République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master en informatique

Option : Réseaux et Systèmes Distribués

Thème

Collecte de certificats dans les réseaux mobiles ad hoc

Présenté par

MAMMERI Lydia

TAALBA Amel

Soutenu le 27 juin 2013 devant le jury composé de :

Dr. TARI Abdelkamel	Président	M.C.A
M. SAADI Mustapha	Examinateur	M.A.A
M^{lle}. OUADA-FARAH Sarah	Examinatrice	Doctorante
Dr. OMAR Mawloud	Encadreur	M.C.B

Promotion 2012-2013

Remerciements

Nous tenons à remercier en premier lieu notre encadreur Dr. OMAR Mawloud, pour son aide, sa patience, sa disponibilité, et ses conseils très précieux.

Nous remerciements également vont aux membres de jury qui ont accepté de juger notre travail.

Nos sincères remerciements s'adressent à nos parents, nos frères et nos soeurs pour leur soutien moral, leur encouragement inconditionnel, et surtout pour la confiance qu'ils nous accordent.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce travail :

A *mes très chers parents.*

A *mes sœurs et frères.*

A *tous mes proches et amis.*

Amel

Je dédie ce travail :

A *mes très chers parents qui ont toujours été là pour moi et qui m'ont toujours soutenu dans mes études.*

A *mes deux grand-mères, que dieu vous protège et vous garde pour nous.*

A *mes trois chers frères : Karim, Nadjim et Said que j'aime tant.*

A *ma précieuse sœur Nassima, les mots ne peuvent résumer ma reconnaissance et mon amour à ton égard.*

A *mes belles-sœurs.*

A *mes nièces et neveu.*

A *tous mes amis avec lesquels j'ai partagé mes moments de joie et de bonheur.*

A *tous mes proches.*

Lydia

Table des matières

Liste des figures

Introduction générale

Un réseau mobile ad hoc est un ensemble d'équipements mobiles interconnectés en utilisant la technologie sans fil. Il n'existe aucun point d'accès ou station de base qui les couvre, c'est la collaboration des nœuds mobiles relayant les informations qui forme l'infrastructure de ce réseau. Les réseaux mobiles ad hoc ont des caractéristiques particulières : liens de communication sans fil, contraintes d'énergie, limitation en termes de bande passante, connectivité non permanente, etc. Dû à l'absence d'infrastructure centralisée, n'importe quel utilisateur se trouvant à la portée du réseau peut s'insérer quelques soient ses intentions, ce qui rend ces réseaux vulnérables aux attaques. Pour remédier à ces attaques, plusieurs services de sécurité doivent être mis au point, tels que : l'authentification, la confidentialité, l'intégrité des données, etc. Pour assurer ces services, il est nécessaire de faire recours aux mécanismes de chiffrements. Il existe deux techniques pour le chiffrement : symétrique et asymétrique. Le chiffrement symétrique consiste à partager une clé entre deux ou plusieurs utilisateurs. Le chiffrement asymétrique consiste à attribuer une paire de clés (publique et privée) à chaque utilisateur. Chaque clé publique doit être certifiée afin de garantir son appartenance à l'utilisateur.

La plupart des solutions proposées pour la gestion des certificats dans les réseaux mobiles ad hoc se basent sur les autorités de certifications. Dû à la nature imposée par les réseaux mobiles ad hoc, le service de certification doit être distribué sur plusieurs serveurs afin de garantir la disponibilité du service de sécurité. Dans

le cadre de notre travail, nous nous intéressons à une autre catégorie de modèles de certification qui se basent sur la notion de graphe de confiance (appelés aussi : modèles de certification anarchiques). Dans ce modèle de certification, les utilisateurs eux-mêmes qui collaborent pour assurer le service de certification. Le service est assuré d'une manière distribuée par les noeuds du réseau, et chacun peut maintenir un dépôt local qui contient les certificats liés aux autres nœuds du réseau. L'authentification entre deux utilisateurs nécessite la vérification de la chaîne de certificats qui relit les deux utilisateurs. "Comment collecter les certificats nécessaires pour cette authentification" est la problématique traitée dans ce mémoire. Nous avons proposé MRCCRP (*Misbehavior Resistance Certificate Chain Recovery Protocol*) qui est un protocole de collecte de certificats résistant au comportement malveillant des noeuds malicieux qui génèrent des faux certificats dans le réseau.

Le reste de ce mémoire est structuré comme suit. Dans le chapitre un, nous présentons l'état de l'art que nous avons fait dans le cadre des protocoles de collecte de certificats. Dans le chapitre deux, nous présentons en détail notre proposition. Dans le chapitre trois, nous présentons la modélisation analytique que nous avons élaborée afin mener une comparaison en termes de robustesse entre notre protocole et deux protocoles concurrents. Enfin, nous clôturons ce mémoire avec une conclusion générale.

1

Etat de l'art sur les protocoles de collecte de certificats dans les réseaux mobiles ad hoc

1.1 Introduction

Les réseaux mobiles ad hoc ou MANET (Mobile Ad hoc NETwork) sont un ensemble de machines mobiles qui communiquent et échangent les informations entre elles à travers des ondes radio, sans aucune infrastructure. Ces réseaux possèdent des exigences spécifiques telles que la mobilité, une capacité de stockage et de calcul limité, des contraintes d'énergie. que nous allons expliquer dans ce chapitre. La caractéristique de mobilité est d'un coté un point fort qui attire progressivement

l'intention de plusieurs chercheurs dans différents domaines, et d'un autre coté un point d'échec car n'importe quel utilisateur se trouvant à portée peut s'y connecter quelque soient ses intentions, d'où la nécessité d'introduire des mécanismes de sécurité dans ces réseaux. Notre thème se dirige vers la cryptographie asymétrique, ayant comme problème la gestion et la distribution des clés publiques.

La délivrance d'un certificat de clé publique pour authentifier la clé publique d'un utilisateur est l'une des solutions proposées et sur laquelle nous nous baserons. Un certificat est un document électronique généralement délivré et géré par une tierce partie de confiance nommée autorité de certification. Selon l'existence ou non d'une autorité de certification, des modèles ont été conçus pour authentifier les utilisateurs du réseau mobile ad hoc.

Dans ce chapitre, nous nous intéressont aux modèles de confiance où aucune autorité de certification n'est employée pour assurer l'authentification entres les nœuds mobiles, et c'est les nœuds eux même qui jouent le rôle d'une autorité de certification. Ce chapitre est organisé en deux parties, nous commençons d'abord par expliquer la problématique de notre thème et présenter quelques généralités sur les réseaux mobiles ad hoc dans la première partie, ensuite dans la deuxième partie nous présentons les solutions proposées pour la collecte de certificats dans ces réseaux et leurs comparaison.

1.2 Problématique

L'élargissement du domaine d'application des réseaux mobiles ad hoc nécessite plus de sécurité pour assurer l'intégrité et la confidentialité des données qui circulent dans le réseau. Le sujet de notre thème entre dans le cadre de l'étude des problèmes de gestion de certificats dans les réseaux mobiles ad hoc. Il s'agit d'étudier des protocoles qui ont été fait pour résoudre ces problèmes, plus précisément l'étude des modèles de certification anarchiques.

Dans les modèles de certification anarchiques il n'ya aucune autorité centrale. Et c'est les nœuds eux même qui jouent ce rôle. Et pour assurer l'authentification entre une paire source-destination, il faut trouver un chemin de certification qui les relie. Et ça est considéré comme un défi a cause des contraintes imposée par les réseaux mobiles ad hoc. Les modèles de certification anarchiques sont divisés en deux modèles :

- les modèles proactifs dont le protocole d'échange de certificats est exécuté entre les noeuds voisins, et le chemin de certification est récupéré directement dans le dépôts des certificats de chaque nœud.
- et les modèles réactifs dont le protocole de collection de certificats est exécuté à la demande.

1.3 Généralité sur les réseaux mobiles ad hoc

Les réseaux mobiles ad hoc (MANETs) ne cessent d'évoluer grâce au développement de la technologie mobile. Les équipements mobiles deviennent de plus en plus petits et puissants en termes de capacité de traitement et de stockage des données. Ceci permet aux nœuds d'assurer des applications et des services plus avancés [1].

1.3.1 Définition des réseaux mobiles ad hoc

Un réseau mobile ad hoc appelé généralement MANET (*Mobile Ad hoc NETwork*) est un réseau temporaire, constitué d'une collection d'entités mobiles (ordinateurs portables, PDAs, Smartphones, capteurs, etc.) interconnectées par une technologie sans fil. Ils ne comportent pas d'infrastructure et les terminaux peuvent apparaître ou disparaître à tout moment suivant leurs capacités à communiquer ou leurs intérêts applicatifs. Un réseau ad hoc peut être utilisé dans les scénarios où l'existence d'infrastructure ne répond pas aux besoins d'applications telles que la sécurité ou le coût. Les applications militaires, les opérations de secours et les conférences sont des exemples où les réseaux ad hoc sont utilisés, mais la sécurité de la

communication est nécessaire dans ces applications. La figure 3.1 illustre un exemple du réseau mobile ad hoc.

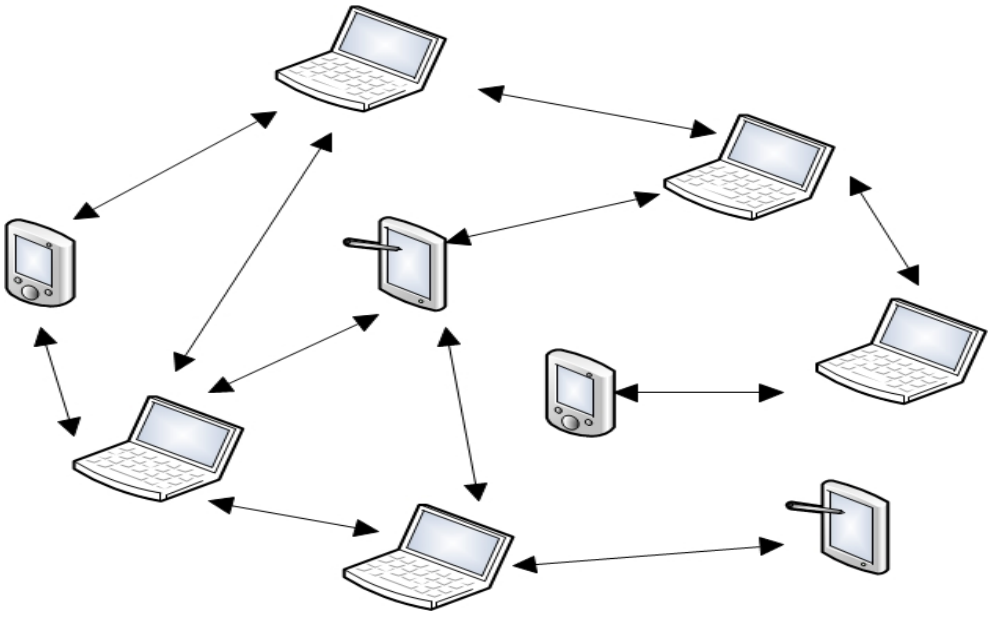


FIGURE 1.1 – Réseau mobile ad hoc.

1.3.2 Caractéristiques des réseaux mobiles ad hoc

Les réseaux mobiles ad hoc sont caractérisés par ce qui suit [2] :

1. **L'absence d'infrastructure :** absence de station de base ou de point d'accès, Cette caractéristique est fondamentale car elle définit la manière dont les unités mobiles communiquent entre elles. Un nœud joue le rôle aussi bien d'un acteur actif dans le réseau émetteur et récepteur mais aussi de routeur pour relayer la communication des autres nœuds du réseau.
2. **Topologie dynamique :** la mobilité et les mouvements des nœuds mobiles font que la topologie du réseau est dynamique car elle peut changer à tout instant de façon rapide et aléatoire. En effet, un lien de communication existant entre deux stations peut se briser à n'importe quel moment, Le mouvement des stations peut aussi conduire à la création de nouveaux liens et ainsi avoir la possibilité de remplacer les routes brisées par de nouvelles.

3. **Contraintes sur la bande passante :** Les réseaux sans fil se basent sur le partage des médiums de communication, alors la bande passante réservée à un hôte sera relativement modeste. Ceci implique des liens sans fil à capacité variable.
4. **Contraintes d'énergie :** Les unités mobiles sont alimentées par des sources d'énergie autonomes (exemple : batteries), ce qui réduit leurs temps de disponibilité dans le réseau. Par conséquent, la consommation d'énergie constitue un véritable problème. Les mécanismes de gestion d'énergie sont nécessaires pour les nœuds dans le but de conserver l'énergie et d'augmenter leur durée de vie. Donc, n'importe quelle solution destinée aux réseaux mobiles Ad hoc doit prendre en compte la contrainte de l'énergie.
5. **Sécurité limitée :** les réseaux mobiles Ad hoc sont considérés comme étant très fragiles en matière d'attaques en tout genre. Les pirates informatiques peuvent intercepter les données d'une manière directe en utilisant des antennes pirates (car les données circulent par voie hertzienne) ou bien obliger une station à consommer une bonne partie de ses ressources d'énergie en l'inondant de toutes sortes de requêtes inutiles.

1.4 Gestion des certificats dans les réseaux mobiles ad hoc

1.4.1 Quelques Définitions

- **Authentification** : L'authentification a pour but de vérifier l'identité dont une entité se réclame. Généralement l'authentification est précédée d'une identification qui permet à cette entité de se faire reconnaître du système par un élément dont on l'a doté. En résumé, s'identifier c'est communiquer son identité, s'authentifier c'est apporter la preuve de son identité. En d'autre terme

La définition la plus large de l'authentification dans les systèmes informatiques englobe la vérification de l'identité.

- **Certificat de clé publique** : Fichier électronique attestant qu'une clé publique appartient à l'entité qu'il identifie (personne physique ou morale ou entité matérielle). Il est délivré par une tierce partie de confiance appelé autorité de certification. L'Autorité de certification en signant le certificat, valide le lien entre une entité identifiée et une bi-clé cryptographique. Le format des certificats est donné comme suit :
  - Version du certificat.
  - Identificateur unique du certificat.
  - Le nom de l'algorithme de signature utilisé.
  - Le nom unique de l'émetteur du certificat.
  - Le nom unique du détenteur du certificat.
  - La clé publique du détenteur du certificat.
  - La période d'expiration du certificat d'attribut.
  - La liste des groupes dont l'accès est autorisée au détenteur du certificat.
  - Indications concernant la nature du certificat.
  - La signature de l'autorité de certification.
- **Autorité de certification (CA)** : c'est la tierce partie de confiance chargée de la création et la signature de certificats. L'autorité de certification a également pour mission la signature des listes de révocations CRL (Certificate Revocation List).
- **Service de certification** : Un ensemble de fonctionnalités liées à la certification qui sont assurées par des nœuds spéciaux dans le réseau et aux qui s'adressent les nœuds clients afin de procurer ou vérifier la validité d'un certificat.
- **Disponibilité du service de certification** : le degré d'accessibilité des nœuds qui assurent le service de certification dans le réseau.

- **Chaîne de certificats** : un ensemble ordonné des certificats nécessaires pour vérifier la filiation d'un certificat à un porteur.

1.4.2 Problèmes de gestion de certificats dans les réseaux mobiles ad hoc

La gestion des certificats dans les réseaux mobiles ad hoc présente des problèmes liés aux contraintes imposées par la nature du réseau : mobilité, limitation de ressources, absence totale ou partielle d'infrastructure, etc. Les problèmes à étudier touchent les points suivants :

- **La disponibilité du service de certification :** dans les réseaux mobiles ad hoc, proposer une seule autorité de certification fixe pour tout le réseau n'est pas une solution souhaitable, car si cette autorité est compromise tout le réseau sera compromis et cette autorité n'est pas adaptée à la dynamique de la topologie du réseau ce qui entraverait la disponibilité des services de sécurité. Ainsi, l'une des exigences fondamentales est que le service de certification puisse assurer la disponibilité du service malgré les éventuelles déconnexions voire partitionnement du réseau.
- **La consommation de ressources :** les nœuds d'un réseau mobile ad hoc peuvent être réduits à des systèmes embarqués mobiles disposant de peu de ressources d'énergie, de bande passante, de stockage et de calcul. De ce fait, les protocoles du service de certification doivent être acceptablement optimisés en termes de calcul, communication, et stockage.
- **La scalabilité :** diverses applications dans les réseaux mobiles ad hoc impliquent un nombre important d'utilisateurs. Dans une telle condition, si le service de certification est assuré par une autorité centrale, cette dernière peut devenir surchargée à cause du nombre important de requêtes de certification. Autrement, si le service de certification est assuré d'une manière distribuée sur plusieurs nœuds du réseau, chaque participant doit maintenir un dépôt local,

qui contient les certificats liés aux autres nœuds du réseau. Par conséquent, le stockage des certificats serait proportionnel à la taille du réseau, ce qui va compromettre les performances du service de certification à large échelle.

- **La gestion de l'hétérogénéité :** comme dans le cas des réseaux avec infrastructure, les autorités de certification pourraient être hétérogènes, de même dans les réseaux mobiles ad hoc. Ceci signifie que deux nœuds ou plus appartenant à différents domaines peuvent avoir le besoin de s'authentifier. Dans ce cas, il doit y avoir une certaine relation de confiance entre les deux domaines.

1.4.3 Classification des protocoles de gestion de certificats dans les réseaux mobiles ad hoc

Pour assurer le service de certification dans les réseaux mobiles ad hoc en respectant les caractéristiques cités précédemment, deux grandes catégories de modèles existent : (1) Modèle Autoritaires, dans cette catégorie le service de certification est réalisé par une ou plusieurs autorités de certifications et (2) Modèles anarchiques, dans cette catégorie de modèles, il n'y a aucune autorité centrale, chaque utilisateur dans le système agit en tant qu'autorité de certification. La propagation de la confiance dans le réseau forme un graphe entre les utilisateurs, appelé graphe de confiance (ou web-of-trust), qui est géré par les utilisateurs eux-mêmes. Ce modèle décentralisé dans sa nature, est bien adapté aux réseaux mobiles ad hoc. Dans cette catégorie, deux opérations principales sont nécessaires : (1) la construction du graphe de confiance, et (2) la découverte des chaînes de certificats. Cette catégorie est divisée en deux sous-catégories qui sont : les modèles proactifs et et les modèles réactifs. Dans ce qui suit on s'intéresse aux modèles de certifications anarchiques, et la figure 3.2 présente la classification des protocoles que nous avons étudiés.

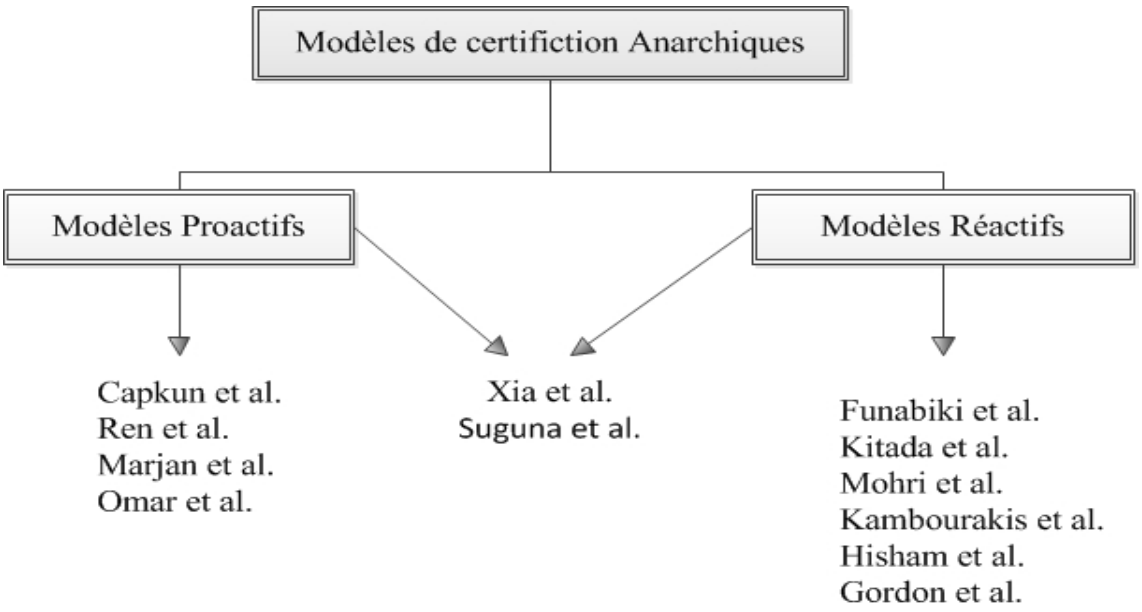


FIGURE 1.2 – Classification des modèles de confiance anarchiques dans les réseaux ad hoc

1.5 Protocoles de gestion des certificats dans les réseaux mobiles ad hoc

Dans ce qui suit nous avons effectué l'étude des différents protocoles de certification anarchique. Ces protocoles sont divisés en modèles proactifs et modèles réactifs.

1.5.1 Modèles proactifs

Dans cette sous-catégorie, chaque nœud dispose d'un dépôt de certificats qui contient la pluparts des certificats délivrés dans le réseau, et à partir duquel il peut construire la chaîne de certificats désirée.

- **Solution de Capkun et al.**

  Capkun et al. [3] ont proposé un service de certification distribué sur tous les nœuds du réseau. Dans ce modèle, il est supposé qu'il existe des relations de confiance sociales entre les différents utilisateurs. Chaque utilisateur génère sa propre paire de clés (privée, publique) et sollicite un ensemble d'utilisateurs

afin de lui délivrer des certificats correspondants à sa clé publique. Si un utilisateur u croit que la clé publique K_v appartient à l'utilisateur v, il lui délivre un certificat signé par sa propre clé privée, les certificats sont délivré avec une durée de validité limitée et chaque certificat contient sa date de délivrance et date d'expiration. De ce fait, un graphe de confiance sera établi entre tous les utilisateurs du réseau. Egalement, chaque nœud du réseau maintient un dépôt local de certificats, qui est mis à jour périodiquement à travers un protocole d'échange de certificats entre les nœuds voisins. Quand un utilisateur u nécessite d'authentifier la clé publique d'un utilisateur v, les deux nœuds fusionnent leurs dépôts locaux et essayent de trouver une chaîne de certificats à partir de l'utilisateur u vers l'utilisateur v dans le dépôt fusionné (cf. Fig. 3.3). Si une telle chaîne n'est pas trouvée, u peut solliciter les helper nodes, qui sont les nœuds voisins à un ou deux sauts. Chaque utilisateur peut révoquer un certificat qu'il a délivré s'il croit que la clé publique lié à un utilisateur par ce certificat n'est plus valide, également si un utilisateur croit que sa propre clé privée est compromise, il peut révoquer la clé publique correspondante.

L'avantage majeur de ce modèle est l'autonomie du service de certification où aucune autorité n'est employée pour assurer le service de certification. Cependant, le graphe de confiance établi entre les utilisateurs peut ne pas être fortement connecté, ce qui peut empêcher l'authentification de certains utilisateurs du réseau. Un autre problème lié à ce modèle est la consommation de ressources. En effet, ce modèle provoque une charge importante de calcul induite par la vérification des chaînes de certificats à chaque authentification. Egalement, chaque utilisateur est requis de maintenir un dépôt local de certificats, qui est enrichi systématiquement par le protocole d'échange de certificats, ce qui provoque une charge importante de stockage et de transmission.

- **Solution de Ren et al.**

  Ren et al. [4] ont proposé une version modifiée du modèle de Capkun et al.

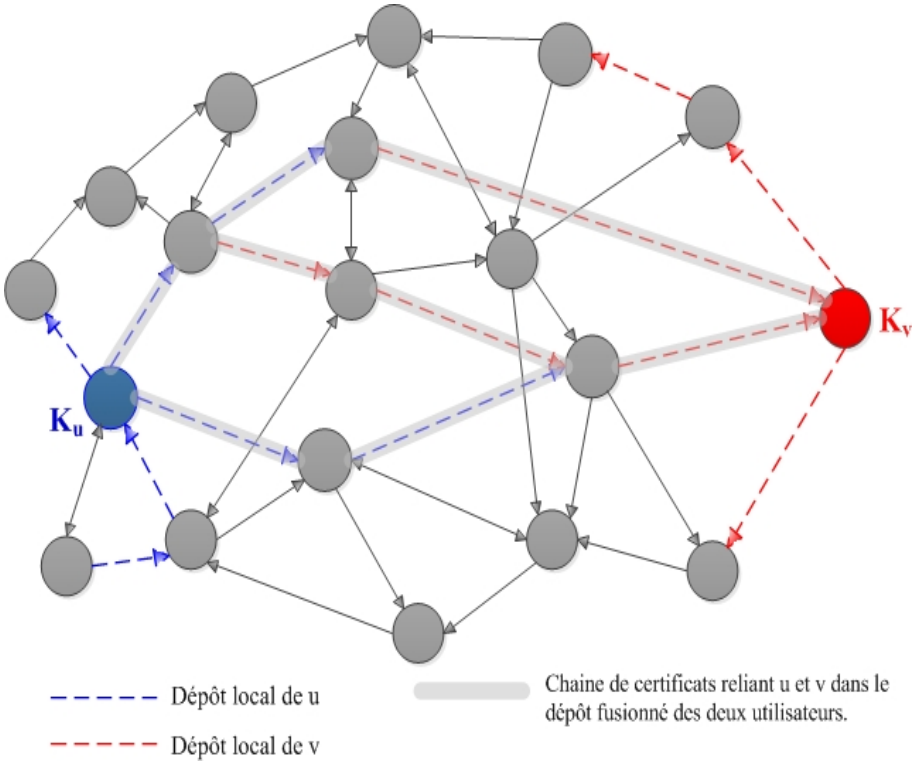


FIGURE 1.3 – capkun et al.

en se basant sur un serveur central pour initialiser le service de certification. Ce serveur initialise le système en distribuant pour chaque nœud une liste contenant un ensemble d'identificateurs de quelques utilisateurs et leurs clés publiques. Ensuite, chaque nœud génère les certificats correspondants. Ainsi, un graphe de confiance sera établi entre les utilisateurs du réseau, et le système continuera son fonctionnement indépendamment du serveur central où l'authentification des nœuds sera assurée à travers des chaînes de certificats.
Comme le modèle de Capkun et al., l'avantage majeur de ce modèle est l'autonomie du service de certification qui ne dépend pas d'une autorité centrale de certification. Cependant, le service de certification est dépendant toujours d'un serveur central à l'étape d'initialisation. En plus, l'efficacité du service de certification de ce modèle est fortement liée à la longueur des listes délivrées par le serveur central. Plus la longueur est grande, plus chaque nœud génère un nombre important de certificats, ce qui augmente la disponibilité du service

de certification. Mais dans ce cas, la charge de stockage sera importante.

- **Solution de Marjan et al.**

  Marjan et al. [11] ont proposé une solution améliorant les deux présentées précédemment, Dans ce modèle l'établissement des paires de clé (publique, privée) se fait localement par les nœuds joignant le réseau, ensuite chacun exécute un mécanisme de près-authentification en utilisant *le side channel* (un canal auxiliaire discret permettant l'échange de la preuve de confiance) pour prouver ses bonnes intentions à ses voisins, et s'ils le croient, ils établissent avec lui une relation de confiance en échangeant leurs clés publique. Ensuite vient l'étape de délivrance de certificats entre les nœuds, ainsi la construction des dépôts locaux de certificats qui ne contiennent que des certificats qu'un nœud délivre et ceux délivrés pour lui. Les auteurs estime qu'un dépôt de certificats associé a chaque nœud dispose d'un espace mémoire fixe, après la construction des dépôts locaux des certificats pour chaque utilisateur, une procédure d'échange de certificats se déclenche et ne s'arrête que si l'espace mémoire des dépôts est plein ou si les nœuds obtiennent tout les certificats du réseau. Pour l'authentification entre deux nœuds S et D, S vérifie d'abords dans son dépôt s'il contient tout les certificats qui mènent vers D, alors il établit lui même le chemin désiré, sinon il le fusionne avec les dépôts des nœuds intermédiaires.

  Ce modèle offre un service d'authentification robuste en effectuant d'abords une près-authentification par le side channel, ensuite l'authentification par le service de certification. Avec l'absence de toute autorité centrale, ce modèle offre un service de certification autonome, mais il peut provoquer une surcharge du stockage vu que les dépôts des nœuds dispose d'un nombre important de certificats.

- **Solution de Omar et al.**

  Le modèle de certification proposé par Omar et al. [6] est totalement distribué,

où la gestion des certificats est exécuté par les nœuds eux même, il se base sur des graphes de confiance où l'authentification de clé publique est établie via des chaînes de certificats. L'idée principale de la solution proposée dans ce protocole est l'inclusion du schéma de cryptographie à seuil (k, n) dans le graphe de confiance afin de résister contre les faux certificats délivré par des nœuds malveillant dans le réseau, chaque nœud i possède sa part privée Si qui est utilisée pour signer les certificats. Dans cette approche une chaîne de certificats partiels est représentée par un graphe direct G (V, E) qui s'appelle graphe de confiance partiel, où V représente les sommets et E les liens. Dans ce graphe les sommets correspondent aux nœuds et les liens correspondent aux certificats partiel, un lien direct entre un sommet i et un sommet j existe si et seulement s'il existe un certificat partiel signé par Si pour relier Kj (clé publique du nœud j) au nœud j. L'approche proposée contient quatre opérations de base :

1. **Phase d'initialisation :** dans cette phase un système dealer est introduit, qui est commun entre tous les membres du système. Durant cette phase chaque nœud obtient sa part privée Si à partir du systeme dealer, ensuite chaque membre génère des certificats partiels aux membres auxquels il fait confiance dans le système.
2. **Joindre le système :** quand un nœud veut joindre le réseau, un groupe composé d'au moins k membres collaborant pour permettre l'accès à ce nouveau nœud dans le système.
3. **Création et échange de certificat partiel :** l'échange de certificat est un mécanisme important qui permet aux nœuds de partager et distribuer les certificats qu'ils délivrent. Le protocole d'échange de certificat est exécuté périodiquement entre les nœuds et leurs voisins. Si un utilisateur i croit que la clé publique Kj appartient à l'utilisateur j, alors l'utilisateur i peut délivrer un certificat pour l'utilisateur j signé avec la part privée Si

de i. un protocole d'échange de dépôt local est exécuté systématiquement entre les nœuds voisins dans le réseau.

4. **Authentification de clé publique :** l'authentification des clés publiques entre les nœuds est effectuée par les chaînes de certificats partiels. Quand un nœud i désir authentifier la clé publique Kj de l'utilisateur j, les deux nœuds fusionnent leurs graphes de confiance partiel, et valident ce graphe en respectant le modèle de confiance basé sur le schéma de cryptographie à seuil, puis le nœud i essaye de trouver une chaîne de certificat du nœud i vers j dans le graphe partiel validé, et s'il trouve cette chaîne alors l'authentification sera effectuée.

Avec ce modèle un adversaire externe n'a aucune façon pour se faire passer à un nœud membre, et il ne peut pas signer un faux certificat partiel sous l'identité d'un nœud membre valide, car pour pouvoir signer un certificat partiel le nœud doit avoir sa part privée correspondant au schéma de cryptographie a seuil. Par contre un adversaire interne peut délivrer plusieurs types de faux certificats, mais l'approche proposée dans ce modèle résiste contre ces faux certificats grâce à l'utilisation des techniques de cryptographie à seuil, d'où la robustesse du système. Les auteurs dans ce modèle ont étudié aussi la valeur optimale du seuil k pour augmenter la disponibilité du service de certification. Les inconvénients de cette approche est que le service de certification est dépendant d'un administrateur (system dealer) à l'étape d'initialisation, en plus la vérification des chaînes de certificats à chaque authentification provoque une charge importante du calcul.

1.5.2 Modèles réactifs

Dans cette sous-catégorie, le protocole de collection de certificats s'exécute à la demande. Quand un nœud a besoin de vérifier un certificat, à cet instant il collecte à travers un protocole distribué la chaîne de certificats appropriée.

- **Solution de Funabiki et al.**

  Funabiki et al. [7] ont proposé un modèle de certification basé sur la clustérisation, Le réseau est constitué de clusters dans chacun : ils existent des nœuds qui délivrent des certificats et un nœud CMN (Certificate Management Node) qui a pour rôle de gérer les certificats et de les stocker dans son propre dépôt. Les nœuds CMNs sont choisis à travers une phase de sélection du nœud ayant un degré maximum de voisinage. Pour établir un chemin de certification d'un nœud source S à un nœud destinataire D, d'abord le nœud S sollicite le CMN de son cluster qui vérifie dans son dépôt s'il dispose d'une chaîne vers la destination sinon il va solliciter d'autres CMNs jusqu'à l'arrivée au CMN du nœud D. Le schéma suivant (cf. Fig. 1.4) présente le protocole de collection des certificats présenté par Funabiki et al. :

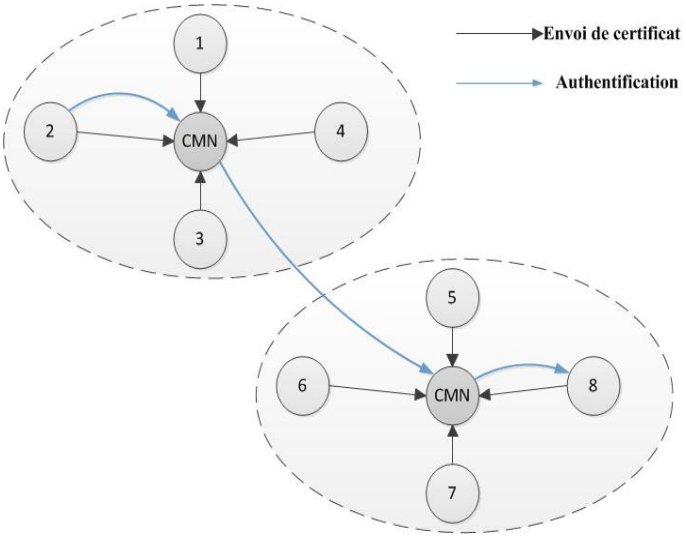


FIGURE 1.4 – Funabiki et al.

Dans ce modèle les nœuds CMNs sont d'un coté un point fort car ils stockent tous les certificats délivrés dans leurs cluster, au lieu que les autres nœuds le font, de cette manière ce modèle offre une disponibilité du service de certification élevée par rapport aux autres protocoles du modèle réactif, et d'un autre coté un point faible car si ces nœuds tombent en panne ou devient compromis ou se déconnecte, cela touche à l'intégrité des certificats. En plus cette idée

converge le modèle proposé vers les modèles centralisés.

- **Solution de Kitada et al.**

  Kitada et al. [14] ont proposé d'établir un graphe de confiance suivant les étapes suivantes :

  1. Création des paires de clé par les noeuds eux même.
  2. Délivrance des certificats de clés publique, ainsi la construction du graphe de confiance.

  Pour l'établissement du chemin de certification depuis un nœud S au nœud D, les auteurs ont proposé d'exécuter deux étapes :

  Etape 1 *Diffusion du paquet de recherche* : Un paquet de recherche est une requête diffusée par le nœud source S à tous les nœuds pour lesquels il a déjà signé un certificat. Tout nœud intermédiaire i qui reçoit cette requête, ajoute son propre certificat et la diffuse à tous les nœuds pour lesquels il a déjà signé un certificat, jusqu'à ce que la requête atteigne la destination D.

  Etape 2 *construction du chemin de retour* : Quand le nœud D reçoit la requête du nœud S, il ajoute aussi son certificat de clé publique et envoie le paquet directement au nœud S, ainsi le nœud S obtient la chaîne de certificats nécessaire pour l'authentification. La figure 1.5 illustre le fonctionnement du protocole d'établissement des chaînes de certificats présenté par Kitada et al. L'avantage principal du modèle de Kitada et al.[14] est que les nœuds ne maintiennent pas des dépôts de certificats. Cependant, ce système provoque une charge importante de transmission, vu que la recherche des chaînes de certificats est faite à travers la diffusion d'un ensemble de certificats qui ne s'arrêtera qu'à l'aboutissement du noeud destinataire. Si le destinataire n'est pas abouti, le processus de recherche pourra s'exécuter indéfiniment.

- **Solution de Mohri et al.**

  Mohri et al. [13] ont proposé une version améliorée du modèle de Kitada et al. [14]. Ils ont proposé de diviser le processus de collection des chaînes de

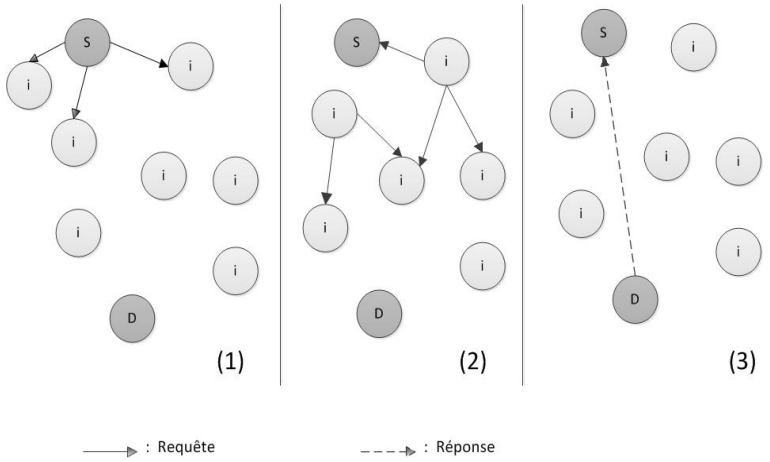


FIGURE 1.5 – Kitada et al.

certificats en deux phases :

1. **Phase de recherche de certificat :** le problème à résoudre dans cette phase est de trouver un seul chemin de certification du nœud source au nœud destinataire, afin de trouver ce chemin le nœud source construit un arbre couvrant en utilisant un algorithme distribué. l'objectif de cet algorithme est que la requête du nœud source arrive à tous les nœuds intermédiaires une seule fois. Chaque nœud intermédiaire ajoute seulement son identifiant à cette requête et l'envoi au nœud suivant. A la fin de ce processus le nœud destinataire reçoit une chaîne d'identificateur de nœud source vers lui, et à partir de celle la il peut passer à la phase de collecte de certificats.

2. **Phase de collecte de certificats :** quand le nœud destinataire reçoit le paquet de recherche, chaque nœud dans l'arbre connait son nœud père. Pour que le nœud source obtienne tous les certificats dans une chaîne de certificat, ils ont proposé la méthode suivante :

   - Le nœud destinataire envoie un paquet réponse contenant son certificat à son nœud père.
   - Chaque nœud intermédiaire qui reçoit le paquet ajoute son propre certificat au paquet et l'envoi aussi à son nœud père. Ce processus est

répété jusqu'à ce que le paquet atteigne le nœud source. Lorsque ce processus est terminé, le nœud source obtient tous les certificats dans une chaîne de certificats.

En comparant avec le modèle de kitada et al.[14] celui-ci minimise la charge de transmission en évitant de rajouter les certificats dans la première phase ce qui permet d'avoir un coût de communication minimale. Cependant, l'obtention d'un seul chemin de certification peut empêcher tous le processus d'authentification, car si l'un des nœuds intermédiaires est compromis ou se comporte comme un nœud malicieux en délivrant des faux certificats, c'est toute la chaine de certificats qui sera rejetée.

- **Solution de Kambourakis et al.**

  La solution de Kambourakis et al. [8] est une amélioration de celle de Kitada et al.[14], ils ont supposé l'existence d'un graphe de confiance entres les nœuds du réseau sous forme d'un arbre binaire, deux méthodes ont été proposées pour la construction de l'arbre binaires.

  1. **création d'arbre :** se déclenche quand un nœud initiateur envoie des invitations à ses voisins indiquant qu'il veut exécuter le protocole dédié, chaque nœud qui reçoit l'invitation et l'accepte l'envoie à son tour à ses voisins. Pour garantir la création d'un arbre binaire deux conditions sont à respecter : (1) Un nœud parent ayant reçu deux réponses positives à ses invitations doit ignorer toute les autres, et (2) Un nœud fils ayant accepté une invitation doit ignorer toute les autres.
  2. **création d'arbre binaire basé sur le parallélisme :** suit les mêmes étapes que la méthode précédente avec la différence que plus d'un seul nœud qui déclenche le protocole, ainsi chaque nœud abouti à la construction d'un sous-arbre binaire de taille petite et qui porte son identité id-racine (identité du nœud racine), ensuite pour former un seul arbre binaire les sous-arbres sont réunis suivant deux conditions : (1) Un nœud qui re-

çoit une invitation pour joindre un arbre qui porte le même id-racine que son arbre doit la rejeter, et (2) Un nœud ne peut pas accepter plus de trois invitations.

La procédure de découverte des chaînes de certificats suit 3 etapes :

(1) Un nœud initiateur S diffuse une requête au nœud destinataire D, chaque nœud intermédiaire i recevant la requête, ajoute son certificat et la diffuse.

(2) A la réception de la requête par le nœud D, il obtient la chaîne necessaire pour répondre au nœud source S. Chaque nœud intermédiaire i dans l'arbre recevant la réponse du nœud D, vérifie le certificat de son expéditeur intermédiaire, et s'il est valide, ajoute son certificat à cette réponse, et l'envoie nœud suivant jusqu'à ce qu'elle atteigne le nœud cible.

(3) Le nœud S reçoit donc le certificat du nœud D et lui envoie le sien de la même façon que dans 2. Et les deux nœuds seront certifiés mutuellement. La figure 1.6 illustre le fonctionnement du protocole d'établissement des chaînes de certificats présenté par Kambourakis et al.

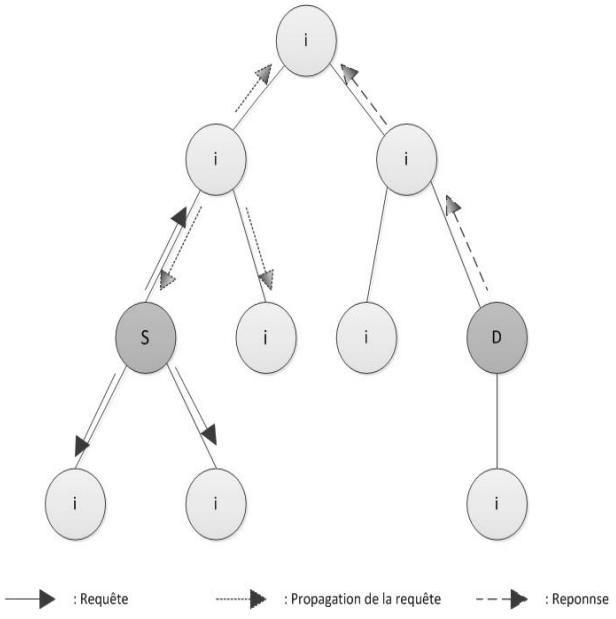


FIGURE 1.6 – Kambourakis et al.

Ce protocole permet le partage de confiance entre les nœuds de l'arbre, et offre un service de certification entre chaque nœud père et ses deux fils. La

structure en arborescence a facilité la découverte de chaîne des certificats, mais vu la mobilité des nœuds il est impossible de maintenir la topologie en arbre. En plus si un nouveau nœud veut joindre le réseau, il ne l'effectuer que par les feuilles ou par les nœuds possédant un seul fils, donc ce modèle peut empêcher certains nœuds de s'authentifier. Un autre problème provoqué par ce modèle est la consommation de ressources, car il implique une charge importante de calcul induite par la vérification des chaînes de certificats à chaque authentification. En plus la structure de l'arbre ne peut collecter qu'un seul chemin de certification entre chaque paire source-destination.

- **Solution de Hisham et al.**

  Dans ce modèle les auteurs [9] ont supposé que la confiance est partagée mutuellement et directement entre certains nœuds voisins, ce nombre est supposé être réparti uniformément. Chaque utilisateur génère localement sa propre paire de clés (privée, publique) et si un nœud croit que la clé publique d'un autre nœud lui appartient, alors il lui délivre un certificat signé par sa clé privée, ce qui permet la création d'un graphe de confiance entre les utilisateurs du réseau. Si un nœud S a besoin de s'authentifier avec un nœud D, il diffuse une requête à ses voisins avec qui il a une relation de confiance directe, et chaque nœud intermédiaire recevant cette requête ajoute son certificat qu'il a déjà signé a son expéditeur, et la diffuse également à ses voisins avec qui il a une relation de confiance directe, et la requête se propage de cette manière dans le réseau jusqu'à ce qu'elle atteigne le nœud D. A la réception de la requête du nœud S, le nœud D obtient la chaîne de certificats nécessaire qu'il utilisera pour répondre au nœud S. Dans le cas d'obtention de plusieurs chaînes le nœud source choisie celle ayant la longueur minimale. La figure 1.7 illustre le fonctionnement du protocole d'établissement des chaînes de certificats présenté par Hisham et al.

  Avec l'absence d'une autorité centrale ce modèle offre un service de certification

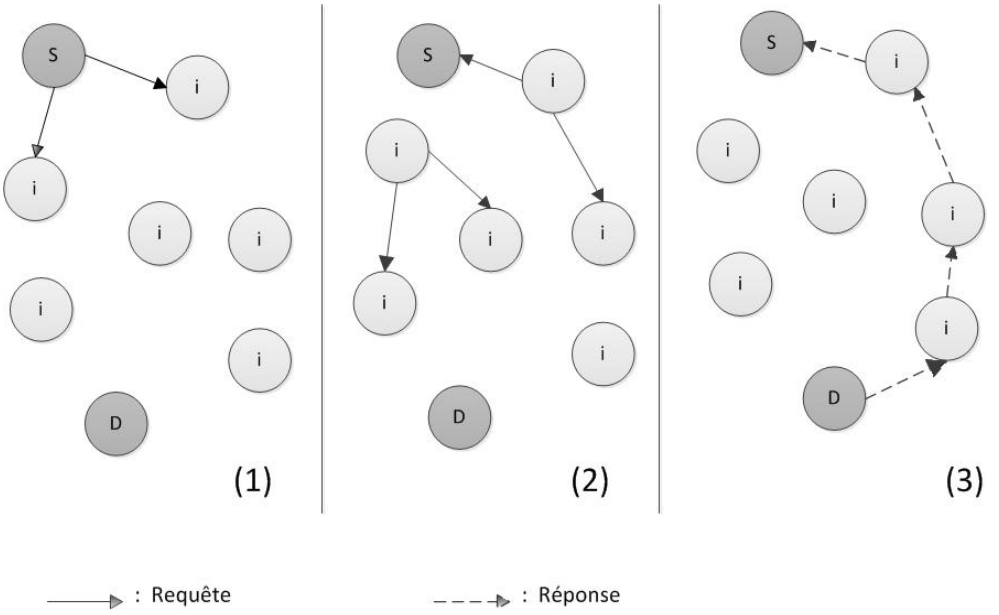


FIGURE 1.7 – Hisham et al.

autonome, en revanche il provoque une charge importante de transmission vu que le message est diffusé à chaque fois, et le nœud destinataire peut recevoir plusieurs fois un même message empruntant des chemins différents, et si le nœud destinataire n'est pas abouti le processus peut boucler indéfiniment. En plus les auteurs ont supposé que le nombre de relations de confiance directe que possède chaque nœud avec ses voisins est fixe donc le graphe de confiance établi entre les utilisateurs peut ne pas être connexe, ce qui peut empêcher l'authentification de certains utilisateurs du réseau.

- **Solution de Gordon et al.**

  Gordon et al. [12] ont également pensé à améliorer la solution de hisham et al.[9], dans ce modèle les nœuds crées eux mêmes leurs propre paires de clés avant de joindre le réseau, ensuite un graphe de confiance est établit comme résultat de la délivrance des certificats entre les nœuds du réseau, chaque certificat sera stocké dans le dépôt local du nœud qui l'a délivré et celui de l'utilisateur qui l'a reçu. Pour l'authentification entre deux nœuds Source S et Destinataire D, le nœud S diffuse la requête d'authentification contenant son certificat à tout ses voisins, quand un nœud intermédiaire i reçoit cette

requête vérifie dans son dépôt local s'il possède le certificat du destinataire s'il le trouve, il envoie deux types de messages : le premier est une réponse au nœud S avec le certificat du destinataire, et le deuxième est envoyé au nœud destinataire avec le certificat du nœud S, ainsi les nœuds obtiennent une chaîne indirecte de certification, mais si les nœuds intermédiaires ne connaissent pas la destination, ils diffusent la requête à leurs voisins jusqu'à ce qu'elle arrive au destinataire qui obtient une chaîne directe de confiance à travers laquelle il peut répondre au nœud source. La figure 1.8 illustre le fonctionnement du protocole de d'établissement d'une chaîne indirecte présenté par Gordon et al.

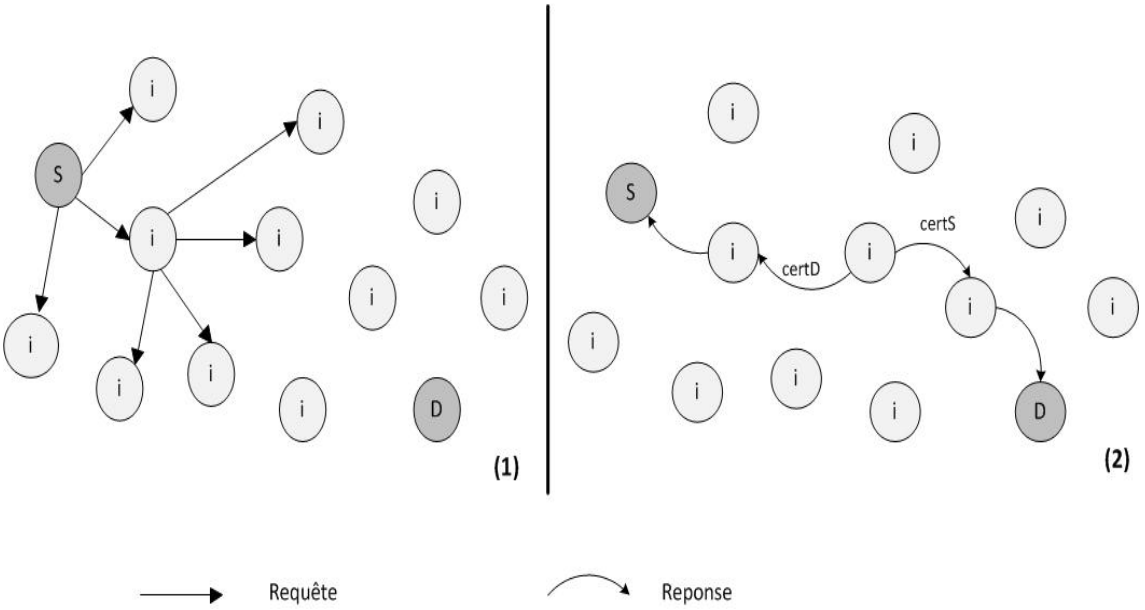


FIGURE 1.8 – Gordon et al.

Cette solution offre un service de certification distribué et autonome vu que c'est les nœuds eux mêmes qui assurent le service de certification sans aucune autorité centrale, mais les chaînes indirecte de confiance n'assurent pas un vrai chemin vers la destination et suffisent par informer la source qu'un autre nœud a authentifié la clé publique du destinataire en lui envoyant son certificat, ce qui rend ce modèle convergeant vers les modèles proactifs, en plus si un nœud malicieux intercepte la requête, il peut répondre au nœud source par un faux certificat ce qui touche à la robustesse du modèle.

- **Solution de Xia et al.**

  OPKM [5] est un protocole de certification executé par les nœuds eux même sans avoir besoin d'une autorité centrale. Il utilise la technologie de diffusion broadcast pour distribuer les certificats de clé publique pour tout les nœuds du réseau, et utilise la signature numérique pour vérifier les messages diffusés contre l'attaque " man-in-the-middle ". Dans OPKM, chaque nœud qui joint le réseau pour la première fois effectue un processus proactif en diffusant le certificat de sa clé publique à ses voisins à un saut. Chaque nœud dans le réseau maintient deux tables pour stocker les certificats de clé publique des autres nœuds, une table s'appelle table de certificat de voisinage (neighbourhood certificate table), dans laquelle un nœud stocke les certificats de clé publique de ses voisins à deux sauts. L'autre table s'appelle table de certificat réseau (network certificate table), dans laquelle un nœud stocke tout certificat de clé publique des nœuds disponible dans le réseau. Pour la génération de certificat dans OPKM, chaque nœud i délivre un certificat de sa clé publique Ci qui contient IDi (identifiant du nœud i), PUKi (la clé publique de nœud i), période de validité du certificat et leurs signatures numérique Di. Di est obtenue par le nœud i en appliquant la fonction de hachage pour le contenue de Ci et chiffrer le hacher avec PRKi (la clé privée du nœud i). Pour la distribution des certificats les auteurs ont proposé deux processus : Un processus s'appelle " neighbourhood certificate distribution ", et l'autre s'appelle " multi-hop certificate distribution ". " neighbourhood certificate distribution " est un processus proactif exécuté par un nœud quand il joint le réseau ou change de position. La distribution de certificat de voisinage (neighbourhood certificate distribution) du nœud i est effectuée par les 3 étapes suivantes :

  Etape1 : le nœud i diffuse un message Mi pour tous ses voisins à 1-saut, telle que Mi contient la requête PKC-REQ et le certificat du nœud i.

  Etape2 : chaque nœud j qui reçoit le message PKC-REQ en premier, valide le

message Mi en vérifiant la signature numérique du nœud i avec la clé publique PKUi contenue dans Mi. Chaque nœud j appartenant au voisin de i à 1-saut met à jour ses table de certificats de voisinage et table de certificats réseau et rebroadcast le message de réponse PKC-REP contenue dans le message Mj, Mj contient le certificat de j et les certificats des voisins de j à 1-saut.

Etape3 : le nœud initial i met à jour ses tables de certificats par rapport aux informations dans le message de réponse Mj envoyé par j. L'autre processus de distribution de certificat est Multi-hop certificate distribution (distribution de certificat multi saut) ce processus est conçu pour les nœuds qui sont à deux sauts ou plus pour échanger leurs certificats de clé publique, se base sur le processus de distribution de certificat au voisinage (neighbourhood certification distribution). Son principe est que chaque nœud intermédiaire ne rediffuse pas un message qui à été déjà transféré par ses deux voisins à 1-saut. Quand un nœud S veut échanger le certificat de sa clé publique avec le nœud destinataire D, il initialise le processus de distribution en diffusant un message MDs pour ces voisins à 1-saut. Le message MDs contient le certificat de S et l'identifiant IDd du destinataire D. chaque nœud i appartenant aux voisins de S a 1-saut valide le message MDs puis ajoute sont certificat Ci dans le message MDs et chiffre le tous avec sa clé privée. Maintenant le message est organisé comme un nouveau message MDi qui sera diffusé par le nœud i pour ses voisins a 1-saut. Ensuite tous les autres nœuds intermédiaires continuent le processus de distribution jusqu'à ce que le message de certification transféré atteigne le nœud destinataire D. La vérification des certificats dans OPKM est exécutée par les deux processus précédent. L'avantage de ce protocole est l'autonomie du service de certification, et il est capable de fournir un service de gestion de clé à la demande pour échanger les certificats dans les communications multi-saut. Il est aussi flexible pour s'adapter à la mobilité des nœuds dans le réseau. Et il peut défendre et prévenir de certains attaque (exp : l'attaque "

man-in-the-middle "). Cependant ce modèle provoque une charge de stockage car chaque nœud dans le réseau maintient une table de certificat de voisinage et une table de certificat réseau d'où il n'assure pas la scalabilité. Ce modèle n'est pas optimisés en termes de calcul car l'utilisation de cryptage de clé publique asymétrique provoque une perte de temps pendant les calculs dans chaque nœud.

- **Solution de Suguna et al.**

  Suguna et al. [10] ont proposé un modèle de certification hybride, où l'établissement des paires de clés se fait localement par chaque participant dans le réseau, et la génération des certificats se fait par approche comportementale ce qui permet la construction d'un graphe de confiance entre les utilisateurs du réseau, les auteurs ont ajouté une métrique nommée *Lien stable* qui donne la durée entre l'établissement d'un lien entre deux nœuds donnés et sa rupture, cette valeur dépond de la distance entre les deux nœuds, leurs vitesses de mobilité et leurs directions. Comme le modèle de hisham et al.[9] ce modèle se base sur la diffusion d'une requête par le nœud source S désirant s'authentifier avec un nœud destinataire D. Le nœud D répond par son certificat signé suivant le chemin spécifié dans la requête reçue, chaque nœud intermédiaire i ajoute à cette réponse son certificat et la valeur *lien stable* entre lui et l'expéditeur de la réponse. A la réception de toute les réponses le nœud S choisit celle ayant la valeur *lien stable* la plus haute pour l'authentification. Et l'ensemble des certificats participant a cette chaîne seront stockés dans la mémoire cache du nœud source pendant la durée du *lien stable*. Ce modèle offre un service de certification distribué et autonome, les dépôts de certificats contiennent les certificats qu'un nœud délivre pour d'autres nœuds et ceux délivrés pour lui.mais peut impliquer une charge de stockage a des moments données a cause du stockage des chaînes dans les mémoires caches. La majorité des problèmes rencontrés dans ce modèle sont liés à la diffusion des requêtes qui peuvent saturer le ré-

seau, en plus avec la métrique ajoutée, la chaîne de certification choisie aura une durée de vie plus longue, mais l'utilisateur peut finir par choisir une longue chaîne alors que d'autres plus courtes existent, en plus cette valeur peut ne pas être exacte car la vitesse de mobilité des nœuds est variable. Comme il peut alourdir le protocole d'authentification car le nœud source doit attendre toute les réponses pour choisir le chemin de certification.

1.5.3 Comparaison des protocoles

Le tableau suivant illustre la comparaison des différents modèles anarchiques :

La comparaison des différents modèles est faite par rapport aux critères de la disponibilité du service de certification et de scalabilité. Il est aussi expliqué pour chaque modèle comment les dépôts de certificats sont gérés et la manière avec laquelle les chaînes de certificats sont collectées. Ces deux critères influencent fortement la capacité des nœuds à collecter les chaînes de certificats et indirectement le taux de réussite du service de certification. On récapitule les principaux points de cette comparaison dans les deux critères suivants :

- **Le critère de disponibilité du service de certification :** on trouve au sommet, les modèles de Capkun et al. [3], Ren et al. [4], Omar et al. [6], Marjan et al. [11], qui assurent un niveau élevé de disponibilité du service de certification. En effet, dans ces modèles, chaque nœud maintient un dépôt local, qui est mis à jour périodiquement à chaque arrivée d'un nœud voisin en utilisant un protocole d'échange de certificats. De ce fait, la collection des certificats se fait localement à l'instant de l'exécution du processus d'authentification, ce qui augmente la disponibilité du service de certification. Quand un nœud client C_i nécessite la vérification de la clé publique d'un autre nœud C_j, les deux communicants fusionnent leurs dépôts de certificats et essayent de trouver une chaîne de certificats à partir du dépôt fusionné. Si une telle chaîne n'est pas

Modèles	Disponibilité	Dépôts de certificats	Découverte de chaînes de certificat	Scalabilité
Capkun et al. Ren et al. Marjan et al. Omar et al.	Elevée	Géré et mis à jour par les nœuds eux mêmes.	Le nœud client collecte directement la chaîne de certificats à partir de son dépôt local.	NON
Funabiki et al.	Moyenne	Géré par les nœuds CMNs.	le nœud client sollicite les nœuds CMNs.	NON
Kitada et al. Hisham et al. Gordon et al.	Faible	contient les certificats générés par le nœud lui-mêmes, et ceux générés pour lui.	Le nœud client diffuse la requête de découverte à tous les nœuds intermédiaires concernés par une chaîne de certificats donnée.	OUI
Mohri et al.	Faible	contient les certificats générés par les nœuds eux mêmes, et ceux générés par d'autres pour lui.	la requéte du nœud client suit la structure d'arbre couvrant.	OUI
Kambourakis et al.	Faible	contient les certificats générés par les nœuds eux mêmes, et ceux générés par d'autres pour lui.	la requéte du nœud client suit la structure d'arbre binaire.	OUI
Xia et al.	Moyenne	Contient les certificats de ses voisins à 2-saut et ceux des nœuds qu'il connaît dans le réseau.	Chaque nœud diffuse la requête de découverte à ces voisins à 1-saut.	Non
Suguna et al.	Faible	contient les certificats générés par le nœud lui-mêmes, et ceux générés pour lui. Existance de memoire cache qui stocke les certificats de la chaîne reçue.	Le nœud client diffuse la requête de découverte à tous les nœuds intermédiaires concernés par une chaîne de certificats donnée.	OUI

TABLE 1.1 – Tableau comparatif entre les protocoles de gestion de certificats.

trouvée, C_i peut solliciter d'autres nœuds (helper nodes) qui se trouvent à un ou deux sauts de son voisinage.

On trouve aussi le modèle de Funabiki et al. qui assure un niveau moyen de disponibilité du service de certification. Ceci est dû à la gestion centralisée des certificats qui sont stockés dans des nœuds spéciaux (CMNs). A l'exécution du processus d'authentification, le nœud client doit solliciter les nœuds CMNs afin de collecter une chaîne de certificats appropriée. De ce fait, la disponibilité du service de certification est fortement liée à la disponibilité des nœuds CMNs, ce qui diminue la disponibilité du service de certification. Les autres protocoles de certification réactifs restant assurent un faible niveau de disponibilité du service de certification, car dans ces modèles les chaînes de certificats sont collectées à la demande ; c'est-à-dire à l'instant même de l'exécution du processus d'authentification auprès des nœuds intermédiaires qui sont concernés par cette chaîne. De ce fait, la disponibilité du service de certification dépend fortement de la disponibilité des nœuds intermédiaires.

- **Le critère de scalabilité :** Dans tous les modèles proactifs, la scalabilité de service de certification n'est pas assurée car le nombre de certificats stocké dans chaque nœud est important. On trouve aussi le modèle de Funabiki et al. qui ce base sur un dépôt central de certificats, qui pourra être surchargé à large échelle, d'où la scalabilité ne sera pas assuré dans ce modèle. Tous les autres protocoles réactifs restant assurent la scalabilité de service de certification, car le dépôt des certificats des nœuds contient que les certificats générés par le nœud lui-même et ceux délivré pour lui.

1.6 Conclusion

Dans ce chapitre, nous avons présenté quelques généralités sur les réseaux mobiles ad hoc et les problèmes de gestion de certificats dans ces réseaux, et nous avons

étudié les différentes solutions proposées en se basant sur les modèles de certifications anarchiques. Cette catégorie de modèle est devisée en deux sous-catégories qui sont : les modèles proactifs et les modèles réactifs. Dans la dernière partie nous avons présenté une comparaison de l'ensemble des solutions étudiées.

2

Misbehavior Resistance Certificate Chain Recovery Protocol

2.1 Introduction

Afin de répondre à la problématique de collecte de certificats dans les réseaux mobiles ad hoc, nous avons proposé MRCCRP (*Misbehavior Resistance Certificate Chain Recovery Protocol*) qui est un protocole réactif. Notre proposition permet de construire et de maintenir les différents chemins de certification existants, ensuite le nœud source procède à la sélection d'un chemin selon deux critères ; à la fois le chemin le plus sûr et le plus court.

2.2 Notre solution

Dans son fonctionnement de base, ce protocole est similaire aux protocoles proposés par Kitada et al. [14] et Hisham et al. [9], les caractéristiques principales de notre protocole sont énumérées comme suit :

- Chaque utilisateur dans le réseau génère sa propre paire de clés (privée, publique).
- Le système n'a besoin d'aucune tierce partie de confiance, le service d'authentification est assuré par les nœuds eux même. Si un nœud u croit que la clé publique K_v appartient à l'utilisateur v alors il lui assigne un certificat.
- Il n'est pas nécessaire d'exécuter un protocole d'échange de certificats.
- Tous les nœuds ont un rôle similaire, et nous n'assignons pas un rôle particulier pour un ou un certain nombre de nœuds.
- Chaque nœud maintient un dépôt de certificats, qui contient les certificats qui sont délivrés pour lui par d'autres nœuds, et ceux qu'il délivre pour les autres nœuds.
- La validation de certificats est réalisée en utilisant les mécanismes de la signature numérique.
- Notre modèle se base sur le graphe de confiance G (V, E), ou V représente les sommets et E les liens. Dans ce graphe les sommets correspondent aux nœuds et les liens correspondent aux certificats. Comme il est illustré sur la figure 2.1, il existe un lien direct entre un nœud u vers un nœud v si et seulement si, il existe un certificat (Cert $u \rightarrow v$) signé par la clé privée de u qui permet de lier la clé publique de v (K_v) à l'identité du nœud v.

Le protocole **MRCCRP** peut être divisé en 3 phases : (1) La phase de recherche des chemins de certification, (2) la phase de sélection du chemin de certification et (3) la phase de la collecte des certificats.

2.2.1 Phase de recherche des chemins de certification

Le problème dans cette phase est de trouver des chemins de certification du nœud source S vers un nœud destinataire D. Pour résoudre ce problème nous avons proposé un mécanisme de collection des identificateurs du nœud destinataire D vers le nœud source S qui s'exécute selon les deux étapes suivantes :

- **Étape 1 :** Selon un protocole de routage existant, le nœud source S envoie une requête *Req* au nœud destinataire D dans laquelle il spécifie qu'il veut initier le processus d'authentification avec lui.

- **Étape 2 :** quand le nœud destinataire D reçoit la requête ***Req***, le nœud D diffuse un message de réponse ***Rep*** vers la source S en ajoutant son identificateur ID_D. Le message ***Rep*** est diffusé par D et envoyé seulement pour les nœuds qui lui font confiance, c'est-a-dire ceux qui ont déjà signé des certificats pour D. Chaque nœud intermédiaire i qui reçoit le paquet ***Rep***, rajoute son identificateur dans ce paquet, et le redirige vers l'ensemble des nœuds qui lui font confiance. Le processus continuera de cette façon jusqu'à ce que les paquets d'identificateurs (***Rep***) atteignent le nœud source S.

  À la fin de ce processus, S reçoit plusieurs paquets des identificateurs de D vers S.

2.2.2 Phase de sélection du chemin de certification

Quand le nœud source S reçoit les différents paquets des identificateurs qui correspondent aux chemins de certification possible, il choisit un selon les deux critères suivant :

1. ***Critère1 :*** Choisir le plus court chemin x, le nœud source choisit le chemin

dont le nombre d'identificateurs est minimale. Celui qui satisfait :

$$Long(x) = \min(Long(i)). \tag{2.1}$$

avec $i = [1..M]$, et M est le nombre de chemins obtenus.

2. ***Critère2*** : Choisir le chemin le plus sûr, On évalue le degré de sûreté d'un nœud selon le nombre de certificats délivrés pour authentifier sa clé publique. C'est-à-dire, si le nombre de certificats délivrés pour authentifier la clé publique d'un nœud u est supérieur au nombre de certificats délivrés pour authentifier la clé publique d'un nœud v, alors le nœud u est plus sûr que v.

   L'idée principale de notre proposition est d'écarter le chemin contenant un nœud dont le degré de sûreté est minimal, car un seul nœud malicieux peut empêcher tout le service de certification comtpé à effectuer à travers le chemin contenant ce nœud. Le chemin x choisi satisfait :

$$g(x) = \max\{h(x)/x \in [1..M]\} \tag{2.2}$$

ET

$$h(x) = \min\{d^-_{n_i}/n_i \in x\} \tag{2.3}$$

   Avec :

   i = [1..M] tel que M est nombre des chemins trouvés.

   $d^-_{n_i}$: représente le nombre d'arc entrant dans chaque sommet n de chaque chemin i, c'est-à-dire le nombre de certificats reçus par chaque nœud n de chaque chemin i.

   h(x) : fonction qui renvoie le minimum de certificats délivrés pour un nœud n appartenant au chemin x.

Pour que le nœud source ait la possibilité de combiner les deux critères cités précédemment, on utilise une fonction objective dont le but est de trouver le chemin x

pour lequel la valeur de cette fonction est maximale, en d'autres termes, il consiste à trouver le chemin x qui maximise cette fonction.

La fonction *f(x)* suivante représente la manière de combiner les deux critères précédent pour choisir la chaîne de plus court chemin et plus sûr en même temps :

$$f(x) = \frac{h(x)}{Long(x)} \tag{2.4}$$

2.2.3 Phase de la collecte des certificats

Après la sélection du chemin de certification, le nœud source S connaît le chemin de certification qu'il doit suivre pour authentifier la clé publique du nœud destinataire D.

Pour obtenir la chaîne de certificats du nœud source S au destinataire D nous suivons les étapes suivantes :

- **Étape 1 :** le nœud source S envoie une requête de certification individuelle (en unicast) pour chaque nœud concerné par la chaîne de certificats sauf pour ceux certifiés par lui-même.
- **Étape 2 :** chaque nœud intermédiaire qui reçoit la requête de certification, répond à la source S par son certificat de clé publique.

2.3 Exemple d'un scénario d'exécution de MRC-CRP

A fin d'expliquer notre protocole, nous avons appliqué les étapes de son fonctionnement sur l'exemple de graphe de confiance de la figure 2.1.

1. **La phase de recherche des chemins :**
   - **Étape1 :** pour que le nœud source S puisse initier le processus d'authentification avec D, il lui envoie la requête ***Req***. comme le montre la figure 2.1.

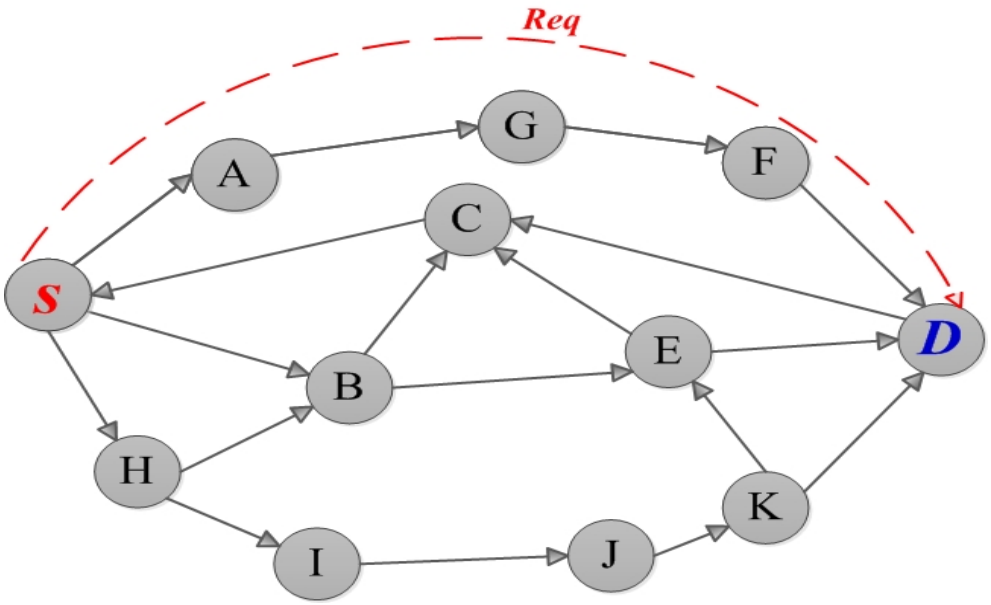


FIGURE 2.1 – Le nœud source S envoi une requête *Req* au nœud D

– É**tape2 :**la figure 2.2 illustre le message de réponse ***Rep*** diffusé par le destinataire D vers les nœuds qui lui font confiance. Chaque nœud intermédiaire qui reçoit le paquet ***Rep*** rajoute son identificateur dans ce paquet, et il le redirige vers l'ensemble des nœuds qui lui font confiance .

À la fin de ce processus, S reçoit les chaînes d'identificateurs suivantes :

Chaîne1= $\{ID_S, ID_A, ID_G, ID_F, ID_D\}$

Chaîne2= $\{ID_S, ID_B, ID_E, ID_D\}$

Chaîne3= $\{ID_S, ID_H, ID_B, ID_E, ID_D\}$

Chaîne4= $\{ID_S, ID_H, ID_I, ID_J, ID_K, ID_D\}$

Chaîne5= $\{ID_S, ID_H, ID_I, ID_J, ID_K, ID_E, ID_D\}$

D'où les chemins de certification possibles dans cet exemple sont :

chemin 1 = {S, A, G, F, D}

chemin 2 = {S, B, E, D}

chemin 3= {S, H, B, E, D}

chemin 4 = {S, H, I, J, K, D}

chemin 5 = {S, H, I, J, K, E, D}

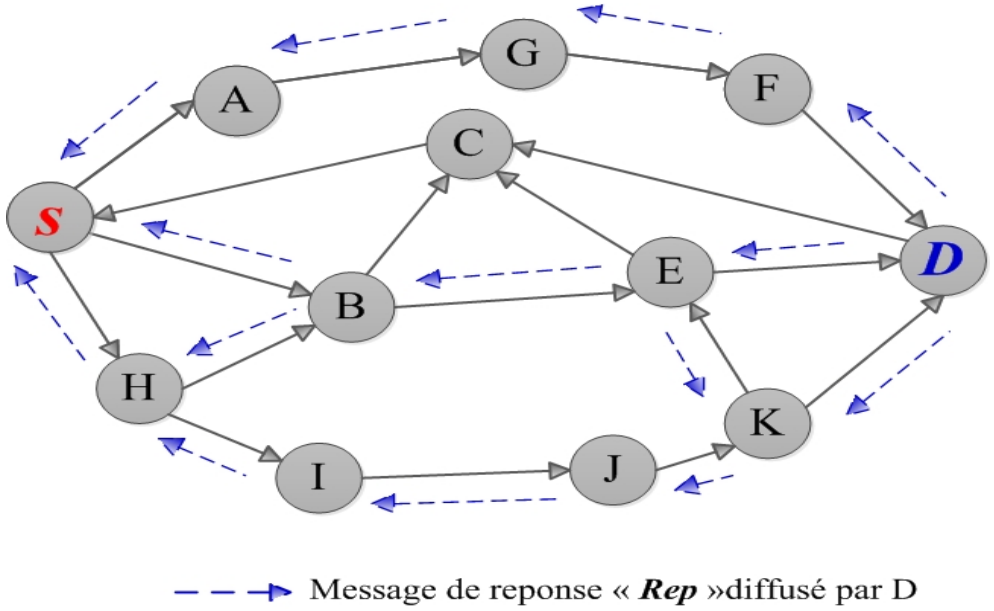


FIGURE 2.2 – Diffusion du message de réponse *Rep* par D vers S

2. **La phase de sélection du chemin de certification :** À la réception de tous les chemins possibles le nœud source S choisit un selon les deux critères cité dans cette phase. Il calcule la valeur de la fonction objective de chaque chemin ensuite il choisit le chemin qui maximise cette fonction.

   La figure 2.3 représente les différents chemins obtenus par le nœud source, et auxquels il procède au calcul de la fonction objective :

   - Calcul de la valeur de la fonction objective de chaque chemin :

   La longueur (Long(x)) de chaque chemin x : x= {1, 2, 3, 4, 5}.

   Long(1)= 4

   Long(2)= 3

   Long(3)= 4

   Long(4)= 5

   Long(5)= 6

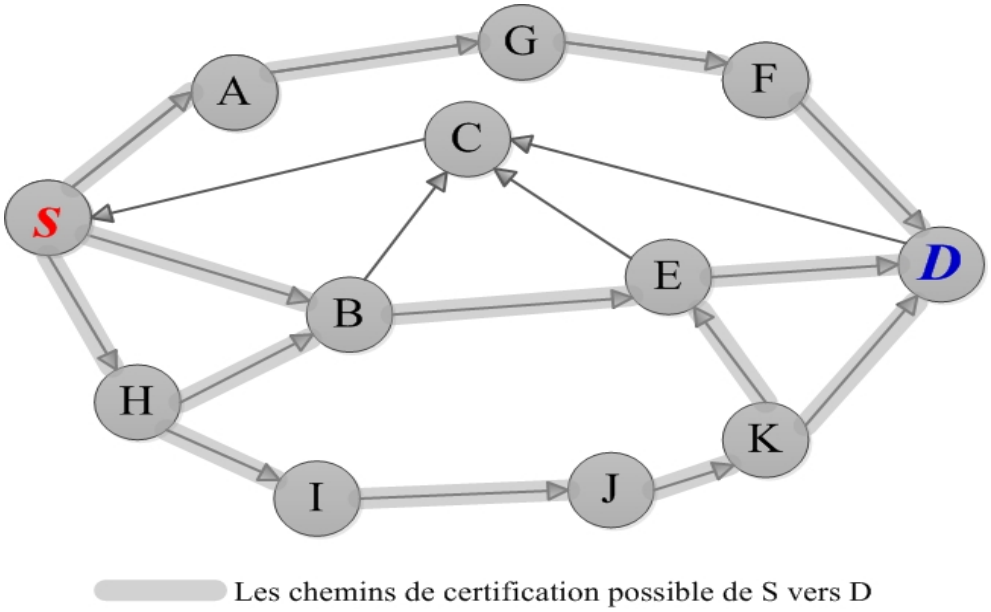


FIGURE 2.3 – Les chemins de certification possible.

Calcul du nombre de certificats délivrés pour chaque nœud à partir des différentes chaînes reçues :

À partir des chaînes des identificateurs reçues, le nœud source S peut calculer le nombre de certificats délivrés pour chaque nœud en calculant le nombre de différents prédécesseurs de chaque nœud dans les différentes chaînes reçues. On tient à rappeler qu'un nœud prédécesseur d'un nœud v dans une chaîne est le nœud qui a délivré un certificat pour le nœud v.

Dans ce qui suit nous allons calculer le nombre de certificats délivrés pour chaque nœud contenu dans les chemins :

chemin 1 = {S, A, G, F, D}

chemin 2 = {S, B, E, D}

chemin 3= {S, H, B, E, D}

chemin 4 = {S, H, I, J, K, D}

chemin 5 = {S, H, I, J, K, E, D}

Soit $d^{-}_{n_i}$: le nombre de certificats délivrés pour chaque nœud n dans chaque chaîne i.

h(x) = min$\{d^-_{n_i}$ / $n_i \in$ x$\}$

♢ **Pour le chemin x=1 :**

$d^-_{A_1}$=1

$d^-_{G_1}$=1

$d^-_{F_1}$=1

$d^-_{D_1}$=3

h(1)= min$\{d^-_{A_1}, d^-_{G_1}, d^-_{F_1}, d^-_{D_1}\}$ = 1.

♢ **Pour la chemin x=2 :**

$d^-_{B_2}$=2

$d^-_{E_2}$=2

$d^-_{D_2}$=3

h(2)= min$\{d^-_{B_2}, d^-_{E_2}, d^-_{D_2}\}$ = 2.

♢ **Pour la chemin x=3 :**

$d^-_{H_3}$=1

$d^-_{B_3}$=2

$d^-_{E_3}$=2

$d^-_{D_3}$=3

h(3)= min$\{d^-_{H_3}, d^-_{B_3}, d^-_{E_3}, d^-_{D_3}\}$ = 1.

♢ **Pour la chemin x=4 :**

$d^-_{H_4}$=1

$d^-_{I_4}$=1

$d^-_{J_4}$=1

$d^-_{K_4}$=1

$d^-_{D_4}$=3

h(4)= min$\{d^-_{H_4}, d^-_{I_4}, d^-_{J_4}, d^-_{K_4}, d^-_{D_4}\}$ = 1.

♢ **Pour la chemin 5 :**

$d^-_{H_5}$=1

$d^-_{I_5}$=1

$d^-_{J_5}$=1

$d^-_{K_5}$=1

$d^-_{E_5}$=2

$d^-_{D_5}$=3

h(5)= min$\{d^-_{H_5}, d^-_{I_5}, d^-_{J_5}, d^-_{K_5}, d^-_{E_5}, d^-_{D_5}\} = 1$.

L'idée principale à appliquer par le nœud source pour le choix de la chaîne la plus sûr est d'éviter celle contenant des nœuds ayant un nombre minimal de certificats délivrés pour eux, car un seul nœud malicieux peut empêcher tous le processus d'authentification.

Calcul de la fonction objective f(x) pour chaque chemin x :

♢ **Pour la chemin x=1 :**

f(1) = $\frac{h(1)}{long(1)}$= $\frac{1}{4}$ = 0.25

♢ **Pour la chemin x=2 :**

f(2) = $\frac{h(2)}{long(2)}$ = $\frac{2}{3}$ = 0.67

♢ **Pour la chemin x=3 :**

$f(3) = \frac{h(3)}{long(3)} = \frac{1}{4} = 0.25$

♢ **Pour la chemin x=4 :**

$f(4) = \frac{h(4)}{long(4)} = \frac{1}{5} = 0.20$

♢ **Pour la chemin x=5 :**

$f(5) = \frac{h(5)}{long(5)} = \frac{1}{6} = 0.16$

Maximiser la fonction objective f(x) :

$Max f(x) = Max(\frac{1}{4}, \frac{2}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}) = \frac{2}{3}$=f(2).

Nous remarquons que la fonction objective est maximisée par le chemin 2, donc le chemin de certification choisi par la source S sera le chemin 2 (S, B, E, D).

3. **La phase de la collecte des certificats :** À la fin de la phase de sélection du chemin de certification le nœud source S doit obtenir la chaîne de certificats du nœud source S vers le destinataire D à travers le chemin S, B, E, D selon

les étapes suivantes :

– **Étape1 :** le nœud source S envoie une requête de demande de certificat individuelle pour chaque nœud concerné par la chaîne de certificats choisie. Cette étape est illustrée dans la figure 2.4.

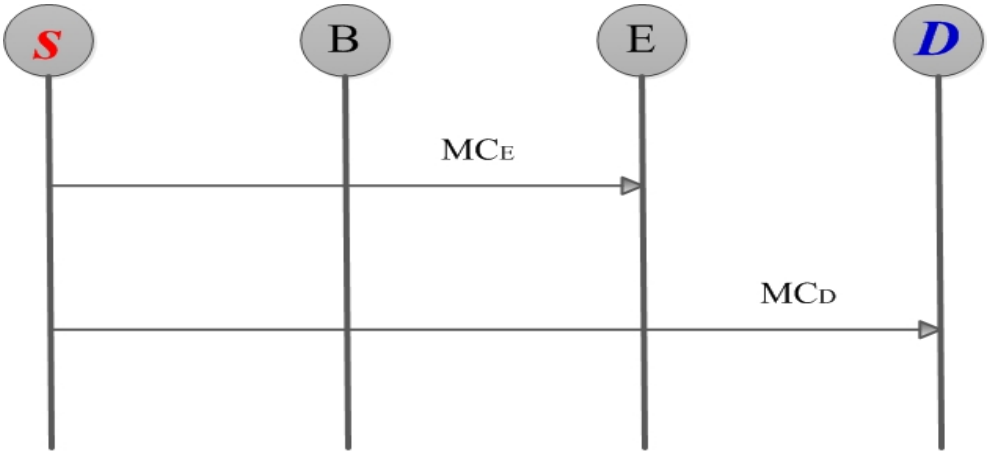


FIGURE 2.4 – Étape d'envoi de demande de certificat.

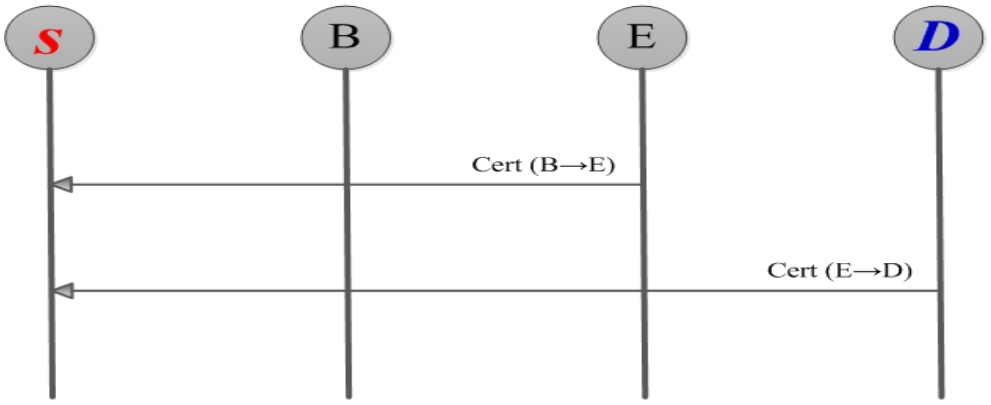
Le nœud source S envoie un message de demande de certificat MC_i pour chaque nœud i concerné par la chaîne de certificats sélectionnée. Le message MC_i contient la requête de certification (CERT-REQ), l'adresse du nœud source (addr-source) et l'adresse du nœud i (addr-i) concerné par cette demande.

MC_i = {CERT-REQ, addr-source, addr-i}.

Dans l'exemple précédent, le nœud source S envoie les requêtes de certifications seulement pour les nœuds E et D, et ne l'envoie pas pour le nœud B, puisque c'est lui-même qui a délivré un certificat pour B. d'où il récupère directement le certificat de B dans son dépôt local.

– **Étape2 :**chaque nœud intermédiaire i qui reçoit le message MC_i envoyé par le nœud source, répond à la source par son certificat de clé publique. Cette

étape est illustrée sur la figure 2.5.



Cert (B→E) : le certificat délivré par B pour E.

Cert (E→D) : le certificat délivré par E pour D.

Etape2: chaque nœud envoi son certificat de clé publique pour le nœud source S.

FIGURE 2.5 – Étape de collecte de tous les certificats nécessaire.

Dans cet exemple les nœuds E et D envoient leurs certificats de clé publique pour le nœud source S. S récupère directement le certificat de B dans son dépôt.

A la fin de cette étape le nœud source S aura la chaîne de certificats de S vers D (S→B, B→E, E→D).

2.4 Vérification de la chaîne de certificats

La vérification de certificats de clé publique dans notre modèle est effectuée à travers la vérification de la signature numérique dans chaque certificat.

Dans notre approche on suppose qu'un certificat délivré par un nœud u pour authentifier la clé publique d'un nœud v (cert u→v) contient la clé publique de u (K_u), la clé publique de v (K_v) et leurs signature numérique (SIG_u).

Cert (u→ v) = $\{K_u, K_v, SIG_u\}$.

$SIG_u = CH_{PR_u}$ [H (K_u, K_v)].

La signature numérique (SIG_u) du nœud u pour le contenu du certificat délivré par u pour v est obtenue en appliquant la fonction de hachage (H (K_u, K_v)) pour le contenu de ce certificat et chiffrer le hacher en utilisant la clé privée de u (CH_{PR_u} [H (K_u, K_v)]).

Dans la chaîne de certificats obtenue dans l'exemple de la figure 2.5 (Cert S → B, Cert B→E, Cert E→D). Pour que le nœud source S puisse valider les certificats Cert B→E et Cert E→D, il vérifie la signature numérique de B (SIG_B) en utilisant la clé publique de B contenue dans le certificat Cert B→E. Et vérifie la signature numérique de E (SIG_E) en utilisant la clé publique de E contenue dans le certificat Cert E→D.

À la fin le nœud source S fait confiance au destinataire D si toutes les signatures sont correctes.

2.5 Conclusion

Dans ce chapitre, nous avons présenté une solution pour la collecte de certificats dans les réseaux mobiles ad hoc, basée sur le graphe de confiance où aucune autorité de certification n'est employée et c'est les nœuds eux même qui assurent le service de certification. Notre modèle offre un mécanisme qui permet de résister aux comportements malicieux des nœuds du réseau, ce mécanisme est décrit dans la deuxième phase de notre proposition. Les autres avantages fournis par notre modèle sont : (1) une faible charge de stockage et transmission : car notre protocole appartient à la catégorie de modèle réactif, et dans la phase de recherche des chemins de certification les nœuds transmettent leurs identificateurs au lieu des certificats ; (2) robustesse et temps d'exécution réduit, car notre solution d'écrit des critères permettant de choisir le plus sûr et plus court chemin.

3

Modélisation analytique et résultats

3.1 Introduction

L'étude d'un système réel n'est que rarement réalisable dans un environnement opérationnel. La représentation du fonctionnement d'un système d'une manière plus ou moins précise est nécessaire pour nous permettre d'approcher son comportement. Actuellement, il existe beaucoup de formalismes de modélisation. Le choix de formalismes est déterminé d'une part par la nature du système à modéliser et d'autre part par les résultats attendus. Dans ce chapitre, nous présentons une modélisation analytique à base de chaînes de Markov, à travers laquelle nous menons une étude comparative entre notre protocole et deux protocoles concurrents.

3.2 Chaîne de Markov stochastique

Une chaîne de Markov est un modèle fournissant un outil de modélisation et d'analyse de performances du système à modéliser, il en existe deux types : les chaînes de Markov à temps discret et les chaînes de Markov à temps continu. Nous nous intéressons aux chaînes de Markov à temps discret (CMTD) qui vérifient la propriété dite sans mémoire connue sous le nom de propriété de Markov. Elle est exprimée avec :

$P(X_{n+1} = j|X_n = i) = P(X_{n+1} = j|X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) pour n \geq 0 et j, i, i_{n-1}, ..., i_0 \in E$

E représente l'ensemble des états.

La classe des processus vérifiant cette propriété est caractérisée par le fait que l'état présent du processus, c'est à dire son état à l'instant n, résume toute l'information nécessaire pour connaître son évolution future. En d'autres termes, la prévision de cette dernière ne peut être améliorée par une connaissance supplémentaire du passé du processus, c'est à dire par la connaissance de ses états aux instants $\leq n - 1$. On peut associer à une chaîne de Markov un graphe orienté appelé graphe des transitions formé de points représentant les états du processus et d'arcs correspondant aux transitions possibles, c'est à dire pour lesquelles les probabilités de transition d'un état i à un état j (p_{ij}) sont non nulles.

3.3 Modélisation

Afin d'évaluer les performances de notre solution, nous avons comparé notre protocole à celui de Kitada et al. [14] et Hisham et al. [9]. Nous avons modélisé les trois protocoles à travers une chaînes de Markov stochastique pour modéliser le processus du choix d'une chaîne de certification pour les trois protocoles. La figure 3.1 illustre le graphe de transition qui montre comment passer d'un état à un autre selon certaines probabilités. Notre modèle comporte un ensemble de M états (tel

que $M = (2 * N) + 1$ et N : le nombre de chaînes de certification possibles) avec leurs différentes transitions.

- **L'ensemble des états E sont :** (N c'est le nombre de chaînes de certification possible dans le réseau et $E = \{Chemin_1, Chemin_2, Chemin_3, ..., Chemin_N, Echec_1, Echec_2, Echec_3, ..., Echec_N$, Succès$\}$).

✓ L'état $\ll Succès \gg$ représente le succès du service de certification.

✓ L'état $\ll Chaîne_i \gg$ représente le choix d'une chaîne i.

✓ L'état $\ll Echec_i \gg$ représente l'échec de la chaîne i.

✓ L'état $\ll Echec \gg$ représente l'échec du service de certification.

- **Les probabilités de transition :**

✓ P_{SC_i} : représente la probabilité de sûrté de la chaîne i.

✓ $1 - P_{SC_i}$: représente la probabilité d'échec de la chaîne i.

- **Graphe de transitions :** la figure 3.5 illustre le modèle de Markov qui modélise le choix d'une chaîne de certification pour les trois protocoles. A la réception des chaînes de certification, le nœud source choisit la première chaîne selon la politique de sélection définie dans chaque protocole. Si cette chaîne réussit dans la vérification de certificats, le service de certification est achevé. Sinon, on passe au choix d'une autre chaîne, ainsi de suite jusqu'à l'obtention d'une chaîne qui permet la réussite du service de certification.

3.3.1 Métriques de performance

Afin d'évaluer les trois protocoles, nous avons défini deux métriques :

1. **Probabilité de succès du service de certification selon le nombre de tentatives :** elle représente la probabilité de succès du service de certification à l'itération j qui est défini comme suit :

$$P^j_{succès} = P_{SC_j} * \prod_{i=1}^{j-1}(1 - P_{SC_i}) \tag{3.1}$$

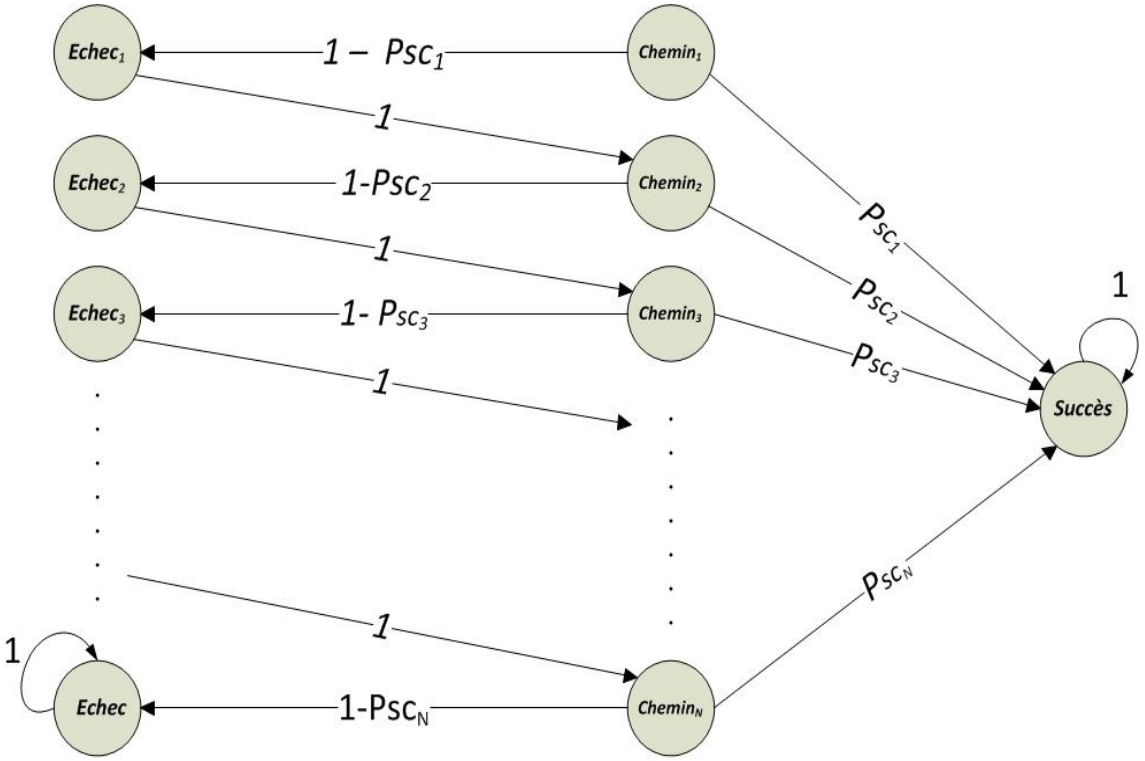


FIGURE 3.1 – Graphe de transition de la chaîne de Markov

Plus que j est petit plus que le système est robuste, et minimise la charge de calcul pour la vérification des certificats.

2. **Probabilité d'échec du service de certification :** la probabilité d'échec à l'itération j signifie que toutes les $j-1$ itérations précédentes ont subit un échec, y compris à l'itération j.

$$P_{echec}^{j} = \prod_{i=1}^{j}(1 - P_{SC_i}) \tag{3.2}$$

Nous exprimons la probabilité d'échec du service de certification, comme le produit des échecs de toutes les chaînes de certification. Cette probabilité est calculée comme suit :

$$P_{echec} = \prod_{i=1}^{N}(1 - P_{SC_i}) \tag{3.3}$$

3.4 Résultats obtenus

3.4.1 Probabilité de succès et d'échec en fonction du nombre de nœuds

Les figures 3.2 et 3.3 illustrent respectivement, la probabilité de succès et d'échec du service de certification en fonction du nombre de nœuds dans le réseau pour les trois protocoles.

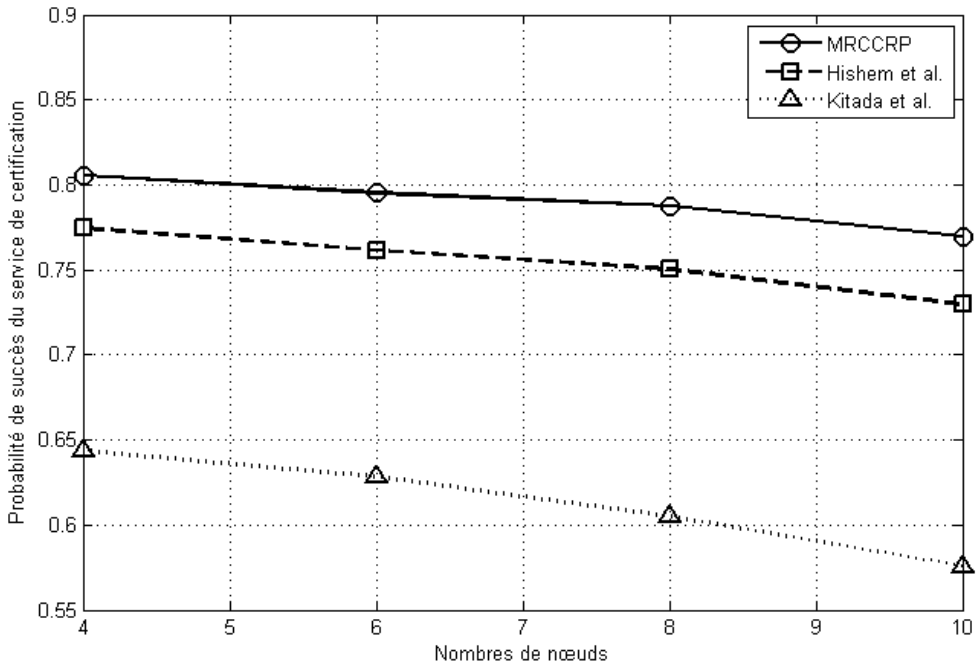


FIGURE 3.2 – Probabilité de succès en fonction du nombre de nœuds

D'après les résultats, nous constatons que même si nous augmentons le nombre de nœuds du réseau, la probabilité de succès du service de certification de notre protocole reste toujours supérieur par rapport au protocole de Hisham et al. et Kitada et al. De même pour la probabilité d'échec du service de certification de notre protocole est inférieur en la comparant avec les deux autres protocoles. Nous constatons aussi que cette probabilité diminue légèrement à cause de la longueur des chaînes impliquées par l'augmentation du nombre des nœuds, d'où la scalabilité est

assurée avec notre protocole.

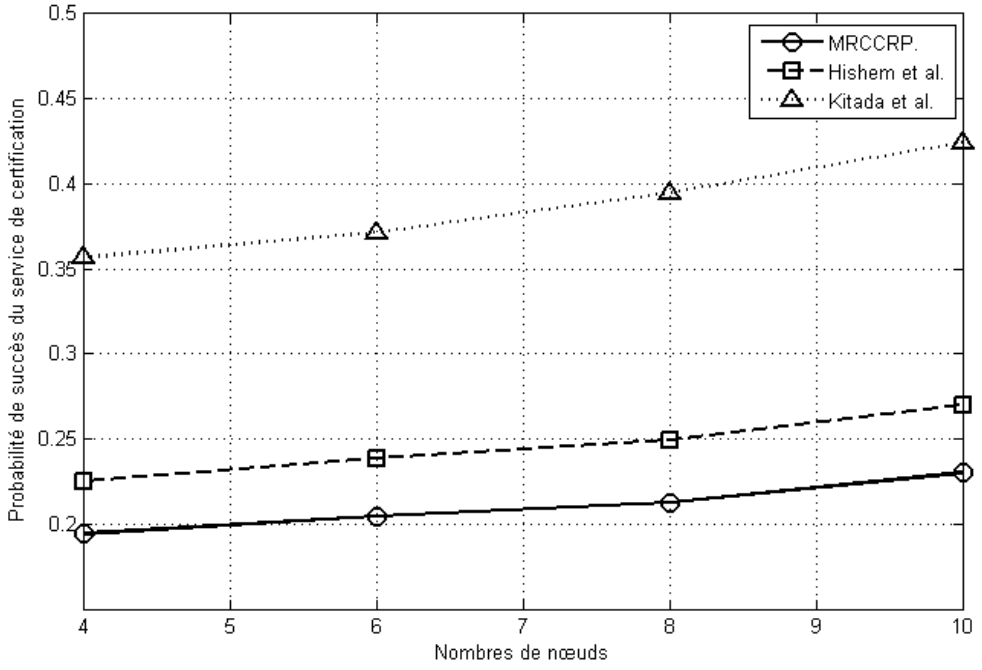


FIGURE 3.3 – Probabilité d'échec en fonction du nombre de nœuds

3.4.2 Probabilité de succès et d'échec en fonction de la densité du graphe de confiance

Les figures 3.4 et 3.5 illustrent respectivement, la probabilité de succès et d'échec du service de certification en fonction de la densité du graphe de confiance. L'axe des abscisses correspond à des valeurs qui représentent la probabilité qu'un nœud i délivre un certificat pour un nœud j. L'augmentation de cette probabilité correspond à l'augmentation de la densité du graphe de confiance.

D'après les résultats, nous constatons la probabilité de succès des trois protocoles augmentent avec l'augmentation de la densité du graphe de confiance. La probabilité de succès de notre protocole est supérieure à celles des deux autres protocoles. De même pour la probabilité d'échec du service de certification de notre protocole est inférieure en la comparant avec les deux autres protocoles. Nous remarquons égale-

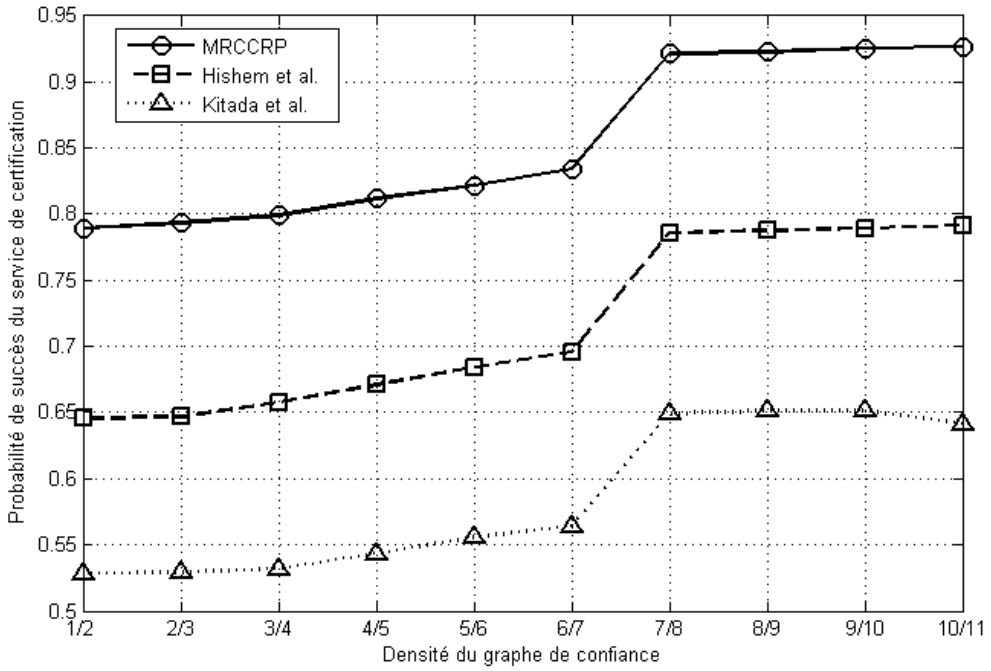


FIGURE 3.4 – Probabilités de succès en fonction de la densité du graphe de confiance

ment que la valeur de probabilité de succès dans notre protocole converge vers 1, ce qui signifie que les chaînes choisies par notre protocoles comporte des nœuds ayant un certain degré de confiance, d'où la robustesse est assurée par notre protocole.

3.4.3 Probabilité de succès et d'échec en fonction du nombre de tentatives

Les tentatives (ou itérations) correspondent aux nombres de fois pour lesquelles le nœud source répète le choix d'une chaîne de certification jusqu'à l'obtention de la chaîne qui réussit le service de certification. Les figures 3.6 et 3.7 illustrent respectivement, la probabilité de succès et d'échec du service de certification en fonction du nombre de tentatives du choix des chaînes. Dans notre protocole cette probabilité diminue avec l'augmentation de nombre de tentatives, car la fonction objective définie dans notre solution nous permet de renvoyer la chaîne la plus sûr dès la première tentative.

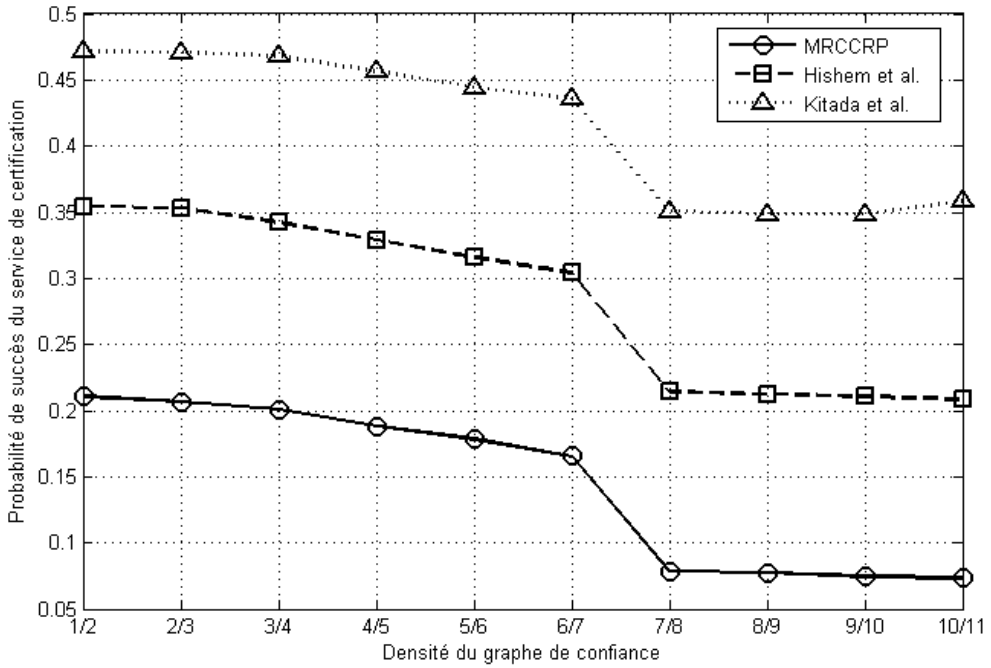


FIGURE 3.5 – Probabilités d'échec en fonction de la densité du graphe de confiance

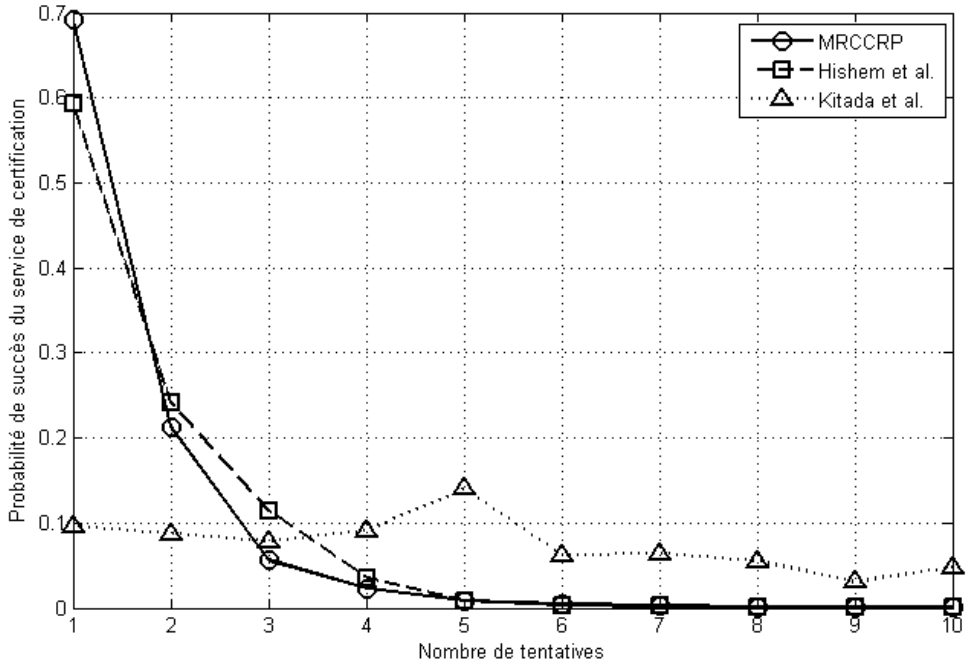


FIGURE 3.6 – Probabilité de succès en fonction du nombre de tentatives

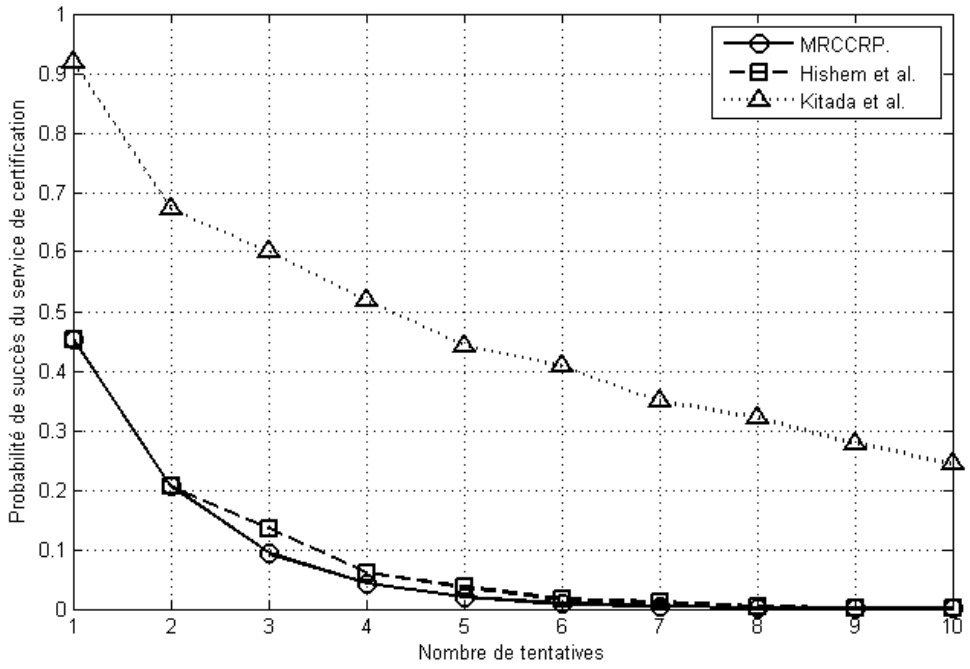


FIGURE 3.7 – Probabilité d'échec en fonction du nombre de tentatives

3.5 Conclusion

Dans ce chapitre, nous avons développé un modèle analytique à base de chaîne de Markov. Nous avons ensuite évalué les performances de notre protocole en comparaison avec ceux de Hishem et al. et Kitada et al. en termes de probabilité de succès du service de certification, d'où les résultats sont très intéressant par rapport à notre protocole.

Conclusion et Perspectives

Un réseau mobile ad hoc est une collection de de nœuds mobiles auto-organisés qui communiquent entre eux sans avoir recours à une infrastructure centrale préexistante. Les réseaux mobiles ad hoc possèdent non seulement les mêmes caractéristiques que les réseaux mobiles, mais aussi un certain nombre de caractéristiques qui leur sont propres et qui les différencient des autres à savoir, l'absence d'infrastructure, la topologie du réseau dynamique, les contraintes sur la bande passante, les contraintes d'énergie, et la sécurité limitée.

Dans ce travail, nous avons présenté un état de l'art sur les modèles de certification anarchiques dons lesquels il n'existe aucune autorité de certification et c'est les nœuds eux-mêmes qui assurent le service de certification. Dans la suite de mémoire, nous avons proposé un protocole de collecte de certificats résistant au comportement malicieux (*MRCCRP : Misbehavior Resistance Certificate Chain Recovery Protocol*). Le protocole proposé appartient à la catégorie des modèles réactifs, dont laquelle le protocole de collection de certificats est exécuté à la demande au moment où un nœud aura besoin de vérifier une chaîne de certificats donnée. Dans notre protocole nous avons supposé l'existence d'un graphe de confiance établi entre les différents utilisateurs dans un réseau. Le protocole s'exécute en trois phases :

1. La première phase consiste à la recherche des différentes chaînes de certification existantes entre une source et une destination dans le graphe de confiance. Pour

ce faire, nous avons proposé un mécanisme de collecte d'identificateurs de la chaîne à travers les nœuds intermédiaires qui se situent entre la source et la destination.

2. La deuxième phase consiste à la sélection du chemin de certification dans laquelle le nœud source choisit un chemin de certification en combinant deux critères : la chaîne la plus courte et la plus sûre.
3. La phase de collecte des certificats, c'est la dernière étape dans notre protocole dans laquelle le nœud source collecte les certificats auprès des nœuds concernés.

Les avantages fournis par notre protocole sont :

1. *Une faible charge de stockage* : car notre solution appartient à la catégorie des modèles réactifs, et les nœuds stockent seulement les certificats qu'ils délivrent et ceux délivrés pour eux.
2. *Une faible charge de transmission* : ce point est remédié par le fait que les nœuds transmettent leurs identificateurs au lieu des certificats.
3. *Une faible charge de calcul et un temps d'exécution réduit* : nous avons défini dans notre solution un mécanisme qui permet de choisir un chemin en combinant deux critères principaux qui consistent au choix de la chaîne la plus courte et plus sûre. Le plus court chemin permet de réduire la charge de calcul, et le critère de sûreté permet d'augmenter le degré de succès du service de certification à travers le premier chemin choisi d'où la réduction du temps d'exécution.
4. *La robustesse* : la solution proposée dans notre modèle permet d'éviter les chemins de certification qui comportent des nœuds qui ont un faible degré de sureté.

Comme perspectives de notre travail, nous prévoyons de converger notre modèle vers la catégorie des modèles de certification hybrides afin d'augmenter la disponibilité du service de certification. Pour ce faire, nous envisageons de développer un

mécanisme qui va permettre aux nœuds de stocker plus de certificats, par exemple un mécanisme qui permettra aux nœuds de stocker les certificats des nœuds les plus souvent rencontré dans les chaînes de certificats. Nous envisageons, également, de prévoir un temps d'arrêt du processus de recherche des chemins de certification à la réception d'un certain nombre de chemins possibles.

Bibliographie

[1] Rachedi A. *Contributions à la sécurité dans les réseaux mobiles ad Hoc*. Thése doctorat. Université d'Avignon et des Pays de Vaucluse, 2008.

[2] Drira K. *Topologie Dynamique Virtuelle Pour Le Routage Dans Les Réseaux Mobiles Ad hoc*. Projet de fin d'étude, Ecole supérieur des communications de tunisie, 2005.

[3] Capkun S, Buttyan L, Hubaux J. *Self-organized Public Key Management for Mobile Ad hoc Networks*. IEEE Transactions on Mobile Computin. 2003.

[4] Ren K, Li T, Wan Z, Bao F, Deng R, Kim K. *Highly Reliable Trust Establishment Scheme in Ad hoc Networks*. Elsevier, Computer Networks, 2004.

[5] Xia L, Steven G, Jill S. *On Demand Public Key Management for Wireless Ad Hoc Networks*. 2007.

[6] Omar M, Challal Y, Bouabdallah A. *Reliable and fully distributed trust model for mobile ad hoc networks*, Computers and Security, 2009.

[7] Funabiki S, Isohara T, Kitada Y, Takemori K, Sasase I. *Public Key Management Scheme with Certificate Management Node for Wireless Ad Hoc Networks*. In Proceedings of the International Multiconference on Computer Science and Information Technology, 2006.

[8] Kambourakis G, Konstantinou E, Douma A, Anagnostopoulos M, Fotiadis G. *Efficient Certification Path Discovery for MANET*. EURASIP Journal on Wireless

Communications and Networking, 2010.

[9] Hisham D, James I. *On Demand Self-Organized Public Key Management for Mobile Ad Hoc Networks*. Department of Electronic and Electrical Engineering, University of Strathclyde, 2010.

[10] Suguna M, Subathra M. *Establishment of Stable Certificate Chains for Authentication in Mobile Ad Hoc Networks*. Department of Computer Science and Engineering, Thiagarajar College of Engineering, 2011.

[11] Marjan K, Bahador S. *Improvement of Self-Organized Public Key Management for MANET*. Journal of American Science, 2012.

[12] Gordron R, Dawoud D. *Trust Establishment in Ad Hoc Networks by Certificate Distribution and Postponed Verification*. School of Electrical, Electronic and Computer Engineering, University of KwaZulu Natal Durban, 2011.

[13] Mohri H, Yasuda I, Takata Y, Seki H. *Certificate Chain Discovery in Web of Trust for Ad Hoc Networks*. In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), 2007.

[14] Kitada Y, Arakawa Y, Takemori K, Watanabe A, Sasase I. *On demand distributed public key management using routing information for wirelss ad hoc netwoks*. IEICE Transactions on Information and Systems, 2005.

Résumé

La gestion de certificats dans les réseaux mobiles ad hoc est un défi majeur à cause des contraintes imposées par la nature de ce dernier. Dans ce présent travail nous nous sommes intéressés aux modèles de confiances à base de graphe de confiance, où c'est les utilisateurs eux-mêmes qui jouent le rôle de l'autorité de certification en gérant les certificats d'une manière autonome. Nous avons proposé MRCCRP (Misbehavior Resistance Certificate Chain Recovery Protocol) qui a comme caractéristique de choisir une chaîne de certification à travers la maximisation d'une fonction objective, et qui permet d'obtenir une chaîne de certification robuste, résistante contre les nœuds malveillants qui délivrent des faux certificats dans le réseau.

Mots clés : Réseaux mobiles ad hoc, authentification, certificat, graphe de confiance.

Abstract

Certificate management in mobile ad hoc networks is a serious challenge, because of constraints imposed by the nature of the network. In the present work, we are interested to models based on Web of trust, where users ensure the role of the certification authority by issuing certificates. We have proposed MRCCRP (Misbehavior Resistance Certificate Chain Recovery Protocol) which has a characteristic to choose a certificate chain through the maximization of an objective function, and provides a robust chain, resistant against malicious nodes which can generate false certificates in the network.

Keywords : Mobile ad hoc networks, authentication, certificate, Web of trust.