République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Kasdi Merbah – Ouargla

FACULTÉ DES SCIENCES ET TECHNOLOGIES ET DES SCIENCES DE LA MATIERE

DEPARTEMENT DE MATHEMATIQUE ET D'INFORMATIQUE

Mémoire

Présenté pour l'obtention du diplôme de :

Magister en Informatique

Option :

Technologie de l'Information et de Communication

Thème

Protocole pour la sécurité des réseaux sans fil peer to peer

Présenté par :

Houda HAFI

Devant la commission d'examen composée de :

Président	**: Pr. Mohammed Benmohamed**	Pr	Université de Constantine
Rapporteur	**: Pr. Azzedine Bilami**	Pr	Université de Batna
Examinateurs	**: Dr. Abdelmadjid Zidani**	MCA	Université de Batna
	Dr. Ahmed Korichi	MCA	Université de Ouargla
	Dr. Ramdane Maameri	MCA	Université de Constantine

Remerciements

Avant tout, je remercie DIEU, le tout puissant, pour la force, la volonté, la santé et la patience qu'il ma donné pour accomplir ce travail.

Je tiens à exprimer mon grand respect et ma gratitude à mon promoteur le professeur : BILAMI Azzedine, pour m'avoir honoré en acceptant de diriger mon mémoire, pour son encadrement de qualité, ses précieuses suggestions scientifiques, sa présence encourageante, et sa patience tout au long de ce travail.

Mes vifs remerciements aux membres de jury, Pr : Benmohammed Mohammed, professeur à l'université Mentouri de Constantine. Dr : Korichi ahmed, maître de conférences à l'université Kasdi Merbah de Ouargla. Dr : Maameri Ramdane, maître de conférences à l'université de Constantine. Dr : Zidani abdelmadjid, maître de conférences à l'université de Batna.

Un grand merci à mon professeur Mr : Korichi ahmed, pour les connaissances qu'il nous a transmis durant la première année magister, j'ai toujours apprécié votre simplicité et votre modestie. Votre compétence scientifique, vos qualités humaines et professionnelles sont pour moi un exemple.

J'adresse mes remerciements infinis à mes parents pour leurs soutiens et leurs encouragements.

Une pensée particulière est adressée à tous mes collègues et amis de Ouargla, particulièrement à : Syhem, Dounia, Meriem, Naima et Halima, les moments passés ensemble ont été très agréables.

Pour terminer, Merci pour tous ceux qui, par leurs remarques et leurs conseils, ont contribué à la réalisation de ce travail.

Pour toutes ces personnes : Merci

Tables des matières

Chapitre 3 : Les solutions au niveau routage

Chapitre 4 : Protocole proposé

Liste des figures

Liste des tableaux

Résumé

RÉSUMÉ : les réseaux Mobile peer-to-peer (MP2P) représentent un paradigme relativement nouveau par rapport à d'autres réseaux sans fil. Dans les dernières années, ils sont devenus très populaires en raison de leur utilisation dans plusieurs applications et essentiellement dans le partage de fichiers sur Internet d'une manière décentralisée. La sécurité des réseaux mobiles P2P représente un sujet de recherche ouvert et un défi majeur au regard de leur vulnérabilité aux différentes attaques, telles que le trou noir, Sybil ... etc. Dans ce mémoire, nous analysons l'attaque black hole (trou noir) dans les réseaux sans fil P2P en utilisant comme protocole de routage AODV. Dans une attaque black hole, un nœud malveillant usurpe l'identité d'un nœud légitime, fabrique des réponses falsifiées avec un numéro de séquence élevé et force ainsi le nœud victime à le choisir comme relai. Nous proposons dans ce travail une solution qui tient compte du comportement de chaque nœud participant au réseau, en apportant une modification au protocole de routage AODV, avec une évaluation des performances par simulation.

MOTS-CLÉS : réseau sans fil, P2P, sécurité, black hole, AODV.

ABSTRACT: Mobile peer-to-peer networking (MP2P) is relatively a new paradigm compared to other wireless networks. In the last years, it has gained in popularity because of its practice in applications such as file sharing over Internet in a decentralized manner. Security of mobile P2P networks represents an open research topic and a main challenge regarding to the vulnerability of these networks and their convenience to different security attacks, such as black-hole, Sybil...etc. In this work, we analyze the black-hole attack in wireless P2P networks using AODV as routing protocol. In a black-hole attack, a malicious node assumes the identity of a legitimate node, by creating forged answers with a higher sequence number, and thus forces the victim node to choose it as relay. We propose a solution based on a modification of the AODV routing protocol, taking into account the behavior of each node participating in the network. Performances of our proposal are evaluated by simulation.

KEYWORDS: mobile wireless network, P2P, security, black-hole, AODV

Introduction Générale

Introduction Générale

Les réseaux sans fil et les réseaux mobiles sont devenus très populaires ces dernières années. Ceci est dû à leurs caractéristiques : Installation simple et facile, absence de câblage, les coûts de matériel ne sont pas prohibitifs, les utilisateurs se déplacent librement au sein de la zone de couverture du réseau. Donc ils peuvent être mis en place facilement et économiquement selon les besoins. Ils offrent en effet un large éventail d'applications, notamment dans les situations géographiques avec des contraintes terrestres telles que les champs de bataille, les applications militaires, et d'autres situations d'urgence et de catastrophe.

Parmi ces réseaux on trouve : les réseaux locaux sans fil, les réseaux de capteurs, les réseaux ad hoc et les réseaux sans fil peer to peer. Dans ce dernier type chaque nœud mobile peut fonctionner non seulement comme un client mais également comme serveur, donc il demande et fournit des services mais différemment du modèle Client/serveur.

Cependant le grand problème de ces réseaux est la sécurité, les travaux de recherche indique que les réseaux sans fil sont plus vulnérables que les réseaux filaires en raison de leurs caractéristiques tels que le milieu ouvert, la topologie dynamique, l'absence d'administration centrale, la coopération distribuée, et la capacité restreinte (en termes de puissance et de calcul). L'utilisation de liaisons sans fil rend ces réseaux plus sujets à des menaces de sécurité physiques que les réseaux câblés, allant de l'écoute passive à l'interférence active. Sans aucune sécurité adéquate, les hôtes mobiles sont facilement capturés, compromis et détournés par des nœuds malveillants. L'adversaire peut écouter et / ou modifier les messages dans le canal de communication, injecter des messages erronés, supprimer des messages, et même passer par d'autres nœuds. Par conséquent, les mécanismes de sécurité dans de tels réseaux sont essentiels pour protéger les données émises par les utilisateurs.

Dans le cadre de ce mémoire, nous nous intéresserons aux problèmes de sécurité dans les réseaux sans fil en général et les réseaux P2P en particulier. Dans cette optique nous projetons de proposer un protocole de sécurité dédié aux P2P.

Ce rapport est organisé en quatre chapitres. Chaque chapitre aborde des points spécifiques.
Il est et structuré comme suit:

Le premier chapitre présente une introduction aux réseaux peer to peer, et les différents concepts liés à ces réseaux. Le deuxième chapitre explique les mécanismes de sécurité fondamentaux utilisés dans la sécurisation des réseaux sans fil. Le troisième chapitre introduit les attaques qui peuvent se produire au niveau routage, en particulier contre le protocole AODV et les solutions présentées dans la littérature. Le dernier chapitre est réservé à la description du protocole proposé et l'analyse de ses performances, le but de ce protocole est la sécurisation des échanges des données dans les réseaux sans fil peer to peer. Enfin, nous terminons le mémoire par une conclusion générale dans laquelle nous résumons l'essentiel de notre travail et nous donnons quelques orientations des travaux futurs.

Chapitre 1 :

Les Réseaux Peer to Peer

1. Introduction :

Ces dernières années, les systèmes pair à pair (en anglais Peer-to-Peer ou bien P2P) sont devenus de plus en plus populaires, cette popularité est dû aux caractéristiques avantageuses offertes par ces systèmes telles que: le passage à l'échelle, tolérance aux panne et le contrôle décentralisé, chaque dispositif peut jouer le rôle de **Serv**eur en offrant ces ressources aux autres nœuds, d'un cli**ent** en consommant les ressources des autres nœuds (Servent). Actuellement, la recherche et l'industrie voient en ce modèle une vraie alternative au modèle classique client-serveur et contribuent à de nombreux travaux dans ce domaine. Parmi ceux ci, nous pouvons citer le partage des fichiers, le calcul distribué ainsi que les espaces collaboratifs.

Dans ce chapitre, nous présentons les réseaux P2P, nous commençons par définir précisément le concept peer-to-peer, puis nous décrivons leurs caractéristiques, par la suite nous parlerons des différentes architectures du modèle et nous présentons pour chacun un exemple, à la fin nous citons quelques domaines d'applications de ces réseaux.

2. Définition:

Au cours de notre étude bibliographique, nous avons trouvé plusieurs définitions pour les systèmes peer-to peer, nous citons ici deux parmi celles proposées par la communauté P2P :

"A distributed network architecture may be called a Peer-to-Peer network (P-to-P, P2P…), if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers . . .). These shared resources are necessary to provide the service and content offered by the network (e.g. file sharing or content workspaces for collaboration). They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource providers as well as resource requestors (Servent-concept)"[Schollmeier, 2001].

"Peer-to-peer systems are distributed systems consisting of interconnected nodes able to self- organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes while maintaining acceptable connectivity and performance,

without requiring the intermediation or support of a global centralized server or authority"[Androutsellis, 2004].

A partir de ces définitions, nous pouvons dire que la technologie P2P ne permet pas seulement le partage des ressources numériques (textes, sons, images, ..), mais elle permet également le partage des capacités de traitement de l'information et de l'espace de stockage (CPU, RAM…), ainsi l'une des spécificités les plus marquantes des réseaux P2P est que les échanges ont lieu directement entre les utilisateurs qui peuvent être contributeurs et consommateurs au même temps.

3. Client/serveur Vs P2P:

Traditionnellement, l'échange de services entre ordinateurs est fondé sur la technique client/serveur, selon cette architecture, il n'y a qu'une seule entité centrale très puissante, le serveur, et plusieurs entités généralement de puissances inférieures, les clients. Le serveur est le seul fournisseur des services aux clients. Un client consomme les services exécutés par le serveur, sans partager aucune de ses propres ressources. L'architecture pair à pair se pose comme une solution de rechange à l'architecture client/serveur en offrant plusieurs avantages par rapport aux autres basés sur le paradigme client/serveur, le tableau 1.1 montre bien les différences entre les deux modèles.

Critère	Modèle Client-serveur	Modèle P2P
Gestion	Supervisé	Auto-organisé
Présence	Permanente	Ad Hoc
Accès aux ressources	Recherche	Découverte
Organisation	Hiérarchique	Distribuée
Mobilité	Statique	Mobile
Disponibilité	Dépendante du serveur	Indépendante des pairs
Nommage	DNS	Indépendant
Modèle de programmation	RPC	Asynchrone

Tableau 1.1: Comparaison des infrastructures client/serveur et P2P

4. Les caractéristiques:

Dans cette section, nous présentons les principales caractéristiques [Raddad AL KING, 2010] que présente le modèle peer-to-peer:

- **Décentralisation:** le fait que chaque nœud gère ses propres ressources permet d'éviter la centralisation de contrôle. Un système P2P peut fonctionner sans avoir aucun besoin d'une administration centralisée ce qui permet d'éviter les goulets d'étranglements et d'augmenter la résistance du système face aux pannes et aux défaillances.
- **Passage à l'échelle:** il s'agit de faire coopérer un grand nombre de nœuds (jusqu'à des milliers ou des millions) pour partager leurs ressources tout en maintenant une bonne performance du système. Cela signifie qu'un système P2P doit offrir des méthodes bien adaptées avec un environnement dans lequel il y a un grand volume de données à partager, un nombre important de messages à échanger entre un grand nombre de nœuds partageant leurs ressources via un réseau largement distribué.
- **L'auto-organisation:** puisque les systèmes P2P sont souvent déployés sur l'Internet, la participation d'un nouveau nœud à un système P2P ne nécessite pas une infrastructure couteuse. Il suffit d'avoir un point d'accès à l'Internet et de connaitre un autre nœud déjà connecté pour se connecter au système. Un système P2P doit être un environnement ouvert ; c'est-a-dire, un utilisateur sur un nœud doit être capable de connecter son nœud au système sans avoir besoin de contacter une personne et sans avoir besoin de passer par une autorité centrale.
- **Autonomie des nœuds:** chaque nœud gère ses ressources d'une façon autonome. Il décide quelle partie de ses données à partager. Il peut se connecter ou/et se déconnecter à n'importe quel moment. Il possède également l'autonomie de gérer sa puissance de calcul et sa capacité de stockage.
- **Hétérogénéité :** à cause de l'autonomie de nœuds possédant des architectures matérielles et/ou logicielles hétérogènes, les systèmes P2P doivent posséder des techniques convenables pour résoudre les problèmes liés a l'hétérogénéité de ressources.
- **Dynamique:** à cause de l'autonomie de nœuds, chaque nœud peut quitter le système à n'importe quel moment ce qui fait disparaitre ses ressources du système. De nouvelles ressources peuvent être ajoutées au système lors de la connexion de nouveaux nœuds. Alors, à cause de l'instabilité de nœuds, les systèmes P2P doivent être capables de gérer un grand nombre de ressources fortement variables. La sortie d'un nœud du système (ou la

panne d'un nœud) ne doit pas mettre le système en échec. Elle doit être tolérée et avoir un "petit" impact sur la performance de tout le système.

5. Classification de l'architecture P2P:

Comme illustré à la figure 1.1, les systèmes informatiques peuvent être classés dans deux catégories différentes : systèmes centralisés et systèmes distribués, ces derniers, à leur tour, sont divisés en deux: le modèle client/serveur et le modèle pair à pair. Le premier peut être soit plat dans le cas où les clients dialoguent seulement avec un seul serveur, soit hiérarchique dans le cas où les clients n'ont de contacts qu'avec les serveurs de plus haut niveau d'eux.

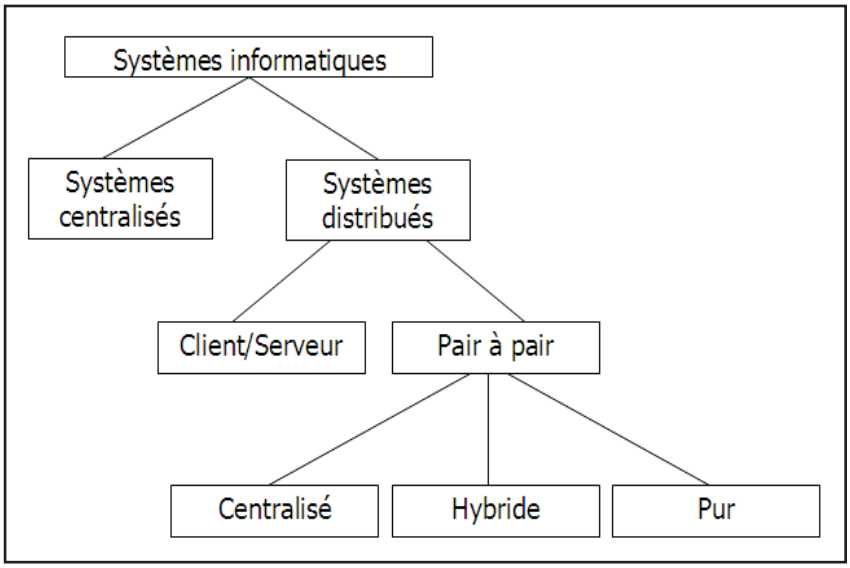

Figure1.1: Taxonomie des systèmes informatiques

Le modèle pair à pair peut être soit centralisé, soit hybride ou bien pur, selon le fonctionnement de recherche du contenu.

5.1. P2P centralisé :

La première génération des systèmes P2P a commencé par le concept de centralisation, il existe un serveur central, joue le rôle d'un annuaire, qui stocke des informations concernant la description des ressources partagées (nom, taille,……) et d'autres sur les utilisateurs qui les hébergent (nom utilisé, IP, nombre de fichiers partagés,……), comme illustré à la figure ci-

dessous les ressources R3, R1, R4, R2 sont hébergées respectivement par les postes P1, P2, P3, P4.

Quand un nœud souhaite partager une ressource, il la déclare au serveur central, celui-ci stocke son adresse IP ainsi un numéro de port donné par le nœud où il pourra être contacté pour un téléchargement.

Lorsqu'un utilisateur recherche un fichier, il envoie une requête au serveur central qui lui répond et transmet la liste des nœuds possédant le fichier demandé, une fois l'utilisateur choisit parmi les réponses indiquées par l'index central celle qui lui convient le mieux, il contacte directement le ou les postes choisis [DIJOUX et al, 2007], Le contenu reste toujours du côté client, ne passant jamais par le serveur.

Le modèle centralisé permet une recherche simple, il est facile à administrer et à contrôler, il est peu coûteux, nécessite qu'un serveur central pour la découverte et une machine hôte pour l'accès à la ressource, cependant il présente plusieurs inconvénients, il n'est pas robuste car la surcharge ou la panne du serveur central rend tout le réseau indisponible, il passe que mal à l'échelle. L'exemple le plus connu reposant sur cette architecture est "Napster".

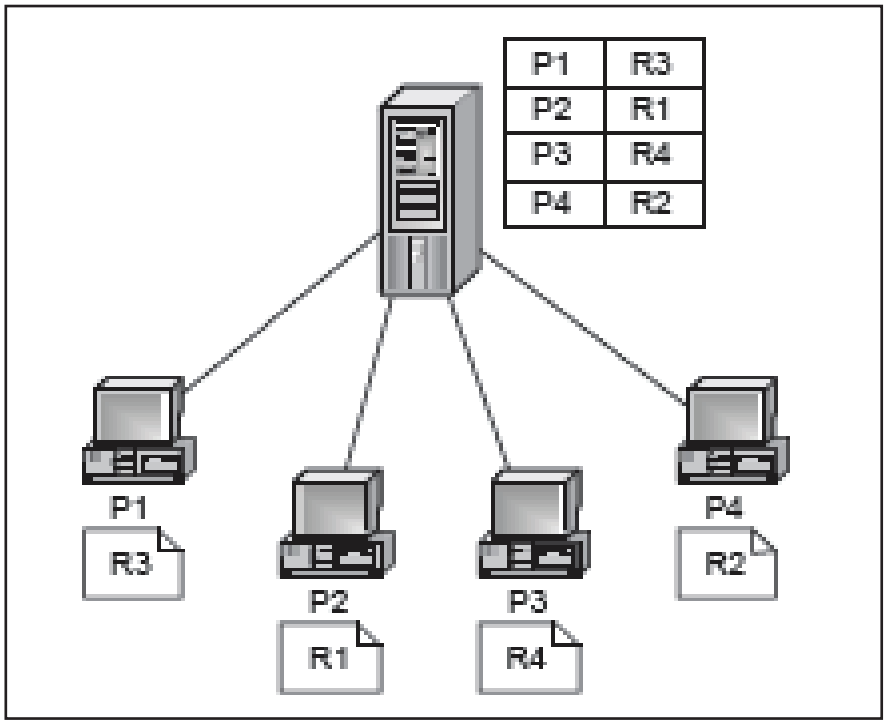

Figure1.2: Modèle centralisé

- **Napster:**

Napster est le premier réseau P2P grand public, adopte une architecture centralisée, cela signifie que les éléments du réseau informent un serveur central des fichiers dont ils disposent et qu'ils le contactent pour obtenir les coordonnées d'un élément possédant les fichiers recherchés. Un utilisateur qui désire partager des fichiers doit exécuter le logiciel Napster sur son ordinateur (client). Etant connecté à Internet, le client établit une connexion TCP avec le serveur central Napster et lui déclare les fichiers partagés, le serveur Napster détient un index avec toutes les adresses IP des clients participants, ainsi qu'une liste de ressources partagées, comme nous avons dit précédemment, le fichier ne transite pas par le serveur central, Le logiciel permet à l'utilisateur de se connecter au pair désiré directement, c'est cette communication directe entre les pairs qui différencie le modèle P2P centralisé du modèle client-serveur classique.

5.2. P2P décentralisé « purs » :

Dans ce type d'architecture, il n'ya plus de serveurs centraux, tous les nœuds sont égaux et jouent le même rôle, ainsi la suppression d'un pair du réseau, n'affecte pas les services offerts (voir figure 1.3), par contre, l'absence d'un serveur central ayant une vue globale sur la localisation des ressources hébergées par les pairs dans le réseau P2P pose le problème suivant: Comment un pair peut découvrir et accéder à une ressource dans ce contexte ? Pour cela deux solutions ont été proposées : la première basée sur l'inondation et la deuxième sur les tables de hachages distribuées "DHT", également connues respectivement sous le nom de réseaux P2P décentralisés non structurés et réseaux P2P décentralisés structurés.

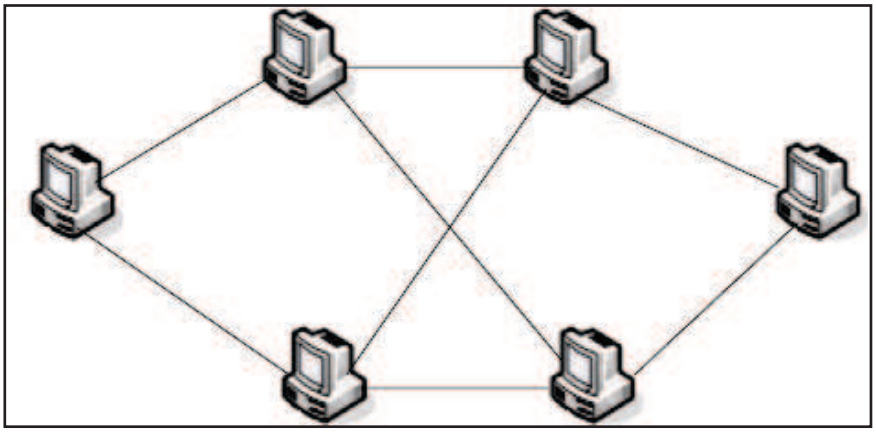

Figure 1.3: le modèle pur

- ✓ **Recherche par inondation:**

Pour découvrir une ressource, une requête sera transmise d'un pair à un autre jusqu'à atteindre le client qui dispose l'objet désiré, afin d'éviter l'inondation du réseau durant un temps trop long, le système associe à chaque requête un temporisateur TTL "Time To Live", la valeur attribuée au TTL est généralement 7. Lorsqu'il arrive à zéro, la requête n'est plus renvoyée.

L'inconvénient majeur de ce mécanisme, provient de l'expiration du TTL avant le parcours de l'intégralité du réseau, ce qui peut aboutir à l'échec d'une recherche bien que l'objet désiré soit disponible sur le réseau P2P, il passe mal à l'échelle et génère une surcharge du réseau par les trames de broadcast diffusées. Cette méthode est utilisée dans le système : Gnutella et FreeNet.

- **Gnutella:**

Gnutella est un réseau P2P décentralisé non structuré, succédant à Napster, crée en mars 2000 par *Justin Frankel* et *Tom Pepper*, constitué d'un ensemble de pairs joignant le réseau d'après certaines règles (voir le tableau 1.2), un client souhaite rejoindre le réseau Gnutella , il commence par identifier les nœuds présents sur le réseau, pour cela il envoie un message d'identification *Ping* à ses voisins, qui l'envoient à leur tour à leurs voisins et ainsi de suite, la retransmission est stoppée lorsque le TTL devient 0. Chaque client recevant un *Ping* répond avec une trame de réponse *Pong* contenant l'adresse IP, le numéro de port, le nombre et la taille des fichiers partagés.

Pour chercher une ressource dans le réseau, le client lance une requête *Query* en spécifiant la vitesse de transfert minimum et les critères de recherche, un servent qui reçoit un descripteur de type *Query* et s'il dispose de la ressource, renvoie une requête de réponse *QueryHit* au voisin qui lui a retransmis la requête, spécifiant son adresse IP et son numéro de port TCP où l'objet peut être téléchargé. La réponse remonte de proche en proche jusqu'au client initiateur. Ce dernier sélectionne ensuite les fichiers à télécharger en envoyant directement une requête de téléchargement au client possédant le fichier. Un message *Push* est utilisé si les données sont derrière un firewall.

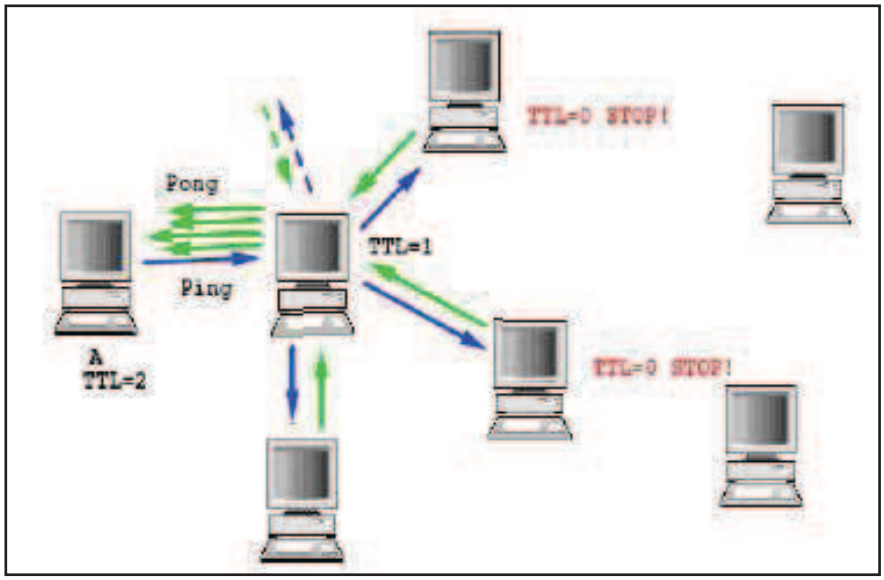

Figure 1.4: fonctionnement de Gnutella

Parmi les applications qui implémentent le protocole Gnutella, on trouve Limewire, BearShare, Gnucleos ou Phex, de plus les échanges peuvent être effectués quelque soit la plate forme (Windows, Linux/Unix, Macintosh, etc.) du client.

Types	Description	Information
Ping	Annonce la disponibilité, et lance une recherche de pair	Vide
Pong	Réponse à un ping	IP + N de port + Nombre et taille de fichiers partagés
Query	Requête	Bande passante minimum demandée+ critères de recherche
QueryHit	Réponse à query si on possède la ressource	IP+ N de port+ Bande Passante+ Nombre de réponses+descripteur
Push	Demande de téléchargement pour les pairs derrière un firewall	IP + index du fichier demandé+Adresse IP+N de port où envoyer le fichier

Tableau 1.2: cinq descripteurs utilisés dans le protocole Gnutella

- ✓ **Les tables de hachages distribuées:**

Une table de hachage distribué "DHT" est une structure de données permettant la découverte et la localisation efficace des ressources via le biais des clés, chaque pair est

responsable d'une partie de la table de hachage, cette dernière utilise une fonction de hachage sécurisée telles que SHA-1 et MD5, Le hachage de l'adresse IP donne l'identifiant d'un utilisateur tandis que celui du nom d'un fichier donne la clé. Il s'agit ensuite de stocker de manière distribuée les couples (clé, identifiant) sur les nœuds du réseau pour qu'à chaque ressource du réseau soit associée l'adresse de l'utilisateur possédant la ressource. De la redondance dans le stockage est également introduite afin que le départ d'un nœud du système n'engendre pas la perte des méta-données qu'il stocke et ne rende impossible l'accès à ces données. La fonction de hachage utilisée assure que pour deux ressources différentes les clés générées le seront aussi.

La figure 5 montre le processus de découverte des ressources en utilisant une DHT: le pair P1 héberge la partie de la table correspondant à la ressource R1 qui possède le même identifiant numérique, à savoir 1, P2 héberge l'entrée de la table pour R2, et ainsi de suite. On remarque que pour chaque pair hébergeant une partie de la table de hachage, ne possède qu'une référence vers une ressource et non la ressource elle-même. À titre d'exemple le pair P4 héberge physiquement la ressource R2 et possède un pointeur vers P3 qui héberge R4, la ressource possédant le même identifiant. Donc pour découvrir une ressource particulière, revient à découvrir le pair d'identifiant le plus proche de celle-ci, ce dernier contient un pointeur vers le pair hébergeant physiquement la ressource.

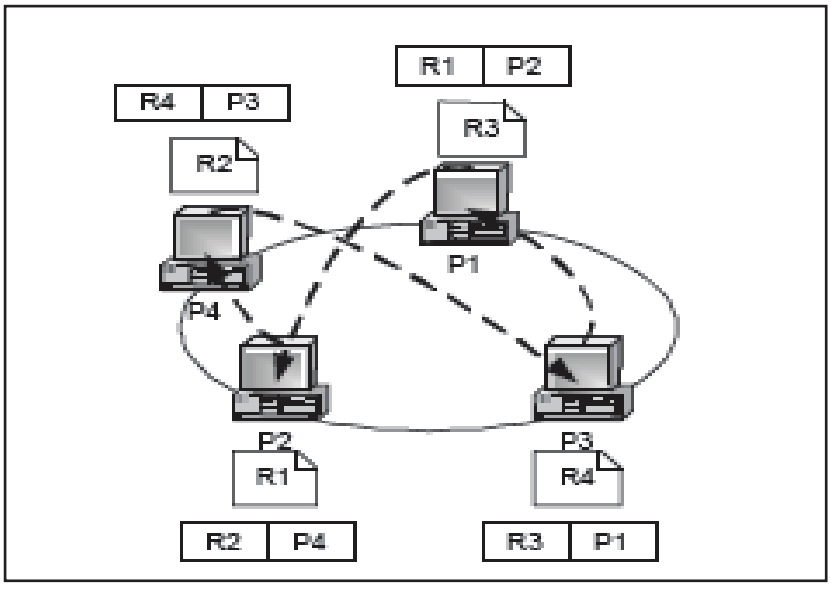

Figure 1.5: la découverte de ressources par une DHT

On confère aux DHT les propriétés [Pujolle G, 2008] suivantes:

- **Fiabilité:** L'utilisation d'un algorithme de découverte et de routage permet, pour une clé donnée, de déterminer le pair d'identifiant le plus proche. Dans des conditions statiques, une réponse négative à une requête signifie que la ressource requise n'est pas disponible dans la communauté.
- **Performance:** Dans des conditions normales de fonctionnement, le nombre de sauts nécessaires est limité. Par exemple, dans une communauté de 10^6 pairs utilisant une base d'identifiants hexadécimale, la longueur moyenne d'une requête avoisine cinq sauts.
- **Passage à l'échelle:** Deux caractéristiques confèrent aux DHT de bonnes performances de passage à l'échelle. La première est liée au nombre moyen de sauts nécessaires au routage des requêtes, qui reste petit, même dans le cas de communautés comptant un grand nombre de participants. La seconde est relative aux tables de routage, qui restent elles aussi d'une taille raisonnable en regard du nombre de participants.
- **Tolérance aux fautes:** Du fait de l'absence de centralisation, qui exclut tout point central, les DHT présentent une bonne tolérance aux suppressions aléatoires de nœuds. Les requêtes peuvent être acheminées même si une partie des nœuds disparaît. Par contre, chaque nœud racine d'une ressource particulière s'apparente à un point central. Des mécanismes de redondance sont souvent mis en place pour éviter l'inaccessibilité d'une ressource présente dans une DHT. Exemples de DHT actuels: Pastry, Chord, CAN (Content adressable Network), Kademlia.

Nous détaillons ici le fonctionnement du réseau structuré Chord

Chord

Chord est un protocole de recherche distribué, qui repose sur une structure en anneau, représentant 2^m valeurs (m est la taille d'un identifiant). Typiquement, m vaut 160, l'identifiant d'un pair est un condensat SHA-1 réalisé à partir de son adresse IP et l'identifiant d'une ressource est le condensat SHA-1 de la donnée stockée. Les ressources sont réparties sur les différents nœuds de l'anneau. Une clé K est attribuée au premier nœud immédiatement supérieur (ou égal) à K, ce nœud est dit successeur de K et est noté successeur(K).

Pour la recherche d'une ressource dans Chord, le nœud demandeur doit obtenir la clé K de la ressource en hachant son nom, et en contactant successeur(K), pour cela une méthode simple mais peu efficace existe, chaque nœud conserve l'adresse IP de son successeur réel dans le cercle (successeur de 8 est 14, successeur de 14 est 21,….etc). Le nœud demandeur peut alors envoyer un paquet de requête contenant son adresse IP et la clé de la donnée qu'il recherche. Le paquet circule dans le cercle jusqu'à arriver au successeur de la clé recherchée. Ce nœud regarde alors s'il possède des informations correspondant à la clé, et les renvoie directement au nœud demandeur vu qu'il a son adresse IP, le résultat suit le chemin dans le sens inverse.

Dans le but d'accélérer les recherches, une table de routage est utilisée appelée "*finger table*". Chaque nœud *p* dispose d'une table de repérage à *m* entrées, L'ieme entrée (1<=i<=m) contient l'identifiant du nœud succédant *p* par au moins 2^{i-1}. i.e successeur ($p+ 2^{i-1}$).

Pour rendre le schéma plus lisible, dans la figure 1.6, la 1ére entrée de la table finger du nœud 8 est le successeur de ($8+2^{1-1}$) qui est le nœud 14, la 2éme entrée correspond au successeur de ($8+2^{2-1}$) à savoir 14, et ainsi de suite pour les autres nœuds.

Quand un nœud *p* reçoit une requête de recherche d'une clé K, il vérifie d'abord si elle existe localement. Si oui il renvoie la valeur associée sinon, il recherche dans la table de routage un nœud avec la plus grande valeur inférieure ou égale à la clé cherchée, puis il transmet la requête au nœud sélectionné et appliquer récursivement.

La figure 1.7 montre la recherche de la clé 54 depuis le nœud 8, tout d'abord le nœud 8 consulte sa table pour déterminer le plus grand prédécesseur de 54, il s'agit du nœud 42, par conséquent la requête est transmise directement vers le nœud 42 qui en prend en charge.

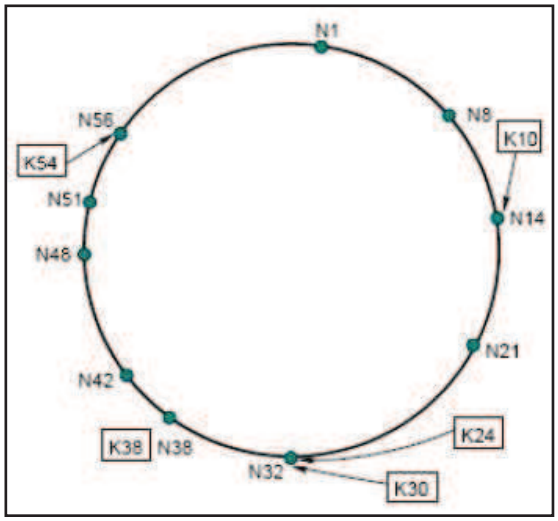

Figure 1.6: Espace d'adressage de Chord

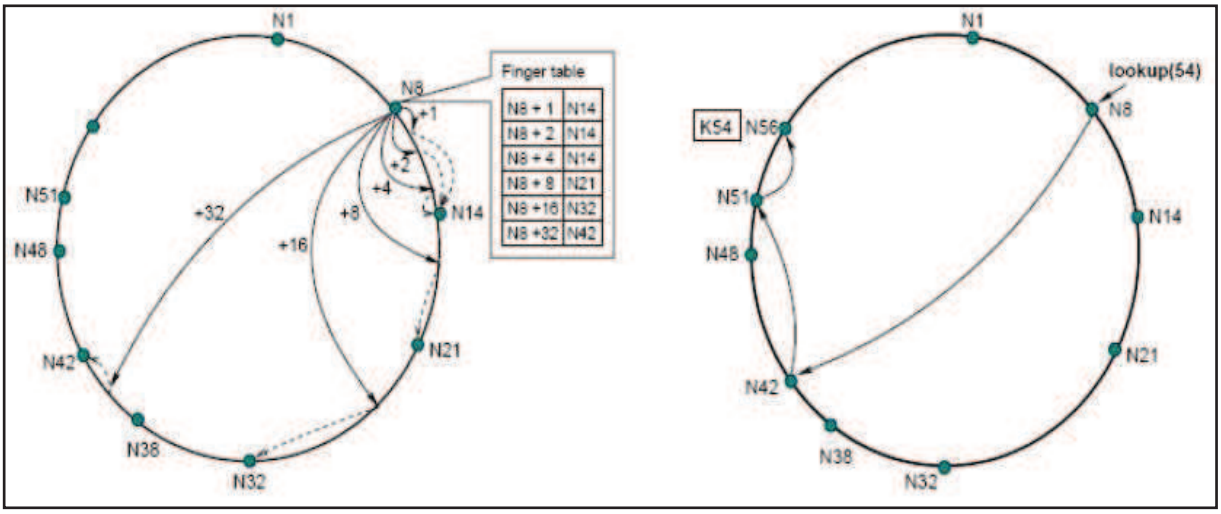

Figure 1.7: Routage d'un objet dans Chord

5.3. P2P hybrides :

Cette solution combine les caractéristiques des modèles centralisés et décentralisés, la décentralisation assure l'extensibilité, la tolérance aux pannes et le passage à l'échelle, cependant la centralisation partielle implique quelques nœuds qui contiennent des données importantes pour le système.

Certains nœuds jouent un rôle particulier, et sont appelés les "Super-pairs", ils disposent d'une forte capacité de calcul et d'une large bande passante, chaque super-pair fait office d'un serveur local pour un groupe de nœuds, comme la montre la figure 1.8, le super-pair P1 gère le groupe des nœuds P2, P3, P4.

Nombreuses applications sont construites selon ce modèle, par exemple : Kazaa, BitTorrent

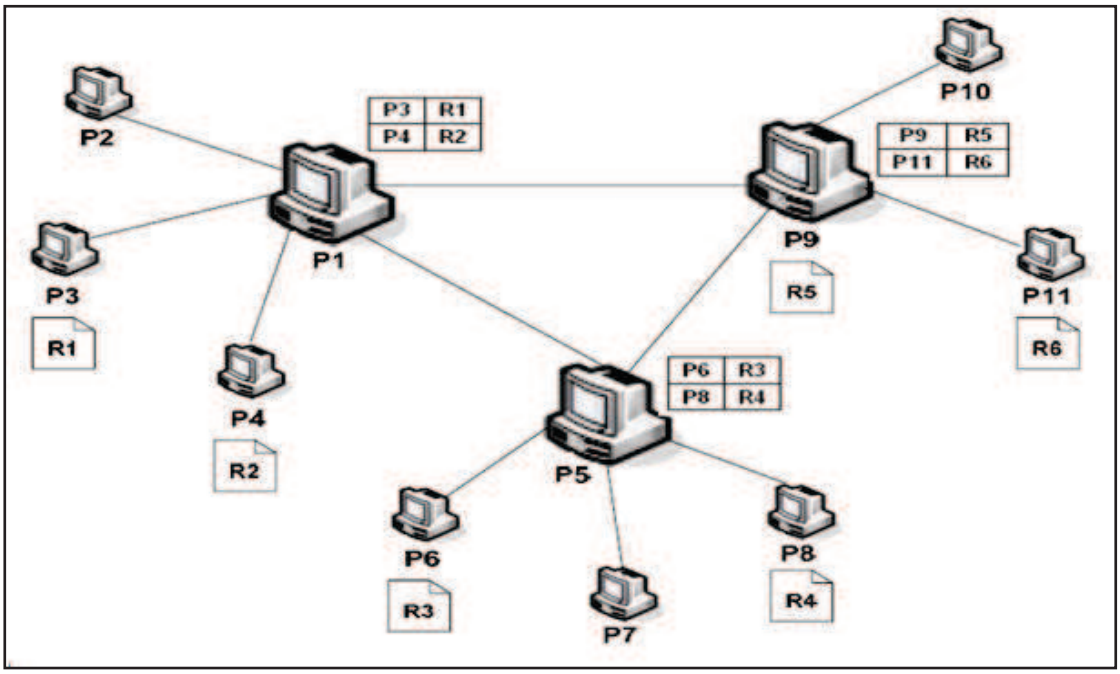

Figure 1.8: Modèle hybride

Remarque:

Certaines classifications des architectures P2P considèrent les modèles hybrides et centralisés comme identiques : le modèle centralisé est un modèle hybride avec un seul super-pair.

- **KaZaA:**

Les pairs disposant d'une connexion rapide, une grande capacité de disque, et une forte puissance de traitement sont immédiatement désignés comme des super-pairs, ces derniers en plus de leur rôle spécifique à savoir l'hébergement de la liste des fichiers partagés par les clients peuvent aussi partager et télécharger des fichiers comme les autres pairs ordinaires. Pour pouvoir se connecter au réseau, chaque nœud contacte un des super-pairs actifs, et lui envoie la liste des fichiers qu'il désire partager afin qu'ils y soient indexés. Il lui envoie également des requêtes sur des fichiers qu'il veut obtenir. Le super-pair lui fournit directement l'adresse IP d'un membre disposant de la donnée voulue soit en cherchant dans l'index local (la liste des ressources hébergées par les postes qu'il gère), soit en diffusant la requête aux autres super-pairs, les données restent toujours distribuées sur les pairs et les échanges se font directement d'un pair à un autre via le protocole HTTP.

6. Champs d'application des réseaux P2P :

Le terme « peer-to-peer » peut se confondre avec les systèmes de partage de fichiers tels que: eMule, Bittorrent, … mais, en réalité, les réseaux P2P peuvent être utilisés à de nombreuses fins :

6.1. Partage de fichiers:

Le partage de fichiers constitue l'application la plus répandue actuellement du réseau P2P. Les internautes peuvent partager ses fichiers ainsi télécharger les fichiers des autres via des logiciels le plus souvent gratuits comme BitTorrent et eMule.

6.2. Le calcul distribué "Grid Computing":

Consiste à utiliser les machines connectées à l'internet pour faire des petites portions d'un grand calcul, en exploitant les ressources (CPU, mémoire….) inutilisées des PC en réseau en vue d'accroitre le potentiel réseau, comme exemple le projet: SETI@home (Search for Extra Terrestrial Intelligence).

6.3. Système de sauvegarde distribué :

Le système de sauvegarde réparti s'appuie sur la coopération des pairs à mettre à disposition leurs espaces de disques inutilisés, un utilisateur peut sauvegarder d'une manière transparente et sécurisée une copie de ses données dans les autres pairs du réseau P2P afin de les récupérer en cas de perte ou de dégâts occasionnés aux données locales. Nous pouvons citer des projets comme Wuala, DisPairSe, OceanStore…et

6.4. Des programmes de messagerie:

De nos jours, il existe des services de messagerie électroniques basé sur le principe P2P, les utilisateurs peuvent envoyer et recevoir des e-mails d'une façon sécurisée, pas besoin d'un serveur central pour stocker temporairement les messages ce qui assure la confidentialité des correspondants, un système d'authentification et de cryptage est utilisé afin de protéger leur contenus, un bon exemple est Jeftel.com. Des logiciels de messagerie instantanée peuvent aussi être vus comme P2P, on citera des exemples comme ICQ, AIM qui certes utilisent un

serveur mais juste pour la résolution d'adresses (les adresses utilisées peuvent être des alternatives aux adresses IP).

6.5. Streaming P2P:

Le streaming P2P est le fait de regarder en direct des flux produits et/ou relayes par d'autres pairs du réseau afin d'éviter ou du moins diminuer la congestion qui pourrait se produire sur les serveurs de téléchargement, tout se déroule entre les personnes qui veulent accéder au fichier, Swarmplayer, qui permet de lire des vidéos en streaming en utilisant Bittorrent, s'annonce comme une vraie révolution dans le domaine.

6.6. Plateformes de développement:

La plupart des logiciels P2P ont été développés de manière spécifique sans référence à des standards propres au p2p, dans le but d'uniformiser les réseaux P2P, des plates-formes sont implémentées pour servir de base au développement des applications P2P, Elles assurent les fonctionnalités de base: gestion des pairs, attribution des identifiants, découverte des ressources, communication entre pairs, sécurité. On peut citer la plate-forme JXTA développé par Sun MicroSystems.

7. Conclusion

Les systèmes P2P représentent un domaine de recherche très actif grâce à leurs actuelles utilisations et leurs applications, ils sont classés en trois grandes familles: les réseaux centralisés, les réseaux purs et les réseaux hybrides, chacun de ces réseaux possède des caractéristiques bien différentes, ils offrent beaucoup d'avantages indéniables tels que: la répartition de la charge et la résistance aux pannes, mais aussi quelques inconvénients liés surtout à l'aspect de sécurité des données circulant dans le système.

Dans le chapitre suivant, nous étudierons les différents problèmes de sécurité qu'on peut rencontrer en utilisant les réseaux sans fil Peer-to-Peer.

Chapitre II :

La Sécurité dans les réseaux Sans fil Peer to Peer

1. Introduction:

Le problème de sécurité ne se posait pas lorsque les entreprises et les universités n'avaient qu'un seul centre d'ordinateurs. Il suffisait de placer un garde à l'entrée de la salle. Maintenant avec la venue des réseaux, l'emploi des liaisons satellites et l'utilisation de l'Internet, la situation a radicalement changé, dans la mesure où un même message transite par plusieurs machines avant d'atteindre son destinataire. A chaque étape, il peut être copié, perdu ou altéré. La sécurité est donc un élément primordial dans tout système, en particulier les systèmes de liaison radio qui sont par leur nature très vulnérable aux attaques.

Dans ce chapitre, nous présentons les propriétés de sécurité à garantir, puis nous étudions les attaques qui peuvent survenir dans les différentes topologies sans fil P2P et les contre-mesures éventuelles contre ces attaques.

2. Les enjeux de sécurité dans un réseau sans fil P2P :

Quelle que soit la nature du réseau, sa politique de sécurité vise à satisfaire les propriétés suivantes [Liorens et al, 2003]:

2.1. Confidentialité des données:

La confidentialité des données est une exigence importante dans la sécurité du réseau. Elle permet d'assurer qu'une communication de données reste privée entre un émetteur et un destinataire. La cryptographie ou le chiffrement des données est la seule solution fiable pour sécuriser le transfert des données.

2.2. Intégrité des données:

L'intégrité peut être vue comme un ensemble de mesures garantissant la protection des données contre les modifications et altérations non autorisées. On peut distinguer les altérations accidentelles dues à l'environnement dur de communication, par exemple une mauvaise couverture des ondes, et les altérations volontaires d'un attaquant. Cela concerne aussi la protection contre l'injection ou la modification des paquets.

2.3. Disponibilité:

La disponibilité est un service réseau qui donne une assurance aux entités autorisées d'accéder aux ressources réseaux avec une qualité de service adéquate.

2.4. Authentification des pairs:

L'authentification peut être définie comme le processus de prouver une identité revendiquée. La confidentialité, l'intégrité des données, et la non-répudiation dépendent tous de l'authentification appropriée. Un système sans cette fonctionnalité ne pouvait pas fournir les objectifs de sécurité mentionnés de manière satisfaisante. Elle est la pierre angulaire d'un réseau sans fil P2P sécurisé.

2.5. Non-répudiation:

Mécanisme permettant de garantir qu'un message a bien été envoyé par un émetteur et reçu par un destinataire, c'est-à-dire aucun des correspondants ne pourra nier l'envoi ou la réception du message.

3. Les modèles d'attaques:

Les réseaux sans fil P2P sont susceptibles aux différentes attaques qui tentent d'exploiter ses différentes vulnérabilités pour mener des manipulations malicieuses. Les attaques peuvent se produire de différentes manières. La classification de ces attaques dépend de plusieurs paramètres :

- **Actif vs Passif:** Une attaque passive obtient les données échangées dans le réseau sans perturber le fonctionnement de la communication, tandis qu'une attaque active implique l'interruption d'information, la modification, ou la fabrication, ce qui perturbe le fonctionnement normal du réseau sans fil P2P [Bing Wu et al, 2006]**.**
- **Interne vs Externe:** Les attaques peuvent aussi être classées en deux catégories, à savoir les attaques externes et les attaques internes, selon le domaine de l'attaque. Les attaques externes sont effectuées par des nœuds qui n'appartiennent pas au domaine du réseau. Les attaques internes sont entreprises par des nœuds compromis, qui font partie du réseau. Les attaques internes sont plus graves par rapport aux

attaques externes car l'attaquant connaît des informations précieuses et secrètes, et possède un accès privilégié au réseau [Bing Wu et al, 2006].

- **Individuelle vs Distribuée:** 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. 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.

4. Notions et mécanismes de bases de sécurité:

Pour s'assurer que seules les personnes autorisées ont accès à l'information et que le service est rendu correctement, un ensemble de mécanismes doivent être mis en place. Parmi ces mécanismes, on peut citer:

4.1. La cryptographie:

La cryptographie est une science qui étudie les outils servant à sécuriser les informations. De tout temps, l'art du chiffrement-déchiffrement a été employé.

Le chiffrement et le déchiffrement des données sont effectués par des algorithmes cryptographiques. Ces algorithmes reposent généralement sur des problèmes mathématiques complexes, difficiles à résoudre, tels que la factorisation des nombres premiers, les logarithmes discrets, etc. [Liorens et al, 2003]

Il existe deux grands types d'algorithmes cryptographiques, ceux dits symétrique ou à clé secrète et ceux dits asymétrique ou à clé publique :

▶ *Algorithmes symétrique:*

Dans les algorithmes symétriques, les clés de cryptage/décryptage sont connues par Alice et Bob. Par exemple, la clé de chiffrement est partagée, et la clé de déchiffrement est facilement calculée à partir d'elle. Dans la plupart des cas, la clé de cryptage et la clé de décryptage sont les mêmes [W.Trappe et al, 2005]. En générale, les algorithmes symétriques sont très rapides, cependant, La distribution des clés constitue le point de faiblesse de ces

algorithmes, les parties qui établissent la session devant posséder la même clé. Pour surmonter cette faiblesse, des protocoles d'échange de clés ont été élaborés, notamment le protocole Diffie-Hellman.

Parmi les algorithmes symétriques les plus connus, on trouve: le DES (*Data EncryptionStandard*), et son successeur l'AES *(Advanced Encryption Standard*).

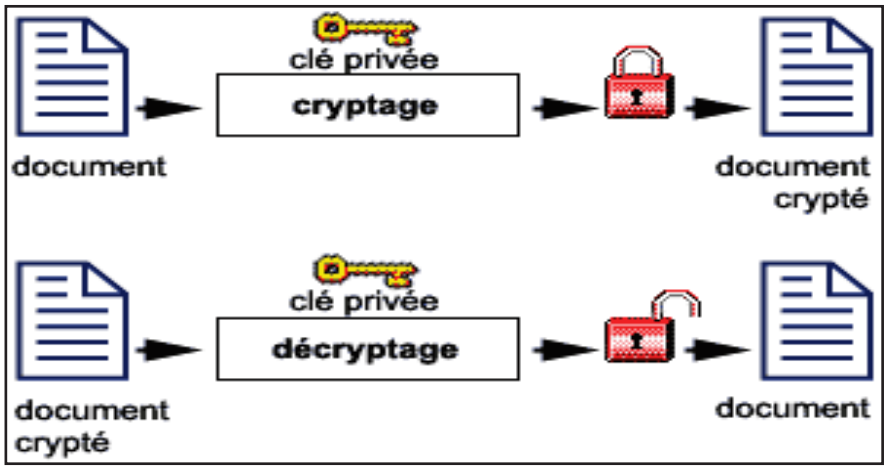

Figure 2.1: Cryptographie symétrique

► *Algorithmes asymétriques:*

L'idée de base de la cryptographie asymétrique est les clés publiques. La clé de chaque personne est séparée en deux parties: une clé publique pour le cryptage disponible pour tout le monde et une clé secrète de déchiffrement qui est gardée secrète par le propriétaire. Bien entendu, il est impossible de déduire la clé privée à partir de la clé publique [Hans et al, 2007].

Les techniques de clé publique sont très utilisées de nos jours pour une distribution sécurisée de clés secrètes, ainsi certaines formes d'authentification et la non-répudiation exigent également les méthodes de la clé publique, telles que les signatures numériques. Le nombre des algorithmes de chiffrement asymétrique est important, les plus connus sont RSA (Rivest Shamir Adleman), les courbes elliptiques, Pohlig-Hellman, Rabin et ElGamal.

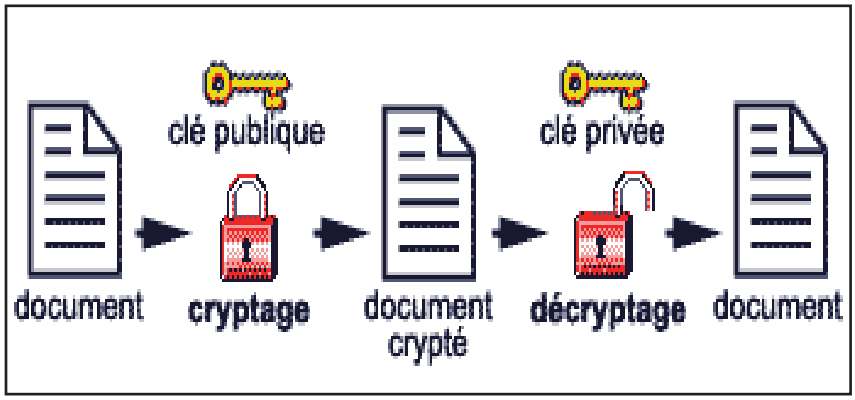

Figure 2.2: Cryptographie asymétrique

4.2. Fonction de hachage:

Une fonction de hachage prend en entrée un texte de longueur arbitraire et renvoie en sortie un texte de taille fixe appelé le condensât ou bien l'empreinte.

Ces fonctions sont utilisées, par exemple, pour la vérification de l'intégrité des messages transmis. On crée pour cela une empreinte du message à transmettre, puis on transmet le message et l'empreinte. À la réception du message, on calcule l'empreinte du message reçu et on la compare à l'empreinte initiale. Si les deux empreintes correspondent, c'est que le message n'a pu être modifié. Les principales fonctions de hachage sont MD5, et SHA-x (x = 1, 256, 384, 512).

Une fonction de hachage H doit avoir les propriétés suivantes [Tiwari, 2010]:

- ✓ Résistance aux attaques par pré-image : étant donné un résultat de hachage h, il est impossible de construire un message m tel que: h = H(m).
- ✓ Résistance aux attaques de seconde pré-image : étant donné un message quelconque m1, il est impossible de trouver un autre message m2 (différent de m1) tel que: H (m1)=H (m2).
- ✓ Résistance aux attaques par collision : il est impossible de trouver deux messages distincts possédant un même haché.

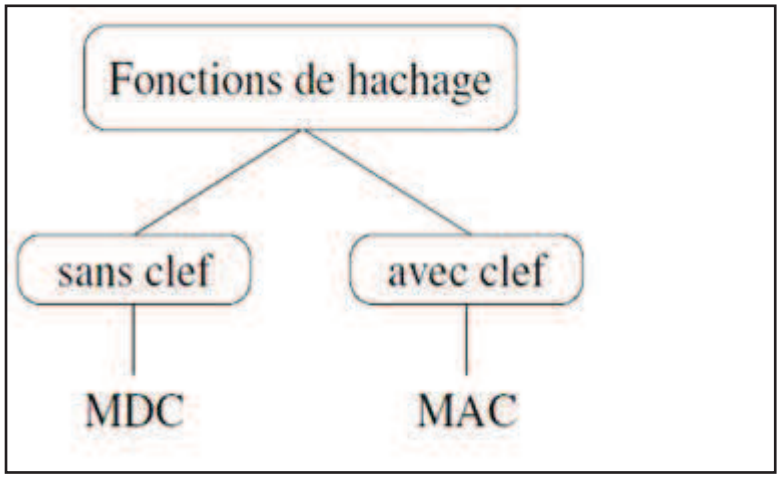

Figure 2.3: Classification des fonctions de hachage

Deux classes de fonctions de hachage:

- ✓ Sans clé: servent à garantir uniquement l'intégrité du message (modification detection code ou MDC en anglais). Exemples: MD4, MD5.
- ✓ Avec clé: garantissent à la fois l'intégrité du message et l'identité de l'expéditeur (message authentication code ou MAC en anglais).

4.3. MAC

Mécanisme basé sur une fonction de hachage avec clé, il consiste à calculer une empreinte à partir d'un message et d'une clé privée pour authentifier l'origine des données et vérifier l'intégrité des données.

Le principe est relativement simple: l'émetteur calcule le MAC et l'émet avec le message. Le récepteur sépare le message du MAC, il lui suffit de calculer de son côté le MAC sur le même message à l'aide de la clé symétrique partagée et de le comparer avec le MAC reçu. Si les deux MAC diffèrent, soit l'émetteur ne possède pas la bonne clé, soit le message a subi des modifications en chemin.

Un MAC assure l'intégrité d'un message mais pas la non-répudiation puisque l'émetteur et le récepteur possèdent la même clef (principe du chiffrement symétrique). L'émetteur peut donc nier avoir signé les données puisqu'il n'est pas le seul à pouvoir le faire. Le MAC est très utile (rapide et efficace) à condition d'avoir mis en place un mécanisme sûr d'échange de la clef secrète entre les différents protagonistes.

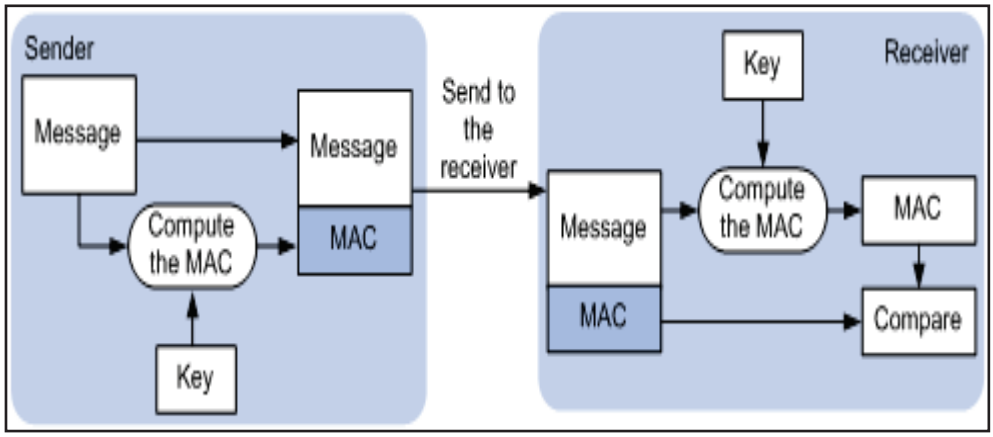

Figure 2.4: Génération du MAC (chiffrement symétrique)

4.4. Signature Electronique:

La signature numérique est un système assurant l'intégrité, l'authentification et la non-répudiation des données, il repose sur la cryptographie asymétrique. L'émetteur crée une empreinte de son message, chiffre l'empreinte avec sa clé privé puis il envoie le message et la signature, le récepteur utilise la clé publique de l'émetteur pour déchiffrer la signature, il recalcule l'empreinte du message et la compare avec celle reçue. Si le condensât nouvellement calculé égale au condensât accompagnant le message alors le message n'a pas été modifié et il est prouvé authentique.

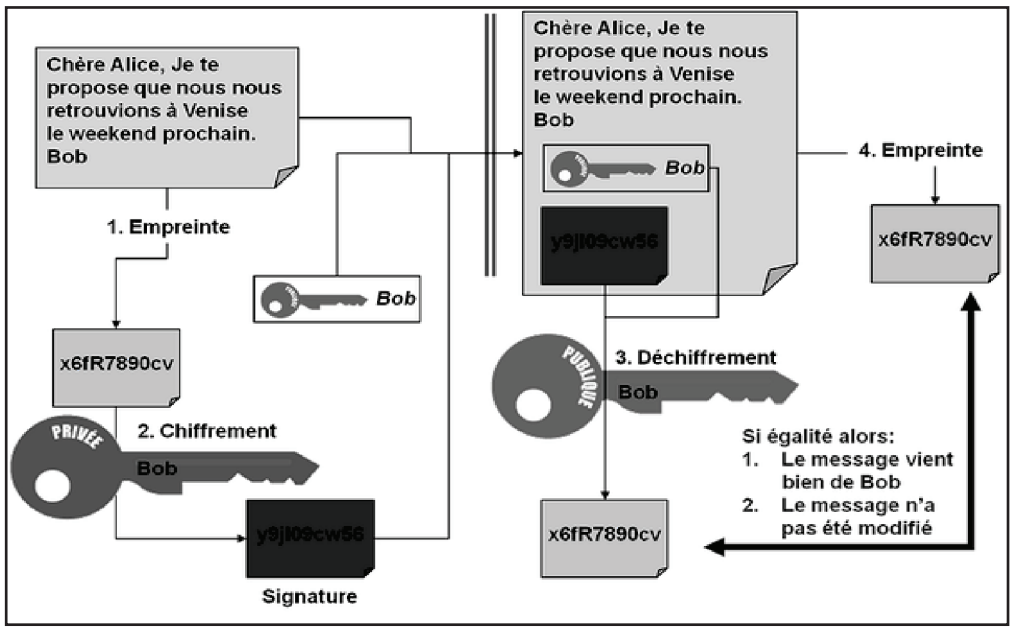

Figure 2.5: processus de création d'une signature numérique

4.5. Certificat:

Le problème de clés publiques est qu'un pirate peut arriver à remplacer la clé publique d'un utilisateur X par la sienne, par exemple sur un annuaire. Et toutes les personnes croyant encrypter pour l'utilisateur X encrypterons pour le pirate. Les systèmes à clés publiques ne garantissent donc pas que la clé est bien celle de l'utilisateur à qui elle est censée appartenir.

Les certificats électroniques servent à cela: ils permettent de lier de façon sûre une clé publique à une entité (utilisateur, serveur, etc.). Un certificat contient les informations suivantes: un numéro de série, une clé publique, l'identifiant du propriétaire de la clé publique, la date de validité (date de début et date de fin de validité), l'identifiant de l'autorité de certification émettrice du certificat, la signature du certificat à l'aide de la clé privée de l'autorité de certification. Toutes ces informations sont signées et délivrées par un tiers de confiance appelé: autorité de certification (souvent notée **CA** pour *Certification Authority*).

La clé publique ayant été préalablement largement diffusée afin de permettre aux utilisateurs de vérifier la signature avec la clé publique de l'autorité de certification.

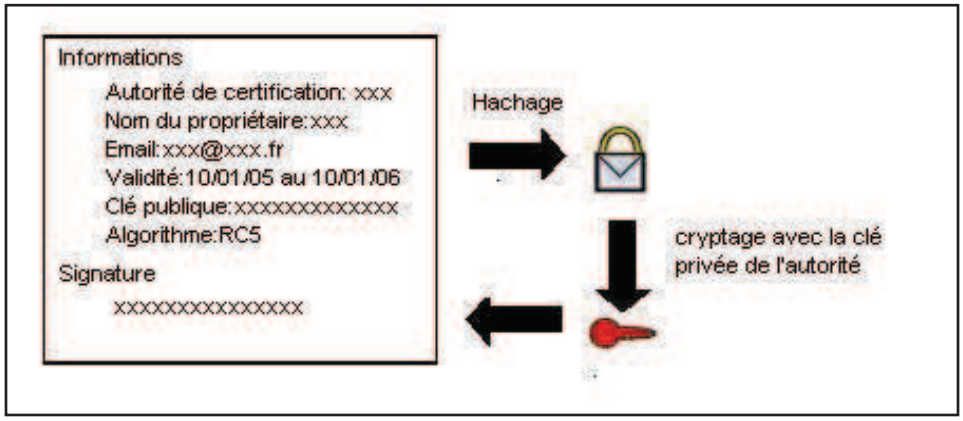

Figure 2.6: création d'un certificat numérique

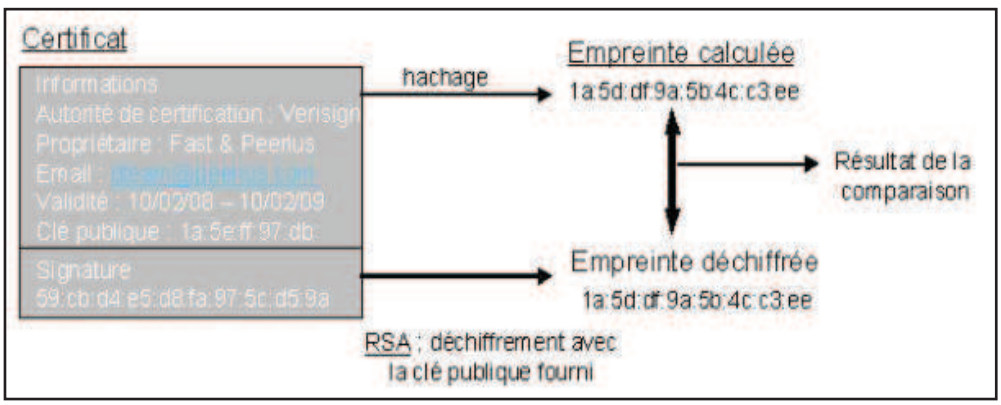

Figure 2.7: vérification du certificat

4.6. PKI:

La norme X.509 **[**R.Housley et al, 1999**]** définit une PKI (Public Key Infrastructure) comme un ensemble de matériels, de logiciels, de personnes et de procédures requis pour créer, contrôler, stocker, délivrer et révoquer les certificats basés sur la cryptographie à clés publiques.

Une PKI ou IGC (Infrastructure de Gestion de Clefs) comprend les éléments suivants:

- *Propriétaires de certificats*: L'utilisateur ou le système qui est le sujet d'un certificat.
- *Autorité d'enregistrement(AE):* Entité chargée de recevoir la demande de l'utilisateur, elle vérifie son identité et valide sa demande s'il est apte à recevoir un certificat. Par la suite, elle passe cette demande entre les mains de l'autorité de certification qui va appliquer les procédures. Elle peut optionnellement créer des clés.
- *Autorité de certification(AC):* Son principal rôle est de générer un certificat signé pour l'utilisateur. De plus, elle est responsable à suivre le statut des certificats et de publier la liste de révocation (CRL) qui contient les numéros de série des certificats révoqués et la signature de l'AC.
- *Dépôts:* Son rôle est de stocker les certificats révoqués et par la même occasion, les certificats en cours de validité afin d'avoir un accès rapide à ces certificats. Un annuaire LDAP est une très bonne autorité de dépôt.

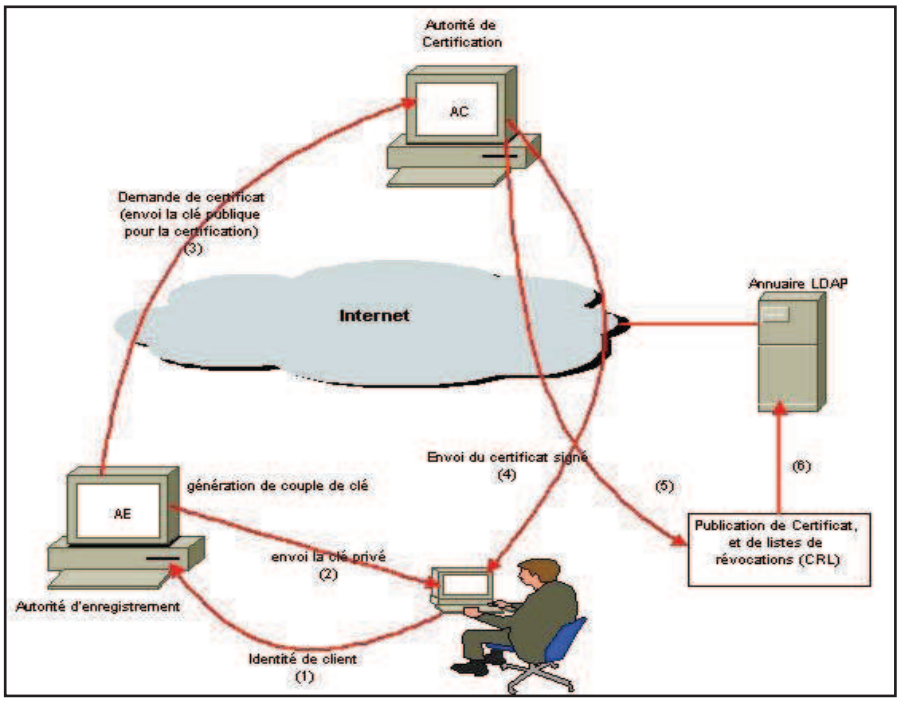

Figure 2.8: Organisation d'une PKI

5. Les attaques possibles dans les réseaux sans fil P2P

Dans cette section, nous allons parler des actions malveillantes les plus répandues dans ces réseaux. Tout d'abord, nous allons examiner les vulnérabilités ou les attaques dans les réseaux sans fil en général, ensuite nous examinerons les attaques spécifiques pour les réseaux P2P.

✓ **Attaque Man-in-the-Middle**

Une situation dans laquelle un attaquant peut lire, insérer et modifier des messages entre deux pairs sans qu'ils sachent que le lien entre eux a été compromis, en modifiant l'adresse IP et le numéro de port dans le message "QueryHit" (réponse à une requête Query, voir chapitre 1), le nœud malveillant peut tromper le pair demandeur, et le laisser connecter et télécharger un contenu altéré du nœud malicieux [John et al, 2006].

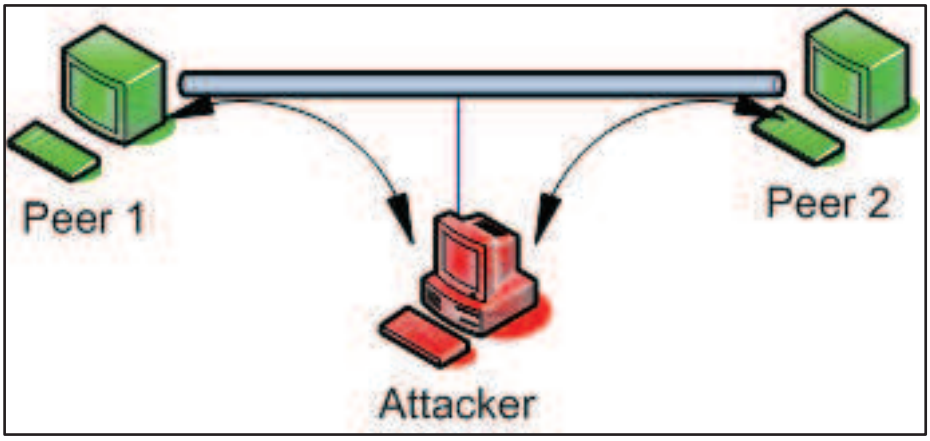

Figure 2.9:l'attaque Man-in-the middle

❖ **Solution:**

Sans serveur ou autorité centrale, l'attaque MitM revient à compromettre l'intégrité des nœuds dans le réseau. De nombreux réseaux utilisent des super-nœuds ou autorité de certification, comme un outil de prévention d'autres formes d'attaque. La prévention la plus largement acceptée de l'altération d'information est l'utilisation de signatures numériques. Ces signatures sont basées sur la cryptographie à clé publique, et permettent d'assurer l'intégrité d'un document. Une telle technologie a été utilisée dans l'e-mail pour de nombreuses années, pour fournir l'authentification d'un message. Cette même méthode peut être utilisée pour détecter si un message a été modifié dans un réseau sans fil P2P. En attachant simplement une signature numérique à la fin du message, et le nœud de réception (ou de n'importe quel nœud

entre les deux) sera en mesure de vérifier que le message est inchangé et il provient de la vraie source.

Après avoir empêché la modification et l'insertion de faux messages, nous avons besoin maintenant d'empêcher l'attaquant MitM d'être capable de lire les messages. Encore une fois la solution vient de la cryptographie à clé publique. Avec la fixation de la signature, l'émetteur peut également chiffrer le message avec la clé publique du destinataire: ce qui rend improbable tout nœud autre que la destination de lire le contenu réel du message [Marling et al, 2006].

- ✓ **Attaque DOS**

Une attaque par déni de service (denial of service en anglais ou DOS) est une attaque qui vise à limiter ou stopper complètement le bon fonctionnement d'une ressource réseau, afin qu'elle ne puisse plus ou mal fournir son service. Il existe de nombreuses formes ou méthodes pour perpétrer une attaque DOS, dans le cas des réseaux Peer-to-peer, la forme la plus commune est d'inonder le réseau par des paquets invalides, ainsi, empêcher le trafic légitime du réseau, Une autre méthode consiste à noyer la victime dans un calcul fastidieux de sorte qu'elle est trop occupée pour faire répondre à toutes les requêtes [Baptiste, 2005].

Quand plusieurs nœuds sont impliqués dans l'attaque, on parle alors d'une attaque DDOS (Distributed Denial Of Service), dans laquelle l'attaquant exploite un grand nombre d'hôtes infectés par des virus, chevaux de Troie ou vers, il peut alors contrôler à distance ces machines (qualifiés de zombies ou des esclaves) et diriger l'attaque à tout autre hôte ou réseau. L'attaquant peut exploiter les applications de partage de fichier P2P de deux façons pour lancer une attaque DDOS: index poisoning et routing table poisoning [Pankaj Kohli et al, 2007].

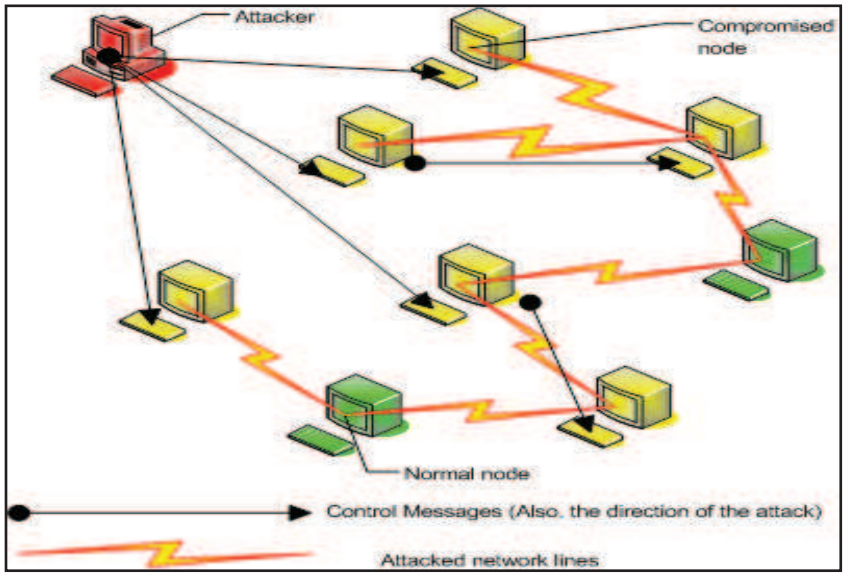

Figure 2.10:l'attaque DDOS

❖ **Solution:**

Pour empêcher une attaque DDOS, une technique connu sous le nom de *pricing* est utilisée, elle consiste à limiter la vitesse des requêtes (voir figure ci-dessous)**.** Quand un attaquant envoie une requête à un nœud, le nœud répond avec une sorte de Puzzle (par exemple: Que pouvez-vous ajouter à la chaîne "adabsdh1" afin de rendre tous les X premiers bits de son hachage SHA-1 nuls?), Ensuite, l'attaquant (demandeur) doit résoudre ce casse-tête et fournir une réponse valide avant que la demande soit examinée. Cette méthode est efficace contre un petit nombre d'attaquants, mais elle peut avoir un impact sur certains pairs légitimes, car ils pourraient percevoir des puzzles trop durs, ce qui conduit à un gaspillage d'énergie.

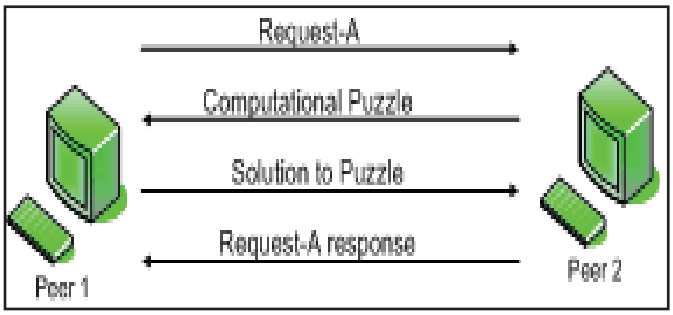

Figure 2.11:Méthode pricing

✓ **Attaque Sybil**

Selon Douceur [J.Douceur et al, 2002] l'attaque Sybil consiste à créer un grand nombre de pairs malveillants dans le réseau pour une même entité dans le but d'attirer le plus de traffic

possible et de gagner plus d'influence par rapport aux nœuds ordinaires. Le nombre d'identités que l'attaquant peut générer dépend des ressources de l'attaquant comme la bande passante, la mémoire et la puissance de calcul. Avec le développement du matériel en termes de capacités de stockage et de traitement ainsi la popularité de l'Internet haut débit, même un attaquant qui utilise un matériel de base peut causer un tort considérable à de grands systèmes [A.Mohaisen et al, 2010].

L'attaque Sybil n'est pas seulement pertinente dans les réseaux P2P, mais également dans d'autres types de réseaux tels que les réseaux ad-hoc et les réseaux de capteurs. Une taxonomie de l'attaque sybil a été proposée par Newsome et al [J.Newsome et al, 2004] selon les trois dimensions suivantes:

- Communication directe vs communication indirecte: si les nœuds Sybil communiquent directement avec les nœuds légitimes, c'est une attaque sybil directe. En revanche, dans une attaque sybil indirecte, les nœuds Sybil peuvent communiquer à travers un ou plusieurs nœuds malveillants se proclament capable de les atteindre.
- Identités fabriquées vs identités volées: un nœud sybil peut fabriquer une nouvelle identité ou bien il peut voler l'identité d'un nœud légitime.
- Simultanéité: l'attaquant peut faire participer toutes ses identités Sybille de façon simultanée dans le réseau, comme il peut alternativement présenter seulement une partie de ses identités sur une période de temps donné.

La figure ci-dessous montre une attaque sybil où un nœud malicieux "AD" est présenté par plusieurs identités: "AD" apparaît comme un nœud 'F' pour 'A', 'C' pour 'B' et 'A' pour 'D', alors quand 'A' veut communiquer avec 'F' il envoie le message à "AD".

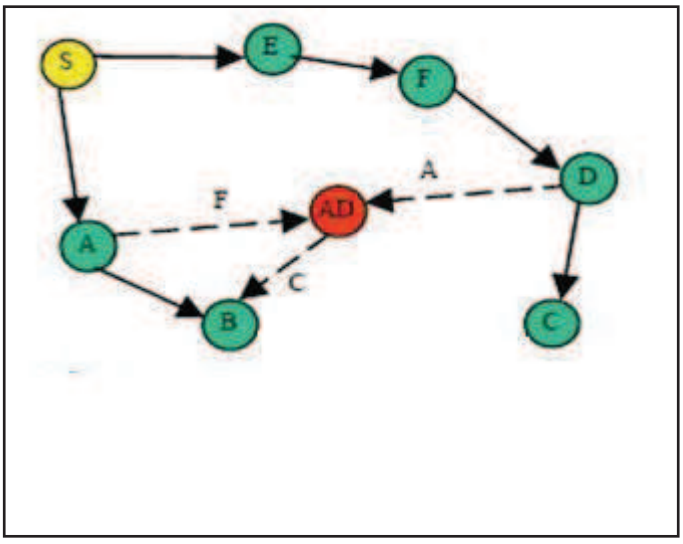

Figure 2.12: l'attaque sybil

❖ **Solution:**

La nature du réseau sans fil P2P rend l'arrêt complet de l'attaque Sybil impossible, il existe plusieurs mécanismes pour la protection contre l'attaque Sybil qui peuvent être divisées en trois catégories:

- **Certification approuvée:**

Elle est l'approche la plus citée dans la littérature pour lutter contre les attaques Sybille [B. Levine et al, 2006], dans laquelle chaque pair doit obtenir un droit d'accès personnel en présentant une ressource administrative supposée unique (adresse IP, carte d'identité). L'hypothèse est qu'un attaquant ne peut pas se faire passer pour plusieurs personnes auprès d'une autorité ponctuelle [François, 2009], cependant la présence de cette autorité va de plus à l'encontre des principes du réseau sans fil pair-à-pair en créant un point de confiance et de faiblesse.

- **Test de ressources:**

Cette technique est utilisé pour vérifier des ressources de calcul ou de stockage, pour accepter une identité, un vérifieur peut lancer un défi exigeant en ressources en broadcastant une requête aux identités et valide seulement les identités dont les réponses ont lieu dans un intervalle de temps donné. Cependant, nous ne pouvons malheureusement pas envisager d'utiliser une telle approche vu qu'elle ne peut malheureusement convenir à des systèmes distribués dans un réseau de large étendue. En effet, la validation peut nécessiter d'importants coûts en termes d'énergie.

- **Test de ressources social:**

Ce mécanisme utilise les relations entre les utilisateurs, chaque pair doit posséder une reconnaissance sociale antérieure pour accéder au réseau. L'hypothèse est qu'un attaquant possède un nombre limité de relations sociales et en tous les cas comparable au nombre de relations d'un utilisateur honnête. La création d'une telle identité n'implique aucun coût matériel et semble juste. Elle a déjà été utilisée dans des systèmes autres que le pair à pair. Par exemple, GMail.

✓ **Attaque Eclipse :**

L'attaque Eclipse est plus général que l'attaque sybil décrite précédemment, elle consiste à contrôler une grande partie des voisins d'un bon pair (un pair sain). Dans cette situation, l'ensemble des nœuds malicieux travaillent ensemble pour tromper le nœud sain en écrivant leurs adresses dans sa table, et donc modifier l'utilisation normale du routage [Lin Wangm, 2006]. L'attaquant cache alors les bons pairs sur le réseau, d'où le nom d'« attaque eclipse »

L'attaque Eclipse est étroitement liée à l'attaque Sybil, elle peut être lancé en utilisant l'attaque sybil, c'est à dire en se forgeant de nombreuses identités sur le réseau, afin d'être présent comme voisin sur un maximum de nœud dans le réseau [Patrick, 2007]. Cependant, les attaques Eclipse sont possibles même en présence d'une défense efficace contre les attaques Sybil, telles que la certification d'identités d'un pair.

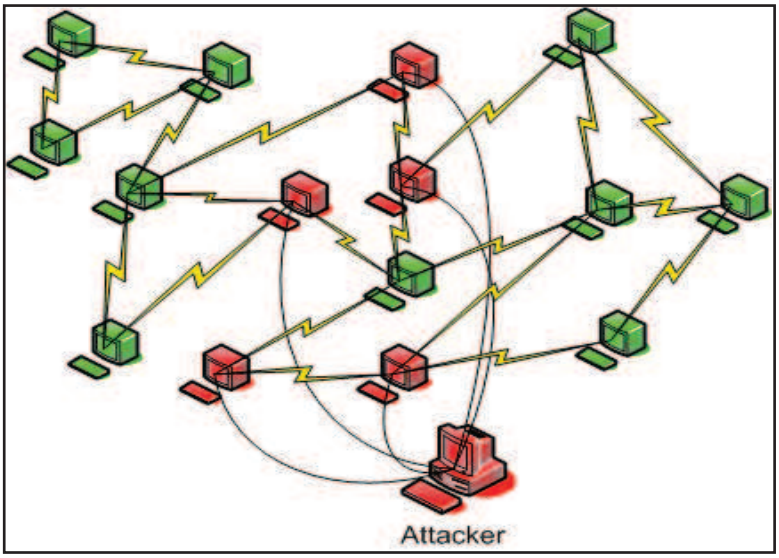

Figure 2.13: attaque Eclipse

❖ **Solution:**

Pour prévenir d'une attaque Eclipse, nous pouvons réutiliser les mêmes mécanismes de la prévention d'une attaque MitM à savoir: les Signatures numériques et la cryptographie à clé publique, qui permettront d'éviter la modification et la lecture passive des messages. Cependant, en raison de l'échelle d'une attaque Eclipse, elle pose toujours une menace pour tout le réseau, (Une attaque MitM n'est pas une menace pour l'ensemble du réseau, tant que la tolérance aux pannes du réseau peut gérer la perte des trois nœuds (2 pairs et un attaquant)), Si les messages sont tous supprimés, donc le réseau est divisé en deux (ou plus) partitions. Etant donné assez d'endroits stratégiques, l'attaquant pourrait partitionner le réseau en autant

de partitions désirées (ce qui limite la taille de chaque réseau, et en limitant l'utilisation de l'ensemble du réseau). Il est important de noter que, avec une attaque Sybil assez grande, il est toujours possible d'exécuter d'une attaque Eclipse [Lin Wangm, 2006].

6. Conclusion:

Les réseaux sans fil P2P sont encore vulnérables à de nombreuses attaques (attaque Sybil, pollution, etc.) pouvant grandement affecter la sécurité et la qualité du service délivré. Détecter et limiter les effets de tels comportements de manière totalement distribuée restent encore aujourd'hui des majeurs auxquels nous nous attachons dans ce mémoire.

Chapitre III :

Les solutions proposées

au niveau routage

1. Introduction :

La sécurité d'un réseau sans fil P2P peut être réalisée à différents niveaux de la couche protocolaire (Application, MAC, Routage, Physique) et sans une certaine forme de sécurité au niveau de l'une de ces couches, un réseau sans fil P2P est vulnérable à plusieurs types d'attaques. Il est sans doute assez facile d'écouter le trafic, de rejouer les transmissions, de manipuler les en-têtes de paquets ou de rediriger les messages de routage. La plus part des protocoles de routage existant permettent l'acheminement efficace des données cependant l'aspect sécurité est négligé ce qui leur rend aussi vulnérables aux attaques menaçant la fiabilité des données en circulation, la question qui s'impose maintenant n'est plus la recherche de la route optimale mais la recherche du chemin le plus sécurisé. Dans ce chapitre, nous allons faire un état de l'art des solutions proposées contre l'effet des attaques au niveau de la couche routage.

2. Protocoles de routages :

Afin de comprendre les attaques sur les protocoles de routages et les solutions proposées contre ces attaques, il est nécessaire de comprendre leur fonctionnement général. Plusieurs protocoles de routage sont concernés par la notion de sécurité (AODV, DSR, OLSR….etc.), dans la suite de ce chapitre on cite les protocoles de routage existant et leur classification. Le protocole AODV fait l'objet principal de ce chapitre.

2.1. Les protocoles de routage existant:

En fonction du mode de fonctionnement de la phase de découverte du chemin et de la mise à jour des informations de routage, les protocoles de routage des réseaux sans fil sont séparés en trois catégories : proactif, réactif et hybride [Rutvij et al, 2011]. Dans les trois sous-sections qui suivent, nous présentons chacune de ces catégories et fournissons des exemples de ces protocoles.

2.1.1. Protocoles de routage proactifs :

Dans ce type de protocole, les nœuds effectuent des mises à jour périodiques des chemins pour qu'un paquet, dès que nécessaire, soit immédiatement transmis. Les protocoles de routage proactifs ont un problème de scalabilité car la taille des tables de routage ainsi que la taille des paquets contenant les informations de topologie ou de distances échangés

augmentent proportionnellement avec le nombre de nœuds dans le réseau. Parmi les différents protocoles de routage proactifs, on peut citer : les protocoles *DSDV* et *OLSR*.

2.1.2. Protocoles de routage réactifs :

Quant à ce type de protocole, ils n'effectuent des découvertes de chemins qu'à la demande, ce qui permet de minimiser la charge de contrôle des protocoles de routage proactifs, parmi ces protocoles on trouve *DSR* et le protocole *AODV* que nous allons détailler dans le paragraphe 2.2.

2.1.3. Protocoles de routage hybrides :

La combinaison d'un protocole réactif et d'un protocole proactif donne lieu à une troisième catégorie qu'on appelle les protocoles hybrides (ou basés sur les zones), le protocole le plus connue est le *ZRP*.

2.2. Le fonctionnement général du protocole AODV :

Ad hoc On demand Distance Vector (AODV) est un protocole de routage réactif ce qui signifie que les routes sont construites à la demande, Si un nœud source veut envoyer des paquets de données vers un nœud destination, il doit établir et maintenir une route vers ce nœud destination durant le temps qu'il en fait usage.

Le protocole AODV est basé sur l'utilisation des deux mécanismes "Découverte de route" et "Maintenance de route" (utilisés par le DSR), en plus du routage nœud-par-nœud, le principe des numéros de séquence et l'échange périodique du DSDV. Il utilise trois types de paquets de routage à savoir : RREQ (Route REQuest), RREP (Route REPly), RERR (Route ERRor).

❖ **Découverte des routes :**

Avec le protocole AODV, chaque nœud doit maintenir une liste de ses voisins actifs. Cette liste est obtenue par un échange périodique des messages HELLO de chaque nœud avec ses voisins immédiats. Quand un nœud source S veut envoyer des données à un destinataire D et qu'aucune route vers cette destination n'est stockée dans la table de routage de la source, le nœud S initialise une procédure de découverte de routes.

La source S envoie à ses voisins une demande de route RREQ (*Route REQest*) qui contient l'adresse de S, l'identifiant de la requête, un compteur de séquence, l'adresse de D et le compteur de nombre de sauts avec une valeur initiale zéro. La source attendra une période RREP_WAIT_TIMEOUT, si une réponse est reçue alors l'opération de découverte de route est terminée, sinon elle rediffuse le RREQ et attend une période plus grande si aucune réponse n'est reçue, elle continuera la rediffusion du RREQ jusqu'à un nombre maximum de tentatives RREQ_RRTRIES (03 tentatives), si après RREQ_RETRIES tentatives d'établissement de route, il n'y a aucune réponse alors le processus est abandonné et un message d'erreur est signalé à l'application. Après une certaine période d'attente (10 s), l'application demande la route et par conséquent l'opération de découverte de route est initiée [Tebbane et el, 2004]. Chaque nœud qui reçoit le message RREQ recherche dans sa table de routage locale s'il existe une route vers le nœud D sinon le nœud qui traite la requête RREQ incrémente le nombre de sauts et la diffuse à nouveau. Lorsque la requête atteint la destination D ou un nœud qui connaît une route vers la destination, une réponse RREP (*Route REPly*) est diffusée sur la même route de réception du RREQ (chemin inverse). La réponse RREP contient l'adresse source, l'adresse de destination, le nombre de sauts, un numéro de séquence de destination et la durée de vie du paquet. La réponse RREP passe par la route inverse vers le nœud source S. Ainsi chaque nœud, sur cette route, enregistre une entrée dans sa table de routage local vers le nœud destination avant de renvoyer le paquet. Une fois la source S reçoit le message, elle commence à envoyer les données vers D. [ABDELLAOUI, 2009]

La Figure ci-dessous donne un exemple de recherche de route dans un réseau avec AODV. Le nœud *Source* veut envoyer du trafic au nœud *Destination*. Il génère un paquet RREQ qu'il diffuse à ses voisins 2, 3 et 4. Ces trois nœuds à leur tour retransmettent le RREQ à leurs voisins. Ce paquet permet, au niveau de chaque nœud, de mettre à jour le saut suivant pour la route construite vers la *Source*. À chaque saut, le nœud calcule le nombre de sauts depuis la *Source*. Le RREQ arrive au niveau au nœud 8 qui voit qu'il est le nœud destination, il génère alors un paquet RREP qui fait le chemin inverse et informe le nœud source du chemin à prendre.

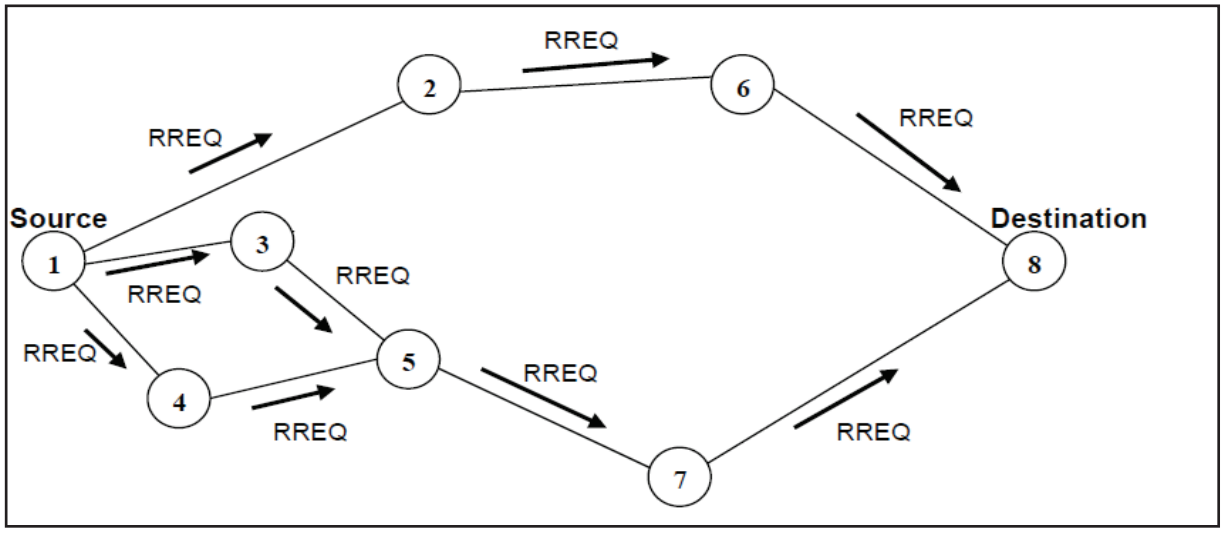

Figure 3.1 : Une demande de route

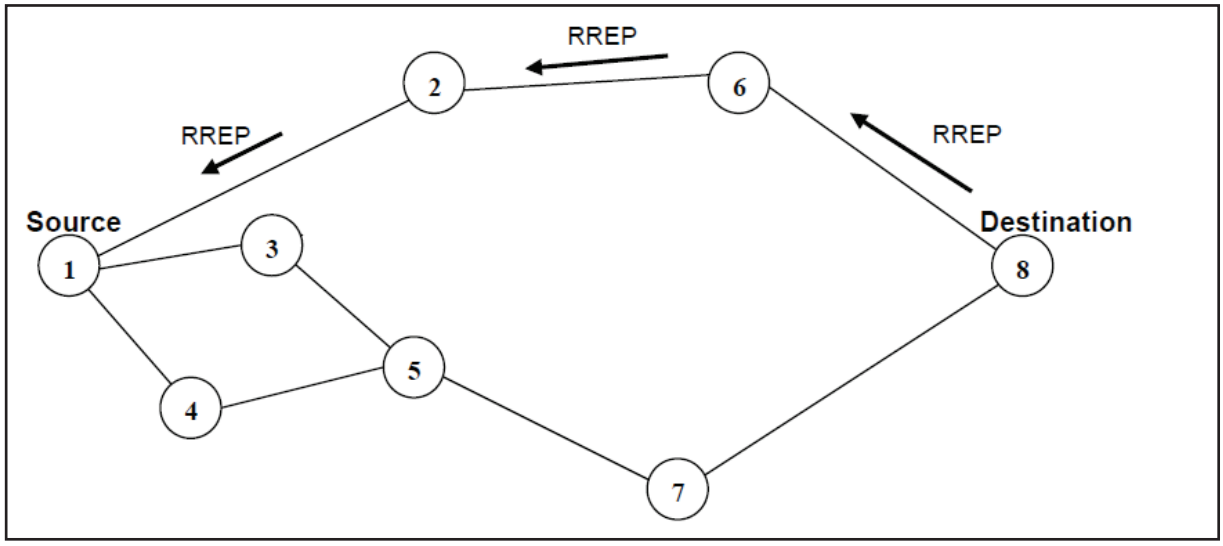

Figure 3.2 : Réponse de route

❖ **Maintenances des routes :**

L'échange des messages HELLO entre les voisins immédiats permet de mettre à jour la liste des voisins de chaque nœud. Lorsqu'un nœud N détecte qu'un autre nœud Q n'est plus accessible (Q a quitté le réseau ou est hors porté radio), N procède à une mise à jour des liens dans sa table de routage. En effet, il recherche dans sa table de routage toutes les routes qui passent par le nœud Q et les détruit avant d'annoncer à ses voisins actifs que la route passant par le nœud Q n'est plus valide. Un message RERR (*Route ERRor*) est envoyé alors au nœud source. Ainsi, la mise à jour est diffusée à travers le réseau saut-par saut et le nœud source initie une nouvelle procédure de recherche de route vers la destination.

3. Les attaques possibles :

Dans cette section, nous nous intéressons aux attaques sur le protocole de routage AODV. Nous présentons comment les nœuds malhonnêtes opèrent pour arriver à leur objectif (perturbation de route, isolation de nœud, consommation des ressources…etc.). Dans [P. Ning et al, 2003] et [Huang et al, 2004] les auteurs divisent les comportements malveillants sur AODV en deux catégories à savoir *atomiques* et *composées,* les premiers est le résultat de la manipulation d'un seul message de routage, les deuxièmes sont définis comme étant une collection d'actions atomiques.

Les manipulations effectuées sur un message de routage peuvent être :

– supprimer un paquet ;

– Modifier un ou plusieurs champs du paquet avant de le retransmettre ;

– Fabriquer une réponse à la réception d'une demande de route RREQ ;

– Fabriquer activement des paquets de routage sans même avoir reçu de messages de routage.

3.1. Attaques élémentaires portant sur les demandes de route : [Mohamed, 2011]

3.1.1. Suppression d'une demande de route :

Un nœud malhonnête pourrait simplement **effacer** la demande de route reçue. En appliquant ce genre de comportement à tout message RREQ reçu, l'attaquant ne participe pas au routage c'est comme s'il ne fait pas partie du réseau. Une autre variante serait d'effacer sélectivement des messages RREQ. Ce comportement peut être comparé à celui d'un nœud égoïste.

3.1.2 Modification d'une demande de route :

À la réception d'une demande de route, le nœud malhonnête **modifie** un ou plusieurs champs qu'il n'est pas supposé modifier avant de retransmettre le message. La modification peut aussi porter sur un champ qu'il a le droit de modifier, mais il ne respecte pas la spécification pour le faire.

Plusieurs champs impliquent des traitements différents lorsqu'ils sont modifiés. Par exemple,
Le champs identifiant de la RREQ associé à l'adresse de la source permet d'identifier de manière unique une demande de route et indique la fraicheur de la demande de route. Puisqu'un nœud n'accepte que le première copie de RREQ, en augmentant cet identifiant, le

nœud malhonnête peut garantir l'acceptation et le traitement de la RREQ modifiée par les autres nœuds.

3.1.3. Fabrication d'une demande de route :

Les attaques décrites précédemment (section 3.1.1 et 3.1.2) sont déclenchées par la réception d'une demande de route. En revanche, les attaques par **fabrication** peuvent être effectuées sans avoir reçu de messages RREQ. Le nœud malhonnête a besoin de collecter certaines informations, en écoutant le trafic par exemple, avant d'injecter le message fabriqué. Par conséquent l'éxécution répétitive de ce type d'attaque peut provoquer l'inondation du réseau par des messages de routage inutiles. Le nœud malhonnête peut orienter le trafic vers une seule destination ou faire croire que le trafic part d'une seule source ou les choisir (source/destination) au hasard.

3.1.4. *Rushing* d'une demande de route :

Dans d'autres cas, le nœud malhonnête peut utiliser la technique du ***rushing*** qui consiste à diminuer le temps de traitement des messages RREQ et les retransmettre plus rapidement afin qu'ils atteignent plus rapidement la destination. Ce qui garantira pour le nœud malhonnête une place sur le chemin.

3.2 Attaques élémentaires portant sur les réponses de route :

3.2.1 Suppression d'une réponse de route :

Ce type d'attaque n'a un sens que si le nœud malhonnête a été choisi sur la **route** reliant la source à la destination. Dans ce cas, la **suppression** de la réponse de route empêche la formation du chemin vers la destination et entraine des messages de contrôle supplémentaires suite à l'initialisation d'un nouveau processus de création de route, ce qui dégrade la qualité de service.

3.2.2 Modification d'une réponse de route :

Comme pour les demandes de route, le nœud malhonnête peut jouer sur le numéro de séquence de la destination et/ou le nombre de saut dans une RREP en augmentant le premier et en diminuant le second. Ces paramètres sont pris en compte lors de la mise à jour du chemin vers la destination : une mise à jour est possible si le numéro de séquence reçu dans la

demande de route est plus grand que celui stocké dans la table de routage ou les numéros de séquence sont égaux et le nombre de sauts reçu est plus petit que celui stocké dans la table de routage. Cette intervention permet de garder le chemin qui passe par le nœud malhonnête même si un autre chemin plus court est proposé par un autre nœud.

3.2.3 Fabrication d'une réponse de route :

Certaines attaques peuvent être effectuées sans pour autant être choisi sur le chemin. C'est le cas des attaques par **fabrication** :

– Fausse réponse : à la réception d'une demande de route, le nœud malhonnête fabrique une réponse de route même s'il n'a pas de chemin valide vers la destination (attaque Blackhole). Dans un autre cas de figure, le nœud malhonnête répond avec une réponse de route même s'il n'est pas supposé le faire lorsque le drapeau D est à 1 (indiquant que seulement la destination doit répondre).

– Réponse active : des réponses de routes sont fabriquées et injectées dans le réseau même sans avoir reçu une demande de route au préalable. Dans ce cas le nœud malveillant peut jouer sur des champs pour produire l'effet désiré. Une variante de cette attaque vise à déborder la table de routage d'une cible en proposant des routes (via RREP) vers des nœuds (nouveaux ou inexistants).

– En écoutant la transmission d'une réponse de route qui ne lui est pas destinée, un nœud malhonnête peut fabriquer et injecter un paquet RREP proposant un chemin plus court et plus frais provoquant la mise à jour du chemin vers la destination qui passe dorénavant par le nœud malveillant.

3.3 Attaques élémentaires portant sur les erreurs de route :

3.3.1 Suppression d'une erreur de route :

Comme c'est le cas pour les RREQ et RREP, en **effaçant** une RERR, un nœud malhonnête peut retarder la détection des liens défaillants. Cependant, l'impact de cette attaque est restreint du fait que les nœuds en amont découvrent le problème et demandent l'établissement de nouvelles routes.

3 .3.2 Modification d'une erreur de route :

Un nœud malhonnête peut **modifier** des erreurs de route avant de les retransmettre. Ainsi, il peut supprimer des destinations non-joignables pour faire croire qu'elles le sont encore et ajouter des destinations qui sont joignables et actives pour faire croire qu'elles ne le sont plus et les désactiver.

3.3.3 Fabrication d'une erreur de route :

Un nœud malhonnête peut **fabriquer** un message d'erreur de route et déclarer autant de routes non-joignables causant l'invalidation des entrées correspondantes dans la table de routage des nœuds recevant le message de contrôle.

3.4. Attaques composés :

Une attaque composée est la combinaison de plusieurs attaques élémentaires afin d'atteindre des objectifs plus évolués. Dans la section suivante nous présentons quelques attaques composées.

3.4.1 Répétition régulière d'attaques élémentaires :

Cette attaque se base sur une répétition régulière d'une attaque élémentaire pour avoir un impact permanent. Par exemple, l'envoi continu de messages de demande de route RREQ fabriqués à destination d'une cible est efficace pour empêcher celle-ci de recevoir les messages des autres nœuds. De même, l'envoi de réponses de route fabriquées est efficace pour empêcher la cible d'envoyer des messages aux autres nœuds (puisque le nœud malhonnête devient le prochain saut vers les autres nœuds). En combinant ces deux attaques composées, un nœud malhonnête peut isoler sa cible.

3.4.2. Création d'une boucle de routage :

La formation d'une boucle de routage dans une route déjà établie est faite en fabriquant deux réponses de routes. Nous présentons cette attaque à travers l'exemple suivant :

On suppose l'existence d'un chemin entre le nœud 1 et le nœud 2 passant par 3, 5 et 4. Le nœud malicieux M fabrique une première réponse de route où l'adresse source est initialisée à 1, l'adresse destination à 2 et un numéro de séquence de la destination 2 supérieur au numéro de séquence actuel. Il fait croire que ce paquet est envoyé par le nœud 3 à

destination du nœud 6 (valeurs à mettre respectivement dans le champ adresse source et destination de l'entête IP). Ceci provoque l'envoi de tout paquet reçu à destination de 2 vers le nœud 3 (voir figure 2.3b). Ensuite, M fabrique une deuxième réponse de route équivalente à la première sauf qu'il fait croire qu'elle est à destination du nœud 4, provenant du nœud 6. Cette action garantie que tout paquet transféré vers 2 à travers 4 est envoyé vers 6, ce qui complète la boucle.

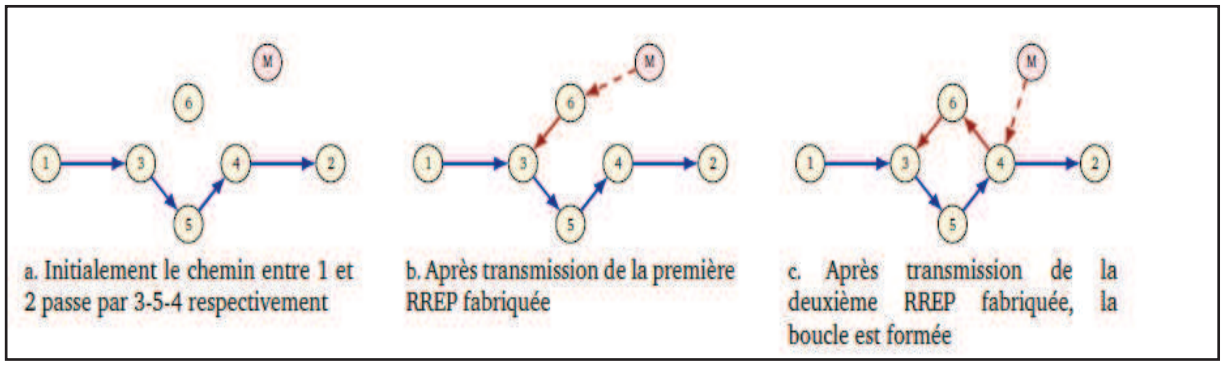

Figure 3.3: Boucle de routage

Parmi toutes les attaques possibles que nous avons présentées, nous allons parler plus en détails de l'attaque Blackhole et parler des différentes solutions pour pallier ce trou de sécurité.

4. Etude de l'attaque Blackhole :

L'attaque blackhole [Maha et al, 2011] est un problème sérieux qui doit être résolu dans les réseaux sans fil P2P, elle peut s'effectuer au moment où un nœud source initie un processus de découverte de route en émettant un paquet RREQ, le nœud corrompu en le recevant va répondre par un paquet RREP avec un numéro de séquence non seulement erroné mais également élevé afin d'augmenter ses chances de faire partie de la route. En effet, si son paquet RREP atteint la source le premier par rapport aux réponses des nœuds légitimes, il peut s'intégrer dans la route pour intercepter et contrôler une partie ou la totalité du trafic échangé au sein du réseau, de façon à pouvoir surveiller, bloquer ou même détourner certains flux du trafic. Le trafic absorbé peut être donc soit redirigé vers un autre nœud soit disparaître complètement.

Cette attaque est grave dans la mesure où les nœuds légitimes vont mettre à jour leurs tables de routage avec des informations fausses et le nœud malveillant n'a pas besoin d'intervenir une seconde fois.

La figure ci-dessous illustre cette attaque où la source S veut transmettre des données vers la destination D, elle diffuse une requête RREQ (Route REQuest), le paquet RREQ va être reçu par les nœuds N1, N2, N3.

Supposons le nœud N3 a une route vers la destination dans sa table de routage, le nœud N3 génère un paquet de réponse RREP et mettre à jour sa table de routage par le nombre de sauts et le numéro de séquence de la destination.

Le numéro de séquence de la destination est un entier de 32 bits associé à chaque route et permet l'utilisation des routes les plus fraîches autrement dit les plus nouvelles. Une route est jugée fraîche que si la base du numéro de séquence de la destination est assez élevée [K.LakSBmi et al, 2010].

Le nœud N3 va envoyer le paquet vers le nœud M, tant que les nœuds N1 et N2 n'ont pas une route vers la destination D, ils seraient à nouveau diffusé le message de contrôle RREQ. Ainsi le paquet RREQ diffusé par le nœud N3 devrait également être reçu par le nœud M (supposons M est un nœud malicieux). Donc le nœud M va générer un faux message de contrôle RREP et l'envoyer au nœud N3 avec un numéro de séquence de destination très élevé, qui serait ensuite envoyé au nœud S.

Cependant, tant que le numéro de séquence de destination est élevé, la route à partir du nœud N3 sera considéré comme plus fraiche et donc la source serait commencée à envoyer des paquets de données au nœud N3. Le nœud N3 serait envoyer les même paquets au nœud malicieux. Le message de contrôle RREQ à partir du nœud N1, finirait par atteindre le nœud D (nœud de destination), ce qui générerait un message de contrôle RREP et la route du retour. Toutefois, le nœud S a déjà reçu un paquet de réponse RREP avec numéro de séquence supérieur à celui de D, le nœud S ignore les deux véritables messages de contrôle RREP.
Pour chaque message de contrôle RREP reçu, la source devrait d'abord vérifier si elle possède une entrée pour la destination dans la table de routage ou non. S'il en trouve un, le nœud source serait de vérifier si le numéro de séquence de destination dans le message de contrôle reçu est plus élevé que celui qu'il a envoyé dans la dernière RREQ ou non. Si le numéro de

séquence de destination est plus élevé, la source met à jour sa table de routage avec le nouveau message RREP, sinon le message de contrôle RREP sera rejeté.

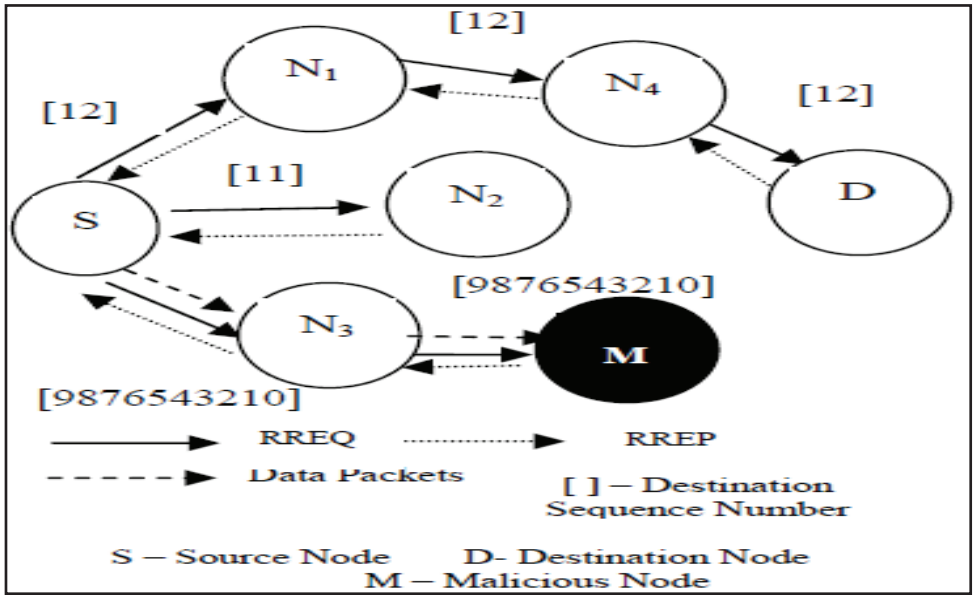

Figure 3.4: Attaque Black Hole

5. Méthodes développées pour la sécurisation au niveau routage :

Plusieurs solutions ont été proposées pour pallier des problèmes de sécurité dans les réseaux sans fil (ad hoc ou P2P). Dans cette section nous allons présenter les différentes propositions dans la littérature pour éviter l'attaque blackhole.

[Deng et al, 2002] ont proposé une solution contre l'attaque trou noir en modifiant le protocole AODV. Dans cette méthode chaque nœud intermédiaire doit inclure l'information « next hop » quand il envoie un paquet RREP. Une fois la source a reçu le paquet RREP et avant d'envoyer les paquets de données, il extrait l'adresse du « next hop » et lui envoie une nouvelle demande de route (Further Request) afin de vérifier qu'il possède une route vers le nœud intermédiaire qui a envoyé le message de réponse, et qu'il a aussi une route vers le nœud destination. Le « next hop » répond avec un paquet de réponse de route (Further Reply) qui comprend le résultat de contrôle. La source vérifie les informations des paquets FRREP et agit selon les règles suivantes:

1) Si le « next hop » possède une route vers le nœud intermédiaire et la destination, la source établit la route reçu du nœud intermédiaire et commence l'envoi des données.

2) Si le « next hop » a une route vers la destination, mais n'a pas de route vers le nœud intermédiaire, la source suppose que le nœud intermédiaire est un nœud malicieux. Ensuite, elle initie l'envoi des données via la nouvelle route à travers le next hop et diffuse un message d'alarme dans le réseau afin d'isoler le nœud malveillant.
3) Si le « next hop » n'a pas de routes vers le nœud intermédiaire et la destination, la source lancera un nouveau processus de découverte de route, et envoie également un message d'alarme afin d'isoler le nœud malveillant.

Le mécanisme proposé est efficace dans la détection de l'attaque Blackhole, cependant, l'envoi d'un paquet FRREQ à partir du nœud source vers le « next hop » et l'attente du paquet FRREP du « next-hop » augmente la charge du routage « overhead » entre la source et le « next hop », surtout quand ce mécanisme est appliqué sur un réseau à grande échelle et la distance entre la source et le nœud malicieux est longue.

[Al-Shurman et al, 2004] ont proposé deux solutions conçues pour cibler l'attaque Blackhole dans le protocole AODV. La première solution proposée consiste à trouver plus d'une route vers la destination (au moins trois routes différentes). La source envoie un paquet RREQ au nœud destination en utilisant ces trois routes. La destination, le nœud malicieux et les nœuds intermédiaires vont répondre à ce paquet. Le nœud expéditeur met ses paquets de données dans un tampon jusqu'à ce qu'il reçoit plus d'une réponse RREP ; lorsque la source reçoit des RREP, si les routes à destination ont des nœuds partagés, la source peut reconnaître une voie sûre vers la destination, et les paquets vont être transmis. Si aucuns nœuds partagés ne semblent être dans ces routes redondantes, l'expéditeur attendra une autre RREP jusqu'à ce qu'un chemin avec des nœuds partagés identifié ou le temps d'attente soit expiré.
Cette solution peut garantir à trouver une route sécurisé vers la destination, mais le principal inconvénient est le délai d'attente. Plusieurs paquets RREP doivent être reçues et traitées par la source. En outre, s'il n'ya pas de nœuds partagés entre les routes, les paquets ne seront jamais envoyé.
La seconde solution proposée exploite le numéro de séquence inclus dans l'en-tête de chaque paquet. Le nœud dans cette situation a besoin d'avoir deux tables supplémentaires; la première table comprend les numéros de séquence du dernier paquet envoyé à chaque nœud dans le réseau.

La deuxième table contient le numéro de séquence reçu de chaque expéditeur. Pendant la phase de réponse de route, le nœud intermédiaire ou la destination doivent inclure le numéro de séquence du dernier paquet reçu de la source qui déclenche la demande de route. Une fois la source reçoit ce RREP, il va extraire le dernier numéro de séquence, puis le comparer avec la valeur enregistrée dans sa table. Si elle correspond, la transmission aura lieu. Si ce n'est pas, ce nœud est un nœud malveillant, alors un message d'alarme sera diffusée pour avertir le réseau sur ce nœud. Toutefois, les deux solutions ont le délai de bout en bout comme inconvénient.

Selon la solution proposée par [Tamilselvan et.al, 2007], la source doit attendre d'autres réponses avec des détails sur le prochain saut avant d'envoyer les paquets de données vers la destination. Une fois un nœud a reçu la première RREQ, il fixe un temporisateur « timer » dans le "Timer Expired Table" pour collecter les nouvelles demandes venant des différents nœuds et les stocker dans un ordre séquentiel, la source va stocker le « numéro de séquence », et « le moment d'arrivé du paquet » dans « Collect Route Reply Table » (CRRT). La valeur "timeout" est basée sur le temps d'arrivé du premier RREQ. Elle vérifie d'abord dans la table CRRT afin de déterminer s'il existe un next hop répété dans les réponses de route reçus, dans ce cas il assume que les chemins sont corrects ou la chance de chemins malveillants est limitée. S'il n'ya pas de répétition, une route aléatoire est sélectionné de la table CRRT. L'inconvénient de la solution proposée est le délai de bout en bout, puisque le nœud source doit attendre d'autres réponses de route avant d'envoyer les données.

[Lalit Himral et.al, 2011] ont proposé une méthode pour trouver les routes sécurisées et prévenir les nœuds malveillants dans les MANET en vérifiant s'il existe une différence importante entre le numéro de séquence du nœud source et le nœud intermédiaire qui a envoyé la première RREP. En règle générale, la première réponse sera du nœud malveillant avec un numéro de séquence de destination très élevée, la réponse sera stockée comme la première entrée dans RR-table. Ensuite, comparez le premier numéro de séquence de destination reçu avec le numéro de séquence du nœud source, s'il existe une grande différence entre eux, il est certainement que cette réponse vient du nœud malveillant, par conséquent supprimer immédiatement cette entrée de la table. La méthode proposée offre les avantages suivants : (1) Le nœud malveillant est identifié dans la phase initiale et il est retiré immédiatement. (2) Aucune modification n'est faite dans les autres opérations du protocole AODV. (3) Une meilleure performance en légère modification. Cependant la méthode ne peut pas trouver de multiples nœuds malicieux.

La solution proposée dans [Payal N et al, 2009] modifie le comportement de l'AODV pour inclure un mécanisme permettant de vérifier le numéro de séquence de RREP reçu. Quand le nœud source reçoit un paquet RREP il compare le numéro de séquence du RREP reçu à une valeur de seuil. Le nœud répondant est soupçonné d'être un trou noir si son numéro de séquence est supérieur à la valeur de seuil. Le nœud source ajoute le nœud suspect à sa liste noire, et se propage un message de contrôle appelé une alarme pour faire connaître la liste noire pour ses voisins. Le seuil est la moyenne calculée de la différence entre le numéro de séquence de destination dans la table de routage et le numéro de séquence de destination dans le RREP dans une période de temps. Le principal avantage de ce protocole est que le nœud source annonce le trou noir à ses voisins afin d'être ignoré et supprimé.

Un algorithme présenté dans [Subash et al, 2011] pour détecter l'attaque trou noir dans un MANET basé sur un prétraitement appelé Pre_Process_RREP, il est simple ainsi il ne change pas le fonctionnement de l'un des nœuds intermédiaires ou de destination. Il n'a même pas modifié le fonctionnement normal de l'AODV. Le processus continue à accepter les paquets RREP et appelle un processus appelé Compare_Pkts (p1 paquets, p2 paquet) qui compare le numéro de séquence de destination des deux paquets et sélectionne le paquet avec un numéro de destination supérieur si la différence entre les deux numéros n'est pas sensiblement élevée. Le paquet contenant exceptionnellement un numéro de séquence de destination élevé est soupçonnée d'être un nœud malveillant et un message d'alerte contenant l'identification du nœud est généré et diffusé vers les nœuds voisins de sorte qu'il peut être isolé du réseau et peut maintenir une liste de ces nœuds malveillants.

6. Conclusion :

Ce chapitre a fusionné les divers travaux liés aux mécanismes de détection de l'attaque Blackhole, les auteurs ont donné plusieurs propositions pour la détection et la prévention de cette attaque, chacune a ses proposes avantages et inconvénients. Dans le chapitre qui suit nous allons mettre le point sur notre protocole proposé et ses performances en utilisant le simulateur NS2.

Chapitre IV :

Approche distribuée pour la sécurité des réseaux Sans fil P2P

1. Introduction :

Comme nous avons vu dans le chapitre précédent diverses solutions ont été proposées pour la sécurisation des réseaux sans fil contre l'attaque Blackhole. L'objectif principal de ce chapitre est de présenter notre contribution dans le cadre de ce mémoire. Nous commençons d'abord par une description du protocole proposé, ensuite nous discutons de l'environnement utilisé pour l'implémentation de la solution, enfin, nous analysons la performance de notre protocole avec les différentes simulations.

2. Le protocole proposé :

La notion de confiance s'est appliquée dans les télécommunications avec la notion de connaissance au préalable des identités. Mais aujourd'hui le développement de nouveaux modèles de communication tels que les réseaux ad hoc, les réseaux sans fil P2P, rendent cette vision de la confiance obsolète [Véronique et al, 2004]. En outre la confiance n'est pas un problème technique, elle est un problème social à opposer à la notion de sécurité : on a besoin de confiance lorsque la sécurité n'est pas suffisante.

Nous envisageons de proposer un nouveau protocole basé sur l'utilisation d'un modèle de confiance capable d'assurer les échanges sécurisés dans les réseaux sans fil P2P, tout en tenant compte des caractéristiques de ces réseaux.

Dans notre modèle, et afin d'évaluer le degré de confiance d'un nœud, chaque nœud dans le réseau maintient une table d'activité, dans cette table il sauvegarde l'identifiant d'un nœud, le nombre des paquets de données, le nombre des paquets de demande de route (RREQ) et le nombre des paquets de réponse (RREP) reçus de ce nœud.

Quand un nœud légitime reçoit un paquet, selon le type du paquet reçu, il augmente le nombre dans sa table d'activité.

Si le paquet reçu est de type RREP, il consulte sa table d'activité pour vérifier l'une des équations ci-dessous, selon les valeurs stockées dans cette table, il décide si le nœud est un nœud de confiance ou bien non.

A chaque fois qu'un nœud blackhole reçoit un paquet de données, il le supprime directement, ainsi quand il reçoit un paquet RREQ, il répond en envoyant une fausse RREP sans consulter sa table de routage et il ne rediffuse pas le RREQ vers les autres nœuds. En se

basant sur ce comportement, un nœud légitime ne recevra aucun paquet de données ou bien un paquet RREQ d'un nœud malicieux, il reçoit que des paquets de réponse RREP, par conséquent, si on suppose que :

NB-D : le nombre des paquets de données reçus d'un nœud X

NB-RREQ : le nombre des paquets RREQ reçus d'un nœud X

NB-RREP : le nombre des paquets RREP reçus d'un nœud X

Si ***(NB-D+NB-RREQ > NB-RREP)*** alors : X est un nœud de confiance

Si ***((NB-D+NB-RREQ!=0) and (NB-RREP >NB-D+NB-RREQ))*** alors : X est un nœud connu

Si ***(NB-D+NB-RREQ=0)*** alors : X est un nœud inconnu et peut être un nœud Blackhole

Dans ce qui suit, nous présentons l'idée générale du protocole :

Step 1 :

Le nœud source S commence la phase de découverte de route

Step 2 :

Chaque nœud intermédiaire reçoit un RREQ stocke le numéro de séquence de la source (SSN)

Step 3 :

Quand un nœud intermédiaire reçoit un RREP, il vérifie d'abord si le nœud existe dans le blacklist, si la condition est vraie, il le supprime directement. Sinon il passe à l'étape 4

Step 4 :

Dans cette étape il vérifie un bit rajouté au format du paquet RREP, pour éviter que plusieurs nœuds vérifient plusieurs fois le même paquet.

Si (le bit = 1) alors :

- Le RREP a été déjà vérifié par un nœud et le nœud suivant n'aura plus besoin de revérifier le paquet (dans ce cas le nœud est jugé soit de confiance, soit connu)
- Rediffuser RREP vers la source

Sinon (le bit =0)

Switch Etat du nœud

Case 1 : Le nœud est jugé de Confiance

- Mettre le bit = 1
- Rediffuser RREP vers la source

Case 2 : Le nœud est jugé Connu

- Mettre le bit = 1
- Rediffuser RREP vers la source

Case 3 : Le nœud est Inconnu (Route non sécurisé, et le nœud peut être un Blackhole)

Si (DSN>>SSN) (pour confirmer)

- Il ne le renvoie pas à la source
- Ajouter le nœud au blacklist
- Supprimer le RREP

Sinon

- Mettre le bit = 1
- Rediffuser RREP vers la source

Finsi

3. Implémentation :

Afin d'implémenter notre protocole, nous avons utilisé le simulateur NS2 [http://www.isi.edu/nsnam/ns/], et nous avons effectué plusieurs modifications à plusieurs niveaux, tout d'abord nous avons implémenté l'attaque, puis nous avons intégré le protocole proposé qui est une version modifié du AODV. Nous avons choisi le protocole AODV car il consomme moins d'énergie, il réduit le surcoût de routage et il est plus adaptable aux réseaux dynamiques.

3.1. Présentation du Simulateur NS2 :

Network Simulator (NS2) est un simulateur à événements discrets orienté objet, écrit en C++ avec une interface qui utilise le langage OTcl (Object Tool Command Langage). A travers ces deux langages il est possible de modéliser tout type de réseau et de décrire les conditions de simulation : La topologie réseau, le type du trafic qui circule, les protocoles utilisés, les communications qui ont lieu….etc. Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. Pour reprendre la terminologie objet, il sert à définir les classes. Quant au langage OTcl, il fournit un moyen flexible et puissant de

contrôle de la simulation comme le déclenchement d'événements, la configuration du réseau, la collecte de statistiques, etc.

Network Simulator offre plusieurs avantages comme :

- Il est open source et gratuit
- Il englobe les contributions de plusieurs chercheurs
- Il peut être étendu à d'autres modèles grâce à sa conception orientée objet et son implémentation en C++
- Il est riche en modèles et en protocoles pour les deux environnements filaires et sans fil
- Les résultats de simulation sont générés dans un fichier trace que l'utilisateur peut exploiter.

Toute simulation sous NS2 se base sur un modèle composé des éléments suivants :

- Nœuds du réseau : Nœuds d'extrémités où le trafic est généré ou consommé plus les nœuds de routage (nœuds intermédiaires)
- Liens de communications entre ces nœuds.
- Agents : représentent les protocoles au niveau transport (TCP, UDP), ces agents sont connectés aux nœuds et sont attachées les uns aux autres pour permettre l'échange de données.
- Application : qui génère le trafic des données.

3.2. Préparation de l'Environnement d'Implémentation :

La préparation de l'environnement d'implémentation consiste à installer le simulateur de réseau NS2 sous le système d'exploitation LINUX UBUNTU 10.10. Nous avons utilisé la version NS2.34. L'installation s'effectue en trois grandes étapes :

- ✓ Copier le package d'installation NS2.34 dans le répertoire USER du système ;
- ✓ Taper la commande d'installation './Install' dans le terminal de commandes, dans le répertoire NS2.34 précédemment copié ;
- ✓ Modifier les variables d'environnement et cela consiste à ajouter les lignes de code ci-dessous dans le fichier '.bashrc' ;

- ✓ Et enfin taper la commande './make' dans le terminal de commandes afin de compiler NS2 et de générer tous les fichiers NS2 nécessaires à son fonctionnement.

#NS2 Path

export

NS=/home/user/ns-allinone-2.34/ns-2.34/ns

export

PATH=${PATH}:/home/user/ns-allinone-2.34/bin:/home/user/ns-allinone2.34/tcl8.4.18/unix/ :/home/user/ns-allinone-2.34/tk8.4.18/unix

export

NAM=/home/user/ns-allinone-2.34/nam-1.14

export

LD_LIBRARY_PATH=/home/user/ns-allinone-2.34/otcl-1.13,/user/ns-allinone-2.34/lib

export

TCL_LIBRARY=/user/ns-allinone-2.34/tcl8.4.18/library

Figure 4.1 : Les lignes ajoutées dans le fichier « .bashrc »

3.3. L'ajout d'un nouveau protocole dans NS2 :

Pour Implémenter notre proposition, nous avons rajouté deux protocoles dans le NS2, à savoir : BHAODV et SBAODV pour Blackhole AODV et Solution Blackhole AODV respectivement. Dans le premier nous avons implémenté l'attaque Blackhole et dans le deuxième nous avons implémenté notre proposition. Dans ce qui suit nous allons parler de l'ajout du protocole SBAODV.

Dans [Ros et al, 2004]. La mise en œuvre d'un nouveau protocole de routage dans le NS-2 est décrite. Pour mettre en œuvre notre contribution, nous avons utilisé les détails

expliqués dans ce papier. Dans notre travail, nous avons tout d'abord intégré notre proposition dans l'AODV, puis nous avons suivi les étapes expliquées ci-dessous en détail:

Tous les protocoles de routage dans NS sont installés dans le répertoire «ns-2.34 ». Nous avons dupliqué le protocole AODV dans ce répertoire et changer le nom de répertoire comme "SBaodv".

Tous les fichiers du dossier dupliqué dans le répertoire sont modifiés pour "SBaodv" tels que SBaodv.cc, SBaodv.h, SBaodv_rqueue.cc, SBaodv_rqueue.h, sauf pour "aodv_packet.h", afin d'utiliser les mêmes paquets de données de l'AODV, ainsi toutes les fonctions, structures et les variables sont modifiés. Après les changements apportés, nous avons modifié deux fichiers communs qui sont utilisés dans NS-2 pour intégrer le nouveau protocole dans le simulateur. Ces modifications sont expliquées ci-dessous :

```
#SBAODV patch
SBAODV {
set ragent [$self create-SBaodv-agent $node]
}
Simulator instproc create-SBaodv-agent { node } {
        set ragent [new Agent/SBAODV [$node node-addr]]
        $self at 0.0 "$ragent start" # start BEACON/HELLO Messages
        $node set ragent_ $ragent
        return $ragent
}
```

Figure 4.2: L'ajout de l'agent SBaodv dans ns-2.34/tcl/ns-lib.tcl

Nous avons aussi rajouté les lignes dans la figure ci-dessous dans le fichier « Makefile » dans le répertoire racine de la «ns-2.34 ».

```
SBaodv/SBaodv_rtable.o SBaodv/SBaodv_rqueue.o \  SBaodv/SBaodv_rtable.o
SBaodv/SBaodv_rqueue.o \
```

Figure 4.3: Modification du fichier Makefile

Après toutes ces modifications, nous avons compilé NS-2 à nouveau en exécutant dans le terminal les instructions « make clean », « make » pour créer les fichiers objet.

Pour le BHAODV, nous avons procédé de la même manière que SBAODV, la mise en œuvre du comportement malicieux est faite sur la couche routage et cela par la modification du protocole de routage AODV.

3.4. Les performances réseaux :

Nous avons évalué notre solution, en utilisant les métriques suivantes :

- *Le Taux de Délivrance (TD)* : c'est le rapport entre le nombre des paquets de données bien reçus par les nœuds destinations à celui généré par les nœuds sources.
- *Le Surdébit de Routage Normalisé (SRN)* :c'est le ratio entre le nombre de paquets du contrôle (RREQs, RREPs, RERRs) générés par le protocole de routage et le nombre de paquets de données bien reçus.
- *Le Débit* : c'est la quantité d'informations transmises par unité de temps
- *Les paquets de données reçus par les destinations*

3.5. Les résultats de simulation :

Après cette phase d'implémentation de la solution sous NS2.34, nous avons développé un script TCL [Kevin Fall et al, 2011] permettant de configurer et d'exécuter les différentes simulations. Les résultats de chaque simulation sont enregistrés dans un fichier trace (.tr) spécifié dans le script TCL.

Nous avons généré un réseau de 20 nœuds et créé une connexion UDP entre les nœuds, nous avons attaché l'application CBR (Constant Bit Rate) qui génère des paquets constants à travers la connexion UDP. La taille des paquets CBR est choisie pour être 512 octets. Durée des scénarios est de 100 secondes et les connexions CBR commence au moment égal à 1,0 secondes et continuer jusqu'à la fin de la simulation, dans un espace de 500 x 500.

Nos simulations sont réalisées en utilisant IEEE802.11 pour la couche MAC et Random Waypoint Model [VASANTHI.V et al, 2011] comme modèle de mobilité des nœuds. Dans ce dernier, un nœud mobile commence par séjourner dans un endroit pendant une certaine période de temps dite temps de pause. Une fois cette période terminé, le nœud se déplace vers une destination choisie aléatoirement et avec une vitesse de déplacement choisie dans

l'intervalle [minspeed, maxspeed]. Une fois cette destination atteinte, il reste immobile pendant le temps de pause spécifié, puis réitère le processus. Pour réaliser cela on utilise l'utilitaire ./setdest de NS2, qui est le générateur aléatoire de scénarios de mouvement des nœuds. Ainsi pour générer des modèles de trafic aléatoire on utilise l'utilitaire ./cbrgen.

Pour exploiter et filtrer les résultats voulus nous avons écrit des scripts AWK. Par la suite nous avons affiché les graphes par le programme Excel.

Le tableau ci-après résume les paramètres de notre modèle d'expérimentation :

Paramètre	Valeur
Simulateur	NS2.34
Temps de simulation	100s
Nombre des nœuds	20
Nombre des nœuds malicieux	1
Pause Time	2s
Terrain de simulation	500*500
Traffic	CBR
MAC	802_11

Tableau 4.1 : paramètres de simulation

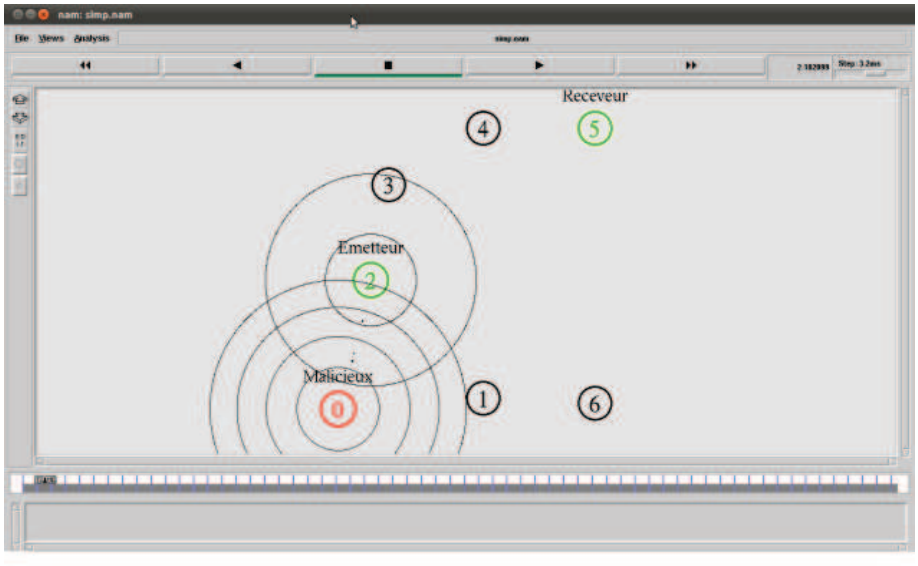

Figure 4.4 : Simulation de l'attaque Blackhole sous l'AODV

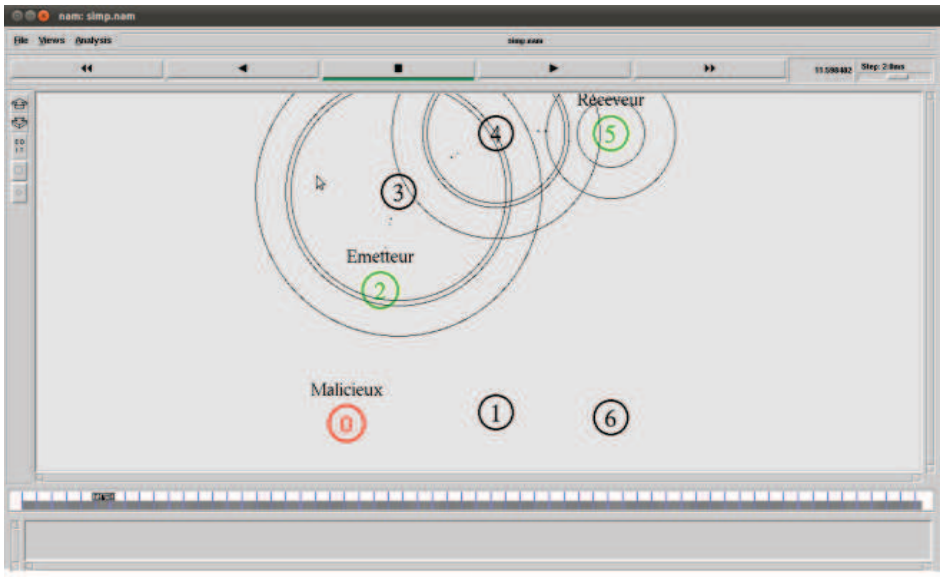

Figure 4.5 : L'attaque sous SBAODV (le nœud malicieux n'a plus d'effet)

Les figures 4.4 et 4.5 montrent la simulation du réseau avec l'attaque et sans attaque après avoir intégré la solution respectivement.

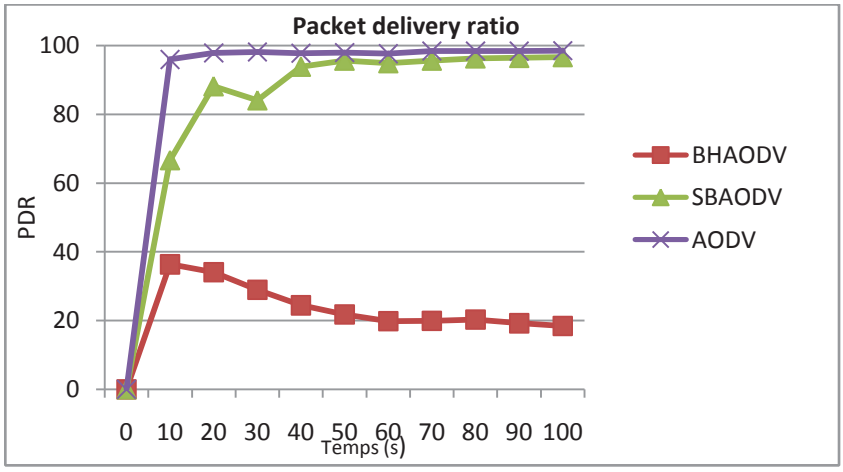

Figure 4.6 : Le taux de délivrance des paquets vs Temps

La figure 4.6 montre une évolution croissante du taux de délivrance de paquets en fonction du temps, on constate que le PDR de SBAODV après la suppression de l'attaque est nettement supérieur à celui de BHAODV, et avec le temps il devient approximativement égal à celui de l'AODV

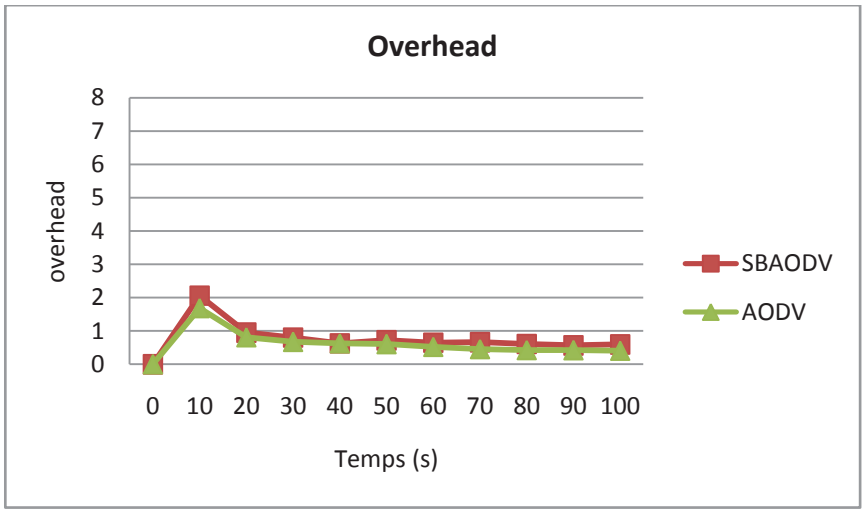

Figure 4.7 : La charge réseau vs Temps

La figure 4.7 illustre la charge de routage en fonction du temps, elle est légèrement plus de l'AODV.

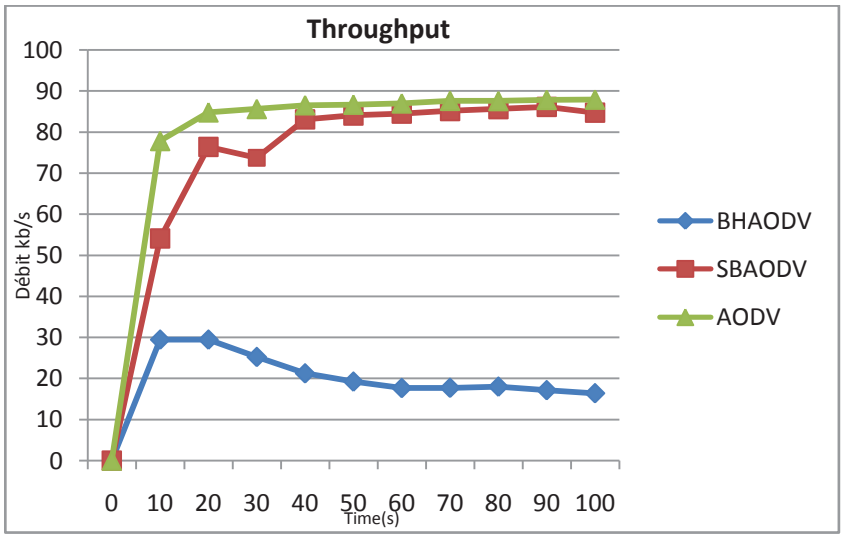

Figure 4.8 : Le débit vs Temps

La figure 4.8 montre le débit en fonction du temps, pour le protocole BHAODV il est très faible, après l'intégration de notre module on remarque que le débit commence à augmenter progressivement.

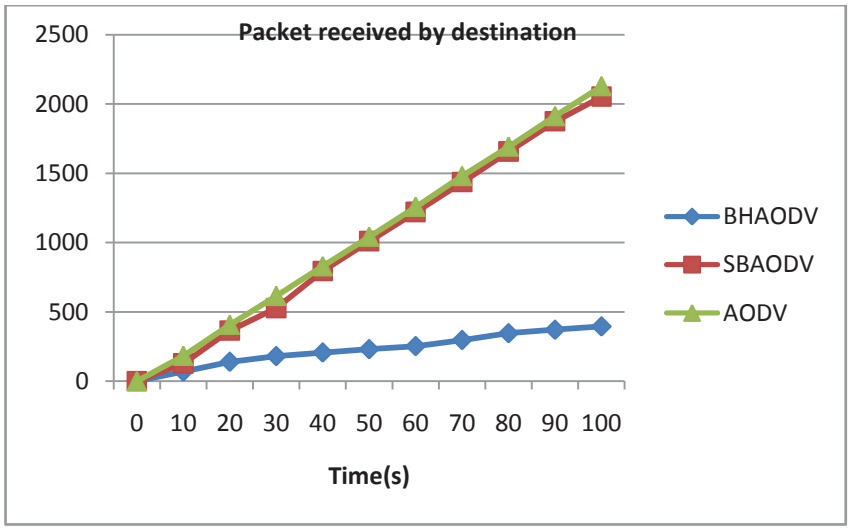

Figure 4.9 : Les paquets reçus par les destinations vs Temps

Dans la figure 4.9, on calcule le nombre de paquets de données envoyés par les nœuds légitimes et reçus par leurs réelles destinations. La courbe de l'AODV sous attaque est très inférieure par rapport aux deux autres. En effet, on voit très bien que l'attaquant a réussi d'isoler les nœuds légitimes et absorber le trafic, les paquets reçus dans ce cas sont ceux des nœuds qui sont loin du nœud malicieux, puisque on a utilisé 20 nœuds, si on réduit le nombre de nœuds (à 7 par exemple) on a bien remarqué que la courbe de BHAODV deviendra 0.

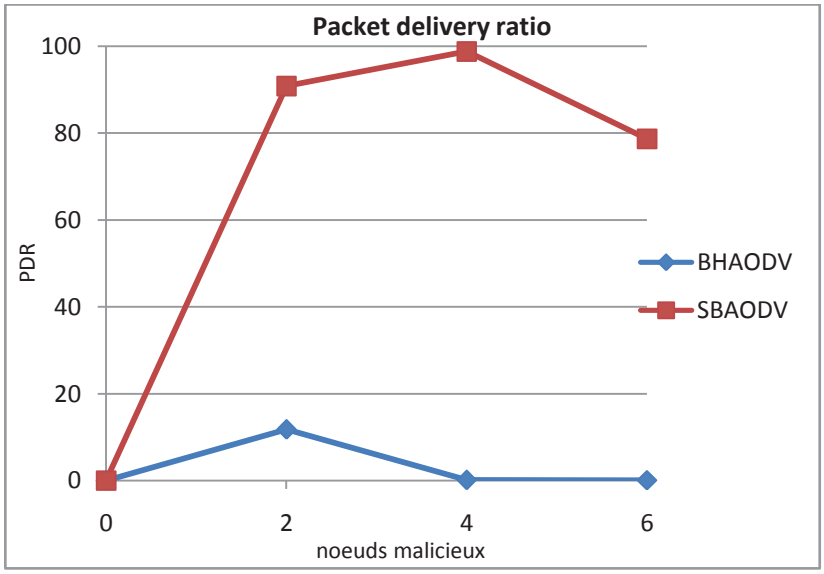

Figure 4.10 : PDR vs Nombre de nœuds malicieux

Comme nous pouvons le constater dans la figure 4.10, avec l'augmentation du nombre de nœuds malicieux le taux de délivrance de paquets pour le BHAODV tend très vite vers zéro, le PDR pour le protocole SBAODV est bien supérieur à celui de BHAODV et il commence à baisser quand le nombre de nœud malicieux vaut 4.

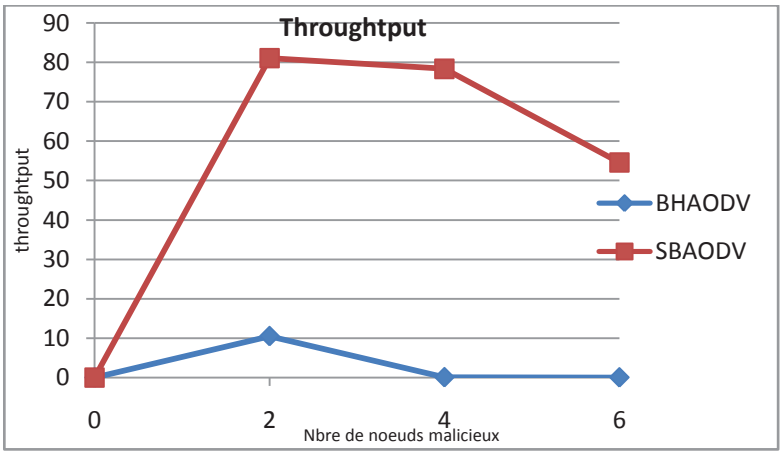

Figure 4.11: Débit vs Nombre de nœuds malicieux

La figure 4.11 montre aussi que le débit pour le protocole BHAODV va rapidement vers le zéro en augmentant le nombre de nœuds malicieux, quant au protocole SBAODV, il commence à décroître à partir de quatre nœuds malicieux.

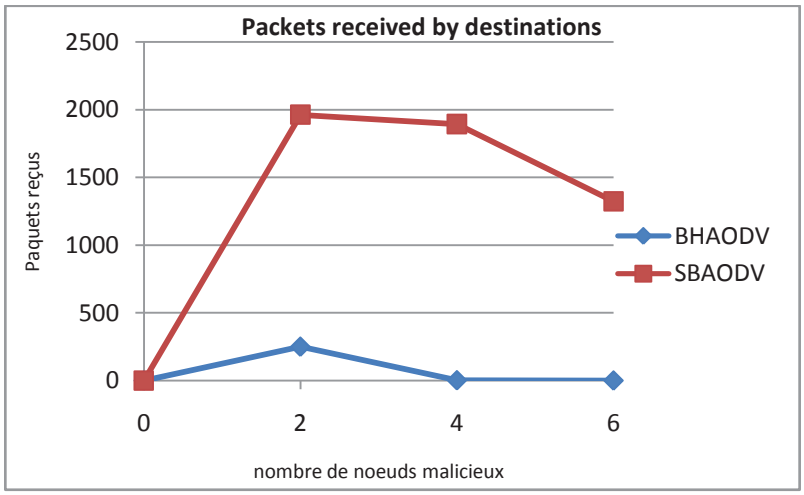

Figure 4.12 : Paquets reçus vs Nombre de nœuds malicieux

Même chose dans la figure 4.12, on remarque que le protocole SBAODV est plus performant en termes de paquets de données reçus par les destinations par rapport à BHOADV.

4. Conclusion :

Dans ce travail, nous avons mis en place un nouveau protocole de sécurité dédié aux réseaux sans fil P2P, qui est une version modifiée du protocole AODV, ce mécanisme consiste à sécuriser le processus de découverte de route, et par conséquent protéger le processus de transfert des données en se basant sur un algorithme de détection d'intrus.

Conclusion Générale

Conclusion Générale

Ce travail a été principalement axé sur la sécurité des réseaux sans fil Peer to Peer, qui représente un vrai challenge, à cause des caractéristiques de ces réseaux. Dans ce mémoire, nous avons proposé un protocole basé sur la notion de confiance pour détecter les actions malhonnêtes et sécuriser l'échange de données dans ces réseaux. Nous nous somme intéressés à la sécurité au niveau routage, plus précisément nous avons basé notre étude sur le protocole AODV.

Les caractéristiques particulières des réseaux sans fil P2P les rendent très vulnérables à plusieurs formes d'attaques. Un exemple spécifique de l'une de ces attaques est l'attaque Blackhole. Ce type d'attaque peut représenter une menace importante pour le bon fonctionnement du réseau.

Dans ce mémoire, nous nous sommes intéressés à l'analyse de l'attaque Blackhole, nous avons proposé une nouvelle approche permettant d'éviter cette attaque, la solution exploite l'activité de chaque nœud dans le réseau pour évaluer son degré de confiance, avec cette valeur le protocole détecte le comportement suspect lié à l'attaque Blackhole. Pour évaluer les performances du protocole, nous avons implémenté la solution sous le simulateur NS2. Ensuite, nous avons effectué un ensemble de simulations et nous avons présenté et interprété les résultats obtenus.

Ce projet nous a offert l'occasion de travailler sous l'environnement Linux, découvrir l'outil de simulation des réseaux NS2, découvrir et enrichir nos connaissances sur des domaines de recherche très vastes, à savoir les réseaux sans fil Peer to Peer, la sécurité des réseaux en général et les systèmes de détection d'intrusion en particulier. Grâce à notre étude, nous avons aussi constaté qu'il ne peut y avoir une sécurité absolue. Cela est dû au nombre important des facteurs qui conditionnent les performances d'un protocole sécurisé à 100%.

Finalement, nous envisageons comme perspectives du travail d'évaluer la capacité de notre proposition à résister à d'autres attaques dans des conditions supplémentaires.

Références Bibliographiques

[**A**ndroutsellis, 2004]: S. Androutsellis-Theotokis and D. Spinellis. "A survey of peer-to-peer content distribution technologies". ACM Computing Surveys, December, 2004.

[A.Mohaisen et al, 2010]: A.Mohaisen, N.Hopper, and Y.Kim, “Keep your friends close: Incorporating trust into social network-based Sybil defenses”, University of Minnesota, Minneapolis, 2010.

[Abdellaoui et al, 2009]: Abdellaoui Rachid and Jean-Marc Robert, “SU-OLSR: A NEW SOLUTION TO THWART ATTACKS AGAINST THE OLSR PROTOCOL”, Ecole de Technologie Supérieure, MONTRÉAL, 2009.

[Al-Shurman et al, 2004]: Al-Shurman, M., Yoo, S. and Park, S, "Black hole Attack in Mobile Ad Hoc Networks", ACM Southeast Regional Conference, 2004.

[**B**ing Wu et al, 2006]: Bing Wu et al, "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks", Department of Computer Science and Engineering, Florida Atlantic University, springer, 2006.

[Baptiste, 2005]: Baptiste Prêtre, “Attacks on Peer to peer networks”, Dept. of Computer Science Swiss Federal Institute of Technology (ETH) Zurich, 2005.

[B. Levine et al, 2006]: B. Levine, C. Shields, and N. Margolin, “A survey of solutions to the sybil attack,” University of Massachusetts Amherst, Tech. Rep., 2006.

[**D**IDOUX et al, 2007] : DIJOUX Alexandre et EMMA Samuel, "Peet-To-Peer", Master Professionnel Système informatique et réseaux Université Claude Bernard, Lyon, 2007.

[Deng et al, 2002]: Deng H., Li W. and Agrawal, D.P., "Routing security in wireless ad hoc networks" Communications Magazine, IEEE, October 2002.

[François, 2009] : François LESUEUR, Thèse de doctorat, université de Rennes 1, 2009.

[Hans et al, 2007]: Hans Delfs et Helmut Knebl, "Introduction to Cryptography, Principles and Applications", 2éme Edition, springer, 2007.

[Huang et al, 2004]: Y. Huang and W. Lee. Attack Analysis and Detection for Ad Hoc Routing Protocols. In *Proc. of 7th International Symposium on Recent Advances in Intrusion Detection RAID*, Springer, September 2004.

[John et al, 2006]: John Risson, Tim Moors “Survey of research towards robust peer-to-peer networks: Search methods”, Computer Networks, *Computer Networks*, Vol. 50, No. 17, 2006.

[J.Douceur et al, 2002]: J. Douceur and J. S. Donath, “The sybil attack,” in In Proceedings for the 1st International Workshop on Peer-to-Peer Systems, pages 251–260, Cambridge, MA, USA, Mar. 2002.

[J.Newsome et al, 2004]: J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proc. Of the third int. symposium on Information processing in sensor networks*, pages 259–268, New York, NY, USA, 2004.

[K.LakSBmi et al, 2010]: K.LakSBmi et al, “Modified AODV Protocol against Blackhole Attacks in MANET”, International Journal of Engineering and Technology Vol.2 (6), 2010, 444-449, 2010.

[Kevin Fall et al, 2011]: Kevin Fall et Kannan Varadhan “The *ns* Manual”, The

[Lin Wangm, 2006]: Lin Wangm, "Attacks Against Peer-to-peer Networks and Countermeasures", Helsinki University of Technology, 2006.

[Lalit Himral et al, 2011]: Lalit Himral, Vishal Vig, Nagesh Chand, "Preventing AODV Routing Protocol from Black Hole Attack" International Journal of Engineering Science and Technology (IJEST), May 2011.

[Liorens et al, 2003] : C. Liorens et L. Levier, "Tableaux de bord de la sécurité réseau", *Eyrolles*, Paris-France, 2003.

[Marling et al, 2006]: Marling Engle & Javed I. Khanm,"Vulnerabilities of P2P Systems and a Critical Look at their Solutions", Computer Science Dept., Kent State University, 2006.

[Mohamed, 2011] : Mohamed Ali Ayachi, « Contributions à la détection des comportements malhonnêtes dans les réseaux ad hoc AODV par analyse de la confiance implicite », thèse de doctorat, université de Rennes 1, 2011.

[Maha et al, 2011]: Maha Abdelhaq et al, "Security Routing Mechanism for Black Hole Attack over AODV MANET Routing Protocol", Australian Journal of Basic and Applied Sciences, 5(10): 1137-1145, 2011 University Kebangsaan Malaysia, 2011.

[Pujolle G, 2008] : Pujolle G, "Les Réseaux", EYROLLES, Paris,-France, 1128, 2008.

[Pankaj Kohli et al, 2007]: Pankaj Kohli and Umadevi Ganugula, 'DDoS Attacks using P2P Networks', Centre for Security Theory and Algorithmic Research, International Institute of Information Technology, India, April 25, 2007.

[Patrick, 2007]: Patrick MARLIER, “Sécurité du Peer-to-Peer”. [en ligne] www.labo-asso.com

[P.Ning et al, 2003]: P. Ning and K. Sun. How to misuse AODV : a case study of insider attacks against mobile ad-hoc routing protocols. In *Proc. IEEE Systems, Man and Cybernetics Society, Information Assurance Workshop (IAW’03)*. IEEE, June 2003.

[Payal N et al, 2009]: Payal N. Raj1 and Prashant B. Swadas2, “DPRAODV: A Dynamic Learning System against Blackhole Attack in AODV based MANET”, IJCSI International Journal of Computer Science Issues, 2009.

[Raddad AL KING, 2010] : Raddad AL KING, "Localisation de sources de données et optimisation de requêtes réparties en environnement pair-à-pair", thèse de doctorat, université de Toulouse, 2010.

[R.Housley et al, 1999]: R. Housley, W. Ford, W. Polk and D. Solo, “Internet X.509 Public Key Infrastructure: Certificate and CRL, Profile”, *RFC 2459*, 1999.

[Rutvij et al, 2011]: Rutvij Jhaveri et al, “Security and Service Discovery Issues in Mobile Ad-hoc Networks”, International Journal of Networking Volume 1, Issue 1, 2011.

[Ros et al, 2004]: F. J. Ros and P. M. Ruiz, “Implementing a New Manet Unicast Routing Protocol in NS2”, Dept. of Information and Communications Engineering, University of Murcia, December 2004.

[Schollmeier, 2001]: R. Schollmeier. "A definition of peer-to-peer networking for the classification of peer-to peer architecture and applications". In Proceedings of the 1st International Conference on Peer-to-Peer Computing - IEEE Computer Society, 2001.

[Subah et al, 2011]: Subash Chandra Mandhata, Dr.Surya Narayan Patro, “A counter measure to Black hole attack on AODV- based Mobile Ad-Hoc Networks” International Journal of Computer & Communication Technology (IJCCT), 2011.

[Tiwari, 2010]: Tiwari Harshvardhan, "Cryptographic Hash Function: An Elevated View", European Journal of Scientific Research, 2010.

[Tamilselvan et al, 2007]: Tamilselvan, L.; Sankaranarayanan, V., "Prevention of Blackhole Attack in MANET," The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, Aus Wireless 2007.

[Tebbane et al, 2004]: N. Tebbane, S.Tebbane, A. Mehaoua, “Simulation et Mesure des performances du protocole de routage AODV,” JTEA'2004, Hamamet, Tunisia 2004.

[Véronique et al, 2004]: Véronique Legrand et Stéphane Ubéda ″Vers un modèle de confiance pour les objets communicants : une approche sociale″, Laboratoire CITI, INRIA ARES, 2004.

[VASANTHI.V et al, 2011]: VASANTHI.V et al, “A DETAILED STUDY OF MOBILITY MODELS IN WIRELESS SENSOR NETWORK”, Journal of Theoretical and Applied Information Technology, 2011.

[W.Trappe et al, 2005] : W. Trappe and L.C.Washington. “Introduction to Cryptography with Coding Theory” (2nd Edition). Prentice Hall, 2005.

Pages web: The network simulator - *ns-2*, available at *http://www.isi.edu/nsnam/ns/*