République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaia

Faculté des Sciences exactes

Département Informatique

Mémoire de fin d'études

En vue de l'obtention du diplôme de Master Recherche en Informatique

Option : Réseaux et Système Distribués

Thème

Accord de clef de groupe dans les réseaux mobiles Ad Hoc

Présenté par

M^r^ ***KIROUANI*** **Youba**

M^r^ ***LATBI*** **Ghilas**

Soutenu devant le jury composé de :

Président	M^r^ ***SIDER*** **Abderrahmane**
Promotrice	M^me^ ***BENSAID*** **Samia**
Examinateur	M^r^ ***SAADI*** **Mustapha**
Examinateur	M^r^ ***LARBI*** **Ali**

Promotion 2012/2013

Dédicaces

A *la mémoire de ma grande mère*

A *ma mère*

A *mon père*

A *mes frères et sœurs*

A *toute ma famille*

A *mes amis et collègues*

KIROUANI Youba

Dédicaces

***A** la mémoire de mon regretté frère*

***A** mon cher père*

***A** ma très chère mère*

***A** mes frères et sœurs*

***A** mon adorable poussin "Sylia"*

***A** toute ma famille*

***A** mes amis et camarades, et tous ceux qui m'ont aidé*

LATBI Ghilas

Remerciements

Nous remercions tout d'abord Dieu le tout puissant de nous avoir donné le courage, la santé, la force et la patience pour réaliser ce modeste travail.

Nous tenons à remercier notre promotrice Mme BENSAID Samia pour son aide précieuse, ses conseils, sa patience ainsi que son soutien constant qui nous a permis de mener à bien ce modeste travail.

Nous tenons à remercier aussi Mr Omar MAWLOUD, chef de département informatique, d'avoir été là pour nous et de lui exprimer notre immense gratitude pour ses conseils, sa générosité, sa disponibilité et sa patience.

Nous tenons à exprimer notre gratitude aux membres de jury pour avoir accepté de juger ce travail.

Un grand merci à tous nos collègues en Master 2 et tous les enseignants du département Informatique, qui nous ont offert un environnement de travail extrêmement agréable.

Un grand merci à nos familles, pour leur soutien qui nous a poussé à chercher au fond de nous la volonté de faire toujours beaucoup plus, à nos amis et tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

A

AES	**A**dvanced **E**ncryption **S**tandard
AODV	**A**d Hoc **O**n Demand **D**istance **V**ector
AT-GDH	**A**rbitrary **T**opology **G**eneralisation of **D**iffie-**H**ellman

B

BD	**B**urmester **D**esmedt

C

CA	**C**ertification **A**uthority
CDS	**C**onnected **D**ominating **S**et
C-H	**C**ircular **H**ierarchical
CRT	**C**hinese **R**emainder **T**heorem
CRTDH	**C**hinese **R**emainder **T**heorem and **D**iffie-**H**ellman

D

DES	**D**ata **E**ncryption **S**tandard
DH	**D**iffie **H**ellman
DoS	**D**enial **o**f **S**ervice
DSDV	**D**estination-**S**equenced **D**istance **V**ector
DSR	**D**ynamic **S**ource **R**outing

G

GK	**G**roup **K**ey

H

Hdr **H**eader
HSR **H**ierarchically **S**egmented **R**outing

I

IGMPv3 **I**nternet **G**roup **M**anagement **P**rotocol **v**ersion 3
IP **I**nternet **P**rotocol

K

KGC **K**ey **G**eneration **C**enter

L

LCM **L**east **C**ommon **M**ultiple

M

MAC **M**edium **A**ccess **C**ontrol
MANET **M**obile **A**d Hoc **NET**work
MD2 **M**essage-**D**igest Algorithm 2
MD4 **M**essage-**D**igest Algorithm 4
MD5 **M**essage-**D**igest Algorithm 5
MK **M**aster **K**ey
MLDv2 **M**ulticast **L**istener **D**iscovery **v**ersion 2

O

OLSR **O**ptimized **L**ink **S**tate **R**outing Protocol

P

PC **P**ersonal **C**omputer
PDA **P**ersonal **D**igital **A**ssistant
PK **P**ublic **K**ey
PKI **P**ublic **K**ey **I**nfrastructure

PKG **P**rivate **K**ey **G**enerator

R

RSA **R**ivest **S**hamir et **A**dleman

S

SN **S**ervice **N**ode
STR The **S**kinny **T**ree Protocol

T

TEK **T**rafic **E**ncryption **K**ey
TFAN **T**ree-based Group Key Agreement **F**ramework for Mobile **A**d Hoc **N**etworks
TGDH **T**ree-Based **G**roup **D**iffie-**H**ellman
TORA **T**emporally-**O**rdered **R**outing **A**lgorithm

Z

ZRP **Z**one **R**outing **P**rotocol

Introdution Générale

Les technologies sans fil offrent aujourd'hui de nouvelles perspectives dans le domaine des télécommunications. Depuis leur émergence dans les années 70, les réseaux sans fil ont gagné un intérêt majeur et une popularité croissante. Ils occupent de plus en plus de place dans les communications personnelles et d'entreprise.

Les réseaux mobiles sans fil, peuvent être classés en deux grandes catégories : les réseaux avec infrastructures qui utilisent généralement le modèle de la communication cellulaire, et les réseaux sans infrastructure ou les réseaux mobiles Ad Hoc.

Un réseau mobile Ad Hoc est une collection d'entités mobiles interconnectées par une technologie sans fil formant un réseau temporaire sans l'aide de toute administration centralisée ou de support fixe. Les caractéristiques des réseaux mobiles Ad Hoc, comme l'absence d'infrastructure et l'utilisation d'un canal radio rendent la sécurité un véritable défi technologique auquel nous sommes confrontés.

Etablir des communications de groupe sécurisées au sein d'un réseau mobiles Ad Hoc est un véritable challenge. En effet, un réseau Ad Hoc est un environnement hostile, qui apporte plusieurs défis de sécurité, dus à ses caractéristiques (liens sans fil, capacités limitées, ...). A ces contraintes de sécurité, s'ajoutent les vulnérabilités du modèle IP multicast qui élimine toute possibilité d'identification des membres de groupe ou de confidentialité de données. Le déploiement des communications de groupe dans un réseau Ad Hoc induit également de nouvelles épreuves à prendre en compte lors de la conception d'une architecture de gestion de clef dans les réseaux mobiles Ad Hoc.

La solution la plus appropriée pour assurer des communications de groupe dans les MANETs est l'établissement d'un protocole d'accord de clef de groupe. Ce protocole doit garantir la confidentialité des données multicast en assurant la gestion et la distribution de la clef de chiffrement de données TEK (*Trafic Encryption Key*). Le contrôle d'accès au groupe multicast est également assuré car seuls les membres détenant la clef du groupe peuvent accéder au flux multicast émis par la source. Le protocole d'accord de clef de

groupe doit également être adapté à la nature et aux caractéristiques des réseaux mobiles Ad Hoc.

Ce projet de fin d'études entre dans ce contexte et il s'intéresse à l'étude de la sécurité des communications de groupe dans les réseaux mobiles Ad Hoc. Ce manuscrit est organisé comme suit : un premier chapitre dans lequel nous présentons les réseaux Ad Hoc à savoir leurs caractéristiques et leurs domaines d'application. Nous décrivons dans le deuxième chapitre les risques et les menaces de sécurité dans les réseaux mobiles Ad Hoc ainsi que le principe de communication de groupe. Le troisième chapitre donne un aperçu sur les protocoles d'accord de clef de groupe dans les réseaux mobiles Ad Hoc. Une proposition d'une amélioration et d'une nouvelle approche pour l'accord de clef de groupe sera décrite dans le chapitre quatre.

Nous achevons ce manuscrit par une conclusion générale qui résume nos objectifs estimés et les limites de notre approche tout en évoquant les problèmes que nous avons rencontrés et les améliorations envisageables.

Chapitre 1

Présentation des réseaux mobiles Ad Hoc

1.1 Introduction

Dans nos jours, le besoin à plus de mobilité et à pouvoir partager ou échanger de l'information a tout moment, en utilisant des dispositifs mobiles (téléphones portables, PDA, PC portables, etc....) a rendu très répandu la notion de réseau sans infrastructure, ou réseaux Ad Hoc.

Ce chapitre est composé de deux parties. Dans la première partie, nous présentons les réseaux Ad Hoc à savoir leurs caractéristiques (absence d'infrastructure, topologie dynamique, contraintes d'énergie,...etc.). La deuxième partie est consacrée pour une étude des risques et des menaces de sécurité dans les réseaux mobiles Ad Hoc.

1.2 Les réseaux mobiles Ad Hoc

1.2.1 Définition

Un réseau mobile Ad Hoc, appelé généralement MANET (*Mobile Ad Hoc NETwork*) [22], est un réseau composé de nœuds mobiles ayant aucune infrastructure ou administration centralisée dont le seul moyen de communication est l'utilisation des interfaces sans fil. La topologie du réseau peut changer à tout moment [31], elle est donc dynamique et imprévisible ce qui fait que la déconnexion des unités soit très fréquente.

1.2.2 Caractéristiques des réseaux mobiles Ad Hoc

En tant que nouveau paradigme des réseaux sans fil, les réseaux Ad Hoc présentent de nombreuses caractéristiques dont certaines leur sont bien spécifiques et les différencient

des autres types de réseaux sans fil :

Une topologie dynamique **:** les unités mobiles du réseau se déplacent d'une façon libre et arbitraire. Par conséquent, la topologie du réseau peut changer, à des instants imprévisibles, d'une manière rapide et aléatoire, ce qui fait que la déconnexion des unités soit très fréquente.

Une bande passante limitée **:** l'utilisation d'un médium de communication partagé (ondes radio) est une caractéristique primordiale des réseaux basés sur la communication sans fil. Ce partage fait que la bande passante réservée à un hôte soit modeste.

Rapidité de déploiement **:** les réseaux Ad Hoc peuvent être facilement installés dans les endroits difficiles à câbler. Par exemple, il est possible d'utiliser un réseau dans des situations d'urgence, pour organiser les secours lors d'une catastrophe naturelle.

Des contraintes d'énergie **:** les hôtes mobiles sont alimentés par des sources d'énergie autonomes donc restreintes, comme les batteries ou les autres sources consommables, dont la durée de vie est limitée.

Erreur de transmission **:** les erreurs de transmission radio dans les réseaux mobiles Ad Hoc sont plus fréquentes que dans les réseaux filaires. Cela est dû à leur sensibilité aux bruits externes.

Nœuds cachés **:** ce phénomène est très particulier à l'environnement sans fil. Un exemple est illustré par la FIGURE 1.1. Dans cet exemple, les nœuds B et C ne s'entendent pas, à cause d'un obstacle qui empêche la propagation des ondes. Les mécanismes d'accès au canal vont permettre alors à ces nœuds de commencer leurs émissions simultanément. Ce qui provoque des collisions au niveau du nœud A.

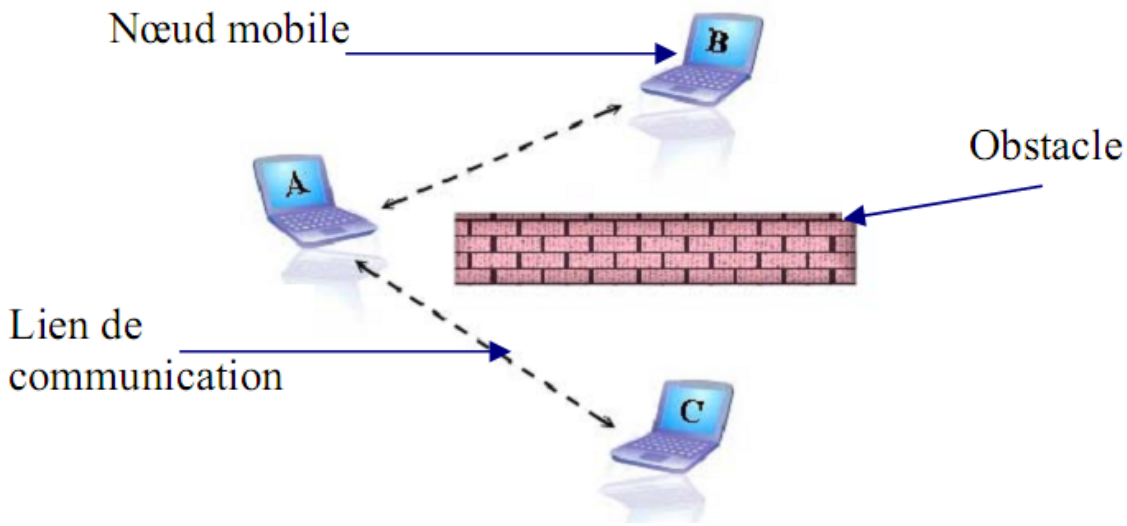

FIGURE 1.1 – *les nœuds cachés.*

Une sécurité physique limitée **:** les réseaux mobiles Ad Hoc sont plus touchés par le paramètre de sécurité, que les réseaux filaires classique. De part la nature immatérielle du support physique, l'écoute clandestine sur un réseau sans fil est facile. Ces réseaux sont également vulnérables aux attaques actives telles que la destruction où la modification non autorisée des données. Il faut donc protéger l'accès aux informations qui circulent dans le réseau.

Absence d'infrastructure **:** les réseaux Ad Hoc se distinguent des autres réseaux mobiles par l'absence d'infrastructure préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.

Communication par lien radio **:** les communications entre les nœuds se font par l'utilisation d'une interface radio. Il est alors important d'adopter un protocole d'accès au médium qui permet de bien distribuer les ressources radio et ceci en évitant le plus possible les collisions et les interférences. Les technologies de communication sans fil sont alors indispensables à la mise en place d'un réseau Ad Hoc.

Equivalence des nœuds du réseau **:** dans un réseau classique, il existe une distinction entre les nœuds terminaux (stations, hôtes) qui supportent les applications et les nœuds internes (routeurs par exemple) du réseau, en charge de l'acheminement des données. Cette différence n'existe pas dans les réseaux Ad Hoc car tous les nœuds peuvent être amenés à assurer la fonction du routage.

1.2.3 Domaines d'application des réseaux mobiles Ad Hoc

Bien qu'historiquement, les premiers projets relatifs aux réseaux Ad Hoc aient débuté dans un cadre militaire, mais leurs domaines d'application s'accordent bien au-delà du cadre militaire.

Les réseaux Ad Hoc sans fil offrent une grande flexibilité ainsi qu'une facilité de mise en place. Ils seront alors d'un grand apport lors des incendies et des catastrophes naturelles, où il sera indispensable de disposer rapidement d'un réseau pour organiser les secours et les opérations de sauvetages [1].

D'une façon générale, les réseaux Ad Hoc sont utilisés dans toute application ou le déploiement d'une infrastructure réseau filaire est trop contraignant, à cause de la difficulté de sa mise en place, et sa durée d'installation.

1.2.4 Modélisation d'un réseau Ad Hoc

Un réseau Ad Hoc peut être modélisé par un graphe non orienté $G_t = (V_t, E_t)$ ou V_t représente l'ensemble des nœuds (i.e. les unités mobiles) du réseau et E_t modélise l'ensemble des connexions qui existent entres ces nœuds. Si $e = (u, v) \in E_t$, cela veut dire que les nœuds u et v sont en mesure de communiquer directement à l'instant t [2].

La figure 1.2. Représente un réseau Ad Hoc de 07 unités mobiles modélisé sous forme d'un graphe.

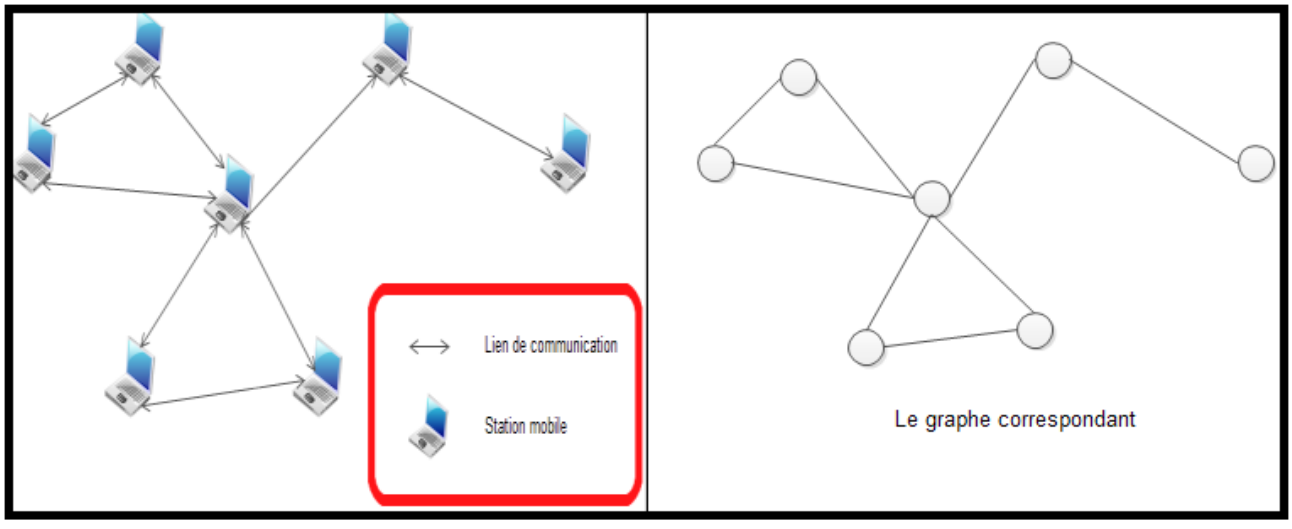

FIGURE 1.2 – *Modélisation d'un réseau Ad Hoc.*

1.2.5 Le routage dans les réseaux mobiles Ad Hoc

Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance (mobilité des nœuds, optimalité des chemins, l'énergie consommée,...etc.).

Si on suppose que les coûts des liens sont identiques, le chemin indiqué dans la figure 1.3. est le chemin optimal reliant la station source et la station destination. Une bonne stratégie de routage utilise ce chemin dans le transfert des données entres les deux stations.

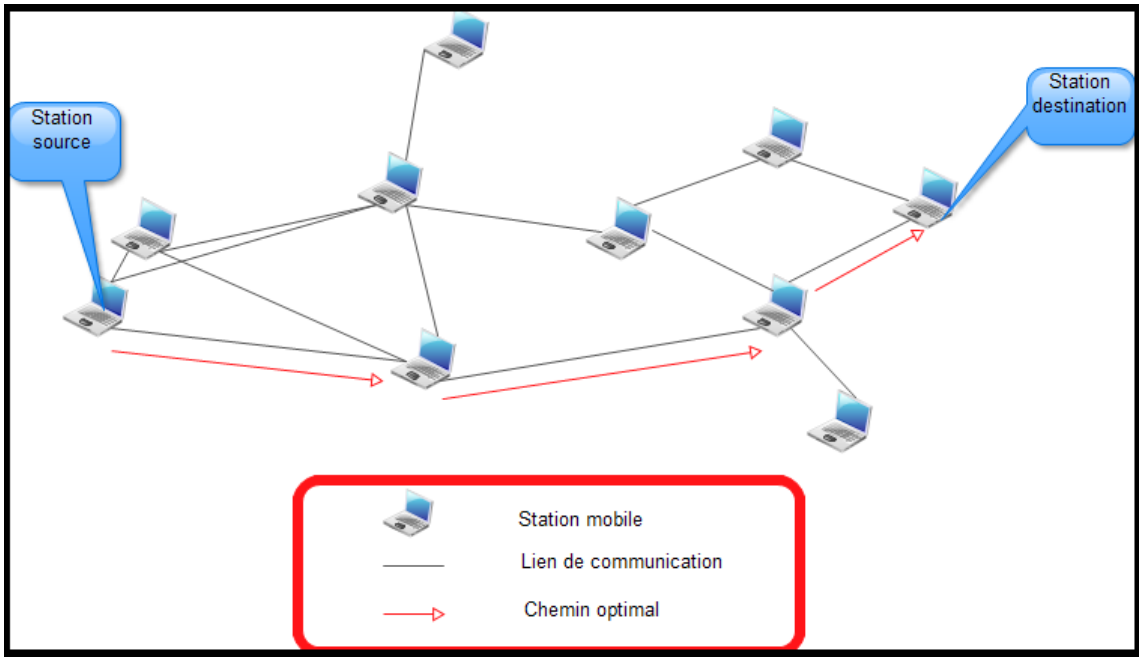

FIGURE 1.3 – *Chemin utilisé dans un routage entre la source et la destination.*

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en deux catégories, les protocoles pro-actifs et les protocoles réactifs.

Les protocoles pro-actifs établissent les routes à l'avance en se basant sur l'échange périodique des tables de routage, alors que les protocoles réactifs cherchent les routes à la demande.

Parmi les protocoles de routage pro-actifs les plus connus, on cite les protocoles DSDV [51], HSR [50] et OLSR [25]. Par contre, les protocoles réactifs les plus utilisés sont principalement les protocoles AODV [49], DSR [23] et TORA [52].

Il existe aussi des protocoles qui combinent les approches réactives et proactives, appelés protocoles hybrides. Leurs principe est de connaître le voisinage de manière proactive jusqu'à une certaine distance, puis de faire appel aux techniques réactives pour chercher des routes. Parmi ces protocoles, on peut citer le protocole ZRP [24].

1.3 Risques et menaces de sécurité dans les réseaux Ad Hoc

La sécurité des réseaux Ad Hoc présente un défi. En effet ces derniers possèdent des caractéristiques qui les rendent plus vulnérable aux attaques. Dans ce qui suit, nous énumérons leurs vulnérabilités ainsi que les attaques possibles.

1.3.1 Fonctions et données à protéger

Peut-on garantir l'intégrité des informations transmises sur les réseaux Ad Hoc ? Pour garantir ces dernières, il faut tout d'abord déterminer les ressources sensibles qu'on doit protéger.

Chaque nœud d'un réseau Ad Hoc participe dans la découverte des routes et échange des informations de routage avec les autres nœuds du réseau.

Si on suppose qu'un nœud malicieux réussit à introduire des informations de routage invalides alors tout le réseau tombe en panne (les nœuds n'arrivent pas à communiquer). Donc on peut déduire que le routage est une fonction sensible qu'il faut protéger pour garantir la disponibilité du réseau.

Les informations relatives au routage comme les nœuds accessibles et les métriques associées aux routes, les informations relatives aux mécanismes de configuration et les informations personnelles des utilisateurs sont considérées comme données sensible qu'il faut protéger.

1.3.2 Vulnérabilité des réseaux Ad Hoc

Les réseaux Ad Hoc présentent quelques faiblesses. Certaines faiblesses sont liées à la technologie sans fil, d'autre aux caractéristiques de ces réseaux. Une liste des attaques les plus probables est ainsi dressée dans cette section.

Une première classification des attaques consiste à distinguer les attaques passives des attaques actives.

Les attaques passives se limitent à l'écoute et l'analyse du trafic échangé (*Sniffing*). Ce type d'attaques est plus facile à réaliser, et il est difficile à détecter puisque l'intrus n'apporte aucune modification sur les informations échangées. L'intention de l'intrus peut être la connaissance des informations confidentielles des utilisateurs ou bien la connaissance des nœuds importants dans le réseau, en analysant les informations de routage, pour se préparer à une attaque active.

Les attaques d'écoute et d'analyse de trafic sont d'autant plus dangereuses dans les réseaux sans fils tels que les MANETs. En effet, les ondes radioélectriques ont une grande capacité à se propager dans toutes les directions avec une portée relativement grande, facilitant ainsi à une personne non autorisée, d'écouter le réseau et d'analyser le trafic. Ces attaques constituent une menace certaine pour la confidentialité des données, ainsi que pour l'anonymat des utilisateurs.

Dans les attaques actives, un intrus tente de supprimer ou de modifier les messages transmis sur le réseau. Il peut aussi injecter son propre trafic ou rejouer d'anciens messages pour perturber le fonctionnement du réseau ou provoquer un déni de service.

Parmi les attaques actives les plus connues, on peut citer [28] :

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

Attaque de Blackhole / Grayhole **:** les techniques *blackhole* et *grayhole* ont pour but de ne retransmettre aucun ou une partie seulement des paquets reçus. Pour le cas des *grayholes*, le choix des paquets retransmis n'est généralement pas lié aux hasards, le but étant par exemple de favoriser une partie du trafic. Toutefois, il est à noter que cette attaque peut être également confondue avec le fait qu'un nœud est soit surchargé, soit incapable de jouer le rôle d'un routeur, ce qui peut compliquer la détection de ce genre d'attaques [29].

Attaque du trou de ver (Wormhole) **:** cette attaque est une variante du trou noir, qui consiste à réinjecter les paquets absorbés en un autre point (souvent distant) du réseau. Elle est réalisée lorsque plusieurs nœuds sont compromis. Le but est alors

de simuler un nœud dans le voisinage. La technique de base utilisée par les nœuds compromis est l'encapsulation des messages, pour former un lien de voisinage virtuel.

Attaque de l'identité multiple (Sybil attack) **:** le principe de cette attaque est qu'un nœud se fait passer pour plusieurs nœuds potentiellement distants, créant des incohérences dans les tables de routage des nœuds voisins [27].

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

Attaque par chantage **:** elle est connue sous le nom anglais de *Blackmail attack* [26]. Un nœud malicieux fait annoncer qu'un autre nœud légitime est malicieux pour éliminer ce dernier du réseau. Si le nœud malicieux arrive à attaquer un nombre important de nœuds, il pourra perturber le fonctionnement du réseau.

Brouillage radio **:** cette attaque a pour but de perturber le canal radio en envoyant des informations inutiles sur la bande de fréquence utilisée. Ce brouillage peut être temporaire, intermittent ou permanent. Le champ d'action de cette attaque doit être pris en compte, son effet est-il suffisamment puissant pour bloquer le réseau tout entier, ou est-il limité à quelques nœuds seulement ?

Relais sélectif de paquets (Selective forwarding) **:** un nœud décide de ne pas transmettre les données de certains nœuds. La raison peut être aussi bien d'ordre énergétique, que liée à une attaque [3].

Privation de mise en veille **:** elle a pour but de consommer toutes les ressources de la victime en l'obligeant à effectuer des calculs ou à recevoir ou transmettre des données inutilement.

Les attaques physiques d'un élément valide du réseau **:** ce sont de dangereuses attaques qui compromettent des nœuds valides du réseau (destruction, altération ou changement physique d'un composant) [1].

1.4 Conclusion

L'étude effectuée sur les réseaux mobiles Ad Hoc nous a permis de connaître leurs différentes caractéristiques (absence d'infrastructure, topologie dynamique, bandes passantes limitées, sécurité physique limitée, contraintes d'énergie,...etc.), et ainsi constater que leur apparition a facilité la mise en œuvre de plusieurs applications mobiles sans infrastructure préexistante (telles que les applications militaires), mais en revanche, a laissé émerger un bon nombre de problèmes dont la sécurité de ce dernier.

A cause de leurs vulnérabilités, les réseaux Ad Hoc sont sujets de très nombreuses menaces. Certaines d'entre elles sont liées à la technologie sans fil, d'autre aux caractéristiques de ces réseaux. Pour remédier à la plupart de ces menaces, la cryptographie est un moyen efficace et puissant. Mais, pour cela les systèmes cryptographiques doivent être associés à une gestion de clefs efficace. Ces sujets seront traités en détail dans les chapitres suivants.

Chapitre 2

Sécurité des communications de groupe

2.1 Introduction

Le réseau MANET est aujourd'hui le vecteur d'avancées considérables en matière d'information et de communication. Avec la naissance de nouvelles applications qui nécessitent les services de communication de groupe telles que les conférences multimédia et le télé-enseignement, le besoin d'échanger des informations en toute sécurité est devenu indispensable.

L'accord de clef de groupe est une opération essentielle dans la sécurité des communications de groupe, il est l'un des facteurs principaux de toute architecture de sécurité des communications de groupe. En effet, il permet la distribution de la clef du groupe à la source pour chiffrer le flux multicast et aux récepteurs pour le déchiffrer, assurant ainsi la confidentialité des données. L'authentification et le contrôle d'accès des membres de groupe sont également assurés, du fait que seuls les membres autorisés qui détiennent la clef du groupe sont capables d'accéder aux données multicast émises par la source.

Ce chapitre est composé de trois parties, nous présentons les principes de communication de groupe dans un premier lieu. Ensuite, nous définissons les concepts de sécurité. Pour finir, nous décrivons l'accord et la gestion de clefs de groupe dans les environnements mobiles Ad Hoc.

2.2 Principe de communication de groupe

2.2.1 La notion de groupe

L'organisation en groupe d'un système consiste à réunir certains objets dans des groupes coopérant et communicant. La notion de groupe multicast est un ensemble de machines représentées (désignées) par une adresse IP unique dite multicast (*adresse du groupe*). L'ensemble des membres d'un groupe est entièrement dynamique, cela signifie que toute machine peut, à tout moment, joindre le groupe ou bien le quitter. Il n'y a aucune restriction sur le nombre de membres d'un groupe ainsi que sur leur localisation physique. Une machine peut être en même temps membre de plusieurs groupes.

Un groupe multicast peut être permanent ou non. Un groupe permanent est en fait un groupe qui possède une adresse assignée par une autorité.
Il n'est toutefois pas nécessaire d'être membre d'un groupe pour y envoyer des messages.

2.2.2 Le modèle IP multicast

La transmission en multicast est le mécanisme le plus efficace et approprié pour les applications orientées vers un groupe d'utilisateurs. Le modèle IP multicast définit par DEERING [39], est une extension du modèle IP. Il définit la notion de groupe, les types d'adressage et le protocole d'adhésion au groupe. Il fut proposé comme une solution pour gérer les communications de groupe.

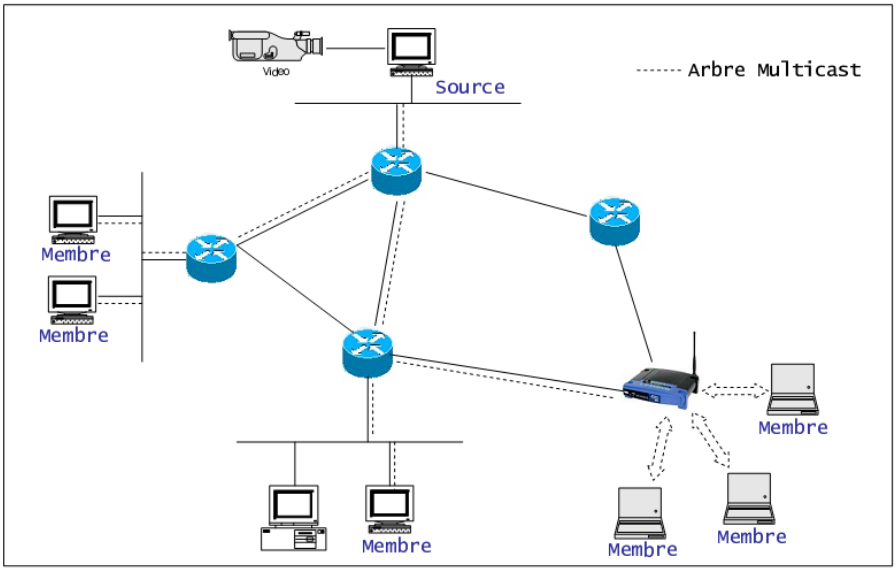

FIGURE 2.1 – *Le modèle IP multicast.*

Deux composantes sont essentielles au fonctionnement du multicast :

1. Un protocole de routage permettant de construire un arbre de diffusion afin d'acheminer les données vers les membres du groupe.
2. Un protocole de gestion de groupes qui se charge des requêtes d'adhésions et de départ tels que MLDv2 [40] ou IGMPv3 [39].

Le modèle multicast est attractif, efficace et adapté au passage à l'échelle. Cependant, ses caractéristiques présentent quelques vulnérabilités auxquelles doivent faire face les services de sécurité à instaurer pour assurer des communications de groupe sécurisées. En effet, dans le modèle IP multicast, aucune identification des hôtes participants au groupe n'est effectuée.
Les adresses des groupes multicast sont publiquement connues ; toute entité du réseau peut ainsi joindre le groupe ou le quitter sans y avoir été invitée ou accéder au flux de données multicast envoyé par la source. Une entité malveillante peut également envoyer des données multicast aux membres du groupe, sans en être adhérente et sans aucune autorisation ou contrôle d'accès. Ces actions peuvent engendrer ses attaques de dénis de services (DoS) et d'atteinte à la confidentialité et la disponibilité de données.

2.3 Concepts de sécurité

2.3.1 La cryptographie

La cryptographie est la science utilisée par les mathématiciens pour chiffrer et déchiffrer les données. L'opération de chiffrement transforme un texte en clair en un texte chiffré appelé cryptogramme, au moyen d'une clef (*qu'on dénomme la clef de chiffrement*). Le déchiffrement est le traitement des données qui permet de reconstruire le texte clair à partir du texte chiffré au moyen d'une clef. (voir figure 2.2).

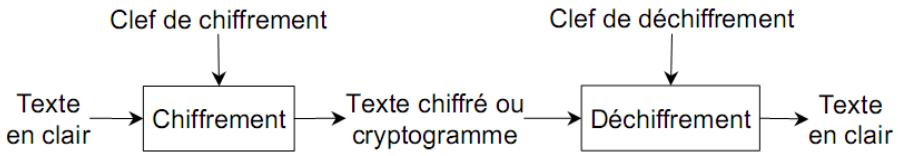

FIGURE 2.2 – *Chiffrement et déchiffrement.*

On distingue deux types de cryptographie : la cryptographie symétrique et la cryptographie asymétrique.

La cryptographie à clef symétrique

Dans la cryptographie symétrique, appelée aussi chiffrement à clef secrète, la même clef est utilisée pour le chiffrement et le déchiffrement. Il faut donc que les deux parties partagent une clé unique (voir figure 2.3). Ainsi, l'expéditeur chiffre son message avec une clef et à la réception du message chiffré, le destinataire déchiffre avec la même clef.

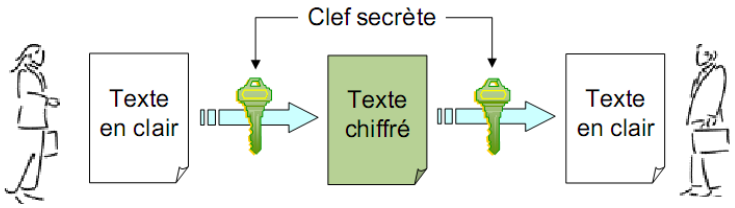

FIGURE 2.3 – *Cryptographie à clé symétrique.*

Le chiffrement à clef symétrique a l'avantage d'être rapide car le nombre de clefs ainsi que les calculs sont réduits. Mais le problème réside dans la manière d'envoyer cette unique clef de chiffrement et de déchiffrement à tous les utilisateurs de façon sécurisée. Parmi les algorithmes de chiffrement symétrique les plus utilisés nous citons DES [46], Triple DES [44] et AES [47].

La cryptographie à clef asymétrique

Dans la cryptographie asymétrique (voir figure 2.4), appelé aussi chiffrement à clef publique, chaque interlocuteur, qui désire communiquer des données confidentielles, possède un couple de clefs : une clef visible appelée clef publique utilisée pour le chiffrement, et une clef secrète appelée clef privée utilisée pour le déchiffrement.

Dans le chiffrement à clef publique, si un texte est chiffré avec la clef publique de l'utilisateur A, il ne sera déchiffré que par la clef privée de A et s'il est chiffré avec la clef privée de A, il ne sera déchiffré qu'avec la clef publique de A.

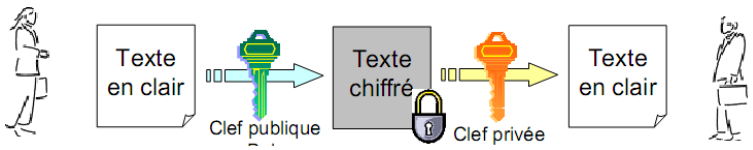

FIGURE 2.4 – *Cryptographie à clef asymétrique.*

Le principal avantage de la cryptographie asymétrique est qu'elle permet à des utilisateurs n'ayant pas d'accord de sécurité préalable d'échanger des messages de manière sûre. L'algorithme RSA [45] est un exemple de chiffrement asymétrique.

2.3.2 Les services de la sécurité

La sécurité informatique se base sur des services qui permettent de faire face à certaines menaces telles que l'écoute, la création ou la modification des données et la répudiation. Ces services de sécurité sont des propriétés que certaines applications exigent et que les protocoles de sécurité utilisés dans ces applications devraient assurer. Parmi ces services[3] :

La confidentialité : il s'agit de garantir le secret de l'information transmise ou archivée. La confidentialité des données dans une communication réseau fournit une protection contre l'analyse du trafic. Ainsi, les données transportées ne peuvent pas être lues par un adversaire espionnant les communications.

L'authentification : il s'agit de garantir l'origine d'une information. En général, l'authentification d'un participant veut dire qu'un agent doit être sûr de l'identité de son correspondant. Différents mécanismes sont utilisés pour assurer l'authentification telle que la signature numérique. On utilise la signature numérique avec un couple de clefs, dont celle permettant de créer les signatures est gardée secrète, et dont l'autre permettant de vérifier la signature est rendue publique.

Le contrôle d'accès : c'est la fonction qui consiste à prévoir les accès non autorisés. Autrement dit, c'est limiter l'accès aux utilisateurs autorisés.

L'intégrité : garantir l'intégrité consiste à assurer que les données sont transmises de la source à la destination sans modifications. Si un intrus modifie les données d'origines envoyées par l'expéditeur, le destinataire devrait être capable de détecter qu'il y a eu des changements.

La non répudiation : la non répudiation est la possibilité de protéger les utilisateurs contre le déni d'envoi et prouver que quelqu'un a bien réalisé une action.

La disponibilité : la disponibilité consiste à garantir que le service offert par le nœud doit être disponible pour les utilisateurs.

L'anonymat : la propriété de l'anonymat procure à un participant la possibilité de faire les transactions sans se faire remarquer, et qui ne peuvent pas être dépistées

par un autre participant.

2.3.3 Les fonctions de hachage

Les fonctions de hachage sont des fonctions à sens unique, et produisent à partir d'un texte d'une longueur quelconque, un condensé (aussi appelée *empreinte* ou *hash*) de longueur fixe, qu'on peut voir comme un résumé du texte (à partir duquel on ne peut pas retrouver le texte d'origine, ou du moins très difficilement [1]).

Les fonctions de hachage doivent être résistantes aux collisions, c'est-à-dire qu'on ne peut pas trouver deux messages différents ayant la même empreinte. Bien évidemment, la taille de l'empreinte étant fixe, il existera toujours des collisions du fait qu'il y a un nombre limité d'empreintes alors que le nombre de messages initiaux est illimité. On peut remarquer que la plupart des fonctions de hachage sont très sensibles ; si l'on change un seul caractère à un message, son empreinte sera totalement changée.
Les fonctions de hachage les plus connues sont : MD2 [43], MD4 [42] et MD5 [41].

2.3.4 Les signatures numériques

Les procédés de signatures sont également appelés signatures électroniques (*digital signatures*). Le but recherché n'est pas seulement la confidentialité, mais l'authenticité du message transmis (fonction d'authentification) et vérifier l'intégrité du message reçu.

Les signatures numériques s'appuient sur la cryptographie à clef publique. Un nœud possède une clef publique qui sert à ses correspondants pour chiffrer des messages qui lui sont destinés, et une clef privée pour déchiffrer les messages qu'il reçoit. Dans le cas de la signature, le nœud utilise sa clef privée pour signer un message, le destinataire du message déchiffre la signature avec la clef publique de l'émetteur.

2.3.5 Le protocole de Diffie-Hellman

L'échange de clefs Diffie-Hellman, du nom de ses auteurs *Whitfield Diffie* et *Martin Hellman*, est un protocole qui permet à deux personnes nommées conventionnellement *Alice* et *Bob* de se mettre d'accord sur un nombre (qu'ils peuvent utiliser comme clef secrète). Les étapes du protocole sont les suivantes [17] :

1. *Alice* et *Bob* choisissent publiquement deux entiers : un nombre premier g supérieur à 2 et p plus petit que g. p et g sont appelés les paramètres de Diffie-Hellman.

1. Par très difficile, nous entendons qui n'est pas faisable en pratique, car cela demanderait beaucoup de temps de calculs, même en réunissant la puissance de calculs de tous les ordinateurs.

2. Chaque participant génère confidentiellement un nombre aléatoire plus petit que *p-1* qui constitue sa clef privée : *Alice* génère x_1 et *Bob* génère x_2.
3. Les deux participants calculent alors une clef publique : *Alice* calcule $y_1 = g^{x_1} \bmod p$ et *Bob* calcule $y_2 = g^{x_2} \bmod p$.
4. *Alice* et *Bob* s'échangent leurs clefs publiques. Chacun d'eux peut calculer alors : $z = (y_1)^{x_2} \bmod p = (y_2)^{x_1} \bmod p$ qui est définie comme clef secrète entre les deux participants.

Même si le protocole de Diffie-Hellman résiste aux attaques passives, il reste vulnérable face aux attaques actives. En effet, si un intrus réussit à se faire passer pour *Bob* auprès d'*Alice* et vice-versa, il pourra partager une clef avec eux et ainsi intercepter les messages qu'ils vont s'échanger par la suite. Cette attaque, plus connue sous le nom de " *man-in-the-middle*", a fait l'objet d'une mise à jour du procédé de Diffie-Hellman. Dans cette nouvelle version, les messages sont authentifiés, ce qui permet d'éviter à un intrus de se faire passer pour l'un des deux participants lors de l'échange de clefs.

2.4 Accord de clef dans les communications de groupe

Contrairement aux communications point-à-point, dont deux utilisateurs partagent une clef secrète établie par un protocole de génération de clef (exemple protocole de Diffie-Hellman) afin de communiquer en sécurité, dans le multicast, une seule clef est utilisée entre tous les membres du groupe pour chiffrer le trafic. Cette clef est appelée clef de chiffrement du trafic (notée TEK).

2.4.1 La clef du groupe

Une clef de groupe (TEK) est une information secrète établie par accord entre les membres du groupe, cela signifie que chaque membre du groupe participe à la génération de la clef de groupe. Cette clef doit être connue uniquement par les membres courant du groupe [16].

La propriété principale d'une clef de groupe (TEK) est sa confidentialité qui se base sur les deux types suivants [15] :

1. **Confidentialité passée (*Backward Secrecy*)** : permet d'assurer qu'un nouveau membre ne peut déchiffrer les communications antérieures à son adhésion.

2. **Confidentialité future (*Forward Secrecy*)** : permet d'assurer qu'un ancien membre d'un groupe ne peut pas lire (déchiffrer) les messages du groupe après l'avoir quitté.

3. **L'indépendance de la clef** : un membre qui possède un ensemble de clefs ne peut pas calculer une autre clef qui n'appartient pas à cet ensemble.

2.4.2 Le facteur d'échelle

Le facteur d'échelle (*Scalability*) est le critère principal qui assure l'évaluation de l'efficacité d'un système de gestion de clefs de groupe, et qui traduit la capacité du système à s'adapter aux changements de la taille du groupe dû à l'ajout, départ des membres ou partition de groupe. Il peut être exprimé par les deux paramètres suivants [15] :

1. **Facteur d'échelle 1-affecte-n** : mesure l'effet d'une adhésion ou départ d'un membre sur les autres membres du groupe.

2. **Facteur d'échelle 1-n'est pas égal à -n** : mesure la capacité du système de gestion de clefs à considérer le groupe comme un tout et ne pas avoir à traiter chaque membre individuellement.

Ces deux paramètres peuvent être mesurés par le coût de communication et le coût de calcul engendrés lors des opérations de mise à jour de la clef de groupe.

2.4.3 Protocoles de gestion de clefs de groupe

Les protocoles de gestion de clefs de groupe sont classés en trois catégories [16] :

Le modèle centralisé

Le principe de ce modèle est la distribution sécurisée de clefs. Cette distribution est assurée par une entité centralisée (un contrôleur de groupe) qui s'occupe de la gestion de clefs de groupe, elle est la source du trafic multicast. L'inconvénient de cette approche est qu'elle repose sur une seule entité centrale qui devient une cible d'attaque. Parmi les protocoles centralisés, on cite *GKMPAN* [55].

Le modèle semi-distribué

Plusieurs travaux considèrent que les solutions centralisées ne sont pas adaptées aux grands réseaux, donc ne répondent pas au facteur d'échelle [56].
L'approche semi-distribuée confie la gestion des clefs aux divers sites qui peuvent ne pas faire partie du groupe. Cette méthode est utilisée dans les applications de diffusion multi-sources. Le but de ce modèle est d'éviter d'avoir un seul point de défaillance.

Le modèle distribué (l'accord)

Dans ce modèle, tous les membres du groupe participent de manière plus au moins équitable à la gestion de clefs de groupe. Chaque membre de groupe fournit une valeur qui est une contribution à la valeur de la clef et participe à son calcul.
Les protocoles de gestion de clefs distribués sont dits protocoles d'accord de clefs ou protocoles contributifs, on cite le protocole *Chiang et al* [57].

2.5 Conclusion

Dans ce chapitre, nous avons introduit la notion de groupe ainsi que le concept d'IP Multicast. Puis, nous avons présenté les principes de communication de groupe. Ensuite, nous avons cité les différents concepts de sécurité à savoir les mécanismes cryptographiques utilisés pour assurer la confidentialité des données. Finalement, nous avons montré le but et l'importance de la gestion de clef de groupe qui est un élément essentiel dans les communications multicast sécurisées. Dans le prochain chapitre, nous présentons quelques protocoles d'accord de clef de groupe dans les réseaux mobiles Ad Hoc.

Chapitre 3

Les protocoles d'accord de clef de groupe dans les réseaux mobiles Ad Hoc

3.1 Introduction

Avec l'évolution des réseaux et principalement des applications, les communications de groupe ou transmission multicast ont suscité beaucoup d'intérêt au cours de la dernière décennie. Elles correspondent à des modèles adéquats pour des applications coopératives comme l'audio conférence, les communications des forces civiles ou militaires.

La nature des applications multipoints possède une grande influence sur la méthode de gestion de clefs de groupe. En effet, certaines applications diffèrent selon la taille du groupe, le nombre de sources autorisées à envoyer des données, et le nombre de destinations ou membres autorisés à recevoir les données envoyées au groupe. Ceci nécessite la proposition d'un protocole d'accord et de gestion de clefs propre à chaque type d'application.

La gestion de la clef du groupe est assurée par une variété de protocoles qui peuvent être classé selon trois approches : centralisée, semi distribuée et distribuée [16]. Dans ce chapitre, nous décrivons les protocoles distribués ou les protocoles d'accord de clefs de groupe qui sont élaborés dans les environnements mobiles Ad Hoc.

3.2 Classification des protocoles d'accord de clef de groupe dans les MANETs

Afin de mieux comprendre la diversité des solutions proposées dans le problème d'accord de clef de groupe dans les réseaux mobiles Ad Hoc, nous proposons dans cette section une classification des protocoles d'accord de clef de groupe pour les MANETs (voir Figure 3.1).

– **Approche à plat**

- ***Algorithmes basés sur l'échange de Diffie-Hellman :*** utilisation des contributions de Diffie-Hellman pour assurer l'authentification et la confidentialité des nœuds membres dans un réseau Ad Hoc.

- ***Utilisation d'un système de chiffrement basé sur l'identité :*** c'est une combinaison de deux approches, une approche de diffusion basée sur l'identité et un système de chiffrement basé sur l'échange de Diffie-Hellman.

– **Approche orientée topologie**

- ***Arbre couvrant :*** c'est le fait d'utiliser un arbre de nœud pour assurer l'accord de clef de groupe dans un réseau.

- ***A base de Clusters :*** consiste à découper le réseau en sous-groupes appelé "*clusters*" en donnant au réseau une structure hiérarchique. Chaque cluster est représenté par un nœud particulier appelé "clusterhead". Ce nœud est élu comme clusterhead selon une métrique spécifique telles que l'identifiant, le degré, le poids, la mobilité, la densité...etc.

- ***Arbre de Clusters :*** consiste à construire un arbre de clusters enraciné sur un certain cluster ayant une certaine métrique particulière (poids, énergie, identité...etc.), c.-à-d. chaque sommet de l'arbre correspond à un cluster.

- ***Clustering en couches :*** est l'utilisation d'une approche basée sur une architecture à plusieurs niveaux.

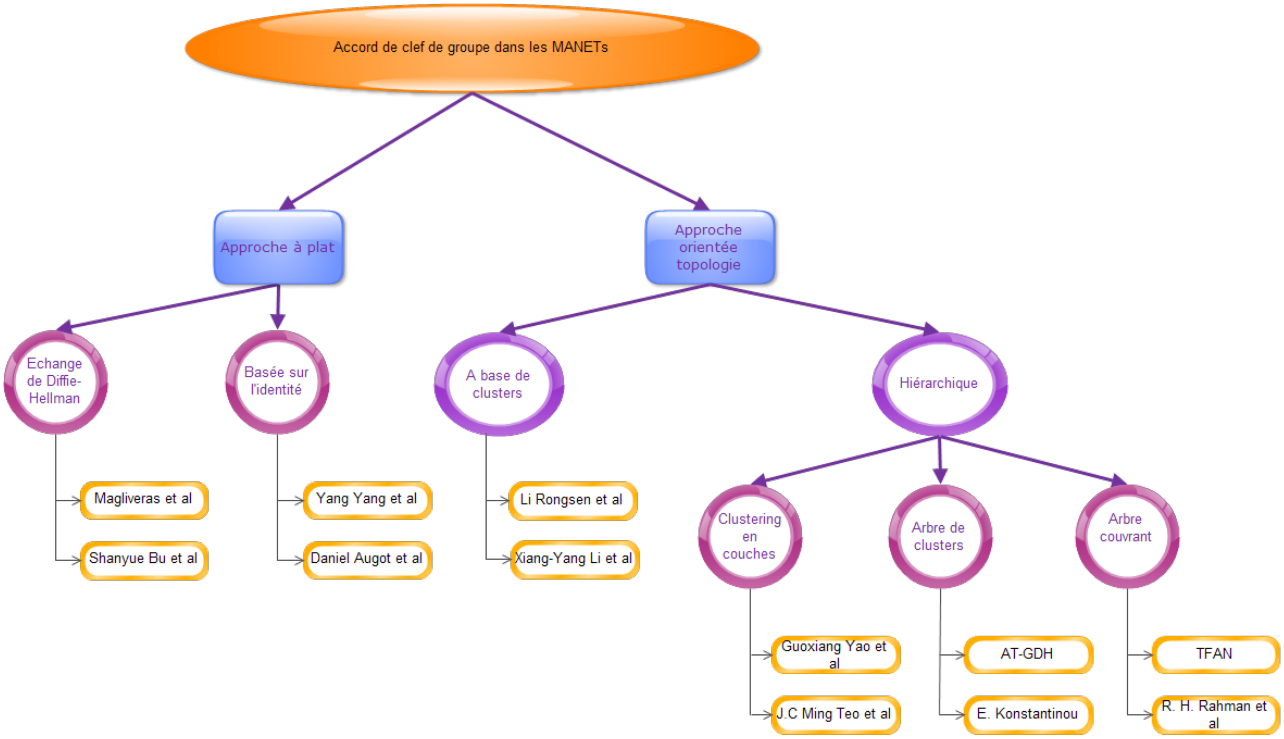

FIGURE 3.1 – *Classification des protocoles d'accord de clef de groupe dans les réseaux mobiles Ad Hoc.*

3.3 Approche à plat

3.3.1 Protocoles basés sur l'échange de Diffie-Hellman

Protocole de Shanyue Bu et al

Shanyue Bu et al [37] ont proposé un protocole d'accord de clefs de groupe qui convient aussi bien aux réseaux sans fils, et en particulier les réseaux mobiles Ad Hoc. Pour satisfaire les conditions de sécurité de tels réseaux, ce protocole utilise un système à clef publique qui est basé sur l'utilisation des signatures numériques. Le principe est que chaque membre P_i génère un nombre aléatoire r_i, le chiffre avec sa clef privée et calcule sa signature numérique. Ensuite, il diffuse le résultat vers tous les membres de groupe. Chaque membre P_j vérifie les signatures reçues, puis calcule sa clef secrète K_j ainsi que son haché H_j, et compare ensuite tous les hachés des autres membres. S'ils sont tous identiques, alors la clef de groupe GK est construite, telle que $GK = H(K_j)$.

Si un membre décide de rejoindre où de quitter le groupe, il envoie une requête JOIN pour tous les membres du groupe, génère ensuite un nombre aléatoire, le chiffre, le signe et diffuse le résultat vers tous les membres du groupe. Chaque membre ayant reçu

toutes les signatures numériques des autres membres, compare tous leurs hachés, s'ils correspondent tous alors la clef finale du réseau est mise à jours.

Le protocole proposé par *Shanyue Bu et al* [37] assure la confidentialité passée (*Backward Secrecy*) et future (*Forward Secrecy*).C'est-à-dire que si un membre quitte le groupe, il lui sera impossible de déchiffrer les messages qui circulent dans le groupe après l'avoir quitté et c'est aussi pareil quand un membre rejoint le groupe (il ne peut pas déchiffrer les communications antérieures).

Protocole de Magliveras et al

Spyros Magliveras et al [10] ont proposé un protocole pour sécuriser une communication de groupe et l'information partagée entre les membres du groupe reste inaccessible pour les autres.

Une clef de groupe est établie parmi tous les membres participants et cette clef est employée pour chiffrer tous les messages destinés au groupe. Et seuls ces derniers peuvent déchiffrer les messages.

Après avoir étudié le protocole CRTDH [32], qui est basée sur le théorème du reste chinois et l'arrangement d'échange de clefs de Diffie-Hellman [17], *Spyros Magliveras et al* [10] ont constaté qu'il possède quelques problèmes " basse efficacité, probablement une petite clef, et en possession du même petit multiple commun " *LCM* ". Ils ont amélioré le protocole en apportant de légères modifications, de telle sorte qu'il soit plus efficace que l'original, comme suit :

1. Sélectionner une contribution privée de Diffie-Hellman x_i, ensuite calculer la contribution publique $y_i = g^{x_i} \bmod p$ (tel que g le générateur et p est le modulo principal utilisé dans le calcul de Diffie-Hellman).

2. Diffuser y_i vers tous les membres du groupe. Quand un membre reçoit toutes les contributions, il calcule la clef (m_{ij}) qu'il va partager avec chacun d'eux (i et j).

3. Chaque membre choisit un p_{ij} tel que le pgcd (p_{ij}, m_{ij}) =1 pour ($i \neq j$), et génère un nombre aléatoire k_i tel que $k_i < \min(m_{ij}, \forall j)$, et calcule ensuite $crt_{ij} = k_i \pmod{m_{ij}}$ (tel que crt : *Chinese Remainder Theorem*). Finalemment, chaque membre calcule la clef finale $GK = \sum k_i$, qui est la clef de groupe.

Evènements

1. ***Adhésion d'un membre :*** quand un nouveau membre souhaite rejoindre le groupe, un membre appartenant au groupe calcule le haché de la clef du groupe et transmet le résultat au nouveau membre. Ce dernier exécute les étapes décrites précédemment, le nouveau groupe calcule la nouvelle clef GK_{new}.

2. ***Départ d'un membre :*** quand un membre quitte le groupe, un membre U_i exécute la troisième étape avec un nouveau K_i, diffuse ensuite le nouveau crt_{ij}, et calcule la nouvelle clef de groupe.

3.3.2 Utilisation d'un système de chiffrement basé sur l'identité

Protocole de Yang Yang et al

Yang Yang et al [20] ont proposé un protocole d'accord de clefs fondé sur une approche de diffusion basée sur l'identité, qui vise à fournir une solution d'accord de clefs efficace et rapide dans les MANETs. Ce système répond non seulement aux exigences des réseaux mobiles Ad Hoc mais réduit aussi le coût de communication. Il combine le système de chiffrement basé sur l'identité [21] avec des applications bilinéaires pour remplacer la configuration d'affiliation de clefs de groupe comme les protocoles antérieurs.

Chaque membre de groupe est désigné par une identité unique qui joue le rôle de l'adresse MAC tel que dans les réseaux filaires. Il s'agit d'une méthode qui évite l'authentification sur des signatures numériques avec un certificat délivré par une autorité de certification (CA) commune. Si un membre décide de rejoindre ou de quitter le groupe, une nouvelle clef de groupe pourrait facilement être construite. Chaque membre de groupe est considéré comme diffuseur. La charge de calcule est la même pour chaque récepteur. Les récepteurs peuvent calculer la clef de session en utilisant leurs propres clefs privées sur l'échange de messages avec d'autres membres du groupe.

Pendant l'éxécution du protocole, le générateur de clefs privées (PKG) génère la clef publique PK (*Public Key*) et la clef secrète MK (*Master Key*) pour le système. Le PKG vérifie l'identité des utilisateurs ID_i et génère ensuite, la clef privée correspondante d_{ID} (après le succès de la vérification). Puis, une clef de session (clef de groupe) doit être établie pour le groupe. Après avoir connu les identités des récepteurs, le diffuseur génère un entête d'encapsulation (*Hdr*) pour la clef de groupe K, et diffuse (S, *Hdr*), tel que S=id_1,...id_n. Après avoir reçu le message(S, *Hdr*), chaque utilisateur ayant comme identité ID tel que ID $\in$ S pourrait participer à la création de la clef de groupe K avec sa propre clef privée $\mathsf{d}_{\mathrm{ID}_i}$.

Après que la clef de session K soit mise en place pour le groupe Ad Hoc, le message M pourrait être chiffré à l'aide d'un système de chiffrement symétrique tel que DES ou AES.

Protocole de Daniel Augot et al

Daniel Augot et al [12] ont proposé un protocole d'accord de clefs de groupe, particulièrement bien adapté aux réseaux Ad Hoc. Il n'exige aucune commande spéciale des membres de groupe, n'importe quel membre peut être probablement choisi comme le chef de groupe pour une session et le rôle peut être facilement tourné parmi les autres membres.

Le principe d'un tel protocole est que chaque chef de groupe (M_l), choisit un nombre aléatoire (N_i) et l'envoie avec son identité vers tous les membres de son groupe. Chaque membre intéressé répond au message du chef de groupe, en lui envoyant son identité U_i, un nombre aléatoire et sa clef secrète. Ensuite, le chef de groupe collecte toutes les clefs secrètes reçus, les ajoutent à sa clef et diffusent le résultat vers tous les membres de groupe. La clef du groupe est alors crée.

Lorsqu'un nouveau membre souhaite rejoindre un groupe, il envoie une requête d'adhésion *request*-JOIN accompagnée de son identité et sa clef secrète à tous les membres du groupe. L'ancien chef du groupe génère une nouvelle clef secrète et l'envoie au nouveau chef qui sera choisit aléatoirement. Ce dernier collecte toutes les clefs secrètes des participants, les ajoutent à sa clef et diffusent le résultat vers tous les membres du groupe. Quand le nouveau membre est autorisé a rejoindre le groupe, il reçoit une requête *request*-JOIN-OK.

Quand un membre souhaite quitter le groupe, il annonce son départ via la requête *request*-LEAVE. Le chef de groupe change sa clef secrète et l'ajoute aux autres clefs des membres de son groupe qu'il a déjà collectées (en supprimant la contribution du membre qui veut quitter). Puis, il envoie le résultat à tous les membres.

3.4 Approche orientée topologie

3.4.1 A base de Clusters

Protocole de Li Rongsen et al

Li Rongsen et al [8] ont proposé un protocole d'accord de clefs de groupe pour les MANETs. Son principe est que chaque nœud peut communiquer avec n'importe quel autre nœud du réseau, soit dans un même cluster, soit dans des clusters différents via le clusterhead de chacun. Le réseau est déjà divisé en quelques clusters qui sont dirigés par des clusterheads. La Figure 3.2. donne un aperçu du réseau.

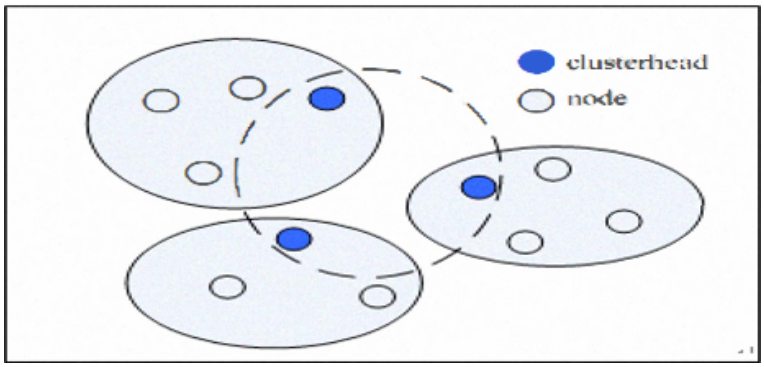

FIGURE 3.2 – *Structure du réseau à base des clusters.*

Quand un nœud A veut communiquer avec le nœud B (dans un même cluster), A génère un nombre aléatoire N_1 et une clef privée r_A (qui sera expirée à la fin de la session), A calcule alors une clef temporaire P_A, la signe et envoie un message msg_1 vers le nœud B qui comprendra l'identité de A (ID_A), l'identité de B (ID_B), la clef P_A, le signé de A ($Sign_A$) et un certificat ($Cert_A$) délivré par l'autorité de certification (*Trusted Authority* où TA). Après que B ai reçu le message msg_1, il vérifie la validité du certificat $Cert_A$, et déchiffre ensuite msg_1. Puis, il calcule une clef temporaire P_B et vérifie la validité de la signature $Sign_A$. Si le nœud B accepte la requête, il génère alors une clef privée r_B et calcule P_B, la signe et envoie à son tour un message msg_2 vers le nœud A (msg_2= ID_B, ID_A, P_B, $Sign_B$, $Cert_B$ et N_1+1.

A la réception du message msg_2, le nœud A vérifie le certificat ($Cert_B$) ainsi que la signature $Sign_B$, et calcule ensuite la clef K_{A-B}=r_A * P_B qui sera égale à K_{B-A}. Donc, les nœuds A et B partagent la même clef de session.

Quand un nœud A veut communiquer avec le nœud B (dans des clusters différents), A génère un nombre aléatoire r_A, calcule la clef P_A, puis la signe ($Sign_A$). Ensuite, A envoie le message msg_1 vers son clusterhead (CH_1) qui comportera l'identité de A (ID_A), $Cert_A$, $Sign_A$ et la clef publique de son cluster RK_{CH_1} qui sera accompagnée de l'identité de B (ID_B), P_A et un nombre aléatoire N_2.

Après la réception de message msg_1, le clusterhead CH_1 collecte des informations sur le clusterhead destinataire (CH_2) selon le nombre de nœud mis en commun entre tous les clusters. CH_1 envoie ces informations via un message msg_2 vers le nœud A, suivi de la clef publique de nœud B (PK_B). A la réception, le nœud A décrypte le message msg_2 et extrait la clef PK_B, et envoie ensuite le message msg_3 qui comprendra ID_A, ID_B, CH_1, CH_2, N_2, P_A ainsi que $Cert_A$ et $Sign_A$ (msg_3 sera chiffré avec la clef PK_B). Le nœud B vérifie la validité de $Cert_A$ et déchiffre le message msg_3. Puis, le nœud B calcule K_{B-A}=r_B*P_A et

choisit une clef privée aléatoire r_B, calcule P_B, le signe et envoie le message msg_4 vers le nœud A (msg_4= ID_B, ID_A, CH_A, CH_2, N_2+1, P_B, $Cert_B$, $Sign_B$). A la réception du msg_4, le nœud A vérifie le certificat $Cert_B$, décrypte le message et calcule K_{A-B}=r_A*P_B qui serait égale à K_{B-A}. Donc, A et B partagent la même clef de session.

Protocole de Xiang-Yang Li et al

Xiang-Yang Li et al [11] ont proposé un protocole d'accord de clefs de groupe pour les réseaux mobiles Ad Hoc. Le principe de ce protocole est de diviser les nœuds en sous-groupes en se basant sur leur localisation géométrique. Chaque nœud ayant la petite identité parmi tous ses voisins, est désigné comme leader du groupe (également appelé Dominateur). Ce dernier diffuse le message "*IamDominator*" vers tous ses voisins à un saut. Les nœuds ayant reçus le message "*IamDominator*" sont appelés nœuds dominés. Ces nœuds dominés diffusent le message "*IamDominatee*" vers tous leurs voisins à un saut. Tous les nœuds intermédiaires qui connectent un leader d'un sous-groupe avec un autre leader d'un autre sous-groupe sont appelés " *Connecteur*", et l'ensemble des connecteurs reliés avec les leaders de groupes forment un ensemble de leaders reliés (*Connected dominating set* où CDS).

La Figure 3.3. donne un aperçu d'un CDS construit.

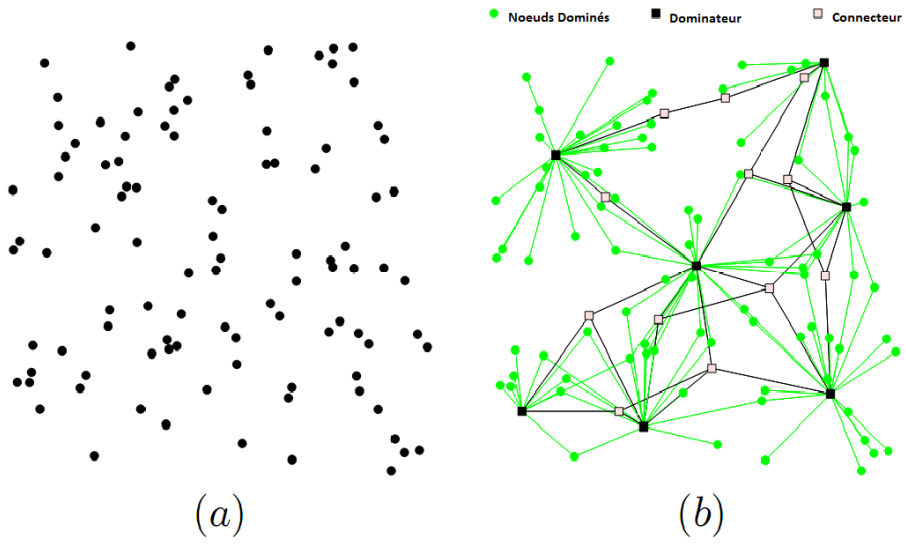

FIGURE 3.3 – *(a) un ensemble de noeuds sans fil. (b) CDS construit.*

Tous les nœuds membres d'un groupe s'accordent sur la clef de groupe (*participent à la création de la clef de groupe*) comme suit :

Chaque nœud dominé M_i génère un nombre aléatoire r_i, et envoie ses contributions (α^{r_i}) vers son nœud dominateur (*leader de groupe*), tel que α représente le générateur

du groupe. A la réception de toutes les contributions, le nœud dominateur génère un nombre aléatoire, et envoie y_i vers tous les nœuds dominés d'un même groupe tel que $y_i = (\alpha^{\frac{(\prod r_j)}{r_i}})$. Chaque nœud membre calcule alors la clef du groupe $S_n = y_i^{r_i} = \alpha^{r_j}$.

Une fois que les clefs des groupes sont construites, tous les leaders des groupes (*nœuds dominateurs*) s'accordent sur la clef du réseau à travers l'épine dorsale, en utilisant les connecteurs comme liens entre les différents leaders des groupes.

La topologie du réseau change fréquemment due à la mobilité des nœuds. Si un nouveau nœud u veut rejoindre le groupe, et l'un de ses voisins à un saut est un leader de groupe, alors u envoie un message JOIN pour rejoindre le groupe du leader et il sera désigné comme nœud dominé. Une mise à jours sera effectuée pour obtenir la nouvelle clef du groupe. Autrement, il sera désigné comme leader, et envoie le message " *IamDominator* " à tous ses voisins à un saut, les informant qu'il est le nouveau leader du groupe. Une mise à jours de clef est alors effectuée, et une procédure de recherche de nouveaux connecteurs pour le nouveau membre sera élaborée.

Si un membre veut quitter le groupe, il envoie un message LEAVE à tous ses voisins. Une mise à jour de la clef de groupe est alors effectuée, et si le membre qui veut quitter et un leader de groupe, alors un nouveau leader de groupe sera désigné (celui avec la petite identité). Une mise à jours des connecteurs (nœuds intermédiaires) et de la clef de groupe sera effectuée.

Xiang-Yang Li et al [11] ont supposé que tous les nœuds sont statiques mais ce n'est pas toujours vrai dans certaines applications ; Les nœuds sans fil se déplaceront autour. Il est facile d'ajouter ou de supprimer un nœud dans un tel protocole, mais ces opérations causent un problème majeur pour la topologie du réseau (changement de la topologie). Pour que le protocole soit efficace, il faut seulement que les nœuds se déplacent pendant un petit intervalle de temps.

3.4.2 Structure Hiérarchique

Protocole de regroupement AT-GDH

Maarit Hietalahti [9] a proposé un protocole d'accord de clefs de groupe basé sur des clusters. Ce protocole est une généralisation du protocole de Diffie-Hellman [17] appelé "AT-GDH" (*Arbitrary Topology Generalisation of Diffie-Hellman*). *M. Hietalahti* [9] utilise un arbre couvrant qui contient seulement les liens utilisés dans l'accord initial, contrairement aux échanges de Diffie-Hellman, qui sont faits seulement avec des voisins à un seul saut, et cela dans le but d'éviter les perturbations des transmissions de voisins.

Le protocole "AT-GDH" peut être employé dans n'importe quelle topologie de

réseau reliée avec des liens bi-directionnels vu qu'un arbre recouvrant peut toujours être construit dans un tel réseau.

- ***Le principe de clusterisation***

  Tout d'abord, le réseau est divisé en groupe avec un mécanisme de clusterisation qui crée des groupes très stables. Chaque groupe à une clef secrète commune, les groupes s'accordent sur une clef de groupe par le protocole "AT-GDH".

  Chaque *cluster-head* peut représenter son groupe et employer la clef de groupe en tant que son exposant secret. Après l'exécution du protocole "AT-GDH", les *cluster-heads* distribuent le dernier message reçu dans leur groupe de sorte que d'autres nœuds puissent également calculer la clef du groupe de réseau.

  Maintenant que les *cluster-heads* ne sont pas nécessairement à une distance d'un saut entre eux, les messages doivent être transmit par relais. Les passages transmettant les messages par relais sont des membres d'un groupe et connaissent déjà le secret de groupe. Cependant, cela peut affecter le temps de communication en empruntant des liens supplémentaires au chemin.

  La Figure 3.4. illustre le fonctionnement du protocole "AT-GDH" [9].

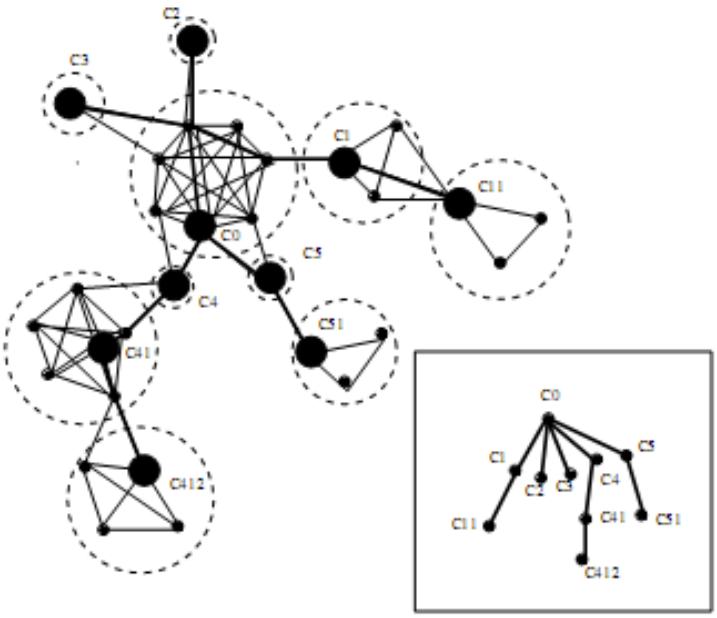

FIGURE 3.4 – *le protocole de regroupement AT-GDH.*

- ***Gestion des clefs***

  Tous les nœuds feuilles choisissent un exposant secret aléatoire r_i, le chiffrent et

envoient le résultat à leurs parents respectifs. Après qu'un nœud ait reçu les clefs chiffrées de tous ses fils, ils choisissent leurs exposants et forment les clefs de type Diffie-Hellman avec leurs fils à plusieurs reprise, utilisant la clef résultante comme nouvel exposant.

Le secret formé avec le dernier fils sert de nouvelle clef privée, que le nœud chiffre et envoie à son parent. Quand ce parent recevra les messages semblables de tous ses fils, il pourra répéter le même calcul. Ceci continue jusqu'à ce que la racine ait reçu toutes les clefs chiffrées de ses fils. Alors, la racine répète le même calcul que tous les autres nœuds parents, la clef secrète est formée ainsi entre la racine et son dernier fils.

Dans la dernière phase du protocole, les clefs chiffrées requises pour extraire la clef de groupe sont propagées vers le haut de l'arbre des parents à leurs fils à partir de la racine.

Protocole de Konstantinou

Elisavet Konstantinou [7] a proposé un protocole basé sur des messages à un seul saut et qui n'exige pas d'émissions, en plus d'économiser l'énergie. Ce protocole est basé sur le regroupement d'un groupe entier dans des petits *clusters* de deux où trois nœuds. Après la formation du groupe, les nœuds devraient avoir une structure qui se compose de trois branches principales, qui sont des arbres binaires avec l'addition d'une arête entre chaque deux fils d'un même parent. Ainsi, chaque nœud appartient à deux *clusters* (groupes) excepté les feuilles, qui appartiennent à un seul *cluster*.

Le principe de clusterisation de ce protocole est réparti en deux phases. Dans la première étape, un nœud est choisit aléatoirement afin d'être le nœud principal de la structure entière (exécution d'un algorithme d'élection de chef). Puis, le chef devrait mettre au courant ses voisins de sa présence en envoyant un message " *I am Alive* " et ajouter les deux premiers voisins qui répondent à son message dans le premier *cluster*. Dans la deuxième étape, chacun des trois nœuds exécutera le même procédé en ajoutant à son nouveau *cluster* les deux premiers nœuds répondant à leur message " *I am Alive* ". Si un seul nœud qui répond au message " *I am Alive* ", alors un *cluster* de deux nœuds (au lieu de trois) peut être construit.

Ce processus continuera jusqu'à ce que tous les nœuds deviennent une partie du *cluster*. Le nœud principal de chaque *cluster* s'appellera le nœud parent du *cluster*.

- ***La variante non-authentifiée du protocole***

  Avant l'exécution du protocole d'accord de clefs de groupe, tous les nœuds devraient

s'accorder sur l'utilisation des mêmes paramètres de courbe elliptique, le même point de base et d'une même fonction de hachage. Le protocole se déroule comme suit :

Phase01 : chaque membre M_i du groupe génère un nombre aléatoire m_i, et calcule m_i P, puis il envoie le résultat aux autres membres de son *cluster*. Ainsi, les feuilles envoient leur résultats aussi à tout au plus deux nœuds et les nœuds internes à 4 où 3 nœuds.

Phase02 : une procédure de création de clef du *cluster* est exécutée simultanément dans chaque *cluster*. en particulier, dans un cluster de 3 membres m_1, m_2 et m_3 a qui correspondent les clefs publiques respectivement m_1P, m_2P, et m_3P, chaque membre calcule sa fonction de hachage en fonction des clefs publiques des deux autres membres. Le résultat de chacun est la clefs secrète commune.

Phase03 : chaque feuille dans la structure partage une clef secrète avec les nœuds du même *cluster*, alors que le reste des membres de groupe partagent deux clefs secrètes, une avec les nœuds du *cluster* du niveau supérieure et l'autre avec les nœuds du *cluster* du niveau inférieur.

Le protocole échoue, si les nœuds commencent à se déplacer loin pendant les trois étapes du protocole d'accord.

- ***La variante authentifiée du protocole***

  Chaque membre du groupe à une identité ID_i et crée sa propre clef publique Q_i=H(ID_i). Le KGC (*key generation center*) choisit un nombre aléatoire S et calcule P_{pub}=S.P (où P est le point de base), chaque membre envoie sa clef publique Q_i au KGC, ce dernier crée le point S_i et l'envoie de nouveau aux membres de groupe. Le S_i sera alors la clef privée de chaque membre du groupe.

  Premièrement, chaque membre du groupe génère un nombre aléatoire et calcule les points P_i, puis il les envoie aux autres membres de son *cluster*. Ensuite, une procédure de création de clef authentifiée de *cluster* est exécutée simultanément sur chaque *cluster*. L'étape trois est identique à celle de la variante non-authentifiée du protocole.

Elisavet Konstantinou [7] voulait dans le meilleur des cas, que la structure basée sur les *clusters* soit équilibrée et que tous les *clusters* formés seront de taille 3, mais

cela est impossible dans la pratique, car cela signifie que chaque branche devrait avoir approximativement une taille de $\log_2 \frac{N}{3}$, où n est le nombre des nœuds du groupe. Cependant, même dans le cas moyen la taille de la plus grande branche aura $o(\log_2 \frac{N}{3})$.

Un avantage important d'un tel protocole est qu'il distribue bien la consommation d'énergie parmi les participants, car chaque nœud a les rôles semblables en termes de calcul et d'échanges de communication.

Protocole de J. C. Ming Teo et al

J. C. Ming Teo et al [6], ont proposé un protocole d'accord de clef de groupe pour les MANETs, particulièrement les réseaux Ad Hoc de large échelle. Ils utilisent un modèle de groupe hiérarchique circulaire (C-H), où le réseau est divisé en sous-groupes de h couches différentes, et chaque sous-groupe est arrangé dans un cercle (voir Figure 3.5). *J. C. Ming Teo et al* [6] souhaitent établir une clef de groupe pour des communications fiables et sécurisées, ils ont donc établi ce modèle C-H qui vérifie $n = c^h$, tel que c est le nombre des membres de chaque sous-groupe, n est le nombre d'utilisateurs et h est le nombre total de couches dans le modèle (C-H).

Ce modèle est basé sur le principe de *Burmester-Desmedt* (BD) [38], qui utilise une structure basée sur l'anneau, où tous les utilisateurs sont arrangés dans un anneau. Le principe de *Burmester-Desmedt* (BD) [38] est de définir les clefs des sous-groupes, pour en déduire au final une clef de tout le réseau. Chaque utilisateur U_i pour $i \in \{1, ..., n\}$, choisit un nombre aléatoire $r_i \in (Z_q^*)$, diffuse $z_i = g^{r_i} \bmod p$ vers tous ses voisins. Ensuite, tous les utilisateurs calculent et diffusent $X_i = g^{r_i r_{i+1} - r_{i-1} r_i} \bmod p$ vers tous leurs voisins. Après avoir reçu tous les messages de ses voisins, chaque utilisateur U_i calcule la clef de groupe commune K= $g^{r_1 r_2 + r_2 r_3 + ... + r_n r_1} \bmod p$.

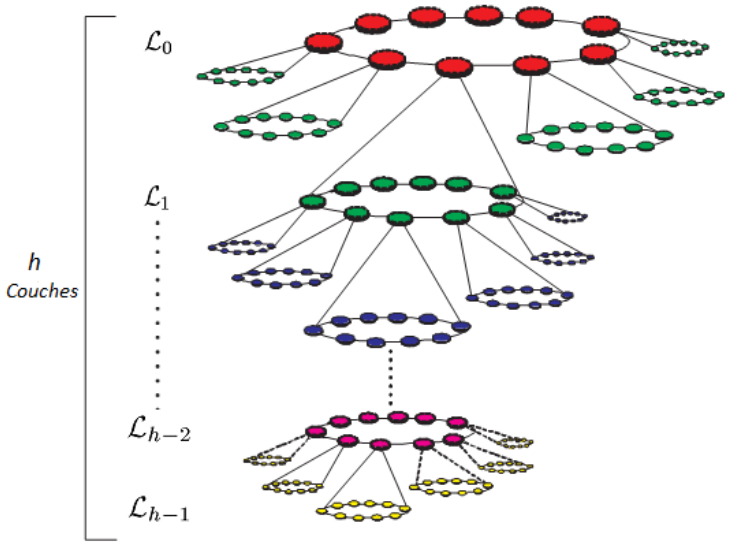

FIGURE 3.5 – *Une illustration du modèle de groupe (C-H) avec c = 10.*

Ce protocole comporte quatre phases. Pendant les trois premières phases les clefs $K_{SG_j^{(L_i)}}$ pour chaque sous-groupe $SG_j^{(L_i)}$ sont calculées et la clef $K_{SG_0^{(L_0)}}$ pour la couche la plus élevée L_0 est la clef finale K de tout le groupe. Par contre, durant la dernière phase, la clef finale K du groupe est chiffrée et diffusée vers tous les sous-groupes en utilisant la cryptographie à clef symétrique.

Phase 01 *:* le protocole d'accord de clef de groupe pour le modèle (C-H) commence à la plus basse couche L_{h-1}. Chaque sous-groupe $SG_j^{(L_{h-1})}$ pour $j \in \{0, ..., c^{h-1} - 1\}$ exécute le protocole *Burmester-Desmedt* (BD) [38] pour obtenir sa clef $K_{SG_j^{(L_{h-1})}}$, ou c est le nombre de membres dans chaque sous-groupe et h représente le nombre total de couches dans le modèle.

Phase 02 *:* pour les couches L_ν tel que $\nu \in \{h-2, ..., 1\}$, chaque membre $U_{j,k}^{(L_\nu)}$ du sous-groupe $SG_j^{(L_\nu)}$ pour $k = jc + l$ tel que $l \in \{0, ..., c-1\}$, exécute le protocole *Burmester-Desmedt* (BD) [38] pour obtenir la clef de sous-groupe $K_{SG_j^{(L_\nu)}}$. Chaque $U_{j,k}^{(L_\nu)}$ va utiliser la clef $K_{SG_k^{(L_{\nu+1})}}$ du sous-groupe $SG_k^{(L_{\nu+1})}$ qu'il contrôle dans la prochaine couche $L_{\nu+1}$ pour calculer et diffuser $z_{k^{(L_\nu)}} = g^{H(K_{SG_j^{(L_\nu)}})} \bmod p$ dans l'arrangement de DB [38]. Cette phase se termine quand tous les sous-groupes de la couche L_1 ont calculé leurs clefs respectives de sous-groupe.

Phase 03 *:* tous les membres $U_{0,k}^{(L_0)}$ du sous groupe $SG_0^{(L_0)}$ de la couche la plus élevée L_0 vont utiliser la clef du sous-groupe $K_{SG_k^{(L_1)}}$ pour calculer et diffuser $z_{k^{(L_0)}} = g^{H(K_{SG_k^{(L_1)}})} \bmod p$. Ensuite, chaque membre $U_{0,k}^{(L_0)}$ utilise la clef de son sous-

groupe et une clef de chiffrement symétrique $E_k(m$ (tel que k est la clef secrète) pour produire et diffuser $E_{K_{SG_k^{(L_1)}}}(K)$ pour leur sous-groupes respectifs.

Phase 04 *(Distribution de la clef) :* chaque membre de sous-groupe $U_{j,k}^{(L_\nu)}$ dans la couche L_ν tel que $\nu \in \{1, ..., h-2\}$ décrypte en premier lieu le message chiffré reçu de la part du contrôleur de son sous-groupe qui appartient à un sous-groupe de la couche précédente $L_{\nu-1}$, pour obtenir la clef finale du groupe K. Ensuite, chaque sous-groupe $U_{j,k}^{(L_\nu)}$ utilise la clef du sous-groupe $K_{SG_k^{(L_{\nu+1})}}$ et une clef de chiffrement symétrique $E_k(m)$ pour produire et diffuser $E_{K_{SG_k^{(L_1)}}}(K)$ pour leur sous-groupes dans la couche $L_{\nu+1}$.
Ce processus se poursuit jusqu'à ce que tous les membres de sous-groupes $U_{j,k}^{(L_{h-1})}$ de la plus basse couche L_{h-1} obtiennent la clef finale de groupe K en déchiffrant le message reçu $E_{K_{SG_j^{(L_{h-1})}}}(K)$.

L'arrangement de *Burmester-Desmedt* (BD) [38] consomme une très grande quantité d'énergie. Cette forte consommation d'énergie est dû à la grande quantité de messages à recevoir par chaque nœud lors du calcul de la clef de groupe K. Par contre, pour le modèle (C-H) proposé par *J. C. Ming Teo et al* [6], il est plus économique en énergie, et c'est principalement en raison de la taille des messages qui doivent être reçus par chaque utilisateur U_1 à U_{n-1} pour calculer la clef de groupe.

Protocole de Guoxiang Yao et al

Guoxiang Yao et al [53] ont proposé un protocole d'accord de clef de groupe basé sur une architecture à deux niveaux qui utilise deux différentes approches. Une approche centralisée dans la couche inférieure, et une autre approche distribuée dans la couche supérieure. L'utilisation de ces deux approches à l'avantage d'augmenter la flexibilité du réseau, et aussi l'efficacité des mises à jour des clefs de groupe. Ce protocole utilise aussi un mécanisme d'authentification dans l'accord de clef qui peut protéger le réseau des attaques actives, et qui permet aussi de réduire le temps de communication. La figure 3.6. illustre un réseau mobile Ad Hoc basé sur une architecture à deux niveaux.

- ***L'accord de clef dans la couche inférieure du réseau***

  La couche inférieure se compose de nœuds ordinaires. Les nœuds ordinaires sont divisés en plusieurs groupes qui appliquent l'accord de clef centralisé.

  Chaque nœud ordinaire (M_i) génère un nombre aléatoire (r_i). Puis, il calcule ($r_1 r_2 ..... r_i$)*G en utilisant toutes les contributions des précédents nœuds, et envoie le tout aux nœuds suivants. Ensuite, chaque nœud M_{n-1} génère un nombre aléatoire (r_{n-1}) et calcule à son tour ($r_1 r_2 ..... r_{n-1}$)*G en utilisant les contribu-

tions des $n-2$ précédents nœuds, et leurs diffusent le résultat comme suit : $M_{n-1} \longrightarrow M_i : (r_1 r_2 .....r_{n-1}) * G$, tel que $i \in [1, n-1]$.

Après avoir reçus les messages envoyés par M_{n-1}, les $n-2$ nœuds calculent séparément $\frac{(r_1 r_2 .....r_{n-1})}{r_i} * G$ utilisant le nombre aléatoire de chacun (r_i). Puis, ils envoient le résultat aux nœuds M_i et envoie aussi $\frac{(r_1 r_2 .....r_{n-1})}{r_i} * G$ au nœud de service (*leader de groupe* où SN_n).

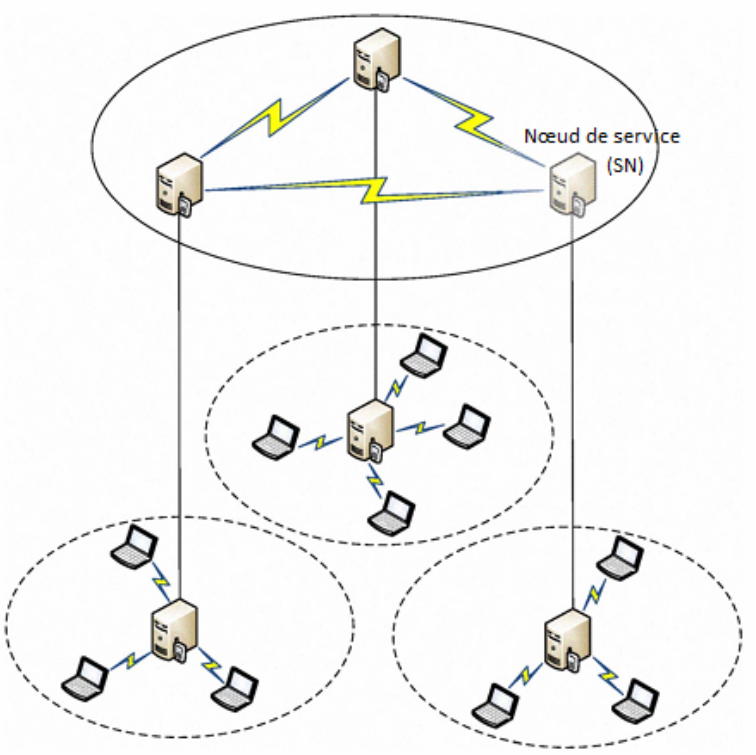

FIGURE 3.6 – *Un réseau mobile Ad Hoc basé sur une architecture à deux niveaux.*

Cependant, chaque nœud de service (*leader de groupe* où SN_n) calcule le secret Q_i en utilisant sa clef privée, le nombre aléatoire (r_n) qu'il a généré, ainsi que les clefs publiques et toutes les contributions de tous les nœuds membres de son groupe. Ensuite, il diffuse Q_i vers tous les nœuds d'un même groupe.

Tous les nœuds ordinaires (M_i), ainsi que les leaders de leur groupe (SN_n) participent à la création de la clef de groupe en utilisant la clef publique de (SN_n), ainsi que les clefs privées (Pk_i) de tous les nœuds ordinaires.

Quand un nouveau nœud veut rejoindre le groupe, le leader du groupe (SN_n) génère un nouveau nombre aléatoire (r_n) pour assurer la confidentialité passée (*Backward Secrecy*). Une mise à jour de la clef sera élaborée. Par contre, si un

membre veut quitter le groupe, le leader SN_n génère un nouveau membre aléatoire r_i pour assurer la confidentialité future (*Backward Secrecy*). Ensuite, une mise à jour sera effectuée pour la nouvelle clef de groupe.

- ***La clef dans la couche supérieure du réseau***

  La couche supérieure, qui est le réseau de base, se compose des nœuds de service avec une capacité de manipulation plus élevée. La méthode d'accord distribuée est appliquée en nœuds de service. Chaque nœud de service participe à la production de la clef de groupe du réseau de base.

  Chaque leader du groupe (SN_n) envoie la clef de son groupe, ainsi que toutes les contributions de tous les membres de son groupe aux nœuds de services (SN_j). A la réception, chaque nœud de service (SN_j) de la couche supérieure calcule la nouvelle clef de groupe (*Key*) en utilisant sa propre clef publique (SK_j) et les contributions de tous les membres, ainsi que leurs clefs privées. Ensuite, chaque nœud de service (SN_i) de la couche supérieure génère un nombre aléatoire (K_i), et calcule Q_{ij} en utilisant la clef publique et la clef privée de SN_j. Enfin, ils diffusent le résultat Q_{ij} vers les autres nœuds de service de la même couche (couche supérieure).

  Après avoir reçu Q_{ij}, le nœud de service (SN_j) calcule la clef de groupe en utilisant sa clef privée Pk_j, sa clef publique Sk_j, et un nombre aléatoire K_j qu'il va génére.

  Si un nœud N_{n+1} veut rejoindre où quitter le groupe de la couche supérieure, il doit générer un nombre aléatoire K_{n+1}, et une mise à jour de la clef sera effectuée.

  Le protocole de *Guoxiang Yao et al* [53] peut résister aux attaques passives et identifier les nœuds malveillants. Car l'attaquant peut obtenir les informations sur les communications dans le réseau pendant le processus d'accord de clef de groupe par l'écoute du canal. Et même si l'attaquant peut capturer Q_i en utilisant les clefs publiques et privées des nœuds, il ne peut pas obtenir la clef de groupe.

Protocole TFAN

Shan-shan Tu et al [35] ont proposé une comparaison entre les protocoles d'accord de clefs de groupe basé sur la structure d'arbre. Ces structures sont basées sur la cryptographie à clef publique et l'échange de clefs de Diffie-Hellman [17]. Ils ont étudié les trois structures les plus adaptées pour l'accord de clefs dans les réseaux Ad Hoc telles que : TGDH [33], STR [34] et TFAN [36].

La structure TGDH a été présenté par *Yondae Kim et al* [33]. Chaque membre envoie sa clef privée, ainsi que sa clef publique vers ses parents, le processus continu jusqu'à ce

que toutes les clefs soient arrivées à la racine, cette dernière calcule la clef de groupe et la diffuse vers tous ses fils, chiffrée avec la clef privée de chacun. Ce processus s'arrête quand toutes les feuilles auront reçus la clef de groupe. En revanche, si un membre veut rejoindre où quitter le groupe, des mises à jours des clefs de tous les parents sont effectuées jusqu'à ce qu'on obtient la clef finale de groupe.

La structure STR a été proposé par *Kim et al* [34]. Chaque nœud feuille génère un nombre aléatoire R_i ainsi qu'une clef publique BR_i, il les envoie vers ses parent, jusqu'à ce que le résultat atteint la racine, cette dernière calcule la clef finale K, la diffuse ensuite vers tous les membres. Si un membre par exemple, veut rejoindre le groupe, c'est seulement les clefs de ses parents respectifs qui seront mises à jours ainsi que la clef de la racine. Par contre, si un membre quitte le groupe, une mise à jours de toutes les clefs de chaque membre ainsi que leurs parents respectifs jusqu'à l'obtention de la clef de tout le réseau.

Lijun Lia and Mark Manulis [36] ont présenté la structure TFAN. Il y a au plus quatre membres dans un sous-arbre, chaque membre diffuse sa clef publique vers la racine de chaque sous-arbre et chaque racine du sous-arbre à son tour les diffusent vers la racine principale. Si un membre veut adhérer au groupe, il envoie une requête JOIN avec sa clef publique vers tous les autres membres. Une mise à jour de la clef sera alors élaborée selon le principe de TGDH [33] où STR [34]. Par contre, si un membre veut quitter le groupe, la clef de la racine du sous-groupe sera mise à jours, ainsi qu'une diffusion des clefs publiques des autres membres vers toutes les racines jusqu'à l'obtention de la nouvelle clef de groupe.

Il est important de choisir la structure d'arbre adéquate pour l'accord de clef de groupe dans les réseaux Ad Hoc. Ceci est l'élément principal pour une communication efficace, et qui affecte directement la sûreté du réseau tout entier.

Protocole de R. H. Rahman et al

L'idée de base de *Rony Hasinur Rahman et al* [48] est de construire et distribuer une clef secrète d'une manière sécurisée parmi tous les nœuds du réseau.

Le protocole proposé par *Rony Hasinur Rahman et al* [48] commence par construire un arbre couvrant (voir figure 3.7), en impliquant tous les nœuds valides. Le nœud initiateur (celui qui exécute le protocole en premier) devient la racine de l'arbre et envoie un message pour tous ses voisins, de ce fait les nœuds ayant reçues le message deviennent les fils de la racine, ces derniers envoient à leurs tour un message similaire à tous leurs voisins excepté leurs parents. Ce processus se répète jusqu'à ce que les nœuds feuilles soient atteints. Si un nœud reçoit plusieurs messages, il ne prend en considération que le message reçu en premier et ignore tous les autres.

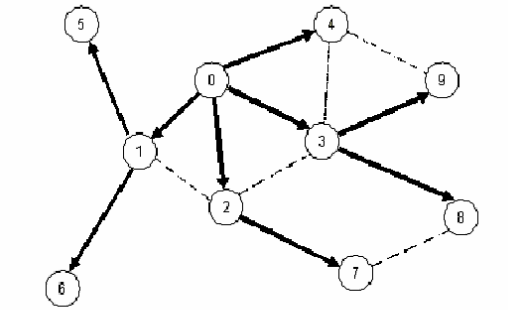

FIGURE 3.7 – *L'arbre final Construit.*

Pour la construction et la distribution de la clef de groupe, *Rony Hasinur Rahman et al* [48] divisent cette tâche en deux phases :

1. *Phase 01 :* chaque nœud feuille génère sa propre contribution et l'envoie en unicast à son parent. A la réception de toutes les contributions de leurs fils, les nœuds internes génèrent aussi leur contributions et envoient le tous en unicast à leurs parents. Le processus se répète jusqu'a ce que la racine soit atteinte.

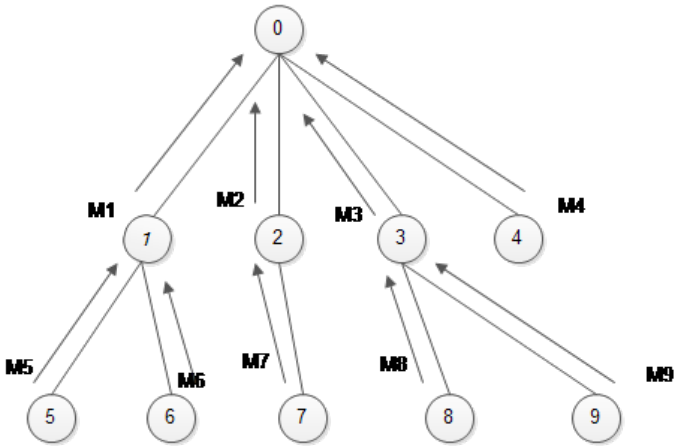

FIGURE 3.8 – *La Phase 01 : collecte de contributions.*

2. *Phase 02 :* A la réception de toutes les contributions de ses fils, la racine envoie en unicast à chacun de ses fils l'information necessaire pour la construction de la clef de groupe. Après avoir reçu le message de la racine, les nœuds internes envoient un message similaire en unicast à leur fils. Le processus s'arrête quand tous les nœuds feuilles auront reçus le message, et ainsi tous les nœuds valides auront la clef du groupe K.

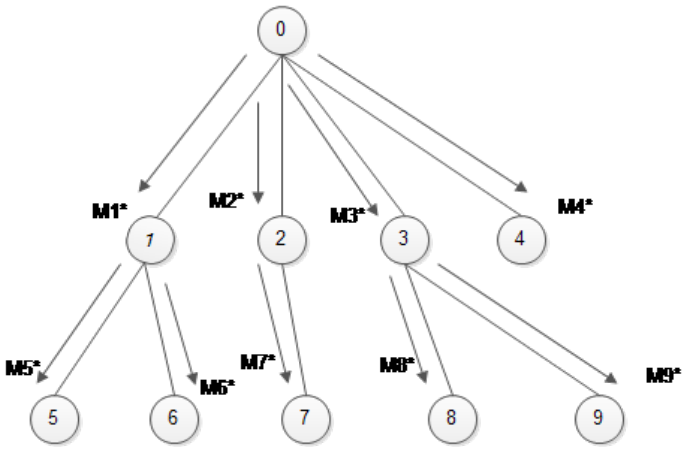

FIGURE 3.9 – *La Phase 02 : distribution et authentification.*

L'idée de *R. H. Rahman et al* [48] est de distribuer en unicast les messages d'une manière sécurisée pour la construction de la clef de groupe parmi tous les nœuds du réseau en utilisant un arbre couvrant. Cela engendre un nombre important de messages envoyés dans le réseau.

R. H. Rahman et al [48] utilisent une approche orientée topologie sans clusterisation. L'utilisation d'un algorithme de clusterisation permet de bien hiérarchiser le réseau et de le diviser en sous-groupes, ce qui permet de réduire le phénomène *1-affecte-n* et l'impact de la redistribution de nouvelles clefs du moment que ça va se faire uniquement dans les groupes concernés. Mais, ce n'est pas le cas dans le protocole proposé par *R. H. Rahman et al* [48], car lors d'un événement provoqué par un nœud, c'est la clef de tout le réseau qui sera mise à jour.

3.5 Tableau comparatif et récapitulatif des protocoles d'accord de clef de groupe

Dans cette section nous donnons un tableau comparatif des protocoles d'accord de clef de groupe que nous avons étudiés (Table 3.1).

			Passage à l'échèle			
Protocoles	**Pré-requis**	**Service de sécurité**	**Surcoût de calcul**	**Surcoût de stockage**	**Nombre de messages**	**Nombre de tour**
Xiang-Yang Li et al (2002)	construction d'un CDS	confidentialité	génération de clef de groupe et des sous-groupes	clef de groupe et de chaque sous-groupe et clef de l'épine dorsale	$\Theta(n)$	$n+2$
Shanyue Bu et al (2011)	système à clef publique	confidentialité, authentification et intégrité	calcule des signatures numériques et du haché	clef de groupe GK	$n-1$	$n-2$
Daniel Augot et al (2007)	système de chiffrement basé sur l'identité	confidentialité	calcule de la clef de groupe	une clef de groupe et les clefs secrètes	$m+1$	3
Konstantinou (2011)	algorithme de clusterisation, centre de génération de clefs (KGC)	confidentialité et authentification	calcule de la clef secrète commune	clefs publiques et clef secrète commune	$4n$	$\log_2(\frac{n}{3})$
J. C. Ming Teo (2005)	algorithme de clusterisation, protocole BD [38]	confidentialité	chiffrement, déchiffrement et calcule de la clef de groupe	clefs de sous-groupes, clef de chaque niveau, clef du réseau	$c+2$	$\log_c n$

			Passage à l'échèle			
Protocoles	**Pré-requis**	**Service de sécurité**	**Surcoût de calcul**	**Surcoût de stockage**	**Nombre de messages**	**Nombre de tour**
Li Rongsen et al (2010)	algorithme de clusterisation, mécanisme de certification	confidentialité et authentification	calcule des certificats, chiffrement et déchiffrement des messages	n clefs temporaires et une clef de groupe	$4n$	$(\frac{4n}{2}) + c$
Guoxiang Yao et al (2010)	mécanisme d'authentification, algorithme de clusterisation	confidentialité et authentification	calcule de la clef de groupe et du secret Q_i	clefs publiques et privées	$n-2$	m
AT-GDH (2008)	algorithme de clusterisation, construction d'un arbre couvrant	confidentialité	chiffrement et déchiffrement de l'exposant secret	clef privée et clef secrète du groupe	$o(n^2) + o(clogc)$	$3 + 2[logc]$
Yang Yang et al (2013)	mécanisme de certification, générateur de clefs privées (PKG)	confidentialité et authentification	calcule de la clef de groupe, génération des clefs secrètes et publiques, calcule de l'entête d'encapsulation (Hdr)	clef de groupe	$o(1)$	q

			Passage à l'échèle			
Protocoles	**Pré-requis**	**Service de sécurité**	**Surcoût de calcul**	**Surcoût de stockage**	**Nombre de messages**	**Nombre de tour**
Magliveras et al (2008)	Protocole de Diffie-Hellman et théorème du reste chinois	confidentialité, authentification et intégrité	chiffrement et déchiffrement des messages, calcule du reste chinois et des contributions publiques	clef de groupe GK	$2n$	2
TFAN (2010)	Diffie-Hellman et cryptographie à clef publique	confidentialité	diffusion des clefs publiques	clef de groupe	$n + 1 + (q2 - q + 1) * (h + 1)$	$q + 2$
R. H. Rahman et al (2008)	arbre couvrant, mot de passe faible P	confidentialité et authentification	chiffrement et déchiffrement des messages	clef de groupe K	$2n - 1$	$2log_k(n)$

TABLE 3.1 – *Tableau comparatif et récapitulatif des protocoles d'accord de clef de groupe.*

Remarque

n représente le nombre total de nœuds dans le réseau, c est le nombre de clusterhead et h est la hauteur de l'arbre.

3.6 Conclusion

Dans ce chapitre, nous avons présenté certains protocoles d'accord de clefs de groupe existants, et nous avons pu constater l'importance de l'établissement de tels protocoles dans les réseaux mobiles Ad Hoc, qui est un élément essentiel dans les communications multicast sécurisées. Ce genre de protocole doit assurer la confidentialité des données en chiffrant le flux du coté de la source et en le déchiffrant du coté des utilisateurs avec la clef de groupe TEK.

Cependant, le but des protocoles d'accord de clef de groupe dans les réseaux mobiles Ad Hoc étant l'optimisation de la génération des clefs, leur distribution et leur maintenace. Nous avons également effectué une étude comparative de certains protocoles d'accord de clef de groupe, en se basant sur des critères tel que le passage à l'échelle, nombre de tours...etc.

Chapitre 4

Proposition d'un protocole d'accord de clef de groupe pour les MANETs

4.1 Introduction

La communication de groupe a été un domaine de recherche très actif pendant longtemps, parce que plusieurs applications l'exigent. Avec l'apparition des besoins de sécurité dans les réseaux sans fil et plus spécifiquement dans les réseaux Ad Hoc, plusieurs protocoles d'accord de clef de groupe ont été adapté pour assurer un secret parfaitement partagé entre les différents membres du groupe. Cet accord est devenu une tache difficile à mettre en œuvre, surtout lorsqu'un membre décide de rejoindre où de quitter le groupe. En effet, la clef de groupe devrait être mise à jour pour préserver la confidentialité au sein de groupe.

Dans ce qui suit, nous proposons une amélioration du protocole de *R. H. Rahman et al* [48], en utilisant un algorithme de clusterisation [5] qui réduira le phénomène *1-affecte-n*. Cette amélioration réduit aussi le nombre de messages envoyés dans le réseau en adaptant la diffusion multicast.

4.2 Problématique

- Comment peut-on réduire l'impact d'un évènement provoqué par un nœud appartenant à un réseau donné sur l'ensemble de ce dernier ?

- Est-il possible d'utiliser la diffusion multicast tout en garantissant que seules les entités autorisées puissent calculer la clef du groupe ?

- Comment garantir la confidentialité passée et future dans une communication de groupe ?

4.3 Hypothèses

- On suppose que tous les liens entre les nœuds du réseau sont bidirectionnels, ce qui veut dire que chaque nœud peut envoyer et recevoir des messages.

- On suppose l'existence d'un PKI (*Public Key Infrastructure*) qui garanti la gestion des clefs publiques et qui permet d'assurer qu'une clef publique est bien celle de l'entité que l'on croit être.

4.4 Principe de la proposition

Nous proposons une amélioration pour le protocole de *R. H. Rahman et al* [48], en utilisant l'algorithme de clusterisation de B. S. Haggar [5] pour diviser le réseau en sous-groupes "*clusters*", car cela permet de réduire considérablement le phénomène "*1-affecte-n*" (C.-à-d. réduire le nombre de nœuds affectés par un événement d'adhésion, départ ou de déplacement d'un nœud). En effet le protocole de *R. H. Rahman et al* [48] utilise une structure hiérarchique sans clusterisation, ce qui provoque la réinitialisation de tout le réseau lors d'un événement. Nous utilisons aussi l'algorithme de construction d'arbre de cluster de [54] qui permet de bien hiérarchisé le réseau.

Notre mélioration permet aussi de réduire le nombre de messages envoyés en unicast, en utilisant dans la phase de distribution de l'information nécessaire pour la construction de groupe une diffusion en multicast.

Dans ce qui suit, nous décrivons brièvement l'algorithme de clusterisation [5] utilisé ainsi que l'algorithme de la construction de l'arbre de cluster [54]. Ensuite, nous présentons les phases de création des clefs des sous-groupes et de la clef du groupe, ainsi que les mises à jour apportées lors d'un événement. Finalement, nous évaluons les performances de notre amélioration en les comparant au protocole de *R. H. Rahman et al* [48].

4.4.1 La clusterisation

Nous avons utilisé pour l'étape de clusterisation l'algorithme de *B. S. Haggar* [5]. Cet algorithme permet de diviser le réseau en clusters, et cela se fait par échange périodique de message "*Hello* " entre chaque nœud du réseau. Chaque message "*Hello*" transmis par un nœud contient son identité *id* et son *statut*. Le nœud dont l'identité est plus grande au sein d'un même cluster sera clusterhead. À la réception de ce message, tous les voisins deviennent alors des nœuds membres, dans le cas où un nœud se trouve être entouré de plusieurs clusterheads, il devient alors un nœud de passage (*Passerelle* où *Gataway*). À la fin du processus, chaque nœud sera dans un des trois états : Clusterhead, nœud membre où nœud de passage. Si deux clusterheads deviennent voisins alors, le premier nœud qui détecte le conflit par l'intermédiaire des messages "*Hello*" reçus compare son

identité à celle du clusterhead voisin. Si son identité est plus petite, il abandonne son statut de clusterhead et devient nœud membre, dans le cas contraire, il conserve le rôle de clusterhead.

4.4.2 Construction de l'arbre de clusters

Pour la construction d'arbre de clusters, nous avons utilisé l'algorithme [54], qui est basé sur les identités associées à chaque cluster. La construction de l'arbre de clusters se fait par échange périodique des messages "*Hello*", chaque message transmis par un nœud u contient : son identité ($\mathtt{id}_u$), son statut ($\mathtt{statut}_u$) et l'identité de son cluster ($\mathtt{cl}_i$). À la fin du processus, le cluster ayant la plus grande identité dans le réseau sera racine de l'arbre de clusters (voir figure 4.1).

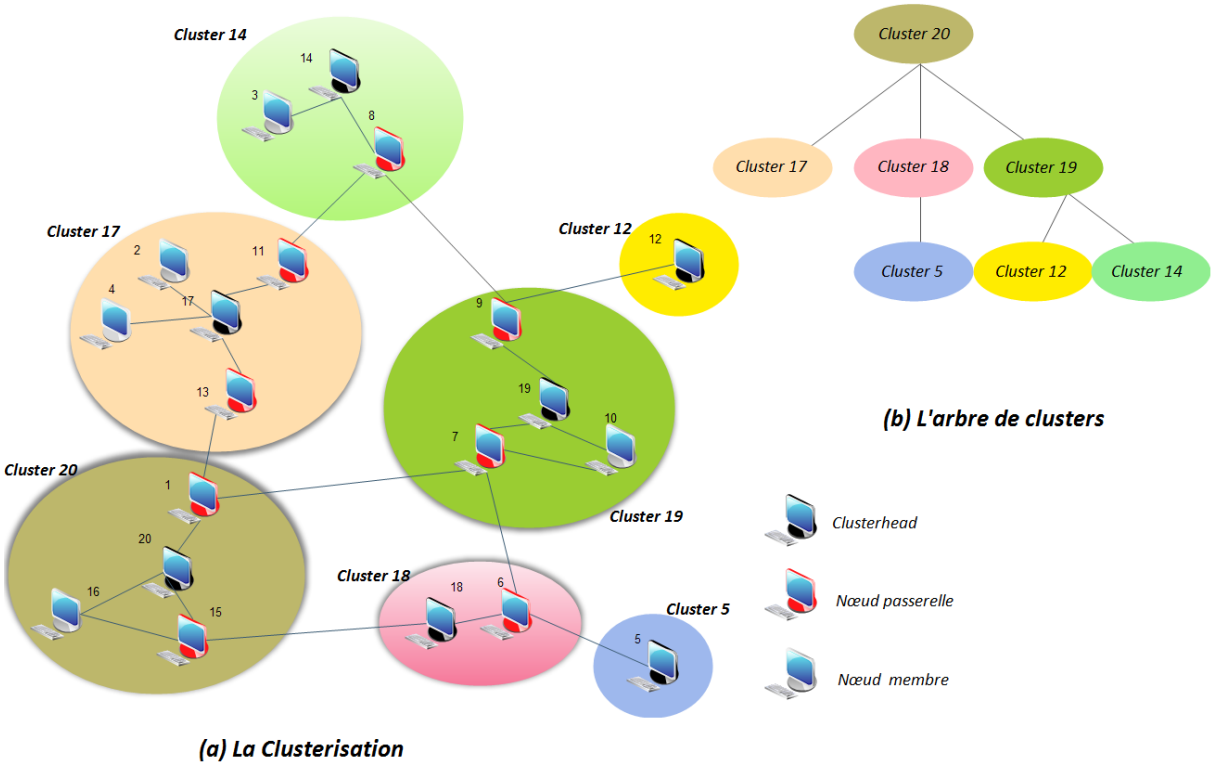

FIGURE 4.1 – *L'arbre de clusters.*

Cependant, chaque nœud u choisit comme père le cluster ayant la plus grande identité qu'il a détecté. Ensuite, le nœud u propage cette information ($\mathtt{cl}_u$, *pere*($\mathtt{cl}_u$)) à ses clusters voisins. À la réception de message du u par un nœud v, si la plus grande identité détectée par v auparavant est plus petite que celle annoncée par u, v choisit $\mathtt{cl}_u$ comme père et propage l'information à ses clusters voisins. Ce processus se répète jusqu'à ce que l'arbre soit construit et que le cluster ayant la plus grande identité du réseau soit choisit comme la racine de l'arbre. À la fin du processus,

chaque cluster connait la racine de l'arbre et la liste des clusters permettant de l'atteindre.

4.4.3 Génération de clefs

Selon l'approche que nous avons utilisée, pour garantir la confidentialité des données nous avons besoin de deux clefs partagées :

1. Une clef du groupe externe (KG) est employée pour chiffrer et déchiffrer les messages diffusés parmi les clusterheads.

2. Une clef de sous-groupe (KR) est employée pour chiffrer et déchiffrer les messages diffusés parmi les membres de chaque cluster (y compris le clusterhead).

Clef intra-cluster (KR)

La clef du sous-groupe (KR), aussi appelée "*la clef intra-cluster* " est employée pour la communication entre un clusterhead et les nœuds membres d'un même cluster. Tous les membres de chaque cluster s'accordent sur la clef du cluster, et chaque clusterhead peut représenter son cluster en employant cette dernière (clef du cluster) en tant que son exposant secret.

Tous les membres d'un cluster participeront à l'établissement de la clef du cluster (KR) d'une manière distribuée, comme suit :

1. Chaque nœud i génère sa contribution (α^{g_i}) et l'envoie à son clusterhead, tel que α est le générateur de groupe.

2. Après avoir reçu toutes les contributions de tous les nœuds de son cluster, le clusterhead génère sa contribution, calcule la clef KR, tel que $KR = (\alpha^{\sum g_i})$ et diffuse l'information nécessaire pour la construction de la clef de cluster comme suit :

   $M_i = (M\prime, M\prime\prime)$, tel que $M\prime$ est la contribution du clusterhead chiffrée avec la clef P_i du cluster (P_i est une clef partagée entre les nœuds du cluster i). $M\prime\prime$ c'est les contributions de tous les nœuds du cluster excepté le clusterhead.

3. A la réception du message M_i, chaque membre d'un cluster déchiffre la partie $M\prime$ avec la clef P_i, récupère la contribution de la racine et calcule la clef du cluster KR.

Exemple

On prend le cluster 14 comme exemple (figure 4.1). les nœuds 3 et 8 génèrent leurs contributions α^{g_3} et α^{g_8} respectivement, et chacun d'eux envoie sa contribution au clusterhead 14. Après avoir reçu les contributions de tous les nœuds de son cluster, le clusterhead 14 génère à son tour sa contribution $\alpha^{g_{14}}$, et diffuse le message suivant aux membres de son groupe :

$M_{14} = (P_{14}(\alpha^{g_{14}}), \alpha^{g_3 g_8})$.

Les nœuds 3 et 8 peuvent finalement calculer la clef du cluster KR, en déchiffrant la partie $M\prime$ du message reçu M_{14} avec la clef P_{14}.

Clef inter-cluster (KG)

La clef inter-cluster KG est établie par tous les clusterheads selon la structure d'arbre définit précédemment, d'une façon distribuée, suivant les étapes ci-après :

1. Chaque clusterhead feuille génère sa contribution $\alpha^{g'_i}$ et l'envoie à son parent.

2. Après la réception de toutes les contributions de ses fils, chaque clusterhead interne génère sa contribution et envoie le tout à son parent comme suit :

   $M_j = (M\prime, M\prime\prime)$, tel que $M\prime = \alpha^{g'_j}$ et $M\prime\prime = \alpha^{g'_{fils}}$.

3. Après avoir reçu tous les messages de ses fils, le clusterhead racine génère sa contribution, calcule la clef de l'arbre de cluster KG, tel que $KG = \alpha^{\sum g'_j}$ et diffuse l'information nécessaire pour la construction de la clef de l'arbre vers chacun de ses fils comme suit :

   $M*_j = (M\prime, M\prime\prime)$, tel que $M\prime$ est la contribution de la racine chiffrée avec la clef P (P est la clef partagée entre les clusterhead), $M\prime\prime$ est la contribution de tous les clusterheads.

4. Une fois que tous les clusterheads ont reçus leurs messages respectifs, ils déchiffrent $M\prime$ avec la clef de l'arbre P, puis ils calculent la clef inter-cluster KG.

Exemple

1. ***La collecte des contributions***

   Tous les clusterheads du réseau génèrent leurs contributions respectives et les envoient aux cluster racine (clusterhead 20). Si on prend l'exemple du clusterhead 17 (voir figure 4.1) alors :

   – Le clusterhead 17 envoie $\mathsf{M}_{17} = (\alpha^{g'_{17}}, -)$.

2. ***La distribution de l'information***

   Après la réception de toutes les contributions de ses fils, la racine (le clusterhead 20) génère à son tour sa contribution, calcule la clef de groupe et envoie à ses fils le résultat suivant :

   $\mathsf{M}*_{20} = (\mathsf{P}(\alpha^{g'_{20}}), (\alpha^{g'_{17}}, \alpha^{g'_{18}}, \alpha^{g'_{19}}, \alpha^{g'_{5}}, \alpha^{g'_{12}}, \alpha^{g'_{14}}))$.

3. Finalement, quand tous les clutsers feuilles reçoivent leurs messages, chaque nœud a tout ce qui'il faut pour construire la clef de l'arbre KG.

4.4.4 Mise à jour des clefs

- ***Adhésion d'un nœud***

  Lorsqu'un nouveau nœud adhère au groupe, il initie une communication avec les nœuds voisins. Ce groupe lance un processus de renouvellement de clefs (étape de génération de clefs de cluster KR). Dans le cas où le nouveau membre sera élu comme clusterhead du groupe, une mise à jour de la clef de cluster KR ainsi qu'une mise à jour de la clef de l'arbre KG seront effectuées.

- ***Départ d'un membre***

  Quand un nœud membre quitte le groupe, la clef KR du cluster à qui appartient ce nœud doit être mise à jour. Ce nœud informe ces voisins de son départ, un processus de renouvellement de clef du cluster est ainsi lancé. Mais si le nœud membre qui quitte le groupe est un clusterhead, la clef du cluster KR et la clef de l'arbre KG doivent être changées. Un nouveau clusterhead sera désigné dans le cluster concerné selon l'algorithme de clusterisation [5]. Celui-ci initie les procédures de génération des clefs (KR et KG).

4.4.5 Analyse de sécurité

- ***Confidentialité passée (*Backward Secrecy*)***

  Notre amélioration permet de garantir la confidentialité passée (*Backward Secrecy*) en appliquant les procédures de génération de clefs lors d'une adhésion d'un nœud au groupe, car il ne peut pas déchiffrer les messages antérieurs.

- ***Confidentialité future (*Forward Secrecy*)***

  Notre amélioration permet aussi d'assurer la confidentialité future (*Forward Secrecy*). Lorsqu'un membre quitte le groupe, il n'est pas en mesure de déchiffrer les messages qui circulent dans le réseau après son départ. Car après chaque évènement de départ, une mise à jour des clefs sera effectuée.

4.4.6 Evaluation des performances

L'évaluation des performances d'un système est une phase indispensable, qui permet de valider ses atouts et de les valoriser, ainsi que de le comparer avec d'autres approches existantes selon des métriques bien spécifiées tel que le nombre de messages envoyés et le nombre de nœud affecté par un évènement.

- **Nombre de messages**

  Dans la Figure 4.2, nous avons comparé les coûts de communications de notre amélioration avec le protocole de *R. H. Rahman et al* [48]. Le coût de communications est présenté par le nombre de messages transmis et reçus. L'idée de *R. H. Rahman et al* [48] est de distribuer en unicast les messages d'une manière sécurisée pour la construction de la clef de groupe parmi tous les nœuds du réseau en utilisant un arbre couvrant. Par contre, notre amélioration est basée sur l'utilisation d'un arbre de clusters, ce qui réduit le coût de communications dans le réseau. *R. H. Rahman et al* [48] utilisent $2n-1$ pour le nombre de messages, nous calculons le nombres de messages selon la formule : $n + 2c - (cf + 1)$, tels que c est le nombre de clusters, cf est le nombre de clusters feuille. Cette formule est calculée comme suit :

  Construction de la clef intra-cluster
  Dans la phase de collecte des contributions, chaque nœud membre envoie sa contribution excepté les clusterhead donc c'est : $n-c$, dans la phase de distribution de l'information, chaque clusterhead diffuse en multicast un message aux membres de son cluster donc c'est c.

  Pour la construction des clefs de tous les clusters, on a $n - c + c$ donc c'est n.

Construction de la clef inter-cluster
Dans la phase de collecte des contributions, chaque clusterhead envoie sa contribution à son père excepté le clusterhead racine donc c'est $c-1$, dans la phase de distribution de l'information, chaque clusterhead parent diffuse un message en multicast à ses fils excepté les clusterheads feuilles $c-cf$, tel que cf est le nombre de clusters feuille.
Donc pour la construction de la clef inter-cluster, on a $c-1+c-cf \Longrightarrow 2c-(cf+1)$.

le nombre de messages total est $n+2c-(cf+1)$.
Nous remarquons dans la Figure 4.2, que notre approche réduit bien le nombre de messages échangés parmi les nœuds membres dans le réseau, et cela est du à la diffusion des messages de la part des clusterheads lors de la construction des clefs de clusters ainsi que la clef de groupe. Mais ce n'est pas le cas pour le protocole de *R. H. Rahman et al* [48], car l'échange des messages est en unicast.

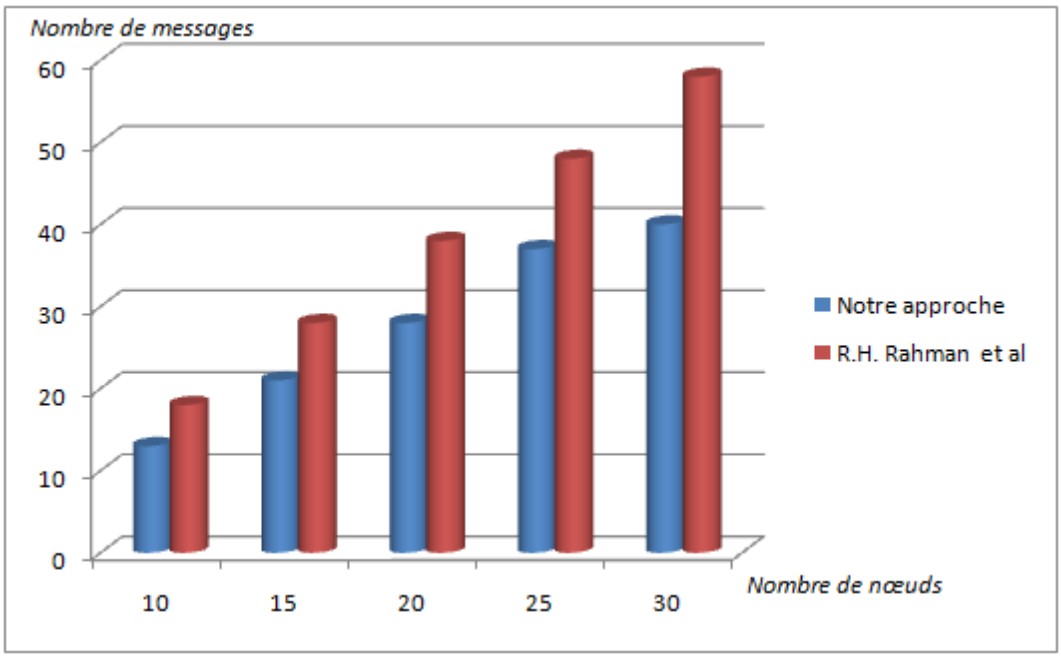

FIGURE 4.2 – *Variation du nombre de messages en fonction du nombre de nœuds dans le réseau.*

- **L'événement 1-affecte-n**

  Pour satisfaire les besoins de confidentialité au sein du groupe dans un environnement mobile, une redistribution de la clef doit être effectuée à chaque fois qu'un membre adhère où quitte le groupe. Ceci consiste en la génération d'une nouvelle clef TEK et la distribuer uniquement aux membres valides du groupe en incluant le nouveau membre dans le cas d'une adhésion et en l'excluant dans le cas d'un départ.

Dans un premier temps, nous nous sommes intéressés à l'étude de l'impact du nombre de nœuds dans le réseau sur le nombre de membres affectés par les changements d'adhésion (*Join* ou *Leave*) : *1-affecte-n*. Selon l'approche de *R. H. Rahman et al* [48], lors d'un évènement c'est tous les nœuds du réseau qui sont affectés, mais c'est pas le cas pour notre solution, car lors d'un évènement provoqué par un nœud membre, c'est seulement les nœuds de son cluster qui seront affectés, et lors d'un évènement provoqué par un clusterhead c'est les nœuds de son cluster ainsi les clusterheads formant l'arbre qui seront affectés.

Nous remarquons dans la Figure 4.3 que notre amélioration réduit bien le phénomène *1-affecte-n* par rapport à la solution proposée par *R. H. Rahman et al* [48], car l'utilisation des clusters qui permet d'organiser le réseau en sous-groupes avec des clefs TEK différentes réduit l'impact de la redistribution de nouvelles clefs du moment que ça va se faire uniquement dans les groupes concernés.

Le nombre de nœuds affectés par un évènement provoqué par un nœud membre est NC, tel que NC est le nombre de nœud du cluster concerné. Par contre, dans le cas d'un évènement provoqué par un clusterhead, le nombre de nœuds affectés est $NC + c$, tel que c est le nombre de clusters.

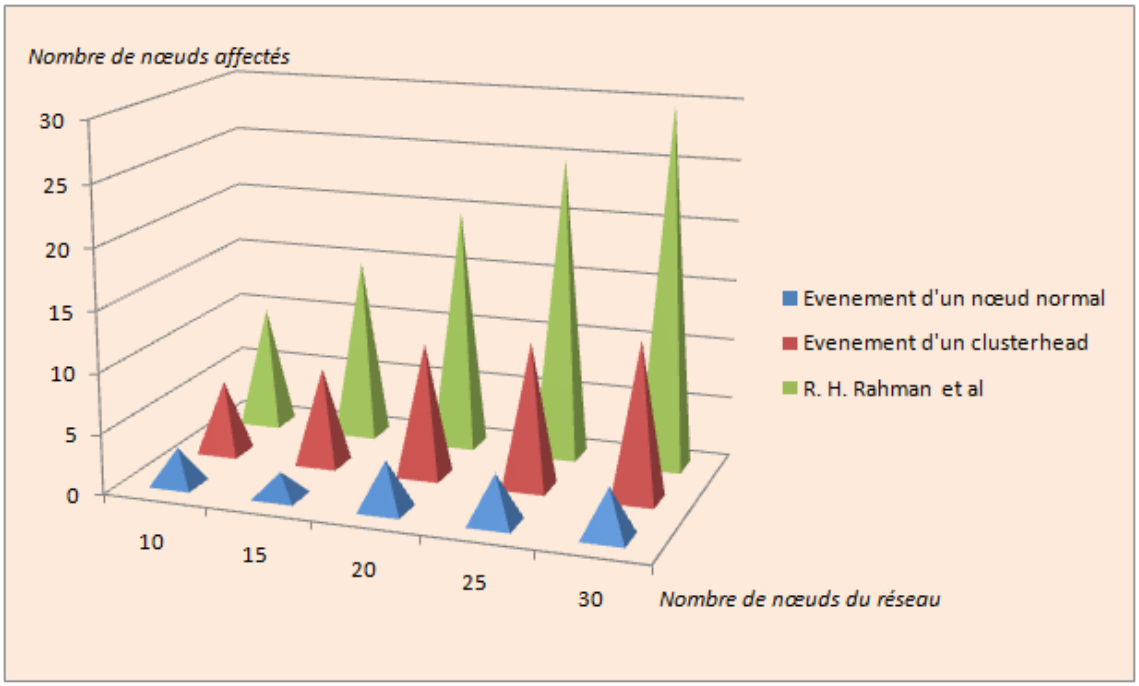

FIGURE 4.3 – *L'événement 1-affecte-n.*

Comme nous pouvons voir dans la Figure 4.3, il est évident que le nombre de nœuds affectés par les événements de départ ou d'adhésion d'un membre au

groupe diminue selon notre amélioration. Car lorsqu'un nœud membre rejoint ou quitte le groupe, c'est seulement la clef du cluster à qui appartient ce nœud qui doit être mise à jour. Mais dans le cas où le nouveau membre sera élu comme clusterhead du groupe ou lorsqu'un clusterhead quitte le groupe, la clef du cluster et la clef du réseau doivent être changées. Par contre, selon la solution proposée par *R. H. Rahman et al* [48], c'est la clef de tout le réseau qui va être mise à jour.

4.5 Conclusion

Dans ce chapitre, nous avons proposé une amélioration du protocole de *R. H. Rahman et al* [48], qui garantit la confidentialité des données. Le principe de cette approche est de construire un arbre de clusters pour pouvoir faire de la diffusion dans le réseau. Nous avons aussi évaluer les performances de notre amélioration en les comparant au protocole de *R. H. Rahman et al* [48], et nous avons montré que notre approche apporte un gain en termes de nombre de messages échangés pour la distribution des clefs aux membres du groupe.

Notre amélioration réduit considérablement le phénomène *1-affecte-n* au sein du groupe, et garantit les secrets passés *(Backward Secrecy)* et futures *(Forward Secrecy)* en appliquant les procédures de génération de clefs à chaque événement, que ce soit un événement de départ, d'adhésion ou de déplacement d'un membre.

Conclusion Générale et Perspectives

L'accord de clef de groupe est une opération essentielle dans la sécurité des communications de groupe. En effet, il permet la distribution de la clef de groupe à la source pour chiffrer le flux multicast et aux récepteurs pour le déchiffrer, assurant ainsi la confidentialité des données.

Dans ce projet nous avons étudié globalement les problèmes de sécurité et de communication de groupe dans les réseaux mobiles Ad Hoc, plus précisément les problèmes d'accord de clef de groupe. Le but de ce projet est de proposer une nouvelle approche utilisant une structure hiérarchique du réseau pour garantir la confidentialité des données permettant de réduire le phénomène 1-affecte-n au sein du groupe. Cette approche doit en plus permettre le passage à l'échelle, tout en s'adaptant aux caractéristiques des réseaux mobiles Ad Hoc.

Nous avons proposé une classification des protocoles d'accord de clefs de groupe dans les réseaux mobiles Ad Hoc. Nous avons comparé ensuite notre solution avec celle proposée par *Rony Hasinur Rahman et al* [48], selon des métriques bien spécifiés telles que le nombre de messages et le phénomène *1-affecte-n*. Et nous avons pu constater l'importance de l'établissement de tels protocoles dans les réseaux mobiles Ad Hoc, et que l'amélioration des performances de ce genre de réseaux peut être réalisée en groupant les nœuds du réseau en sous-groupes. Nous avons mis en avant notre contribution en proposons une amélioration basée sur la construction d'un arbre couvrant à base de clusters. Le cluster ayant l'identité la plus grande devient racine de l'arbre.

A travers l'évaluation des performances de nos contributions, et après les avoir comparées avec le protocole de *Rony Hasinur Rahman et al* [48], nous avons pu montrer que la clusterisation [5] apporte une économie en termes de délai moyen de transmissions de clefs. Nous avons montré également, dans ce sens, que notre amélioration à des meilleures performances que celle proposé par *Rony Hasinur Rahman et al* [48].

Comme perspectives à nos travaux, nous prévoyons dans un premier temps d'étendre

notre amélioration basée sur une approche d'accord de clef de groupe, en définissant un système de gestion de confiance entre les chefs de groupe multicast, nous prévoyons également de simuler nos travaux. Notre objectif est d'aboutir à un modèle d'évaluation de performance uniforme, permettant de comparer les différentes approches selon des métriques bien spécifiques. Ce modèle permettra entre autres d'évaluer l'impact de la dynamicité d'adhésion des membres au groupe multicast, d'évaluer le support du passage à l'échelle, ainsi que de chiffrer le surcoût de stockage, de calcul et de communication à travers des simulations réelles des protocoles étudiés.

Bibliographie

[1] R. Touati et R. S. Chellali, *"Sécurité des communications de groupe dans les réseaux Ad Hoc mobiles (MANETs)",* Memoire Master Recherche, Université A. Mira de Bejaia, 2011.

[2] N. Mansouri, *"Protocole de routage multichemin avec équilibrage de charge dans les réseaux mobiles Ad Hoc",* Cycle de formation des ingénieurs en Télécommunications, Ecole supérieure des communications de Tunis, 2007.

[3] O. Cheikhrouhou, *"Sécurité des réseaux Ad Hoc",* Mémoire d'ingénieur en Génie Informatique, Ecole Nationale d'Ingénieurs de Sfax, 2005.

[4] B. S Haggar, *"Les protocoles de routage dans les réseaux Ad Hoc",* Memoire Master Recherche STIC, Université de Reims - UFR Sciences, 2007.

[5] B. S. Haggar, *"Clusterisation Auto-stabilisante pour les Réseaux Ad Hoc",* 10èmes journées Doctorales en Informatique et Réseaux, Belfort, France, pages 31-36, 2009.

[6] J. C. Ming Teo et C. How Tan, *"Energy-Efficient and Scalable Group Key Agreement for Large Ad Hoc Networks",* PE-WASUN'05, Copyright ACM, pages 114-121, Canada, 2005.

[7] E. Konstantinou, *"Efficient cluster-based group key agreement protocols for wireless ad hoc networks",* Journal of Network and Computer Applications 34, pages 384-393, 2011.

[8] D. Congwei, L. Rongsen et D. Wenhua, *"An Efficient Key Agreement Protocol in Cluster-Based MANETs",* International Conference on Computer Application and System Modeling (ICCASM), 2010.

[9] M. Hietalahti, *"A Clustering-based Group Key Agreement Protocol for Ad-Hoc Networks",* Electronic Notes in Theoretical Computer Science 192, pages 43-53, 2008.

[10] S. Magliveras, W. Wei et X. Zou, *"Notes on the CRTDH Group Key Agreement Protocol",* The 28th International Conference on Distributed Computing Systems Workshops, pages 406-411, 2008.

[11] Xiang-Yang Li, Y. Wang et O. Frieder, *"Efficient Hybrid Key Agreement Protocol for Wireless Ad Hoc Networks",* In Proceedings of the Eleventh International Conference On Computer Communications and Networks, pages 404-409, 2002.

[12] D. Augot, R. Bhaskar, V. Issarny et D. Sacchetti, *"A three round authenticated group key agreement protocol for ad hoc networks",* Pervasive and Mobile Computing 3, pages 36-52, 2007.

[13] M. S. Bouassida, *"Sécurité des communications de groupe dans les réseaux ad hoc",* Thèse de Doctorat, Université de Henri Pointcaré - Nancy 1, 2006.

[14] Y. Challal, H. Bettahar et A. Bouabdallah, *"Tutorial sur la sécurité dans les communications Multicast",* 3rd International Conference : Sciences of Electronic, Tunis, SETIT 2005.

[15] M. Allouti et C. Yousfi, *"Implémentation d'un protocole d'accord de clefs de groupe dans les réseaux Ad Hoc auto-configuré",* Mémoire d'ingénieur en Genie Informatique, Université A. Mira de Bejaia, 2010.

[16] Z. Azzoug, *"Evaluation et implémentation d'un protocole d'accord de clef de groupe avec clusterisation dans les réseaux sans fil Ad Hoc",* Memoire Master Professionnel, Université A. Mira de Bejaia, 2011.

[17] W. Diffie et M. Hellman, *"New Directions in Cryptography."* IEEE Transactions on Information Theory, Vol. 22, No. 6, pages 644-654, 1976.

[18] J. Katz et M. Yung, *"Scalable protocols for authenticated group key exchange,"* In Advances in Cryptography - Crypto'03, Lecture Notes in Computer Science Vol. 2729, pages 110-125, 2003.

[19] M. Steiner, G. Tsudik et M. Waidner, *"Diffie-Hellman key distribution extended to group communication,"* 3rd ACM Conference on Computer and Communications Security, ACM Press, pages 31-37, 1996.

[20] Yang Yang, Yupu Hu, M. Chunhui Sun, Chao Lv et L. Zhang, *"An Efficient Group Key Agreement Scheme for Mobile Ad-Hoc Networks,"* The International Arab Journal of Information Technology, Vol. 10, No. 1, 2013.

[21] D. Boneh et X. Boyen, *"Efficient Selective-ID Secure Identity-Based Encryption without Random Oracles,"* In Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology-EUROCRYPT, Switzerland, vol. 3027, pages 223-238, 2004.

[22] M. Conti, I. Chlamtac et J. J-N.liu, *"Mobile ad hoc networking : imperatif and challenges,"* Proceedings of IEEE International Conference on Communications, Elsevier 2003.

[23] Y. Hu, D. Johnson et D. Maltz, *"The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,"* IETF RFC 4728, 2007.

[24] R. M. Pearlman et Z. J. Haas, *"The Zone Routing Protocol (ZRP) for Ad Hoc Networks,"* IETF Internet Draft, 2002.

[25] T. Clausen et P. Jaquet, *"Optimized Link State Routing Protocol (OLSR),"* RFC 3626 (Experimental), 2003.

[26] C. Karlof et D. Wagner, *"Secure routing in wireless sensor networks : attacks and countermeasures,"* ad hoc Networks 1, pages 293-315, 2003.

[27] Z. Liu, A. W.Joy et R. A.Thompson, *"A Dynamic Trust Model for Mobile ad hoc Networks,"* Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems,(FTDCS'04), 2004.

[28] X. Perséguers, *"La sécurité dans les réseaux de capteurs sans fil,"* Memoire Master, Ecole Polytechnique Fédérale de Lausanne (EPFL) et Centre Suisse d'Electronique et de Microtechnique (CSEM), 2005.

[29] V. Gayraud, L. Nuaymi, F. Dupont, S. Gombault et B. Tharon, *"La Sécurité dans les Réseaux Sans Fil ad hoc,"* Actes du symposium SSTIC03, 2003.

[30] A. Adolf, *"La Sécurité dans les Réseaux Sans Fil ad hoc,"* Memoire Master Recherche, Informatique et Sciences de l'Information et de la Communication, Université de Limoges, 2007.

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

[32] R. Balachandran, B. Ramamurthy, X. Zou et N. Vinodchandran, *"CRTDH : An efficient key agreement scheme for secure group communications in wireless ad hoc networks,"* Proceedings of IEEE International Conference on Communications (ICC), pages 1123-1127, 2005.

[33] Yondae Kim, A. Perring et G. Tsudik, *"Tree-Based Group Key Agreement,"* ACM Transactions on Information and System Security, vol. 7, no. 1, 2004.

[34] Y. Kim, A. Perring et G. Tsudik, *"Group Key Agreement Efficient in Communication,"* IEEE Transactions on Computers, vol 53, no 7, pages 905-921, 2004.

[35] Shan-shan Tu, Chun-bo Ma et fa-liang AO, *"Based on Tree Structure Group Key Management fo Ad Hoc Networks,"* IEEE 978-1-4244-5326-9/10, 2010.

[36] L. Liao et M. Manulis, *"Tree-based group key agreement framework for mobile ad-hoc networks[J],"* 2007 Elsevier B. V, Future Generation Computer System 23, pages 787-803, 2007.

[37] S. Bu et H. Kou, *"A New Ad Hoc group Key Agreement Scheme,"* IEEE International Conference on Computer Science and Network Technology, 978-1-4577-1587-7/11, 2011.

[38] M. Burmester et Y. Desmedt, *"A Secure and Efficient Conference Key Distribution System,"* Proc. Advances in Cryptography - Eurocrypt '94, Lecture Notes in Computer Science vol. 950, pages 275-286, 1995.

[39] I. Kouvelas, B. Fenner, B. Cain, S. Deering et A. Thyagarajan, *"Internet Group Management Protocol, Version 3,"* RFC 3376, 2002.

[40] R. Vida et R. Costa, *"Multicast Listener Discovery Version 2 (MLDv2) for IPv6,"* IETF RFC 3810, 2004.

[41] R. Rivest, *"MD5 - Message-Digest Algorithm 5,"* MIT Laboratory for Computer Science and RSA Data Security, Inc. RFC 1321, 1992.

[42] R Rivest, *"The MD4 Message-Digest Algorithm,"* MIT Laboratory for Computer Science and RSA Data Security, Inc. RFC 1320, 1992.

[43] R. Rivest., *"The MD2 Message-Digest Algorithm,"* MIT Laboratory for Computer Science and RSA Data Security, Inc. RFC 1115, 1992.

[44] R. Housley, *"Enveloppe de clé Triple-DES et RC2,"* RSA Laboratories, RFC 32317, 2001.

[45] R. Rivest, A. Shamir et L. M. Adelman, *"A Method for Obtening Digital Signatures and Public Key Cryptosystems,"* Communication of the ACM, vol. 21, pages 120-126, 1978.

[46] A.J. Menezzs, P.C. van Oorschot et S.A. Vanstone, *"Handbook of applied cryptography,"* CRC Press, New York, 1997.

[47] *"NIST. - FIPS 197, Advanced Encryption Standard (AES),"* nov. 2001. CF. http ://csrc.nist.gov/encryption/aes/

[48] R. H. Rahman et M. L. Rahman, *"An Efficient Group Key Agreement Protocol for Ad-Hoc Networks,"* 5th International Conference on Electrical and Computer Engineering, ICECE 2008, Dhaka, Bangladesh, pages 20-22, 2008.

[49] C. E. Perkins et E. M. Royer, *"The Ad hoc On-Demand Distance Vector Protocol,"* In C. E. Perkins, editor, Ad hoc Networking, Addison-Wesley, pages 173-219, 2000.

[50] S. Patel, S. Rizvi et K. Elleithy, *"Hierarchically Segmented Routing (HSR) Protocol for MANET,"* Department of Computer Science and Engineering University of Bridgeport, 2003.

[51] C.E. Perkins et P. Bhagwat, *"Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,"* Comp. Comm. Rev., pages 234-244, 1994.

[52] V.D. Park et M.S. Corson, *"A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,"* In Proceedings of IEEE INFOCOM'97, Kobe, Japan, pages 1405-1413, 1997.

[53] G. Yao, Saizhi Ye, WeihuaXu et L. Wei, *"A Verifiable Group Key Agreement Protocol Based on Two-tier Architecture in Wireless Ad hoc Network,"* IEEE 2nd International Conference on Industrial and Information Systems, 978-1-4244-8217-7/10/, 2010.

[54] B S. Haggar, *"Auto-organisation et routage dans les réseaux mobiles ad hoc",* Thèse de Doctorat, Université de Reims Champagne-Ardenne, 2011.

[55] S. Zhu, S. Setia, S. Xu and S. Jajodia, *"GKMPAN : An Efficient Group Rekeying Scheme for Secure Multicast in Ad Hoc Networks",* In MobiQuitous, pages 42-51, 2004.

[56] V. Varadharajan, M. Hitchens and R. Shankaran, *"Securing NTDR Ad Hoc Networks",* In IASTED International Conference on Parallel and Distributed Computing and Systems, pages 593-598, Anaheim California, 2001.

[57] T. Chiang and Y. Huang, *"Group Keys and the Multicast Security in Ad Hoc Networks"*, In Proceedings on the 2003 Internationnal Conference on Parallel Processing Workshops (ICPP Workshops), page 385, 2003 .

Résumé

Un réseau mobile Ad Hoc est une collection d'entités mobiles interconnectées par une technologie sans fil, formant un réseau temporaire, sans l'aide d'aucune infrastructure fixe ni d'administration centralisée. Cette flexibilité en temps et en espace induit de nouveaux défis envers l'architecture de sécurité à mettre en œuvre pour assurer des communications multicast sécurisées.

Pour mieux répondre à cette exigence, les protocoles d'accord de clef de groupe s'avèrent être la solution. En effet, l'accord de clef de groupe est une opération essentielle dans la sécurité des communications de groupe, il est l'un des facteurs principaux de toute architecture de sécurité des communications de groupe. Il permet la distribution de la clef du groupe à la source pour chiffrer le flux multicast et aux récepteurs pour le déchiffrer, assurant ainsi la confidentialité des données. L'authentification et le contrôle d'accès des membres de groupe sont également assurés, du fait que seuls les membres autorisés qui détiennent la clef du groupe sont capables d'accéder aux données multicast émises par la source.

Dans ce rapport, nous nous intéressons au problème d'accord de clef dans les communications de groupes dynamiques. Tout d'abord, nous faisons une présentation détaillée du problème en montrant les exigences et défis. Ensuite, nous présentons les solutions existantes. Finalement, nous présentons notre amélioration.

Mots-clés : Réseaux Mobiles Ad Hoc, Sécurité, Cryptographie, Diffie-Hellman, Communication de Groupe (Multicast), Gestion de Clefs, Accord.

Abstract

A mobile Ad Hoc network is a collection of mobile entities wirelessly connected, and able to form a network on the fly without using fixed infrastructure or centralized administration. This flexibility in time and space induced new challenges towards security to ensure multicast communications.

To meet them, the group key agreement protocols proved to be the solution. Indeed, the group key agreement is a fundamental element in the multicast communication security. It allows the distribution of the group key the source to cipher the multicast flow and the receivers to decipher it, thus ensuring the data confidentiality. The authentification and the access control of the group members are also assured, owing to the fact that only the authorized members who hold the group key are able to reach multicast data transmitted by the source.

In this report, we focus on the problem of key agreement in dynamic group communications. Firstly, we make a detailed presentation of the problem by showing the requirements and challenges. Then, we present the existing solutions. And finally, we present our improvement.

Keywords : Mobile Ad Hoc Network, Security, Cryptographiy, Diffie-Hellman, Group Communication (Multicast), Key Management, Agreement.