RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A.Mira, Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de Cycle

Pour l'Obtention du Diplôme Master Recherche en Informatique
Option : Réseaux et systémes distribués

Thème

Gestion de clé de groupe dans les réseaux mobiles Ad Hoc

MEHDIOUI Feriel & HADDADOU Souraya

Président du jury : D^r TALANTIKITE Hassina née NACER, Maitre de conférence B, Université de Béjaïa.

Examinateurs : M^{me} BOUTRID Samia , Maître assistante B, Université de Béjaïa.

M^r ATTROUCHE Abd alghani, Doctorant, Université de Béjaïa.

Promotrice : M^{me} BENSAID Samia, Maître assistante A, Université de Béjaïa.

Juin 2012

Remerciements

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

Nous tenons à remercier notre promotrice Mme Samia BENSAID d'avoir cru en nous dès le début et de lui exprimer notre immense gratitude pour son encadrement, ses conseils, son soutien constant, sa générosité, sa disponibilité et sa patience... nous avons beaucoup apris à vos côtés. Merci de nous avoir fait découvrir les plaisirs du travail bien fait, nous espérons pouvoir travailler avec vous dans le futur.
Nos remerciements vont également à l'ensemble du jury pour avoir accepté de juger notre travail.
Nous tenons aussi à remercier tous nos professeurs et toute l'équipe pédagogique de l'université qui ont travaillé avec abnégation pour nous permettre de suivre ce cursus.

De même nous remercions tout ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail, soit avec leur support, leur amitié ou leur amour.

Dédicaces

A ma mère et mon père,
A mon frère et ma soeur
A tous mes ami(es).
A tous ceux que j'aime.

Votre affection ma prodiguée beaucoup de courage afin de de mener à terme ce projet.
Veuillez trouver dans ce travail le témoignage de ma reconnaissance.

Feriel

Dédicaces

A ma mère et mon père,
A mes soeurs Saida et Nacera, et à ma cousine Hanane
A Yacine.
A toute ma famille
A Sonia, Feriel et Roza
A tous mes ami(es).
A tous ceux que j'aime.

Votre affection ma prodiguée beaucoup de courage afin de de mener à terme ce projet.
Veuillez trouver dans ce trvail le témoignage de ma reconnaissance.

Soraya

Table des matières

Liste des figure

Liste des tableaux

Liste des abréviations

A

AODV : Ad-hoc On demand Distance Vector routing.
ACL : Liste De Contrôle D'accés

B

BDGKA :Burmester-Desmedt Groupe Key Agrement

C

CBRP : Cluster Based Routing Protocol.
CTCKM : Composite Tree Cluster Key Management
C-H : Hierarchique Circulair

D

DH : Diffie Hellman.
DSR : Dynamic Source Routing.
DT : Distrust

E

ECGK : Efficient Clustering scheme for Group Key management in MANETs.

F

FSR : Fisheye State Routing.

G

GDH : Goup Diffie - Hellman.
GPS : Global Positionning System.

H

HKMS : Hierarchical Key Management Scheme.

I

IEEE : Institute of Electrical and Electronics Engineers.

K

KEK : Key Encyption Key.

L

LKH : Logical Key Hierarchy.
LKT : Logical Key Tree.

M

MANET : Mobile Ad-hoc Networks.
MD2 :Message Digest 2
MD4 :Message Digest 4
MD5 :Message Digest 5

O

OLSR : Optimized Link State Radio.

P

RIPEMD-160 :RACE Integrity Primitives Evaluation Message Digest of 160 bit
PT : Partiel Trust

R

RREQ : Route Request.

S

SEGK : Simple and Efficient Group Key management scheme.
SHA : Secure Hash Algorithm

T

TEK : Trafic Encryption Key.
TRP : Two Round key agreement Protocol
TT : Total Trust

Z

ZRP : Zone Routing Protocol.

Annexe

Le protocole TRP est un protocole d'accord de clé à deux tours, l'initiateur du protocole devient leadeur du groupe, il commence par diffuser un message "INIT" pour lancer le processus de calcule de la clé qui s'étale sur deux tours :

Tour 1 : chaque participant i répond au message "INIT" en choisissant aléatoirement un secret r_i et envoie la version aveuglé g^{r_i} à l'initiateur.

Tour 2 : le leadeur du groupe l soulève le secret aveuglé reçu des membres a la puissance de son propre secret r_l et le diffuse à tous les participants $g^{r_i r_l}$. Chaque participant Vérifie si sa contribution est incluse, supprime son secret r_i à partir de $g^{r_i r_l}$ pour obtenir g^{r_l} , la clé du cluster est calculé par tout les membres de la manière suivante :

$$TEK_c = g^{r_l} * \prod g^{r_i r_l} = g^{r_l}(1 + \sum r_l) \text{ ou } i = 1, n \text{ et } i \neq l.$$

Introduction générale

Les communications ont toujours été au centre des rapports entre personnes et entre communautés. Depuis les colonnes de fumées des indiens, l'écho des tambours des africains, le courriel en passant par le courrier traditionnel et le téléphone, les moyens de communication ont pris diverses formes et se sont raffinés dans le temps. Les télécommunications connaissent une croissance phénoménale, les réseaux informatiques aussi.

L'avènement de l'Internet est une véritable révolution dans le monde des média car il a ouvert une grande vitrine sur le monde et servi comme base de lancement à de nouvelles applications de masse comme le multimédia. Mais ces applications sont généralement soutenues par un mode de communication de point à point. Or, ce mode de communication a pour conséquence, entre autre, une importante consommation de la bande passante. Il fallait trouver des méthodes de communication de masse plus efficaces et le multicast est une des techniques de communication qui sont en plein essor.

Les réseaux sans fil basés sur l'utilisation du protocole IEEE 802.11 fonctionnent, le plus souvent, selon un mode centralisé dans lequel une station de base joue le rôle de concentrateur ou de commutateur fournissant un point d'entrée et de sortie unique pour toutes les machines. Se libérer de cette contrainte de centralisation est possible grâce au mode Ad Hoc qui permet à deux stations mobiles de communiquer sans intervention d'une entité centrale. La mise en place de ces réseaux est totalement spontanée et autonome. Ce nouvel environnement, permet aux usagers, une libre mobilité et ne pose aucune restriction sur la localisation.

Les réseaux ad hoc sont idéaux pour les applications caractérisées par une absence d'une infrastructure préexistante, telles que les applications militaires et les autres applications de tactique comme les opérations de secours (incendies, tremblement de terre, etc.) et les missions d'exploration.

Une importante activité de recherche s'est concentrée sur les réseaux ad hoc ces dernières années. Elle a fait paraître de nombreux problèmes complexes (qualité de service, sécurité, routage, etc.). En particulier, plusieurs travaux de recherche se sont intéressés au problème de sécurité.

Le premier objectif de notre travail est de présenter une taxonomie basée sur un critère particulier qui est la topologie et de classer les articles synthétisés. Le second objectif est de proposer un protocole de gestion de clé de groupe dans les communications de groupe des réseaux ad hoc, en minimisant le nombre de mise à jour à effectuer par les leaders du groupe

et le nombre de messages à envoyé et en économisant l'énergie consommée par les stations mobiles.

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

- ◇ Le premier chapitre décrit les réseaux mobiles Ad hoc, leurs caractéristiques, les différents modes de routage existant et met en relief les contraintes de sécurité que peut avoir ce type de réseau.
- ◇ Dans le deuxième chapitre, nous mettons l'accent sur la notion de multicast et les différents services de sécurité.
- ◇ Le chapitre trois consiste en une étude synthétique des travaux de recherche qui ont été faits, et qui se font à l'heure actuelle, dans le but de résoudre le problème de sécurité dans les communications de groupe et la proposition d'une taxonomie orientée sur une topologie.
- ◇ Le chapitre quatre est dédié à la proposition d'un protocole de gestion de clé de groupe dans les communications de groupe basée sur une topologie à clique maximum.

Enfin nous terminons par une conclusion générale.

CHAPITRE 1

LES RÉSEAUX AD HOC

Introduction

Les réseaux ad hoc sont la frontière ultime en matière de communication sans fil. Cette technologie permet aux nœuds du réseau de communiquer directement avec chacun des émetteurs et récepteurs sans fil sans avoir besoin d'une infrastructure fixe. C'est une fonction très caractéristique des réseaux ad hoc, à l'égard du réseau sans fil plus traditionnel, tels que les réseaux cellulaires et sans fil LAN, dans lequel les nœuds communiquent avec les autres via l'intermédiaire des stations de base (antennes radio câblée).

Dans ce chapitre nous définissons ce qu'un réseau ad hoc, énumérons les caractéristiques de ce réseau, citons les différents domaines d'application du réseau, les méthodes de routage et les contraintes de sécurité existantes.

1.1 Définition d'un réseau mobile ad hoc

Un réseau mobile Ad Hoc, appelé généralement MANET (Mobile Ad hoc Network), c'est un réseau relativement dense dont les unités mobiles se déplacent dans un territoire quelconque. Le seul moyen de communication est l'utilisation 'des ondes radio' qui se propagent entre les différents nœuds mobiles, sans l'aide d'une infrastructure préexistante ou administration centralisée[7].

1.2 Caractéristiques d'un réseau ad hoc

Parmi les caractéristiques d'un réseau ad hoc, nous citons :

- **Conservation d'énergie**
  Les unités du réseau ad hoc sont généralement équipées de batterie, l'un des principaux objectifs de la conception est d'utiliser cette quantité limitée d'énergie le plus efficacement possible.
- **Absence d'infrastructure et variation de la topologie**
  En principe, les nœuds du réseau peuvent être placés dans une certaine région et ils

sont généralement mobiles. Par ailleurs, la topologie du réseau peut varier avec le temps, à cause de la mobilité des nœuds.

- **Faible qualité de communication**
  En général, une communication sur un canal sans fil est beaucoup moins fiable que dans un canal câblé. Par ailleurs, la qualité de communication est influencée par des facteurs environnementaux (conditions météorologiques, la présence d'obstacles), qui sont variables dans le temps. Ainsi, les applications pour les réseaux ad hoc doivent être résistantes aux variations des conditions.
- **Bande passante limitée**
  Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé (ondes radio). Ce partage fait que la bande passante réservée à un hôte soit limitée.
- **Erreur de transmission et interférence**
  Les erreurs de transmission radio sont plus fréquentes que dans les réseaux filaires (absence de l'écoute du canal), de plus, les liens radios ne sont pas isolés, deux transmissions simultanées sur une même fréquence ou, en utilisant des fréquences proches peuvent interférer.
- **Faible securité**
  Les réseaux mobiles ad hoc sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que les données transférées doivent être minimisées.

1.3 Domaines d'application

Les réseaux ad hoc sont utilisés dans toute application où le déploiement d'une infrastructure réseau filaire est trop contraignant, soit parce qu'il est difficile de le mettre en place, soit parce que la durée d'exploitation du réseau ne justifie pas de câblage à demeure, parmi celle-ci :

- Les services d'urgence : opération de recherche et de secours des personnes, opération militaire, tremblement de terre, feux, inondation, dans le but de remplacer l'infrastructure filaire ;
- Home network : partage d'applications et communications des équipements mobiles ;
- Applications commerciales : paiement électronique à distance ou l'accès mobile à l'Internet ;
- Réseaux de capteurs : applications environnementales (climat, activité de la terre, suivi des mouvements des animaux...) ou domestiques (contrôle des équipements à distance) ;
- Réseaux en mouvement : informatique embarquée et véhicules communicants ;

– Réseaux maillés : c'est une technologie émergente qui permet d'étendre la portée d'un réseau ou de le densifier.

1.4 Modélisation

Un réseau ad hoc peut être modélisé par un graphe $G_t = (V_t, E_t)$. Où : V_t représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau et E_t modélise l'ensemble des connections qui existent entre 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*. La figure 1.1 représente un réseau ad hoc de 10 unités mobiles sous forme d'un graphe.

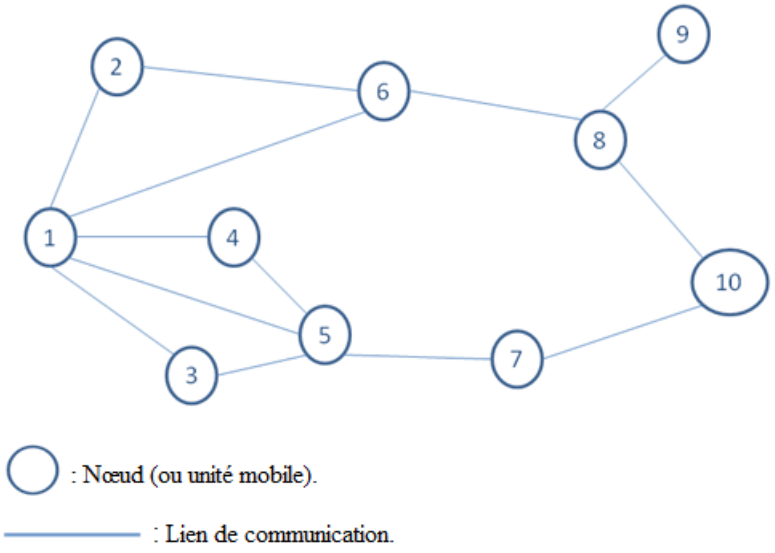

FIGURE 1.1: La modélisation d'un réseau ad hoc.

1.5 Routage pour les réseaux ad hoc

Le routage est une méthode à travers laquelle une information donnée est transitée depuis un certain émetteur vers un ou plusieurs destinataires bien précis. Le principal but de toute stratégie de routage est de mettre en œuvre une bonne gestion d'acheminement qui soit robuste et efficace. D'une manière générale, toute stratégie de routage repose sur des méthodes et des mécanismes que l'on peu regrouper en trois grandes classes : les protocoles de routage proactifs, les protocoles de routage réactifs et les protocoles de routage hybrides[42].

1.5.1 Protocoles de routage proactif

Le principe de base des protocoles proactifs est de maintenir à jour les tables de routage, de sorte que lorsqu'une application désire envoyer un paquet à un autre mobile, une route sera immédiatement connue. Dans le contexte des réseaux mobiles Ad Hoc, les nœuds peuvent apparaître ou disparaître de manière aléatoire et la topologie même du réseau peut changer ; cela signifie qu'il va falloir un échange continuel d'informations pour que chaque nœud ait une image à jour du réseau. Les tables sont donc maintenues grâce à des paquets de contrôle, et il est possible d'y trouver directement et à tout moment un chemin vers les destinations connues en fonction de divers critères. On peut par exemple privilégier les routes comportant peu de sauts, celles qui offrent la meilleure bande passante, ou encore celles où le délai est le plus faible.

Nous citons quelques protocoles de routage :

- OLSR [1](Optimized Link State Routing) est un protocole de type etat des liens.
- FSR [2](Fisheye State Routing) est un protocole de type vecteur distance.

1.5.2 Protocoles de routage réactif

Les protocoles de routage réactifs (dits aussi : protocoles de routage à la demande) représentent les protocoles les plus récents proposés dans le but d'assurer le service du routage dans les réseaux sans fil. Les protocoles de routage appartenant à cette catégorie, créent et maintiennent les routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte globale de routes est lancée, et cela dans le but d'obtenir une information spécifiée, inconnue au préalable. Plusieurs approches peuvent être appliquées dans la découverte des routes. La méthode classique de recherche de route consiste à inonder le réseau avec une requête " RREQ " (Route REQuest). Les nœuds voulant communiquer à travers le réseau lancent des requêtes à la recherche de routes permettant l'acheminement des paquets d'information.

Nous citons quelques protocoles de routage :

- AODV [3](Ad hoc On demand Distance Vector) est un protocole de type vecteur distance.
- DSR [4](Dynamic Source Routing) est un protocole de type etat de lien.

1.5.3 Protocoles de routage hybrides

Les protocoles hybrides combinent les deux idées des protocoles proactifs et réactifs. Ils utilisent un protocole proactif, pour apprendre le proche voisin (par exemple voisinage à deux ou trois sauts) ; ainsi, ils disposent des routes immédiatement dans le voisinage. Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes. Avec ce découpage, le réseau est partagé en plusieurs zones

et la recherche de routes en mode réactif peut être amélioré. A la réception d'une requête de recherche réactive, un nœud peut indiquer immédiatement si la destination est dans le voisinage ou non, et par conséquent savoir s'il faut aiguiller la requête vers les autres zones sans déranger le reste de sa zone.

On peut citer quelques protocoles de routage :

- ZRP [5](Zone Routing Protocol).
- CBRP [6](Cluster Based Routing Protocol).

1.6 Contraintes de sécurité

Parmi les problèmes liés à la sécurité des réseaux ad hoc :

1. Les dénis de services apparaissent comme les attaques les plus faciles à réaliser par un attaquant. La criticité de telles attaques dépend fortement du contexte d'utilisation mais n'est jamais complètement négligeable. Les modèles de dénis de services qui suivent se dégagent plus particulièrement dans le cas de réseau sans fil ad hoc :
   - Brouillage du canal radio pour empêcher toute communication.
   - Tentative de débordement des tables de routages des nœuds servant de relais.
   - Non coopération d'un nœud au bon fonctionnement du réseau dans le but de préserver son énergie. L'égoïsme d'un nœud est une notion propre aux réseaux ad hoc. Un réseau ad hoc s'appuie sur la collaboration sans condition de ses éléments. Un nœud refusant de jouer le jeu peut mettre en péril l'ensemble.
   - Tentative de gaspillage de l'énergie de nœuds ayant une autonomie de batterie faible ou cherchant à rester autonome (sans recharge) le plus longtemps possible. Ces nœuds se caractérisent par leur propension à passer en mode veille le plus souvent possible. L'attaque consiste à faire en sorte que le nœud soit obligé de rester en état d'activité et ainsi de lui faire consommer toute son énergie (sleep deprivation torture).
   - Dispersion et suppression du trafic en jouant sur les mécanismes de routage.
2. Les attaques passives d'écoute et d'analyse du trafic constituent une menace certaine pour la confidentialité et l'anonymat.
3. L'usurpation de l'identité d'un nœud en leurrant les mécanismes de contrôle d'accès permet de nombreuses attaques actives rendant particulièrement critiques la protection des mécanismes de routage.
4. L'attaque physique d'un élément valide d'un réseau sans fil ad hoc, entrainant la compromission du nœud, se révèle comme étant un point faible de ces réseaux.

   Enfin, il apparait clairement que les attaques sur les mécanismes de routage sont particulièrement critiques.

Conclusion

L'étude effectuée sur les réseaux mobiles ad hoc nous a permis de connaître leurs différentes caractéristiques, et ainsi constater que leur apparition a facilité la mise en œuvre d'applications mobiles ne supportant pas d'infrastructure préexistante (telles que les applications militaires), par conséquent, a dévoilé un bon nombre de problèmes telle que la sécurité de la communication.

CHAPITRE 2

SÉCURITÉ DES COMMUNICATIONS DE GROUPE

Introduction

L'évolution récente et sur tout très étendue de l'Internet a favorisé la naissance de nouvelles applications qui nécessitent des services de communication de groupes comme la télé enseignement et les conférences multimédia. Dans ces applications, un ou plusieurs membres transmettent des informations à plusieurs récepteurs. Les informations transmises ont besoin d'être sécurisées. Pour ce faire, les membres du groupe utilisent une clé pour chiffrer les messages et ne donner accès aux informations échangées qu'aux membres légitimes. La gestion de clé de groupe est une opération essentielle dans la sécurité des communications de groupe, elle permet de garantir les principaux services de sécurité tels que la confidentialité, l'authentification, l'intégrité et le contrôle d'accès.
Dans ce chapitre, nous allons introduire le concept d'IP Multicast ainsi que les principes de communication de groupe et les différents mécanismes cryptographiques utilisés pour assurer la confidentialité des données.

2.1 Principe de communication de groupe

2.1.1 Le multicast

Le multicast (multipoint) est un mécanisme de communication efficace pour des applications orientées groupe tels que la vidéo-conférence, les jeux interactifs, la télévision sur Internet, et plusieurs applications dédiées à un groupe d'utilisateurs. Contrairement à l'unicast qui permet à une machine source d'envoyer ses données vers une machine destinataire, ou au broadcast qui permet à une source de diffuser ses données vers toutes les machines du réseau, le multicast permet à une machine source de transmettre une même copie de ses données vers un groupe identifié de destinataires. La figure 2.1 illustre ces différents modes de communication.

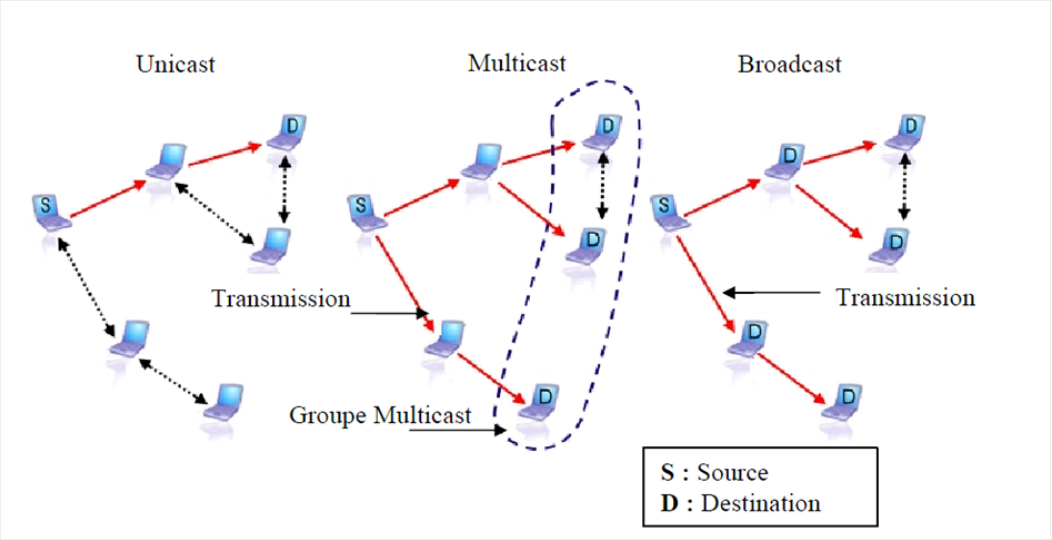

FIGURE 2.1: : Les différents modes de communication .

2.1.2 Notion du groupe multicast

Un groupe multicast est constitué d'un ensemble de machines (stations) avec une adresse IP unique appelée adresse du groupe. Un groupe multicast est très dynamique c'est-à-dire qu'une station peut joindre le groupe ou le quitter à tout moment. Il n'ya aucune restriction sur le nombre des membres d'un groupe. La direction du groupe est souvent coordonnée par une autorité unique (la source multicast). Les participants du groupe sont seulement intéressés par la réception des messages multicast qui se fait grâce à des protocoles de routage adaptés aux communications multicast, mais n'ont pas besoin de point de vue cohérent sur la composition du groupe. En cas de plusieurs-vers-plusieurs multicast, la direction du groupe peut être répartie entre plusieurs autorités.

2.1.3 IP multicast

Le modèle IP multicast, défini par DEERING [8], est une extension du modèle IP [8]. Il définit la notion de groupe, les types d'adressage et le protocole d'adhésion au groupe. Le multicast permet à une ou plusieurs sources d'envoyer des données à destination de l'ensemble des membres d'un groupe, sans effectuer d'envois multiples. Les routeurs se chargent de la réplication des paquets pour les acheminer aux différents membres. Deux composantes sont essentielles au fonctionnement du multicast :

- Un protocole de gestion de groupes se charge des requêtes d'adhésion et de départ tels qu'IGMPv3 [10] ou MLDv2 [11].
- L'architecture doit aussi disposer d'un protocole de routage permettant la construction d'un arbre de diffusion afin d'acheminer les données vers les membres du groupe.

2.1.4 Vulnérabilités d'IP multicast

Le routage multipoint permet une distribution efficace des données aux membres du groupe en utilisant un arbre couvrant la ou les sources et l'ensemble de récepteurs. L'IP multicast présente certaines vulnérabilités parmi elles, on trouve [12] :

- **IP multicast ne supporte pas la notion de groupe fermé :** L'adhésion et le départ du groupe s'effectuent sans permission préalable. N'importe quel utilisateur peut donc adhérer à un groupe et recevoir les messages destinés à ce groupe.

- **L'accès au groupe n'est pas contrôlé :** Un intrus ne faisant pas partie d'un groupe peut envoyer des données à ce dernier, perturber la session multicast et éventuellement causer des congestions dans le réseau.

- **Les données envoyées au groupe peuvent traverser plusieurs canaux non sécurisés avant d'atteindre tous les membres**, ceci augmente les éventualités d'écoute pour les intrus.

2.2 Concepts de sécurité

2.2.1 Services de sécurité

Un système ouvert vers l'extérieur est exposé à des menaces telles que l'écoute, la création, la modification et la destruction non autorisée des données.

Les services de sécurité qui permettent de pallier ces menaces sont :

- **L'authentification** : il existe deux types d'authentification : l'authentification d'entités qui permet à une entité d'être sûre de l'identité d'une seconde entité et l'authentification de l'origine de données qui permet à une entité d'être sûre qu'une deuxième entité est la source originale d'un ensemble de données.

- **Le contrôle d'accès** : permet de limiter l'accès aux utilisateurs autorisés.

- **La confidentialité** : consiste à rendre l'information intelligible à tous ceux qui pourraient l'intercepter. C'est donc la brique essentielle à la création d'une session multipoint privée.

– **L'intégrité** : consiste à vérifier que les données n'ont pas été altérées frauduleusement.

– **la non répudiation** : permet de protéger les utilisateurs contre le déni d'envoi et de réception.

2.2.2 Cryptographie

La cryptographie est un mécanisme de base permettant d'assurer la plupart des services de sécurité. Elle désigne l'ensemble des techniques permettant de transformer un texte original (texte en clair), en texte incompréhensible (texte chiffré), et d'effectuer l'opération inverse, La transformation du texte clair en texte incompréhensible s'appelle le Chiffrement l'opération inverse quant à elle s'appelle le Déchiffrement. Le chiffrement et le déchiffrement utilisent un algorithme, qui n'a généralement pas besoin d'être secret, mais qui fait intervenir une information tenue secrète dite clé (figure 2.2).

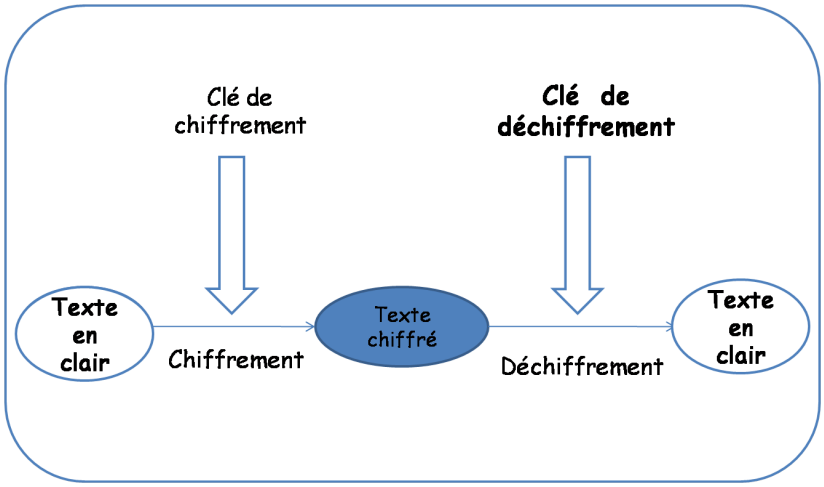

FIGURE 2.2: Chiffrement et déchiffrement.

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

2.2.2.1 Cryptographie symétrique

Dans la cryptographie symétrique, aussi appelée cryptographie à clé secrète, une seule et même clé est utilisée pour le chiffrement et le déchiffrement. Ce type de cryptographie a l'avantage d'être rapide car le nombre de clés et les calculs sont réduits. Mais le problème qui se pose est la manière d'envoyer cette unique clé à tous les utilisateurs de façon sécurisée.

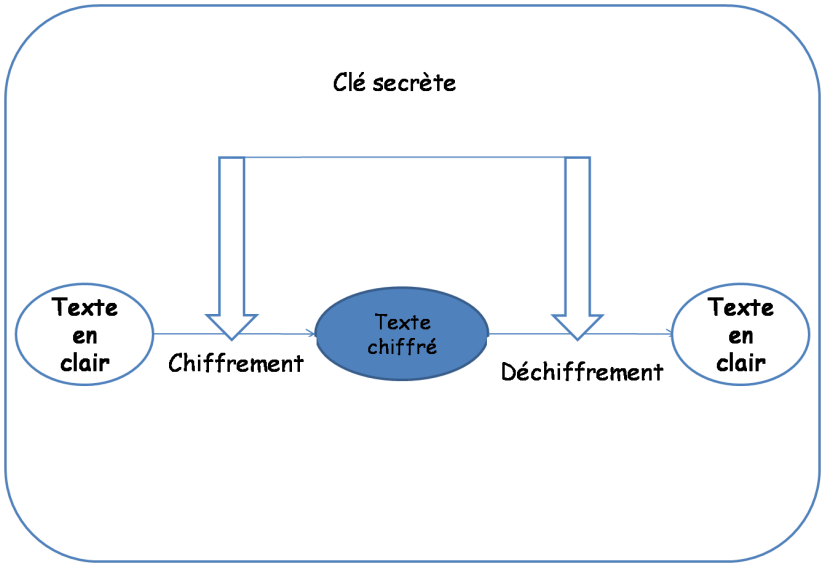

FIGURE 2.3: Cryptographie symétrique.

2.2.2.2 Cryptographie asymétrique

Dans la cryptographie asymétrique, aussi appelée cryptographie à clé publique, chaque entité détient un couple de clés : une clé visible appelée clé publique utilisée pour le chiffrement et une clé secrète appelée clé privée utilisée pour le déchiffrement (voir la figure 2.4). L'avantage de cette cryptographie est qu'elle permet à des utilisateurs n'ayant pas d'accord de sécurité préalable d'échanger des messages de manière sûre.

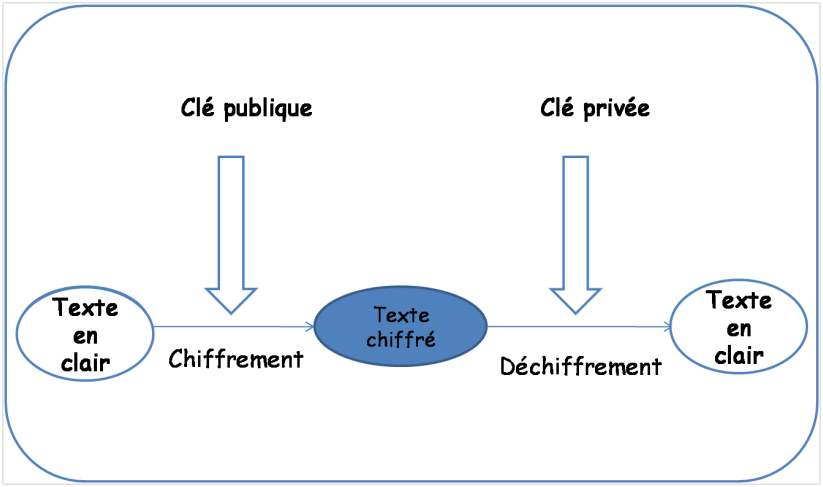

FIGURE 2.4: Cryptographie asymétrique.

2.2.3 Fonctions de hachage

On nomme fonction de hachage une fonction particulière qui, à partir d'une donnée de taille variable fournie en entrée, calcule une autre donnée de taille fixe appelée haché(condensé). Cette fonction doit être telle qu'elle associe un et un seul haché à un texte en clair c'est-à-dire que la moindre modification du document entraîne la modification de son haché. D'autre part, il doit s'agir d'une fonction à sens unique afin qu'il soit impossible de retrouver le message original à partir du condensé. Ce haché représente, alors l'empreinte digitale du document, les algorithmes de hachage les plus connus sont : MD2 [45], MD4 [44], MD5 [43], SH-1 [46] et RIPEMD-160 [47].

2.2.4 Signature numérique

La signature numérique est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur. les signatures numériques s'appuient sur la cryptographie à clé publique. Un nœud possède une clé publique qui sert à ses correspondants pour chiffrer des messages qui lui sont destinés et le nœud déchiffre les messages qu'il reçoit avec sa clé privée. Dans le cas de la signature, le nœud utilise sa clé privée pour signer un message. Le destinataire du message déchiffre la signature avec la clé publique de l'émetteur.

2.2.5 Protocole de Diffie-Hellman

IE Protocole de Diffie-Hellman [13], propose à deux entité A et B de pouvoir définir une clé secrète même si une autre entité E écoute leur communication. Les étapes de la procédure de ce protocole sont les suivantes :

- Les deux participants choisissent, ensemble et 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.
- Chaque participant génère confidentiellement un nombre aléatoire plus petit que *p-1* qui constitue sa clé privée : *A* génère x_1 et *B* génère x_2.
- Chaque participant calcule alors une clé publique : A calcule $y_1 = g^{x_1} \bmod p$ et *B* calcule $y_2 = g^{x_2} \bmod p$.
- Les deux participants s'échangent leurs clés publiques. Chacun des deux participants peut maintenant calculer $z = (y_1)^{x_2} \bmod p = (y_2)^{x_1} \bmod p$ qui constituera le secret que les deux participants peuvent utiliser comme clé secrète symétrique.

2.3 Gestion de clé dans les communications de groupe

La confidentialité des communications de groupe conduit à ce qu'uniquement les membres valides peuvent avoir accès au contenu destiné au groupe. Généralement, une clé symétrique est utilisée pour chiffrer les donnés par la source du trafic et les déchiffrer au niveau des récepteurs. Cette clé est appelée Clé de Chiffrement de Trafic " TEK " [14].

2.3.1 Clé du groupe

La TEK est secrète et doit être connue uniquement par les membres appartenant au groupe. La clé TEK est établie entre les membres du groupe soit par distribution, soit par accord [12].

- **Distribution :** dans ce cas, une entité du groupe (ou une tierce partie de confiance) génère une clé secrète et la distribue de manière sécurisée à l'ensemble des membres du groupe.
- **Accord** (ou contribution) dans ce cas, chaque membre du groupe participe à la génération de la clé de groupe. Dans cette situation chaque participant génère un élément de la clé. L'ensemble des éléments générés va servir pour la génération de la clé.

2.3.2 Propriétés d'une clé de groupe

La clé du groupe doit impérativement assurer une propriété principale qui est la confidentialité, les personnes qui n'ont jamais fait partie du groupe ne devront avoir accès à aucune clé qui permet de déchiffrer un flux multicast adressé au groupe ; cette propriété est basée essentiellement sur les deux impératifs suivantes [15].

- **Confidentialité Future :** les utilisateurs qui ont quitté le groupe ne devront plus avoir accès à aucune clé future du groupe. Cela assure qu'un membre ne pourra pas déchiffrer les données après son départ du groupe ;
- **Confidentialité Passée :** Un nouveau membre qui vient de joindre le groupe ne devra pas avoir accès à aucune ancienne clé. Cela assure qu'un membre ne peut déchiffrer les données émises avant son adhésion.

2.3.3 Facteur d'échelle

Le facteur d'échelle traduit la capacité du système à s'adapter aux variations de la taille du groupe et à prendre en charge les événements qui modifient cette taille (ajout, départ, partitionnement). Il peut être exprimé par deux paramètres [12] :

- **Facteur d'échelle 1-affecte-n :** mesure l'effet d'une adhésion ou départ d'un membre du groupe sur les autres membres du groupe, (n étant le nombre de membres dans le groupe).

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

Conclusion

Le déploiement des communications de groupe sécurisées dans les MANETS induit également de nouvelles épreuves à prendre en compte. En effet, aux contraintes de sécurité, s'ajoute la vulnérabilité d'IP multicast, qui de part sa nature élimine toute possibilité d'identification des membres du groupe ou de confidentialité des données.
Dans le chapitre suivant, nous présenterons un état de l'art des protocoles de sécurisation des communications de groupe dans les réseaux MANETS, et nous les discuterons selon des métriques bien spécifiques.

CHAPITRE 3

PROTOCOLES DE GESTION DE CLÉ DE GROUPE DANS LES MANETS

introduction

La partie la plus essentielle pour toute architecture de sécurisation des communications de groupe est la gestion de clé de groupe, elle a pour objectif de protéger les applications multicast, elle permet d'établir, de distribuer et de renouvler la clé de groupe. Aujourd'hui de nombreuses approches de gestion de clé de groupe dans les réseaux ad hoc sont proposées.

Dans ce chapitre, nous presentons une classification des protocoles de gestion de clé de groupe dans les Manets. Le critere de classification est le type de topologie du réseau. Puis, nous décrivons quelques uns de ces protocoles et dans la dernière partie de ce chapitre, nous exposons une étude comparative de certaines de leurs caractéristiques.

3.1 Protocoles de gestion de clé de groupe dans les MANETs

Les protocoles de gestion de clé de groupe peuvent être classés selon trois approches [11] :

Approche centralisée : Nous retrouvons dans cette approche les protocoles dont la génération et la distribution de la clé du trafic se fait par une seule entité centrale.

Approche décentralisée : Dans cette approche le groupe multicast est divisé en sous groupe. Où chaque sous groupe est géré séparément par un contrôleur local, responsable de la sécurité des membres de son sous-groupe et de la gestion d'une clé de chiffrement TEK locale au sein de son sous groupe. Chaque contrôleur local génère et distribue la TEK locale à ces membres locaux.

Approche distribuée : Dans cette approche la gestion de la clé du groupe est à la charge de tous les membres du groupe multicast qui collaborent et coopèrent pour partager une clé de groupe.

Les auteurs dans [], proposent de classifier les protocoles de gestion de clé de groupe se-

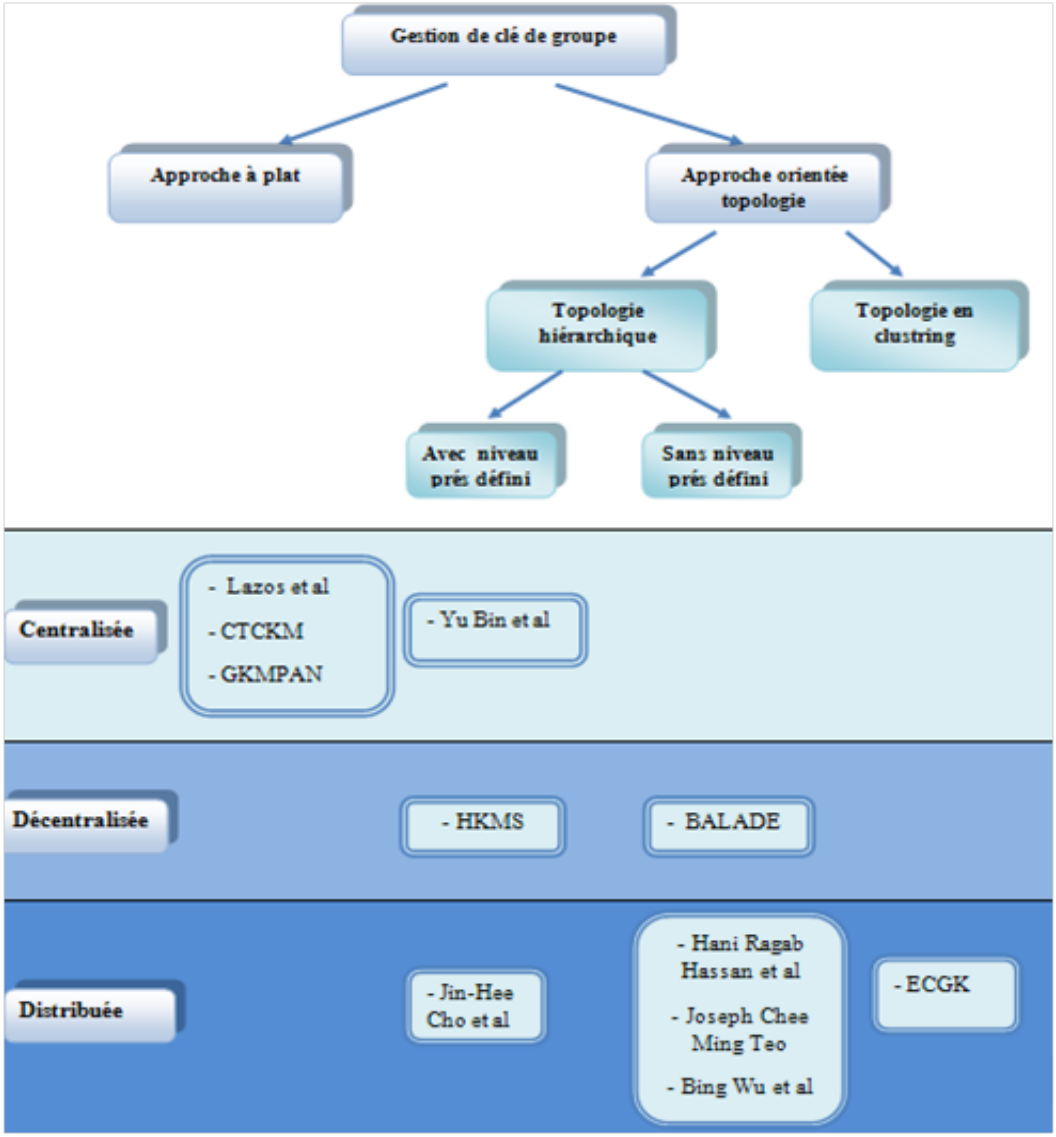

FIGURE 3.1: Taxonomie des protocoles de gestion de clé de groupe dans les MANETs.

lon deux approches, à plat et orientée topologie. Dans notre travail, nous proposons d'affiner cette classification comme illustré sur la figure 3.1.

Nous proposons de classer les protocoles de gestion de clé de groupe selon deux types d'approches : l'approche à plat et l'approche orientée topologie, Nous présentons dans ce qui suit chacunes de ces approches et nous discutons de leurs forces et leurs faiblesses.(figure 3.1)

3.1.1 Approche à plat

Dans cette approche, il n'existe aucune organisation préalable du groupe multicast et tous les membres du groupe partagent une TEK commune. La gestion de cette clé unique

est centralisée sur un serveur unique ou répartie entre tous les membres du groupe. Les protocoles Lazos et al [15], Mu Haibing Liu et al [16], et GKMPAN [17] illustrent ce type d'approche.

1. Protocole de Lazos et al

Lazos et al [15] proposent un protocole de gestion de clé centralisé qui prend en compte la contrainte de l'énergie limitée dans les réseaux ad hoc. Ce protocole améliore le schéma de distribution de clé LKH [18] et l'adapte au contexte des réseaux ad hoc statiques, en optimisant la consommation de l'énergie, avec l'utilisation de l'information de localisation géographique des membres (GPS). L'idée de base est que les membres géographiquement proches peuvent être atteint par un message de diffusion, ou utilisent le même chemin pour accéder au flux multicast. L'algorithme de clustérisations K-means [15] est utilisé pour former des groupes de forte corrélation et pour déduire l'arbre de distribution de la clé de groupe. Le processus de distribution de clés, basé sur l'algorithme K-means est composé de plusieurs étapes :

- **Etape 1** : Tous les membres du groupe sont affectés à un seul cluster.
- **Etape 2** : Chaque cluster est divisé en deux sous clusters via l'algorithme K-means.
- **Etape 3** : Une procédure de raffinement est utilisée pour équilibrer le nombre de membre par cluster.
- **Etape 4** : Répéter les étapes 2 et 3 jusqu'à l'aboutissement de clusters formés d'un seul ou deux membres. Les clusters formés d'un seul membre sont fusionnés, si cela est possible.
- **Etape 5** : Faire correspondre la hiérarchie de clusters en une hiérarchie logique de distribution de clés LKH.

La figure 3.2 illustre le résultat de l'exécution de l'algorithme K-means, les membres géographiquement proches sont frères dans l'arbre de distribution de clés LKH.

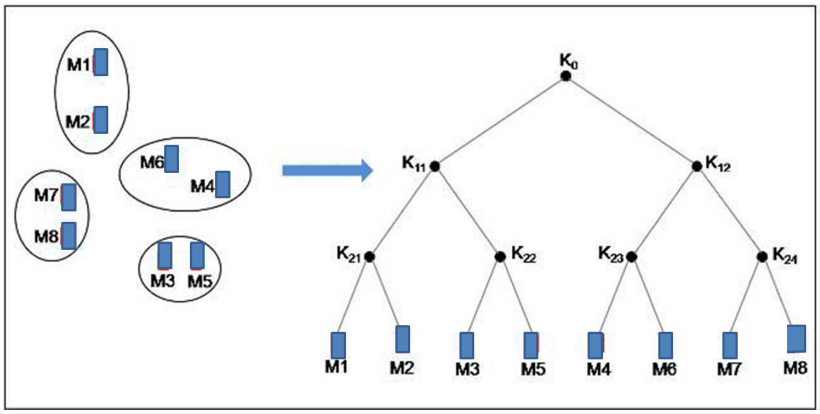

FIGURE 3.2: Processus de distribution de clés basé sur l'algorithme K-means.

2. Protocole de Mu Haibing Liu

Mu Haibing Liu et al [16] proposent le protocole CTCKM "composite tree-cluster key management" basé sur le protocole LKT[19]. Les utilisateurs de chaque groupe sont dans un réseau à topologie à plat, et la politique de gestion de la clé locale est centralisée.

• Initialisation

Durant cette phase, les utilisateurs rejoignent et deviennent les membres du groupe multicast. Chaque utilisateur diffuse un paquet de détection, comportant son poids P_i, et son identificateur ID. L'utilisateur qui reçoit ce message compare son propre P_i avec celui des autres utilisateurs et détermine son rôle dans le groupe. Les cluster-heads négocient et maintiennent la clé de groupe dans un mode en arbre. Pour les communications multicast une fonction de hachage ainsi que l'algorithme DH [12] sont utilisés. Un utilisateur ordinaire dans chaque cluster stocke une clé de groupe, une clé de cluster, ainsi que ses clés asymétriques, tandis qu'un cluster-head conserve certaines clés internes dans l'arborescence à clé logique (LKT), et une liste de clés publiques de ses membres, outre la clé de groupe et la clé de son cluster.

• Gestion des membres de groupe

✓ Adhésion d'un nœud ordinaire

Quand un nœud souhaite adhérer au groupe multicast, il envoie une requéte "add-req", qui comprend les informations des utilisateurs (certificat et clé publique). Le membre ordinaire qui reçoit ce message le transmet à son cluster-head. Si le cluster-head approuve la demande, il envoie un message d'approbation au nouveau membre et commence à mettre à jour la clé de groupe. Il génère une nouvelle clé de cluster K_c' et négocie une nouvelle clé de groupe K_g' avec les autres cluster-heads. Il diffuse K_c' et K_g' aux membres de son cluster chiffrés avec l'ancienne clé de cluster et envoie au nouveau membre K_c' et K_g' chiffrés avec sa clé publique. Les cluster-heads peuvent également diffuser la nouvelle clé de groupe K_g' à leurs membres chiffrée avec les clés de cluster respectives.

✓ Adhésion d'un cluster-head

Cela arrive lors de la jointure de multiple clusters ou de leurs partitionnement. Tous les cluster-heads, y compris le nouveau, négocient une nouvelle clé de groupe et la diffuse aux membres de leurs clusters et modifient l'architecture LKT.

✓ Départ d'un membre ordinaire

Lorsqu'un nœud ordinaire quitte le groupe, les clés internes sont encore un secret pour lui. Les clusters heads peuvent négocier une nouvelle clé de groupe avec les clés internes du deuxième niveau la nouvelle clé est calculée comme suit : $K_g' = h(K_{21} \bigoplus K_{22} \bigoplus i)$, ou i est un nombre aléartoire. La nouvelle clé de groupe est envoyée par chaque cluster-head à ses membres cryptée avec leurs clés publiques respectives.

✓ **Départ d'un cluster-head**

Le cluster-head diffuse un message de départ pour informer tous ses membres et les autres cluster-heads. Pour cela :

- Il peut choisir un autre nœud pour gérer le cluster et lui passer sa chaîne de clé de cluster, de sorte que le nouveau cluster-head puisse négocier une nouvelle clé de groupe avec les autres.
- Le cluster-head sortant peut également s'associer à un autre cluster-head afin d'unir leurs membres dans un même cluster si le nombre de leurs membres est de petite taille.
- Il laisse le problème de sélection de cluster-head à ses membres de cluster qui vont relancer le processus d'initialisation. Si le nombre des membres du cluster est petit, ils peuvent être transférés à d'autres clusters ce qui engendre la disparition du cluster.

✓ **Transfert**

Dans ce cas, il n'est pas nécessaire de mettre à jour la clé de groupe parce que le nœud est toujours un membre valide du groupe, mais la clé du cluster auquel ce membre appartient ou vient de se joindre à besoin d'être mise à jour.

3. Le protocole de Sencun Zhu et al

Sencun Zhu et al[17] proposent le protocole GKMPAN "an efficient and scalable group rekeying protocol for secure multicast in ad hoc networks" qui est basé sur une phase de pré-distribution de listes de clés aux membres du groupe multicast et sur de nombreuses phases de renouvellement de clés, à la charge d'un serveur de clés.

• **Phase de pré-distribution de clés**

Dans cette phase, chaque membre du groupe u obtient avant le déploiement du réseau ad hoc les clés suivantes :

-Un ensemble R_u, qui contient m clés parmi l, l étant le nombre total de clé à pré-distribuer à tous les membres du groupe K_1 ; K_2 ;... ; K_i. l_u est un ensemble d'identificateurs de clés correspondant à l'ensemble R_u. Les clés de R_u sont utilisées comme clés de chiffrement de clés ($KEKs$). L'algorithme de pré-distribution de clés permet à chaque nœud i qui connaît l'identité d'un autre nœud j, de déterminer l'ensemble des identificateurs de clés l_j et ainsi de déterminer quelle(s) clé(s) utilisée pour pouvoir communiquer avec lui de façon sécurisée.

-Une clé de groupe K_g générée par le serveur de clés, qui est utilisée pour sécuriser la communication entre les membres du groupe. Une clé secrète est partagée entre le serveur de clés et chaque membre individuellement.

-L'authentification des données de la source est assurée par le protocole TESLA [20]. Pour cela, la première clé de chaînage TESLA (clé de validation) est chargée dans chaque nœud.

-Des membres peuvent joindre le groupe dans GKMPAN, même après la phase de pré-distribution de clés. Le serveur de clés peut ajouter des entités dans le groupe pour compenser

les membres exclus. Avant d'ajouter un membre u dans le groupe, le serveur de clés déploie son ensemble R_u ainsi que la clé du groupe courante. Suite à cet évènement le serveur de clés peut décider de renouveler la clé du groupe K_g pour assurer la confidentialité passée et diffuser un message de renouvellement de clé du groupe $K'_g = fK_g(0)$, f est une fonction pseudo-aléatoire.

• Renouvellement de clés

La phase de renouvellement de clés comprend deux étapes ;

✓ Distribution sécurisée de la clé de groupe

Le serveur de clés génère et distribue la clé du groupe. Ce processus de distribution de clés est réalisé, saut par saut, en chiffrant la clé du groupe par les clés pré-déployées $KEKs$. Le serveur de clés ne délivre la clé qu'à ses voisins immédiats (à un seul saut), qui acheminent la clé à leurs voisins de façon récursive et sécurisée. De cette manière, GKMPAN exploite la propriété de communication multi-sauts dans les réseaux ad hoc, où les membres du groupe jouent le rôle d'hôte et de routeur.

✓ Exclusion d'un membre du groupe et mise à jour des clés compromises

Lors de la révocation d'un membre malicieux u, toutes les clés incluses dans R_u du membre exclu sont compromises et doivent donc être renouvelées par les autres membres détenant ces clés. Le serveur de clé détermine M, l'identificateur de la clé non compromise, qui est la plus connue parmi les membres restants. Puis, le serveur de clé génère une clé intermédiaire $K_{im} = f_{k_M}(K_g)$ et calcule la nouvelle clé de groupe $K'_g = f_{k_{im}}(0)$. Il diffuse un message de révocation à tous les membres du groupe contenant M, l'identifiant du membre exclu (u) et la nouvelle clé $f_{K'_g}(0)$. Ce message sera authentifié par le protocole TESLA, qui permet aussi d'assurer la tolérance aux pertes de messages.

- Les nœuds qui possèdent la clé K_M peuvent calculer la clé intermédiaire K_{im} après avoir reçu le message de révocation, ils transmettent la clé K_{im} à tous leurs voisins qui ne possèdent pas cette clé intermédiaire via un chemin logique sécurisé par des clés non compromises ($P - R_u$) pour que u ne puisse pas deviner la clé K_{im}.
- Tous les nœuds (sauf u) calculent la nouvelle clé $K'_g = f_{k_{im}}(0)$ et la vérifient avec celle contenue dans le message de révocation.
- Après avoir calculé la nouvelle clé de groupe par tous les nœuds non compromis, les membres du groupe doivent renouveler leurs clés et toutes les clés $KEKs$ connues du membre exclu doivent également être renouvelées. A la fin, tous les nœuds effacent la clé K_{im} et les clés K_i initiales.

5. Tableau comparatif

Critères	Lazos et al	CTCKM	GKMPAN
Facteur d'échelle 1 affect n	Non	Non	Non
Service de sécurité	Confidentialité des nœuds	Confidentialité des nœuds	-Révocation des nœuds Confidentialité des données
Surcoût de calcul	Algorithme de cautérisation K-means en O (ns)	- Algorithme d'initialisation -calcul des clés -algorithme DH	Déchiffrement et rechiffrement de la TEK (saut par saut) Algorithme de reconnaissance des clés communes entre les membres
Surcoût de stockage	Clés de l'arbre LKH Contrôleur O(N) Membre O (log N)	-clés publiques des membres -la clé de groupe, la clé de cluster, des clés internes dans l'arborescence à clé logique	m clés pré distribuées, choisis parmi l ; Clés d'authentification TESLA ; Clé secrète entre le serveur et tout membre
Surcoût de communications	Initialisation du groupe (N message de LKH)	-diffusion des paquets de détection, les requêtes ADD	Acheminement de la clé du groupe saut par saut, dépendant de m et l.
Vulnérabilité	Source dédiée aux réseaux ad hoc statiques	Le cluster head	Serveur de Clés

Table 3.1-Tableau comparatif des protocoles de l'approche à plat

Critique

L'approche à plat souffre du problème 1-affecte-n, où un unique changement d'appartenance à un groupe (adésion ou départ) se traduit par un processus de renouvelement qui perturbe tous les membres du groupe. En outre, la plupart des protocoles dans cette approche ont besoin d'un serveur central. Donc, ils ne sont ni évolutifs, ni tolérants aux pannes.

3.1.2 Approche orientée topologie

Cette approche propose d'organiser les membres du groupe selon une topologie virtuelle. Les régimes les plus courants sont le regroupement et les arbres hiérarchiques, mais d'autres structures de groupe sont également proposées. L'objectif principal de l'utilisation d'une topologie virtuelle est de diminuer le coût du processus de renouvellement des clés afin de faire face au problème 1-affecte-n.

Dans cette approche nous classerons les protocoles selon deux types de topologies, la premiére étant la topologie hierarchique, la seconde est la topologie en clustring. La topologie hiérarchique est partagée en deux catégories : la premiére catégorie est une topologie hiéarachique à niveau où nous présenterons les protocoles : Nen-Chung et al [21], Jin-Hee Cho et al[22] et Yu Bin et al[23], la seconde catégorie est une topologie hiérarchique sans niveau prédéfini dans laquel nous présenterons les protocoles : Mohamed-Salah Bouassida et al [24], Hani Ragab Hassan et al [14], Joseph Chee Ming Teo et al [25] et Bing Wu [26], enfin dans la topologie en clustring nous présenterons le protocole de Kaouther Drira et al [27].

3.1.2.1 Topologie hiérarchique

Une topologie hierarchique est une topologie divisée en niveaux. Le sommet de haut niveau est connecté à plusieurs nœuds de niveau inférieur, dans la hiérarchie. Ces nœuds peuvent être eux mêmes connectés à plusieurs nœuds de niveau inférieur, cette architecture forme un arbre.

3.1.2.1.1 Topologie hiérarchique à niveau

Une topologie hiérarchique à niveau est une topologie hierarchique avec un nombre de niveau fixé à l'avance.

1. Le protocole de Nen-Chung et al

Nen-Chung et al [21] proposent le protocole HKMS "The Hierarchical Key Management Scheme" basé sur une architecture à deux niveaux (*L1* et *L2*). Les sous groupes contiennent des leaders dans chacun des deux niveaux, qui sont élus en fonction de leur poids. Ainsi

nous obtienderons, un *L1*-sous groupe géré par un leader appelé *L1-head* et un *L2*-sous groupe géré par *L2-head*. La figue 3.3 illustre cette architecture.

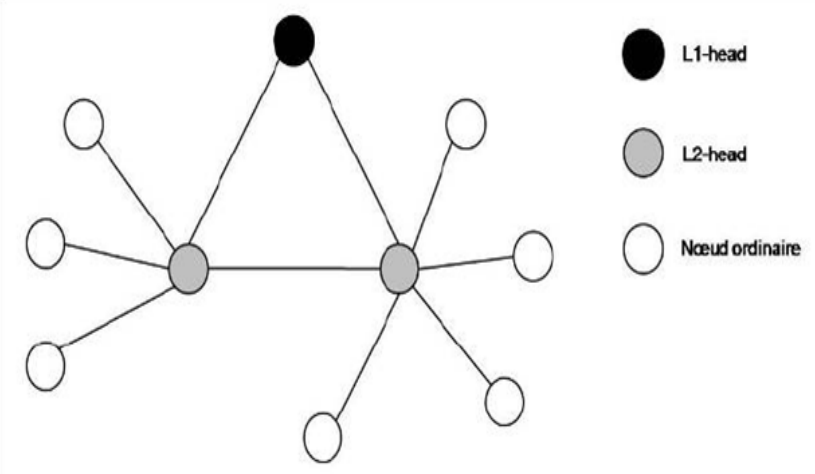

FIGURE 3.3: La structure d'un sous groupe selon HKMS.

• **Procédure 1 : Sélection du L1-head**

- **Etape 1 :** Chaque nœud diffuse un message *"hello"* contenant la valeur de son poids à tout ses voisins à deux sauts au maximum.

- **Etape 2 :** Le nœud avec le poids le plus élevé est sélectionné pour être le *L1-head*.

- **Etape 3 :** Les autres nœuds s'enregistrent auprès du *L1-head* sélectionné et lui envoient toutes les informations les concernant.

• **Procédure 2 : Sélection du L2-Head**

- **Etape 1 :** Tous les nœuds envoient leurs données de localisation au *L1-head*.

- **Etape 2 :** Après avoir reçu toutes les informations sur les nœuds, le *L1-head* classifie tous les nœuds en *L2*-sous groupe.

- **Etape 3 :** Les nœuds comparent leur poids et sélectionnent celui qui a le poids le plus élevé pour être le *L2-head*, le *L2-head* gère le sous groupe et communique avec le *L1-head* et les autres *L2-heads*.

• **Gestion de clé**

Après la génération de l'arbre, tous les nœuds envoient un paquet d'enregistrement au *L1-head* qui génère une clé *L1GK* en utilisant RSA [28], et la transmet à tous les *L2-heads* membres de son *L1*-sous groupe. Ensuite, chaque *L2*-sous groupe génère sa clé *L2GK* en fonction de *L1GK* et la transmet à tous ses membres. Pour la communication entre les sous groupes, le protocole de Diffie-Hellman [12] est utilisé pour sécuriser la transmission. Tout d'abord les sous groupes voisins sont reliés en utilisant les nœuds de transmission dans chaque sous groupe. La clé de transmission K_c est utilisée pour le chiffrement et le déchiffrement des messages entre deux nœuds dans différents sous groupes. Un nœud vou-

lant communiquer envoie l'information au *L1-head* afin qu'il génère les clés de communication K_c qui seront utilisées pour relier les sous groupes. Une fois que la localisation du nœud de destination est connue par le *L2-head*, le protocole de *DH* sera utilisé pour générer une clé privée K_{DH} qui sera connue uniquement par le nœud source et le nœud de destination. Cette clé K_{DH} sera utilisée pour le premier chiffrement, quand le paquet sera transmis (pour que le paquet ne soit pas intercepté par un autre nœud). la figuere 3.4 illustre le processus de communication selon HKMS.

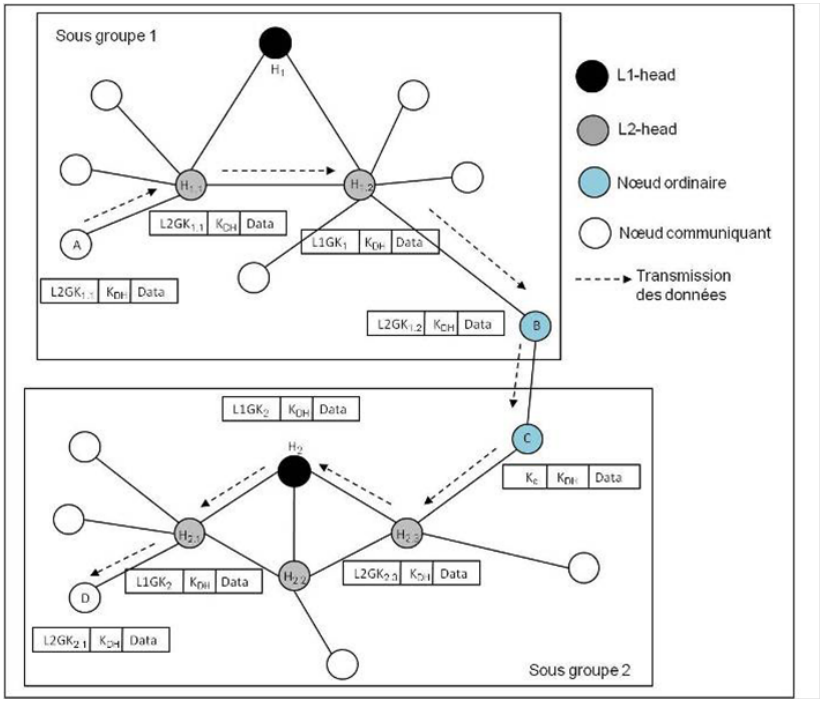

FIGURE 3.4: Processus de communication selon HKMS.

• Adhésion d'un nœud au sous groupe

- **Etape 1 :** Le nœud envoie un message de demande d'adhésion aux nœuds voisins.
- **Etape 2 :** Après avoir reçu le message de demande d'adhésion, le nœud voisin le transmet à son *L2-head*.
- **Etape 3 :** *L2-head* envoie un message de réponse au nouveau nœud.
- **Etape 4 :** Le nouveau nœud reçoit le message de réponse, et il est autorisé à rejoindre le *L2*-sous groupe.
- **Etape 5 :** *L2-head* régénére une clé *L2*-sous groupe (*L2GK*) par la mise à jour des informations *L2*-sous groupe et l'envoie à tous les nœuds *L2*-sous groupes.

• Départ d'un membre ordinaire

- **Etape 1 :** Avant de quitter le sous groupe, le nœud envoie une demande de départ à son *L2-head*.

- **Etape 2 :** Après avoir reçu la demande de départ, *L2-head* envoie une réponse au membre sortant, puis il génère une nouvelle clé *L2GK* et la transmet à tous les membres restant.

• Départ d'un L2-head

- **Etape 1 :** Avant de quitter le sous groupe, *L2-head* envoie une demande de départ aux nœuds ordinaires et à son *L1-head*. Après avoir reçu cette demande, les nœuds ordinaires et le *L1-head* envoient un message de réponse au *L2-head*. S'il y a des nœuds qui ne répondent pas, une autre demande leur sera envoyée jusqu'a ce qu'ils répondent.
- **Etape 2 :** un nouveau *L2-head* sera sélectionné parmis les membres ordinaires dans le *L2*-sous groupe du nœud partant, en fonction de son poids.
- **Etape 3 :** Le nouveau *L2-head* envoie les informations de mise à jour du *L2*-sous groupe au *L1-head*.
- **Etape 4 :** Après avoir reçu l'information de mise à jour du nouveau *L2-head*, *L1-head* régénére une nouvelle clé *L1GK* pour le *L1*-sous groupe et l'envoie à tous les *L2-heads*.
- **Etape 5 :** *L2-head* génère une nouvelle clé *L2GK* pour *L2*-sous groupe et la transmet à tous ses membres.

• Départ d'un L1-head

- **Etape 1 :** Avant de quitter le sous groupe, le *L1-head* envoie une demande de départ aux *L2-heads*. Après avoir reçu cette demande, chaque *L2-head* envoie une réponse au *L1-head*.
- **Etape 2 :** Un nouveau *L1-head* est élu parmi les *L2-head* en fonction du poids.
- **Etape 3 :** Simultanément, les nœuds ordinaires de *L2*-sous groupe précédemment gérés par le nouveau *L1-head* choisissent de la même façon un nouveau *L2-head*.
- **Etape 4 :** Tous les *L2-heads* doivent envoyer leurs informations concernant leurs *L2*-sous groupe au nouveau *L1-head* et seront ainsi enregistrés.
- **Etape 5 :** Après avoir reçu les informations des *L2-heads*, le *L1-head* régénére une nouvelle clé de sous groupe (*L1GK*) et l'envoie à tous les *L2-heads*.
- **Etape 6 :** Après avoir reçu la nouvelle clé de sous groupe (*L1GK*), chaque *L2-head* génère pour son *L2*-sous groupe une nouvelle clé *L2GK* et la transmet à tous ses membres.

2. Protocole de Jin-Hee Cho et al

Jin-Hee Cho et al[22] proposent un protocole "Performance optimization of region-based group key management in mobile ad hoc networks", le groupe est reparti en sous-groupes dans une hiérarchie à deux niveaux, le schéma de gestion de clé utilisé est une hybridation entre le décentralisé et le distribué.

• Gestion des clés

Initialement dans une région, un nœud peut prendre le rôle d'un "*leader*" pour exécuter *GDH[éç]*. S'il y a plusieurs initiateurs, le nœud qui a le plus petit id gagne en tant que leader, il exécutera *GDH* et générera une clé régionale K_r. Une fois qu'un leader est élu dans chaque

région, tous les leaders dans les groupes exécuteront *GDH* et se mettront d'accord sur une clé secrète de leader K_{RL}, pour des communications sécurisées entre les dirigeants.
Un "*leader*" peut prendre le rôle d'un "*super-leader*" ou "*coordinateur*" pour exécuter *GDH* parmi les autres leaders. S'il y a de multiples leaders qui lancent l'exécution de *GDH*, le leader qui a le plus petit *id* gagne en tant que coordinateur pour exécuter *GDH* et générer K_{RL}. Une fois que K_{RL} est générée, une clé de groupe K_g est obtenue comme suit :

$$K_g = MAC\ (K_{RL}, c).$$

Où *MAC* est une fonction de hachage cryptographique sécurisée, K_{RL} est la clé utilisée comme la clé secrète leader pour *MAC*, et c est un compteur qui est incrémenté à chaque fois qu'un événement se produit. Une fois que K_g est générée, les dirigeants diffusent la clé K_g de groupe aux membres. K_g est utilisée pour les communications de données sécurisées entre les membres du groupe à travers les régions.

• Adhésion d'un nœud

Tous les nœuds sont équipés d'un GPS qui leur permet de localiser dans qu'elle région ils se trouvent et de savoir s'ils ont franchi les frontières. Une opération d'adhésion est lancée par un nœud communiquant avec ses nœuds voisins pour connaître le leader de la région dans un groupe. Quand un nouveau membre rejoint le groupe, un message "*hello*" qui comprend, son identifiant et des informations sur sa localisation est envoyé à ses nœuds voisins pour les informer de son intention de rejoindre le groupe. Les nœuds voisins reçoivent le message "*hello*" et le transmettent à leur leader régional. Ce dernier authentifie l'identité du nœud avec sa clé publique, puis il agit comme un coordinateur impliquant tous les membres du sous groupe, incluant le nœud qui veut rejoindre le groupe pour exécuter *GDH* et générer une nouvelle clé K_r. Ensuite, le leader met à jour la liste des membres régionaux, et diffuse la liste régionale à tous les membres de la région.

• Départ d'un membre ordinaire

Quand un membre ordinaire quitte le groupe, il informe son leader régional. Lorsque ce dernier reçoit le message du nœud désirant quitter le groupe, il met à jour la vue régionale et la diffuse à ses membres. Après cet événement de départ, une nouvelle clé K_r est générée par l'exécution de *GDH* et distribuée aux membres régionaux. Ensuite, tous les leaders de groupe sont informés de la modification de la vue du groupe et informe tous leurs membres, et régénèrent une clé de groupe, ils la distribuent à leurs membres correspondants en la cryptant avec leur clé respective K_r.

• Départ d'un leader

Quand un leader quitte le groupe, la clé du leader devrait être changée. Ainsi, un nouveau leader est élu pour remplacer le leader partant. Comme il s'agit d'un changement de membre 'leader', tous les leaders, y compris le leader nouvellement élu, exécuteront *GDH* pour

générer une nouvelle clé de leader. Ensuite, chaque leader génère de manière autonome une nouvelle clé de groupe et la distribue à ses membres à l'aide de la clé régionale K_r.

• Franchissement de limite par un membre non leader
Si un membre non leader traverse une frontière régionale, un changement d'appartenance régionale se produit, les clés régionales dans les deux régions concernées sont renouvelées en se basant sur *GDH*, et la vue régionale des membres dans ces deux régions est mise à jour.

•Franchissement de limite par un membre leader
Si un membre leader traverse une frontière régionale, il y'aura un changement de leader, en plus de toutes les opérations considérées dans le cas de la frontière traversée par un membre non leader. Un nouveau leader dans la région de départ est élu, la clé du leader est renouvelée parmi tous les leaders et la vue est mise à jour.

•Déconnexion et reconnexion d'un membre du groupe
Les membres peuvent être déconnectés volontairement ou involontairement pour détecter le membre qui est en panne dans le groupe ; chaque hôte mobile envoie périodiquement un message "*I-am-alive*" à son leader de sorte que celui là soit au courant des membres qui sont dans sa région. Si un leader ne reçoit pas le message pendant une certaine période de temps d'un membre, il considère le membre déconnecté. Les déconnexions sont traitées comme des événements de départ dans ce protocole. Si le membre déconnecté est un leader, un nouveau leader est élu par la suite avec un protocole d'élection d'un nouveau leader.

•Élection d'un leader
Lors d'un départ ou d'une déconnexion d'un leader, une élection d'un nouveau leader se déclenche dans la région concernée. Un membre de la région après avoir raté un message de balisage de son leader régional peut initier l'exécution de *GDH* en fonction de la vue régionale. Le nouveau leader annonce alors qu'il est le nouveau leader dans la région en diffusant un message "*I-am-a-new-leader*" chiffré avec la clé K_R.

3. Protocole de Yu Bin et al

Yu Bin et al [23] proposent le protocole "the Three-layered Group Key Management Architecture for MANET", basé sur une architecture à trois niveaux ; les nœuds du premier niveau sont les nœuds passerelles, ceux du deuxième niveau sont les nœuds passerelles du second niveau ou sous passerelles et les nœuds du troisième niveau sont appelés cellules. La figure 3.2 illustre un exemple d'un MANET avec une architecture à trois niveaux.

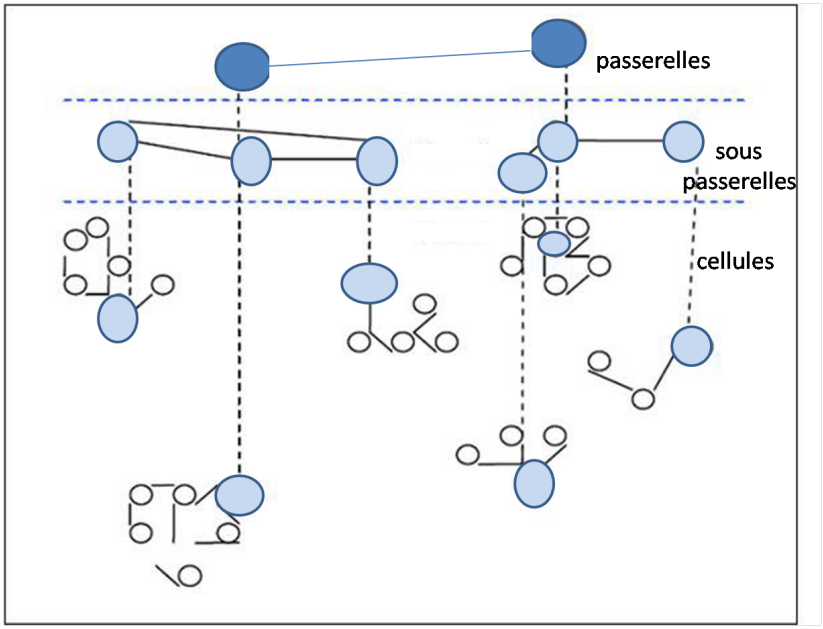

FIGURE 3.5: Un réseau MANET avec infrastructure virtuelle à trois niveaux.

Dans ce protocole, deux architectures sont proposées :

• Architecture : deux -Distribuées -une Centralisée

Dans cette architecture, la clé de groupe est générée à partir de la couche inférieure vers la couche supérieure. Tout d'abord, chaque sous passerelle génère et distribue une clé de sous groupe pour les cellules placées sous son contrôle. Puis, cette sous-passerelle contribuera au partage sécurisé de cette clé de sous groupe, en utilisant l'algorithme distribué de gestion de clé pour générer la clé de sous groupe pour la sous passerelle sous un nœud passerelle. Le nœud passerelle utilisera la clé du sous groupe comme un partage sécurisé pour calculer la clé de groupe des MANETs. Le premier niveau et le second niveau de cette architecture utilisent l'algorithme distribué de la gestion de clé, et les clés de groupe du troisième niveau sont contrôlées par un schéma de gestion de clé centralisé.

•Architecture : deux -Centralisées -une Distribuée

Dans le premier niveau, chaque nœud génère et distribue une clé de sous groupe pour les sous passerelles sous le controle d'un schéma de gestion de clé centralisé. Dans cette architecture, le premier niveau utilise un schéma de gestion de clé distribué, alors que le deusième et le troisième niveaux utilisent une approche de gestion de clé centralisée.

✓ Renouvèllement de la clé

- Quand un nœud rejoint le réseau, la clé de groupe devrait être renouvelée pour assurer la confidentialité passée de telle sorte que le nouveau membre ne puisse pas accéder aux informations d'avant son adhésion. Si l'adhésion d'un nœud crée une nouvelle passerelle ou une sous passerelle, la clé de groupe doit être renouvelée. Dans le cas où le nouveau nœud est une cellule, la clé de groupe de la cellule devrait être renouvelée, ensuite la clé du sous

groupe et la clé de groupe devraient être recalculée niveau par niveau.
- Quand un nœud quitte le réseau, la clé de groupe doit être modifiée pour assurer la confidentialité future de telle sorte que le nœud partant ne puisse pas accéder aux informations qui seront échangées. Similaire à la procédure de jointure, la clé de groupe sera soit recalculée à partir du niveau inferieur au niveau supérieur.

3.1.2.1.2 Topologie hiérarchique sans niveau prédéfini

Une topologie hiérarchique sans niveau prédéfini est une topologie hierarchique dont le nombre de niveau n'est pas fixé.

1. Protocole de Mohamed-Salah Bouassida et al

Mohamed-Salah Bouassida et al [24] proposent un protocole extensible de gestion de clé de groupe appelé BALADE, qui assure une distribution décentralisée des clés. Il élimine complètement le surcoût induit par le cryptage et le décryptage des opérations sur le flux multicast.

Dans ce protocole trois types d'acteurs sont definis : le premier correspond au contrôleur globale du groupe multicast (CG), le second acteur représente le contrôleur local (CL) qui se charge de la sécurité d'un sous groupe, le troisième correspond à un membre ordinaire du groupe. La figure 3.5 illustre le schéma global de cette architecture.

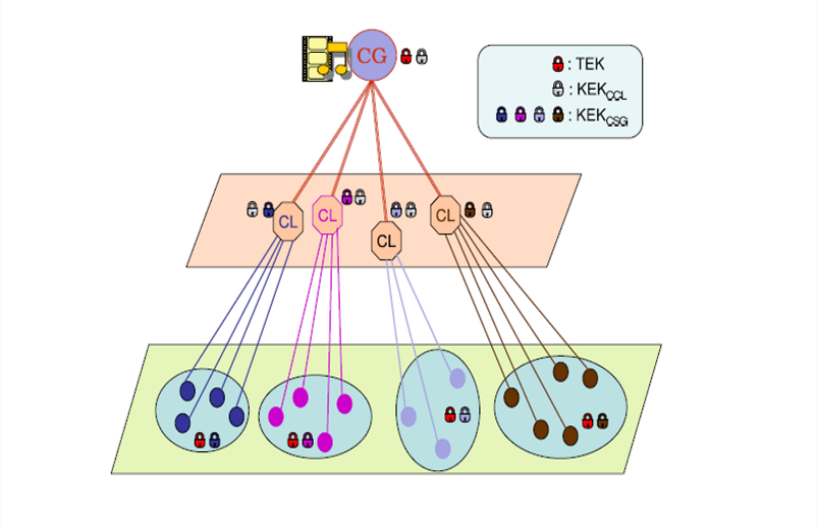

FIGURE 3.6: Architecture globale de BALADE.

• Initialisation

Le processus de clustérisation sera déclenché en utilisant l'algorithme OMCT [24].

• Mise en place de la liste de contrôle d'accès (*ACL*) :

Cette liste est construite hors ligne par l'administrateur, elle est envoyée ensuite au contrôleur global du groupe correspondant.

• Génération et distribution de la clé

Le contrôleur global génère la clé *TEK* et la distribue au groupe des contrôleurs locaux, chiffrée avec sa clé privée. A leur tour, les contrôleurs locaux envoient la *TEK* à leurs membres locaux, cryptée avec leur clé de cluster locale respective.

• Renouvellement de la TEK

Le renouvèllement de la *TEK* se fait après chaque envoi de donnée multicast.

✓ Adhésion d'un membre

Lorsqu'un membre désire rejoindre le groupe multicast, il doit s'authentifier auprès d'un *CL* qui vérifie s'il est autorisé à rejoindre le groupe ou pas, à l'aide de l'*ACL* du groupe. Si le nouveau membre est accepté au sein du cluster concerné, il sera ajouté à la liste des membres locaux.

✓ Départ d'un nœud ordinaire

Le nœud sortant envoie une requête de départ qui sera prise en charge par le contrôleur local du cluster auquel il appartient, le *CL* efface ce membre de sa liste de membres locaux et enclenche un renouvellement de la clé locale du cluster.

✓ Départ d'un contrôleur local

Le *CL* voulant quitter le groupe multicast demande à ces membres de rejoindre d'autres clusters via un message. Puis, il envoie un message de départ au groupe de contrôleurs locaux, une mise à jour de la clé partagée avec ce membre se fera par le *CG*.

2. Protocole de Hani Ragab Hassan et al

Hani Ragab Hassan et al [14] proposent un protocole de "Gestion de Clés dans la Communication de Groupes Hiérarchiques ", c'est une architecture efficace de distribution de clés, et ce dans le cas où le groupe est organisé en plusieurs classes disjointes. Les sous classes respectent une certaine hiérarchie.

• Modèle Hi-KD

L'idée principale de Hi-KD (Hash-based Hierarchical Key Distribution for Group Communication) est de lier les clés entre elles, de manière à permettre le calcul des clés inférieures, clés du groupe, le centre génère aléatoirement une clé de base *K*. Cette clé servira de clé initiale pour la classe *1*, et sera notée ainsi K_1^1. Par la suite, *S*(un centre de distribution de

clés) calcule les clés des autres classes c. Les autres clés sont calculées à l'aide de la formule suivante :

$$\mathrm{K}_{c+1}^{1} = H(K_c^1), c \in \{1, \ldots, C-1\}$$

Où H est une fonction de hachage et C le nombre de classe.
Ainsi, chaque utilisateur peut déduire, à partir de la clé de sa classe, les clés des classes inférieures. L'ensemble des clés à un moment donné est appelé chaîne de clés. Il suffira alors d'envoyer à chaque classe sa propre clé.

• Renouvellement partiel
Un renouvellement R_t se produisant à la période *p* sera procédé de la manière suivante : - *S* génère une clé K_t^{p+1}.
- *S* calcule la suite $K_j^{p+1}, i < j \leqslant C$ en utilisant la fonction de hachage H.
- *S* envoie chaque clé K_j^{p+1} à la classe correspondante j, $i \leqslant j \leqslant C$. Et envoie un message de notification de mise à jour de la chaîne des clés (contenant le couple t,K_t^{p+1}) aux classes supérieures.

• Renouvellement d'une plage
Un renouvellement R_t, *u* se déroule de la manière suivante :
- S génère une clé K_t^{p+1}.
- S calcule la suite K_j^{p+1}, $t < j \leqslant u$.
- S envoie chaque clé K_j^{p+1} à la classe correspondante j, $t \leqslant j \leqslant u$. Et envoie deux messages de notification de mise à jour de la chaîne des clés :
- Un message aux classes supérieures à t leurs indiquant que K_t^{p+1} est à utiliser à partir de t
- Un deuxième message indiquant que K_{u+1}^{p+1} est à utiliser à partir de $u+1$, pour les classes supérieures à $u+1$.
Les classes concernées par ces messages mettent à jour leurs tables de clés.

3. Protocole de Joseph Chee Ming Teo et al

Joseph Chee Ming Teo et al [25] proposent un protocole d'accord de clé de groupe (*GKA*) qui permet de sécuriser les communications de groupe dans un grand réseau ad hoc de *n* utilisateurs considérés comme un seul groupe, ce dernier est représenté en une structure hiérarchique circulaire (*C-H*). (La figure 3.6)

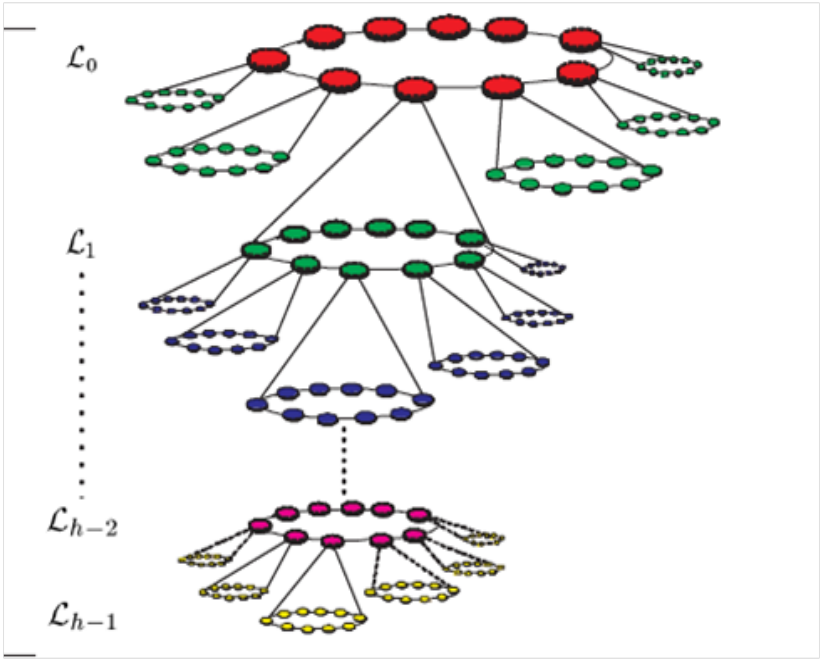

FIGURE 3.7: Illustration du modèle de groupe (C-H).

Le déroulement du protocole GKA comprend quatre phases.

• **Phase 1 :** Le protocole *GKA* pour le modèle (*C-H*) commence à la couche inférieure L_{h-1}. Chaque sous groupe $SG_j^{(L_{h-1})}$ pour *j* $\in\{0,\ldots,\mathrm{C}^{h-1}-1\}$ exécute le protocole *BD GKA* [30] afin d'obtenir sa clé de sous groupe $K_{SG_j^{(L_{h-1})}}$. Où C est le nombre de membres dans chaque sous groupe et h représente le nombre total de couches dans le modèle (C-H).

• **Phase 2 :** Pour les couches L_v où *v* $\in$ { h-2,..., 1 }, chaque membre de sous groupe $U_{(j,k)}^{(L_v)}$ du sous groupe $SG_j^{(L_v)}$ ou *j* $\in$ { 0,..., c^v-1 } et $(k=j+l)$ pour *l* $\in\{0,\ldots,c-1\}$, qui est aussi le contrôleur du sous groupe $U_{SG_k}^{(l_{v+1})}$ du sous groupe $SG_k^{(l_{v+1})}$ dans la couche L_{v+1} le protocole *BD GKA* se déroulera au sein de son sous groupe $SG_j^{(L_v)}$ pour obtenir sa clé de sous groupe $K_{SG_j^{(L_v)}}$. Cette phase se termine lorsque tous les sous groupes jusqu'à la couche *L1* auront calculé leurs clés de sous groupes respectives.

• **Phase 3 :**

Calcul de la clé : Tous les membres du sous groupe $U_{(0,k)}^{(L_0)}$ du sous groupe $SG_0^{(L_0)}$ à la plus haute couche L_0 utiliseront la clé de sous groupe $SG_k^{(L_1)}$ du sous groupe $k_{SG_k^{(L_1)}}$ pour calculer *k* et diffuser $z_k^{(L_0)} = g^{H(K_{SG_k^{(L_1)}})} \bmod p$ dans le premier tour du protocole *BD GKA*. Après avoir terminé l'exécution de *BD GKA*, chaque $U_{(0,k)}^{(L_0)}$ sera en mesure de calculer la clé de sous groupe $K_{SG_0^{(L_0)}}$, qui est aussi la dernière clé de groupe *K* de l'ensemble du groupe. Chaque $U_{(0,k)}^{(L_0)}$ utilise la clé de sous groupe $K_{SG_k^{(L_1)}}$ de son sous groupe $k_{SG_k^{(L_1)}}$ et un algorithme de chiffrement à clé symétrique $E_k(m)$(où *m* est le message pour le chiffrement et *k* est la clé secrète) à produire et à diffuser $E_{K_{SG_k^{(L_1)}}}(K)$ à son sous-groupe respectif $SG_k^{(l_1)}$.

Phase 4 : Distribution de la clé : Chaque membre de sous groupe $U_{(j,k)}^{(L_v)}$ à la couche L_v ou $v \in \{1,\ldots,$ h-2 $\}$ déchiffre le message chiffré $E_{K_{SG_j^{(L_v)}}}$ *(K)* reçu par le contrôleur du sous groupe auquel il appartient $U_{SC_j}^{(L_v)}$ qui fait partie aussi d'un sous groupe dans la couche L_{v-1} pour obtenir la dernière clé de groupe *K*. Chaque $U_{(j,k)}^{(L_v)}$, qui est également le contrôleur d'un sous groupe dans le sous groupe $SG_k^{(L_{v+1})}$ à la couche L_{v+1}, utilise la clé de sous groupe $K_{SG_k^{(L_{v+1})}}$ et un algorithme de chiffrement à clé symétrique E_k *(m)* pour produire et diffuser $E_{K_{SG_k^{(L_{v+1})}}}$*(K)* à son sous groupe $SG_k^{(L_{v+1})}$ à la couche L_{v+1}. Ce procédé se poursuit jusqu'à ce que tous les membres du sous groupe $U_{(j,k)}^{(L_{h-1})}$ à la couche L_{h-1} obtiennent la clé finale du groupe K en déchiffrant le message reçu $E_{K_{SG_j^{(L_{h-1})}}}$ *(K)*.

4. protocole de Bing Wu

Bing Wu et al[26] proposent le protocole SEGK "Simple and Efficient Group Key management scheme", le schéma de base repose sur la formation d'un arbre dans les MANETs. Deux arbres multicast sont construits et entretenus en parallèle pour assurer la tolérance aux pannes.

• Construction des doubles arbres multicast

L'initiateur du groupe est responsable de l'envoi de messages de rafraîchissement périodique aux membres pour maintenir la connexion de la double arborescence multicast. Après une période de temps prédéfinie de fonctionnement, un membre du groupe pourrait décider d'agir en tant que coordinateur du groupe et informe le groupe. Tous les membres doivent avoir des tours pour agir en tant que coordinateur du groupe. Les deux arbres multicast peuvent être utilisés en permettant à un arbre dans un état actif et à un autre dans un état inactif la sauvegarde des données. Après une période de temps prédéterminée pour la phase d'initialisation du groupe, les arbres multicast sont formés.(La figure 3.6)

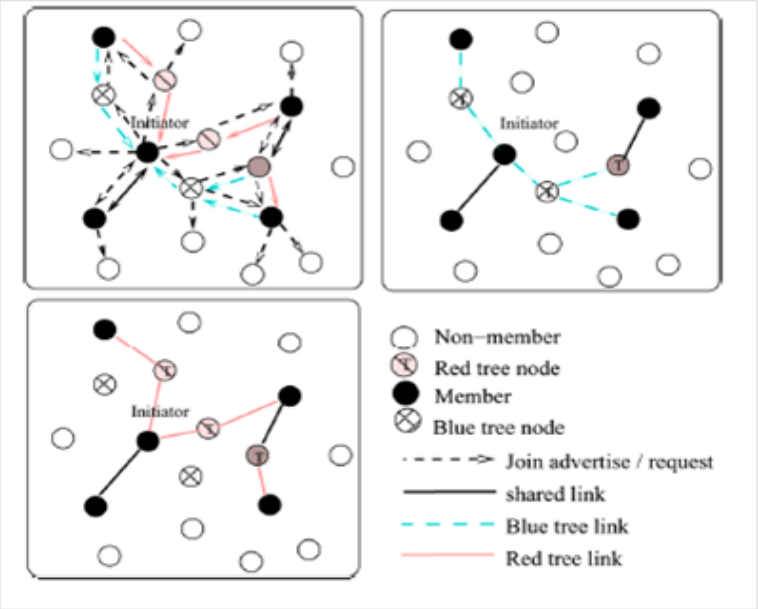

FIGURE 3.8: Double arbre multicast.

• Détection des membres sortants

Nous définissons un départ physique et un départ logique. Pour le départ physique, les nœuds se déplacent sur la plage du réseau quant au départ logique, les nœuds restent encore à l'intérieur du réseau, sans participer à l'activité du groupe.

• Protocole d'établissement de clé de groupe

✓ Initialisation de la clé de groupe

- **Etape1 :** M_c est le groupe d'initiateur, il fournie une clé aléatoire chiffré $br_c = g^{r_c}$ et deux clés virtuels br et br_0. Chaque membre M_i, $i \in [1, n]$ et $i \neq c$ répond avec une clé aléatoire chiffré br_i.

- **Etape2 :** L'initiateur M_c calcule la clé intermédiaire $k_i = (br_i)^{k_{i-1}}$ et envoie en multicast la clé intermédiaire chiffré $bk_i = g^{k_i}$ ∀i ∈ $[1, n]$; tel que $k_0 = r$, à tous les membres.

- **Etape 3** : Tout M_i, $i \in [1, n]$ calcule $k_i = (bk_{i-1}^{r_j}(k_0 = r))$ et récursivement $k_j = (br_j)^{k_{j-1}}$,$j \in \forall[i, n]$, $K_G = br_0^{k_n}$.

✓Adhésion d'un nœud

- **Etape 1 :** Un nouveau membre M_{n+1} génère une clé aléatoire r_{n+1} et diffuse la valeur br_{n+1}. Le coordinateur envoie en unicast la clé bk_n et br_0 chiffrée au nouveau membre.

- **Etape2 :** Chaque M_i, $i \in [1, n]$ calcule $k_{n+1} = (br_{n+1})^{k_n}$, le nouveau membre calcule aussi $k_{n+1} = (bk_n)^{r_{n+1}}$ et M_i, $i \in [1, n+1]$ peut calculer $K_G = br_0^{k_{n+1}}$.

✓ Départ d'un membre

- **Etape1 :** M_c notifie le groupe pour le départ d'un membre M_l, M_c génère une clé r' et

envoie en multicast la valeur br', ainsi que la clé intermédiaire $k_i = (br_i)^{k_{i-1}}$ et $bk_i = g^{k_i}$, à tout les membres sauf à l.
- **Etape2 :** Tout M_i, $i \in [1,n]/\{l\}$ calcule la clé intermédiaire $k_i = bk_{i-1}^{r_j}(k_0 = r')$ et récursivement $k_j = (br_j)^{k_{j-1}}$, $\forall j \in [i,n]\ /\{l\}$, $K_G = br_0^{k_n}$.

3.1.2.2 Topologie en clustring

Une topologie en clustring est une topologie qui permet de regrouper les nœuds proches, ces derniers seront gérés par un leader appelé cluster-head qui est choisi suivant un procéssus d'élection distribué.

1. Protocole de Kaouther Drira et al

Kaouther Drira et al [27] proposent le protocole ECGK "An efficient clustring scheme for groupe key management in MANETs" qui utilise un schéma de clustering basé sur la confiance. Dans cette approche, l'ensemble des nœuds est divisé en clusters. Chaque cluster contient un cluster noyau (core) incluant un cluster-head et des membres du noyau, ainsi qu'un cluster peripherie contenant les membres de la périphérie (figure 3.7).

Deux seuils de confiance sont définis : $S_{min}[1,0]$ et $S_{max}[0,1]$. La relation de confiance entre deux nœuds *(i, j)* est définie par la valeur de confiance tv *(trust value)* qui les relie. Elle peut être :

- **Totale (TT)**

  Si et seulement si : $tv(i,j) \in [S_{max},1]$ et $tv(j,i) \in [S_{max},1]$
- **Partielle (PT)**

  Si et seulement si :$tv(i,j) \in [S_{max},1]$ et $tv(j,i) \in [S_{min},S_{max}]$
  ou $tv(j,i) \in [S_{max},1]$ et $tv(i,j) \in [S_{min},S_{max}]$
  ou $tv(i,j) \in [S_{min},S_{max}]$ et $tv(j,i) \in [S_{min},S_{max}]$
- **Méfiance (DT)**

  Si et seulement si : $tv(i,j) \in [1,S_{min}]$
  ou $tv\ (j,i) \in [1,S_{min}]$

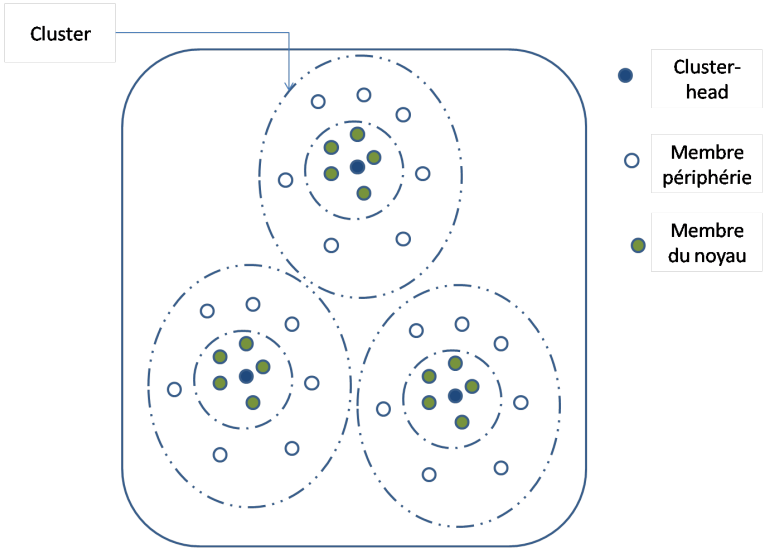

FIGURE 3.9: Topologie en clustring .

• Le protocole de clusterisation

Ce protocole s'exécute en trois phases :

- Phase 1 : Election du cluster-head

Tous les nœuds diffusent un message *"Hello"* pour avoir les informations des autres nœuds. Le nœud ayant le plus grand nombre de relations de confiance totales *TT-edge* se déclare cluster-head. Dans le cas d'égalité de la valeur *TT-edge*, le nœud ayant le plus grand nombre de voisins est désigné cluster-head. S'il y a une égalité parfaite dans les deux critères précédents, le nœud possédant le plus grand *ID* sera privilégié. Une fois qu'un nœud est élu cluster-head, il met son identité dans le champ *CH* du message *"Hello"* et met à zéro le champ *Hop-CH*.

- Phase 2 : Formation du noyau

Le noyau contient tous les voisins ayant une relation *TT* avec le cluster-head. Leur message *"Hello"* possède l'identité du cluster-head dans le champ *CH* et la valeur *"1"* dans *Hop-CH*. Un nœud peut avoir une relation *TT* avec plusieurs clusters-heads, dans ce cas, ce nœud choisit le cluster-head ayant le plus grand *TT-edge*.

- Phase 3 : Formation de la périphérie

⋆ **Etape 1** (membres de la périphérie avec des relations *TT*) : Après la constitution des noyaux, s'il y a des nœuds qui n'ont adhéré à ancun groupe et qui ont des relations *TT* au moins avec un noyau, rejoignent le cluster avec lequel ils ont la plus grande valeur de confiance.

⋆**Etape 2** (membres de la périphérie avec des relations *PT*) : Un nœud qui a une relation *PT* avec au moins un nœud privilégie toujours le cluster avec lequel il a la plus courte distance (nombre de sauts) du cluster-head. Le voisin avec lequel il a une relation *PT* et une plus courte distance au cluster-head devient son parent dans un arbre routé vers le cluster-head. Cet arbre simplifiera la communication entre les clusters.

• Maintenance de la topologie

1. Élection d'un nouveau cluster-head

Plusieurs facteurs impliquent l'élection d'un nouveau cluster-head. Nous citons deux cas :

⋆ **L'échec du cluster-head :** Quand un cluster-head échoue, les membres du noyau ne reçoivent aucun de ses messages (*hello*). Dans ce cas, les membres du noyau réinitialisent le champ *CH* de leurs messages *hello* à *NULL*. Lors de la réception de ces messages modifiés, les membres de la périphérie sous-jacents mettent également les champs *CH* de leurs messages *hello* à *NULL*. Cela lance la phase de clusterisation de ces nœuds et réorganise le cluster.

⋆ **Modification de la valeur TT-edge :** Un cluster-head qui réalise que l'un de ses voisins a une plus grande valeur TT-edge réinitialise le champ CH de ses messages hello à NULL. Lors de la réception de ces messages mis à jour, les membres du cluster propagent cette réinitialisation. Cela conduit à la situation de l'échec du cluster-head et au lancement de l'algorithme de clustérisation.

2. Rupture des relations de confiance

Un nœud contenant une relation TT ou PT avec d'autres nœuds fait partie d'un cluster, cette relation peut se brisée en raison de la mobilité des nœuds et de la valeur de confiance qui diminue à cause des mauvaises interactions échangées. Dans le cas où un nœud n'a plus de relation TT ou PT avec son cluster, il devient malicieux et doit être exclu du cluster. Le nœud rejoindra, ensuite, un autre cluster s'il a des relations TT ou PT avec d'autres nœuds dans le réseau.

3. Echec d'un membre du cluster ou de la périphérie

Lorsqu'un élément du noyau ou un élément de périphérie échoue, les membres de la périphérie qui en dépendent sont isolés à partir du cluster. Ces membres ré-exécutent la phase 3 de l'algorithme de clusterisation pour sélectionner un nouveau parent dans le cluster ou adhérer à un autre cluster.

4. Gestion des nouveaux nœuds

Quand un membre du groupe j s'authentifie auprès d'un membre du groupe voisin *i*, *i* affecte la valeur *1* à sa relation de confiance avec *j* ($tv(i, j) = 1$). Cela signifie que i accorde une relation *TT* à l'arrivée du nouveau membre du groupe, et par conséquent, il aura accès à la session du groupe. Toutefois, si *i* et *j* ne parviennent pas à s'authentifier on leurs attribue une relation de méfiance.

• Gestion de clé

Les membres du cluster et le cluster-head forment une zone d'accord de clé. Cela signifie que tous les membres du noyau contribuent au calcul de la *TEK* du cluster. La périphérie est une zone de distribution de clé. Cela signifie que les membres de la périphérie obtiennent la *TEK* du cluster via les membres du noyau. Chaque cluster c conserve une TEK_c clé de

chiffrement du trafic qui sert à chiffrer/déchiffrer le flux multicast par tous les membres du cluster. Cette clé TEK_c sera encryptée avant d'être distribuée en utilisant la clé *KEK* (key encryption key).

1. La gestion des clés Intra-cluster

- Établissement des clés

Les membres du cluster c calculent la TEK_c en passant par deux étapes :

⋆ **Etape1 :** Chaque membre du cluster c génère un secret aléatoire r_i, et envoie une version chiffré $br_i = gr_i$ mod p au cluster head.

⋆**Etape2 :** Le cluster-head l élève chaque secret reçu des membres du noyau à la puissance de son propre secret r_l et le diffuse à tous les voisins ($g^{r_i r_l}$ mod p). Chaque membre du noyau supprime son secret à partir de $g^{r_i r_l}$ pour obtenir g^{r_l}, la clé du cluster est calculée par tous les membres de la manière suivante : TEK$_c = g^{r_l} * \prod g^{r_i r_l} = g^{r_l}(1 + \sum r_l)$/i = 1,n et i$\neq l$.

⋆ **Adhésion d'un nœud**

Un message 'join' est envoyé du nœud désirant adhérer vers le nœud parent auquel il est rattaché, l'élément du cluster génère un nouveau secret le chiffre et le transmet au cluster-head. l'exécution de l'étape 2 du processus d'établissement des clés sera déclenchée, tous les nœuds concernés calculent une nouvelle clé *KEK*, le cluster envoie un message $new - TEK - RQ$ au cluster-head qui génère un nouveau secret et exécute la deuxiéme étape du TRP afin de calculer une nouvelle TEK.

⋆ **Départ d'un nœud**

Un membre quitte le groupe volontairement ou en étant exclu à cause de l'apparition d'une relation de méfiance (DT).

Cas d'un départ volontaire : Le nœud quittant le groupe envoie un message $'LEAVE'$ à son parent dans le cluster. ce message est transmis a travers les nœuds du cluster jusqu'à ce qu'il atteigne le nœud responsable du départ.

Cas d'exclusion d'un nœud : Un nœud est relié à un cluster en ayant une relation *TT* ou *PT*, dans le cas ou cette relation est brisée le parent du nœud envoie un message $'LEAVE'$ vers le cluster, les nœuds du noyau mettront à jour les clés *KEK*, et enverront un message *NEW-TEK-RQ* au clusterhead, et une clé *TEK* est recalculée.

2. Gestion des clés Inter-cluster

Les différents cluster-heads assurent la communication entre les clusters. Chaque cluster-head calcule une clé TEK avec chaque cluster-head du cluster adjacent en utilisant le protocole Diffie-Hellman[R]. Comme suit :

⋆ Chaque cluster-head h Génère un secret r_h et envoie une version chiffrée $(g^{r_h} modp)$ dans le message *Contact-Adj-CH* vers le bas du cluster.

⋆ Lorsque le message atteint une passerelle, il est transmis à la passerelle correspondante au cluster adjacent si le lien entre les deux passerelles est un *TT* ou un lien *PT.*

⋆ Le message remonte jusqu'aux cluster-head du cluster adjacent.

⋆ Chaque paire de clusterhead x et y peut calculer une clé TEK $K = g^{r_x r_y} mod p$ en utilisant le protocole de Diffie-Hellman.

• **Transfert de données**

Afin d'assurer la sécurité des communications les membres du groupe doivent suivre les règles suivantes :

⋆ Les données multicast suivent uniquement les relations *TT* ou *PT*.

⋆ Pour diffuser certaines données, un membre les chiffre avec la *TEK* de son cluster et les transmet à ses voisins dans un message cluster-diffus.

⋆ A la réception d'un message Cluster-Diffuse pour la première fois, chaque nœud le transmet à ses voisins. Si le nœud est un cluster-head, il le décrypte et le recrypte avec la TEK qu'il partage avec chaque cluster-head adjacent et le transmet à chacun d'eux dans un message *CH-to-CH*. Un message *CH-to-CH* parcourt l'arborescence du cluster jusqu'à atteindre une passerelle.

⋆ Une passerelle qui reçoit un message d'un autre cluster *CH-to-CH* pour la première fois l'envoie à son cluster-head

Tableau comparatif

Critères	BALADE	SEGK	Yu Bin et al	Jin-Hee Cho et al	HKMS	Hani Ragab Hassan et al	Joseph Chee Ming Teo et al	Kaouther Drira et al
Facteur d'échelle l affect n	Oui	oui	oui	oui	oui	Oui	oui	oui
Service de sécurité	- Authentification et contrôle d'accès - Confidentialité des données	Confidentialité des données	Confidentialité des nœuds	Confidentialité des données	Confidentialité des données	Confidentialité des données	Confidentialité des données	Confidentialité des données
Surcoût de Calcul	- Déchiffrement et rechiffrement de la TEK par les CLS -Algorithme OMTC	-génération de la clé r' et des clés virtuelles	-Génération de clé de groupe et des sous groupes –Recalcule périodique de la clé de groupe	-calcul de clé KR -exécution de GDH	O(m) -calcule de (l1GK &L2GK)	Calcul des clés de classe inférieure	Calcule de la clé de sous groupe à chaque couche	
Surcoût de Stockage	- Clés locales de clusters KEKs ACL distribuée par les CLs du groupe	passage des nœuds en mode coordinateur	-Clé de groupe. -Clés de chaque sous groupe. -Clés de cellules.	-la vue régionale du groupe -liste régionale	O (*n*)	- une clé par chaque membre	Chaque membre stocke la clé du sous groupe auquel il appartient	
Surcoût de communications	Signalisation de l'algorithme OMCT CG : N+K message CL : c message MG : 1 message	-message périodique de maintenance de connexion	-Distribution de clé de groupe -Distribution des clés de sous groupe. -Distribution des clés de cellules.	-message périodique de maintenance de lien -diffusion de la liste régionale	O (*mk*)	O(n)	Les diffusions causées par l'exécution du protocole BD lors du calcul des clés	-message' hello' - TT_edge,CH -O(n) + communication entre les clusters heads
Vulnérabilité	Contrôleur global	Non	-Nœud Passerelle -Nœuds sous passerelles.	-Inondation GPS -coût élevé de calcul des mises à jour	*L1-head*	Non	Non	Le cluster-head

Table 3.2-Tableau comparatif des protocoles de l'approche orientée topologie .

Critiques

L'inconvénient d'une telle approche est son coût de maintenance en particulier dans les MANETs. En raison de la mobilité, le maintien de la structure du groupe peut entraîner des frais généraux de communication énormes.

Conclusion

Un protocole de gestion de clé de groupe doit assurer la confidentialité des données en chiffrant le flux du coté de la source et en les déchiffrant du coté du récepteur. De plus, l'authentification et le contrôle d'accès peuvent être assurés car seul les membres détenant la clé de groupe peuvent accéder aux données.

Dans ce chapitre, nous avons étudié certains protocoles de gestion de clé de groupe dans les MANETs, en les classant suivant une taxonomie choisie (à plat, orienté topologie). Dans le prochain chapitre nous pésenterons un protocole de gestion de clé de groupe dans les MANETs qui est orienté topologie.

CHAPITRE 4

CONTRIBUTION

Introduction

La gestion de clé est une tâche difficile à mettre en œuvre. En effet, à chaque fois qu'un membre rejoint ou quitte le groupe la clé devrait être mise à jour pour préserver la confidentialité au sein du groupe. Une solution de sécurité répondant à ces besoins se basera nécessairement sur un système efficace de gestion de clés durant la session. Dans une session multicast sécurisée, le rôle du système de gestion de clés est d'assurer la confidentialité des données échangées tout au long de la session. Ceci passe par le déclenchement dun processus de renouvellement des clés après chaque événement qui survient dans la session. Après l'arrivée d'un nouveau membre dans le groupe, les clés doivent être renouvelées pour garder la confidentialité passée et éviter que le nouveau membre n'accède aux messages échangés avant son arrivée. De même, quand un membre quitte le groupe, le renouvellement des clés assurera la confidentialité future et empêchera l'ancien membre d'accéder aux messages échangés après son départ.

L'approche à plat souffre du problème 1-affecte-n, où un unique changement d'appartenance à un groupe (adésion ou départ) se traduit par un processus de renouvelement qui perturbe tous les membres du groupe. En outre, la plupart des protocoles dans cette approche ont besoin d'un serveur central. Donc, ils ne sont ni évolutifs, ni tolérants aux pannes et c'est pour cela que nous avons choisit une approche orientée topologie afin d'en tirer profit des différentes caractéristiques que peut offrir cette organisation qui est celle des clique maximum que nous retrouvons dans divers domaines tels que le routage dans les réseaux[32] et les intérference[33].

Dans ce chapitre, en premier lieu, nous présentons les différentes étapes nécessaires au bon fonctionnement d'un protocole de gestion de clé dans un environnement mobile. En second lieu, nous illustrons le protocole par un exemple concret, et enfin, nous étudions les différents avantages et inconvénients du protocole proposé.

4.1 Motivation

Dans notre approche, nous utilisons une topologie spécifique, qui est celle de clique maximum, dans ce qui suit nous levons l'ambiguité sur les differentes notions de clique en définissant les notions : clique, clique maximale et clique maximum.

Définition 1
*Une **clique** dans un graphe non orienté $G = (V, E)$ est un sous ensemble de l'ensemble des sommets $C \subseteq V$, tel que pour chaque paire de sommets de C, il existe une arête reliant les deux. Ceci est équivalent à dire que le sous graphe induit par C est complet.*

Définition 2
*Une **clique maximale** est une clique qui ne peut pas être étendu en incluant un sommet adjacent, qui est, une clique qui n'existe pas exclusivement dans l'ensemble des sommets d'une plus grande clique.*

Définition 3
*Une **clique maximum** est une clique maximale qui a la plus grande cardinalité possible dans un graphe donné.*

Les services que propose cette topologie sont les suivant :

- Subdiviser le graphe (réseau) en sous groupe de manière à réduire le facteur 1 affect n, lors d'un départ d'un nœud. Ainsi le renouvelement de la clé de ne s'effectuera qu'au sein du sous groupe concerné.
- Les cluster-heads sont élus d'une manière déterministe, il n'y a aucune confusion.
- Tous les cluster-heads sont à portée ce qui facilite la communication entre eux en un temps bref sans passer par des nœud intermédiaire.
- La consommation d'énergie lors de l'envoie d'un flux multicast au niveau de la clique maximum (les sommets de la clique maximum représentent les cluster-heads du groupe) est inferieur à celle consomée dans le cas ou les cluster-heads sont non interconnectés.

4.2 Principe de la proposition

Le but de ce travail est d'assurer un service de sécurité qui englobe la confidentialité passée et la confidentialité future lors d'une communication de groupe dans un réseau ad hoc, et de réduire l'influence du départ d'un nœud ou de son arrivée, et d'effectuer des mises à jour de la clé si nécessaire en un temps minimum, ainsi que la conservation de l'énergie en exploitant les caractéristiques de la topologie de clique maximum.

4.2.1 Etablissement de la topologie

Dans les topologies en clustering les nœuds du groupe sont répartis en clusters afin de faciliter la gestion du trafic et des clés. Pour pouvoir diviser le groupe en clusters plusieurs protocoles de clusterisations[34][35] sont utilisés pour élire un leader dans chaque cluster en se basant sur plusieurs paramétres (l'identifiant, le nombre de voisins, l'énergie, le poids, etc). Dans notre topologie les cluster-heads representent les sommets de la clique maximum.

◇ **La phase initiale**

Nous optons pour une clustérisation basé principalement sur le concept des cliques maximums ; soit $G = (V; E)$ un graphe non orienté, représentant un réseau ad hoc, où V est l'ensemble des sommets, qui sont les stations ad hoc, et E est l'ensemble des liens, qui relient les stations qui sont à portée de la transmission de chaque autre station. Si un sous graphe C de G forme une clique maximum alors les sommet de l'ensemble C representent les cluster-heads du groupe(figure 4.1).

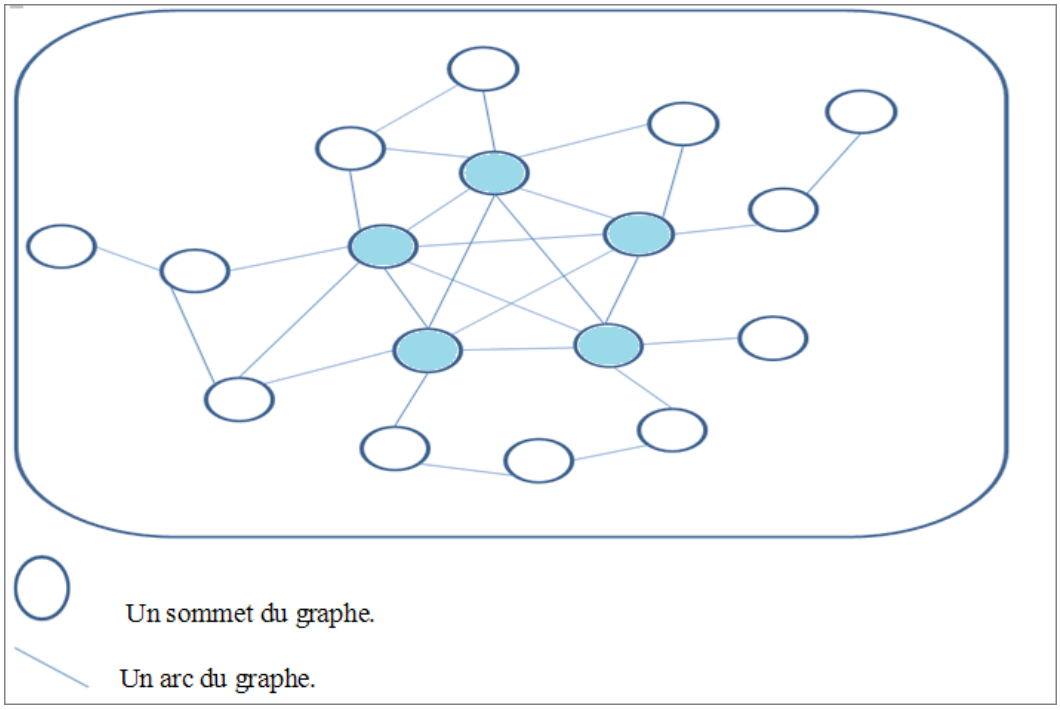

FIGURE 4.1: Réseau initiale.

◇ **Recherche d'une clique maximum**

La recherche d'une clique maximum, est un problème NP complet[36], plusieurs solutions approximatives ont été proposées [37], [38], [39],pour la détermination d'une clique maximum dans un graphe. Le choix de l'algorithme de recherche d'une clique maximum c'est posé sur l'algorithme de Gupta[39] pour sa compléxité moins coûteuse qui est de $O(m\triangle^2)$ où m est le nombre d'arc du graphe et $\triangle$ est le degré maximum du graphe.

◇ **Formation des clusters**

Les nœuds restants seront affectés à chaque cluster-head. La construction des clusters se fait de la façon suivante :

Etape 1 : Détermination des cluster-heads

- Tous les nœuds qui appartiennent à la clique maximum sont définis cluster-heads.
- Les nœuds cluster-heads s'échangent des messages "Hello" pour collecter les informations de chacun (ID, clé publique).
- Chaque cluster head envoie un message "I am cluster-head" à tous ses voisins à un saut, pour les informer que c'est lui le cluster-head et pour qu'il leurs communique sa clé publique.
- Chaque nœud qui reçoit ce message s'enregistre auprès du cluster-head émetteur et sera placé dans son cluster.

Etape 2 : Affectation des nœuds aux cluster- heads

- Tous les nœuds ayant un lien direct avec un des cluster-heads de la clique maximum feront partie de son cluster.
- Si un nœud reçoit plus d'un message "I am cluster-head", ce nœud privilégiera le cluster-head ayant la plus grande valeur d'énergie et s'enregistrera auprès lui.
- Tous les nœuds restants ayant un lien avec un nœud faisant déjà partie d'un cluster, feront eux aussi partie de ce cluster.

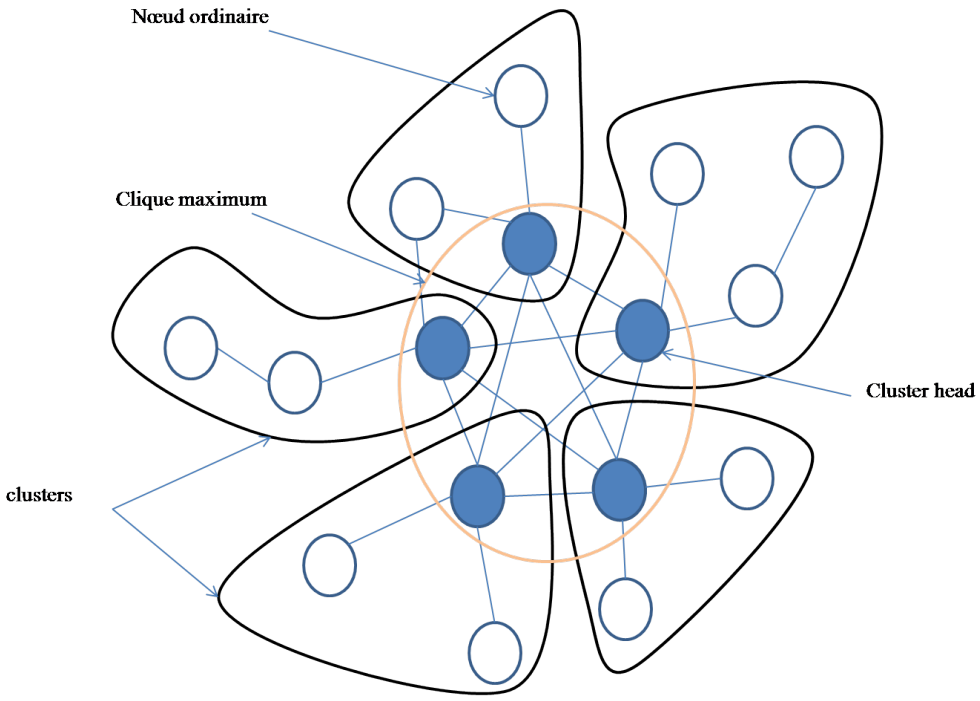

FIGURE 4.2: Schéma de clustérisation.

4.2.2 Génération de clés

Afin d'assurer la confidentialité des données et de n'avoir accès au flux multicast que par les membres légitimes du groupe, des clés de chiffrement et de déchiffrement sont nécessaires. Celles-ci sont obtenues comme suit :

1. Etablissement de la clé intra-cluster TEK_{intra}
La clé du cluster est générée par le cluster-head de manière centralisée, il génére $K_{cluster}$ puis il la hache pour obtenir la TEK_{intra}. De ce fait la clé intra-cluster est calculée comme suit :

TEK_{intra}= H ($K_{cluster}$). Où H est une fonction de hachage

Le cluster-head chiffre cette clé avec les clés publiques des membres de son sous groupe et leurs envoie en unicast.

2. Etablissement de la clé inter-cluster TEK_{inter}
Au départ le nœud initiateur sera le nœud qui initiera la communication. Les autres nœuds de la clique maximum seront initiateurs à tour de rôle. Les membres de la clique maximum calculent TEK_{inter} en utilisant les deux tours de l'amélioration du protocole d'accord de clé TRP [41] proposé dans l'article de Kaouther Drira [27].

Tour 1 :
Chaque membre de la clique maximum i génère un secret aléatoire r_i et envoie sa version chiffré $br_i = g^{r_i} modp$ à l'initiateur de la communication j.
Tour 2 :
L'initiateur élève à chaque secret d'un membre de la clique maximum son propre secret r_j et diffuse à tout les membres de clique $g^{r_i r_j} modp$ pour tout i dans $C - \{j\}$, Où C est l'ensemble des éléments de la clique maximum. Chaque membre supprime son secret à partir $g^{r_i r_j}$ pour obtenir g^{r_j}.
Tous les membres calcule la clé inter-cluster eux même :

$$TEK_{inter} = g^{r_j} * \prod_{i \in C-\{j\}} (g^{r_i r_j}).$$

4.2.3 Mise à jour des clés

Dans le but d'assurer de parfaites confidentialités passée et future, à chaque fois qu'un changement d'adhésion a lieu dans un cluster, un processus de renouvèlement des clés aura lieu.

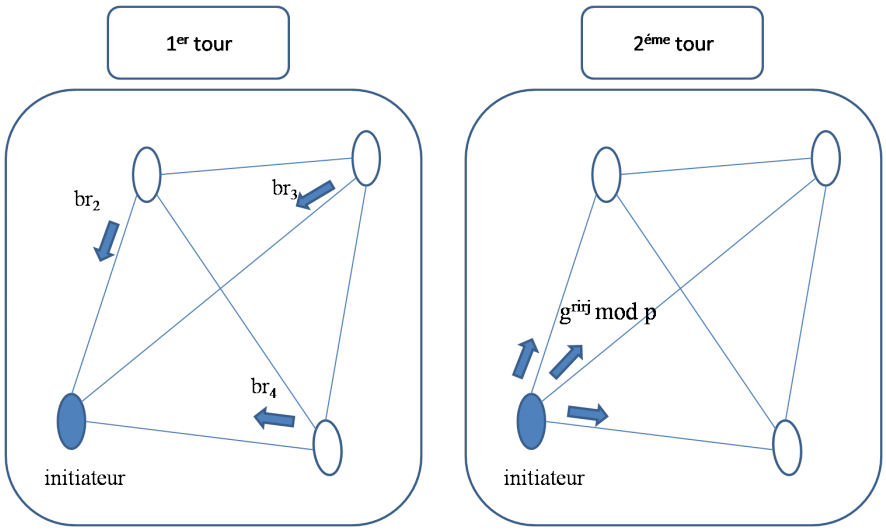

FIGURE 4.3: Etablissement de clé inter.

◇ **Adhésion d'un nœud**

Lorsqu'un membre souhaite joindre le groupe multicast, il envoie une requête d'adhésion à tous ses voisins. Chaque nœud recevant cette requête la transmet à son cluster-head. Si le nœud est autorisé à rejoindre un cluster, une requête favorable lui est transmise. Plusieurs réponses favorables de la part d'autres cluster-head peuvent être attribuées à un même nœud. Ce dernier privilégiera le cluster-head ayant la plus grande valeur d'énergie et si deux cluster-head ont deux valeur d'énergie égales alors le nœud adhérera au cluster-head qui a le plus grand identifiant.

Afin d'assurer la confidentialité passée le cluster-head générera une nouvelle TEK_{intra} et après l'échange de clé publique entre le cluster-head et le nouveau membre, le cluster-head diffuse la nouvelle clé TEK_{intra} aux ancienx membres du cluster chiffrée avec l'ancienne TEK_{intra} et l'envoie au nouveau membre chiffrée avec sa clé publique.

◇ **Départ d'un membre ordinaire**

Quand un nœud ordinaire quitte son cluster, il informe le cluster-head de son départ en lui envoyant une requête de départ. A son tour, le cluster-head regénére une nouvelle TEK_{intra}, et l'envoie à tout les membres de son cluster.

◇ **Départ d'un cluster-head**

Quand un cluster-head quitte le réseau, les nœuds appartenant à son cluster enverront des demandes d'adhésion aux cluster-heads les plus proches, ces derniers répondront par une réponse favorable ou défavorable selon le besoin du cluster. Lors de la réception des réponses les nœuds ordinaires décideront à quels clusters adhérer.

Le départ d'un cluster-head n'affecte pas sur la topologie de clique maximum vu que les nœuds de ce sous groupe restent toujours interconnectés. Cependant après plusieurs départs de cluster-heads on perd l'utilité de la clique maximum. Donc nous fixons un seuil

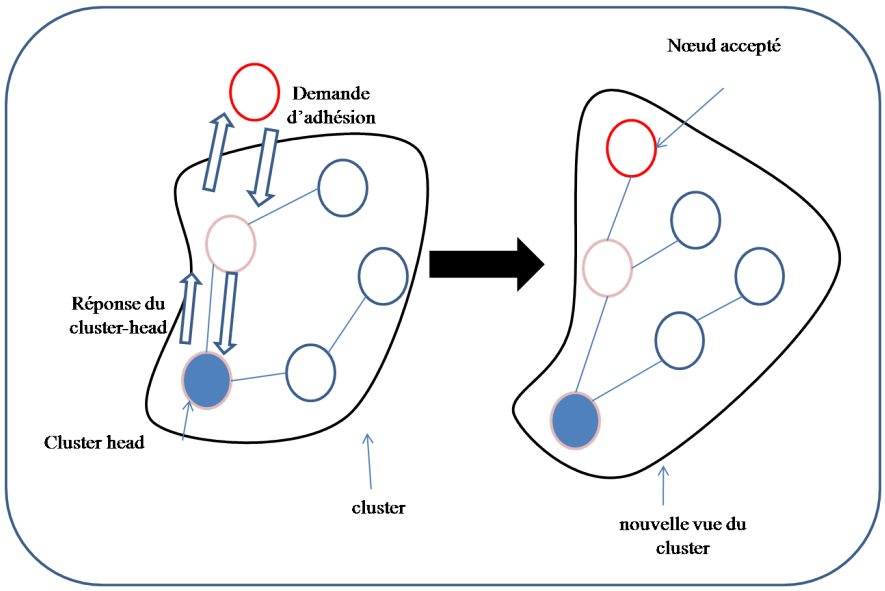

FIGURE 4.4: Adhésion d'un nœud.

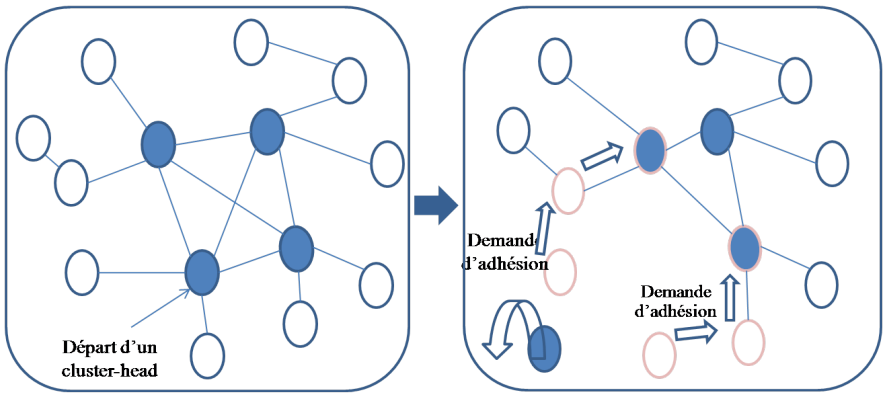

FIGURE 4.5: Départ d'un cluster-head.

$'S'$. Après le franchissement de ce seuil, nous ré-exécutons l'algorithme de recherche d'une clique maximum(figure4.5).

4.2.4 Transfert de données

Les données multicast sont transférées de la manière suivante :

– Quand deux nœuds se trouvant dans un même cluster veulent communiquer, ils chiffrent et déchiffrent le message avec leur clé de cluster TEK_{intra}.

– Dans le cas où ils font partie de deux clusters différents, le nœud émetteur envoie le message à son cluster-head chiffré avec la clé de son cluster, le cluster-head le déchiffre, le chiffre à nouveau avec la clé inter-cluster et l'envoie au cluster-head du nœud de destination. Ce dernier le déchiffre, ensuite, le chiffre avec la clé de son cluster et l'envoie au destinataire.

4.3 Analyse de l'énergie

L'énergie est un facteur majeur pour les réseaux sans fil et dans le contexte de mobilité du fait que les stations du réseau sont en activité permanente (transmission, réception, ...). Cette activité ne pourrait être présente et continue qu'à travers les batteries des stations présentant l'inconvénient d'être de charge limitée.

Dans cette partie, nous montrons que notre topologie "Clique maximum" rend le coût de consommation d'énergie lors de la transmission d'un flux multicast invariable quelque soit le nombre de cluster-heads (nombre de nœud dans la clique maximum). Cela est du au fait que tous les cluster-heads sont interconnectés entre eux (tous les cluster-heads sont apporté l'un de l'autre), ce qui facilite le routage et réduit la consommation d'énergie.

En générale le coût de consommation d'énergie $E_{Tx}(B)$ pour B bits peut etre calculé en utilisant la formule suivante [25].

$$E_{Tx}(B) = B \cdot \varepsilon Tx.$$

où εTx désigne l'énergie consommée par la radio de communications pour transmettre un bit. Un modèle plus détaillé pour la transmission est donnée dans [40]. pour transmettre un message B-bits sur une distance d, la radio dépense :

$$E_{Tx}(B, d) = B \cdot \varepsilon Tx + B \cdot d^2 \cdot \varepsilon Tx_a mp.$$

Le coût de la consommation d'énergie lors de la réception est donné par la fonction suivante :

$$E_{Rx}(B) = B \cdot \varepsilon Rx.$$

où εRx se réfère à l'énergie consommée par la radio recevoir un bit.

Dans notre modèle le coût de consommation de l'énergie lors d'un flux multicast dans la clique maximum peut être représenter par la fonction suivante :

$$EC_{TX}(B) = \beta.$$

ou β représente le coût énergitique consommé lors de l'envoi d'un flux multicast ou
$\beta = B \cdot \varepsilon Tx$.
Quant à la réception la fonction est comme suit :$EC_{Rx}(B)$= $\beta\prime$, ou$\beta\prime$= $(B \cdot \varepsilon Rx)(n-1)$
donc le coût énergitique d'une communication est : $EC_{com}(B) = EC_{TX}(B) + EC_{Rx}(B)$

Par contre dans un réseau où les cluster-heads ne forment pas une clique maximum, le coût de consommation d'énergie augmente avec l'augmentation des nœuds cluster-heads, cela est dû au fait que certain cluster-heads ne sont pas à portée l'un de l'autre, ce qui fait que lors d'une transmission le message parcourt X station avant d'arriver à la station destinatrice cela implique que le coût de consommation d'énergie augmente, dans ce cas il sera donné par cette fonction :

$$E_{TX}(B) = \beta \cdot X + \beta.$$

Le coût énergitique de la réception est le suivant :

$$E_{Rx}(B) = \beta\prime, \text{ où } \beta\prime = (B \cdot \varepsilon Rx)(n-1)$$

Le coût énergitique d'une communication est :$E_{com}(B) = E_{TX}(B) + E_{Rx}(B)$.

Soit le coût de consommation d'énergie pour l'envoi d'une donnée de 1024 bits est de 11.1mJ, le coût de réception de cette donnée est de 7.69mJ [25] en utilisant les formules précedantes, et en variant le nombre des cluster-heads nous obtenons le graphe suivant qui illustre le coût de consommation de l'énergie lors d'une communication dans un réseau dont le nombre de cluster-head est variant.

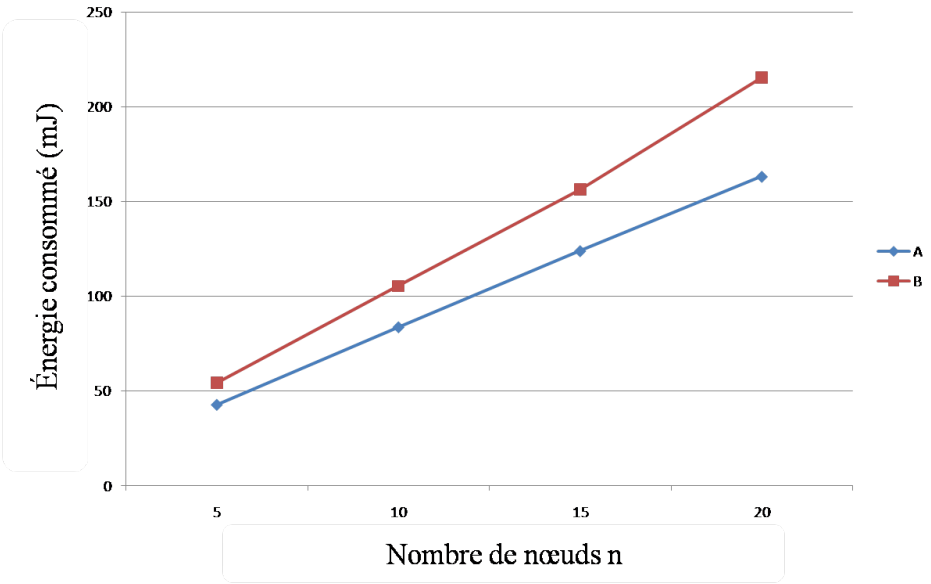

FIGURE 4.6: Graphe illustrant le cout d'énergie consomer lors d'une communication.

A :représente la consommation de l'énergie lors d'une communication multicast dans la clique maximum.

B : représente la consommation de l'énergie lors d'une communication multicast dans le modèle ou les cluster-heads ne sont interconnectés.

Ce graphe est obtenu en utilisant les valeurs calculées dans le tableau 4.1.

Nombre de nœuds	Positionnement des cluster-heads	Cout de l'énergie consommée lors de l'envoie
N=5	Cluster-heads de la clique maximum	(11 ,1 + (7 ,69 *4)) = 42,9 mJ
	Cluster-heads non interconnectés	(11 ,1 + (11,1) + (7 ,69 *4)) = 54,1 mJ
N=10	Cluster-heads de la clique maximum	(11 ,1 + (7 ,69 *9)) = 82,6 mJ
	Cluster-heads non interconnectés	(11 ,1 + 2(11,1) + (7 ,69 *9)) = 104,4 mJ
N=15	Cluster-heads de la clique maximum	(11 ,1 + (7 ,69 *14)) = 121,7 mJ
	Cluster-heads non interconnectés	(11 ,1 + 3(11,1) + (7 ,69 *14)) = 155 mJ
N=20	Cluster-heads de la clique maximum	(11 ,1 + (7 ,69 *19)) = 161,1 mJ
	Cluster-heads non interconnectés	(11 ,1 + 4(11,1) + (7 ,69 *19)) = 205,6 mJ

Table 4.1- tableau illustratif des valeurs du du graphe.

Nous supposons que le nombre de stations intermédiaires par lesquelles le flux multicast transite quand le nombre de nœud est égale à 5 est d'une seule station intermédiaire, Il est de deux lors que le nombre de nœud est égal à 10, de trois lorsque n =15 et 4 lorsque n=20.

4.4 Avantages et Inconvénients

4.4.1 Inconvénients

- Renouvèlement de clé du cluster à chaque départ et arrivée d'un nœud afin d'assurer la confidentialité future et passée.
- Surcoût de calcul induit par le chiffrement et déchiffrement lors de l'envoie des données intra cluster.
- Si le nombre de départ des cluster-heads atteint un seuil fixé 'S' cela engendre une réinitialisation de l'algorithme de recherche d'une clique maximum.

4.4.2 Avantages

- Subdivision du groupe en clusters permet de réduire le facteur 1 affecte n.
- Etablissement déterministe des cluster-heads.
- Construction de la clé inter-cluster par accord.
- Après le départ d'un cluster-head ; les nœuds qui faisaient parti de son cluster seront admit auprès d'un autre cluster.
- Conservation de l'énergie par les nœuds de la clique maximum et même par les autre nœuds puisque nous ne les faisons pas participer au routage.
- La clé inter-cluster n'est pas renouveler à chaque départ d'un cluster-head.

Conclusion

Dans ce chapitre, nous avons présenté notre protocole de gestion de clé dans les communications de groupe dans un réseau ad hoc. Ce protocole est basé essentiellement sur une topologie à clique maximum à plusieurs caractéristiques qui aide à améliorer les apports de ce scénario.

La création de la clé inter cluster est distribuée, alors que la clé intra cluster est générée d'une façon centralisée. L'architecture proposée économise essentiellement l'énergie lors des envois des données multicast.

Conclusion générale

La facilité de déploiement des réseaux mobiles Ad Hoc ainsi que leur caractère spontané en font une solution présentant de nombreux intérêts pour le domaine militaire et civil. Cependant, comme nous l'avons vu, les réseaux ad hoc possèdent des propriétés fondamentales qui rendent difficile la confidentialité des données circulant dans un groupe multicast : ces réseaux exigent que tous les usagers collaborent ensemble pour assurer l'intégrité des informations transmise à d'autres usagers du même groupe. Cette hypothèse et plusieurs autres caractéristiques (la mobilité, l'énergie, bande passante etc.) rendent le problème de sécurité dans ces réseaux un axe de recherche capital.

La solution la plus appropriée, pour assurer des communications de groupe dans les MANETs est l'établissement d'un protocole de gestion de clé de groupe. Ce protocole doit garantir la confidentialité des données multicast en assurant la gestion et la distribution de la clé de chiffrement de données TEK. Le contrôle d'accès au groupe multicast est également assuré car seuls les membres détenant la clé du groupe peuvent accéder aux flux multicast émis par la source. Le protocole de gestion de clé de groupe doit également être adapté à la nature et aux caractéristiques des réseaux ad hoc.

Dans le cadre de ce projet,nous avons étudié certaines approches existantes de gestion de clé de groupe dans les MANETs, en les classant selon leurs topologies (approche à plat et approche orienté topologie) et nous avons tenté de construire une solution de gestion de clé de groupe dans les MANETs, qui s'articule autour du principe de clique maximum en essayant de réduire l'énergie dépensée lors des envois multicast et le nombre de mise à jour lors des départs et des arrivées des nœuds.

En guise de perspectives, ce travail peut être enrichi par des simulations afin de mesurer les forces et les faiblesses du protocole et de concrétiser des résultats pour d'éventuelles comparaisons et améliorations.

Bibliographie

[1] T. Clausen and P. Jaquet. *Optimized Link State Routing Protocol (OLSR)*. IETF RFC 3626, 2003.

[2] X. Hong M. Gerla and G. Pei. *Fisheye State Routing Protocol (FSR) for Ad Hoc Networks*. IETF Internet Draft : draft-ietf-manet-fsr-03.txt, 2002.

[3] S. Das E. Belding-Royer and C. Perkins. *Ad-Hoc On demand Distance Vector routing (AODV)*. IETF RFC 3561, 2003.

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

[5] R. Marc Pearlman and Zygmunt J. Haas. *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*. IETF Internet Draft, 2002.

[6] J. Li M. Jiang and Y.C. Tay. *Cluster Based Routing Protocol (CBRP)*. IETF Internet Draft, 1999.

[7] M S. Corson J. Macker, *Réseaux mobiles ad hoc (MANET) :Routing Protocol Performance Issues and Evaluation Considerations*, IETF RFC 2501, 1999.

[8] I. Kouvelas B. Fenner B. Cain S. Deering and A. Thyagarajan. *Internet Group Management Protocol*. IETF RFC 3376, 2002.

[10] R. Vida and L. Costa. *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*. IETF RFC 3810, 2004.

[11] N. Badache H. Bettahar and D. Tandjaoui H. Seba, A. Bouabdallah. *Gestion de clés et sécurité multicast : Etude et perspective. Annals of Telecommunications*, 58(7-8), pages 1090−1129, 2003.

[12] W. Diffie and M.E. Hellman. *New directions in cryptography*. IEEE Transactions on Information Theory, 22(6), pages 644−654, 1976.

[13] S. GHAROUT. *Sécurité des communications dans les groupes dynamiques*. Thése de doctorat, Université de Technologie de Compiègne, 2009.

[14] H. Ragab A. Bouabdallah H. Bettahar ET Y. Challal. *Gestion de Clés dans la Communication de Groupes Hiérarchiques*, 2005.

[15] L. Lazos and R. Poovendram. I*n Energy-Aware Secure Multicast Communication in Ad Hoc Networks Using Geographical Location Information*, IEEE International Conference on Acoustics Speech and Signal Srocessing, pages 201 − 204, 2003.

[16] M. Haibing L. Yun Z. Changlun. *A Compossite Muticast Key Management Scheme for MANET*. Proceedings of 6^{th} Intertional Conference on ITS Telecommunications, pages 794−797, 2006

[17] S. Zhu, S. Setia, S. Xu, S. Jajodia, *GKMPAN : An efficient group rkeying scheme for secure multicast in ad-hoc networks*, Proc. First Annual International Conference on Mobile and Ubiquitous Systems : Networking and Services, pages 42−51, 2004.

[18] M. Gouda C. Wong and S. Lam. *Secure group communications using key graphs*. ACM SIGCOMM, pages 68−79, 1998.

[19] H. Lu. *A novel high-order tree for secure mtulticast key management*. IEEE Transactions on computers, 54 (2), 2005.

[20] D. Tygar A. Perrig R. Canetti and D. Song. *The tesla broadcast authentification protocol*. RSA Laboratories Cryptobytes, 5(2), 2002.

[21] N C. Wang and S-Z. Fang. *A hierarchical key management scheme for secure group communications in mobile ad hoc networks*. Journal of Systems and Software, 80(10), pages 1667−1677, 2007.

[22] J H. Choa, I R. Chena, D C. Wang. *Performance optimization of region-based group key management in mobile ad hoc networks*. Performance Evaluation, 65, pages 319−344, 2008.

[23] S. Bin, Y. Bin, *The three-layered group key management architecture for MANET*. 11th International Conference, Advanced Communication Technology. 02, Pages 1378−1381, 2009.

[24] M S. Bouassida, I. Chrisment, O. Festor. *Group key management in MANETs*. Int J Netw Secur IJNS 6(1), pages 67−79, 2008.

[25] J. Teo C. Tan. *Energy-Efficient and Scalable Group Key Agreement for Large Ad Hoc Networks*, In ACM PE-WASUN05, pages 114−121, 2005.

[26] B. Wu J. Wu and Y. Dong. *An efficient group key management scheme for mobile ad hoc networks*, Int. J. Security and Networks, 4, pages 125−134, 2008.

[27] H. Seba K. Drira and H. Kheddouci. *ECGK : An efficient clustering scheme for group key management in manets*. Computer Communications, 33 (9), pages 1094 − 1107, 2010.

[28] J. Pieprzyk and C-H. Li. *Multiparty key agreement protocols*. IEE Proceedings Computers and Digital Techniques, 147(4), pages 229 − 236, 2000.

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

[30] M. Burmester and Y. Desmedt. *A Secure and Efficient Conference Key Distribution System*, Proc. Advances in Cryptography - Eurocrypt, 839, pages 275−286, 1995.

[31] J.B. MacQueen. *Some Methods for Classification and Analysis of Multivariate Observations*. Proceedings of 5^{th} berkeley symposium on mathematical statistics and probability,University of California Press, 1, pages 281−297, 1967.

[32] A. Puri.*OPtimizing Traffic Flow in Fixed wireless NEtworks*, in IEEE Wireless Communications and Networking Conference, 2, pages 904−907, 2002.

[33] R. Gupta J.Walrand. *Approximating Maximal Cliques in Ad hoc Networks*. Proc. PIMRC, Barcelona, Spain, 2004.

[34] C C. Chiang H-K. Wu W. Liu and M. Gerla. *Routing in Clustered Multuhop*, Mobile Wireless Networks With Fading ChannelL, IEEE singapore international conference on networks, 1997.

[35] A. Ephremides J E. Wieselthier and D J. Baker. *A Design Concept For Reliable Mobile Radio Network With Frenquency Hopping Signaling*, Procedings of the IEEE, 75(1), January 1987

[36] G. Chartrand and P. Zhang. *Chromatic graph theory*,Taylor Francis Group, 2008

[37] C. Bron and J. Kerbosch. *Algorithm 457 finding all cliques of an undirected graph*. Commun ACM, 16(9), pages 575−577, 1973.

[38] E. Tomita A. Tanaka and H. Takahashi. *The worst-case time complexity for generating all maximal cliques and computational experiments*. Theoretical Computer Science, 363(1), pages 28−42, 2006.

[39] R. Gupta J. Walrand and O. Goldschmidt. *Maximal Cliques in Unit Disk Graphs : Polynomial Approximation*, OPNET Technologies Inc, 2006

[40] W. Heinzelman A. Sinha A. Wang and A. Chandrakasan. *Energy-Scalable Algorithms and Protocols for Wireless Microsensor Networks*, Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 00), June 2000.

[41] V. Issarny D. Augot, R. Bhaskar and D. Sacchetti. *A three round authenticated group key agreement protocol for ad hoc networks*. Pervasive and Mobile Computing, 3(1), pages 36−52, 2007.

[42] M. Abolhasan T. Wysocki and E. Dutkiewicz. *A review of routing protocols for mobile ad hoc networks*, Ad Hoc Networks, 2(1), pages 1−22, 2003.

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

[44] R. Rivest, *The MD4 Message Digest Algorithm*, MIT and RSA Data Security, Inc. IETF RFC 1320, 1992.

[45] B. Kaliski, *The MD2 Message-Digest Algorithm*, RSA Laboratories. IETF RFC 1115, 1992

[46] D. Eastlake and P. Jones, *US Secure Hash Algorithm 1 (SHA1)*, Cisco Systems. IETF RFC 3174, 2001.

[47] J. Kapp, *Test Cases for HMAC-RIPEMD160 and HMAC-RIPEMD128*, Reaper Technologies. IETF RFC 2286, 1998.

Résumé

Un réseau ad hoc est une collection de nœuds mobiles communiquant entre eux par des liaisons sans fil. Son déploiement est facile et moins coûteux. 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 répondre à ces défis, La solution la plus appropriée pour assurer des communications de groupe sécurisées est la mise en place d'un protocole de gestion de clé au sein du groupe, assurant les services de sécurité. Cela grace à la distribution et la mise à jour des clés du groupe à chaque fois qu'il y a un changement dans la composition du réseau dû aux adhésions et aux départs. Le grand handicap est que les nouvelles clés devraient être redistribuées à tous les membres du groupe, ce qui devient coûteux quand la taille du groupe augmente.

Dans ce rapport, nous nous intéressons au problème de gestion de clés dans les communications de groupes. Nous faisons une présentation détaillée du problème en montrant les exigences et les défis. Ensuite, nous présentons les solutions existantes dans la littérature. Enfin, nous présentons notre contribution basée sur le concept des cliques maximums, qui permet de réduire le coût énergitique consommé lors d'une communication multicast.
Mots clés : Réseaux mobile ad hoc, Sécurité, Communication de groupe, Gestion de clés.

Abstract

An ad hoc network is a collection of n ow mobile nodes communicating with each other through wireless links. Its deployment is easier and cheaper. This flexibility in time and space induces new challenges towards the security architecture to be implemented to ensure secure communications.for multicast address these challenges, the most appropriate solution to ensure secure group communications is the development establishment of a key management protocol within the group, providing security services. This thanks to the distribution and updating the group key whenever there is a change in the composition of the network due to accessions and separations. The biggest handicap is that the new keys should be distributed to all members of the group, which becomes expensive when the group size increases.

In this report, we address the problem of key management in group communication. We make a detailed presentation of the problem by showing the requirements and challenges. Next, we present the existing solutions in the literature. In the end, we present our contribution based on the concept of maximum clique , which allows to lessen the cost efficiency curves consumed during a multicast communication.
Keywords : Mobile ad hoc network, Security, Group communication, Key management.