Dédecace

J'ai toujours pensé faire ou offrir quelque chose à mes parents en signe de reconnaissance pour tout ce qu'ils ont consenti comme efforts, rien que pour me voir réussir, et voilà, l'occasion est venue.

A ceux qui m'ont donné la vie, symbole de beauté, et de fierté, de sagesse et de patience. A ceux qui sont la source de mon inspiration et de mon courage, à qui je dois de l'amour et de la reconnaissance.

A mes chers parents.

A la personne qui a été prés de moi, ma soutenu, encouragé, «Ma Femme».

A mes frères et soeurs.

A tous mes neveux.

A tous mes amis que j'aime.

Merci pour tout.

Remerciement

Tout d'abord, je remercie ALLAH pour la volonté, la force, la santé et la patience qu'il m'a donné afin de réaliser ce travail. Je tiens à adresser mes plus chaleureux remerciements à Tamine Karim, professeur à l'université de limoge (France) et lui exprimer toute ma reconnaissance pour son encadrement, ses conseils, son soutien constant, sa confiance et sa patience, ainsi que pour ses remarques pertinentes et ses contributions considérables tout au long de la réalisation de ce travail. J'ai eu l'honneur et le plaisir de travailler sous sa direction pendant mon stage de Magister et j'espère pouvoir continuer à travailler avec lui dans le futur. Je remercie profondément HARZRLLAH Abd elkrim, Chef du département d'informatique, pour l'interét qu'il accorde aux étudiants en leur fournissant les conditions favorables et pour ses encouragements et son soutien. Mes vifs remerciements s'adressent également à tous les enseignants de l'école doctorale qui ont contribué à sa création et sa réussite. Je remercie tout particulièrement les membres de jury qui ont accepté de juger ce travail. Je souhaite aussi exprimer ma profonde gratitude à mes chers parents et le reste de la famille pour m'avoir apporté du réconfort et pour m'avoir soutenu dans tous mes efforts. Je remercie également toute l'équipe de ma promo pour leur soutien moral, ainsi que tous mes amis. Enfin, je remercie toute personne qui, a, à des degrés divers, contribué sur le plan intellectuel, technique, moral, affectif ou encore matériel à l'achèvement de ce travail.

Résumé

Comme résultats des développement significatifs dans les ordinateurs mobiles, et les technologies des communications sans fils, les appariels mobiles ont gagné en communciation, calcul, et ressources mémoire pour être interconnecté. Par définition les réseaux mobiles ad hoc (MANETs : Mobile Ad Hoc Networks), se différencient des autres réseaux existants par le fait qu'ils ne se relient à aucune forme d'infrastructure fixe, le réseau n'a pas de stations de base, point d'accès, des serveurs à distance,...etc.Toutes les fonctionnalités du réseau sont assurées par les noeuds formant le réseau, chaque noeud joue le rôle de hôte et routeur reliant les données pour établir la connexion entre la source et la destination qui ne se trouvent pas dans la portée de transmissions l'un par rapport à l'autre. les MANETs ont montré leur importances, car ils sont meilleurs que leurs homologues les réseaux filaires dans plusieurs utilisations civile ou militaire. Cependant, la prolifération de ce paradigme de réseau dépend fortement du facteur de sécurité. Un système efficace et sécurisé de gestion de clés est un élément primordial pour fournir la plupart des services de sécurité au sein d'un réseau ad hoc. Un système de gestion de clés avec la proprieté de haute disponibilité est au centre de fournir la sécurité réseau via les mécanisme cryptographiques. De ce fait, plusieurs mécanismes basés sur la cryptographie qui résoudre les problèmes de sécurité des MANET ont un recours direct à une infrastructure de gestion de clés éfficace et sécurisée; ceci laisse les techniques de gestion de clés comme domaine de recherche ouvert dans le domaine de sécurité des réseaux ad hoc. Dans ce travail, nous proposons un nouveau schèma de gestion de clés pour les MANETs entièrement auto-organisé, pour faciliter la distribution des clés publiques dans le réseau. Dans cette solution, les noeuds échangent leurs clés publiques directement en clair, s'ils sont des voisins, d'autre part si les noeuds ne sont pas voisins (c'est-à-dire ne se retrouvent dans la portée de transmission l'un par rapport à l'autre), le noeud source envoie une requête clé à la destination à travers le maximum de chemins possibles, la destination envoie sa clé publique à la source après avoir authentifié sa requête clé, par les chemins inverses de la requête clé. De cette façon, la gestion de clés est assurée par la collaboration de tous les noeuds du réseau. De plus, notre nouveau protocole de gestion de clés fournit une flexibilité pour s'adapter au passage à large échelle de la taille du réseau. Les résultats de simulation montrent l'efficacité, la robustesse et la sécurité de notre nouveau schéma comparé aux autres techniques publiées dans la littérature.

Mots Clés : *MANET, Sécurité, Gestion de clés, PKI.*

Abstract

As a result of significant advances in mobile computing and wireless communication technology, mobile devices have gained sufficient communication, computation, and memory resources to be interconnected. By definition, mobile ad hoc networks (MANETs) differentiate themselves from existing networks by the fact that they rely on no fixed infrastructure, the network has no base stations, access points, remote servers, etc. All network functions are performed by the nodes forming the network ; each node performs the functionality of host and router, relaying data to establish connectivity between source and destination nodes not directly within each other's transmission range. MANET, have shown their significance because they are better than their wired counterparts in many civil and military uses. However, the proliferation of these networks strongly depends on the security factor. Secure key management with a high-availability feature is at the center of providing network security via cryptographic mechanisms. In fact, many cryptographic-based mechanisms that solve MANET security problems have a direct reliance on an efficient and secure key management infrastructure. This leaves key management techniques as an open research area in the ad hoc network security field. In this work, we propose a new scheme for public key management, for fully auto-organized MANET, to facility the publics keys distribution in the network. in this solution, nodes exchange their Public key directly if they are neighbors, in the other side if the nodes are not neighbors, the source node sends a request for public key to the destination node par the maximum of ways as possible, the destination sends his public key to the source node after authenticate his request, by the inverse routes of the request. In this way, key management is performed by collaboration of all the nodes of the network. In addition, our new protocol provides flexibility to scale with the network size. Simulation results show the efficiency, robustness and security of our scheme compared to other techniques in the literature.

Key words *:- MANET, Security, Key Management, PKI.*

Table des matières

Table des figures

Introduction générale

Contexte du travail

Les réseaux mobiles sans fil ont connu un très fort développement ces dernières années pour répondre à la hausse constante des besoins en mobilité. Dans un futur proche, ces technologies constitueront le socle d'environnements persuasifs (ubiquitaires) dans lesquels les utilisateurs pourront accéder à des services, communiquer, travailler avec d'autres usagers en tout lieu, à tout instant et depuis n'importe quel équipement mobile.

Les réseaux mobiles Ad Hoc (MANET) représentent une composante clé de cette évolution et leurs fondements seront inévitablement intégrés aux futures générations de réseaux sans-fil. Ces réseaux auto-organisés, peuvent être formés spontanément à partir d'un ensemble d'entités mobiles communicantes, sans nécessiter la présence d'une infrastructure fixe préexistante. Les entités mobiles qui constituent le réseau peuvent être de formes variées : ordinateurs portables, téléphones mobiles, PDA, capteurs, etc., et présentent par conséquent des capacités non homogènes en termes de communication, de puissance de calcul et de stockage. Elles sont libres de se déplacer de manière aléatoire et de s'organiser arbitrairement, si bien que la topologie du réseau est fortement dynamique dans le temps et dans l'espace. Les entités mobiles peuvent intervenir en tant que routeurs pour assurer l'acheminement des paquets entre elles par sauts successifs. Elles communiquent donc soit directement lorsqu'elles se trouvent dans un même voisinage, soit par communication multi-sauts le cas échéant en faisant appel à des nœuds intermédiaires. Cette technologie constitue donc une solution très attractive pour construire des réseaux dynamiques de terminaux mobiles qui allient la simplicité de construction à un coût relativement modeste.

L'utilisation des réseaux ad hoc est de plus en plus répandue et les applications de cette technologie sont multiples : déploiement d'un réseau en cas de sinistre majeur rendant inopérable l'infrastructure réseautique existante, déploiement d'un réseau militaire dans une zone d'opération, déploiement d'un réseau de capteurs, etc. Cependant, l'apparente simplicité du concept cache de nombreux défis scientifiques et techniques. En particulier, la sécurité dans les MANETs constitue l'un des principaux obstacles à un large déploiement de ces réseaux. En effet, la sécurité est devenue le sujet d'une activité de recherche soutenue qui vise à développer des solutions afin de fournir la protection des communications entre les nœuds dans un environnement potentiellement hostile, tout en assurant les services de base de la sécurité tels que la confidentialité, l'authentification, l'intégrité, la disponibilité et la non-répudiation.

Problématique et objectifs

Un schéma de gestion de clés (key Management) efficace et sécurisé est la partie fondamentale de toute architecture de sécurité utilisée pour assurer l'authentification et la confidentialité des communications. Ces conditions de sécurité peuvent être assurées en utilisant la cryptographie à clé publique. Ainsi, les clés sont une condition préalable pour initialiser un service réseau sécurisé. Cependant, la gestion de clés dans les MANETs est un problème fondamental qui a été abordé par plusieurs chercheurs aboutissant toutefois à des résultats assez limités. La difficulté accrue pour fournir le matériel cryptographique approprié aux entités du système est due à la conjonction de plusieurs facteurs qui s'étendent du manque de sécurité physique des nœuds au manque des relations de confiance à priori et au manque d'infrastructure et d'une autorité centrale.

Dans les réseaux basés infrastructure, les certificats sont gérés par une infrastructure à clé publique (PKI : Public Key Infrastructure) qui inclut une autorité de certification (Certification Authority) jouant le rôle d'une tierce partie de confiance (TTP : Trusted Third Party) et qui certifie l'authenticité des clés publiques. Toutefois, les modèles de PKI classiques ne sont pas transposables aux MANETs pour les raisons déjà mentionnées. Outre le problème du passage à l'échelle et de disponibilité des nœuds, il n'est pas souhaitable d'avoir une autorité de certification centralisée car cette dernière va constituer un point de vulnérabilité pour les MANETs : la sécurisation des communications ne peut plus être assurée si la CA [1] est indisponible ou compromise. La plupart des approches basée sur autorité fait usage d'une autorité en ligne en plus à l'autorité hors ligne pour fournir d'importantes fonctions de gestion de clés tels que le renouvellement de certificat. Il est clair que l'utilisation d'une autorité en ligne est problématique dans les MANETs, à la fois sous forme partielle ou totalement distribuée. La seule solution qui émerge est d'éliminer complètement toute forme d'autorité en ligne ce qui fait l'objet de notre travail qui consiste à réaliser un protocole de gestion de clés pour les MANETs entièrement auto-organisé en éliminant toute sorte de TTP [2].

Organisation du mémoire

Ce mémoire est organisé en cinq chapitres. Dans Le premier chapitre, nous définissons dans un premier temps, des concepts de base des réseaux ad hoc et dans un second temps nous parlerons les protocoles de routage dans ces réseaux. Dans le deuxième chapitre, nous allons aborder un point clé, en analysant les menaces et défis de sécurité dans les réseaux ad hoc.

Le troisième chapitre, a pour but de discuter et de synthétiser les travaux de recherche en termes de gestion de clés dans les réseaux ad hoc. Ainsi, nous présentons dans ce chapitre une taxonomie des schémas de gestion de clés dans les réseaux ad hoc, proposée dans la littérature. Notre contribution est présentée dans le chapitre quatre dans lequel nous décrivons le schéma que nous avons proposé afin d'assurer une gestion robuste et sécurisée de clés adaptée au contexte des MANET.

1. autorité de certification
2. tièrce partie de confiance

Dans le chapitre cinq, nous présentons la validation expérimentale de notre technique en donnant les mesures de performance de notre nouveau protocole à travers une simulation réalisée et implémentée à l'aide de l'environnement NS-2.

Première partie

État de l'art

Chapitre 1

Les réseaux ad hoc et la sécurité

Dans ce chapitre, nous introduisons le concept des réseaux ad hoc et leur contexte d'utilisation et décrivons brièvement les différentes architectures que nous rencontrons dans la littérature. Nous abordons par la suite les problèmes et défis de sécurité des réseaux ad hoc, nous énumérons les caractéristiques et les vulnérabilités induites par ces réseaux et présentons quelques attaques de sécurité possibles.

1.1 Introduction

Les réseaux sans fil (Wireless Networks) constituent de plus en plus une technologie émergente permettant à ses utilisateurs un accès à l'information et aux services indépendamment de leurs positions géographiques. Par ailleurs, les réseaux Mobiles Ad Hoc (MANET), caractérisés par une structure dynamique et une facilité et une rapidité de déploiement, suscitent un intérêt particulier. Comparé à un environnement statique, ce nouvel environnement permet aux unités de calcul une libre mobilité et ne pose aucune restriction sur la localisation des usagers.

Néanmoins, la sécurité dans les environnements ad hoc présente un véritable défi technologique pour avoir un large déploiement de ce type de réseau. En effet, la nature des liens sans fils et la topologie dynamique des réseaux ad hoc les rend plus vulnérables aux attaques que les réseaux filaires. De plus, l'absence d'une infrastructure élimine toute possibilité de déploiement d'une ligne de défense ou une autorité de défense centralisée. Un nœud mobile n'opère pas seulement comme un simple hôte mais aussi comme un routeur en acheminant les paquets aux autres noeuds du réseau qui ne peuvent pas être dans la même portée de transmission. Cela expose la couche réseau à plusieurs attaques. Sécuriser un réseau ad hoc revient donc à instaurer les différents services de sécurité dans ce réseau, tout en prenant en compte les caractéristiques propres à ce type de réseaux.

1.2 Les réseaux sans fil

L'évolution récente de la technologie sans fil et l'apparition des unités de calculs portables poussent aujourd'hui les chercheurs à faire des efforts afin de réaliser le but ultime « offrir l'accès

à l'information n'importe où et n'importe quand ». Dans les réseaux sans fil les ordinateurs communiquent via les ondes radio, ou au moyen de rayonnement infrarouge. Les ondes radio sont le support le plus utilisé, les fréquences disponibles se trouvent dans la bande des micro-ondes, autour de 2.4 GHz (bande ISM) et 5 GHz (bande U-NII) [67]. Les standards pour le support hertzien, au jour d'aujourd'hui, sont le IEEE 802.11 avec ses branches principales 802.11a (Wi-Fi5), 802.11b (Wi-Fi), 802.11g, Hiper-LAN de l'ETSI, et Bluetooth.

1.2.1 Architectures sans fil

Dans le cadre des réseaux sans fil, le standard IEEE 802.11 définit trois modes de fonctionnement différents :

- En mode infrastructure ou BSS (Basic Service Set), les machines (nœuds) sont en connexion à travers un point d'accès (voir la Figure 1.1 (a)), ce type de réseaux est appelé aussi réseaux cellulaires à simple saut (Single-hop) qui nécessite d'avoir une station de base fixe (ou points d'accès dans le glossaire 802.11) pour contenir un domaine de service, la zone de couverture de ce point d'accès est appelée l'ensemble de services de base BSS. Dans une telle architecture les paquets sont routés à travers la station de base.
- En mode point à point ou IBSS (Independent Basic Service Set), les nœuds mobiles peuvent communiquer directement les uns avec les autres s'ils sont mutuellement joignables. S'ils ne sont pas dans la même cellule, les paquets sont acheminés à travers la station de base (voir la Figure 1.1 (b)).
- En mode Ad Hoc, ce sont des réseaux ad hoc au vrai sens du mot, il n'existe aucune infrastructure, il y a juste des nœuds mobiles de faible puissance formant un réseau. Dans ce model il n'y a pas de station de base et à cause de la portée de transmission limitée, de multiples sauts peuvent être nécessaires pour que les nœuds communiquent à travers le réseau, les fonctions de routage sont incorporées dans chacun des nœuds.

Un réseau sans fil est beaucoup plus souple que son homologue filaire, dans la mesure où les nœuds ne sont pas connectés par des câbles et peuvent être totalement mobiles. Pourtant, un réseau sans fil est beaucoup plus vulnérable pour ce qui concerne la connectivité des nœuds et la sécurité des données.

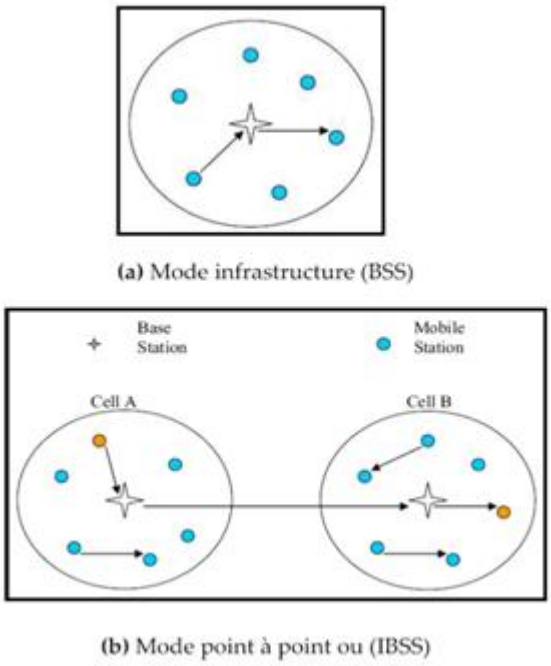


FIGURE 1.1 – Architectures des réseaux sans fil

1.2.2 Vulnérabilités des réseaux sans fil

La caractéristique la plus évidente des réseaux sans fil est que la communication a lieu sur un canal sans fil (qui est habituellement un canal radio), Ce type de technologie sans fil souffre d'un certain nombre de vulnérabilités mentionnées ci-après :

- Le canal radio peut être facilement écouté : en plaçant une antenne à un repérage approprié, un adversaire peut intercepter l'information transmise par une victime. L'écoute (eavesdropping) est souvent utilisée pour réaliser des attaques passives, ce genre d'attaque consiste à écouter le réseau et analyser les données capturées sans interaction avec ce dernier.
- Les données peuvent être altérées : un adversaire tente de modifier le contenu des messages échangés entre les entités sans fil, ce type de menaces est classé dans les attaques actives.
- L'absence d'une liaison filaire rend plus facile d'usurper les identités et de se faire passer pour des utilisateurs légitimes.
- Le canal peut être brouillé, notamment pour perpétrer une attaque de dénis de service (DoS) et ainsi empêcher les autres entités de transmettre leurs données.
- Généralement dans les réseaux sans fil, les entités qui forment le réseau se caractérisent par une capacité de stockage et puissance de calcul limitées. Ce problème est atténué en réduisant la complexité des opérations effectuées par les stations mobiles, néanmoins, cela peut mener à une dégradation de l'efficacité des protocoles de sécurité utilisés.
- Une sécurité physique limitée : cela est justifiée par le fait que le canal de communication radio est relativement vulnérable et peut être une cible facile pour espionner de manière passive.

1.3 Définition des réseaux mobiles Ad Hoc (MANET)

Un réseau mobile ad hoc « MANET » [1] peut être défini comme une collection d'entités mobiles interconnectées par une technologie sans fil formant un réseau temporaire sans l'aide de toute administration ou de toute infrastructure fixe [40]. Aucune supposition ou limitation n'est faite sur la taille du réseau cela veut dire qu'il est possible que le réseau ait une taille très grande. Les réseaux Ad Hoc sont auto organisés, ce qui implique que la connectivité doit être préservée autant que possible automatiquement lorsque la topologie du réseau change suite à l'apparition, la disparition ou au mouvement de certains nœuds.

Un réseau ad hoc peut être multi sauts dans le cas où les nœuds mobiles le constituant ne sont pas localisés dans la même portée (ne sont pas adjacents). Ainsi, le chemin entre un nœud source et un nœud destination peut impliquer plusieurs sauts où d'autres nœuds du réseau servent de relais. En effet, dans ce type de réseau, chaque nœud mobile n'opère pas seulement comme un simple hôte mais aussi comme un routeur, en acheminant les paquets aux autres nœuds du réseau qui ne peuvent pas être dans la même portée de transmission sans fil. Chaque nœud participe à un protocole de routage ad hoc qui lui permet de découvrir des chemins "multi-sauts" à travers le réseau à tout autre nœud.

1.4 Caractéristiques des réseaux mobile ad hoc

Il est important de reconnaître les propriétés ou les caractéristiques des réseaux mobiles ad hoc (MANET), puisque ces propriétés peuvent également être vues comme des contraintes rencontrées par les chercheurs lors de la conception des protocoles de sécurité pour les MANETs. Bien que ces contraintes soient détaillées dans différents articles [15] ; [32] ; [30], les protocoles de sécurité sont fréquemment publiés et n'adhèrent pas aux contraintes de conception fondamentale. Ces contraintes constituent la base des analyses du protocole. Quand un nouveau protocole ou schéma (système de gestion de clés) est publié pour les MANETs, sa faisabilité est mesurée par :

1. son degré de satisfaction des contraintes fondamentales des MANET,
2. la permission d'un compromis justifiable entre la sécurité, les besoins généraux en mémoire, le calcul et la communication.

Il faut noter que les MANETs n'adhèrent pas tous aux caractéristiques détaillées dans cette section. Les caractéristiques des MANET et leurs applications possibles sont fortement liées ; diverses applications demandent des MANET avec des variantes dans les caractéristiques données.

Par exemple, un MANET "ouvert" ou "publique" aura le caractère auto-organisé, et donc les utilisateurs finaux vont mettre en place et gérer eux-mêmes le réseau. Cela signifie qu'une autorité de confiance déconnectée (hors ligne) peut ne pas être disponible. En revanche, les MANETs utilisés dans des applications militaires n'auront pas la caractéristique d'auto-organisée mais fera

1. Mobile Ad hoc NETworks

usage d'une autorité hors ligne pour initialiser les nœuds, l'approche basée sur l'autorité offre un contrôle robuste pour l'accès aux services du réseau.

Un autre exemple de caractéristiques variables émerge de MANET formé par les nœuds de capture ou des ordinateurs portables. Il est clair que les systèmes conçus pour les MANETs formés par les ordinateurs portables n'auront pas la même limitation dans la mémoire, l'énergie (batterie) et les ressources de calcul que ceux formés par les nœuds de captures.

Il est donc évident qu'une description claire du système de gestion de clés de l'application prévue est nécessaire. L'application peut aussi dicter les caractéristiques des MANET et le degré d'influence de certaines de ces caractéristiques sur la conception du système adéquat.

1.4.1 Infrastructure réseau

Il n'y a aucune infrastructure fixe ou préexistante dans un réseau ad hoc ; toutes les fonctions du réseau (routage, sécurité, gestion de réseau, etc.) sont effectuées par les nœuds eux-mêmes et vu la portée de transmission limitée des nœuds, la diffusion des données est réalisée dans un mode multi-sauts ; les nœuds peuvent donc être considérés comme des hôtes et des routeurs.

Bien que le manque d'infrastructures ouvre de nouvelles opportunités pour les attaques, les auteurs croient que ce manque peut aider à assurer la survie du réseau dans un environnement très hostile. Cela est vrai non seulement du côté sécurité réseau mais aussi lorsque les utilisateurs du réseau sont sous attaque physique.

Les réseaux ad-hoc peuvent être formés spontanément, sans une connaissance préalable de l'emplacement physique et de l'environnement du réseau. Le manque d'infrastructures des MANET le rend ainsi approprié pour des applications diverses où les réseaux traditionnels ne répondent pas. Certains chercheurs ont déjà abordé les questions de sécurité dans les réseaux ad-hoc hybrides (Par exemple, [60]). Les réseaux ad hoc hybrides combinent l'infrastructure des réseaux conventionnels avec les multi-sauts. Ce dérivé des réseaux ad-hoc trouvera son utilité là où l'infrastructure fixe peut être étendue grâce à des réseaux multi-sauts ou lorsque la fonctionnalité (et performances) des réseaux multi-sauts peut être améliorée en s'appuyant sur d'autres infrastructures.

1.4.2 Topologie du réseau

Les nœuds dans les réseaux ad-hoc peuvent être mobiles résultant d'une topologie dynamique faiblement connectée. Dès que la mobilité des nœuds est non restrictive, la topologie peut être imprévisible. Le réseau cependant, montre les modèles de mobilité globale qui ne peuvent pas être complètement aléatoires [16].

La topologie est faiblement connectée due à des erreurs transitoires fréquentes de la connectivité sans fil. Les utilisateurs peuvent par conséquent expérimenter l'indisponibilité des services essentiels du réseau.

La mobilité des nœuds et la connectivité sans fil permettent aux nœuds de rejoindre et quitter spontanément le réseau ce qui rend le réseau amorphe. Les services de sécurité doivent être capables de mesurer de manière transparente et demeurer disponibles lors des changements de la topologie du réseau.

1.4.3 Auto-organisation

les MANETs ne peuvent être reliés à aucune forme d'administration centralisée ou de contrôle, ce qui est essentiel pour éviter les attaques qui visent un seul poste [78]. Un MANET auto-organisé ne peut s'appuyer sur aucune forme de tierce partie de confiance « Trusted Third Party » déconnectée (TTP), le réseau peut donc être initialisé par une TTP distribuée en ligne.

Un MANET pur ou totalement auto-organisé ne repose sur aucune forme de TTP [15] [16], la TTP en ligne est également éliminée. Les nœuds doivent donc avoir des dispositifs compatibles avec le logiciel installé. Dans le cas extrême, les nœuds n'auront pas à partager un ensemble commun de paramètres du système de sécurité. Le manque d'une TTP peut forcer les utilisateurs finaux à participer activement à la mise en place des associations de sécurité. les MANETs (totalement) auto-organisés ont des implications de sécurité inhérentes :

- les MANETs totalement auto-organisés sont «ouverts» de nature : de façon similaire à l'internet, tout utilisateur peut rejoindre le réseau aléatoirement. Le contrôle d'accès aux applications devrait être fourni à la couche application avec un degré d'interaction avec l'utilisateur.
- Chaque utilisateur sera lui même son propre domaine d'autorité, et donc responsable de la production et de la distribution de son propre matériel de chiffrement. Comme l'a souligné [26], un nœud peut générer plus d'une identité quand il n'y a pas de TTP déconnectée. Il est donc clair qu'il sera très difficile (voire impossible) de limiter les utilisateurs à une seule et même unique identité dans une configuration (totalement) auto-organisée.
- Le réseau sera toujours vulnérable à l'adversaire actif initié. En fait, le modèle de l'adversaire Dolev-Yao [25] est trop restrictif [11] ; par exemple, il ne parvient pas à saisir l'information alors qu'un adversaire peut gagner des connaissances détaillées sur le protocole utilisé.
- Il sera difficile de tenir des nœuds malveillants responsables de leurs actes, car ils peuvent toujours rejoindre le réseau sous une identité différente (nouvelle).

1.4.4 Ressources limitées

Les nœuds ont une capacité de calcul, mémoire et ressources énergétiques limitées, contrairement à leurs prédécesseurs les réseaux filaires. Les nœuds sont des appareils électroniques à main, de petite taille qui n'entravent pas la mobilité de l'utilisateur. Dans une tentative pour maintenir un coût faible de ces appareils, ils sont alimentés par un petit CPU, accompagnés d'une ressource mémoire limitée. Comme les appareils sont mobiles, ils fonctionnent sur batterie. Cela implique une durée de fonctionnement souvent courte avec possibilité d'une panne de courant due à l'épuisement de la batterie pendant l'exécution d'une fonction liée au réseau.

Les dispositifs peuvent avoir une bande passante et une portée de transmission limitée. Si on suppose que le développement de la technologie de circuit intégré (IC) va continuer à la suite de la loi de Moore, l'atténuation du problème des limitations de calcul et de la mémoire sera juste une question de temps. Il est peu probable que la bande passante et la portée de transmission (dans le cas de la communication par transmission radio) s'améliorent tout en respectant la

consommation d'énergie tant qu'elles sont toutes les deux dépendantes de la loi de Shannon et donc limitées [64]. Afin d'atteindre une bande passante plus haute, un ratio de signal à bruit (SNR) plus grand est exigé. Ce dernier nécessite à son tour une énergie de transmission plus élevée [64]. Une énergie de transmission plus élevée signifie l'épuisement de la batterie. Cette dernière n'évolue pas de manière significative étant donné le taux actuel de l'avancement dans la technologie de la batterie [58]. Un protocole de sécurité qui ne parvient pas à optimiser le nœud et les ressources réseau ne sera tout simplement pas adopté dans la pratique.

1.4.5 Sécurité physique pauvre

Les nœuds sont mobiles et ne peuvent donc pas être enfermés dans une pièce sécurisée ou un placard. Ces petits appareils portatifs sont facilement compromis soit perdus ou volés. Il est donc fort probable qu'un adversaire peut physiquement compromettre un ou plusieurs nœuds et effectuer tous les tests et analyses. L'adversaire peut aussi utiliser les nœuds pour attaquer les services du réseau distribué telle qu'une autorité de certification distribuée en ligne [78]. Une faible sécurité physique n'est pas appropriée aux MANET "ouverts" : les adversaires n'ont pas à capturer les nœuds physiquement pour devenir un initié ou pour effectuer des analyses sur les protocoles. La faible sécurité physique des appareils mobiles peut entraîner de graves problèmes dans les MANETs «fermés» de type militaire où les nœuds compromis physiquement peuvent être utilisés pour lancer des attaques actives d'initiés sur le réseau.

1.4.6 Support Physique Partagé

Le moyen de communication sans fil est accessible à toute entité possédant les équipements appropriés et les ressources adéquates. En conséquence, l'accès au canal ne peut pas être restreint. Les adversaires sont donc en mesure d'écouter les communications et injecter des faux messages dans le réseau sans limitation. Le canal partagé et les nœuds faibles en sécurité physique soulignent à nouveau que les mécanismes de sécurité doivent être en mesure de faire face avec le pire des cas l'adversaire initiés actifs.

1.4.7 Systèmes distribués

En considérant les propriétés ci-dessus, les nœuds dans les réseaux ad hoc ont une relation symétrique. Cela implique qu'ils sont tous égaux et doivent donc se répartir équitablement la responsabilité de fournir les fonctionnalités du réseau. Ce n'est pas seulement pour les raisons de sécurité mais aussi pour assurer la fiabilité et la disponibilité des services réseaux qui placent le même fardeau sur les ressources de calcul, de mémoire et d'énergie des participants du réseau [78] ; [15].

1.5 Les applications des réseaux ad hoc mobiles

Pour comprendre la portée de MANET et l'utilité de leurs caractéristiques uniques, les applications potentielles des réseaux ad hoc sont brièvement examinées. Les réseaux ad hoc ont

des applications dans deux grands domaines : militaire et les environnements commerciaux.

1.5.1 Applications Militaires

L'origine des réseaux qui ne se reposent sur aucune infrastructure préexistante a pu être retracée au début des années 1970 avec les projets DARPA et PRNET [65] ; [30]. L'application des réseaux ad hoc dans un environnement militaire est particulièrement attrayante en raison de leur manque d'infrastructures et de leur nature auto-organisée. Considérons les réseaux conventionnels qui s'appuient sur des infrastructures telles que les stations de base : les infrastructures introduisent des points de vulnérabilité qui peuvent être attaqués, et si éliminés, ça va briser le fonctionnement de l'ensemble du réseau. Dans les scénarios champ de bataille, une communication robuste et garantie est essentielle, avec des conséquences potentiellement fatales si elle est compromise. Les réseaux ad hoc peuvent continuer à exister même dans le cas :

- des nœuds devenus déconnectés suite à la connectivité sans fil fragile,
- des nœuds compromis ou éteints,
- des nœuds déplacés hors de portée,
- des nœuds endommagés pendant les attaques physiques contre les utilisateurs,
- d'un dysfonctionnement ou d'épuisement de la batterie des nœuds.

Des applications telles que les réseaux de capteurs [2], les systèmes de communication positionnelle [57] ; [23] et tactiques des réseaux ad hoc [40] continueront à être les forces qui guide le développement des réseaux ad hoc.La principale caractéristique des MANET de type militaire est l'utilisation d'une autorité déconnectée « offline authority ».

Dans les MANETs basés sur l'autorité, les nœuds partagent des relations préétablies initialisées par l'autorité déconnectée. La présence ou l'absence d'une relation a priori sécurisée a un impact fondamental sur la conception des stratégies de gestion de clés pour les MANETs.

1.5.2 Applications commerciales

Les applications commerciales des réseaux ad hoc incluent l'établissement de la connectivité terrain où les réseaux traditionnels tels que les réseaux cellulaires ne sont pas financièrement viable et ne peuvent pas fournir une couverture suffisante ou contourner le besoin.

Les réseaux privés ou les réseaux personnels (le cas de la téléconférence, visioconférence, communication peer-to-peer, les réunions ad hoc, ou plus généralement toutes sortes d'applications collaboratives) sont les applications possibles des réseaux ad hoc. Il est prévu que ces applications s'accélèrent dès que la flexibilité et la commodité des réseaux ad hoc auto-organisés soient pleinement appréciées et les protocoles sont mis en œuvre avec des produits disponibles dans le commerce. Par exemple, les réseaux cellulaires qui étaient autrefois considérés comme une technologie non pratique sont maintenant devenus une nécessité.

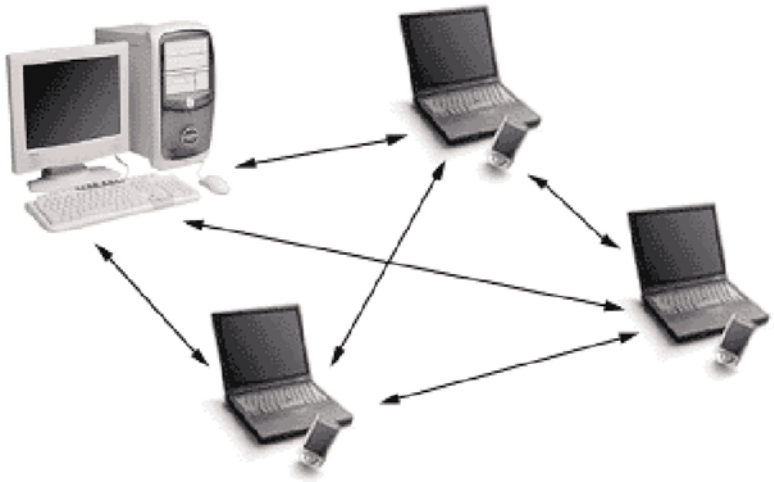
Les situations d'urgence causées par l'instabilité géopolitique, les catastrophes naturelles ou celles causées par l'être humain, pourraient endommager les infrastructures des réseaux existantes ou les rendre peu fiables. Par exemple, l'ouragan Katrina a frappé la Nouvelle-Orléans, Louisiane, le 29 août 2005. La tempête a détruit la plupart de l'infrastructure de communication

fixe qui recouvrait environ 90.000 miles carrés des États-Unis, une région presque aussi grande que le Royaume-Uni [28].

Afin de lancer une opération de secours efficace en cas catastrophe, la communication est très importante même entre un groupe localisé de secouristes. les MANETs "ouverts", permettent aux secouristes de divers pays d'établir une communication à la volée, éliminant ainsi la perte du temps pour la mise en place et la gestion des réseaux traditionnels à infrastructure fixe. La recherche et les missions de secouristes pourraient aussi être menées dans des endroits ne permettant pas l'accès aux réseaux de communication existants. Les missions de recherche et de sauvetage peuvent également être classées avec la catégorie des applications militaires.

Les réseaux ad hoc véhiculaires permettent aux véhicules se déplaçant le long d'une autoroute d'échanger des données pour la surveillance de la congestion du trafic, les communications inter véhicules et l'alerte précoce des dangers potentiels comme un accident, l'obstruction de route ou un véhicule bloqué. Plusieurs projets de recherche ont été entrepris pour traiter les réseaux ad hoc véhiculaires [51] ; [59].

1.6 Le routage dans les réseaux ad hoc



Les réseaux ad hoc sont multi-sauts. Il peut donc arriver qu'un mobile veut communiquer avec un autre qui n'est pas dans sa portée de communication directe. Les messages vont devoir être transmis de proche en proche jusqu'à la destination : c'est ce que l'on appelle "le routage". La technique la plus basique est l'inondation, où chaque mobile réémet tous les paquets qu'il reçoit pour la première fois.

Évidemment, l'inondation consomme beaucoup de ressources (bande passante et énergie) et n'est pas optimale. De nombreux protocoles de routage ont donc été proposés pour rendre les communications multi-sauts plus efficaces (moins de réémissions, chemins plus courts, etc.) que l'inondation basique.

1.6.1 Contraintes de routage dans les réseaux ad hoc

L'accomplissement de la tâche de routage, inhérente à tout réseau, est compliqué dans les réseaux ad hoc par l'utilisation de communications via radio : la radio est en effet le médium le plus hostile à la propagation de l'information, du fait notamment des interférences entre utilisateurs et de la complexité du traitement du signal. D'autre part, le routage ad hoc est aussi compliqué par la mobilité des éléments susceptibles d'acheminer le trafic (c'est-à-dire les utilisateurs eux-mêmes). N'ayant pas été prévus pour ces dernières complications, les algorithmes de routage classiques ne peuvent donc pas être utilisés tels quels. Ils doivent être optimisés (si ce n'est entièrement revus) pour être efficaces dans les réseaux ad hoc : s'adapter aux communications radio en réduisant au maximum le trafic de contrôle nécessaire au bon fonctionnement du réseau, et en même temps rester en mesure de suivre dynamiquement la mobilité des éléments du réseau. Ce sont ces contraintes qui sont au fondement des algorithmes de routage ad hoc.

1.6.2 Mécanismes de routage

Il existe une grande diversité de techniques ou encore architectures de routage ad hoc ; ainsi, les protocoles de routage dans les réseaux ad hoc peuvent être classés suivant plusieurs critères, citant par exemple : la consommation de l'énergie, le rôle de chaque nœud dans le réseau, le nombre de messages émis, ... etc.

1.6.3 Les différentes familles de protocoles de routage MANET

Dans les travaux menés à l'IETF, plusieurs familles de protocoles de routage ad hoc se sont rapidement dégagés. Chaque protocole peut ainsi être classifié en tant que réactif, proactif, ou hybride. On peut différencier ces protocoles par la méthode utilisée pour découvrir le chemin entre le nœud source et le nœud destination.

Les protocoles proactifs

La méthode proactive identifie en temps réel un sous-ensemble des liens entre les éléments du réseau qui soit suffisant pour que toute paire d'éléments du réseau soit connectée à chaque instant (éventuellement à travers plusieurs liens consécutifs). En effet, les protocoles proactifs recherchent à intervalle régulier les différentes routes disponibles dans le réseau. Leur principe de base 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 soit immédiatement connue. Les messages à portée globale ne doivent alors contenir que l'information sur les liens appartenant à ce sous-ensemble, au lieu de toute l'information sur tous les liens, ce qui constitue une économie certaine. Afin d'identifier de manière distribuée le sous-ensemble de liens essentiels pour que chaque paire d'éléments du réseau soit connectée, les algorithmes de sélection ou d'auto proclamation de relais sont réutilisés. Avec un algorithme d'auto proclamation, il est par exemple suffisant que chaque nœud relais envoie périodiquement un message à portée globale contenant l'information à propos de tous ses liens avec tous ses voisins pour qu'il soit garanti que chaque nœud du réseau possède assez d'information pour trouver un chemin jusqu'à tout autre élément dans le réseau.

Dans le contexte des réseaux ad hoc les nœuds peuvent apparaître ou disparaître de manière aléatoire et la topologie même du réseau peut être changée ; 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 fonctions 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. L'avantage premier de ce type de protocole est d'avoir les routes immédiatement disponibles quand les applications en ont besoin, mais cela se fait au coût d'échanges réguliers de messages (consommation de bande passante) qui ne sont certainement pas tous nécessaires (seules certaines routes seront utilisées par les applications en général). Le protocole OLSR est un exemple type de cette catégorie.

Les protocoles réactifs

La méthode réactive se fonde sur une économie plus poussée, et identifie en temps réel un sous-ensemble de liens suffisant pour acheminer le trafic utilisateur du moment. Les protocoles réactifs ou encore "on demande" entreprennent la recherche d'une route uniquement avant de transmettre un paquet. Leur principe est de ne rien faire tant qu'une application ne demande pas explicitement d'envoyer un paquet vers un nœud distant. Seuls les liens dont on a besoin, font l'objet d'information de routage. Pour cela, les messages à portée globale ne sont plus envoyés périodiquement, mais seulement à la demande [Percher et Jouga], lorsque du trafic utilisateur doit être acheminé vers une destination vers laquelle un chemin n'est pas connu à ce moment-là. Dans ce cas, un message à portée globale est diffusé dans tout le réseau, en tant que requête pour établir un tel chemin. La destination (ou un nœud connaissant un chemin vers la destination) recevra alors cette requête diffusée, et pourra y répondre pour établir un chemin, envoyant une réponse rebroussant le chemin pris depuis la source de la requête. Cela permet d'économiser de la bande passante et de l'énergie. Une fois que ce chemin est trouvé, il est inscrit dans la table de routage et peut être utilisé. En général, cette recherche se fait par inondation (un paquet de recherche de route est transmis de proche en proche dans tout ou partie du réseau). L'avantage majeur de cette méthode est qu'elle ne génère du trafic de contrôle que lorsqu'il est nécessaire et donc elle réduit de manière drastique la quantité d'information de routage à transmettre, si un nombre réduit de chemins sont utilisés à travers le réseau, et que les nœuds ne sont pas trop mobiles. Les principales contreparties sont que l'inondation est un mécanisme coûteux qui va faire intervenir tous les nœuds du réseau en très peu de temps et qu'il va y avoir un délai à l'établissement des routes.

Le protocole AODV est un exemple de protocole de routage ad hoc qui utilise certaines de ces techniques de réduction de la quantité d'information de routage.

Les protocoles hybrides

Les protocoles hybrides combinent les approches réactive et proactive. Le principe est de connaître notre voisinage de manière proactive jusqu'à une certaine distance (par exemple trois ou quatre sauts), et si jamais une application cherche à envoyer quelque chose à un nœud qui

n'est pas dans cette zone, d'effectuer une recherche réactive à l'extérieur. Avec ce système, on dispose immédiatement des routes dans notre voisinage proche, et lorsque la recherche doit être étendue plus loin, elle en est optimisée (un nœud qui reçoit un paquet de recherche de route réactive va tout de suite savoir si la destination est dans son propre voisinage. Si c est le cas, il va pouvoir répondre, sinon il va propager de manière optimisée la demande hors de sa zone proactive). Selon le type de trafic et les routes demandées, ce type de protocoles peut cependant combiner les désavantages des deux méthodes : échange de paquets de contrôle réguliers et inondation de l ensemble de réseau pour chercher une route vers un nœud éloigné.

1.7 Conclusion

Les réseaux ad hoc font partie d'un domaine de recherche qui n'a pris son essor que très récemment. Le facteur qui a déclenché cet intérêt fut l'arrivée de technologies relativement bon marché qui ont favorisé la conception et le déploiement de tels réseaux. Les principales normes de réseaux sans fil qui sont à l'origine de cet intérêt pour les réseaux ad hoc sont HiperLAN1 [ETS1] et HiperLAN2 [ETS2], Bluetooth [Blue] et 802.11 de l'IEEE [LAN]. Après notre étude des différents protocoles de routage qui existent, nous avons vu que ces protocoles utilisent une variété de techniques afin de résoudre le problème de routage. Une technique proactive maintient les routes fraiches périodiquement tandis qu'une technique réactive établit des découvertes de routes sur demande. Chaque technique peut donner des résultats très satisfaisants dans un réseau particulier et ne pas convenir du tout pour d'autres réseaux ad hoc. En résumé, cette étude théorique nous a permis de nous familiariser avec un environnement de routage récent. Elle présente le point de départ vers l'étude de mécanismes de sécurité appliqués dans le routage MANET.

Chapitre 2

La Sécurité dans les MANETs

La sécurité est un sujet très important à traiter, notamment pour les applications de MANET dites sensibles à la sécurité. Effectivement, les réseaux ad hoc sont connus pour leur manque d'infrastructure préexistante et d'administration centralisée, ils sont de ce fait considérés comme difficiles à sécuriser. Nous avons déjà signalé que les opérations de gestion de communication dans les MANETs sont gérées d'une manière très différente que celles des réseaux traditionnels, il en est de même pour leurs sécurité.

2.1 Les défis de sécurité

Les réseaux ad hoc constituent un nouveau paradigme dans les systèmes de communications sans fil, ils se distinguent par des propriétés et caractéristiques spécifiques comparativement aux réseaux traditionnels qui sont basés infrastructure.

Ces caractéristiques et propriétés amènent aussi de nombreux défis de gestion de la sécurité. Les caractéristiques principales des MANET incluent une topologie dynamique, une rupture de liens fréquente et une perte de données considérables, des contraintes d'énergie, une bande passante et portée de transmission limitées, une protection physique des nœuds relativement faible, une coopération distribuée pour les communications multi saut et l'absence d'une autorité centrale. Avec ces caractéristiques particulières, les réseaux ad hoc sont exposés au risque de violations de la sécurité. Nous présentons ci-après les thèmes critiques qui empêchent les solutions traditionnelles de sécurité d'être directement appliquées aux MANET.

- Un réseau ad hoc est dynamique en raison des changements fréquents de sa topologie et de ses membres. Il est ainsi indispensable que les mécanismes de sécurité puissent s'adapter rapidement à ces changements. La topologie dynamique du réseau aboutit à un changement significatif dans les relations de confiance entre les nœuds. Ainsi, l'implémentation de sécurité basée sur la confiance fait face à de grands défis et les solutions statiques ne s'appliquent pas dans un tel environnement dynamique.
- La mobilité des nœuds peut également causer la rupture de liens et la perte fréquente de données, puisque les nœuds peuvent joindre et quitter le réseau sans aucun avertissement préalable. Ainsi les liaisons entre les nœuds ne seront pas garanties, ce qui peut avoir un impact important sur la communication et qui affecte aussi la mise en œuvre de mécanismes de sécurité.
- Les nœuds mobiles d'un réseau ad hoc incluent typiquement des équipements portables tels que des « laptops », des PDA, des téléphones cellulaires et des micro capteurs. Ces dispositifs sont inévitablement alimentés par batterie. La durée de vie de la batterie devient, de ce fait, cruciale pour la communication sans fil et le calcul distribué. Cela peut être une cible des nœuds malicieux qui tentent de bloquer les communications et les services via des attaques par consommation de batteries. En outre, les solutions de sécurité traditionnelles sont conçues pour une forte capacité des nœuds ce qui ne s'adapte pas aux environnements ad hoc. Ceux-ci exigent donc de nouvelles solutions qui soient économes en termes de puissance de calcul, du coût de l'énergie et de la charge du trafic. De plus, la solution doit être équitable au niveau de la consommation des ressources du réseau.
- Les nœuds ad hoc ont une protection physique relativement limitée. Il y a donc une probabilité non négligeable que ces derniers soient compromis. Ainsi il ne faut pas considérer seulement les attaques provenant de l'extérieur mais aussi de l'intérieur des nœuds compromis. Ces attaques ne peuvent donc pas être prévenues avec de simples mécanismes d'authentification et les schémas de clé seuls ne sont pas suffisants pour y faire face. Par conséquent, des mécanismes plus sophistiqués doivent être conçus. Ainsi, il est très important de considérer des solutions de sécurité spécifiques aux MANET contre les nœuds compromis, car certaines attaques très sophistiquées peuvent être commises par conspiration des nœuds compromis.

- Le manque d'organisation et de confiance a priori peut également influencer la sécurité des MANET. En effet, lors du déploiement du réseau les nœuds s'auto-organisent et il n'y a pas une confiance a priori entre ces nœuds. De plus, la gestion et la distribution de clés peuvent être difficiles à réaliser en raison de l'absence d'une autorité centrale. Les solutions de sécurité dans les réseaux traditionnels s'appuient généralement sur des relations de confiance établies par une tierce autorité de confiance. Elles utilisent des primitives cryptographiques pour authentifier les nœuds et sécuriser les échanges de données. Afin d'utiliser ces moyens dans les MANETs, nous devons étudier comment établir des autorités de confiance et/ou des relations de confiance entre les nœuds sans l'aide d'aucune infrastructure ou autorité centrale.
- Exigence de coopération, en raison du manque d'une infrastructure les fonctions de base de gestion du réseau doivent être effectuées d'une façon distribuée par la collaboration d'un ensemble de nœuds ordinaires. Ainsi, l'exécution des opérations de base du réseau peut être fortement affectée par un manque malveillant ou accidentel de coopération.

En raison de ces défis de sécurité, les réseaux ad hoc imposent de nouveaux besoins de sécurité et exigent des mécanismes robustes, tolérants aux pannes et ayant un faible surcoût. De plus, un réseau ad hoc peut se composer de centaines voire de milliers de nœuds, il est donc important que la solution de sécurité proposée permette le passage à l'échelle.

2.2 Les vulnérabilités

Tout système peut avoir des faiblesses qui sont dues à une mauvaise conception ou à certaines caractéristiques liées à la nature même de ce système. L'exploitation de ces faiblesses et les capacités de l'intrus présentent, évidement, une menace contre ce système. Il est donc évident qu'avant de proposer toute solution de sécurité, il est primordial d'identifier d'abord les vulnérabilités et les failles pouvant être exploitées pour lancer une éventuelle attaque contre le système.

Parmi les vulnérabilités intrinsèques des réseaux ad hoc, certaines résident dans leur mécanisme de routage, d'autres dans l'utilisation des liens sans fil et d'autres encore dans leurs mécanismes d'auto-configuration. Ces fonctionnalités de base reposent sur une confiance complète entre tous les nœuds participants au réseau. Dans le cas du routage, le transport des paquets dans le réseau repose sur la véracité des informations fournies par les autres nœuds. De plus, la livraison d'un paquet vers une destination dans les MANETs repose sur un routage multi-sauts, et donc nécessite la coopération totale des nœuds intermédiaires. Un nœud malicieux pourrait, en refusant de coopérer, tout simplement bloquer, modifier, ou ralentir le trafic qui passe par lui.

L'utilisation des liaisons sans fil rend ces réseaux très vulnérables aux attaques s'étendant de l'écoute clandestine passive à l'interférence active. Un attaquant doit juste être dans la portée radio d'un nœud afin d'intercepter le trafic réseau.

Le manque de frontières sécurisées dans les MANETs représente une autre vulnérabilité. En effet, il n'y a pas une ligne de défense claire comparativement aux réseaux traditionnels où l'intrus doit passer par plusieurs lignes de défense tels que les pare-feux pour visiter le réseau.

Cependant, dans les MANETs l'intrus n'aura pas besoin d'obtenir un accès physique pour visiter le réseau, il peut communiquer avec n'importe quel nœud dans sa portée de transmission et ainsi rejoindre le réseau automatiquement. Ces réseaux sont donc vulnérables par le fait d'être physiquement accessibles, et les nœuds peuvent être facilement compromis, ce qui peut avoir comme conséquence la révélation des clés cryptographiques. Un autre problème pour protéger les MANETs est la difficulté de distinguer entre des nœuds compromis et des nœuds saints à partir de leurs comportements. Par exemple, un nœud égoïste ne peut pas être considéré comme nœud malicieux ou compromis, car ce nœud n'effectue aucune attaque mais il est possible qu'il refuse de coopérer pour préserver son énergie électrique.

De plus, l'absence d'une administration centralisée peut conduire à un état de vulnérabilité. En effet, l'établissement de confiance entre les nœuds, la gestion et la distribution des clés cryptographiques est une tâche critique sans aucune administration centralisée et la plupart des modèles de confiance utilisés dans les réseaux traditionnels sont basés sur l'existence d'une infrastructure et d'une administration centralisée. L'application directe de ces modèles dans les MANETs peut générer des vulnérabilités considérables. La bande passante limitée peut être également un point de vulnérabilité et une cible à des attaques malveillantes telles que le déni de service (DoS), un nœud malicieux peut envoyer massivement des requêtes afin de consommer la bande passante et de rendre les services indisponibles aux autres nœuds. Outre les contraintes de bande passante, la portée de transmission des nœuds est relativement faible. Ainsi, de multiples sauts à travers des nœuds intermédiaires sont nécessaires pour faire passer les messages. Cette coopération distribuée est basée sur l'hypothèse que les nœuds intermédiaires sont dignes de confiance et n'ont pas un comportement malveillant. Toutefois, cette hypothèse n'est pas fiable, car il n'y a aucune garantie que ces nœuds se comporteront comme prévu.

Pour récapituler, un réseau ad hoc est vulnérable en raison de ses caractéristiques de milieu ouvert, des changements dynamiques de la topologie, des algorithmes basés sur la coopération, du manque de point centralisé de contrôle et de gestion et du manque d'une ligne de défense claire.

2.3 Les menaces de sécurité

Une approche pragmatique pour établir un système de sécurité est de considérer les menaces auxquelles le système pourrait faire face après déploiement. Une classification des menaces dans les réseaux ad hoc est suggérée dans [68] selon laquelle on distingue deux types de menaces :

- Les menaces externes, où l'attaquant est limité à l'écoute et l'analyse du trafic échangé. De plus l'attaquant vise à provoquer la congestion ou la perturbation des nœuds durant la prestation de services.
- Les menaces internes où l'attaquant se donne les moyens pour gagner l'accès normal au réseau et participer aux activités de ce dernier, soit par une usurpation d'identité pour obtenir l'accès au réseau comme nouveau nœud, ou de compromettre directement un nœud et l'utiliser comme base pour mener des attaques malveillantes.

2.3.1 Les menaces externes

En présence d'un protocole d'authentification pour protéger les couches supérieures du réseau, des menaces externes sont dirigées aux couches liaison de données et physique. La sécurité de la couche physique est intrinsèquement difficile à fournir en raison de la nature mobile des nœuds ad hoc. Nous divisons les menaces externes en deux grandes catégories :

1. L'écoute clandestine passive, où l'adversaire écoute simplement les signaux transmis et
2. L'interférence active, où l'adversaire envoie des signaux ou des données destinés à perturber le réseau d'une certaine manière.

L'écoute clandestine passive

Ceci peut permettre aux nœuds non autorisés d'écouter et de recevoir des messages comprenant des informations de routage ou même des clés cryptographiques. Les nœuds non autorisés sont en mesure de recueillir des données qui peuvent être utilisées pour déduire la topologie du réseau, ou des informations sur les clés cryptographiques qui seront utilisées pour faire de la cryptanalyse. Par conséquent, il est nécessaire de développer des techniques pour cacher ces informations. L'écoute clandestine est également une menace pour la confidentialité de la localisation d'un nœud.

L'interférence active

La menace principale dans cette catégorie est l'attaque de déni de service (DoS) provoquée, en bloquant la voie de transmission sans fil, ou en déformant les communications. L'effet de cette attaque dépend de sa durée, et le protocole de routage utilisé ainsi que la robustesse des services de sécurité. Le type le plus sérieux de l'attaque DoS s'appelle « sleep deprivation » [25]. Dans cette attaque, l'énergie des nœuds est délibérément gaspillée. Avec la puissance et les ressources limitées des nœuds ad hoc, la prévention de telles attaques est de la plus grande importance. Il y a également des menaces pour l'intégrité, par exemple où un attaquant externe peut essayer de rejouer d'anciens messages, ou de changer l'ordre des messages.

2.3.2 Les menaces internes

Les menaces posées par les nœuds internes sont très graves, car ces derniers internes disposent de l'information nécessaire pour participer à des opérations distribuées. Les nœuds internes peuvent mal se comporter dans une variété de façons ; nous identifions quatre catégories de comportements malveillants des nœuds : les nœuds échoués, les nœuds égoïstes, les nœuds compromis et les nœuds malicieux.

Les nœuds échoués

Les nœuds échoués sont simplement ceux incapables d'effectuer une opération ; ceci peut être dû à plusieurs facteurs, comprenant des problèmes de manque d'énergie et ceux liés aux événements environnementaux. Un défi majeur réside dans le routage ad hoc où certains nœuds

échouent à acheminer des paquets de données ou même des messages de routage afin de mettre à jour les structures de donnés. Cette information est cruciale et peut affecter les mécanismes de sécurité mis en place, telle que l'authentification. La menace d'avoir des nœuds échoués est plus sérieuse si ces derniers sont nécessaires en tant qu'élément participant d'une manière distribuée à un service de sécurité.

Les nœuds compromis

Les nœuds compromis possèdent les mêmes caractéristiques que les nœuds échoués, comme la non-transmission ou le non-acheminement des paquets de données reçus. En plus, ils peuvent également envoyer de fausses informations pour perturber certaines fonctions de base du réseau comme le routage, ce qui présente une menace à l'intégrité du réseau. Les nœuds compromis peuvent aussi divulguer des informations confidentielles comme par exemple des clés privées, ou conspirer pour lancer une attaque active. Les mécanismes de sécurité proposés pour les environnements ad hoc doivent être en mesure d'identifier et d'isoler ces nœuds compromis.

Les nœuds égoïstes

Dans certains cas, des nœuds ad hoc présentent un comportement égoïste, ils peuvent refuser de transférer des données ou de coopérer avec d'autres nœuds pour garantir les opérations et les services de base du réseau. Les nœuds égoïstes se comportent ainsi pour des raisons de performances ou pour préserver de l'énergie à cause de la capacité limitée des batteries et du manque de ressources dans les MANETs. Il est important de préciser que les nœuds égoïstes n'effectuent aucune action pour compromettre l'intégrité du réseau par injection active d'information de contrôle erronée.

Les nœuds malicieux

Les nœuds malicieux visent à perturber délibérément le fonctionnement correct des différentes opérations de base de gestion du réseau tels que le routage, compromettre les nœuds du réseau ou rendre indisponible les services du réseau si possible. Ils peuvent lancer différentes attaques sur les mécanismes de base du réseau ou même sur les mécanismes de sécurité. L'impact des actions des nœuds malicieux devient plus important lorsque ces nœuds sont le seul lien entre des groupes de nœuds voisins.

2.4 Les attaques sur les MANETs

Les attaques contre un réseau ad hoc peuvent venir de nœuds malicieux qui ne sont pas une partie valide du réseau et qui tentent de joindre le réseau sans autorisation. Ces nœuds sont généralement appelés des nœuds externes. Les réseaux sont généralement protégés contre les nœuds malicieux grâce à l'utilisation de techniques cryptographiques. Ces techniques permettent aux nœuds de vérifier d'une manière sécurisée l'identité des autres nœuds, et peuvent donc essayer de prévenir tout dommage causé par les nœuds malicieux. Nous considérons également

des attaques à partir des nœuds qui sont autorisés pour faire partie du réseau, on les appelle nœuds internes. Des nœuds internes peuvent lancer des attaques parce qu'ils ont été compromis par des nœuds malicieux.

Le succès d'une attaque dépend de la vulnérabilité du système et l'efficacité des contre-mesures. Les attaques peuvent être divisées en deux catégories principales :

- Les attaques passives. Dans ce type d'attaques, un adversaire écoute passivement le trafic échangé sans modifier ou insérer des données. Ces attaques visent principalement la confidentialité du système. Toutefois, ce processus de collecte d'informations peut conduire à des attaques actives plus tard.
- Les attaques actives. Sont des attaques où l'adversaire effectue une action malveillante en plus d'écouter passivement le trafic. Par exemple un adversaire pourrait choisir de modifier des paquets, d'injecter des paquets, ou même de perturber des services réseau.

Une attaque qu'elle soit passive ou active peut être classée dans l'une des deux catégories suivantes :

- Attaques sur les mécanismes de base de gestion du réseau tel que le routage. La prévention de ces attaques nécessite des mécanismes de sécurité qui reposent souvent sur des algorithmes cryptographiques.
- Attaques sur les mécanismes de sécurité notamment les mécanismes de gestion de clés. En raison des particularités des MANET, les solutions, nous verrons dans le reste de ce document, exigent une attention particulière.

Dans ce qui suit, nous dressons la liste d'attaques les plus probables contre les réseaux ad hoc.

2.4.1 Attaques de déni de services (DoS)

Les attaques DoS constituent une menace sévère de sécurité dans n'importe quel système distribué. Dans les MANETs ces attaques visent à mettre en péril la disponibilité des nœuds du réseau et plus particulièrement des entités qui jouent le rôle de serveurs ou de contrôleurs au sein du réseau ad hoc. Les attaques DoS sont possibles à diverses couches, à savoir, couche physique, couche MAC, couche réseau et également sur les applications s'exécutant dans de tels réseaux.

Les modèles de dénis de services qui suivent s'appliquent plus particulièrement dans le cas de réseau ad hoc [69] :

- Brouillage du canal radio pour empêcher toute communication.
- Tentative de débordement des tables de routage des nœuds servant de relais.
- La non coopération des nœuds au bon fonctionnement du réseau pour préserver de l'énergie. Cette attaque est connue sous le nom de « Selfishness » et peut être détectée grâce à des mécanismes de réputation et de détection des comportements égoïstes.
- Tentative de gaspillage de l'énergie des nœuds ayant une autonomie de batterie faible. Cette attaque est connue sous le nom de « Sleep Deprivation » et 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.
- Attaques sur les mécanismes de sécurité eux même.

2.4.2 L'attaque de consommation de ressources (Sleep Deprivation)

Habituellement cette attaque [25] est possible seulement dans les réseaux ad hoc lorsque la durée de vie de la batterie est un paramètre critique. Les nœuds fonctionnant à batterie essayent de conserver l'énergie en transmettant seulement si absolument nécessaire. Dans cette attaque, un nœud malicieux agit sur un nœud cible avec l'intention d'épuiser la batterie de ce nœud victime. Par exemple, un adversaire peut essayer de consommer la puissance de batterie d'un nœud en lui envoyant des requêtes de routage, ou en expédiant des paquets inutiles à ce nœud, ou en falsifiant le routage pour acheminer une quantité massive de trafic vers ce nœud.

2.4.3 Attaque par usurpation de l'identité d'un nœud

Cette attaque appelée aussi Spoofing constitue une menace sévère de sécurité dans les MANETs. Si une authentification appropriée des parties n'est pas pris en charge, un nœud malicieux peut se faire passer pour un nœud légitime auprès des autres nœuds afin de récupérer illégalement des informations confidentielles ou d'injecter des messages erronés dans le réseau. Une partie malveillante peut avoir sa clé publique certifiée même sans qualifications appropriées. Ce type d'attaque met en péril l'authentification et le contrôle d'accès des membres du réseau ad hoc.

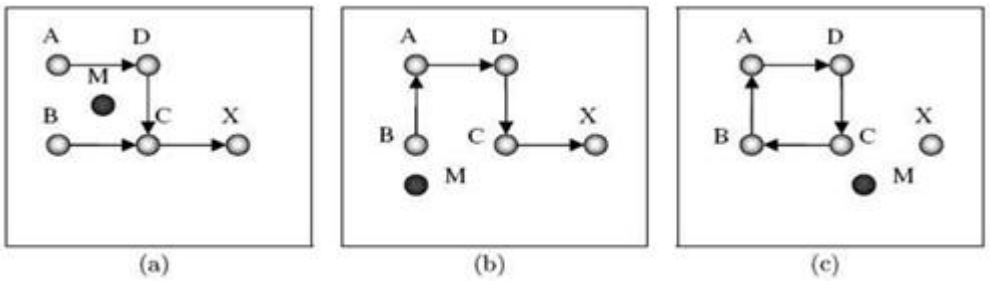


FIGURE 2.1 – Une topologie en boucle formée par une attaque de spoofing par le nœud malicieux M.

La Figure 2.1 illustre un exemple d'une attaque de spoofing qui a pour conséquence de créer une topologie en boucle dans un réseau ad hoc. On suppose qu'il y a un chemin entre les quatre nœuds et la destination X, dans un processus de découverte de route, le nœud malicieux M peut falsifier ces chemins comme suit : M se fait passer pour A ensuite il se rapproche de B, il envoie à B un RREP qui contient un saut vers X ce qui est moins de celui envoyé par C, B change donc sa route vers X en traversant A comme c'est illustré dans la Figure 2.1 (b). Ensuite, M se fait passer pour B et se déplace pour se rapprocher de C et lui envoyer un RREP contenant un nombre de sauts vers X inférieur à celui envoyé par X lui-même. C change alors sa route vers B. À ce point une boucle est crée et le nœud X est isolé.

2.4.4 Les attaques passives d'écoute clandestine et d'analyse de trafic « Men In The Middle »

Ces attaques appelées aussi « sniffing », consistent à écouter le réseau dans lequel transitent des paquets de données. Ces données, qui peuvent être confidentielles, sont récupérées à la volée et de manière illégale. Les attaques d'écoute et d'analyse de trafic sont plus dangereuses dans les environnements ad hoc. En effet, les ondes radio ont intrinsèquement une grande capacité à se propager dans toutes les directions avec une portée relativement grande, facilitant ainsi à une personne non autorisée d'écouter le réseau, décoder les messages, envoyés et obtenir des informations sensibles (Figure 2.2). Ces attaques constituent une menace certaine pour la confidentialité des données ainsi que pour l'anonymat des utilisateurs. L'émetteur n'est pas en mesure de détecter que la transmission a été interceptée. Toutefois, cette attaque peut être évitée en employant un schéma cryptographique pour protéger les données transmises. Cela, néanmoins, nécessite des stratégies efficaces de distribution de clés pour qu'elles soient transmises à tous les nœuds d'une manière sécurisée.

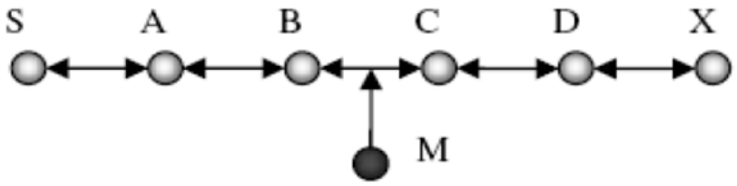


FIGURE 2.2 – Un simple réseau ad hoc avec un nœud malicieux M

2.4.5 Les attaques sur le mécanisme de routage

Le routage est l'une des fonctions les plus critiques dans les MANETs, il est également l'une des principales cibles des adversaires. Dans les MANETs, les attaques contre le routage sont généralement classées en deux catégories : attaques sur les protocoles de routage et attaques sur l'expédition/livraison de paquets [70]. Les attaques sur les protocoles visent à bloquer la propagation des informations de contrôle de routage. Les attaques sur la transmission de paquets essayent de perturber la livraison de paquets le long des chemins de communication.

Nous présentons ci-dessous quelques attaques visant à compromettre le mécanisme de routage :

- ***L'attaque Black hole***. Dans cette attaque, un nœud malicieux utilise un protocole de routage pour annoncer aux autres qu'il est le nœud ayant le plus court chemin vers la destination [31]. Le nœud malicieux recevra donc le trafic envoyé vers la destination et peut donc choisir de tuer ce trafic pour réaliser un déni de service.
- ***Packet Dropping***. Un nœud malicieux peut tuer des paquets contenant des informations de contrôle de routage destinés aux autres nœuds valides du réseau.
- ***L'attaque wormhole***. L'attaque du trou de vers (wormhole) [71] consiste à capturer

le trafic à partir d'un point du réseau et le rejouer à partir d'un autre point distant. Dans cette attaque, un nœud malicieux envoie le trafic capturé à un autre nœud malicieux situé à plusieurs sauts, ce dernier nœud injecte de nouveau le trafic reçu dans le réseau créant ainsi un tunnel de trou de ver entre les deux nœuds malicieux pour faire croire aux autres nœuds que les deux extrémités du tunnel sont voisine. La sévérité de l'attaque de trou de ver vient du fait qu'il est difficile de la détecter, et elle est efficace même dans un réseau où la confidentialité, l'intégrité, l'authentification et la non-répudiation sont préservées.

- ***Rushing Attack***. Cette attaque affecte les protocoles de routage réactifs, dans ce type de routage un nœud ne transmet que le premier message de type Route Request reçu et rejette le reste. L'adversaire peut exploiter cette caractéristique du routage réactif, il transmet rapidement toutes les Route Request. En conséquence, les nœuds qui reçoivent ces requêtes précipitées, les retransmettent et rejettent les autres Route Request qui arrivent bien après. En résultat, les routes vont passer par l'adversaire, ce qui le met dans une position avantageuse.

2.4.6 L'attaque Sybil

Dans cette attaque [9] un nœud malicieux pratique une usurpation d'identité multiple. Les identités multiples peuvent être obtenues soit par l'usurpation de l'identité d'autres nœuds, soit en faisant usage de fausses identités. Ces identités peuvent être utilisées simultanément ou étalé sur une période de temps. Cette attaque peut affecter plusieurs services dans un réseau ad hoc. Par exemple elle peut affecter le routage par trajets multiples, où un ensemble de chemins disjoints peuvent tous passer par le même nœud malveillant qui emploie plusieurs identités Sybil. Les mécanismes d'attribution équitable de ressource seront également affectés puisqu'un nœud malicieux peut réclamer plus que sa part en employant les diverses identités de Sybil. Les mécanismes basés sur la confiance peuvent également être touchés par cette attaque. Une approche simple pour détecter les identités de Sybil pourrait être de fournir un certificat de clé publique à chacune des identités. Le problème, cependant, est le besoin d'autorité centrale qui distribue les certificats.

2.4.7 Attaque par chantage

Elle est connue sous le nom anglais de Blackmail attack. Un nœud malicieux fait annoncer qu'un autre nœud légitime est malicieux pour éliminer ce dernier du réseau. Si le nœud malicieux arrive à attaquer un nombre important de nœuds, il pourra perturber le fonctionnement du réseau.

2.4.8 Les attaques physiques d'un nœud valide du réseau

Ces attaques compromettent des nœuds valides du réseau (destruction, altération ou changement physique d'un composant) et s'avèrent être des attaques particulièrement dangereuses.

2.5 Conclusion

Les réseaux ad hoc offrent une grande flexibilité de déploiement et sont plus en plus utilisés. Ces réseaux représentent également un environnement hostile qui apporte plusieurs défis de sécurité, dus à leurs caractéristiques et spécificités (liens sans fil, capacité de calcul limitée, absence d'infrastructure, etc.). Dans ce chapitre, nous avons d'abord décrit brièvement les réseaux ad hoc et leurs caractéristiques particulières, ensuite, nous avons présenté les défis majeurs en sécurité relevés dans ce domaine. Nous avons aussi présenté les vulnérabilités des réseaux ad hoc induites par leurs caractéristiques ainsi que les attaques potentielles. Ces dernières ciblent principalement les fonctions du réseau tel que le routage et les mécanismes de sécurité. Il est clair que les mécanismes de sécurité utilisés dans les réseaux traditionnels ne peuvent pas être directement appliqués aux réseaux ad hoc vu de la différence architecturale qui existe entre ces deux environnements. Les réseaux ad hoc nécessitent donc le développement de nouveaux mécanismes robustes, qui s'adaptent à leur nature et tiennent compte de leurs différentes caractéristiques et leurs vulnérabilités. Dans le chapitre suivant, nous présentons les mécanismes et solutions de sécurité des réseaux ad hoc proposés dans la littérature.

Chapitre 3

Gestion de clés dans les réseaux mobiles ad hoc

En guise d'introduction à la gestion de clés, nous allons examiner brièvement la classification des problèmes de sécurité dans les MANETs. L'objectif est de positionner le problème de la gestion de clés peer-to-peer dans le domaine de la sécurité des MANET. La principale observation est que les techniques cryptographiques sont souvent au centre de la résolution des problèmes de sécurité et de là ont besoin de la gestion de clés [78] ; [36].

Avant d'introduire les différents groupes de protocoles, nous allons aborder en outre ce qu'on entend par gestion de clés et fournir des définitions et terminologies pour les différentes propriétés et exigences des protocoles (schèmas) de gestion de clés.

3.1 Motivation pour la gestion de clés dans les réseaux mobiles ad hoc

Malgré l'évolution des MANET au cours de la dernière décennie, il y a encore un certain nombre de problèmes liés à la sécurité qui sont ouverts [78] ; [36]. Cela signifie que, bien que des solutions aient été proposées, aucune ne semble satisfaire toutes les contraintes de MANET. La Figure 3.1 illustre les domaines étudiés au sein des MANET avec un accent sur les questions de sécurité. Cette liste met en évidence les principaux domaines de la sécurité du réseau ad hoc et pourrait être élargie.

Comme l'illustre la Figure 3.1, la recherche dans le domaine de la sécurité des MANET s'occupe d'une variété d'aspects différents. Les chercheurs dans le domaine de la sécurité des réseaux ad hoc se sont initialement basés sur la sécurité des protocoles de routage [78]. L'objectif de ces protocoles (Ariadne , SEAD [34], l'ARAN [22], SRP [30]) est double :

– Fournir un mécanisme de routage robuste contre la topologie dynamique des MANET.

– Fournir un mécanisme de routage qui offre une protection contre les nœuds malveillants.

Les protocoles de routage peuvent utiliser différents mécanismes de sécurité pour atténuer les attaques sur l'infrastructure de routage.

Certains de ces mécanismes sont :

- L'exploitation de la redondance,
- La diversité du codage,
- La découverte du chemin à la demande,
- Les techniques d'entretien du chemin,
- Les mécanismes de tolérance des défauts ou d'intrusions,
- et les mécanismes cryptographiques.

FIGURE 3.1 – Les dimensions des réseaux ad hoc [29].

Par exemple, les régimes de routage peuvent exploiter la redondance en créant de multiples itinéraires de la source à la destination (facilement réalisé par ZRP [31], DSR [37], TORA [53] et AODV [55]) [78]. En envoyant des données via toutes ces routes, la redondance fera en sorte que toutes les données arrivent à destination. Un mécanisme alternatif à l'envoi des données via les routes redondantes serait la diversité du codage [4].

La diversité du codage prend avantage des routes redondantes dans la bande passante en ne retransmettant pas les messages. Elle transmet les informations de manière redondante uniquement pour la détection et la correction des erreurs. Tous ces mécanismes ont des degrés d'efficacité divers. Il est largement reconnu que les mécanismes cryptographiques peuvent fournir les techniques les plus puissantes pour assurer la disponibilité, l'intégrité et la confidentialité des informations de routage [36]. Cette observation demeure également vraie pour la plupart des autres problèmes de sécurité mis en évidence dans la Figure 3.1 [36]. Si les mécanismes de mise en réseau de base sont pris en considération, l'identification des menaces révèle que les techniques cryptographiques peuvent aussi être utilisées pour atténuer les attaques qui exploitent les communications over-the-air, les mécanismes d'accès au canal et la découverte des voisins [36] La gestion sécurisée des clés avec une fonctionnalité de haute disponibilité offre la sécurité réseau via des mécanismes cryptographiques [49]. Cependant, la plupart des protocoles de routage et les mécanismes de base liés aux réseaux néglige la tâche cruciale de la gestion sécurisée des clés et assume la préexistence et le pré-partagé des clés secrètes et/ou des paires de clés privées ou publiques [78]. En fait, beaucoup de mécanismes basés sur la cryptographie qui résolvent les problèmes de sécurité des MANET ont un recours direct à une infrastructure de gestion de clés efficace et sécurisée.

3.2 Définition de la gestion de clés

Une relation de chiffrement est l'état où les nœuds du réseau partagent le matériel de chiffrement pour une utilisation dans les mécanismes cryptographiques [49]. Le matériel de chiffrement peut inclure les paires de clés publiques/privés, les clés secrètes (cryptographie symétrique), les paramètres d'initialisation et les paramètres non secrets soutenant la gestion de clés dans les différentes instances. La gestion de clés peut être définie comme un ensemble de techniques et de procédures qui supportent la création et la maintenance des relations de chiffrement entre les parties autorisées [49].

En résumé, la gestion de clés intègre des techniques et des procédures pour établir un service supportant [49] :

1. l'initialisation des utilisateurs du système au sein du réseau,
2. la génération, distribution et installation du matériel de chiffrement,
3. le contrôle de l'utilisation du matériel de chiffrement,
4. la mise à jour, la révocation, la destruction du matériel de chiffrement,
5. le stockage, la sauvegarde/la restauration et l'archivage du matériel de chiffrement,
6. l'amorçage et la maintenance du matériel de chiffrement.

L'authentification est la base de la communication sécurisée. Sans un solide mécanisme d'authentification en place, les objectifs de sécurité restants (la confidentialité, l'intégrité des données et le non reniement) ne sont dans la plupart des cas pas réalisables. Pour être associée à une identité, l'authentification est réalisée au moyen de la vérification de quelque chose de connu. Dans le domaine électronique, le propriétaire de l'identité doit avoir un secret publiquement vérifiable associé à son identité, sinon, le nœud peut être usurpé.

L'authentification en général, dépend du contexte d'utilisation [49]. La gestion de clés se préoccupe de l'authenticité des identités associées aux six services cités ci-dessus, c'est un concept qui peut sembler banal au premier abord, mais qui n'est pas facilement atteint. L'authentification des utilisateurs est particulièrement difficile (et impossible dans la plupart des configurations réseaux) sans l'aide d'une autorité de confiance. La fonction fondamentale des systèmes de gestion de clés est la mise en place du matériel de chiffrement. La gestion de clés peut être subdivisée en l'accord de clés et le transport de clés [49].

L'accord des clés permet à deux ou plusieurs parties de dériver le matériel de chiffrement partagé comme une fonction d'information associée à chacun des protocoles participants tel que aucun parti ne peut prédéterminer la valeur résultante [49].

Dans les protocoles de transport des clés, l'une des parties crée ou obtient autrement le matériel de chiffrement et le transfert en toute sécurité à l'autre partie ou aux autres parties [49]. Les deux protocoles, accord de clés et transport de clé peuvent être réalisés en utilisant soit des techniques symétriques ou asymétriques. Un système d'établissement des clés hybride fait usage des deux techniques symétriques et asymétriques dans une tentative d'exploiter les avantages de ces deux techniques [49].

3.3 Exigences des systèmes de gestion de clés

Les services de gestion de clés doivent respecter les caractéristiques de sécurité suivantes :

3.3.1 La confidentialité

Les méthodes de gestion de clés doivent garantir le secret de la clé, qui est d'assurer l'incapacité des adversaires ou des parties non autorisées à connaitre le matériel de chiffrement (ou même une partie du matériel de chiffrement).

3.3.2 L'authentification par clé

C'est la propriété selon laquelle seules les entités de communications identifiées et authentifiées peuvent avoir accès aux matériels de chiffrement cryptographiques. L'authentification par clé dans le cadre d'une session de communication entre deux parties peut être soit unilatérale ou mutuelle :

- unilatérale : signifie l'authentification du matériel de chiffrement d'une seule partie,
- mutuelle : implique le matériel de chiffrement des deux parties.

La possession de clé est en fait indépendante de l'authentification par clé. L'authentification par clé, sans savoir si le destinataire a effectivement la bonne clé, est désignée comme l'authentification par clé implicite.

3.3.3 Confirmation de clés

Si la confirmation de clés est fournie par un protocole d'établissement de clés, les entités communicantes prouvent la possession de matériel de chiffrement authentifié. L'authentification des clés avec confirmation des clés produit une authentification par clé explicites.

3.3.4 Le rafraîchissement des clés

Cette propriété améliore la sécurité en assurant de nouvelles clés indépendantes entre les différentes sessions de communication. En séparant les sessions de communication, l'information disponible à des fins de cryptanalyse est limitée, ce qui rend l'attaque cryptanalyse plus difficile [49].

3.3.5 Le secret de transmission parfaite (PFS)

Assure que la corruption des clés à long terme ne peut pas être le résultat de la corruption des clés des précédentes sessions [62] ; [3] ; [49].

3.3.6 Résistance aux attaques clés connues

Un système de gestion de clés est vulnérable aux attaques clés connues (KKA) si une clé corrompue d'une session précédente ou un sous-ensemble de clés d'une session précédente permet ce qui suit [62] ; [3] ; [49] :

1- un adversaire passif peut compromettre les clés d'une session ultérieure,

2- un adversaire actif peut usurper l'identité des participants des autres protocoles.

3.3.7 Forward secrecy

Un système de gestion de clés avec la propriété Forward Secrecy empêche un adversaire de découvrir les clés ultérieures à partir d'un sous-ensemble contigu d'anciennes clés corrompues [42].

3.3.8 Backward secrecy

Un système de gestion de clés avec la propriété Backward secrecy empêche un adversaire de découvrir les clés précédents depuis un sous-ensemble contigu d'anciennes clés corrompues [42].

3.3.9 L'indépendance des clés

Elle garantit qu'un adversaire passif, qui connait un sous-ensemble de clés ne peut pas découvrir d'autres clés [42]. L'indépendance de clés inclu forward secrecy et backward secrecy mais n'implique pas la fraîcheur clés.

3.3.10 Disponibilité

La caractéristique de haute disponibilité empêche la dégradation des services de gestion de clés et veille à ce que le matériel de chiffrement soit fourni aux nœuds dans le réseau au moment adéquat.

3.3.11 Robustesse

Le système de gestion de clé doit tolérer les défaillances matérielles et logicielles, les liens asymétriques et unidirectionnels et la fragmentation / partitionnement du réseau en raison de la limite et des erreurs de la connectivité sans fil [63].

3.3.12 Survie

La survie est la capacité du service de gestion de clés à rester disponible même en la présence de menaces et de failles. La survie va au-delà de la sécurité et de la tolérance aux pannes et se concentre sur la prestation de service même lorsque le système est en partie corrompu ou les expériences ont échoué (la survie inclus donc la robustesse). La récupération rapide des services est nécessaire lorsque les conditions s'amélioreront [63]. La survie comprend aussi la « robustesse byzantine », ce qui implique que le service de gestion de clés devrait être en mesure de fonctionner correctement, même si certains mauvais comportements des nœuds participants tentent de perturber son fonctionnement. Plus précisément, les services de gestion de clés incluant la fonction de survie doivent se concentrer sur la prestation des services essentiels (par exemple les services de certification dans l'infrastructure à clé publique) et la préservation du matériel de chiffrement (certificats de clé publique, clé de session, etc.) La survie peut être résumée par les trois R [63] comme suit :

Résistance : la capacité du système à se défendre contre les attaques ou de les tolérer ;

Reconnaissance : la capacité du système à détecter les attaques en cours et à surveiller l'étendue des dégâts ou des corruptions.

Récupération : la caractéristique principale de la survie, c'est la capacité de maintenir les services pendant l'attaque, limiter l'étendue des dégâts et rétablir les services endommagés après une attaque.

3.3.13 Efficacité

Le service de gestion de clés doit être efficace en matière de communication, de calcul, de mémoire et des ressources énergétiques.

3.3.14 Extensibilité

L'extensibilité assure l'efficacité et la disponibilité tant que le nombre de nœuds dans le réseau change rapidement et de manière significative. Le système de gestion de clés devrait donc être à la taille du réseau de manière transparente.

3.4 Gestion de clés Peer-to-Peer pour les réseaux mobiles ad hoc

Dans ce qui suit, nous allons mettre l'accent sur la gestion de clés Peer-to-Peer pour les réseaux mobiles ad hoc (MANET). Les protocoles actuels sont classés dans les sous-ensembles suivants :

1. Autorité de certification partiellement distribuée ;
2. Autorité de certification entièrement distribuée ;
3. La gestion de clés basée sur l'identité ;
4. La gestion de clés basée sur l'enchaînement de certificat ;
5. La gestion de clés basée sur les clusters ;
6. La gestion de clés basée sur le pré-déploiement ;
7. La gestion de clés fondée sur la mobilité,
8. La gestion de clés en parallèle.

La plupart des sous-ensembles ci-dessus utilisent la cryptographie à clé publique en raison de sa supériorité dans la distribution des clés et parce qu'elle assure l'authentification et l'intégrité de la réalisation et élimine les rejets [78] ; [49].

Les systèmes de clés symétriques ont besoin d'un canal qui assure l'intégrité des données et la confidentialité. Cette dernière propriété ne peut pas être toujours disponible sans une autorité de confiance ou un canal latéral sécurisé (telle qu'une interface infrarouge).

- Le sous-ensemble d'autorité de certification partiellement distribuée est un groupe de protocoles qui distribue la confiance à un sous-ensemble d'entités de communication dans le réseau. L'approche atténue le point de vulnérabilité inhérent à l'autorité de certification centralisée. Les protocoles considérés pour la mise en œuvre de cette méthode ont été présentés dans [78] et [73].
- Le sous-ensemble des protocoles d'autorité de certification entièrement distribuée préserve la relation symétrique entre les entités de communication dans les MANETs en distribuant la charge de gestion de clés à toutes les entités de communication. Chaque nœud agréé dans le réseau reçoit une part de l'autorité de certification de la clé secrète permettant ainsi aux voisins de demander le service de certification. Le protocole qui a introduit cette méthode a été présenté en [47].
- Le sous-ensemble de protocoles de gestion de clés basée sur l'identité emprunte les concepts depuis le protocole d'autorité de certification partiellement distribuée mais utilise un crypto système basé sur l'identité pour réduire le besoin de stockage par rapport aux

crypto système à clé publique traditionnels. Le protocole de [41] est considéré comme représentatif de ce groupe de protocoles.

- Dans le sous-ensemble de protocoles de gestion de clés basée sur l'enchaînement des certificats, les entités de communication peuvent authentifier les certificats en recherchant les certificats qui sont enchaînés entre eux. L'enchaînement des certificats peut être expliqué par l'exemple suivant : Une partie A veut communiquer avec la partie C, ceci exige de la partie A l'authentification du certificat de la partie C. Les deux parties n'ont pas un historique de communication, mais la partie A approuve le certificat d'une troisième entité, la partie B. La partie B informe la partie A qu'elle approuve le certificat de la partie C. La partie A qui approuve le certificat de la partie B, approuve aussi le certificat de la partie C suite à la recommandation de la partie B. Il y a donc une chaîne de certificats entièrement connectée entre la partie A et C grâce à la partie B qui permet à la partie A d'authentifier le certificat de la partie C sans aucune communication précédente.
- Le sous-ensemble des protocoles de gestion de clés fondé sur les clusters(grappes) repose sur un algorithme de clustering qui divise le réseau en petits groupes. Les membres du groupe dans la même proximité peuvent surveiller leurs voisins et donner des recommandations aux membres des autres groupes sur l'authenticité des certificats de leurs voisins. Le sous-ensemble de protocoles basé sur les clusters est introduit dans le protocole présenté en [52].
- Le sous-ensemble des protocoles de gestion de clés basée sur le pré-déploiement fait usage d'une autorité déconnectée « offline authority » pour distribuer à chaque nœud le matériel de chiffrement avant la formation du réseau. Il est largement convenu que les techniques de pré-distribution des clés sont idéalement adaptées pour établir une connectivité sécurisée à grande échelle dans les réseaux de capteurs distribués [27]. Les limites des réseaux de capteurs rendent les techniques classiques d'établissement de clés inadaptés (telle que la cryptographie à clé publique).
- Le sous-ensemble des protocoles de gestion de clés basée sur la mobilité exploite la mobilité et les rencontres des nœuds pour créer des associations de sécurité et justifier l'authentification mutuelle entre les utilisateurs. Contrairement aux sous-ensembles discutés précédemment, les protocoles de ce groupe introduisent un changement dans le paradigme par rapport aux tentatives précédentes visant à assurer la gestion de clés dans les MANETs totalement auto-organisés. Plutôt que d'essayer de trouver des solutions adaptées pour les réseaux filaires conventionnels, les protocoles de ce groupe utilisent les caractéristiques uniques de MANET à leur avantage. La combinaison de n'importe quelle approche de gestion de clés décrite ci-dessus donne lieu à ce qui est appelé le sous-ensemble de gestion de clés parallèles. En utilisant deux ou plusieurs approches en parallèle, les avantages de l'une sont utilisés pour atténuer les inconvénients de l'autre. Ce sous-ensemble peut être représenté par le système présenté dans [75], qui combine une autorité de certification partiellement distribuée [74] avec une gestion de clés basée sur l'enchaînement des certificats [15].

3.4.1 Approche d'autorité de certification partiellement distribuée

Une des premières approches pour résoudre le problème de gestion de clés dans les MANETs a été publiée dans [78]. Cette approche a ensuite été étendue dans les [71], [72], [73].

Le modèle du système

[78] ont proposé un service de gestion de clés publiques distribuées pour les réseaux ad hoc où la confiance est répartie entre un ensemble de nœuds en laissant ces derniers partager le secret du système. L'autorité de certification distribuée (DCA), illustrée dans la Figure 3.2 [78], se compose de n nœuds serveurs qui, dans leur ensemble, ont une paire de clé publique/privée K/k. La clé publique K est connue pour tous les nœuds du réseau, tandis que la clé privée k est divisée en n parts $(S_1, S_2, S_3, ..., S_n)$, une pour chaque serveur.

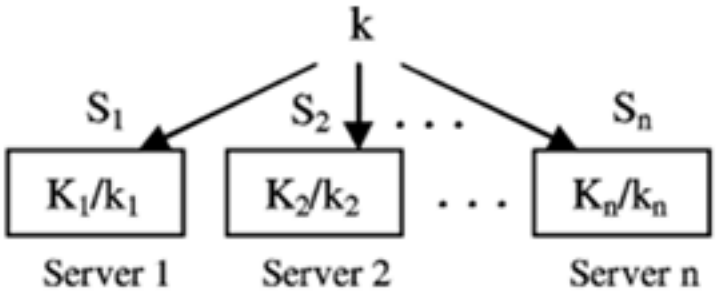


FIGURE 3.2 – Service de configuration gestion de clés

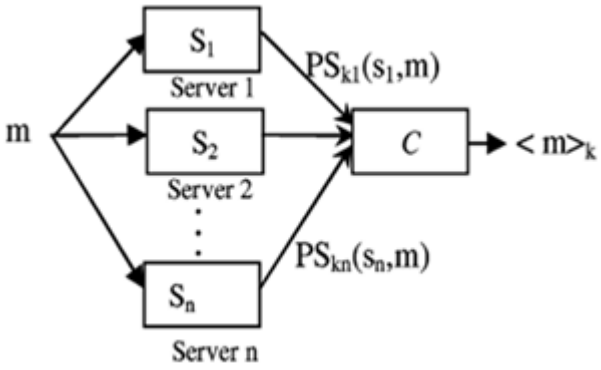


FIGURE 3.3 – Génération de signature à seuil

L'autorité de certification distribuée (DCA) signe un certificat par la production d'un groupe de signature à seuil, comme le montre la Figure 3.3 [78]. Chaque serveur génère une signature partielle à l'aide de sa clé privée et soumet cette signature à un combinateur C. Le combinateur peut être n'importe quel serveur et nécessite au moins $t + 1$ des parts afin de reconstruire avec succès la signature numérique.

L'analyses du système

La phase d'initialisation

Le système tel que proposé dans [78] exige une tierce partie de confiance déconnectée (TTP) « offline trusted third party » pour construire le service de gestion de clés publiques distribuées. Avant la création du réseau, la TTP utilise un système de partage des secrets à seuil [61] pour générer des parts $(S_1, S_2, S_3, ...S_n)$ de la clé privée k de la DCA. Ces parts sont distribuées aux n nœuds arbitraires (serveurs) qui, ensemble, forment la DCA. La TTP doit également émettre la clé publique authentique K de la DCA à tous les nœuds du réseau. Afin d'éviter que les nœuds non autorisés obtiennent des services de certification de la DCA, l'autorité déconnectée devra émettre chaque nœud avec un certificat signé par la clé privée de la DCA. Si le certificat de chaque nœud est également stocké sur les serveurs DCA, les associations sécurisées préétablies sont disponibles pour authentifier les demandes de certification. Si avant la formation du réseau, l'autorité déconnectée ne délivre pas chaque nœud avec son propre certificat, le système est soumis à l'attaque « Sybil » [26].

L'extraction du certificat

Les nœuds qui demandent un certificat, doivent contacter avec succès au moins $t+1$ serveurs parmi les n serveurs formant la DCA. Comme l'illustre la Figure 3.3, le système de signature à seuil proposé dans [78] fait usage d'un nœud combinateur C pour combiner les signatures numériques partielles des $t+1$ serveurs. Tout nœud peut être choisi comme un combinateur, puisque aucune information supplémentaire au sujet de la clé privée k n'est divulguée à C. Il est toujours possible pour un nœud combinateur d'être corrompu par un adversaire ou être indisponible en raison de l'épuisement de la batterie ou de la mauvaise connectivité. Comme solution, [78] ont proposé la sélection de $t+1$ nœuds comme combinateurs pour s'assurer qu'au moins un combinateur peut reconstruire avec succès la signature numérique. Tous les nœuds du réseau, y compris les combinateurs, peuvent vérifier la validité de la signature en utilisant la clé publique K de l'autorité de certification.

La proposition présentée en [71] [72], [74] diffère de la proposition originale de [78], puisque le système de signature à seuil de Yi et Kravets ne nécessite pas un nœud combinateur C pour construire la signature du groupe.

Dans [71], [72], [74], la DCA est appelée une autorité de certification mobile (MOCA). Dans le cadre de la MOCA, le schéma de communication est un un-plusieurs et vice versa, ce qui signifie qu'un nœud qui nécessite des services du certificat doit contacter au moins $t+1$ nœuds de la MOCA et recevoir les réponses de chacun d'eux. La combinaison de signatures partielles est donc effectuée par le nœud demandeur de la certification.

La proposition originale de [78] ne précise pas quel est le protocole de communication (mécanisme de récupération du certificat) utilisé par un nœud afin qu'il prenne contact avec le service de gestion de clés. La proposition de [71] [72], [74] se base principalement sur le système de communication un-à-plusieurs-à-un entre un nœud et la MOCA. Le protocole de certification MOCA permet à un nœud ayant besoin de services de certification de diffuser des paquets de sa demande de certification (CREQ). Tout nœud de la MOCA qui reçoit les paquets (CREQ)

répond par une réponse de certification (CREP) contenant sa signature partielle sur le certificat.

- Si le nœud demandeur reçoit avec succès $t+1$ CREP valables dans le délai imparti, il peut reconstruire le certificat complet.
- Si le certificat est vérifié et il est correct, la demande de certification est réussie.
- Si le nombre de CREP est insuffisant après l'expiration du délai, le processus CREQ du nœud échoue et le nœud peut lancer une autre requête.

La CREQ et la CREP sont similaires à la requête de route (RREQ) et la réponse de route (RREP) des protocoles de routage à la demande dans les réseaux ad hoc (par exemple AODV [55] et DSR [9]). Lorsque les paquets CREQ sont acheminés vers les nœuds, un chemin inverse est créé pour l'expéditeur. Le chemin inverse est associé au délai imparti pour permettre aux paquets CREP d'aller au nœud qui demande le service du certificat. Dans ce cas, les protocoles de routage à la demande et le protocole de certification MOCA peuvent s'entraider en partageant les informations de routage [74].

Inondation : dans la première implémentation du protocole de certification MOCA présenté dans [71], les inondations sont utilisées pour la diffusion des données fiables. Comme le montre les résultats de simulation présentés dans [71], la technique des inondations donnent une approche efficace pour contacter au moins $t+1$ serveurs, mais les coûts de communication seront très élevés. Pour empêcher l'inondation due à une diffusion potentielle, des IDs de diffusion sont utilisés (de façon similaire à ceux de [55]) de sorte que toutes les CREQ générées par la même requête sont étiquetées avec le même ID pour permettre aux nœuds intermédiaires de stopper les demandes déjà transmises.

β–**Unicast** : afin de réduire la quantité du trafic qui résulte de l'inondation tout en gardant un niveau de service acceptable, la méthode β-unicast a été introduite dans le deuxième rapport du service MOCA de [72]. La méthode repose sur des unicast multiples au lieu de l'inondation en utilisant les informations mises en cache de la table de routage. Les études dans [71] ont montré que dans des scénarios de trafic de certification raisonnables, le poste client met en cache un nombre modéré d'itinéraires vers les nœuds MOCA.

Le paramètre β représente le seuil du nombre de routes en cache requis par un nœud pour utiliser l'unicast au lieu de l'inondation. L'inondation est donc la méthode par défaut pour communiquer avec la MOCA dans le cas où le nœud ne parvient pas à accumuler assez d'informations sur les tables de routage dans son cache. Si pour être stable, le nœud perçoit le réseau avec une mobilité relativement faible, les $t+1$ itinéraires mis en cache peuvent être suffisants pour initier plusieurs unicast CREQ. Afin de garantir une réponse d'au moins $t+1$ serveurs, les [72] introduisent une marge de sécurité α. Dans le cas d'instabilité, les nœuds devraient envoyer α CREQ supplémentaires pour augmenter la probabilité du succès jusqu'au $t+1$ serveurs. La somme de α et le crypto seuil t sont appelés le seuil unicast, il est représenté par β. S'il ya plus de (β) routes suffisantes dans le cache local du nœud, le choix des itinéraires à utiliser aura un impact sur la performance et pour cela, trois systèmes ont été proposés dans le [72] :

les nœuds MOCA aléatoires : un nombre β aléatoire de nœuds MOCA sont sélectionnés dans la table de routage.

les nœuds MOCA les plus proches : en utilisant les informations disponibles sur le nombre de sauts dans la table de routage, les β nœuds de MOCA avec le nombre de sauts minimum sont choisis.

les nœuds MOCA les plus récents : les β routes les plus récemment ajoutées sont utilisées pour le β-unicast.

La révocation du certificat

La révocation du certificat n'a pas eu beaucoup d'attention dans [78] ou [71] [72], [74]. L'approche d'une simple liste de révocation certifiée (CRL) a été proposée. Dans le cadre de la MOCA, $t+1$ nœuds doivent convenir pour révoquer un certificat. Chaque nœud de la MOCA génère une révocation certifiée qui contient des informations sur la révocation du certificat. Le nœud MOCA diffuse ainsi son certificat de révocation partiellement signé à travers le réseau. Les nœuds qui reçoivent $t+1$ certificats partiels devront reconstruire la révocation du certificat et mettre à jour leur CRL locale.

Le renouvellement du certificat

Les certificats expirés peuvent être renouvelés par l'envoi d'un message CREQ à n'importe quels $t+1$ nœuds MOCA. Chaque nœud MOCA mettra à jour le contenu du certificat avec des informations pertinentes (par exemple avec un nouveau délai d'expiration) et renouvelle le lien entre la clé publique et l'identité des nœuds en générant une signature partielle.

La mise à jour partagée

Les services de gestion de clés dans [78] et [74] emploient le partage proactif du rafraichissement des clés [33] pour déjouer les adversaires mobiles et à s'adapter aux changements dans le réseau. Un adversaire qui tente de briser le système doit compromettre plus que le seuil des t serveurs dans l'intervalle du temps que prend la mise à jour des clés.

Discussion et commentaires sur les approches d'autorité de certification partiellement distribuée

Un des avantages des propositions de l'autorité de certification distribuée, est qu'il remédie au manque d'infrastructure de serveur dans les MANETs en distribuant les fonctionnalités d'une autorité centrale au sein d'un groupe d'utilisateurs. Le système partiellement distribué offre des services de sécurité avec une plus grande disponibilité par rapport à une approche serveur centralisé. Le système, cependant, ne possède pas de mécanisme pour la révocation des certificats ou les mécanismes de synchronisation pour mettre à jour les serveurs. La solution présentée dans [78] émane de son prédécesseur filaire, à savoir, une autorité de confiance, un serveur spécialisé et des nœuds combinateurs. La plateforme MOCA proposée dans [74] et contrairement à la proposition initiale de la DCA [78] n'a pas besoin d'un serveur combinateur C pour résoudre efficacement le problème d'assurer la disponibilité du C. La solution a toujours une approche largement centralisée bien que le système de seuil permette à t serveurs DCA d'être corrompus

sans sacrifier le service de gestion de clés. Une des principales hypothèses des solutions proposées dans [78] et [74] est la présence d'une TTP déconnectée qui habilite les serveurs ou distribue le matériel de chiffrement avant la formation du réseau. L'hypothèse d'une TTP déconnectée rend le système totalement inadapté aux MANET auto-organisés. L'information distribuée par la TTP déconnectée rend la solution non vectorielle puisque tous les certificats du réseau doivent être connus a priori par les serveurs de la DCA afin de fournir un contrôle d'accès aux services de certification.

La surcharge de communication mise en place par [78] et [74] reste un sujet de discussion tant que les nœuds doivent contacter la DCA à chaque fois qu'ils ont besoin de la certification. Chaque nœud peut communiquer avec la DCA en inondant le réseau avec des demandes de certification. La méthode des inondations est un moyen efficace de communiquer avec t sur les n nœuds de la DCA [74]. Le problème est que chaque nœud de la DCA réponde avec une réponse de certification provoquant un « effet tempête » des paquets inverses de $O(n)$ [74].

[74] ont tenté d'optimiser le système en étudiant les différentes techniques de diffusion des données qui permettent de contacter la DCA au moyen de la simulation. [17] ont étudié le manycast qui semble être une technique prometteuse pour contacter seulement un sous-ensemble de membres d'un groupe avec un minimum de coût de communication. Le manycast apporte de meilleures performances par rapport aux techniques des inondations et des β-unicast en particulier s'il est intégré dans le protocole de routage.

L'analyse de ces systèmes a montré qu'ils sont soumis aux insuffisances suivantes : la distribution de la clé privée du système est réalisée avec un système de partage de secrets à seuil [61]. Le plus important dans la sécurité cryptosystème à seuil est le choix des paramètres de sécurité (n, t) où n est le nombre total des nœuds formant la DCA et t est le seuil de nœuds qui doivent être corrompus pour rendre le système non sécurisé. Le processus de choix des paramètres (n, t) appropriés est une tâche très difficile. Le choix de ces paramètres implique inévitablement un compromis entre la sécurité du système et la disponibilité des nœuds de la DCA. Certains facteurs à considérer lors du choix (n, t) sont :

1. l'environnement réseau du MANET ;
2. la sécurité physique des utilisateurs des nœuds ;
3. la bande passante nécessaire ;
4. la fréquence des demandes de certification de la DCA ;
5. la mobilité habituelle des utilisateurs ;
6. les capacités des attaquants ;
7. la disponibilité des nœuds de la DCA.

Par exemple, les MANETs sont soumis à des erreurs fréquentes de la connectivité sans fil, des ressources d'énergie limitées (une faible durée de vie de la batterie) et les portées de transmission limitées. Les nœuds formant la DCA peuvent donc être souvent indisponibles pour l'interprétation des services de certification. Un environnement hostile, tels que celui trouvé dans les applications militaires, requiert que t soit augmenté en tant que possible. Cependant, le problème est que l'augmentation de t au-delà d'un certain point "sûr" peut empêcher les utilisateurs de contacter avec succès t sur les n nœuds de la DCA. L'observation fondamentale est

que la prise en considération de tous ces facteurs lors du choix (n, t), tout en gardant à l'esprit le compromis sécurité /disponibilité, est un problème difficile à résoudre et il est au centre de la sécurité fournie par les cryptosystèmes à seuil.

Dans les cryptosystèmes à seuil, il est impossible de supposer qu'un adversaire mobile ne peut pas compromettre plus que les t actionnaires pendant toute la vie du système [78] ; [33] ; [47]. Cela force les nœuds formant la DCA à exécuter un protocole de renouvellement partagé [33] dans une période T variable. Le choix de T est influencé par des facteurs similaires à ceux qui affectent la sélection de (n, t). Comme les membres de la DCA n'ont pas d'accès à une TTP centrale en ligne, le protocole de renouvellement partagé doit être entièrement distribué.

La structure d'accès initial $\Gamma_p^{(n,t)}$ du système de distribution ne restera pas constante [24]. Supposer que les mêmes actionnaires doivent être présents à tout moment est irréalisable et le compromis sécurité/disponibilité sera altéré (en présélectionnant (n, t) à la volée) en fonction de la vulnérabilité du système, en changeant l'environnement réseau et selon la fonctionnalité courante du cryptosystème. Les utilisateurs peuvent aléatoirement rejoindre ou quitter le groupe de la DCA, donc la DCA exposera les membres dynamiques du groupe comme associé à des pairs de groupes dynamiques [62]. Les paramètres de sécurité (n, t) devront être ajustés pour permettre l'adhésion dynamique.

Cela oblige les nœuds de la DCA à exécuter un protocole de redistribution des secrets [24] ; [69] afin de distribuer les parts à une nouvelle structure d'accès $\Gamma_{p'}^{(n',t')}$ à chaque changement d'adhésion ou d'ajustement du compromis sécurité/disponibilité. Les coûts associés à la mise à jour des secrets entièrement distribués et les protocoles de la redistribution des secrets peuvent ne pas être idéaux ou pratiques dans les MANETs. Les études sur les mises à jour des secrets entièrement distribués et les protocoles de redistribution des secrets [69] ; [33] [24] existants ont montré que les mécanismes de mise à jour/recomposition de secrets distribués ont un coût de communication et de calcul élevé . Ces calculs devront être exécutés plus fréquemment que prévu dans les MANETs. Le partage de la mise à jour requiert $O(n + t)$ messages de chaque membre de la DCA, tandis que chaque membre de la DCA effectue $O(nt)$ exponentiations et génère $O(t)$ nombres aléatoires. Le partage du protocole de renouvellement a des coûts similaires avec $O(n' + t')$ messages pour chaque membre de la DCA, $O(n't')$ exponentiations et $O(t')$ nombres aléatoires générés par chaque membre de la nouvelle structure d'accès $\Gamma_{p'}^{(n',k')}$. Le réseau dans son ensemble a un coût de communication de $O\left(n^2 + nt\right)$ et $O\left(n'^2 + n't'\right)$ pour la mise à jour et la redistribution des secrets respectivement. Dans l'analyse des modèles, un système de diffusion synchrone a été supposé. Il est généralement convenu que le réseau à large synchronisation n'est pas facilement réalisable dans les MANETs. Sans un système de diffusion synchrone, il n'est pas clair si la mise à jour de secrets entièrement distribués et les protocoles de redistribution sont possibles en se défendant contre les attaques connues [77]. Actuellement, il n'existe pas un système de partage, de mise à jour et de redistribution des secrets disponibles adaptés aux MANET.

Les nœuds dans les MANETs ont une relation symétrique, donc ils sont tous égaux et devraient se partager et distribuer équitablement la responsabilité de fournir les fonctions du réseau. Ceci n'est pas seulement important pour des raisons de sécurité mais pour permettre au réseau d'assurer des services fiables et disponibles qui mettent la même charge des ressources

de calcul, de mémoire et d'énergie sur l'ensemble des participants du réseau. L'utilisation d'une DCA comme une TTP en ligne viole la relation symétrique entre les nœuds du réseau. Quand la DCA est formée par tous les nœuds du réseau, comme en [47], la relation symétrique est préservée. Ceci, cependant, met en péril la sécurité globale du système puisqu'un attaquant peut compromettre n'importe quels t nœuds de l'ensemble du réseau pour briser le seuil du cryptosystème. S'il existe une hétérogénéité entre les participants du réseau, l'utilisation des nœuds avec les ressources les plus avancées pour les nœuds de la DCA n'est pas seulement non propice mais favorise les attaques « par déni de service ». La responsabilité inégalement partagée est à éviter dans les systèmes entièrement distribués car elle motive la localisation des zones de vulnérabilité. Il faut, cependant, noter que la notion d'hétérogénéité peut être exploitée dans les paramètres réseau où les nœuds n'ont pas de relations symétriques, tels que ceux trouvés dans les réseaux militaires [74]. Dans un tel scénario, l'exploitation des nœuds endommagés est toléré pour le bénéfice ou la "survie" de réseau dans son ensemble.

Savoir comment les membres de la DCA vont collaborer pour signer les certificats est une autre question importante mais négligée. Ceci est un problème plus difficile de ce qu'on peut penser. En effet, la conception du groupe de signatures est beaucoup plus complexe que la conception d'un parti unique de signatures. Ce qui est nécessaire pour les systèmes basés sur l'autorité distribuée proposés dans [78], [74], et [47] est un schéma multi-signature à seuil [67] ; [44]. Les systèmes « multi-signature à seuil » se distinguent des systèmes « de groupe de signatures à seuil » [54] par le fait que dans le second et par définition, les signataires individuels restent anonymes. Dans les systèmes de groupe de signatures à seuil et exceptionnellement pour le chef du groupe, il est difficile de tirer les identités à partir du groupe de signatures. En revanche, dans les systèmes multi-signature à seuil, les signataires individuels sont publiquement traçables et ne bénéficient pas de l'anonymat. Par conséquent, la propriété de traçabilité des systèmes multi-signatures à seuil permet aux signataires individuels d'être tenus responsables dans le domaine public. C'est pourquoi, la traçabilité des signataires individuels est indispensable dans les MANETs. L'état de l'art courant des systèmes multi-signatures à seuil [67] ; [44] est notoirement défectueux [50] ; [66], [68] ; [70]. Ce qui est encore plus décourageant est que tous les systèmes existants de groupe des signatures proposés à ce jour sont pour les réseaux conventionnels. Il est largement connu que dans la plupart des cas, les solutions adaptées pour les MANETs, exigent un changement de paradigme en vue d'atténuer les conséquences des caractéristiques uniques de MANET. La solution existante dans le sous-ensemble autorité de certification partiellement distribuée n'a pas pour objectif de briser le cycle de l'interdépendance de la sécurité routage [7].

Un cryptosystème à seuil peut avoir d'autres applications dans les MANETs, mais elles ne sont pas idéales pour la réalisation de la gestion de clés.

3.4.2 Approches d'autorité de certification entièrement distribuée

En [43] et [47], une solution de gestion de clés publiques a été proposée. Elle repose sur l'approche initialement présentée dans [78] et utilise aussi un système de signature à seuil (n, k) pour former une autorité de certifications distribuées (DCA). Le système accroît la disponibilité de la fonction de [78], en choisissant n de façon à ce qu'il touche tous les nœuds du réseau. La

clé privée SK de la DCA est donc partagée entre tous les nœuds du réseau et permet à un nœud demandant le service de la DCA de contacter n'importe quels k nœuds voisins à un seul saut. Contrairement à [78], aucune différence n'est faite entre le serveur et les nœuds clients à l'égard de services de certification. La solution inclut un mécanisme de mise à jour pour éviter aux adversaires les plus puissants de compromettre le service de certification.

Le système et le modèle adversaire

Le modèle du réseau considère un réseau ad hoc sans fil avec l'insécurité, les erreurs fréquentes et les canaux de communication à bande passante limitée. Le réseau n'a pas d'infrastructure et sa topologie change dynamiquement. Les hypothèses suivantes sont faites :

1. Chaque nœud v_i possède un identifiant unique (ID) et il est capable de découvrir ses voisins à un seul saut.
2. Chaque nœud possède un certificat valide signé par la clé privée SK de la DCA liant son ID à une clé publique P_{v_i}. Un certificat signé par SK peut être vérifié par la clé publique authentique PK de la DCA.
3. La communication à un-saut est plus fiable que la communication multi-sauts.
4. A chaque instant, chaque nœud possède plus de k voisins à un-saut.
5. La détection de la mauvaise conduite des nœuds est plus facile et plus pratique entre les voisins à un-saut contrairement aux voisins multi-sauts.
6. La mobilité du réseau est caractérisée par la vitesse du nœud avec la plus haute vitesse S_{max}. La proposition de [47] et [43] tente à pallier aux deux types d'attaques :
   - Déni de service (DoS).
   - L'intrusion d'adversaires ou l'effraction de nœud.

Les adversaires peuvent émettre des attaques DoS sur les différentes couches de la pile réseau. Comme défini dans [33], les adversaires peuvent être caractérisés par un des deux modèles :

- **Modèle I** : pendant la durée de vie de l'ensemble du réseau, un adversaire ne peut pas attaquer avec succès et compromettre plus de k nœuds.
- **Modèle II** : si la durée de vie du réseau est divisée en créneaux T, un adversaire ne peut pas attaquer avec succès et compromettre plus de k nœuds dans T. [47] et [43] ont tenté de se défendre contre les adversaires du Modèle II avec un mécanisme évolutif de mise à jour parallèle.

Analyse des systèmes

Phase d'initialisation de service de certification localisée

Le système tel que proposé dans [47] et [43] nécessite une tierce partie de confiance déconnectée (TTP). La conception basée sur l'RSA a un système DCA avec une paire de clés RSA $\{SK, PK\}$. Avant la formation du réseau, la TTP distribue pour chaque nœud un certificat signé avec la clé privée SK de la DCA. Le certificat est une liaison entre l'ID unique des nœuds et une clé publique qui peut être vérifié avec la clé publique authentique PK de la DCA connue

par tous les nœuds du réseau. Le service de certification localisée ne crée ou ne délivre jamais un certificat initial. Sa fonctionnalité est le renouvellement des certificats expirés ou la réémission des certificats (ou mécanismes de détection d'intrusion) corrompus par des adversaires. La TTP déconnectée distribue les k premiers nœuds partagés. En termes plus précis, la TTP distribue aux k premiers nœuds du réseau un partage polynômial P_v de la signature de l'exposant du certificat SK selon un polynôme aléatoire $f(x)$ tel que $P_v = f(v)$. A l'issue de cette tâche, la TTP n'a plus de rôle dans le fonctionnement du réseau. Le polynôme $f(x)$ peut être défini comme $f(x) = SK + \sum_{i=1}^{k-1} f_i x^j$, où $f_1, f_2, ..., f_{k-1}$, sont uniformément répartis dans un intervalle fini F.

Auto-initialisation localisée

[43] ont proposé un algorithme utilisé pour distribuer des parties de SK aux nœuds qui rejoignent le réseau. Un nœud qui rejoint le réseau doit avoir un certificat liant son ID et sa clé publique signée par SK. Seul un nœud joint à un certificat valide et donc vérifiable avec PK peut obtenir une part de la SK. Depuis que leur architecture est basée sur le système de partage de secrets à seuil de [61], seule une coalition de k nœuds peut délivrer le nœud initialisé avec une part de la SK. Le protocole d'auto-initialisation peut être résumé en quatre étapes :

1. Un nœud qui veut rejoindre le réseau envoie aux nœuds voisins une demande de service avec des informations sur la coalition locale.
2. Chaque voisin (membre de la coalition) sélectionne un nombre aléatoire (nonce) pour tout nœud dont l'ID est plus faible que les IDs des autres nœuds. Un système de brassage complet est utilisé par les nœuds pour échanger ces nonces. Dans l'échange Peer-to-Peer de nonces, les nonces sont niés par le nœud ayant l'ID le plus faible. Les nonces sont cryptés avec la clé publique du destinataire membre de la coalition.
3. Les paquets brassés chiffrés ou les nonces sont ensuite acheminés aux membres de la coalition.
4. Chacun des membres de la coalition calcule une part partielle secrète mélangée SS de son partage polynômial P_v. Comme il est possible de dériver SS depuis P_v, SS n'est plus visible au nœud v_x par l'ajout de la somme de tous les nonces à la SS. Après le décryptage des nonces, chaque membre de la coalition calcule $SS' = SS + \sum nonces$ et transmet ses parts partielles calculéeSS' à v_x.

Délivrance et renouvellement des certificats

Comme décrit dans la phase d'initialisation, chaque nœud détient une part de la clé privée SK selon le polynôme aléatoire $f(x)$. En [43], le nœud v_i localise premièrement une coalition β de k voisins $\{v_1, ..., v_k\}$ et diffuse des requêtes de certification pour les voisins sélectionnés. Chaque nœud $v_j \in \beta$ consulte ses données de surveillance et rend la décision d'accorder ou de refuser la certification. Si v_i est certifié comme légitime, v_j renvoie un certificat partiel Pv_jau nœud demandeur v_i. Le nœud récupère alors le certificat à partir des certificats partiels à l'aide de l'algorithme «k-bounded coalition» comme donné à [43]. Même si les membres de la coalition

sont l'objet d'attaques, ce système reste fonctionnel du moment que seulement k signatures partielles sont nécessaires à partir des nœuds voisins. On compte cependant deux inconvénients dans l'approche [43] :

1. Si l'un des nœuds $v_j \in \beta$ a échoué pendant le processus de requête, tous les autres certificats partiels deviennent inutiles.
2. Quand v_j reçoit une demande de v_i, la surveillance enregistrée peut ne pas fournir suffisamment d'informations pour accorder la certification. Ce sera le cas dans un réseau à haute mobilité où v_j et v_i ne se sont jamais rencontrés auparavant ou ont eu une interaction insuffisante.

Afin de résoudre le premier inconvénient, la coalescence dynamique est introduite. Cette méthode provient du fait que la coalition peut être formulée de manière dynamique par tout nœud répondant, au lieu d'être précisée a priori par le nœud demandeur v_i. [47] illustrent comment le dynamisme de la coalescence peut surmonter la première difficulté identifiée ci-dessus. Pour résoudre le second inconvénient et s'accommoder à la mobilité, la certification est accordée s'il n'y a pas de mauvais enregistrements trouvés. La disponibilité des enregistrements n'est donc pas prise comme une raison suffisante pour refuser la certification.

Révocation de certificats

Les enregistrements de certificat maintenus par le nœud v_j se composent de deux composantes : données surveillées (certificats et le comportement) des nœuds voisins et une liste de révocation des certificats (CRL). La CRL est une liste contenant les ID des utilisateurs et les accusateurs. Si un nœud v_j déduit par les données de surveillance directe qu'un nœud voisin est corrompu, il le marque «condamné» dans sa propre CRL. Le v_j accusateur inonde le réseau avec des accusations signées contre ce nœud condamné. Supposons maintenant que le nœud v_j reçoit une accusation signée contre un nœud accusé, il vérifie dans la CRL si le nœud a été préalablement marqué «Condamnés». Si c'est le cas, le message est considéré comme étant une confirmation de l'accusation et il est abandonné, sinon, le nœud est marqué comme «suspect». Afin d'éviter l'accusation des nœuds légitimes, au moins k accusations contre un nœud sont nécessaires avant qu'il soit marqué «Condamné».

Les mises à jour parallèles partagées

Pour qu'un système de distribution partagé puisse être robuste contre le modèle adversaire II, des mises à jour périodiques sont requises [33]. Ceci empêche un attaquant de compromettre plus de k parts de secret entre les mises à jour périodiques partagées. La proposition de [43] donne deux approches pour réaliser les mises à jour partagées.

1. La première approche est un processus basé sur l'auto-initialisation localisée.
2. La seconde approche est les fonctions des mises à jour parallèles partagées avec une convergence plus rapide. La mise à jour est effectuée par la distribution d'un nouveau polynôme aléatoire $f_{UPDATE}(x)$ dont les coefficients sont cryptés avec SK pour assurer l'authenticité. La nouvelle part du nœud dans SK peut alors être évalué en collaboration des k

nœuds voisins en tant que $P_{vUPDATE} = f_{UPDATE}(v_i)$. Chaque voisin renvoie sa part partielle de mise à jour à $v_i$de manière similaire à celle du service de certification. L'approche de mise à jour parallèle partagée est donc constituée de trois étapes :

a- la génération collaborative de la mise à jour polynomiale $f_{UPDATE}(x)$;
b- la propagation robuste des coefficients du polynôme de mise à jour,
c- l'évaluation distribuée des mises à jour partagées.

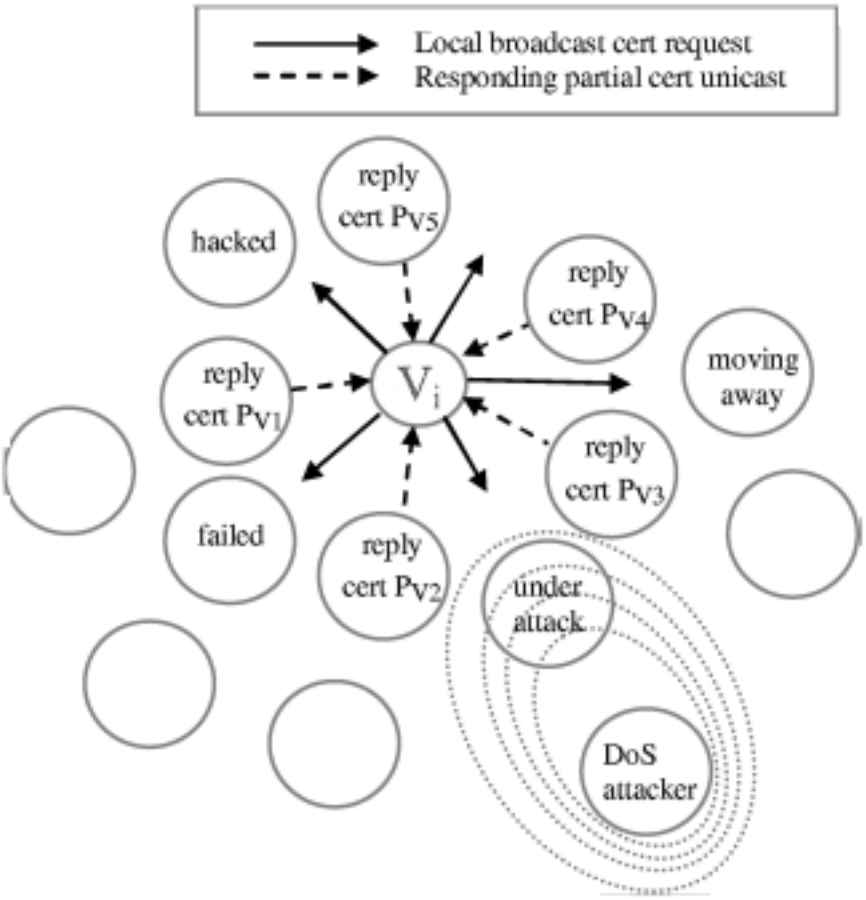


FIGURE 3.4 – Coalition dynamique

Discussion et commentaires sur l'approche d'autorité de certification entièrement distribuée

La proposition d'autorité de certification entièrement distribuée représente une amélioration pour les systèmes originaux partiellement distribués CA et ceci par la distribution équitable de la charge d'avoir une part de la clé secrète sur l'ensemble des nœuds du réseau ce qui préserve efficacement la relation symétrique entre les participants du réseau. Puisque maintenant n'importe quels k nœuds dans un voisinage local peuvent renouveler ou délivrer un certificat, la

disponibilité du service de gestion de clés est augmentée. Les nœuds qui ne sont pas en possession d'une part de la clé secrète peuvent également contacter au moins k nœuds pour obtenir une part.

Malgré ces améliorations dans la conception, la proposition souffre de la plupart des points de faiblesses discutées dans le paragraphe précédent comme dans la proposition initiale [78], avant la formation du réseau, une autorité de confiance déconnectée doit émettre pour chaque nœud un certificat liant un identifiant ID unique à une clé publique. Cette exigence rend la proposition inadaptée pour les réseaux ad hoc totalement auto-organisés. De même, à [78] et [74], cette solution est également non vectorielle, du moment que toutes les identités doivent être connues à priori. Depuis que le nombre k est un compromis entre la sécurité et la disponibilité et comme le réseau s'étend, la solution nécessite un moyen d'ajustement k. Tel que souligné dans [15], il n'est pas clair comment k sera ajusté dans un réseau où il y a une augmentation ou une diminution rapide dans la densité des nœuds.

Depuis que n'importe quels k nœuds dans le réseau peuvent être corrompus pour casser le système, l'augmentation de la disponibilité du service de certification se fait également au prix de la sécurité. De ce fait, l'hypothèse d'un adversaire modèleII n'est réalisable que si n couvre l'ensemble du réseau. Pour cette raison, k doit toujours être choisi assez grand pour assurer la sécurité du système.

L'hypothèse que chaque nœud doit toujours avoir au moins k voisins à un saut est une limitation : les nœuds peuvent se retrouver avec moins de k voisins plus fréquemment que prévu. Comme dans le cas de [78] et [74], il n'est pas clair si cette approche peut briser le cycle de l'interdépendance de la sécurité de routage [7].

3.4.3 Approches de l'identité des principaux dirigeants

La cryptographie basée sur l'identité [39] ; [8] ; [18] provient de la nécessité de réduire le coût de stockage de la mémoire des principaux systèmes publics classiques et de la charge d'obtention des clés publiques authentiques explicitement. Les clés publiques d'un schéma basé sur l'identité ne sont rien d'autre que l'identité des utilisateurs eux-mêmes. Les identités qui sont connues du public des données doivent identifier de façon unique les utilisateurs.

Les schémas de signature basés sur l'identité sont spécifiés par quatre algorithmes [8] :

1. Configuration : l'algorithme prend comme paramètres de configuration la sécurité d'entrée et retourne pour le système une paire de clés public/privé K_m/k_mmaître. La clé principale privée n'est connue que par la tierce partie de confiance (TTP) ou le générateur de clé privée (PKG) du système.
2. Extraction : l'algorithme prend en entrée la clé privée principale et une identité ID et renvoie la clé privée personnelle correspondant à l'ID.
3. Chiffrement : l'algorithme de chiffrement prend en entrée la clé publique maître K_m, l'ID du destinataire et un message m et renvoie les cryptogrammes correspondants. Cet ID sert comme clé publique du destinataire.
4. Déchiffrement : l'algorithme prend en entrée la clé publique maître, un texte chiffré et la clé privée personnelle et renvoie le message d'origine crypté avec l'identité correspondant

à la clé privée personnelle. Les clés privées personnelles dans un système de cryptographie basé sur l'identité peuvent aussi être vues comme un certificat à clé symétrique implicite, c'est la clé privée personnelle cryptée avec la clé privée principale du PKG.

Modèle de système

En [41], La cryptographie basée sur l'identité [39] ; [8] est combinée avec un seuil de cryptographie [49] pour éviter le coût très élevé du calcul de la cryptographie à clé publique. Cette solution est similaire à celle de [78] avec l'autorité de certification remplacée par un générateur de clés privées à seuil (PKG). Les algorithmes discutés ci-dessus réalisent collectivement un cryptosystème basé sur l'identité. Tous ces algorithmes devront être adaptés à un système de chiffrement à seuil et mis en œuvre sur les nœuds formant l'autorité distribuée qui est le PKG [1].

Analyse des systèmes

Phase d'initialisation

À la formation du réseau, les utilisateurs s'accordent sur une politique d'émission de clé et l'échange des paramètres pertinents de la sécurité. Ceux-ci doivent être mutuellement acceptés et les nœuds n'approuvant pas ces paramètres peuvent choisir de suspendre le processus de formation réseau. L'ensemble initial des nœuds forme alors le service de génération de clé privés à seuil (PKG), qui génère de manière distribuée une clé publique/privée maître. La clé privée maître est donc distribuée à n nœuds, tenant chacun une part de la SK. Un adversaire avec moins du seuil t parts ne peut pas récupérer la clé privée principale. La clé publique principale à son tour, est donnée à tous les membres joignant le réseau.

Phase d'enregistrement

Après la phase d'initialisation, le PKG peut commencer à distribuer aux utilisateurs leurs clés privées en fonction de leurs identités. Pour leur clé privée personnelle, [41] ont proposé que les nœuds utilisent leur contrôle d'accès au support (MAC) ou une adresse de couche réseau comme une identité lorsqu'ils communiquent avec l'autorité distribuée. Les nœuds contactent au moins t des serveurs PKG qui forment le PKG où chacun répond avec une part de la clé privée. Dès réception de t parts correctes, l'utilisateur peut calculer sa clé privée. [41] ont montré que les nœuds qui sont incapables de contacter t ou plus de serveurs peuvent se déplacer dans le réseau à la recherche de plus de parts.

Discussion et commentaires sur l'approche de la gestion de clés basée sur l'identité

Dans la phase initiale du système proposé par [41], les nœuds se prononcent sur un ensemble mutuel de paramètres de sécurité. Tout nœud qui n'est pas satisfait du choix des paramètres peut choisir de ne pas participer au réseau. [41] ont déclaré que leur système est indépendant de la négociation initiale. Cette indépendance est difficile à voir puisqu'il est de la responsabilité

1. PKG : générateur de clé publique

du protocole de gestion de clé de réussir à initialiser tous les utilisateurs du système dans un domaine [49]. Si les adversaires sont capables d'influencer la sélection des paramètres du système, ils seront en mesure de forcer les nœuds à ne pas participer au réseau.

La proposition n'aborde pas la question de savoir comment l'ensemble initial des nœuds formera le PKG ou comment un nœud va obtenir une clé privée du PKG distribué. La distribution de la clé privée maître telle que mentionnée dans [41] peut être faite en utilisant l'algorithme présenté dans [29]. L'intégration du système de génération de clés distribuées [29] dans l'algorithme de la configuration du système de signatures basées sur l'identité [18] va permettre la génération d'une clé privée maître distribuée. Un problème est noté avec l'implémentation des algorithmes d'extraction des systèmes de signature existants basés sur l'identité dans les systèmes distribués. Il est à noter que les algorithmes d'extraction [18], [8], et [39] sont conçus pour une entité obtenant une clé privée personnelle à partir d'un PKG centralisé. Tout service centralisé dans les réseaux ad hoc est un point de vulnérabilité. Les algorithmes d'extraction devront être modifiés pour la négociation d'une clé privée avec un PKG distribué vu les observations faites dans le paragraphe (Discussion et commentaires sur l'approche d'autorité de certification partiellement distribuée) sur le partage du secret, la mise à jour du secret, la redistribution du secret et les systèmes multi-signature à seuil.

La proposition n'a pas évité les points faibles de la cryptographie basée sur l'ID. Le problème majeur avec les systèmes cryptographiques basés sur l'identité est qu'ils ne cèdent que la confiance du niveau 1 [56] si la clé privée des utilisateurs est connue par l'autorité de confiance. Dans les réseaux classiques, ceci n'est pas vraiment un problème, mais dans les MANETs où l'autorité de confiance est distribuée entre les serveurs en ligne ou émulée par une entité déconnectée arbitraire, cela peut ne pas être faisable.

Supposons que le nœud peut négocier une clé privée personnelle avec un PKG distribué ; le problème majeur qui se pose est comment transférer le PKG de façon sécurisée vers le nœud demandant sa part de clé privée personnelle ? Dans le système proposé dans [41], le nœud demandeur ne partage aucun secret avec le PKG, par exemple, une clé symétrique commune, et les nœuds n'ont pas de paires de clés publique/privée. Il n'est donc pas clair comment un nœud va obtenir sa clé personnelle du PKG en la présence d'un adversaire. Ce problème ne peut être résolu que par la mise en place d'une voie de communication sécurisée ou par la pré-distribution du matériel de chiffrement commun, ce qui n'est pas idéal pour les réseaux adhoc.

La proposition échange un problème difficile par un autre. Dans le cas d'une solution pour PKI [2], la principale préoccupation est l'authentification des clés publiques. En [41], le problème est d'authentifier l'identité d'un nœud avant d'envoyer les parts de la clé privée personnelle correspondant à l'identité.

La solution est vulnérable à l'attaque man-in-the-middle sur les nœuds qui joignent le réseau [41]. Cette approche également ne peut pas briser le cycle d'interdépendance de la sécurité de routage [7].

2. PKI : infrastructure à clé publique

3.4.4 Approches basées sur le chainage des certificats

Une propositions présentées dans [15] prend un pas de plus pour répondre aux contraintes de MANET. Contrairement aux solutions précédentes, l'infrastructure de la clé publique (PKI) dans cette proposition ne nécessite aucune tierce partie de confiance. Cela rend le système adéquat aux MANET totalement auto-organisée. Chaque nœud émet ses propres certificats à d'autres nœuds de manière similaire à PGP [3] [80]. Cette proposition diffère du PGP par le fait qu'il n'y a pas des répertoires de gestion de certificats centralisés (serveurs de certificats en ligne) mais les certificats sont plutôt stockés et distribués par les nœuds dans une nature auto-organisée. Chaque nœud maintient, dans son voisinage un référentiel de certificats limité composé de nœuds certifiés comme illustré dans la Figure 3.5 [15]. Quand un nœud u veut valider le certificat d'un autre nœud v, les nœuds combinent leurs référentiels de certificat et u tente de trouver une chaîne de certificats de clés publiques valides.

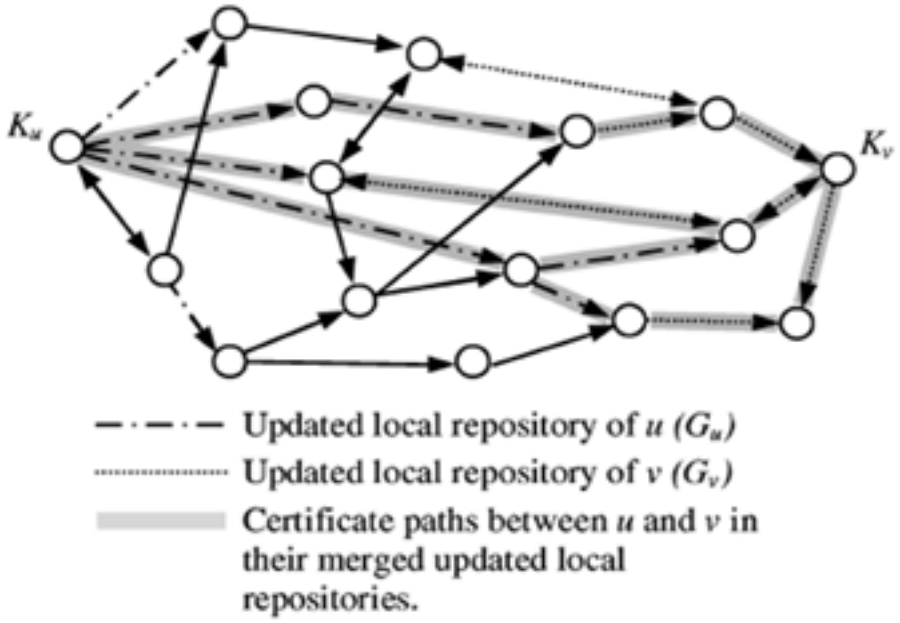


FIGURE 3.5 – Graphe de certificat et chemins entre les utilisateurs u et v dans leur répertoire marger locale

Modèle du système

[15] ont présenté leur système en termes d'un modèle abstrait. Dans leur modèle, les clés publiques et les certificats du système sont représentés comme un graphe de certificats orienté $G(V, E)$, où V et E représentent respectivement l'ensemble des sommets et l'ensemble des arêtes (Figure 3.5). Les sommets du graphe représentent les clés publiques et les arêtes représentent

3. Pretty Good Privacy

les certificats. Plus précisément, il ya une arête orientée du sommet K_u vers le sommet K_w s'il y a un certificat signé avec la clé privée de u qui lie K_w à une identité. Une chaîne de certificats à partir d'une clé publique K_uà une autre clé publique K_v est représentée par un chemin orienté du sommet K_u vers le sommet $K_v$dans G. Ainsi l'existence d'une chaîne de certificats à partir de K_v vers K_u signifie que le sommet K_v est accessible à partir du sommet K_u dans G (noté par $K_u \rightarrow_G K_v$).

Analyse du système

Phase d'initialisation

La phase initiale du système est exécutée en quatre étapes :

- chaque nœud crée une paire de clés publique/privée ;
- chaque nœud crée ses propres certificats ;
- chaque nœud envoie les certificats à d'autres nœuds et construit un graphe de certificat qui ne peut pas être mis à jour ;
- Les nœuds échangent les certificats et créent des référentiels de certificats mis à jour.

Chacune de ces étapes est illustrée dans la Figure 3.6 [15] et elle est expliquée plus en détail ci-dessous. La numérotation des étapes est gardée compatible avec la numérotation utilisés dans [15].

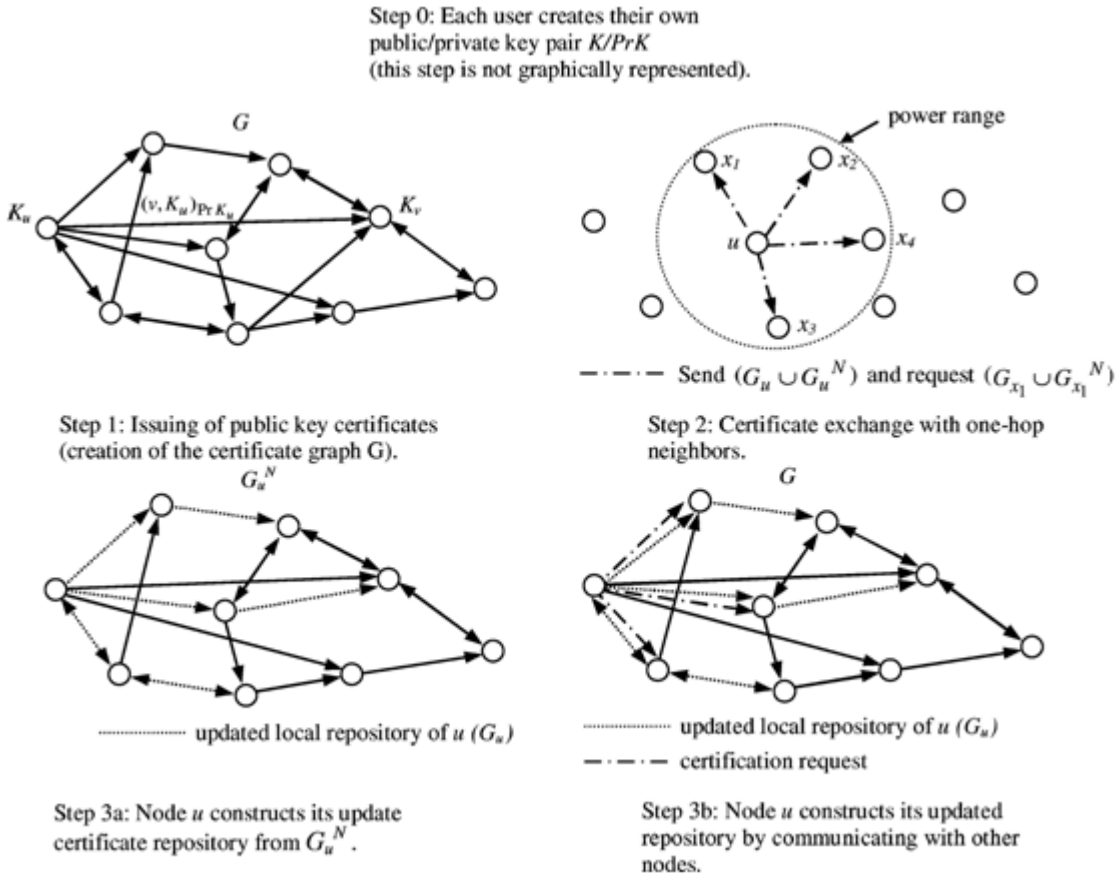


FIGURE 3.6 – Les quatre étapes dans la phase initiale de la proposition de chaînage de certificat

Etape 0 : Création de paires de clés publique/privée

de manière similaire au PGP [80], les utilisateurs créent localement leur propre clé privée et clé publique correspondante.

Étape 1 : Création de ses propres certificats et l'émission des certificats de clés publiques

Les certificats individuels des clés publiques sont délivrés par les utilisateurs eux-mêmes avec une durée de validité limitée. Quand un nœud u a confiance dans la liaison clé publique/identité du nœud v, il peut délivrer un certificat (recommandation) de se porter garant pour la liaison. Les nœuds utilisent ces certificats pour commencer la construction d'un graphe de certificats G.

Étape 2 : L'échange de certificats

Le mécanisme d'échange de certificats permet aux utilisateurs de partager et de distribuer des certificats dans leurs dépôts. Les certificats sont stockés au moins deux fois :

- par l'émetteur du certificat et
- par l'utilisateur à qui le certificat est délivré.

Le processus d'échange de certificat consiste en l'échange périodique de certificats entre les nœuds et leurs voisins. Les échanges sont asynchrones. Les utilisateurs envoient leurs sous-graphes G_u mis à jour et les sous-graphes G_u^N non mis à jour à leurs voisins qui utilisent les certificats pour créer ou étendre leurs sous-graphes non mis à jour. Le message ne contient que la valeur de hachage des certificats. Le nœud compare les valeurs de hachage avec celles qui sont dans ses référentiels et demande uniquement les certificats dont la valeur de hachage est négative. Le processus d'échange de certificat a un coût de communication faible car les échanges des certificats sont effectués localement à un seul saut seulement.

Étape 3 : La construction de la mise à jour de référentiels des certificats

Les référentiels non mis à jour (sous-graphes) fournissent des nœuds avec une vue incomplète du graphe de certificats. Un référentiel de certificats à jour est construit par le nœud u en sélectionnant un sous-graphe $G_u$de G. Ceci peut être réalisé de deux manières :

- les nœuds utilisent le même algorithme de construction de référentiel local pour explorer une partie pertinente du graphe de certificat G (étape 3a, Figure 3.6).
- ou bien les nœuds construisent un référentiel mis à jour en communiquant avec le graphe de certificat de leurs voisins (étape-3b, Figure 3.6).

Révocation de certificats

Les utilisateurs peuvent révoquer tout certificat délivré à d'autres utilisateurs s'ils perdent leur confiance dans la liaison clé publique/identité. De même les utilisateurs peuvent également révoquer leur propre certificat s'ils croient que leur clé privée a été corrompue. [15] ont proposé deux systèmes de révocation : explicites et implicites.

- Dans le système explicite, l'utilisateur émet aux nœuds de son voisinage local un message de révocation explicite. Ce sera très probablement les utilisateurs qui demandent en fonctionnement normal des certificats à partir du nœud de révocation.
- Le système de révocation implicite est basé sur le délai d'expiration des certificats. Le système suppose que les utilisateurs vont établir une communication et échangent une version mise à jour du certificat dans le délai imparti.

Discussion sur l'approche de chaînage des certificats

Le système de gestion de clés entièrement auto-organisé présenté dans [36] et [15] était le premier à donner une indication que la gestion de clés, sans aucune forme de TTP, peut être possible dans les MANETs. Le système auto-organisé est une avance sur les efforts précédents [78] ; [74] ; [47] ; [41] ; [52] pour la gestion de clés dans les MANETs puisqu'il élimine toute forme de TTP en ligne et la plupart des problèmes associés à ces systèmes. Le système entièrement auto-organisé présenté dans [36] et [15] était conçu pour les MANETs "ouverts" et donc pas vraiment comparable avec les systèmes qui supposent une autorité de confiance hors ligne. [36] et [15] ont posé un problème très difficile :

Comment peut-on explicitement authentifier la clé publique d'un utilisateur sans aucune forme d'autorité de confiance en ligne ou hors ligne ? En MANET, ce problème doit être résolu avec une connectivité sporadique (exemple : un canal infrarouge), tout en optimisant les

ressources de communication et de calcul. Le problème peut aussi être défini en termes d'établissement de confiance. Si un utilisateur A approuve le certificat d'un autre utilisateur B, l'utilisateur A a confiance que la clé publique contenue dans le certificat appartient à l'utilisateur B. Les utilisateurs A et B ont donc une relation de confiance directe. L'utilisateur A peut soutenir l'authenticité du certificat de l'utilisateur B en signant le certificat avec sa propre clé privée ; tout autre utilisateur du réseau qui approuve le certificat de l'utilisateur A approuvera également le certificat de l'utilisateur B s'il est en mesure de vérifier la recommandation de l'utilisateur A et donc peut vérifier sa signature. Si un utilisateur recommande également les certificats des autres utilisateurs, un modèle de confiance hiérarchique est créé. Le chaînage de certificats utilisé dans les PGP [80] évolue naturellement à partir d'une combinaison de modèle de confiance hiérarchique et direct. [36] et [15] ont adapté avec succès l'approche d'authentification de chaînage de certificat pour les MANETs. La principale différence entre leurs systèmes et PGP [80] est que le dernier stocke les certificats dans un référentiel centralisé. Dans le système [36] et [15], les certificats sont diffusés et stockés par tous les nœuds sans aucune assistance d'une autorité de confiance.

La relation de confiance prend du temps pour se former et nécessite une interaction des utilisateurs. Le système totalement auto-organisé [36] [15] a donc introduit un retard dans la mise en place d'associations de sécurité. En conséquence, la solution peut rencontrer un problème dans la phase initiale, lorsque le nombre de certificats délivrés est insuffisant pour produire un graphe de certificats suffisamment dense.

Une chaîne de confiance fournit une authentification faible [21] ; [1].Une hypothèse courante dans la pluparts des protocoles d'authentification distribués est que la confiance est implicitement transitive [1]. La confiance transitive signifie, par exemple, que si Alice a confiance en Bob qui lui aussi a confiance en Clark, alors Alice aura également confiance en Clark. Ceci a été montré pour être généralement faux [21]. [38] ont défini une chaîne de confiance transitive valide comme une chaîne où chaque liaison contient le même objet de confiance. [1] ont fait remarquer que les chaînes de confiance transitive valables remplissent quatre conditions en référence à l'exemple ci-dessus. Il est conclu depuis [21], [1] et [38] qu'il est très difficile de garantir la validité des chaînes de confiance transitive avec plus de deux liens. En général, les utilisateurs ne sont pas capables de prendre des décisions de sécurité intuitive. Le public a un manque de connaissances les plus élémentaires (comme ce qui est un certificat de clé public) ce qui rend tout système qui repose sur « le raisonnement des utilisateurs » concernant la sécurité vulnérable à l'attaque. Des exemples de systèmes qui permettent d'éviter ces «raisonnement de l'utilisateur» ont été présentés dans [15], [13] et [48]. Dans une chaîne de confiance de deux liens, Alice a une relation de confiance directe avec Bob, donc, basée sur cette relation de confiance directe, Alice s'appuie sur les recommandations de Bob. Dans la pratique, une relation de confiance directe implique qu'Alice et Bob se connaissent personnellement. Ceci signifie qu'Alice peut vérifier avec Bob s'ils partagent le même objectif de confiance ou les mêmes conditions de confiance. Si la chaîne est portée à trois liaisons (quatre utilisateurs), alors Alice aura une relation de confiance indirecte avec Clark, et donc ne peut pas le connaître personnellement. La relation de confiance indirecte diminue les capacités d'Alice à assurer qu'elle et Clark ont les mêmes conditions de confiance.

Par ailleurs, si une chaîne devient plus forte son lien devient plus faible. Si un nœud le long

de la chaîne est corrompu ou soumis à un comportement byzantin, il peut aboutir à une fausse authentification. Dans les réseaux ouverts ou entièrement auto-organisés, cela peut être encore plus pertinent que dans les réseaux fermés puisqu'un adversaire n'a pas à compromettre les nœuds pour participer aux mécanismes d'échange de certificat.

3.4.5 Approche de gestion de clés basée sur les clusters

Le système de gestion de clés proposé dans [52] provient de l'approche de chaînage des certificats [15]. Un modèle de réseau basé sur les clusters construit avec l'algorithme zonal [20] a été supposé. L'algorithme zonal partitionne le réseau en différents sous-ensembles en utilisant un algorithme distribué pour trouver l'arbre couvrant minimum (MST). Une fois le réseau partitionné et le MST déterminé pour chaque sous-ensemble, l'algorithme calcule les ensembles dominants faiblement connectés de chaque région. A la fin, il fixe les frontières des clusters, ceci relie les régions qui ne peuvent pas être jointes par l'inclusion des nœuds additionnels dans les ensembles. Les nœuds regroupés ensemble dans la même région d'un groupe se voient attribuer un identifiant unique. Les nœuds apprennent les identifiants ID des groupes des autres nœuds en échangeant des messages.

Système/Modèle de confiance

Chaque utilisateur est responsable de la création de sa propre paire de clés publique/privée et de la génération de son propre certificat. Tout nœud peut signer le certificat de clé publique d'un autre nœud dans le même groupe sur demande. Les nœuds sont supposés avoir certains éléments de surveillance qui leur permettent d'observer le comportement des autres nœuds dans leur groupe et d'affecter à chaque nœud une valeur de confiance. La valeur de confiance est définie comme un indicateur d'authentification qui représente la garantie avec laquelle un nœud demandeur s peut obtenir la clé publique correcte d'un nœud cible t. La confiance entre les nœuds dans le même groupe est dénommée confiance directe. La relation de confiance entre les nœuds dans les différents groupes est considérée comme confiance sur recommandation. Le modèle de confiance est montré dans la Figure 3.7 [52]. Chaque nœud doit donc avoir une table de confiance pour stocker les valeurs de confiance et les clés publiques associées des nœuds qu'il "connait" dans le réseau.

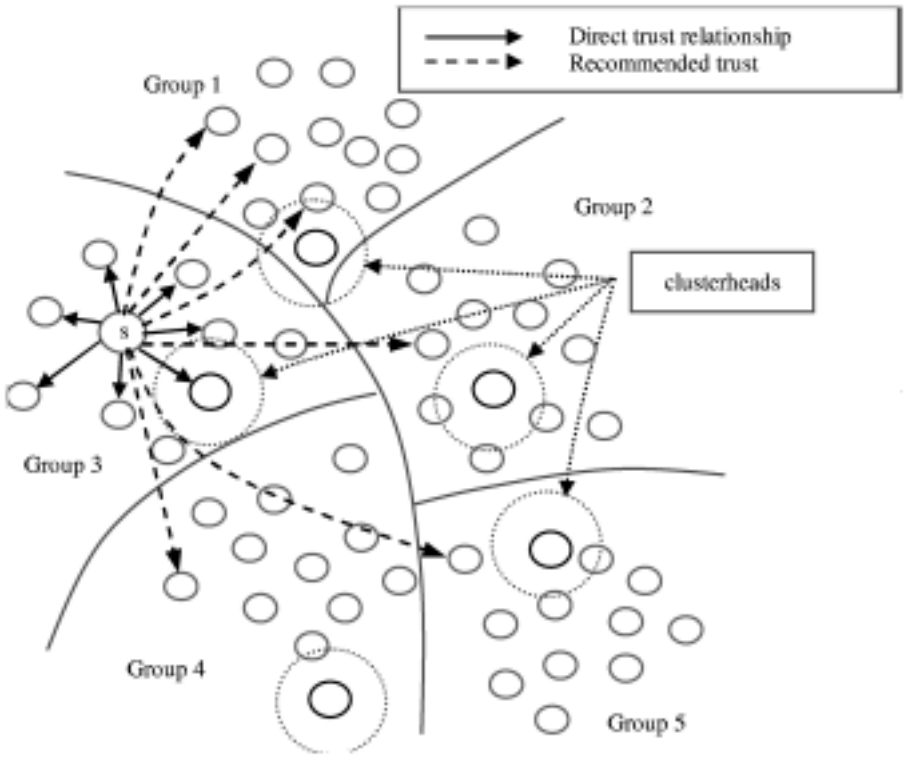


FIGURE 3.7 – Modèle de confiance basé sur les clusters.

La certification de la clé publique et la mise à jour de la valeur de confiance

Si un nœud s veut obtenir le certificat de la clé publique de certains autres nœuds t, s éffectue la procédure suivante [52] :

1- Le nœud s regarde l'ID du groupe de t, noté φ_t.

2- Le nœud s consulte sa table de confiance et trie les valeurs de confiance des nœuds. Soit $i_1, ..., i_n \in I$, où i_n désigne le nœud avec la plus grande valeur de confiance.

3- Le nœud s envoie alors des messages de demande de certification à chaque nœud du sous-ensemble I. Ces nœuds sont aussi appelés des introducteurs.

4- Le nœud s recueille les messages de réponse $m \in M$ de I où $m = \{P_{k_t}, V_{i_k,t}, ...\} Sk_{i_k}$ avec P_{k_t} désigne la clé publique de t, $V_{i_k,t}$ désigne la valeur de confiance à partir de i_k à t et Sk_{i_k} désigne la clé secrète de i_k qui est utilisée pour générer une signature sur m.

5- Le nœud s compare les clés publiques reçues de I et suit les votes à la majorité. Soit $i_{good} \in I_{good}$ et $i_{bad} \in I_{bad}$, où I_{good} sont les nœuds considérés honnêtes et I_{bad} le reste des nœuds perçus malhonnêtes. Le nœud s considère la clé publique de t reçue de I_{good} authentique.

6- Les valeurs de confiance des nœuds $\in I_{bad}$ sont réduites à zéro. Le nœud s calcule et met à jour la valeur de confiance de t en utilisant l'équation suivante où i_k désigne les nœuds et n le nombre de nœuds dans I_{good} :

$$V_{s,i_k,t} = V_{s,i_k} \odot V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}} \quad (1)$$

$$et$$

$$V_t = 1 - \prod_{k=1}^{n} (1 - V_{s,i_k,t}) \quad (2)$$

La certification de la clé publique et les procédures de mise à jour de la valeur de confiance sont illustrés dans la Figure 3.8 [52] et la Figure 3.9 [52] respectivement.

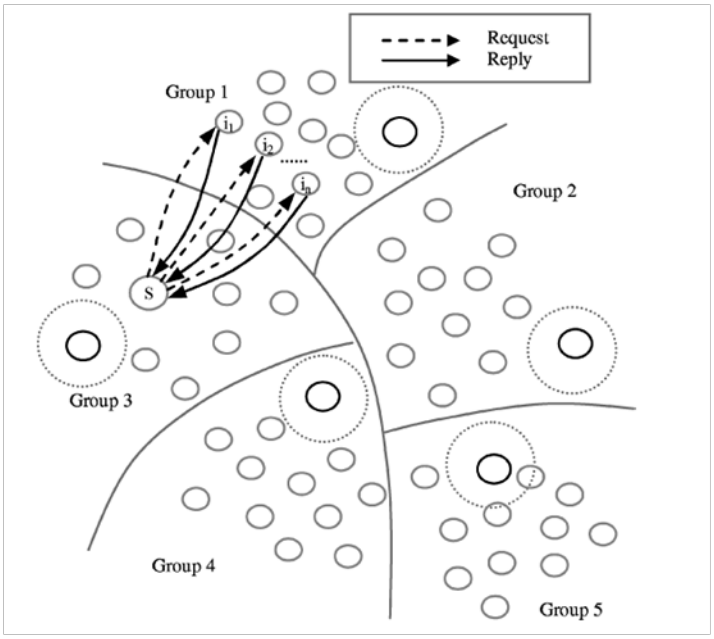


FIGURE 3.8 – Certification de clé publique

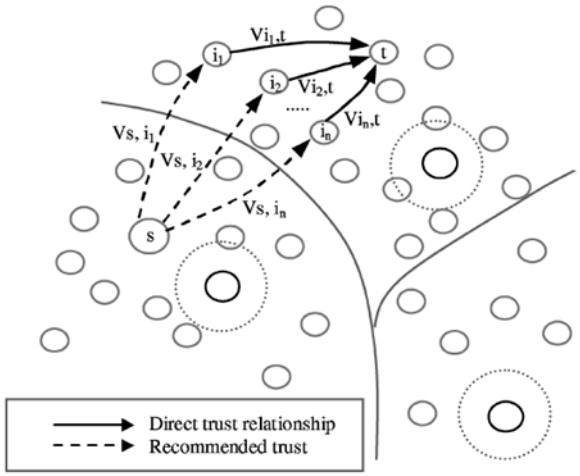


FIGURE 3.9 – La mise à jour de la valeur de confiance

Explication de l'étape 6

Après avoir reçu, décrypté et comparé les valeurs de confiance de I à l'étape 5, s peut calculer la nouvelle recommandation de la relation de confiance de s à t par l'intermédiaire des nœuds dans $i_k \in I$ en utilisant l'équation (1). Les nœuds $i_{bad} \in I_{bad}$ ne font pas de contribution puisque leurs valeurs de confiance sont réduites à zéro. L'opérateur $\odot$ est défini dans [5] et il est donné comme $V_1 \odot V_2 = (1 - (1 - V_2)^{V1})$. Dans [5], les dérivés de la formule $V_1 \odot V_2$ont été utilisés pour calculer la nouvelle relation de confiance entre V_1 et V_2 basée sur les valeurs de confiance directe et les valeurs de recommandation. Une fois $V_{s,i_k,t} \forall i_k$ calculée, s peut calculer la dernière valeur de confiance V_t de$t$ après la certification de la clé publique en utilisant l'équation (2).

Discussion sur l'approche de gestion de clés basée sur les clusters

Dans cette discussion, nous allons nous concentrer sur certaines questions liées à l'analyse des systèmes de gestion de clés basée sur les clusters.

- Lorsque les MANETs s'agrandissent, les calculs des routes deviennent de plus en plus coûteux [20]. La tâche des algorithmes de routage peut être simplifiée en confinant la découverte des routes à une sous-structure du réseau. En introduisant le clustering dans le réseau, les messages locaux peuvent être transmis sur des chemins courts dans le même cluster tandis que les messages pour les longues distances traversent de longs chemins à partir d'un cluster à un autre. On attribue pour chaque cluster un chef en tant que représentant du cluster. Le chef du cluster prend la responsabilité de participer dans le calcul d'itinéraire inter-cluster et de transférer des messages de longue distance. Le transfert des messages longue distance nécessite pour le chef du cluster l'utilisation de plus de puissance de transmission, ce qui contribuera significativement à l'épuisement des ressources d'énergie du nœud [31].
- La nature dynamique des MANET rend le clustering très problématique. La principale préoccupation est la sélection, la configuration, la maintenance et le remplacement du chef

de cluster. Le rôle d'un chef du cluster doit pouvoir être effectué par n'importe quel nœud du réseau. Les chefs des clusters ne sont donc pas différents des autres nœuds dans les MANETs et exposent des caractéristiques similaires. A cause d'une faible connectivité et les échecs de route, les clusters peuvent également se partitionner. L'entretien fréquent des chefs des clusters et les membres de cluster est inévitable et peut causer des calculs non pratique et des surcouts de communication.

- La mission des chefs des clusters est très difficile car les nœuds ne peuvent pas prendre volontairement cette responsabilité [12] ou avoir forcément les capacités nécessaires pour supporter cette charge supplémentaire.
- Les chefs des clusters sont devenus des points d'attaque centraux très prisés par les adversaires. Par exemple, un adversaire qui peut s'imposer comme un chef de cluster contrôle efficacement l'ensemble du cluster ce qui ne fonctionnera pas dans les MANETs entièrement auto-organisés.

[52] ont simulé leur système en GloMoSim [76]. Les 100 nœuds dans le réseau de 600m × 600m ont été divisés en cinq groupes à mobilité variée entre 0-10 m/s. L'objectif de la simulation était de tester le système de gestion de clés publiques en présence de nœuds malveillants. Un pourcentage m des nœuds a été mis en adversaire et retourne donc toujours des clefs publiques et des valeurs de confiance fausses. Le système n'a pas été mis en œuvre en parallèle avec un algorithme de clustering mais a été divisé en groupes fixes. Il a donc été supposé que la structure de clustering du réseau a déjà été construite. Si le système devait être mis en œuvre sans l'algorithme de clustering (par exemple, l'algorithme zonale [20]), il peut être anticipé que la simulation serait de produire des résultats significativement différents. Les systèmes doivent toujours être simulés dans un cadre réaliste car la performance ne peut pas être déterminée si le protocole de l'étude est simulé avec des systèmes centraux à son fonctionnement. Ces systèmes complémentaires à leur tour peuvent consommer des ressources supplémentaires du réseau ce qui va certainement influencer les résultats obtenus à partir des simulations. Les systèmes de gestion de clés à base de cluster ne résolvent pas les inconvénients de l'approche du chaînage des certificats [15] . Les systèmes de gestion de clés dans les MANETs ne doivent pas s'appuyer sur les fonctionnalités et le fonctionnement correct des autres systèmes. Une approche de la gestion de clés basée sur les clusters repose sur l'efficacité d'un système de clustering. Ainsi, les attaques sur le protocole de clustering introduisent d'autres vulnérabilités qui peuvent compromettre le système de gestion de clés et par conséquent les mécanismes de sécurité du réseau en entier.

3.4.6 La gestion de clés basée sur le pré-déploiement

Les limites de la mémoire, de l'énergie et de la puissance de calcul des nœuds capteurs proviennent des contraintes imposées à leur coût et à leurs dimensions physiques. Dans un déploiement incontrôlé, qui est normalement le cas avec les réseaux de capteurs à grande échelle, les nœuds sont aléatoirement dispersés sur la zone cible. Cela implique une topologie imprévisible du réseau. Par ailleurs, des capteurs peuvent être rajoutés et soustraits après le déploiement et peuvent aussi rencontrer des environnements réseaux hostiles [27]. Ces caractéristiques entraînent la pré-distribution des clés qui est la seule technique pratique connue pour l'établissement de clés appropriés aux réseaux de capteurs sans fil à grande échelle [27]. Les réseaux de

capteurs rentrent donc dans les MANETs basés sur l'autorité.

Modèle du système

[27] ont proposé un système de pré-déploiement aléatoire des clés pour les réseaux de capteurs distribués. Une autorité déconnectée charge chaque nœud capteur avec une liste de clés avant son déploiement. Une fois dans l'environnement opérationnel, deux nœuds trouvent une clé commune dans leurs listes et utilisent la clé partagée pour sécuriser les communications ultérieures. Le système ne garantit pas une clé partagée entre toutes les paires de nœuds mais soutient plutôt l'existence d'une telle clé avec une certaine probabilité. Les nœuds qui ne peuvent pas trouver une clé commune peuvent établir une association de sécurité via une séquence de connexions sécurisées. Le système de gestion de clé se compose de trois parties :

- la distribution,
- la révocation,
- et l'auto-révocation de la clé (rekeying).

Le mécanisme de distribution des clés est encore subdivisé en trois phases :

- pré-distribution de clés,
- la découverte de la clé partagée,
- et l'établissement du chemin de la clé.

En [27], des nœuds contrôleurs spécialisés fournissent des services de révocation de la clé pour les réseaux de capteurs.

Analyse des systèmes

La distribution des clés

La phase de pré-distribution des clés repose sur les étapes hors ligne suivantes :

- L'autorité déconnectée A génère une plage P de clés et de leurs identifiants.
- Pour chaque capteur,A tire au hasard k clés de P sans remplacement. Les k clés forment le porte-clés du nœud capteur.
- A charge chaque capteur avec son porte-clés et les identificateurs des clés.
- Les identificateurs des clés de chaque porte-clés et l'identité du nœud capteur associé sont stockés dans un nœud de confiance.
- Pour finir, A charge le i^{ieme}nœud contrôleur avec une clé partagée $K^{ci} = E_{K_x}(ci)$ où $K_x = K_1 \oplus, ..., \oplus K_k$ où K_i représente les clés du porte-clés du nœud et l'identité du contrôleur est donnée par ci. E_{K_x} est un cryptage avec la clé du nœud K_x.

La phase de la découverte de la clé partagée commence après le déploiement des capteurs. Chaque nœud diffuse ses identificateurs clés en texte brut. Les nœuds découvrent les clés partagées dans leur portée de transmission par les moyens de comparaison. Alternativement, les nœuds peuvent cacher les modèles des clés partagées qui permettent de découvrir les clés privées partagées. L'exemple donné par [27] fonctionne comme suit :

Chaque nœud diffuse une liste $\alpha, E_{k_i}(\alpha)$ où α est un challenge aléatoire et K_i pour $i = 1, ..., k$ sont toutes les clés dans le porte-clés du nœud. Le décryptage de $E_{k_i}(\alpha)$ avec la bonne clé révélera α et garantit l'établissement d'une clé partagée avec le nœud diffusant. La probabilité que deux nœuds partagent une clé commune dans leur porte-clés a été donnée par :

$$\frac{k!(P-k)!(P-k)!}{P!k!(P-2k)!} \tag{3}$$

À l'issue de la phase de la découverte de la clé partagée, les nœuds peuvent mettre en place un graphe connexe des liaisons sécurisées. La phase d'établissement du chemin de la clé est utilisée pour établir un chemin de clé entre les nœuds qui ont échoué à trouver une clé partagée dans leur portée de transmission au cours de la phase de découverte de la clé partagée. Si le graphe est connexe, un nœud peut trouver un chemin vers le nœud voisin avec lequel il a échoué à découvrir une clé commune. Le nœud peut trouver un chemin de clé et envoyer la clé vers le nœud voisin via un ou plusieurs nœuds de confiance intermédiaires.

La révocation des clés

Les nœuds contrôleurs (ci) ont des capacités améliorées telles que la mobilité et les longues portées de transmission. Pour révoquer les clés d'un nœud spécifique, le nœud contrôleur diffuse une liste signée des identificateurs des clés du nœud cible. Une clé K_e est utilisée pour signer le message. Le contrôleur unicast un chiffrement de K_e à tous les nœuds. K^{ci} qui est partagée entre le nœud et le contrôleur, est utilisée pour crypter K_e. Les nœuds décryptent $E_{K^{ci}}(K_e)$ et utilisent K_e pour vérifier la signature dans la liste des identificateurs des clés. Les nœuds localisent et suppriment eux-mêmes les clés affectées de leurs porte-clés. Si les clés révoquées affectent tous les autres liens sécurisés, le nœud redémarre la phase de découverte de la clé partagée et si nécessaire la phase d'établissement du chemin de la clé.

L'auto-révocation (rekeying)

Les clés expirées sont retirées du porte-clés du nœud. Dans le cas où une ou plusieurs connexions sécurisées sont perdues, le nœud redémarre la phase de découverte de clé partagée et éventuellement la phase d'établissement du chemin de la clé.

Discussion et commentaires sur la pré-distribution de clés

Le schéma proposé dans [27] identifie la pré-distribution de clés comme la technique la plus pratique pour établir une connectivité sécurisée dans les réseaux de capteurs. Après que le nœud ait accompli les phases de la découverte de clé partagée et d'établissement du chemin de la clé, la nature probabiliste du mécanisme de distribution des clés peut laisser un nœud avec un graphe partiellement connecté. [81] ont proposé qu'un nœud doive augmenter sa portée de transmission ou demander aux nœuds voisins de partager leurs communications pour un petit nombre de

sauts. Le mécanisme, appelé « extension de la portée », impose aux nœuds d'augmenter progressivement leur puissance d'émission et de répéter les phases de découverte de la clé partagée et d'établissement du chemin de la clé jusqu'à ce qu'ils soient entièrement connectés. [81] ont plus tard amélioré dans [27] ont proposé un q-composite système de pré-distribution aléatoire des clés qui améliore la résistance du réseau face aux plus faibles adversaires. Ceci, cependant, génère une grande vulnérabilité aux attaques à grande échelle. En revanche dans [27], les nœuds doivent trouver q clés communes avant qu'ils puissent établir une connectivité sécurisée. [82] ont combiné le système en [6] avec l'approche de la pré-distribution aléatoire des clés présentée dans [27] et [81]. Contrairement à l'espace clé unique (utilisé dans le système [6]), [82] ont utilisé des espaces clés multiples. Le système de Blom garantit un graphe complet car n'importe quelle paire de nœuds peut établir une clé commune. Due à l'extension de l'espace de clé unique à clés multiples, [82] ont sacrifié une connectivité complète pour une meilleure résistance à la capture des nœuds et la réduction du stockage de l'informations. [82] ont également montré que les espaces clés multiple permettent une meilleure résistance par rapport aux précédents schémas [27] ; [81] en utilisant la même quantité de mémoire.

[45] ont amélioré la résistance et l'extensibilité de [27] et [81]. [45] ont construit leur système sur la pré-distribution aléatoire des clés et la pré-distribution des clés basée sur le polynomial [10] et le système est essentiellement équivalent à celui de [82]. Le système permet à tout groupe de v parties d'établir une clé commune. La clé est parfaitement secrète avec le respect de la coalition des t autres parties [82]. [82] ont souligné que [10] atteignent seulement la borne inférieure du stockage de la mémoire si tous les groupes de taille v peuvent calculer une clé commune et si le réseau résiste à la plupart des t nœuds capturés. [45] ont également proposé un système basé sur une grille à clés pré-distribuées. Un serveur de configuration construit une grille $m \times m$ avec un ensemble de $2m$ polynomiaux où $m = \left\lceil \sqrt[n]{N} \right\rceil$. Le paramètre n est la dimension de l'hyper cube et N le nombre total des capteurs dans le réseau. Chaque capteur est attribué à un croisement unique sur la grille et chargé avec le partage polynomial. En utilisant ces partages, les nœuds capteurs peuvent effectuer la découverte des polynômes partagés et la découverte du chemin de la clé. [46] étendent le système basé sur une grille à un système multidimensionnel basé sur l'hyper cube. Une caractéristique intéressante de ce système est qu'il peut fournir une parfaite résistance contre les nœuds capturés, par exemple, lorsque $n = 4$ et $N = 20000$.

Il est clair que le sous-ensemble aléatoire assigné au système de pré-distribution des clés de [45], [46], et [82] améliore significativement la résistance de la pré-distribution de clés par rapport aux efforts précédents. Cependant, comme souligné dans [79], [45] et [19], deux principales propriétés de ces méthodes d'affectation de sous ensembles aléatoires peuvent être améliorées :

1- Après qu'un certain nombre de capteurs ait été corrompu, la fraction de liens corrompus entre les nœuds non corrompus augmente de façon exponentielle si d'autres nœuds sont encore corrompus.

2- Afin d'assurer un graphe connexe, ces régimes requièrent un déploiement des capteurs avec une densité suffisante. Une densité suffisante des nœuds ne peut pas être garantie si on suppose que les nœuds capteurs peuvent échouer (par exemple à cause de l'épuisement de la batterie). Bien que les systèmes basés sur la grille présentés dans [45] et [46] peuvent faire face aux mauvaises connectivités, ils montrent également une dégradation exponentielle de la sécurité après qu'une certaine fraction de

nœuds ait été corrompue.

PIKE [4] [19] améliore la condition de densité des capteurs des sous-ensembles aléatoires assignés au système de pré-distribution de clés en utilisant une approche similaire au système basé sur une grille proposé par [45] et [46]. Chacun des n nœuds partage une paire de clés unique avec $\sqrt{n}$ d'autres capteurs (si $[\sqrt{n}]$ n'est pas carré). Les nœuds qui ne partagent pas les paires de clés pré-chargées utilisent des nœuds de confiance intermédiaires pour établir les chemins des clés [19]. [79] avancent que le réseau PIKE à large communication pour l'établissement du chemin des clés est inapproprié aux réseaux des capteurs à grande échelle.

D'autres systèmes [46] ; [83] ; [35] utilisent le déploiement d'informations pour améliorer les méthodes probabilistes de pré-distribution des clés. Cependant, toutes ces approches supposent que le déploiement des emplacements des capteurs peut être prédéterminé dans une certaine mesure [79] ; [45]. Les propositions récentes [79] ; [45] ont soutenu que la connaissance précise a priori des emplacements des capteurs est peu probable en pratique. [79] exploitent un modèle de déploiement basé sur les groupes pour améliorer les performances et la résistance des approches probabilistes. Comme le suggèrent les précédents systèmes [46] ; [83] ; [35], la probabilité que les capteurs au sein du même groupe soient voisins après le déploiement est élevée. L'idée principale est de pré-charger chaque capteur avec un ensemble de clés minutieusement choisies.

Chaque paire de capteurs dans le même groupe partage une clé commune. Les techniques de pré-chargement des clés garantissent qu'après le déploiement les groupes soient solidement liés via des nœuds associés ; chaque nœud peut partager une clé commune unique avec jusqu'à t agents dans tout autre groupe. Les nœuds qui ne sont pas dans le même groupe utiliseront donc au plus deux intermédiaires de confiance pour mettre en place un chemin clé. [45] ont proposé un modèle de déploiement basé sur le groupe pour éliminer le besoin d'informations sur l'emplacement prévu et pour améliorer la sécurité et la performance des systèmes existants de pré-distribution des clés. Le cadre général requiert des nœuds d'être pré-chargés avec une instance de la clé de pré-distribution du groupe D_i. Cela permet à des nœuds au sein du même groupe d'utiliser des techniques directes d'établissement de clés. Les nœuds sont également pré-chargés avec une instance D_i' de la clé de pré-distribution du groupe croisé. Les nœuds avec les mêmes D_i' forment un groupe croisé. Les nœuds qui ne sont pas dans le même groupe doivent trouver une "passerelle" entre leurs groupes respectifs. Les passerelles sont formées par deux capteurs du même groupe croisé. Le domaine de la pré-distribution des clés pour les réseaux de capteurs nécessite actuellement une analyse exhaustive des systèmes existants en termes de sécurité, de performance et de mise en œuvre.

3.4.7 Approche de gestion de clés basées sur la mobilité

[14] [16] ont proposé des systèmes d'établissement de clé à mobilité assistée pour les MANETs. Contrairement aux sous-ensembles précédemment discutés, les protocoles de [14] [16] introduisent un changement dans le paradigme des tentatives antérieures pour la gestion de clés dans les MANETs. La plupart des régimes de gestion de clés existants pour les MANETs tentent de modifier des solutions adaptées pour les réseaux filaires traditionnels qui ne peuvent pas être

4. Peer intermidiairies for key establishment

toujours idéaux pour les MANETs. La proposition de [14] [16] est d'établir des systèmes de clés Peer-to-Peer qui reposent sur la mobilité des utilisateurs pour apporter des nœuds dans leur portée de transmission. Cela leur permet d'échanger leur matériel de chiffrement sans compter sur une infrastructure de routage sécurisée et rompt efficacement le cycle d'interdépendance de sécurité de routage [7].

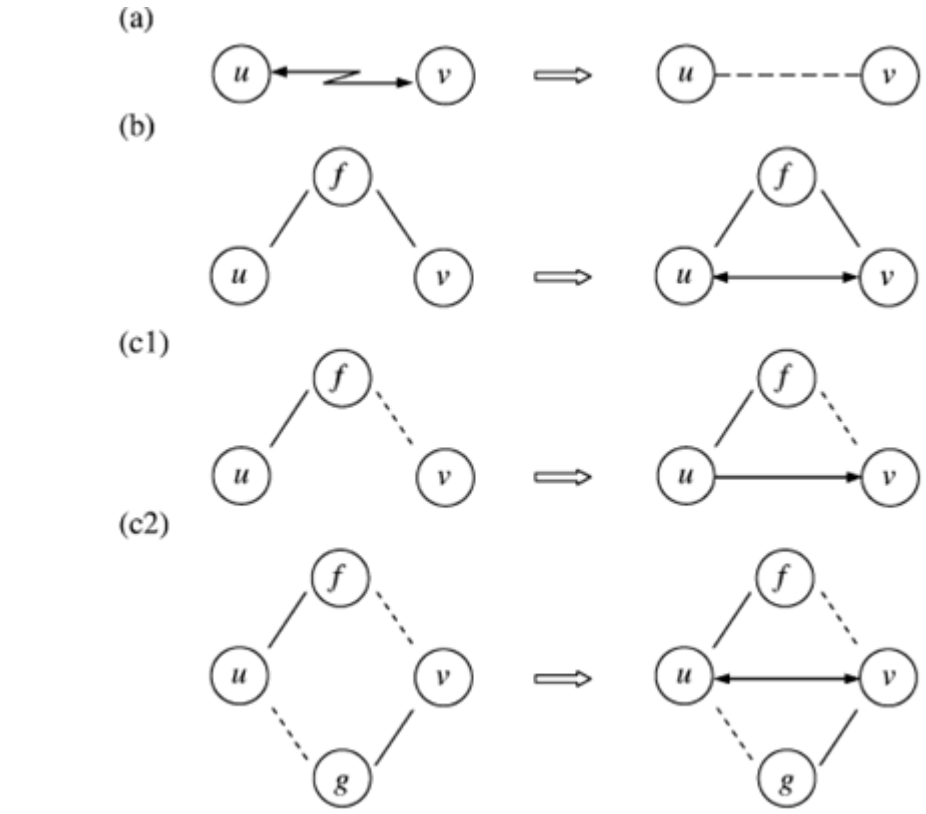
Modèle du système

[14] [16] ont examiné deux modèles : un modèle entièrement auto-organisé et un modèle avec une autorité de confiance déconnectée. Si un système de cryptographie à clé publique est utilisé, deux utilisateurs u et v partagent une association de sécurité à deux voies s'ils échangent leurs triplets (U, k_u, a_u) et (V, k_v, a_v), où (U, V) sont les noms des utilisateurs u et v, (k_u, k_v) sont leurs clés publiques, et (a_u, a_v) sont leurs adresses des nœuds. Dans une configuration à clé symétrique, les clés publiques sont remplacées par une clé partagée k_{uv}. Les utilisateurs sont équipés de connexion sans fil avec un canal latéral intégré (comme une interface infrarouge). Le canal latéral est utilisé pour créer des associations de sécurité lorsque les utilisateurs se rencontrent physiquement dans le réseau. Ceci constitue une authentification visuelle pour les utilisateurs et leur permet de lier les noms d'utilisateur au matériel de chiffrement. Le mécanisme de l'établissement de l'association de sécurité peut être amélioré par l'utilisation de nœuds amis [16].

Analyse des systèmes

Approches de la clé publique

La Figure 3.10. illustre les trois principaux mécanismes d'établissement de clés proposés dans [14][16]. Le premier [mécanisme (a)] permet aux utilisateurs d'établir une association de sécurité directement sur le canal latéral sécurisé lors d'une rencontre physique. Le canal latéral assure l'intégrité des données en éliminant l'adversaire actif. Le couplage du mécanisme(a) avec la confirmation de la clé et une défense contre les attaques répétées aboutit au protocole 1 décrit ci-dessous [14] [16] :



Échange de matériel de chiffrement via un canal latéral sécurisé.
Association sécurisé établie directement via le canal latéral sécurisé.
Associations de sécurités existantes établies entre les amis.
Association de sécurité à un seul sens établie via les amis.
Association de sécurité à deux sens établi via les amis.

FIGURE 3.10 – Etablissement des associations de sécurité directe et amis assistés

Protocole 1 : Mécanisme(a)

msg1 (canal latéral sécurisé)	$u \rightarrow v : a_u \| [\xi_u = h(r_u \| U \| K_u \| a_u)]$
msg2 (canal latéral sécurisé)	$v \rightarrow u : a_v \| [\xi_v = h(r_v \| V \| K_v \| a_v)]$
msg3 (canal radio)	$u \rightarrow v : r_u \| U \| K_u \| a_u$
msg4 (canal radio)	$v \rightarrow u : r_v \| V \| K_v \| a_v$
	$u : h(r_v \| V \| K_v \| a_v) = \xi_v ?; V ?; match(k_v, a_v)$
	$v : h(r_u \| U \| K_u \| a_u) = \xi_u ?; U ?; match(k_u, a_u)$
msg5 (canal radio)	$u \rightarrow v : \sigma(r_v \| U \| V)$
msg6 (canal radio)	$v \rightarrow u : \sigma(r_u \| V \| U)$

Dans msg1 et msg2, les utilisateurs échangent leurs adresses (a_u, a_v), les valeurs de hachage (ξ_u, ξ_v) de leurs nombres aléatoires et leurs triplets. Leurs adresses sont nécessaires afin d'échan-

ger le matériel de chiffrement sur l'interface radio dans les étapes suivantes. Dans msg3 et msg4, les utilisateurs échangent leurs triplets et nombres aléatoires sur l'interface radio. Chaque nœud vérifie si le hachage des nombres aléatoires et des triplets reçu au cours de la liaison radio correspond aux valeurs de hachage reçues sur le canal latéral. A la fin des messages, les utilisateurs s'envoient une signature ($\sigma()$) sur la liaison radio. La vérification des signatures avec les clés publiques reçues sert de preuve que u et v connaissent les clés privées correspondantes. Il faut noter que les triplets et les signatures auraient pu être échangés sur le canal latéral mais le protocole 1 minimise la quantité de données transmises sur ce canal en envoyant les hachages (ξ_u, ξ_v) comme un code de contrôle d'intégrité sur le canal latéral et cela permet à u et v d'échanger le reste des informations sur l'interface radio en éliminant la possibilité de l'attaque « man-in-the-middle ».

Mécanisme (b) : utilise un ami commun f pour générer et distribuer à chaque nœud de nouveaux certificats. Puisque f partage le matériel de chiffrement à la fois avec u et v, l'utilisateur peut vérifier les certificats reçus de f.

Mécanisme (c1) : améliore l'accord de clé lors de rencontres physiques avec des associations de sécurité d'amis et il est simplement une combinaison des mécanisme (a) et (b) détaillés ci-dessus. Ce mécanisme peut être utilisé pour établir une association de sécurité à sens unique ou à deux voies entre des utilisateurs u et v.

Mécanisme (c2) : il sera discuté ultérieurement, il est plus appliqué dans une configuration à clé symétrique.

Approches clés symétriques

Dans un cadre de clé symétrique, les trois mécanismes illustrés dans la Figure 3.10 restent applicables mais dans un contexte différent. Avec le mécanisme (a), les utilisateurs utilisent le canal latéral pour échanger toutes les informations nécessaires du matériel de chiffrement afin de mettre en place une clé partagée entre eux. Le canal latéral dans la configuration à clé symétrique doit également assurer la confidentialité en plus de l'intégrité des données. Pour éviter l'attaque des adversaires passifs, les utilisateurs doivent être informés d'activer leur canaux latéraux qui n'ont pas d'utilisateurs à une portée sécurisée. Dans le mécanisme (b), les utilisateurs ont un ami commun f qui joue le rôle d'une autorité de confiance ou d'intermédiaire de confiance. Il y a des protocoles bien établis qui peuvent être utilisés dans une telle configuration [49]. Le mécanisme (c2) peut être utilisé si u et$v$ ne partagent pas un ami commun et au cas où ils ne veulent pas que la tierce partie de confiance connaisse leur clé partagée. Un ami de u appelé f et un ami de v appelé g sont utilisés par u et v comme deux chemins séparés pour échanger les contributions des clés. Le protocole 2 [14] [16] présenté ci-dessous explique le mécanisme (c2) plus en détail :

Protocole 2 : Mécanisme (c2)

msg1 $u \rightarrow v : f, r_u$
msg2 $v \rightarrow u : g, r_v$
msg3 $u \rightarrow g : u, d_{u \rightarrow g}, request, v, k_u, r_v k_{ug}$
msg4 $g \rightarrow v : g, d_{g \rightarrow v}, reply, u, k_u, r_v k_{vg}$
msg3' $v \rightarrow f : v, d_{v \rightarrow f}, request, u, k_v, r_u k_{vf}$
msg4' $f \rightarrow u : f, d_{f \rightarrow u}, reply, v, k_v, r_u k_{uf}$
$uv : k_{uv} = h(k_u || k_v)$

Dans le protocole 2, les utilisateurs u et v utilisent les messages 1 et 2 pour échanger des nombres aléatoires (r_u, r_v) et les noms (f, g) de leurs amis. Dans les messages 3 et 4 et les messages 3' et 4' u envoie k_u à v via g quand v envoie k_v à u via f. Tous les messages sont cryptés avec une clé symétrique partagée k_{xy} où $x \in [u, v]$ et $y \in [g, f]$. Afin d'éviter toute ambiguïté, chaque message inclut la direction d et le but du message (demande ou réponse). Les utilisateurs u et v génèrent une clé partagée k_{uv} en prenant le hachage de la concaténation de leurs contributions individuelles.

Discussion et commentaires sur les approches de gestions des clés basées sur la mobilité

La caractéristique principale des réseaux ad hoc est le manque d'infrastructures. Les nœuds, par conséquent, sont responsables de toutes les fonctionnalités réseau dont le routage qui est le plus important. Dans les réseaux ad hoc stationnaires, les nœuds peuvent éprouver des ruptures fréquentes avec l'augmentation du trafic dans le réseau. La mobilité des nœuds augmente significativement la fréquence de ces ruptures. Cette connectivité sporadique entraine une faible disponibilité et des coûts de communication élevés pour les systèmes de gestion de clés qui reposent sur l'infrastructure de routage. Une autre raison pour ne pas s'appuyer sur l'infrastructure de routage est que toute attaque sur le protocole de routage peut rendre le système de gestion de clés non sécurisé. Le cycle d'interdépendance de la sécurité de routage [7] force les systèmes de gestion de clés à être indépendants du mécanisme de routage dans tous les cas. Le contournement de ce problème nécessite un changement complet avec le respect des solutions de gestion de clés trouvés dans les réseaux filaires traditionnels. Le fait que tous les réseaux auto-organisés ne supportent aucune forme d'autorité de confiance, pas même pendant l'initialisation hors ligne, ajoute une nouvelle dimension à ce problème. [14] [16] ont détaché le système de gestion de clés de l'infrastructure de routage en exploitant la mobilité des utilisateurs. La caractéristique de mobilité des MANET considérée comme un facteur limitant sera une aide aux mécanismes d'établissement des clés. La dépendance de la mobilité pour amener les utilisateurs à une «portée sécurisée» afin d'utiliser leurs canaux latéraux sécurisés pour l'établissement de clés est l'inconvénient majeur de [14][16]. Il faut remarquer que la mise en place d'un nombre suffisant d'associations de sécurité peut prendre un certain temps. Les résultats de simulations montrent que le temps de convergence diminue avec l'augmentation de la mobilité des utilisateurs. La proposition est donc de trouver une forte application comme une solution complémentaire à d'autres solutions de gestion de clés et qui est adaptée pour établir

des associations de sécurité dans la couche application dans un cadre d'auto-organisés [16].

3.4.8 Approches de gestion de clés parallèle

[75] ont proposé une approche de gestion de clés multiples en combinant une autorité de certification distribuée et le chaînage des certificats. La proposition connue sous le nom de « gestion de clé composite » est basée sur deux principes fondamentaux : tout d'abord, la gestion de clés doit être partagée entre plusieurs nœuds, et deuxièmement, une tierce partie de confiance est requise. Comme proposé dans [15], les certificats sont stockés et distribués par les nœuds dans une nature auto-organisée. [75] ont montré comment une DCA peut être utilisée en parallèle avec le chaînage de certificat pour éliminer certains points faibles de l'approche de chaînage de certificats. L'approche augmente la disponibilité du service de gestion de clés puisque les nœuds peuvent utiliser leurs services pour obtenir le matériel de chiffrement.

Analyse des systèmes

Métriques d'authentification

En introduisant une métrique d'authentification (valeurs de confiance), une tentative est faite pour offrir aux utilisateurs un outil de calcul du niveau de confiance qu'ils peuvent placer dans une instance d'authentification. Donc les utilisateurs attribuent une valeur de confiance pour les certificats basés sur leur relation avec le propriétaire du certificat. Dans [75], la valeur de confiance a également été étendue pour intégrer la DCA comme une tierce partie de confiance.

Modèle de confiance

Dans l'approche chaînage des certificats [15], les utilisateurs forment une chaîne de confiance en délivrant des certificats à d'autres nœuds dans le réseau avec lequel ils ont une certaine relation ou ont des raisons suffisantes pour faire confiance à la liaison entre l'identité du nœud et la clé publique. [75] ont illustré ce concept au moyen d'un exemple : si Alice est certaine que Bob est le propriétaire d'une paire de clé publique / privée, Alice délivre un certificat contenant l'ID de Bob, la clé publique de Bob et d'autres attributs tels le paramètre de la durée de vie du certificat. Alice génère alors une signature numérique sur le certificat de Bob attestant l'authenticité du certificat.

De même que pour [15], la gestion de clés composites schématise les relations de confiance entre les nœuds dans un graphe de certificats où les arêtes représentent un certificat numérique et les sommets des clés publiques. Les arêtes sont également couplées avec une valeur de confiance fixée par l'émetteur du certificat et un niveau de confiance est attribué au certificat obtenu.

Un exemple d'une chaîne de certificat est donné dans la Figure 3.11. [75]. Dans l'exemple, si Bob veut authentifier Alice, Bob doit calculer une valeur de confiance pour la longueur de la chaîne entière. Cela se fait en multipliant d'abord ensemble toutes les valeurs de confiance de chaque arête pour former ce qu'on appelle une valeur de confiance première. Pour obtenir la valeur de confiance finale, la longueur de la chaîne d et les probabilités p des nœuds dans la chaîne corrompue doivent également être considérées. La valeur de confiance première doit être

donc multipliée par un facteur d'atténuation $(1-p)^{d-1}$ qui donne la valeur de confiance finale pour la chaîne dans son ensemble. L'utilisateur utilise la valeur de confiance finale pour prendre une décision d'accorder l'authentification ou de rejeter la chaîne.

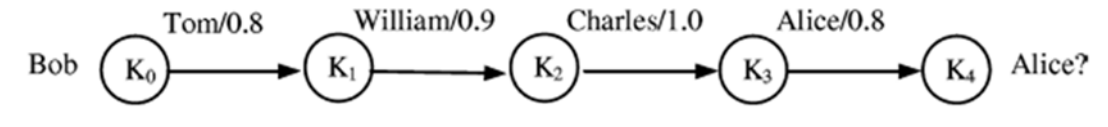


FIGURE 3.11 – Exemple de chaînage de certificat

Niveau de sécurité de la DCA

Lorsque des certificats dans le graphe de la certification sont affectés d'une métrique d'authentification, un niveau de sécurité (SL) reflétant la probabilité de corruption par un adversaire est attribué à la DCA. Le niveau de sécurité est calculé comme suit :

$$SL = 1.0 - \frac{\binom{n}{k}}{\binom{M}{c}}, \tag{4}$$

Avec n représente le nombre des nœuds serveurs formant la CA, k représente le seuil cryptographique, et M représente le nombre total des nœuds dans le réseau, et c le nombre de nœuds que les adversaires les plus puissants peuvent compromettre dans un délai fixe. Un exemple de modèle du système est donné dans la Figure 3.12. [75] montre la composition de chaînage de certificats et une DCA. Les certificats sont assignés d'une métrique d'authentification et la DCA d'un niveau de sécurité, comme expliqué ci-dessus.

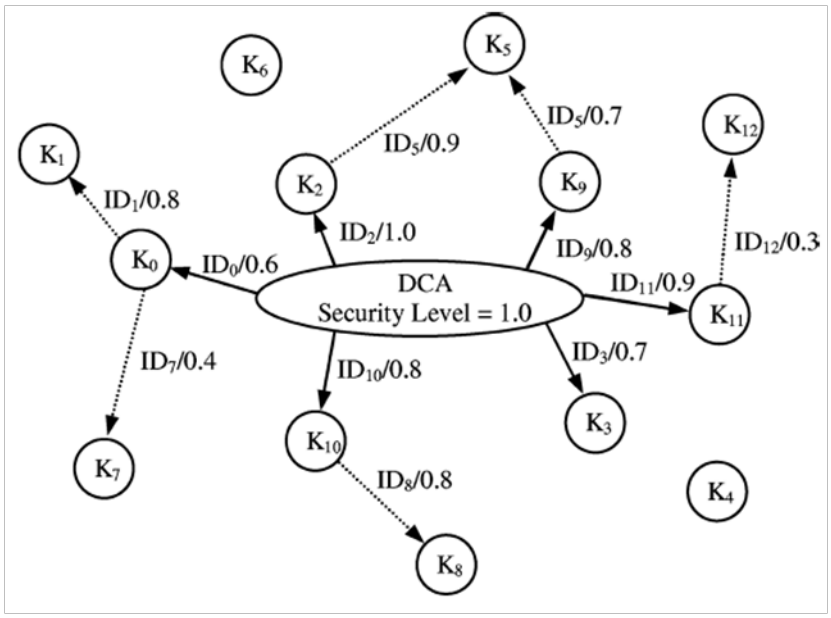


FIGURE 3.12 – Exemple de modèle système montrant la DCA composée de chaînage de certificat à un saut

Discussion et commentaires sur les approches de gestion de clés parallèles

L'approche de combiner simplement le système d'autorité de certification distribuée [78] ; [74] et le système de chaînage de certificats [15] ne traite pas correctement le problème de gestion de clés dans les MANETs du moment que le système de gestion de clés composite [75] hérite tous les points faibles de l'approche d'autorité de certification distribuée. En fait, il dégrade davantage la sécurité car il ne résout que le problème de disponibilité des [15] tout en héritant sa faible propriété d'authentification. Le système proposé dans [15] a été conçu pour les MANETs ouverts ou entièrement auto-organisé. [75] ont affirmé qu'ils avaient amélioré le [15] mais en réalité, ils ont proposé un système pour une application toute différente. .

La combinaison de deux ou plusieurs approches de gestion de clés pour éliminer les inconvénients de l'autre ne sera pas dans la plupart des cas efficace pour les MANETs. L'approche fondée sur la mobilité [14][16] peut être ajoutée en tant que complément à certains systèmes pour améliorer le processus d'échange de certificats en exploitant la mobilité des utilisateurs. Par exemple, la combinaison de l'approche basée sur la mobilité et l'approche de chaînage des certificats sera plus appropriée que la combinaison de l'un des deux avec l'approche d'autorité de certification distribuée car les inconvénients de cette dernière sont inévitables. Il faut donc chercher des façons de combiner les systèmes de gestion de clés existants mais de sorte à ne pas créer de désavantages dans le nouveau processus.

3.5 Conclusion

Cette étude qui présente les Peer-to-Peer ou la gestion des paires de clés dans les réseaux mobile ad hoc (MANET) a montré que les protocoles existants peuvent être regroupés en plusieurs catégories. Chaque catégorie a été discutée en introduisant le protocole original. Cette façon de catégoriser les protocoles disponibles permet de comprendre les approches existantes pour la gestion de clés dans les MANETs et de discuter les points forts et les points faibles de chacune d'elles.Des conclusions ont été fournies séparément pour les systèmes de gestion de clés conçus pour les MANETs totalement auto-organisés et les systèmes adaptés aux MANET basés sur l'autorité. Pour les systèmes de gestion de clés totalement auto-organisés, les approches fondées sur la mobilité [16] sont les plus réalisables sur la couche application. Les obstacles à éliminer sont la dépendance de la mobilité pendant l'amorçage « bootstrapping » de la sécurité de routage et la minimisation des interactions de l'utilisateur sur la couche application. Les approches présentées dans [13] et de [48] résolvent le dernier problème en réduisant efficacement l'interaction avec l'utilisateur jusqu'à la «pression d'un bouton ». Il est important de distinguer les mécanismes de gestion de clés destinés à la sécurisation des services au niveau des applications de ceux utilisés pour sécuriser l'infrastructure du routage. Une défaillance dans la capacité des utilisateurs à juger l'honnêteté ou l'intention des autres utilisateurs ne devrait pas compromettre la sécurité de tout service de base du réseau. La plupart des approches basées sur l'autorité font usage d'une autorité en ligne en plus d'une autorité hors ligne pour fournir d'importantes fonctions de gestion clés tel que le renouvellement du certificat. L'utilisation d'une autorité en ligne à la fois sous forme partielle ou totalement distribuée est problématique dans les MANETs. La seule solution qui émerge est d'éliminer complètement toute forme d'autorité en ligne. [16] ont proposé un système basé sur l'autorité où chaque nœud est pré-chargé avec un certificat par l'autorité hors ligne. Après la formation du réseau, chaque nœud devient son propre domaine d'autorité et distribue son certificat aux nœuds de sa portée de transmission. Cette solution n'est malheureusement pas complète car elle ne traite pas le renouvellement et la révocation du certificat. Par ailleurs, le système est dépendant de la mobilité et échoue dans une faible mobilité ou dans les réseaux ad hoc stationnaires. Les systèmes de gestion de clés basés sur les techniques de pré-distribution des clés proposés pour les réseaux de capteurs peuvent être une autre alternative qui permet de résoudre le problème de gestion de clés dans les MANETs basés sur l'autorité. Une autre observation est liée à des critères utilisés pour l'analyse des systèmes de gestion de clés dans les MANETs. Les systèmes de gestion de clés sont conçus soit pour les réseaux «ouverts» (auto-organisés) soit pour les réseaux « fermés » (basé sur autorité) et par conséquent destinés à différentes applications. les MANETs «ouverts» ont une implication inhérente à la sécurité et doivent être analysés en conséquence. Il n'est donc pas toujours possible de comparer les systèmes qui supposent l'existence d'une autorité de confiance avec ceux qui sont entièrement auto-organisés. Cette étude enfin, confirme que les mécanismes de gestion de clés proposés pour garantir la sécurité des réseaux conventionnels ne sont pas nécessairement adéquats ou adaptables aux MANET. De nouvelles techniques, conçues spécifiquement pour les MANETs, sont nécessaires.

La gestion de clés est un domaine important qui aura besoin d'une résolution avant le déploiement des réseaux ad hoc à grande échelle et bien que cette gestion ait évoluée, elle reste

encore un domaine de recherche avec place à l'innovation.

Deuxième partie

Un protocole de gestion de clés Peer-to-Peer dans les MANETs.

Chapitre 4

Proposition d'un nouveau protocole d'échange des clés dans les MANETs.

4.1 Introduction

Les modèles classiques de PKI ne peuvent pas s'appliquer aux réseaux mobiles ad hoc (MANET) dû à l'absence d'une partie centrale de confiance et à d'autres caractéristiques qui les distinguent des réseaux basés infrastructure. En effet, dans les architectures centralisées de PKI, le service de certification est fourni par une autorité centrale de certification (CA) pour tout le réseau. Cependant, cette approche n'est pas souhaitable pour les MANETs car elle peut mener à un état de vulnérabilité et de faiblesse. La disponibilité de la CA ne peut être garantie suite à une défaillance, compromission ou une attaque DoS, et donc, la sécurisation des communications entre les membres du réseau ne sera pas assurée.

Les schémas de CA distribuée en se basant sur la technique de cryptographie à seuil semblent être une bonne solution au problème de déploiement d'une PKI dans les MANETs, ces modèles permettent de partager la confiance d'une CA entre un ensemble de nœuds et les opérations de la CA (comme la signature d'un certificat digital) sont assurées par collaboration de ces nœuds. Selon ce principe, la clé privée distribuée de la CA est partagée entre les nœuds du réseau en utilisant un schéma de partage de secret. Cependant, le fait de fournir des informations sur la clé privée du service aux nœuds ad hoc peut conduire à un état de vulnérabilité du moment que ces informations (parts de clé) peuvent être utilisées pour révéler la clé privée partagée, ouvrant de ce fait la porte à l'usurpation d'identités et la révocation malveillante des certificats valides. Du fait que la sécurité physique des nœuds ad hoc est relativement pauvre, cela expose les nœuds à plusieurs attaques actives en particulier le cas de l'adversaire mobile, qui attaque, compromet et prend le contrôle d'un nœud pour une période limitée avant de passer à une autre victime. Sur une longue période du temps, un adversaire mobile peut compromettre suffisamment de nœuds pour combiner

leurs parts et ainsi découvrir la clé privée du service. Le partage de secret seul, ne peut donc pas se défendre contre cette attaque.

Afin de limiter ce type d'attaque, certaines solutions emploient le protocole proactif de rafraîchissement (PSS) qui permet aux nœuds de rafraîchir périodiquement leurs parts en générant de nouvelles parts à partir des anciennes, de cette façon, un adversaire ne peut pas combiner des parts de différentes versions pour reconstruire le secret. On définit le concept de la fenêtre de vulnérabilité qui exprime le temps durant lequel les nœuds rafraîchissent leurs parts. La fenêtre de vulnérabilité définit donc l'intervalle de temps dans lequel un adversaire mobile ne doit pas compromettre un nombre de nœuds dépassant le seuil de vulnérabilité. Néanmoins, cette hypothèse n'est pas toujours garantie notamment avec des réseaux à large échelle constitués de grand nombre de nœuds, puisque la fenêtre de vulnérabilité peut augmenter avec l'accroissement du nombre de nœuds partageant la clé privée, ou suite aux attaques DoS qui ralentissent les serveurs et/ou augmentent le délai de livraison de messages. Nous constatons donc, que les schémas classiques à seuil proposés pour les MANETs ne tiennent pas compte d'un aspect très important qui est la sécurité et la robustesse lors du passage à l'échelle du réseau.

Dans ce contexte, notre contribution vise à améliorer la robustesse et la sécurité des systèmes de gestion de clés publiques dans les MANETs. Dans cette contribution, nous avons proposé un schéma de gestion de clés pour les MANETs. Ce schéma tient compte de l'absence d'infrastructure dans les MANETs, et garantit une flexibilité au passage àl'échelle du réseau, tout en assurant les services requis de sécurité (la confidentialité, l'authentification, la tolérance aux intrusions et la disponibilité). Pour assurer l'authentification et la distribution sécurisée des clés sans aucune autorité centrale, nous avons choisi d'utiliser les caractéristiques des MANET à notre faveur.

4.2 Les Avantages et les inconvénients de notre solution

4.2.1 Avantage du protocole de gestion de clés

1. résistance à l'attaque Men-in-the-middle.
2. l'absence de toutes formes d'autorité de certification DCA « autorité de certification partiellement distribuée » ou « autorité de certification entièrement distribuée » rend la solution plus adaptée aux MANET puisqu'il n'existe aucune forme de centralisation.
3. l'absence de toutes forme tierce de confiance que ce soit TTP déconnecté (off line) ou TTP connecté (on line) rend la solution plus adaptée aux réseaux Mobile Ad hoc, puisque on ne va partager aucun matériel de chiffrement à priori.
4. la solution utilise le chiffrement asymétrique, les nœuds n'ont pas besoin d'avoir une TTP pour partager les secrets.
5. le service d'authentification est assuré par un autre mécanisme, qui est assuré par les nœuds eux mêmes, puisque les nœuds ne vont partager aucun secret entre eux, car la solution utilise le chiffrement asymétrique ; donc les nœuds échangent leur clés publiques avec leurs voisins directs, ce qui leur permet de s'assurer de l'authenticité de leurs voisins,

et dans le cas des nœuds qui ne se retrouvent pas à la portée l'un de l'autre ils vont utiliser le mécanisme qu'on a précédemment décrit dans le chapitre précédent.

6. la solution diminue considérablement les coûts de trafic, puisque les nœuds ne vont pas contacter une autorité de certification qu'elle soit partiellement ou entièrement distribuée, les communications vont être directement entre les nœuds, on n'a pas de l'inondation ni du β-unicast ; et les nœuds n'ont pas à contacter plusieurs nœuds qui forment les serveurs de certification.
7. la disponibilité du système de gestion de clés est toujours assurée, puisque on n'a pas des nœuds qui assurent la tâche et le service va se perturber avec leur départ, puisque tous les nœuds se partagent la responsabilité de la disponibilité du service, et n'importe quel nœud du réseau peut reprendre la tâche de celui qui a pris départ.
8. le coût de calcul est considérablement diminué, puisque les nœuds n'ont pas à rassembler les parties du secret partagé et à recalculer à chaque fois le nouveau secret comme dans les systèmes d'autorité de certification partiellement ou entièrement distribuées. Le nœud recalcule d'une façon autonome ces paires de clés une fois la durée de clé est expirée, et il n'a qu'a redistribué sa clé publique à ses voisins. Dans ce cas là, on n'a pas besoin des protocoles de redistributions comme dans le cas de DCA.
9. La relation symétrique entre les nœuds de réseau est préservée, puisque le système ne va pas favoriser un ensemble de nœuds sur le reste des nœuds du réseau, et donc chaque nœud assure sa part de responsabilité.
10. Cette solution permet facilement le passage aux réseaux à grand échelle, car la solution ne n'appuie sur aucune forme de centralisation, ni sur aucune forme de TTP.
11. Notre solution est comparable au système chaînage des certificats en ce qui concerne les relations de confiance directe entre les nœuds voisins, et indirecte lorsqu'il s'agit de deux nœuds éloignés, ou chaque nœud intermédiaire va se baser sur la confiance directe avec son voisin jusqu'à l'arrivée au nœud destination. Les clés sont diffusées et stockées par tous les nœuds sans aucune assistance d'une autorité de confiance.

4.2.2 Inconvénients du protocole de gestion de clés

1. Le coût de stockage de la mémoire est néanmoins augmenté puisque le nœud doit avoir les clés publiques de tous ces voisins dans le réseau, en plus les clés publiques des nœuds qui ne sont pas à portée lorsqu'il veut communiquer avec eux. À l'inverse des systèmes de gestion de clés basées sur l'identité.
2. Comment peut-on explicitement authentifier la clé publique d'un utilisateur sans aucune forme d'offline ou online autorité de confiance ? En MANET, ce problème doit être résolu avec une connectivité sporadique, tout en optimisant les ressources de communication et de calcul.
3. En conséquence, la solution peut rencontrer un problème dans la phase initiale, lorsque le nombre des nœuds dans le réseau est insuffisant pour arriver au seuil lorsqu'il s'agit des nœuds éloignés.

4.3 Principe général du nouveau protocole

Comme dans tout protocole de communication sécurisé, avant que deux nœuds se mettent d'accord sur une session de communication entre eux, ils doivent s'authentifier l'un auprès de l'autre.

Un nœud A veut demander une clé publique à un nœud B (dans le but de lui envoyer plus tard un message chiffré par une méthode de chiffrement asymétrique). Le noeud A veut aussi s'assurer qu'une fois la clé publique reçue, c'est bien le nœud B qui lui a envoyé cette clé (authentification du noeud B). Le protocole se déroule suivant les étapes suivantes :

Etape 1 : Connexion au réseau et échange des clés entre les nœuds voisins

A la création du réseau tous les nœuds créent leur propre paire de clés Privée/Publique. Chaque nœud dans le réseau échange sa clé publique avec tous ses voisins dès qu'il a rejoint le réseau.

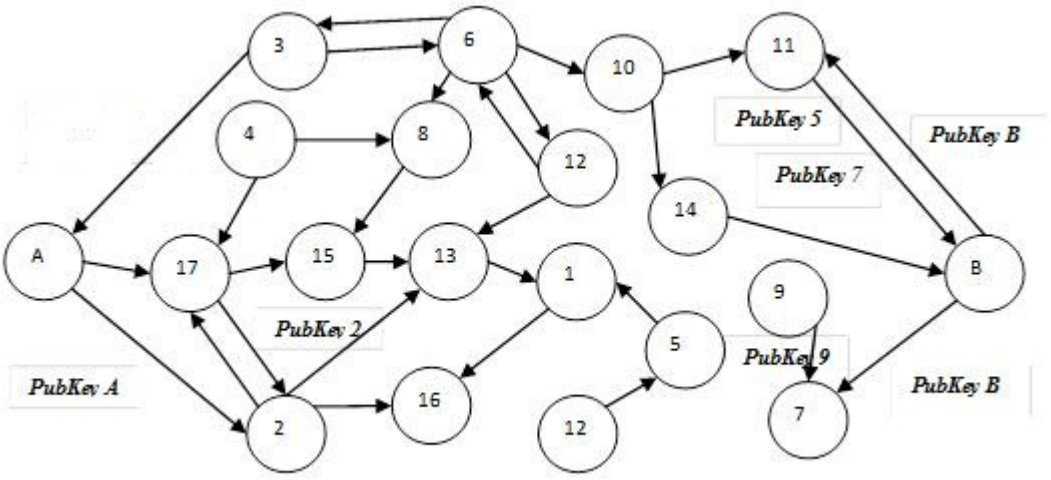


FIGURE 4.1 – Echange de clés entre tous les nœuds voisins dans le réseau

Etape 2 : Connexion d'un nouveau nœud au réseau

Dés qu'un nouveau nœud rejoigne le réseau, il va immédiatement envoyer sa clé publique à tous ses voisinsqui sont les nœuds qui se trouvent à sa portée de transmission, et récupère lui aussi les siens.

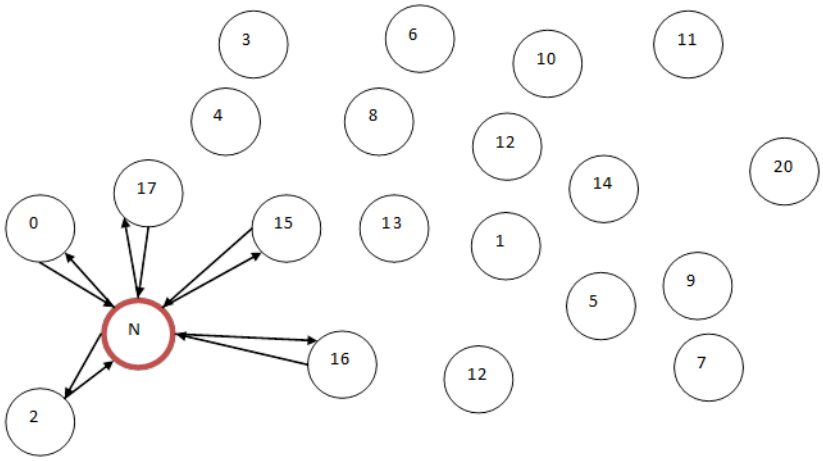


FIGURE 4.2 – Connexion d'un nouveau nœud au réseau et l'échange des clés avec ces voisins

Etape 3 : Echange des clés entre deux nœuds éloignés

Quand un nœud nommé Src veut récupérer la clé publique d'un nœud éloigné qu'on va le nommer Dest et que se dernier ne se retrouve pas à sa portée (ou dans son voisinage), il procède comme suit :

1. Il envoie une requête de type « demande clé publique à Dest » à chacun de ses voisins après l'avoir crypté avec la clé publique du nœud voisin correspondant.
2. Chaque nœud voisin décrypte la requête envoyée par le nœud Src avec sa propre clé privée.
3. Chaque noeud ayant reçu la requête, choisit à son tour parmi ses voisins celui qui se rapproche le plus du nœud Dest puis envoie la requête cryptée à l'aide de la clé publique du nœud voisin choisi. N.B : si un nœud est un voisin commun d'un certain nombre de nœuds qui ont déjà reçu la requête de la clé publique, et il remplit toutes les conditions nécessaires pour être choisi par les nœuds pour lui transmettre la requête, s'il est choisi par tous ces nœuds, on risque de réduire considérablement le nombre de routes qui mènent de Src à Dest. D'un autre côté, si deux nœuds voisins reçoivent la requête et ils remplissent toutes les conditions nécessaires pour être choisi l'un par rapport à l'autre, chacun envoie la requête à l'autre sans pouvoir arriver à la destination.et pour régler ce problème on doit éliminer de la table de routage des nœuds, ceux qui ont déjà envoyé ou reçu la requête.
4. C'est ainsi que de proche en proche, la requête arrive vers le nœud Dest à travers différents chemins. Le nœud Dest récupère ainsi différentes requêtes à travers différents nœuds voisins, qu'il décrypte ensuite avec sa propre clé privée.
5. Le nœud Dest procède alors à l'authentification de la requête en contrôlant et vérifiant ce qui suit :

   a. Le nombre de requêtes reçues doit être supérieur ou égal à une certaine valeur seuil déjà fixée.

   b. Il faut que toutes les requêtes soient identiques.

c. Il faut que toutes les requêtes viennent d'un même nœud source Src.

d. Les trois conditions précédentes doivent être vérifiées dans un délai bien déterminé (le temps initial étant celui où la première requête a été reçue * seuil).

6. Si l'une des conditions précédentes n'est pas vérifiée alors la requête est rejetée et le nœud Src doit initier une nouvelle requête de demande clé auprès du nœud Dest et cela après l'expiration du délai de réponse qui est aussi fixé.
7. Si toutes les conditions précédentes sont vérifiées alors le nœud Dest considère que la requête est valide et procède à l'envoi d'une clé publique au nœud Src. Le nœud Dest répond à la requête initiale en construisant une réponse contenant la clé publique générée par le nœud Dest. Cette réponse est envoyée le long des chemins inverses traversés par la requête du nœud Src vers Dest. Chaque chemin est calculé de façon locale par chaque nœud intermédiaire traversée par la requête, en stockant dans sa table de routage le nœud d'où provient la requête.
8. Lorsque le nœud Src reçoit les différents paquets de type réponse, il procède à l'authentification de cette réponse et par conséquent à l'authentification de la clé publique générée par le nœud Dest. Pour réaliser cela, il procède comme suit :

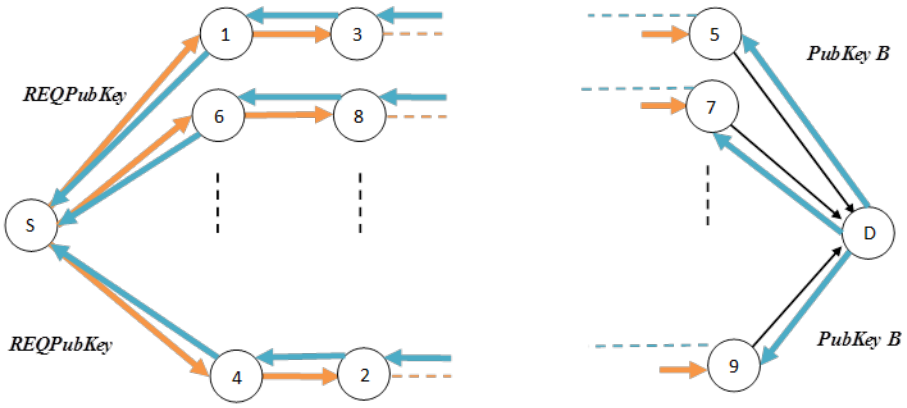
   a. Le nombre de réponses reçues doit être supérieur ou égal à un certain seuil déjà fixé (décidé en fonction de la densité des nœuds dans les réseau, et a nombre des voisins de nœuds).

   b. Après avoir décrypté toutes les réponses, le Src vérifie que toutes les clés publiques sont bien identiques.

   c. Toutes les clés doivent provenir du nœud Dest.

   d. Toutes les conditions précédentes doivent être vérifiées dans un délai de temps fixé à l'avance.

Si la phase d'authentification réussit alors la clé publique du nœud Dest sera considérée comme authentique (le nœud Src peut envoyer des données chiffrées au nœud Dest avec cette clé) sinon le nœud Src doit recommencer de nouveau le processus décrit à l'étape 3 du protocole.



1- S envoi sa requête de clé à tous ses voisins (1 ;6 ;4...etc.)
2- Parmi les voisins du 4, celui le plus proche de la destination D : 2
3- Parmi les voisins du 6, celui le plus proche de la destination D : 8
4- Parmi les voisins du 1, celui le plus proche de la destination D : 3
5- A partir du nœud 5, on arrive à la destination D

FIGURE 4.3 – Echange de clés entre deux nœuds éloignés dans le réseau

4.4 Les algorithmes utilisés par notre protocole

4.4.1 Algorithme 1 : échange des clés publiques entre les nœuds à la création du réseau

Algorithme 4.1 Echange des clés publiques entre les nœuds à la création du réseau

```
Pour (i =1 jusqu'à n ) faire // où n représente le nombre total des nœuds dans le réseau
{
        // A la création du réseau tous les nœuds créent leur propre paire de clés Privée/Publique.
        // Chaque nœud dans le réseau échange sa clé publique avec tous ses voisins dès qu'il a rejoint le
réseau.
        Pour ( j = 2 jusqu'à n) faire
        {
              // chaque nœud i envoie sa clé publique à tous ces voisins
              // donc il vérifie si le nœud j figure parmi ses voisins, il lui envoie sa clé publique.
              Si ( SiVoisins (Noud(i), Noud(j) ) ) alors
              {
                   Noud(i).SendPubKey(j)
              }fin Si
        }fin Pour
}

// cette fonction est assurée par l'agent « Security_agent » est permis la sélection des voisins de chaque
nœud
// qui se retrouve dans sa portée de transmission.

Function selectionVoisins( noud(i) )
{
          Pour ( i =1 jusqu'à n ) faire
          {
               Pour ( j = 2 jusqu'à n) faire
               {
                    Si ( Distance ( Noud(i), Noud(j) ) <= 2 sauts )) alors
                    {
                         // si le nœud j se trouve à deux au maximum du nœud i donc on le met
                         // dans la liste des voisin de i.
                         Voisins ( Noud(i), Noud(j) ) = vrai
                    } fin Si
               }fin Pour
          }fin Pour
}

Bool Function SiVoisins (Noud(i) , Noud(j) )
{
        // cette fonction retourne vrai si le nœud j figure dans la liste des voisins, sinon elle retourne
        // faux
       Retourne Voisins (Noud(i) , Noud(j)) // c'est un tableau à deux dimensions contient la valeurs 1 si
les nœuds en question sont des voisins, 0 sinon
}
```

4.4.2 Algorithme 2 : l'arrivée d'un nouveau nœud sur le réseau

Algorithme 4.2 L'arrivée d'un nouveau nœud sur le réseau

```
// lorsqu'un nœud arrive sur le réseau, il détermine ses voisins, puis il leurs envoie sa clé publique
// et récupère la sienne.

selectionVoisins( Noud(n+1) )   // cette fonction permet la sélection des voisins du nouveau nœud.
{
        Pour (i =1 jusqu'à n ) faire
        {
            Si ( SiVoisins(Noud(n+1),Noud(i) ) ) alors
               {
                // si le nœud i est voisin du nouveau nœud (n+1) , il lui envoie sa clé publique
                 // et récupère la sienne.
                 Noud(n+1).SendPubKey(Noud(j) )
                 Noud(j).SendPubKey(Noud(n+1))
               }fin Si
        }
}
```

4.4.3 Algorithme 3 : envoi d'une requête clé entre deux nœuds éloigné x et y

Algorithme 4.3 Envoi d'une requête clé entre deux nœuds éloignés x et y

```
Si ( SiVoisins ( x , y )) alors
{
      // Le nœud y est un voisin de x alors appliquer l'algorithme 2
} fin Si

Sinon
{
      // envoyer la requête à tous les voisins de x
      Pour (i =1 jusqu'à n ) faire
            {
            Si (SiVoisins (Noud(x), Noud(i) ) ) alors
               {
                     // crypter la requête clé par la clé publique des voisins
                     crypteReq(PubKey(i))
                     // envoyer la requête après l'avoir cryptée
                     Noud(x).SendReqPubKey(i)
               }fin Si
             }fin Pour

       // Dans le cas d'un nœud intermédiaire (on n'est pas encore arrivé à y).
       Si (Noud(i) != Noud(y)) alors
          {
            // après avoir décrypté la requête, le Nœud va choisir parmi ses voisins
            // celui le plus proche de Noeud(y).
            decrypteReq(PrivKey(i))
            Pour ( j ∈ Voisins ( Noud(i) , Noud(j) )) faire
                  {
                    // le nœud j appartient à liste des voisins du i.
                    // Choisir parmi tous les voisins, celui le plus proche de y
                    // crypter la requête clé par la clé publique de ce voisin.
                    crypteReq(PubKey(j))
                    // envoyer la requête après l'avoir cryptée.
                    Noud(i).SendReqPubKey(j)
                  } fin Pour
          } fin Si // on doit refaire cette opération pour tous les nœuds intermédiaires entre x et y

         Sinon Si (Noud(i) == Noud(y)) alors
                {
                  // la requête est arrivée à sa destination
                  decrypteReq(PrivKey(y))
                } fin Sinon
}
```

4.4.4 Algorithme 4 : réception des requêtes clé par le nœud y

Algorithme 4.4 Réception des requêtes clé par le nœud y

Si (nombreReq ≥ seuil ET SourceDesReq == x ET toutes les requêtes sont identiques ET le temps de la requête n'a pas encore expiré) Alors
{
// la requête est acceptée et le nœud Noud(y) doit répondre au nœud x en lui envoyant
// sa clé publique en suivant les chemins inverses de la requête.
}nfin Si

Sinon
{
// la requête est refusée et le Noeud(x) doit relancer une nouvelle requête pour avoir
// la clé publique du nœud Noeud(x) après l'expiration du délai de l'opération.
}fin Sinon

4.4.5 Algorithme 5 : réponse de la Requête par le nœud y

Algorithme 4.5 Réponse de la requête par le nœud y :

```
Si ( SiVoisins (y , x )) alors
{
     // Le nœud x est un voisin de y alors appliquer l'algorithme 2
} fin Si

Sinon
{
      // Dans le cas d'un nœud intermédiaire (on n'est pas encore arrivé à x )
      Si (Noud(i) != Noud(x)) alors
          {
           // lorsqu'un nœud reçoit la clé publique il la décrypte avec sa propre clé privée.
           decryptePubKey (PrivKey(i))
           // crypter la clé par la clé publique du voisin, qui lui a envoyé la requête clé
           // auparavant (chemin inverse).
           cryptePubKey (PubKey(j))
           // envoyer la clé publique de y après l'avoir cryptée.
           Noud(i).SendPubKey(j)
         } // on doit refaire cette opération pour tous les nœuds intermédiaires entre x et y

       Sinon Si (Noud(i) == Noud(x)) alors
              {
                // la clé est arrivée à sa destination finale, le nœud x décrypte la clé reçue avec sa
clé
                // privée
                decryptePubKey (PrivKey(y))
              } fin Si
}
```

4.4.6 Algorithme 6 : réception des clés de y par le nœud x

Algorithme 4.6 Réception des clés de y par le nœud x

Si (nombrePubKey≥ seuil ET SourceDesPubKey == y ET toutes les clés sont identiques ET le temps de l'opération n'a pas encore expiré) Alors
{
// La clé est accepté et le nœud Noud(x) possède donc la clé publique de y et peut lui
// envoyer les informations en toute sécurité.
}fin Si

Sinon
{
// la clé est refusée et le Noud(x) doit relancer une nouvelle requête pour avoir la clé
// publique du nœud Noud(x) après l'expiration du délai de l'opération.
}

4.5 Implémentation, tests et résultats à l'aide du simulateur réseauNS2

NS est un outil logiciel de simulation de réseaux informatiques. Il est principalement bâti avec les idées de la conception par objets, de réutilisabilité du code et de modularité. Il est devenu aujourd'huiun standard de référence en ce domaine. Le simulateur NS actuel est particulièrement bien adapté aux réseaux à commutation de paquets et à la réalisation de simulations de petite taille. Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unipoint ou multipoint, des protocoles de transport, de session, deréservation, des services intégrés, des protocoles d'application comme HTTP. De plus le simulateur possède déjà une palette de systèmes de transmission (couche 1 de l'architecture TCP/IP), d'ordonnanceurs et de politiques de gestion de files d'attente pour effectuer des études de contrôle de congestion. La liste des principaux composants actuellement disponible dans NRREQ : la réponse à la requete clés par catégorie est :

Application	Web, ftp, telnet, générateur de trafic (CBR, ...)
Transport	TCP, UDP, RTP, SRM
Routage	Statique, dynamique (vecteur distance) et routage multipoint (DVMRP, PIM).
Gestion de file d'attente	RED, DropTail, Token bucket
Discipline de service	CBQ, SFQ, DRR, Fair queueing
Système de transmission	CSMA/CD, CSMA/CA, lien point à point

NS est devenu l'outil de référence pour les chercheurs du domaine. Ils peuvent ainsi partager leurs efforts et échanger leurs résultats de simulation.

4.6 Scénario de simulation

Nous avons mené une série de simulations afin d'évaluer les performances de notre système de gestion de clés. Nous avons utilisé pour cela le simulateur NS-2, dans lequel nous avons implémenté nos algorithmes décrits précédemment.

L'utilitaire « setdest » de NS-2 a été utilisé pour générer les scénarii de mobilité des nœuds selon le modèle de mobilité «RandomWaypoint ». Dans cette simulation, notre modèle est établi selon les paramètres suivants :

Pour réaliser des simulations interprétables et évaluables, on a créé notre propre paquet nommée « Security_Packet ». Ce paquet contient toutes les informations nécessaires qui permettront de suivre l'acheminement des clés d'un nœud source à un nœud destination, et de garder la trace des chemins empruntés pour pouvoir revenir au nœud source par les chemins inverses, dès qu'il arrive au nœud destination.

Le paquet contient aussi toutes les informations concernant la donnée qu'il transporte, que ce soit la requête de demande de clé publique, ou la clé publique elle même envoyée par le nœud destination. Les données sont bien sûr cryptées.

L'adresse du nœud source et destination ainsi que la taille de la donnée à transmettre figurent aussi parmi les données de ce paquet.

Ret	0 : le premier paquet reçu par le nœud. 1 : l'accusé de réception du paquet envoyé.
send_time	L'instant où on a envoyé un paquet.
rec_time	L'instant où on a reçu un paquet.
Encrypte	0 : la donnée du paquet n'est pas cryptée. 1 : la donnée du paquet est cryptée.
seq	Numéro de séquence du paquet.
LastPacket	1 : indique la dernière séquence du paquet.
SourceInit	Contient l'adresse du nœud source.
destinationFinal	Contient l'adresse du nœud destination.
typeData	REQ : une requête clé. KEY : la clé publique. RREQ : la réponse à la requete clé.
Data	La donnée portée dans le paquet.

Un agent créé et nommé « Security_packetAgent » est responsable de la génération des clés des nœuds, de cryptage et décryptage des données que ce soit une requête de clé ou la clé elle-même. Les clés générées par cet agent sont d'une taille de 512 à 1024 bits. L'agent est tenu de déterminer la liste des voisins des nœuds, de préserver la liste des nœuds formant les chemins de la source à la destination . . .etc.

4.7 Les paramètres de la simulation

Dans nos simulations, l'environnement est un réseau de taille 1000m x 1000m, dans lequel se trouvent généralement 100 nœuds ou plus, mais ce nombre est varié pour tester l'effet de la densité de nœuds sur les différents facteurs de performance à étudier le long de ce chapitre. Nous effectuerons des simulations d'une durée de 150000 secondes.

4.7.1 Mobilité des nœuds

Le mouvement des nœuds sur une simulation a un impact très important. En effet,la vitesse, la direction et la fréquence du mouvement ont un impact considérable sur la transmission d'information. Dans nos simulations, nous nous sommes basés sur le « Random Waypoint Model ». Ce modèle est devenu par la suite le standard dans la recherche sur les réseaux sans fil. Il fournit des scénarii de simulation où tous les nœuds mobiles se déplacent aléatoirement dans la surface de simulation. En variant les caractéristiques de la mobilité, on s'attend à ce qu'il y ait un impact significatif sur les performances des protocoles de routage. Dans ce modèle, les nœuds choisissent une destination au hasard et s'y rendent à une vitesse comprise dans l'intervalle [0, 5]. On peut aussi choisir le temps de pause qui représente le temps pendant lequel les nœuds restent immobiles entre chaque mouvement. Tout ceci est présent dans l'outil setdest de NS qui prend comme paramètres :

1. la version de l'outil setdest 1 ou 2,
2. le nombre de nœuds dans le réseau : 100
3. la superficie de la simulation qui varie de (600m x 600m) à (1000m x1000m) ...etc.
4. la vitesse de mobilité qui varie entre 0 et 5m/s.

4.8 Trafic entre les nœuds

Dans NS 2, il existe un générateur permettant de créer plusieurs types de trafic tel que : TCP (Transport Control Protocol), CBR (Constant Bit Rate)...etc. Dans notre solution on n'a pas besoin d'un générateur automatique du trafic, puisqu'on a créé notre propre type de paquet «Security_packets », et c'est à nous aussi de choisir les moments pour envoyer nos paquets, ce qui va réduire considérablement le trafic sur le réseau.

4.9 Analyse de la sécurité de notre protocole : la résistance à l'attaque Man-In-The-Middle

4.9.1 Principe de l'attaque Man-In-The-Middle

L'attaque de l'homme du milieu (HDM) ou Man-In-the Middle attack (MITM) est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis.

L'attaquant doit d'abord être capable d'observer et d'intercepter les messages d'une victime à l'autre.

Supposons que Alice et Bob veulent communiquer ensemble, mais un attaquant peut se placer entre les deux, intercepter la clé envoyée par Alice et envoyer à Bob une autre clé, en se faisant passer pour Alice. De même, il peut remplacer la clé envoyée par Bob à Alice par une autre clé, en se faisant passer pour Bob. L'attaquant peut ainsi faire « croire » à Alice et Bob qu'ils communiquent directement alors que toutes les communications passent par l'attaquant qui possède les clés qu'il faut pour chiffrer et déchiffrer les messages. Ce type d'attaque est généralement possible lorsque les deux communicants établissent au préalable une communication à deux sens pour s'échanger des informations pour le chiffrement (exemple le protocole Diffie-Helman). Notre protocole peut être aussi sensible à l'attaque Man-In-The-Middle, puisque le nœud Src et le nœud Dest échangent des informations dans les deux sens dans le but d'établir une clé publique du nœud Dest pour le nœud Src.

Grâce aux simulations effectuées et aux résultats obtenus, nous allons montrer dans ce qui suit que notre protocole résiste bien à l'attaque Man-InThe-Middle.

4.9.2 Résistance du protocole contre l'attaque Man-In-The Middle

Dans cette partie, nous allons étudier et estimer la probabilité de réussite de l'attaque de type Man-In-The-Middle, sur le protocole d'échange de clé publique authentifié que nous avons décrit plus précédemment.

Afin d'estimer cette probabilité, nous considérons ce qui suit : .

Nous supposons qu'il existe une « coalition» (ou groupe de nœuds) pouvant jouer le rôle d'attaquant de type de Man-In-The-Middle (ces nœuds peuvent en effet s'échanger des informations interceptées lors des différents échanges entre les nœuds Src et Dest). Cette «coalition» de nœuds peut se trouver le long de «tous les chemins » entre Scr et Dest (durant le processus d'envoi de la clé publique de Dest à Src.

Nous considérons les paramètres suivants :

1. nbn : le nombre de nœuds total du réseau.
2. nbmoyen : le nombre de nœuds intermédiaires moyen contenus sur tout chemin reliant deux nœuds quelconques du réseau.
3. nbchemins : le nombre de chemins moyen reliant deux nœuds quelconque du réseau.
4. nbm : le nombre de nœuds constituant la «coalition» de l'attaquant Man-In-The-Midde.

A l'aide de tous ces paramètres, nous déduisons que la probabilité que pour tout chemin, aucun nœud ne soit compromis est donnée par la formule suivante :

$$P = \frac{C_{nbmoyen}^{nbn-nbm}}{C_{nbmoyen}^{nbn}} \quad (C : \textit{calcul de combinaisons}) \qquad (1)$$

Par conséquent, la probabilité qu'il existe au moins un nœud compromis le long de tous les chemins est de :

$$P - 1 \qquad (2)$$

L'attaque Man-In-The-Middle décrite plus haut pourra réussir s'il existe au moins un nœud compromis (appartenant à la «coalition) le long de tous les chemins reliant deux nœuds quelconques du réseau. On en déduit que la probabilité de réussite de l'attaque Man-In-The-Middle est donnée par :

$$P_{Man-In-The-Middle} = (1 - P)^{nbchemins} \qquad (3)$$

Afin d'estimer la valeur de cette probabilité, nous considérons les données suivantes recueillies lors des différentes simulations réalisées sur notre protocole :

Nous avons réalisé des tests sur un réseau de $nbn = 100$ nœuds.

Nous supposons qu'il y a $nbm = 23$ nœuds malicieux formant une «coalition» dans le réseau (ceci correspond à un réseau «pollué» à 28,75%, ce qui est considérable pour un réseau adhoc).

Nos tests ont permis de constater que le nombre moyen de nœuds intermédiaires dans un chemin reliant deux nœuds quelconques (Src et Dest) est de $nbmoyen = 4$ et le nombre de chemins moyens entre deux nœuds quelconques du réseau est de $nbchemins = 8$.

En appliquant la formule de calcul donnée en (3), nous trouvons que :

$$P_{Man-In-The-Muddle} = 0,0087517767$$

Cette probabilité très faible, nous permet ainsi de vérifier que notre protocole résiste bien à l'attaque de type Man-In-The-Middle.

Conclusion Générale

Le problème de la sécurité dans le domaine des réseaux est un problème décisif car les données transmises sont potentiellement sensibles et il est souvent aisé de les intercepter ou de les manipuler, notamment dans un réseau pourvu d'une composante sans fil, ou pour lequel tout nœud joue le rôle supplémentaire de routeur. Particulièrement, dans les MANETs où au lieu d'utiliser des routeurs pour transmettre les données comme dans les réseaux traditionnels, il faut que chaque nœud participe dans le processus de routage. Malgré l'évolution des MANET au cours de la dernière décennie, il y a encore un certain nombre de problèmes liés à la sécurité qui sont ouverts. Cela signifie, que bien que des solutions aient été proposées, aucune ne semble satisfaire toutes les contraintes de MANET.

La principale observation est que les techniques cryptographiques sont souvent au centre de la résolution des problèmes de sécurité et de là ont besoin de la gestion de clés. La gestion sécurisée des clés avec une fonctionnalité de haute disponibilité offre la sécurité réseau via des mécanismes cryptographiques. Cependant, la plupart des protocoles de routage et les mécanismes de base liés aux réseaux néglige la tâche cruciale de la gestion sécurisée des clés et assume la préexistence et le pré-partagé des clés secrètes et/ou des paires de clés privées ou publiques. En fait, beaucoup de mécanismes basés sur la cryptographie qui résolvent les problèmes de sécurité des MANET ont un recours direct à une infrastructure de gestion de clés efficace et sécurisée.

Dans l'état de l'art de ce travail, nous avons étudié globalement la gestion des paires de clés dans les réseaux mobile ad hoc (MANET), et nous avons montré que les protocoles existants peuvent être regroupés en plusieurs catégories. La plupart des approches basées sur l'autorité font usage d'une autorité en ligne en plus d'une autorité hors ligne pour fournir d'importantes fonctions de gestion de clés telles que le renouvellement du certificat. L'utilisation d'une autorité en ligne à la fois sous forme partielle ou totalement distribuée est problématique dans les MANETs. La seule solution qui émerge est d'éliminer complètement toute forme d'autorité en ligne.

Dans la deuxième partie de ce travail, nous avons présenté un nouveau protocole de gestion de clés pour les MANETs, qui permet l'authentification et la distribution efficace de clés TEKs [1], en éliminant toute forme de TTP (Third Trusted Party). Du point de vue sécurité, l'authentification et l'échange de clés au sein de notre modèle sont à la charge de tout le réseau ad hoc, donc il n'y a pas de nœuds particuliers (serveurs, leaders, ...) facilitant la tâche d'usurpation ou violation, de plus une telle récupération de la clé ou attaque ne pourrait avoir lieu. L'échange des clés à travers plusieurs chemins a montré la haute résistance de notre protocole aux attaques qui ciblent

1. Trafic Key Encryption

les MANETs notamment l'attaque Men-In-The-Middle qui est l'attaque la plus fréquente vu le moyen de communications utilisé par ces derniers, qu'on a montré à travers plusieurs simulations effectuées par l'outils NS2.

Malgré les avantages apportés par notre protocole, on envisage tester réellement sur les réseaux mobiles ad hoc, et comparer les résultats obtenu à ceux de la simulation pour voir, si ce protocole est vraiment applicable et donne des résultats pour les MANETs.

Bibliographie

[1] ABDUL-RAHMAN.A, AND HAILES.S. A distributed trust model. In Proceedings of the ACM New Security Paradigms Workshop. 1997.

[2] I. F.AKYILDIZ, W.SU, Y.SANKARASUBRAMANIAM, AND CAYIRCI. A survey on sensor networks. IEEE Commun. Mag. 40, 8 (Aug.), 102–114. 2002.

[3] G.ATENIESE, M.STEINER, AND G.TSUDIK. Authenticated group key agreement and friends. In Proceedings of the 5th ACM Conference on Computer and Communications Security. 1998.

[4] E.AYANOGLU, C.-L.I, R.D.GITLIN, AND J. E.MAZO. Diversity coding for transparent self-healing and fault-tolerant communication networks. IEEE Trans. Commun. 41, 11, 1677–1686. 1993.

[5] T.BETH, B.MALTE, AND K.BIRGIT. Valuation of trust in open networks. In Proceedings of the Third European Symposium on Research in Computer Security. 1994.

[6] R. BLOM. An optimal class of symmetric key generation systems. In Proceedings of EUROCRYPT'84. 1985.

[7] R.B.BOBBA, L.ESCHENAUER, V.D.GLIGOR, AND W.ARBAUGH. Bootstrapping security associations for routing in mobile ad-hoc networks. In Proceedings of the IEEE Global Telecommunications Conference. 2003.

[8] D.BONEH, AND M.FRANKLIN. Identity-based encryption from weil pairing. In Proceedings of the Conference on Advances in Cryptology (CRYPTO'01). 2001.

[9] J.BROCH, AND D. B.JOHNSON. The dynamic source routing protocol for mobile ad hoc networks. IETF Internet Draft. October. 1999.

[10] C.BUNDO, A.DE SANTIS, A.HERZBERG, S.KUTTEN, U.VACCARO, AND M.YUNG. Perfectly-secure key distribution for dynamic conferences. In Proceedings of CRYPTO'92. 1993.

[11] L.BUTTYAN. Building blocks for secure services : Authenticated key transport and rational exchange protocols. Ph.D. dissertation. University Technique de Budapest, Budapest, Hungary. 2001.

[12] L.BUTTYAN, AND J.P.HUBAUX. Stimulating cooperation in self-organizing mobile ad hoc networks. ACM Mobile Netw. Appl. 8, 5, 579–592. 2003.

[13] M.CAGALJ, S.CAPKUN, AND J.HUBAUX. Key agreement in peer-to-peer wireless networks. Proc. IEEE (Special Issue on Cryptography and Security) 94, 2, 467–478. 2006.

[14] S.CAPKUN, L.BUTTYAN, AND J.-P.HUBAUX. Mobility helps security in ad hoc networks. In Proceedings of MobiHoc. 2003a.

[15] S.CAPKUN, L.BUTTYAN, AND J.-P.HUBAUX. Self-organized public-key management for mobile ad hoc networks. IEEE Trans. Mobile Comput. 2, 1, 52–64. 2003b.

[16] S.CAPKUN, J.HUBAUX, AND L.BUTTYAN. Mobility helps peer-to-peer security. IEEE Trans. Mobile Comput. 5, 1, 43–51. 2006.

[17] C.CARTER, S.YI, P.RATANCHANDANI, AND R.KRAVETS. Manycast : Exploring the space between anycast and multicast in ad hoc networks. In Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MOBICOM'03). 2003.

[18] J.C.CHA, AND J. H.CHEON. An identity-based signature from gap diffie-hellman groups. In Proceedings of the Conference on Public Key Cryptography (PKI'03). 2003.

[19] H.CHAN, AND A.PERRIG. PIKE : Peer intermediaries for key establishment in sensor networks. In Proceedings of INFOCOM'05. 2005.

[20] Y.P.CHEN, AND A.L.LIESTMAN. A zonal algorithm for clustering ad hoc networks. Int. J. Foundat. Comput. Sci. 14, 2, 305–322. 2003.

[21] B.CHRISTIANSON. Why isn't trust transitive. In Proceedings of the InternationalWorkshop on Security Protocols. 1996.

[22] B.DAHILL, E.LEVINE, E.ROYER, AND C.SHIELDS. A secure routing protocol for ad hoc networks. Tech. rep. UM-CS-2001-037. University of Massachusetts, Amherst, MA.2001.

[23] N. J. DEARHAM. Development, implementation and quantification of an ad-hoc routing protocol for mobile handheld terminals. M. S. thesis in Electronic Engineering. Department of Electrical, Electronic and Computer Engineering, University of Natal, Durban, South Africa.2003.

[24] Y.DESMEDT, AND S. JAJODIA. Redistributing secret shares to newaccess structures and its applications. Tech. rep. ISSE-TR-97-01. Department of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, Fairfax, VA. 1997.

[25] D. DOLEV, AND A.C.YAO. On the security of public key protocols. IEEE Trans. Inform. Theor. 29,2, 198–208. 1983.

[26] J.R.DOUCEUR. The Sybil attack. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02). 2002.

[27] L. ESCHENAUER, AND V. D.GLIGOR. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS'02). 2002.

[28] FEM 2005. U.S Federal Emergency Management Agency (FEMA) : Information on federally declared disasters. Available online at http ://www.fema.gov.2005.

[29] R.GENNARO, S.JARECKI, H.KRAWCZYK, AND T.RABIN. Secure distributed key generation for discrete log based cryptosystems. In Proceedings of the Conference on Advances in Cryptology (EUROCRYPT'99). 1999.

[30] Z. J.HAAS, J.DENG, B.LIANG, P.PAPADIMITRATOS, AND S.SAJAMA. Wireless ad hoc networks. In Encyclopedia of Telecommunications, J. Proakis, Ed. John Wiley, New York, NY. 2002.

[31] Z.J.HAAS, AND M.PERLMAN. The performance of query control schemes for zone routing protocol. In Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'98). 1998.

[32] Z.J.HAAS, AND S.TABRIZI. On some challenges and design choices in ad-hoc communications. In Proceedings of the IEEE Military Communications Conference (MILCOM'98). 1998.

[33] A.HERZBERG, S.JARACKI, H.KRAWCZYK, AND M.YUNG. Proactive secret sharing or : How to cope with perpetual leakage. In Proceedings of the Conference on Advances in Cryptology (CRYPTO'95). 1995.

[34] Y.-C.HU, D.B.JOHNSON, AND A.PERRIG. SEAD : Secure efficient distance vector routing for mobile wireless ad hoc networks. In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications. 2002b.

[35] D.HUANG, M.MEHTA, D.MEDHI, AND L.HARN. Location-aware key management scheme for wireless sensor networks. In Proceedings of the ACM Workshop on Security for Ad Hoc and Sensor Networks (SASN). 2004.

[36] J.-P.HUBAUX, L.BUTTYAN, AND S.CAPKUN. The quest for security in mobile ad hoc networks. In Proceedings of MobiHoc'01. 2001.

[37] D.B.JOHNSON, AND D.A.MALTZ. Dynamic source routing in ad-hoc wireless networks. In Mobile Computing, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 153–181. 1996.

[38] A.JOSANG, E.GRAY, AND M.KINATEDER. Analysing topologies of transitive trust. In Proceedings of the First International Workshop on Formal Aspects in Security and Trust (FAST'03). 2003.

[39] M.JOYE, AND S.-M.YEN. ID-based secret-key cryptography. ACM Operat. Syst. Rev. 32, 4, 33–39. 1998.

[40] J.JUBIN, AND J.D.TORNOW. The DARPA Packet Radio Network Protocol. IEEE 75, 1, 21–32. 1987.

[41] A.KHALILI, J.KATZ, AND W.A.ARBAUGH. Towards secure key distribution in truly ad-hoc networks. In Proceedings of the IEEE Workshop on Security and Assurance in Ad-Hoc Networks.KIM, Y., PERRIG, A., AND TSUDIK, G. 2000. Simple and fault-tolerant key agreement for dynamic collaborative groups. In Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS'00). 2003.

[42] Y.KIM, A.PERRIG, AND G.TSUDIK. Tree-based group key agreement. ACM Trans. Inform. Syst. Sec. 7, 1, 60–96. 2004.

[43] J.KONG, P.ZERFOS, H.LUO, S.LU, AND L.ZHANG. Providing robust and ubiquitous security support for mobile ad-hoc networks. In Proceedings of the Ninth International Conference on Network Protocols (ICNP'01). 2001.

[44] LEE, W.-B. AND CHANG, C.-C. 1999. (t, n) Threshold digital signature with traceability property. J. Inform. Sci. Eng. 15, 5, 669–678.

[45] D.LIU, AND P.NING. Establishing pairwise keys in distributed sensor networks. In Proceedings of the 10th ACM Conference on Computer and Communication Security. 2003a.

[46] D.LIU, AND P.NING. Location-based pairwise key establishments for static sensor networks. In Proceedings of the ACM Workshop on Security for Ad Hoc and Sensor Networks (SASN). 2003b.

[47] H.LUO, P.ZERFOS, J.KONG, S.LU, AND L.ZHANG. Self-securing ad hoc wireless networks. In Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02). 2002.

[48] J. M.MCCUNE, A.PERRIG, AND M. K.REITER. Seeing-is-believing : Using camera phones for humanverifiable authentication. In Proceedings of the IEEE Symposium on Security and Privacy. 2005.

[49] A.MENEZES, P.VAN OORSCHOT, AND S.VANSTONE. Handbook in Applied Cryptography. CRC Press, Boca Raton, FL. 1996.

[50] M.MICHELS, AND P.HORSTER. On the risk of disruption in several multiparty signature schemes. In Proceedings of the Advances in Cryptology (ASIACRYPT'96). 1996.

[51] R.MORRIS, J.JANNOTTI, F.KAASHOEK, J.LI, AND D.DECOUTO. Carnet : A scalable ad hoc wireless network system. In Proceedings of the 9th ACM SIGOPS European Workshop. 2000.

[52] E.C.H.NGAI, M.R.LYU, AND R. T.CHIN. An authentication service against dishonest users inmobile ad hoc networks. In Proceedings of the IEEE Aerospace Conference. 2004.

[53] V.D.PARK, AND M. S.CORSON. A highly adaptable distributed routing algorithm for mobile wireless networks. In Proceedings of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97). 1997.

[54] H.PEDERSEN. How to convert any digital signature scheme into a group signature scheme. In Proceedings of the 5th International Workshop on Security Protocols. 1997.

[55] C.E.PERKINS, AND E. M.BELDING-ROYER. Ad-hoc on-demand distance vector routing. In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99). 1999.

[56] H.PETERSEN, AND P.HORSTER. Self-certified keys—concepts and application. In Proceedings of the Third Conference on Communication and Multimedia Security. 1997.

[57] T. A.-M.QUAZI. Design and implementation of an on-demand ad-hoc routing algorithm for a positional communication system. M. S. thesis in Electronic Engineering, Department of Electrical, Electronic and Computer Engineering, University of Natal, Durban, South Africa. 2003.

[58] S.RAVI, A.RAGHUNATHAN, P.KOCHER, AND S.HATTANGADY. Security in embedded systems : Design challenges. ACM Trans. Embedd. Comput. Syst. 3, 3, 461–491. 2004.

[59] M.RAYA, AND J. P.HUBAUX. The security of vehicular ad hoc networks. In Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05). 2005.

[60] N. B.SALEM, L.BUTTYAN, J.-P.HUBAUX, AND M.JAKOBSSON. Node cooperation in hybrid ad hoc networks. IEEE Trans. Mob. Comput. 5, 4, 365–376. 2005.

[61] A.SHAMIR. How to share a secret. Commun. ACM 22, 11, 612–613. 1979.

[62] M.STEINER, G.TSUDIK, AND M.WAIDNER. Key agreement in dynamic peer groups. IEEE Trans. Parall. Distrib. Syst. 11, 8, 769–780. 2000.

[63] J.P.G.STERBENZ, R.KRISHNAN, R. R.HAIN, A. W.JACKSON, D.LEVIN, R.RAMANATHAN, AND J.ZAO. Survivable mobile wireless networks : Issues, challenges, and research directions. In Proceedings of the ACM Workshop on Wireless Security (WiSe'02). 2002.

[64] H.TAUB, AND D. L.SCHILLING. Principles of Communication Systems, 2nd Ed. McGraw-Hill, New Delhi, India. 1991.

[65] C.-K.TOH. Ad Hoc Mobile Wireless Networks : Protocols and Systems. Prentice Hall PTR, Englewood Cliffs, NJ. 2001.

[66] Y.-M.TSENG, AND J.-K.JAN. Attacks on threshold signature schemes with traceable signers. Inform. Process. Lett. 71, 1, 1–4. 1999.

[67] C.-T.WANG, C.-H.LIN, AND C.-C.CHANG. Threshold signature schemes with traceable signers in group communications. Comput. Commun. 21, 8, 771–776. 1998.

[68] G.WANG, X.HAN, AND B.ZHU. On the security of two threshold signature schemes with traceable signers. In Proceedings of Applied Cryptography and Network Security, First International Conference (ACNS 2003). 2003.

[69] T. M.WONG, C.WANG, AND J. M.WING. Verifiable secret redistribution for archive system. In Proceedings of the First International IEEE Security in Storage Workshop. 2002.

[70] T.-S.WU, AND C.-L.HSU. Cryptanalysis of group-oriented (t, n) threshold digital signature schemes with traceable signers. Comput. Stand. Interfac. 26, 5, 477–481. 2004.

[71] S.YI, AND R.KRAVETS. Practical PKI for ad hoc wireless networks. Tech. rep. UIUCDCS-R-2002-2273, UILU-ENG-2002-1717. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL. 2001.

[72] S.YI, AND R.KRAVETS. Key management for heterogeneous ad hoc wireless networks. Tech. rep. UIUCDCS-R-2002-2290,UILU-ENG-2002-1734.Department ofComputer Science,University of Illinois. at Urbana-Champaign, Urbana, IL. 2002a.

[73] S.YI, AND R.KRAVETS. Key management for heterogeneous ad hoc wireless networks. In Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP'02). 2002b.

[74] S.YI, AND R.KRAVETS, MOCA : Mobile certificate authority for wireless ad hoc networks. In Proceedings of the 2nd Annual PKI Research Workshop (PKI 2003). 2003.

[75] S.YI, AND R.KRAVETS. Composite key management for ad hoc networks. In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems : Networking and Services (MobiQuitous'04). 2004.

[76] X.ZENG, R.BAGRODIA, AND M.GERLA. GloMoSim : A library for parallel simulation for large-scale wireless networks. In Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS '98). 1998.

[77] R.ZHANG, AND H.IMAI. Round optimal distributed key generation of threshold cryptosystem based on discrete logarithm problem. In Proceedings of the Conference on Applied Cryptography and Network Security (ACNS'03). 2003.

[78] L.ZHOU, AND Z. J.HAAS. Securing ad hoc networks. IEEE Netw. (Special Issue on Network Security) 13, 6, 24–30. 1999.

[79] L.ZHOU, J.NI, AND C. V.RAVISHANKAR. Efficient key establishment for group-based wireless sensor deployments. In Proceedings of the ACM Workshop on Wireless Security (WiSe'05). 2005.

[80] P.ZIMMERMANN. The Official PGP User's Guide. MIT Press, Cambridge, MA. 1995.

[81] H.CHAN, A.PERRIG, AND D.SONG. Random key predistribution schemes for sensor networks. In Proceedings of the IEEE Symposium of Privacy and Security. 2003.

[82] W.DU, J.DENG, Y. S.HAN, AND P. K.VARSHNEY. A pairwise key pre-distribution scheme for wireless sensor networks. In Proceedings of the 10th ACMConference on Computer and Communications Security (CCS). 2003.

[83] W.DU, J.DENG, Y.HAN, S.CHEN, AND P.VARSHNEY. A key management scheme for wireless sensor networks using deployment knowledge. In Proceedings of INFOCOM'04. 2004.