Remerciements

Nous remercions avant tous DIEU tout puissant qui nous a donné la force, le courage et la volonté pour réaliser ce travail.

Un grand merci à nos familles surtout nos parents pour leurs encouragements et leurs suivis avec patience tout au long de notre projet.

Nous tenons à remercier vivement notre promoteur Mr OMAR Mawloud d'avoir accepté de nous guider tout au long de ce travail.

Nous remercions également nos très chers amis, camarades et tous ceux qui ont contribué de prés ou de loin à la réalisation de ce travail.

Enfin, nous tenons à remercier les membres de jury d'avoir accepté de juger notre travail.

Merci à tous.

Dédicaces

Je dédié ce modeste travail particulièrement à :

La mémoire de mon père que j'aimerai toujours, pour son soutien tout au long de ma vie, sa présence, son encouragement, ses conseils ..., enfin pour tous, que Dieu t'évacue dans son vaste paradis.

Ma chère mère pour ces nombreux sacrifices, sa présence et son encouragement, que dieu te garde.

Tiziri.

Dédicaces

Je dédié ce modeste travail à :

Toute ma famille.

Tous mes adorables amis (es).

Wissam.

Table des matières

Table des figures

Introduction générale

Depuis quelques années, Internet suscite un engouement croissant, tant dans les domaines de recherche, de l'éducation et celui des affaires. Le secteur de la communication connait une évolution considérable, cette diversité de services et d'utilisateurs est principalement due au fait qu'Internet regroupe un grand nombre de réseaux différents (filaires et sans fils).

La technologie sans fil forme un réseau dynamique, interconnectant en générale, des entités mobiles sans l'aide de toute administration ou de tout support fixe. 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 énorme. Les réseaux mobiles sans fil, peuvent être classés en deux grandes classes : les réseaux avec infrastructure et les réseaux sans infrastructure ou les réseaux ad hoc.

L'évolution dans le domaine de la communication sans fil et l'informatique mobile gagne de plus en plus de popularité, aujourd'hui ces réseaux sont utilisés dans le domaine militaire, dans les opérations de sauvetage et pour le contrôle d'environnement (chaleur, humidité), etc. La communication au sein de l'environnement mobile se base essentiellement sur la transmission radio, ce dernier engendre de nouvelles caractéristiques telles que : une fréquente déconnexion, un débit de communication, des ressources modestes, des sources d'énergie limitées et des failles de sécurité.

Les réseaux Internet sont basés sur l'hypothèse que certaines conditions sont remplies : une bande passante doit être suffisante, les nœuds doivent toujours être alimentés en énergie, les liens sont connectés en point à point de manière permanente, avec un délai de bout en bout faible entre l'émetteur et le récepteur dont la plupart des applications Internet sont basées sur le protocole TCP/IP. Cependant, ces conditions sont très difficiles à obtenir dans certains scénarios comme la liaison entre des satellites et le sol.

Les communications autres qu'Internet (mobile, satellite, interplanétaire) sont réalisées sur des liaisons indépendantes. Chacune nécessitant des caractéristiques qui lui sont propres, dans laquelle les délais de transmissions s'allongent ou dans les réseaux sans fils, lorsque les nœuds sont placés de façon très éparse. De manière générale, elles ne sont pas compatibles avec Internet et entre elles.

Les réseaux DTN sont introduits pour la communication entre plusieurs régions. Ils désignent des réseaux capables de transmettre des informations de bout en bout, même lorsque le réseau n'est pas connecté en permanence, c'est une couche au dessus des réseaux régionaux incluant l'Internet. Contrairement à Internet, ces réseaux peuvent supporter des délais plus importants et variables, des longues périodes de déconnexion, un taux d'erreur élevé et l'asymétrie de transfert de donnée. Dans un réseau DTN, les nœuds ou les passerelles doivent mettre les paquets dans des tampons et les envoyer au prochain saut le plus approprié lorsque la connexion vers ce nœud devient disponible.

Les réseaux DTN présentent toutes les limitations des réseaux sans fil, ils sont très sensible en matière de sécurité. La sécurité est un défi majeur ayant un grand impact sur le futur déploiement de ces réseaux ainsi que leurs applications. Le développement des mécanismes de sécurité instituant les relations de confiance entre les entités communicantes de même que la sécurité des transferts de données, s'avère d'une importance capitale.

Dans le cadre de notre travail, nous allons proposer un modèle pour « la gestion des clés publiques dans les réseaux DTN ». Pour cela, on le subdivise en quatre chapitres, compris entre une introduction générale et une conclusion générale avec quelques perspectives.

Dans le premier chapitre, nous présentons des généralités sur les réseaux DTN. Nous illustrons leurs caractéristiques, leurs architecture, les domaines de leurs applications, leurs fonctionnement, ainsi que les problèmes de sécurité rencontrés.

Dans le deuxième chapitre, nous nous intéressons dans la première section à la cryptographie et quelques notions de base liées à celle-ci. Dans la deuxième section, nous nous basons sur l'infrastructure à clés publiques (ICP), en expliquant les différentes phases de cycle de vie des clés et des certificats ainsi que ses composantes principales.

Dans le troisième chapitre, nous avons réalisé un modèle de gestion des clés publiques pour une architecture spécifique qui est le réseau DakNet. Ce dernier tire profil des notions traditionnelles de sécurité qu'il repose sur un modèle de certification distribué et sur les relations de confiance sociales entre les nœuds. Par la suite, nous traitons les phases de ce système de certification distribué dont chacune, nous illustrons ses étapes sous un protocole spécifique.

Dans le quatrième chapitre, nous avons introduit quelques thématiques de la théorie des graphes, pour mettre en valeur notre graphe de confiance. Pour cela, nous utilisons le principe de coloration de graphes.

Chapitre 1

Généralités sur les réseaux DTN

Introduction

Les réseaux sans fils ad hoc ont montré leurs faiblesses et la difficulté de leurs mises en place dans certains environnements à cause des contraintes géographiques et la mise en œuvre associée est très difficile dans certains environnements. Pour cela, les réseaux DTN répondent à ces problèmes et les résolvent, aujourd'hui les DTN permettent à de nombreuses régions isolées de bénéficier des avantages d'Internet et d'accéder aux multiples services offerts par celles-ci.

Ce chapitre met l'accent sur les réseaux DTN présentant un bref aperçu sur ses caractéristiques, ses exigences, par la suite, son architecture de référence, son fonctionnement ainsi qu'une vue générale sur sa sécurité.

1.1 Définition des DTN

Un réseau tolérant aux délais (DTN : Delay Tolerant Network) est un réseau interconnectant plusieurs réseaux régionaux (réseaux filaires, ad hoc, réseaux téléphoniques...) sujets à des déconnexions. Il offre une connectivité intermittente entre ces dernières en raison des difficultés rencontrées dans l'environnement à savoir : le climat, la mobilité, les pannes d'énergie,etc.

1.2 Caractéristiques des DTN

Un réseau DTN est un réseau spécifique, il présente plusieurs caractéristiques à savoir :

- **Une connectivité intermittente**
  Comme les réseaux DTN sont des réseaux sujets à des déconnexions, il n'existe pas un chemin de bout en bout entre une source et une destination, il est donc nécessaire d'implanter de nouveaux protocoles, car le protocole TCP/IP ne peut pas fonctionner.

- **Des délais de propagation longs et variables**
  Les réseaux DTN utilisent la commutation de messages "Store and Forward", ce qui

implique un temps d'attente important dans les files des nœuds intermédiaires [10], le temps de propagation est aussi important [35], par conséquent, il est nécessaire d'implanter de nouveaux protocoles, car les protocoles qui comptent sur le retour d'acquittement rapide ne peuvent pas fonctionner.

- **Un taux d'erreurs élevé**
  Des erreurs d'envoi de données peuvent se produire sur un réseau DTN à cause d'une difficulté d'acheminement des données ce qui implique leurs retransmissions saut par saut.

- **Une vitesse de transfert asymétrique**
  La vitesse de transfert de données d'un nœud vers un autre nœud, est largement différente de celle du transfert inverse.

- **Un débit de transmission faible**
  Les débits de transmission assez lents ,entre 8 et 256 kb/s.

1.3 Les exigences des DTN

Tout réseau DTN mis en œuvre vise à répondre aux exigences suivantes :

- **Coût faible**
  Tout réseau informatique essaye au maximum de réduire le coût associé aux équipements ou à l'installation du réseau, le réseau DTN doit aussi réduire le coût associé à son déploiement.

- **Fiabilité**
  La fiabilité est une condition nécessaire pour un réseau DTN. Un utilisateur doit être convaincu que le réseau fournit les services de façon fiable, même si avec des retards importants.

- **Efficacité**
  Les réseaux DTN fournissent un compromis sur la connectivité, il est encore très important de fournir des services aussi efficacement que possible.

- **Environnement hétérogène**
  Les réseaux DTN doivent soutenir tous les types de périphériques des utilisateurs capables d'utiliser l'Internet. De même, ils doivent être en mesure d'utiliser n'importe quel réseau (fixe, GSM...) disponible qui est possible d'utiliser.

- **Mobilité**
  La mobilité des utilisateurs doit aussi être prise en considération car elle est très importante.

- **Sécurité**
  La sécurité est une exigence très importante pour tous les réseaux. Les réseaux DTN en particulier sont très sensibles en matière de sécurité, ainsi de nouveaux mécanismes doivent être mis au point.

1.4 Architecture de référence des DTN

L'architecture des DTN définit un réseau de plusieurs réseaux régionaux, où chacun d'eux représente une région ou un environnement précis, dans chaque région se trouve un ensemble de nœuds qui communiquent en échangeant des messages dans une même région ou entre deux régions différentes.[21]

1.4.1 Les entités de communication

1.4.1.1 Nœuds

Un nœud DTN est une entité pour l'envoi ou la réception des messages, il peut être une source, destination ou souvent nœud intermédiaire, tel que chaque nœud est identifié par un Tuple. Durant une communication donnée, un nœud ne joue qu'un seule rôle mais lors de cette communication, il peut jouer les trois rôles suivants au cours du temps :

- **Hôte :** Pour l'envoi et/ou la réception des bundles (pas de diffusion).

- **Routeur :** Pour la diffusion des bundles au sein d'une seule région DTN. Un routeur peut jouer le rôle d'un hôte.

- **Passerelle :** Pour la diffusion des bundles entre deux ou plusieurs régions DTN, une passerelle peut jouer le rôle d'un hôte, elle se base sur la commutation de messages plutôt que sur la commutation de paquets. Cependant, elle fournit l'interopérabilité entre des protocoles spécifiques pour une région et ceux spécifiques pour une autre.

1.4.1.2 Régions

Une région DTN (cf.figure 1.1) est une zone géographique limitée, et identifiée par un nom unique et connu, tel que ce dernier est enregistré dans un annuaire [31]. Elle peut être un réseau Internet terrestre, un réseau tactique militaire, une planète,... .

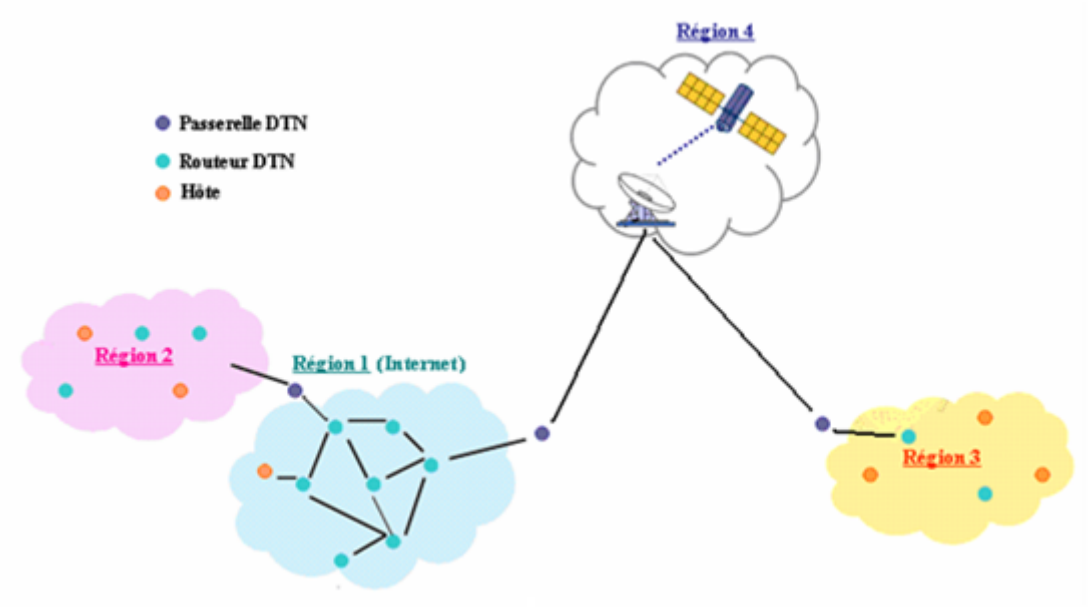


FIGURE 1.1 – Régions DTN

Les régions DTN sont caractérisées par :

1. Chaque région doit avoir un espace identifiant partagé par tous les nœuds de la région, et doit spécifier des conventions de nommage internes, afin d'être employées pour identifier les entités.

2. Chaque nœud de la région est doté d'un seul identifiant tiré de l'espace identifiant. un unique identifiant est appliqué sur le nœud destiné à recevoir des données provenant des autres nœuds DTN.

3. Si chaque membre potentiel de la région a pu atteindre les autres membres de la même région, sans passer par d'autres nœuds DTN se trouvant à l'extérieur de celle-ci, en utilisant un ou plusieurs protocoles connus au niveau de chaque nœud, alors il doit être considéré comme un membre de la région.

4. Un nœud DTN ne doit pas être atteint directement. Ceci peut demander une opération de Store and Forward et/ou de transmission par les autres nœuds de la même région.

1.4.1.3 Tuples

Un tuple désigne le nom d'un nœud DTN, il est composé de deux champs, il peut être représenté comme suit : Nom-nœud (identificateur-région, identificateur-entité).

1. **L'identificateur (nom) de la région :** chaque région DTN est identifiée par un identifiant unique, structuré d'une façon hiérarchique et connu par toutes les

régions du réseau, il est utilisé pour le routage inter-régions.

2. **L'identificateur (nom) de l'entité :** une entité peut être un hôte, un protocole, une application ou une agrégation de tous ceux là, il est utilisé pour le routage intra-région.

1.4.2 Le fonctionnement des DTN

1.4.2.1 La commutation de message «Store and Forward»

Les DTN se basent sur le concept de commutation (transport) de messages « Store and Forward », afin de résoudre les problèmes de perte de connectivité, de latence élevée et de taux d'erreurs important. [34]

Principe de Store and forward (cf.figure 1.2)
Cette technique consiste en la transmission de messages (ou parties de messages) d'une zone de stockage à une autre, le long d'un chemin qui mène à la destination.

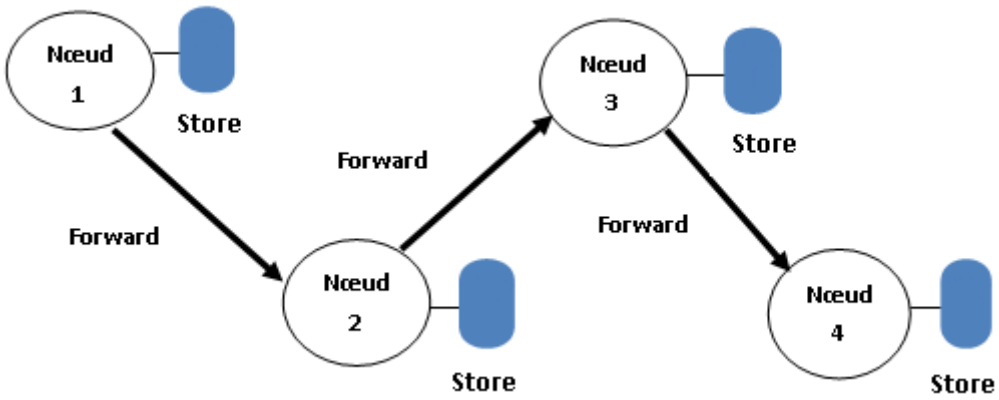


FIGURE 1.2 – Store and Forward

Chaque nœud DTN détient un espace de stockage pour stocker et conserver indéfiniment les données dans cet espace. Les nœuds DTN ont besoin d'un espace de stockage persistant très important (disque dur) pour l'une ou les raisons suivantes :

- Le lien de communication entre la source et la destination peut être indisponible pour une durée indéterminée.
- Un nœud du réseau peut émettre ou recevoir des données plus rapidement ou de manière plus fiable que les autres nœuds du réseau.
- Un message transmis doit être retransmis en cas d'erreurs sur le réseau, ou s'il n'est pas accepté pour être transféré.

Lors d'un transfert de données d'un nœud à un autre, la taille du message (bundle) doit être connue, afin de réserver l'espace adéquat et la bande passante pour le transfert.

1.4.2.2 Le protocole bundle

La technique de commutation de messages « Store and Forward » nécessite la superposition d'une nouvelle couche protocolaire « bundle protocol » au protocole existant, afin d'assurer la transmission d'un message (bundle) de bout en bout et stocker puis transférer les messages entre les nœuds du réseau.

Le « bundle protocol » permet de relier les spécificités des couches inférieures des différentes régions, ce qui assure la communication à travers des réseaux de natures différentes (ad hoc, réseaux téléphoniques, réseaux filaire... .), sa particularité est de pouvoir fonctionner au dessus de plusieurs protocoles, grâce à une sous couche appelée « couche de convergence » (CLA). Cette dernière lui permet d'assurer l'interopérabilité entre les couches inférieures des différentes régions hétérogènes [9].

1.4.2.3 Le transfert de garde

La couche bundle permet la retransmission des données perdues ou corrompues de nœud en nœud, en utilisant le transfert de garde (custody tarnsfert) qui consiste en la persistance d'un message au niveau d'un nœud appelé « gardien ».[32]

Principe du transfert de garde
Chaque transfert est effectué entre la couche bundle des deux nœuds source et destination, tel que le nœud source initie le transfert.

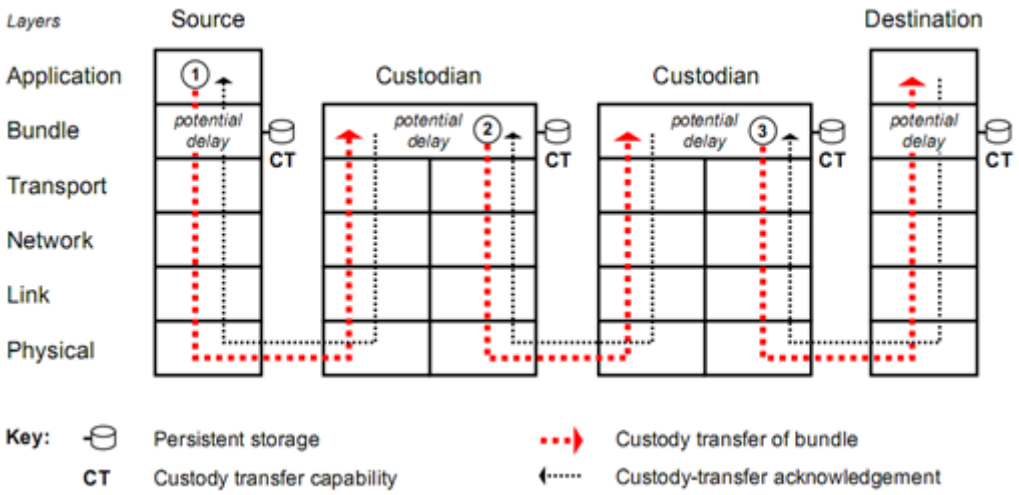


FIGURE 1.3 – Service bundle « custody tranfer » [32]

- Le gardien de la couche bundle source envoie une requête de demande de « transfert de garde » au nœud destinataire et arme une horloge (temporisateur), qui détermine le délai de retransmission de l'acquittement.
- Lors de la réception de la requête, le nœud destinataire peut accepter ou rejeter la demande.

◦ Si la requête est acceptée (demande de transfert bundle et la bonne réception), le destinataire répond par un acquittement.
◦ S'il n'ya pas d'acquittements retournés, après l'expiration du délai de garde, l'émetteur retransmis le bundle.

Le bundle reste sauvegardé jusqu'à ce que :

- Un autre nœud accepte la garde.
- Expiration du délai de retransmission du bundle.

1.5 Applications des DTN

1.5.1 KioskNet

Le réseau KioskNet (cf.figure 1.4) est doté d'une architecture de communication qui combine à la fois les moyens de transports existants, les équipements physiques et le transfert sans fil des données. Le transfert de données se base sur le concept de la commutation de messages Store and Forward, il est composé de : Kiosques, utilisateurs, ferries, passerelles (proxy) et routeurs (gardiens).[33]

• **Kiosques :**
Un Kiosque est considéré comme une plate-forme centrale dans le village, il se compose de : contrôleur de kiosque, un serveur fournissant le démarrage du réseau, un système de fichiers réseau et l'utilisateur des mécanismes de gestion et de la connectivité réseau. [33]

• **Utilisateurs :**
Les utilisateurs sont les clients de réseau, ils sont connectés à un kiosque soit par l'intermédiaire d'un terminal public connecté à Internet ou par leurs propres moyens (PDA, PC, etc.).

• **Ferries :**
Les ferries sont les moyens de transport (bus, voiture,...) utilisés par les contrôleurs de Kiosques, pour se connecter à Internet et transporter les messages, ils peuvent être vus comme des passerelles.

• **Passerelles :**
Les passerelles sont situées dans des zones où les connexions à internet à haut débit sont présentes, elles reçoivent les messages des ferries et les transmettent à la destination appropriée. Une passerelle peut jouer le rôle d'un proxy, dans le cas où une sélection des données est nécessaire avant transmission.

• **Routeurs :**
Un routeur peut être un utilisateur, un ferry ou une passerelle. Un utilisateur doit s'enregistrer auprès de l'un des routeurs, ce dernier s'appel le gardien de l'utilisateur qui permet de stocker les données de cet utilisateur, et peut également participer à l'acheminement

de ses «bundles». Un ferry est un routeur dynamique tandis qu'une passerelle est un routeur fixe.

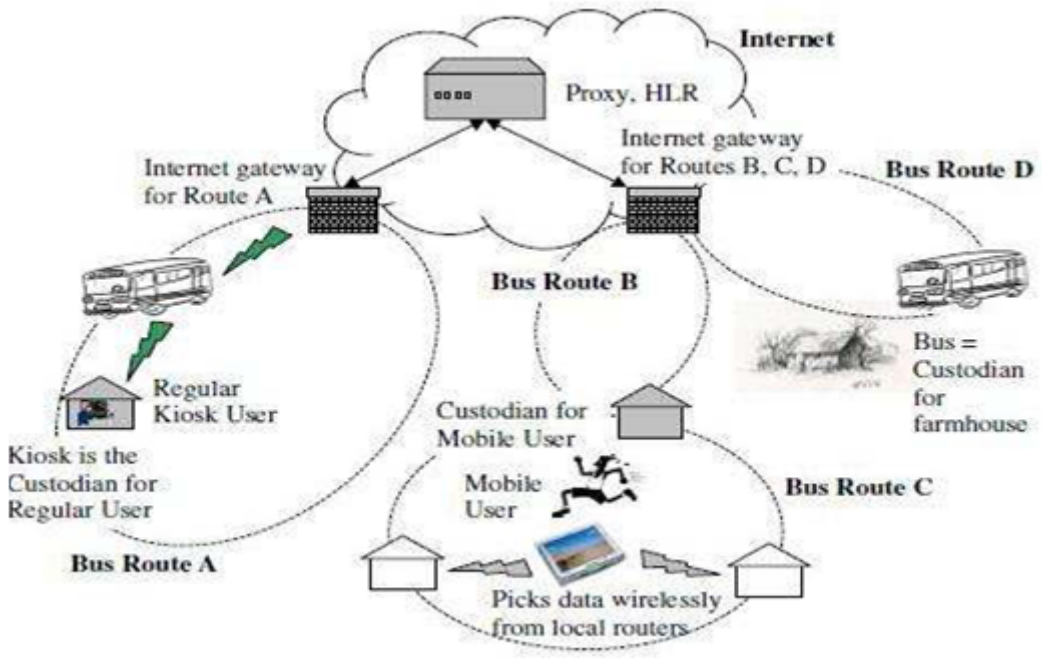


FIGURE 1.4 – KioskNet [33]

1.5.2 ZebraNet

Le projet ZebraNet est né pour répondre aux initiatives des chercheurs en biologie, qui vise à faire une étude sur des animaux, afin de comprendre leurs interactions et leurs influences chacun sur l'autre, comme la migration des animaux sauvages.

ZebraNet est fondé sur des nœuds de suivis, composés d'un système GPS miniature à faible énergie, doté d'un CPU utilisateur programmable avec un stockage de données non volatile, et des émetteurs-récepteurs radios pour communiquer avec les autres nœuds ou avec la station de base.

L'objectif est de récolter des échantillons de données GPS sur les activités des zèbres en trois minutes et enregistrer des informations détaillées sur leurs mouvements pendant trois minutes toutes les heurs pendant une année, sans l'intervention humaines, puis les fournir au centre de recherche au Kenya.

Le but de ZebraNet est d'attacher à chaque zèbre un collier équipé d'un GPS, puis rassembler les données aux niveaux de chaque collier et les transmettre à la station de base. Si cette dernière est à portée de nœud émetteur alors la transmission sera directe, sinon d'autres colliers seront pris comme étant sauts intermédiaires pour tracer un chemin vers elle.

1.5.3 Utilisation des DTN dans les communications satellitaires

Les réseaux DTN ont été initialement introduits dans les communications interplanétaires, pour cela, un scénario a été proposé sur la figure 1.5 tel qu'une station sur Mars veut transmettre les résultats de ses recherches à une autre station sur la Terre.

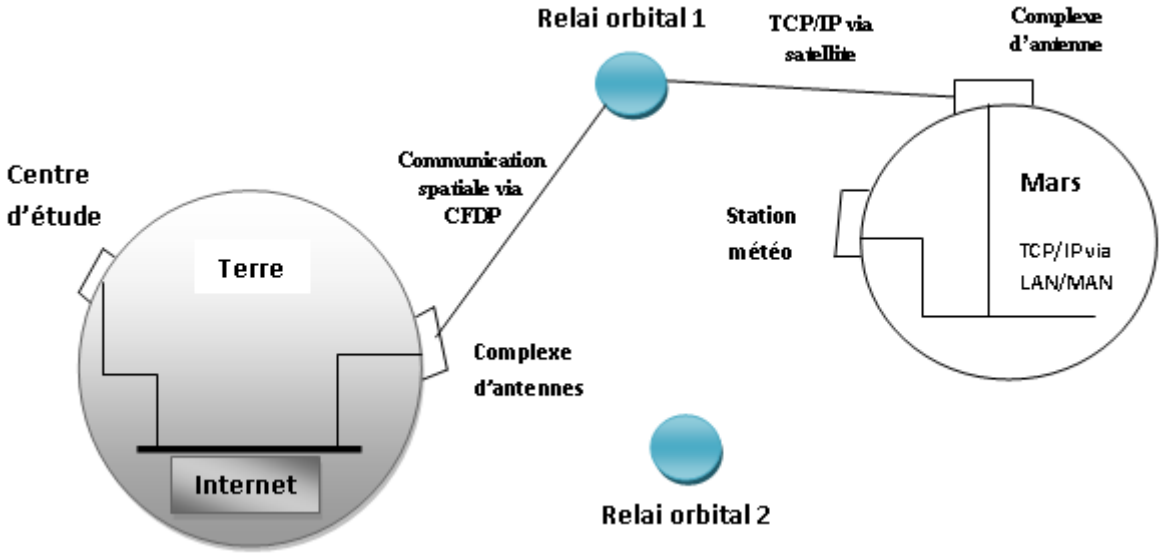


FIGURE 1.5 – Exemple de communication satellitaire

Les communications suivent l'ordre suivant :

- Un réseau local filaire ou sans fil.
- Atteindre un complexe d'antennes qui transmet le message à un relais orbital.
- Le satellite transmet le message à un complexe d'antenne situé sur la Terre.
- Les messages sont ensuite acheminés jusqu'au centre d'étude via Internet.

1.6 Le routage dans les réseaux DTN

Les nœuds dans les réseaux DTN sont sujets à des déconnexions, du fait de ses caractéristiques. Différentes solutions permettent d'assurer un routage correct au sein des réseaux DTN.

- Persistance des liens : il existe des contacts en permanence et des contacts à la demande.
- Connexions programmées : nécessitent la mise en place d'une synchronisation temporelle sur l'ensemble du DTN.
- Connexions opportunistes : en fonction des éléments présents dans l'environnement.

Les différents contacts existant entre deux nœuds DTN sont illustrés dans la figure 1.6 :

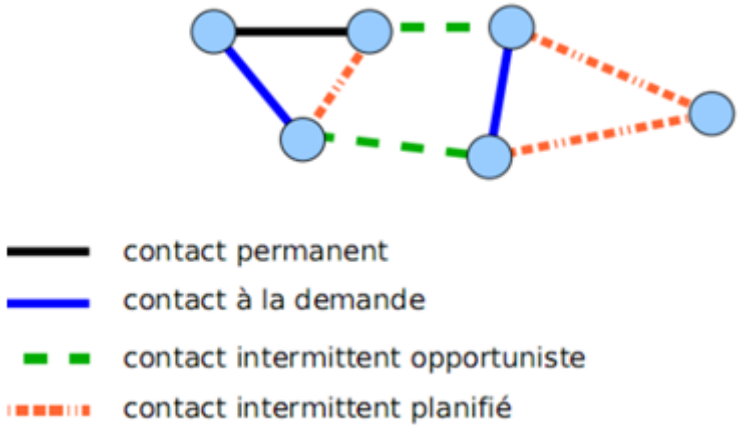


FIGURE 1.6 – Différents contacts entre deux nœuds du réseau DTN

Les nœuds DTN utilisent des vecteurs qui contiennent des éléments d'identifications du nœud ainsi que leurs localisations, ils peuvent également communiquer des informations qui permettent de programmer leurs prochaines communication.

1.7 Protocoles de routage dans les DTN

Les protocoles de routage dans les DTN consistent à augmenter le taux de délivrance de données. Les protocoles de routage peuvent être classés en différentes familles selon le moment auquel ils initient la découverte de route :

1.7.1 Routage proactif

Les protocoles répondant à ce type de routage sont capables de calculer les routes de toute la topologie d'un sous réseau connecté. Cependant, ils échouent lorsqu'ils sont appelés à déterminer un chemin vers un nœud qui n'est plus accessible. Malgré cet inconvénient, les protocoles proactifs peuvent fournir des éléments utiles aux algorithmes de routage DTN en leurs désignant l'ensemble des nœuds accessibles pour le choix du prochain saut.

1.7.2 Routage réactif

Dans ce routage, les protocoles fonctionnent seulement sur un sous réseau connecté de la topologie globale. Les routes peuvent varier avec le temps lorsqu'il s'agit de chemins prévisibles, et peuvent être pré-calculées en utilisant des connaissances sur les futures topologies.

1.7.3 Routage source

Le routage source permet de déterminer le chemin complet que doit suivre le message, depuis le nœud source. Ce chemin est codé dans le paquet du message, il est déterminé une fois et ne change pas lorsque le message traverse le réseau.

1.7.4 Routage par saut

Ce routage permet au message d'utiliser l'information sur les contacts disponibles et les files d'attentes à chaque saut.

1.7.5 Routage hiérarchique

Ce type de routage facilite la hiérarchisation du réseau en topologie abstraite et peut ne pas générer le plus court chemin, il exige que la source connaisse l'adresse hiérarchique de la destination. Son avantage est qu'il est logarithmique et n'a pas besoin d'informations sur la position des nœuds.[7]

1.7.6 Routage par la réplication

La réplication consiste à augmenter la probabilité d'avoir un message qui trouvera son chemin vers la destination, et à diminuer la durée moyenne de sa délivrance.

1.7.7 Routage par la connaissance

La connaissance permet de montrer comment obtenir des informations sur l'état du réseau, et comment une stratégie est utilisée dans le but de prendre une décision de routage.

1.8 Classification des protocoles de routage DTN

Les protocoles de routage dans les DTN utilisent deux principales propriétés : la « réplication » où plusieurs copies du message sont utilisées, et la « connaissance » qui indique comment obtenir des informations sur l'état du réseau. Les éléments de base pour classifier les protocoles de routage de réseau DTN sont les suivants :

1.8.1 L'inondation

L'inondation consiste en la délivrance de multiples copies par message pour un ensemble de nœuds appelés « relais ». Les relais stockent les messages jusqu'à se connecter avec la destination où les paquets de données seront délivrés tel que les protocoles utilisant cette stratégie ne doivent avoir aucune information sur le réseau.

1.8.2 L'expédition

L'expédition utilise l'information sur la topologie du réseau pour sélectionner le meilleur chemin, sur lequel le message est envoyé en allant d'un nœud à l'autre.

1.8.3 Les oracles de connaissance

Le problème de routage DTN a beaucoup de variables d'entrées, par exemple : les caractéristiques de la topologie dynamique et la demande de trafic. La connaissance des

variables facilite le calcul des routes optimales. Cependant, avec des connaissances partielles, la capacité de calcul des routes optimales est entravée, et les performances du routage résultant s'avèrent inférieures.

Les oracles sont des éléments utilisés pour encapsuler des connaissances particulières sur le réseau, requises par différents algorithmes. Ainsi, nous pouvons distinguer quatre types d'oracles :

1.8.3.1 Oracle de l'état des contacts

Cet oracle prend en charge les statistiques globales des contacts. Il fournit en particulier, le temps d'attente moyen jusqu'au prochain contact.

1.8.3.2 Oracle de contact

Cet oracle prend en charge tous les contacts entre deux n ?uds à un temps donné.

1.8.3.3 Oracle de file d'attente

Cet oracle donne des informations sur le taux d'occupation du buffer instantanément, à n'importe quel nœud, à n'importe quel moment, et peut être utilisé pour les routes autour des nœuds congestionnés.

1.8.3.4 Oracle de demande de trafic

Cet oracle prend en charge toute demande présente ou future du trafic. Il est capable de fournir l'ensemble des messages injectés dans le réseau à tout moment.

1.8.4 La connaissance partielle

Les algorithmes de cette catégorie calculent les chemins en utilisant une ou plusieurs de ces informations : l'état des contacts, les contacts et les files d'attente. Chaque message est routé indépendamment de la future requête, car les connaissances sur le trafic ne sont pas utilisées. Ces algorithmes sont tous basés sur l'attribution de coûts aux liens, et le calcul d'une sorte de chemin à coût minimum.

1.8.5 La connaissance complète

Certains algorithmes calculent leurs routes sans tenir compte du trafic, ce qui rend leurs performances non optimales, car ces algorithmes n'auront à se préoccuper ni du trafic, et donc ni des contraintes des buffers.

1.9 Vulnérabilités et attaques dans les DTN

Les réseaux DTN présentent plusieurs failles de sécurité qu'un attaquant peut exploiter.

1.9.1 Vulnérabilités dans les DTN

Les vulnérabilités des réseaux DTN sont semblable à celle des réseaux fixes et les réseaux ad hoc. Quiconque posséder le récepteur adéquat peut écouter ou perturber les messages échangés [35], ces vulnérabilités peuvent se résumer en :

- Les nœuds eux mêmes sont des points de vulnérabilités du réseau, car un attaquant peut compromettre un élément laissé sans surveillance.
- L'absence d'infrastructure fixe pénalise l'ensemble du réseau dans la mesure où il faut faire abstraction de toute entité centrale de gestion, pour l'accès aux ressources.
- Le support sans fil permet l'écoute du trafic.

1.9.2 Attaques

N'importe quelle action qui compromet la sécurité des données est une attaque. Dans les réseaux DTN, on trouve souvent les attaques de déni de service (DoS) et l'usurpation d'identités, on distingue deux types d'attaques :

1. **Attaques passives :** Dans ce type d'attaques, un attaquant est limité à l'écoute et l'analyse du trafic échangé, cette dernière prive le réseau de la confidentialité des messages échangés.

2. **Attaques actives :** Dans ce mode d'attaques, l'attaquant se donne les moyens d'agir sur la gestion, la configuration et l'exploitation du réseau. L'attaquant peut :
   - Injecter son propre trafic.
   - Modifier le fonctionnement d'un nœud.
   - Usurper l'identité d'un nœud.
   - Rejouer des messages.
   - Modifier des messages transitant sur le réseau.
   - Provoquer un déni de service.

1.10 Sécurité des réseaux DTN

Le domaine de la recherche sur les réseaux s'intéresse beaucoup au développement des réseaux DTN et à leurs sécurité. La sécurité des réseaux DTN est très difficile à mettre en œuvre, les mécanismes traditionnels sont inadaptés pour ces derniers, pour répondre à ces contraintes plusieurs schémas de sécurité ont été proposés.

1.10.1 Travaux existants sur la sécurité des réseaux DTN

1. **Farrell et al.**, (2009) : ils ont expliqué comment protéger les réseaux DTN, et pourquoi il est préférable (ou pas préférable) d'utiliser des mécanismes spécifiques.[22]

2. **Symington et al.**, (2010) : ils traitent la sécurité du protocole bundle, ils ont proposé des méthodes pour la protection de l'intégrité des bundles entre deux sauts.[24]

3. **Lu et al.**, (2010) : en exploitant les contacts sociaux, ils ont proposé un protocole préservé pour les DTN véhiculaires, ils ont démontré que leurs protocole peut réaliser des préservation conditionnels privées et résister à de nombreuses attaques existantes dans ces derniers.[25]

4. **Patra et al.**, (2008) : **Seth and Keshav**, (2005) ; **Kate et al.**, (2007) : ils ont suggéré l'emploi de la cryptographie basée sur l'identité (Identity Based Cryptography IBC). Les IBC permettent la création d'un chemin sécurisé de bout en bout. L'expéditeur encrypte toutes les données avec la clé publique du destinataire, et ces dernières ne peuvent être décryptées que par le destinataire, cela apporte de la confidentialité, de l'intégrité et un accès authentifié.[18]

5. **Bhutta et al.**, (2009) : ils ont donné une analyse sécurisée pour les réseaux DTN, et indiqué que la gestion des clés seule ne peut pas être suffisante pour les réseaux DTN, due a l'hétérogénéité des réseaux, comme réseaux satellitaires, réseaux sensors, etc. [25]

1.10.1.1 Travaux existants sur la gestion des clés publiques

La gestion des clés publiques est très difficile a mettre en place dans un réseau DTN, peu de recherches ont été menées sur ce sujet, cependant une approche de distribution des clés publiques a été proposée pour Pocket DTN par : Zhongtian Jia, XiaodongLin, Seng-HuaTan, LixiangLi, YixianYang (2011).[27]

Cette approche se base sur deux canaux cryptographiques :

- Un canal traditionnel sans fil : pour la transmission des clés publiques.
- Un canal manuel (conversation face à face ou téléphonique) : pour la transmission des vérifications de l'information.

Le schéma de distribution des clés proposé est décomposé en trois parties :
- L'échange des clés entre propriétaires.
- L'échange des clés publiques entre transporteurs.
- Approbation des clés publiques.

Conclusion

Les réseaux DTN sont une approche très intéressante pour communiquer dans des environnements difficiles et inaccessibles, ils ont montré leurs efficacités dans plusieurs domaines de la vie courante. Cependant, ces derniers présentent plusieurs failles de sécurité, ce qui empêchent leurs bon fonctionnement dans certains cas, il est donc nécessaire voir indispensable de mettre en avant une infrastructure pour leurs sécurisation.

Chapitre 2

Cryptographie et gestion des clés publiques

Introduction

L'informatique évolue de jour en jour, aujourd'hui, traitant une masse très importante de données, surtout avec l'apparition des réseaux. Les réseaux informatiques ont tous un trait commun, qui est le partage de données et de ressources, cependant chaque réseau est unique en raison des protocoles utilisés, des services offerts, de son emplacement physique, du milieu dans lequel il est utilisé et de sa configuration. Ces derniers sont vulnérables et peuvent être corrompus par des intrus malveillants, leurs sécurisation est indispensable et leurs exigences en termes de sécurité diffèrent d'un réseau à un autre. Dans ce chapitre, nous donnons quelques notions de base liées à la cryptographie et à l'infrastructure de gestion de clés.

2.1 Définition de la cryptographie

La *cryptographie* est une science qui utilise des mathématiques pour chiffrer et déchiffrer des données. Elle permet de stocker des informations sensibles et de les transmettre à travers des réseaux non sûrs de telle sorte qu'elles ne puissent pas être lues par personne à l'exception du destinataire convenu. La cryptographie offre des services de sécurité tels que l'authentification, la confidentialité, l'intégrité et la non-répudiation [15].

2.2 Notions de base de la cryptographie

2.2.1 Clés

Une clé est une valeur numérique codé en bits utilisée avec un algorithme cryptographique pour produire une donnée chiffrée spécifique.

2.2.2 Chiffrement / Déchiffrement

Le chiffrement est le processus cryptographique utilisant un algorithme avec une clé pour transformer une donnée en claire en une donnée chiffrée de manière à la rendre incom-

préhensible afin d'assurer le service de confidentialité.

Le déchiffrement est le processus inverse qui applique une transformation sur une donnée chiffrée de manière à le ramener dans sa forme original.

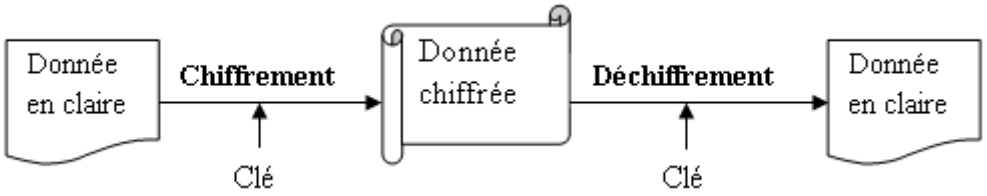


FIGURE 2.1 – Chiffrement / Déchiffrement

2.2.3 Chiffrement symétrique

La cryptographie symétrique (cf.figure 1.2) utilise une même clé pour chiffrer et pour déchiffrer des données.

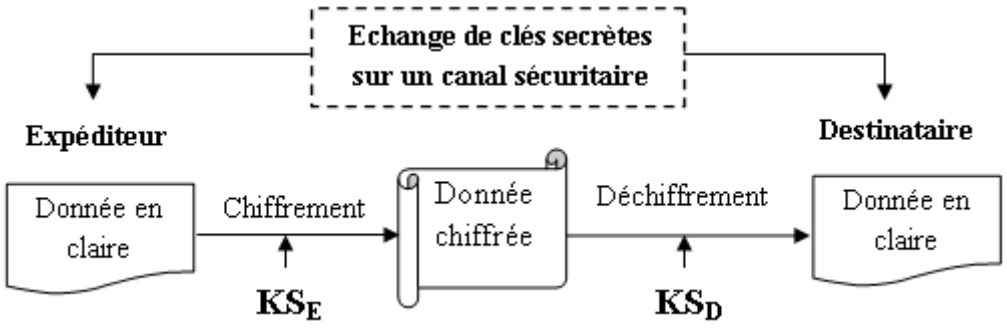


FIGURE 2.2 – Chiffrement symétrique

Il existe aujourd'hui deux grandes catégories de chiffrement à clés symétriques : le chiffrement par blocs, utilisant plusieurs algorithmes à savoir : DES (Data Encryption Standard)[2], 3DES [2] , AES (Advanced Encryption Standard) [5] et *Blowfish* [13], et le chiffrement par flot, dont l'algorithme le plus utilisé est RC4 [29].

2.2.4 Chiffrement asymétrique (à clé publique)

La cryptographie à clé publique repose sur un schéma asymétrique (cf.figure 1.3) utilisant une paire de clés : une *clé publique*, publiée dans des annuaires/serveurs de clés et accessible à tout le monde, et une *clé privée*, gardée secrètement et n'est connue que par son propriétaire.

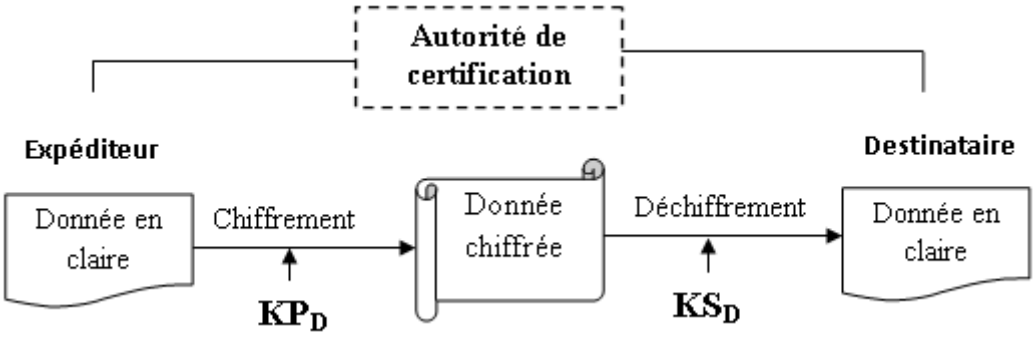


FIGURE 2.3 – Chiffrement asymétrique

L'expéditeur chiffre la donnée avec la clé publique du destinataire KP_D et l'envoie au destinataire. Ce dernier déchiffre la donnée avec sa clé secrète KS_D. Parmi les algorithmes de chiffrement asymétrique les plus répondus nous citons : Elgamal[30], RSA (Rivest Shamir et Adleman)[30], Diffie-Hellman [30] [14] et DSA (Data Structures and Algorithms) [16].

2.2.5 Fonction de hachage

Comme le processus de chiffrement/déchiffrement est long, la parade trouvée était d'utiliser une fonction de hachage (cf.figure 1.4)

FIGURE 2.4 – Hachage

La fonction de hachage calcule un haché de longueur fixe et petite, à partir d'une donnée de longueur variable et grande. Il est impossible de retrouver la donnée originale à partir du haché. Les fonctions de hachage souvent utilisées sont : MD4, MD5 et SHA-1 [17][1].

2.2.6 Signature numérique

La signature d'une donnée est calculée à l'aide de la clé privée du signataire permettant à la personne qui reçoit cette donnée de contrôler l'authenticité de son origine et de vérifier que l'information en question n'a pas être modifiée. Nous illustrons sur la figure 1.5 les étapes de passage pour la signature numérique :

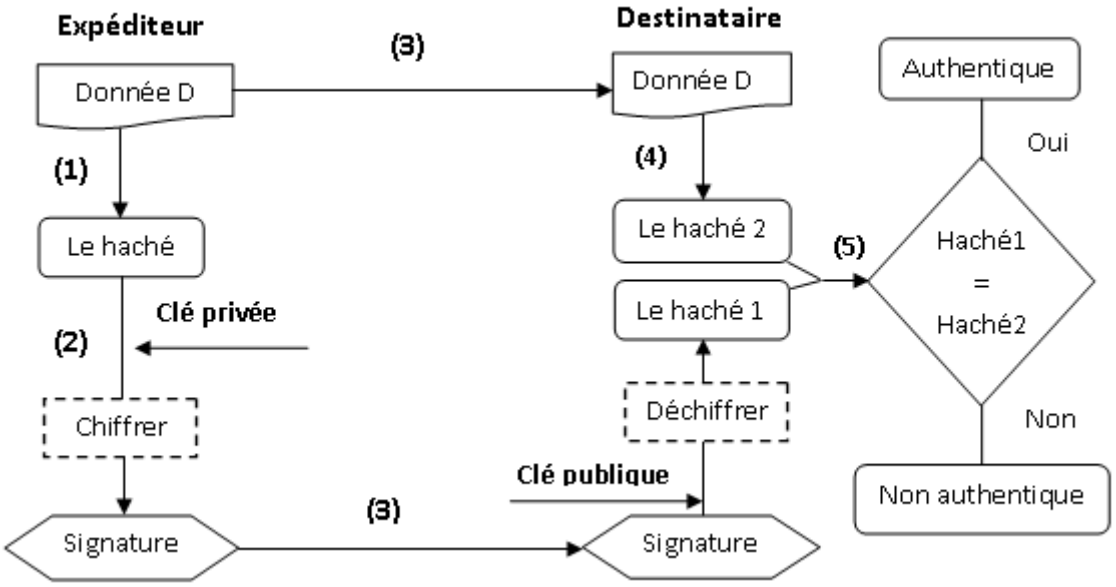


FIGURE 2.5 – Signature numérique [28]

(1) L'expéditeur calcule le haché de la donnée D en utilisant une fonction de hachage.

(2) Il chiffre le haché avec sa clé privée.

(3) Le destinataire reçoit la donnée signée par l'expéditeur, il le déchiffre avec la clé publique de ce dernier.

(4) Le destinataire calcule le haché de la donnée.

(5) Si les deux hachés sont égaux, alors l'expéditeur est authentifié.

2.2.7 Certificat à clé publique

Un certificat est un document numérique liant une clé publique à une personne, une application ou un service. Il permet de valider des clés publiques. Un certificat numérique contient généralement les informations [8] suivantes :

- La version.
- Numéro de série.
- Algorithme de chiffrement utilisé pour signer le certificat.
- Nom de l'autorité de certification émettrice.
- Date de début de validité.
- Date de fin de validité.
- Clé publique du propriétaire.
- Signature numérique de l'autorité de certification.

2.3 Infrastructure à clés publiques (ICP)

2.3.1 Définition

Une ICP est une architecture de sécurité qui a été introduite pour fournir un certain niveau de confiance pour échanger des informations sur Internet. Elle permet la gestion des clés publiques et des certificats électroniques, et d'assurer les services [19] de sécurité suivants :

1. **Authentification :** Service qui permet à un utilisateur de s'assurer qu'une entité est réellement qui elle prétend être.

2. **Intégrité :** Service qui permet à un utilisateur de s'assurer que des données n'ont pas été altérées de manière intentionnelle ou non.

3. **Confidentialité :** Service qui permet à un utilisateur de s'assurer que seule l'entité à qui sont destinées les données pourra les interpréter correctement.

4. **Non-répudiation :** Service qui permet à un utilisateur de s'assurer que les entités d'extrémité demeurent honnêtes quant à leurs actions.

Les avantages d'une ICP sont multiples, nous citons entre autres :

- La certitude de la qualité des informations envoyées et reçues par une voie électronique.
- La certitude de la source et la destination de cette information.
- L'assurance de la durée et le moment de cette information.
- La certitude de la vie privée de cette information.

2.3.2 Mise en œuvre d'une infrastructure ICP

La fonction principale d'une ICP est de permettre la distribution et l'utilisation des clés publiques et des certificats numériques. Une ICP est une fondation sur laquelle d'autres applications et d'autres composantes de sécurité du réseau sont construites. Les systèmes qui nécessitent souvent des mécanismes de sécurité basés sur l'ICP comprennent le courrier électronique, diverses applications de cartes à puce, l'échange de valeur avec l'e-commerce, banque à domicile, et les systèmes électroniques postaux.

2.3.3 Les composantes de l'ICP

L'ICP se compose en cinq entités [19] [3] qui sont les suivantes :

1. **Entité d'Extrémité (EE) :** Abonné (client) et/ou système propriétaire d'une clé publique.

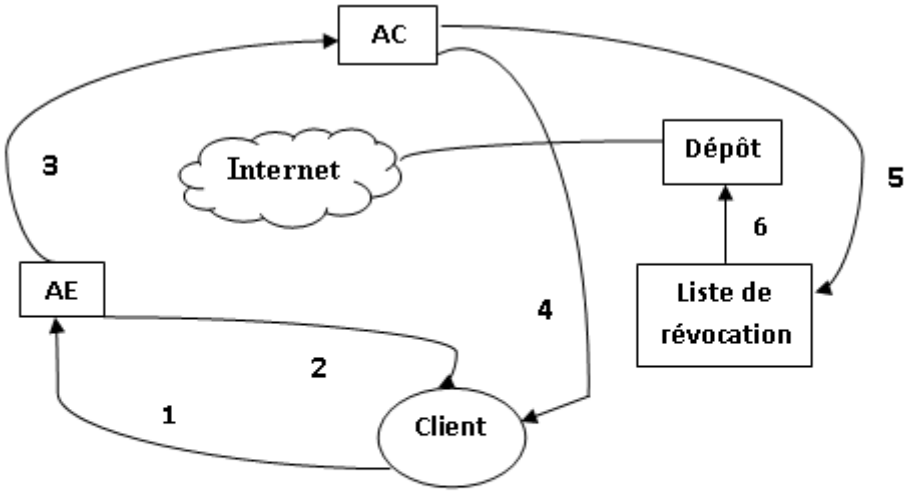
2. **Autorité de Certification (AC) :** Permet de signer les demandes de certificat et les listes de révocation.

3. **Autorité d'Enregistrement (AE) :** Permet de générer les certificats et d'effectuer des vérifications d'usage sur l'identité de l'entité de l'utilisateur final.

4. **L'émetteur de liste de révocation de certificats :** La composante qui s'acquitte de la génération des listes de révocation de certificats. Cette tâche est souvent réalisée par l'AC.

5. **Le dépôt :** Systèmes distribués où sont stockés les certificats ainsi que les listes de révocation, afin que les utilisateurs puissent les récupérer librement.

Nous illustrons sur la figure 1.6, l'organisation d'une ICP :



1. Identité du client
2. Envoie la clé privée
3. Demande de certificat (contient la clé publique)
4. Envoie du certificat signé.

FIGURE 2.6 – Les composantes et les entités de l'ICP

2.3.4 La gestion du cycle de vie des clés et des certificats

Un des principaux rôles de l'ICP est de s'acquitter de la gestion du cycle de vie des clés et des certificats. Le cycle de vie des clés et des certificats se divise en trois phases distinctes (cf.figure 1.7).

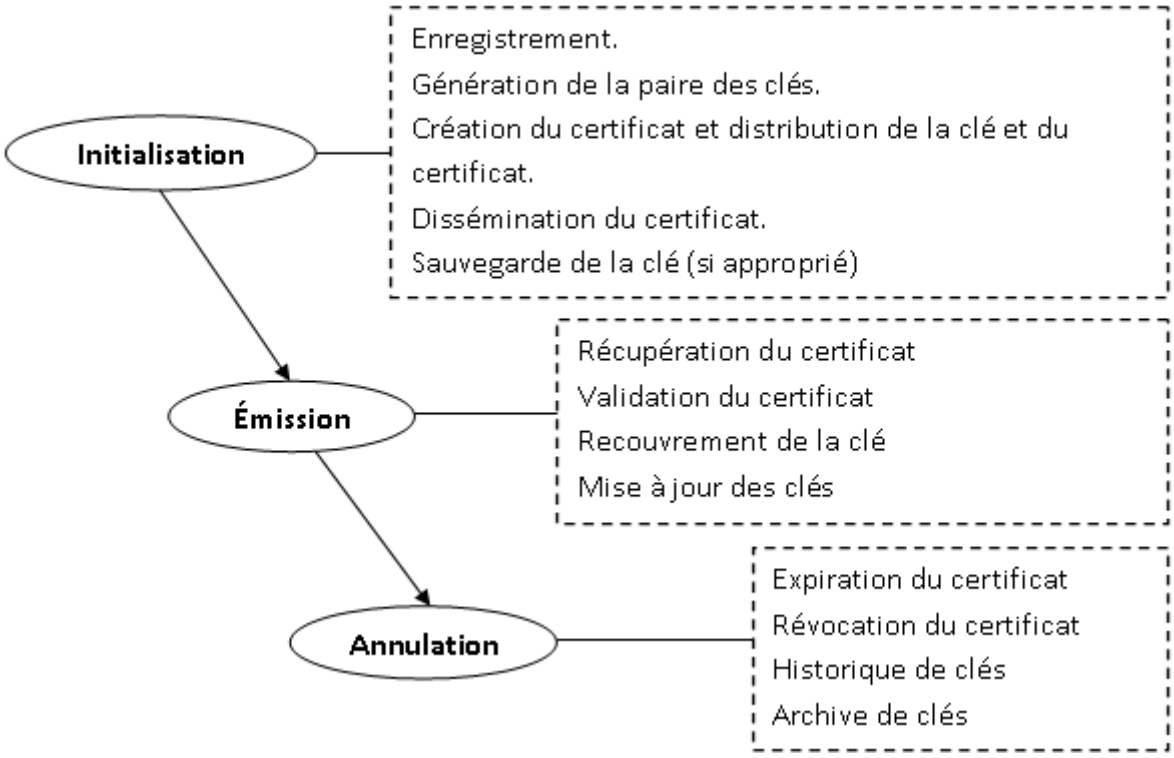
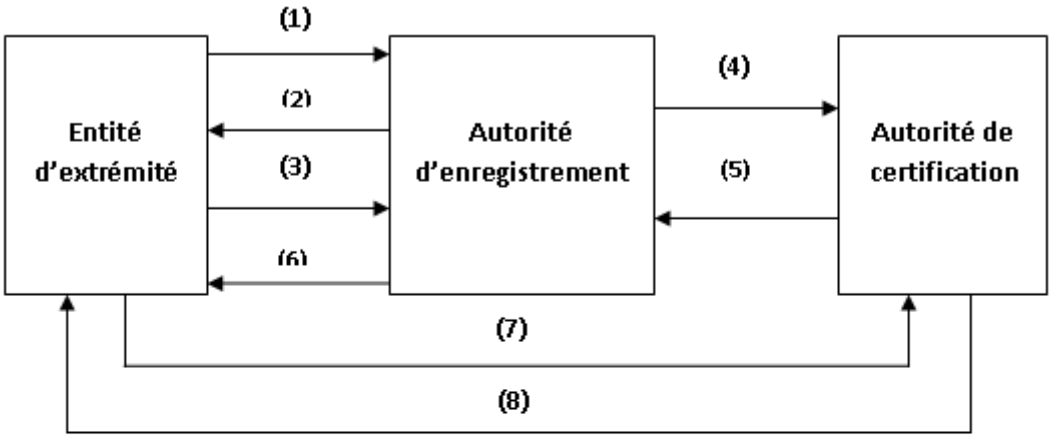


FIGURE 2.7 – Cycle de vie des clés et des certificats [8]

2.3.4.1 Phase « Initialisation »

Phase initiale du cycle de vie durant laquelle une entité effectue des démarches pour intégrer l'ICP.

1. **Enregistrement**
   C'est lors du processus d'enregistrement (cf.figure 1.8) qui est réalisé l'ensemble des vérifications quant à l'identité de l'entité d'extrémité qui désire s'abonner à l'ICP. Les vérifications à réaliser sont dictées par la politique de l'ICP.



(1) Requête de formulaire d'enregistrement.
(2) Formulaire d'enregistrement
(3) Soumission du formulaire d'enregistrement
(4) Requête d'enregistrement
(5) Réponse de l'enregistrement
(6) Résultat de l'enregistrement
(7) Requête de certificat
(8) Réponse de certificat

FIGURE 2.8 – Processus d'enregistrement auprès d'AE [12]

La requête de certificat qui est réalisée par l'entité d'extrémité nécessite la spécification d'un mot de passe. Ce dernier aura préalablement été communiqué à l'entité d'extrémité par l'AE suite à l'enregistrement.
L'enregistrement peut aussi s'effectuer en contactant directement l'AC. L'utilisation ce AE est un choix d'implantation qui se justifie en fonction des besoins et de la configuration de l'ICP.

2. **Génération de la paire de clés**
   C'est par ce processus qu'est générée la paire de clés d'une entité d'extrémité. Cette génération peut s'effectuer tant du côté de l'entité d'extrémité elle même, que du côté de l'AC.

3. **Création du certificat et sa distribution**
   C'est par ce processus que l'entité d'extrémité reçoit son certificat ayant été signé par l'AC, ainsi que la clé privée associée lorsque cette dernière fut générée par l'AC.

4. **Dissémination du certificat**
   C'est par ce processus que le certificat d'une entité d'extrémité est publié afin que les utilisateurs de l'ICP puissent l'utiliser. De façon générale, cette opération consiste à placer le certificat dans un dépôt.

5. **Sauvegarde de la clé**
   C'est par ce processus que la clé privée d'une entité d'extrémité peut être confiée à une AC, afin de constituer une copie de sauvegarde. Cette technique offre à une entité d'extrémité la possibilité de récupérer sa clé privée dans le cas où celle-ci deviendrait inaccessible (perte ou corruption).

2.3.4.2 Phase « Émission »

Phase durant laquelle le certificat d'une entité a été émis et il est considéré comme étant valide.

1. **Récupération du certificat**
   C'est par ce processus qu'un utilisateur de l'ICP peut obtenir le certificat d'une entité d'extrémité afin de chiffrer des données destinées à cette entité, ou de valider une signature ayant été apposée par celle-ci.

2. **Validation du certificat**
   C'est par ce processus qu'un utilisateur de l'ICP s'assure de la validité du certificat d'une entité d'extrémité. Le processus de validation implique un certain nombre de vérifications telles que :

   - Construction et la validation du chemin de certification.
   - Vérification de l'intégrité du certificat.
   - Vérification de l'intervalle de validité du certificat.
   - Vérification du statut du certificat (révoqué ou non) à partir d'une liste de révocation.
   - Vérification du respect des politiques d'utilisation du certificat.

3. **Recouvrement de clés**
   C'est par ce processus qu'une entité d'extrémité peut récupérer sa clé privée après que celle-ci ait été perdue ou endommagée.

4. **Mise à jour de clés**
   C'est par ce processus qu'une entité d'extrémité peut procéder à l'obtention de nouvelles clés avant que celles-ci n'arrivent à échéance. Lors de cette opération, l'AC procède à la génération d'une nouvelle paire de clés, ainsi que d'un nouveau certificat pour cette entité.

2.3.4.3 Phase « Annulation »

Phase finale du cycle de vie durant laquelle le certificat d'une entité devient invalide.

1. **Expiration du certificat**
   C'est par ce processus que le certificat d'une entité d'extrémité arrive à échéance. Si l'on ne pose aucune action, le certificat de l'entité ne sera tout simplement plus

valide lorsque celui-ci atteindra la fin de sa période de validité.

2. **Révocation du certificat**
   C'est par ce processus que le certificat d'une entité d'extrémité peut être annulé avant qu'il arrive à échéance (expiration naturelle).

3. **Historique de clés**
   C'est par ce processus que la clé d'une entité d'extrémité est conservée afin de permettre de déchiffrer des données après la période de validité du certificat contenant la clé publique correspondante.

4. **Archive de clés**
   C'est par ce processus que les clés de chiffrement de données et de vérification de signatures d'une entité d'extrémité sont conservées afin de permettre à des tiers de confiance de procéder ultérieurement à certaines vérifications.

Conclusion

Dans ce chapitre, nous avons présenté les notions de base liées à la cryptographie à clé publique et à l'ICP. Le cryptographie est utilisé pour le chiffrement, déchiffrement et signature. Le chiffrement des informations garantit la confidentialité en empêchant la divulgation involontaire, et la signature des messages authentifie l'expéditeur et assure que la donnée n'a pas été modifiée depuis qu'elle a été envoyée, ainsi qu'une ICP gère les clés et les certificats pour les personnes, les programmes et les systèmes.

Dans le chapitre suivant, nous allons proposer un système de certification distribué basé sur le chiffrement asymétrique et un certificat à clé publique.

Chapitre

3 Stratégie de gestion des clés publiques proposée

Introduction

Les réseaux DTN sont par nature très sensibles aux problèmes de sécurité, les mécanismes traditionnels ne peuvent pas être étendus à des réseaux où les nœuds sont sujets à des déconnexions pour de longues périodes, et où les communications de bout en bout sont inexistantes.

La sécurité des réseaux DTN demande donc la mise en œuvre de nouveaux mécanismes de sécurité. Pour cela, dans ce chapitre, nous nous montré une stratégie de gestion des clés publiques pour une architecture spécifique du réseau DTN.

3.1 Architecture ciblée du réseau DTN

Les réseaux DTN sont utilisés dans plusieurs domaines d'applications de la vie quotidienne : en astronomie, en biologie et pour communiquer même dans les endroits isolés. Notre proposition de gestion des clés publiques se restreint à une architecture spécifique des réseaux DTN qui est le réseau DakNet.

3.1.1 Réseau DakNet

Le réseau DakNet dont le nom dérive du mot indien « poste », a été monté par un groupe de chercheurs de l'institut Indienne de technologie à Kanpur et de MLA (Media Lab Asia).

DakNet combine les moyens physiques de transports (bus, moto, vélo, camion)et le transfert sans fil de données, afin d'étendre la connectivité Internet qui constitue le lien ou le hub central, comme les cybercafés ou les bureaux de postes. Son architecture est dotée d'une infrastructure de communication connectée à un ensemble de sous-réseaux sans infrastructure, chacun de ces derniers couvre une région géographiquement isolée à travers des passerelles de communication pour la distribution de la connectivité numérique aux villageois isolés qui en sont dépourvu pour des contraintes géographiques ou économiques.

3.1.2 Exemple d'application

Un modèle de communication est illustré dans la figure 3.1 comme un exemple de cas d'application. Dans chaque région isolée il y a un certain nombre d'utilisateurs, qui ont besoin de recevoir/transférer des données de/vers le réseau avec infrastructure. Cette communication est assurée à travers les bus urbains de chaque région, qui se déplacent régulièrement à partir de la région vers la partie avec infrastructure.

Chaque bus est doté d'un équipement sans fil, qu'il lui permet d'ouvrir une session de transfert avec les utilisateurs qui se trouvent dans sa portée de communication lors de son passage. Les données des utilisateurs récoltées seront transmises à son arrivée à la partie du réseau avec infrastructure. Inversement, lors de son retour il rapporte les données destinées aux utilisateurs de la région.

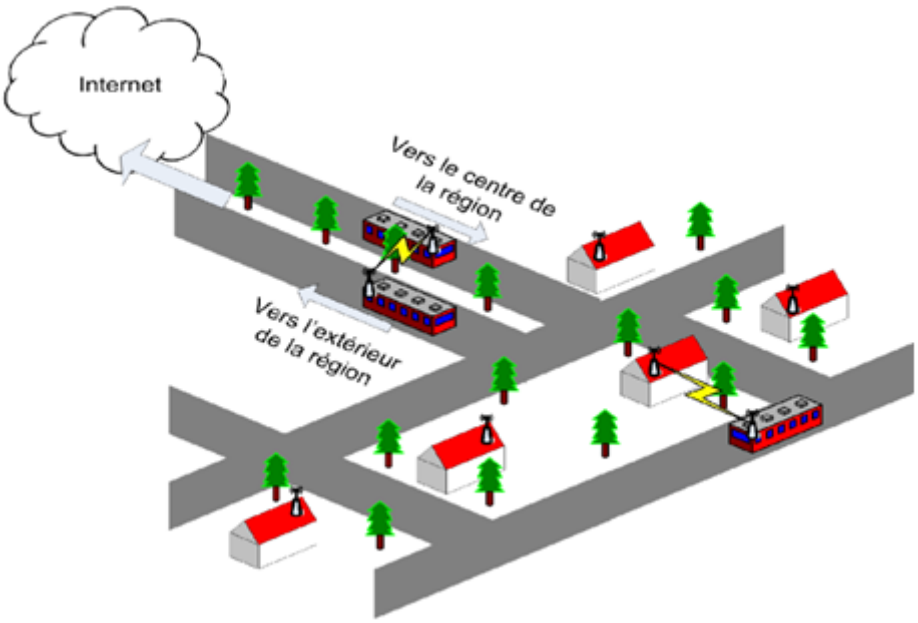


FIGURE 3.1 – L'architecture ciblée du réseau DTN

Nous considérons aussi qu'il est possible qu'un bus réplique les données détenues vers les bus qui le croise sur sa trajectoire. Cette opération est importante dans le sens où le premier arrivé s'en charge de les délivrer au destinataire, ce qui améliore relativement la fiabilité des communications. Également, la réplication des données minimise le risque de perte, dans le cas ou un nœud transporteur peut tomber en panne. Même s'il y'a qu'un seul bus qui traverse le village une fois par jour, c'est suffisant pour fournir quotidiennement des services d'informations, malgré le fait que les nœuds transporteurs ne fournissent pas la transmission de données en temps réel, mais une quantité de données importante peut se déplacer à la fois.

Ce modèle de communication peut être utilisé dans d'autres cas d'applications. Par exemple, dans le domaine militaire (figure 3.2), les données recueillies sur les champs de bataille (l'emplacement de l'ennemi, la nouvelle stratégie mise en place en cas de

changement de stratégie...) peuvent être échangées entre la zone de bataille et le centre de commandement à travers des avions militaires. L'objectif est d'améliorer les communications entre unités sur les champs de bataille, et échanger plus d'informations, visant à améliorer leurs défense. Dans ce cas il ne s'agit pas de mettre en place un réseau tolérant aux délais mais plus aux perturbations (plus particulièrement le brouillage, et les contraintes géographiques et les contraintes climatiques aussi) bien que ces dernières entrainent à leur tour des délais importants et des retards considérable.

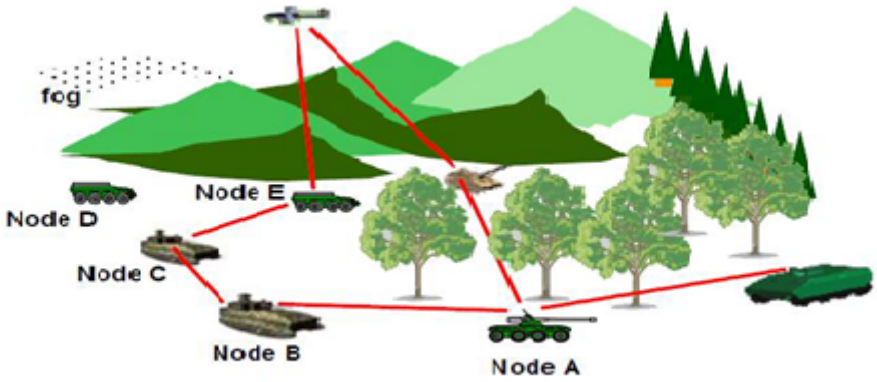


FIGURE 3.2 – Application des DTN dans le domaine militaire

Les nœuds A, B, C et E collettent des informations sur les champs de bataille et les envois (par des dispositifs sans fil) pour l'avion militaire qui sert de relit entre deux régions séparées par des perturbations (montagnes), ce dernier envoie les informations collectées au centre de commandement, si une réponse favorable est attendue, elle sera retransmise à l'avion, puis l'avion la distribue aux nœuds du réseau.

3.2 Stratégie proposée

3.2.1 Principe de la stratégie

Notre système repose sur un modèle de certification distribué basé sur les relations de confiance sociales entre les nœuds constituant le réseau, où chaque nœud (client ou transporteur) joue le rôle d'une autorité de certification pour délivrer et signer des certificats à clés publiques pour les autres nœuds. La stratégie traite les points suivants :

1. **Mise en œuvre de la confiance entre les nœuds :** selon les relations sociales entre les nœuds où en construisant une chaine de confiance.
2. **Délivrance des certificats :** chaque nœud délivre à un nœud dont il fait confiance un certificat à clé publique signé avec sa propre clé privée.
3. **Authentification lors d'une session de transfert :** ceci en échangeant les certificats à clé publique entre les nœuds communicants.

4. **Mise à jour de la confiance des noeuds transporteurs**.
5. **Révocation des certificats :** annulation d'un certificat.
6. **L'échange sécurisé de messages**.

3.2.2 Choix de la stratégie

Dans notre approche nous avons opté pour un système de certification distribué basé sur la confiance entre les nœuds à un saut. Le maintien d'une autorité de certification centrale pour délivrer des certificats demeure inapplicable dans notre approche de gestion des clés publiques. Elle a comme inconvénients principales :

- La non disponibilité des services : à cause de la mobilité des nœuds et le taux important des erreurs.
- Consommation importante d'énergie : les équipements sans fil.
- Insécurité : une autorité centrale est sujette à de nombreuses attaques, qui peuvent toucher à l'intégrité des donnés contenues dans le système et la disponibilité des services.
- La surcharge : à cause des requêtes massives de certification.

Le choix d'une autorité de certification distribuée permet de relever ces inconvénients, elle s'adapte aussi avec les caractéristiques des réseaux DTN, à savoir la mobilité des nœuds, leurs disconnectivité et les longs délais de réponse.

3.2.3 paramètres initiaux

notre stratégie met en avant les paramètres suivants, afin d'assurer le bon déroulement de ces fonctionnalités :

1. Chaque nœud génère une paire de clés.
2. Chaque nœud dispose d'une liste des clés publiques valides .
3. Chaque nœud dispose d'une liste des clés publiques révoquées (LRC).
4. On suppose que les clés publiques sont échangées manuellement entre les différents nœuds constituant le réseau.
5. On suppose qu'il existe une relation sociale entre les nœuds dans le but d'établir des relations de confiance.
6. Il n'existe pas une autorité de certification centrale, les certificats numériques sont créés, sauvegardés et échangés par les nœuds eux-mêmes d'une manière distribuée.

3.2.4 Confiance

La confiance est un mécanisme de coordination des échanges permettant d'unir les différents acteurs du réseau, elle aide à éliminer les nœuds malveillants du réseau. L'établissement de ces relations donne de meilleurs résultats de performance et une bonne sécurité.

3.2.4.1 Degré de confiance

Dans un réseau DTN, le degré de confiance est déterminé par le degré de réputation d'un nœud. La réputation est l'espérance portée dans la réalisation d'un objectif fictif. Il existe deux types de nœuds selon leurs réputations :

Bons nœuds : Ils ont une bonne réputation car ils coopèrent régulièrement et réussissent leurs tâches correspondantes, à savoir le bon acheminement des données vers la destination.

Mauvais nœuds : Ils ont une mauvaise réputation car ils adoptent un comportement égoïste. Un nœud transporteur malicieux peut perturber le fonctionnement du transfert des données des utilisateurs en les récoltant sans les faire suivre aux destinataires et sans les répliquer vers les autres nœuds transporteurs.

La valeur de réputation d'un nœud est calculée en utilisant les relations sociales. De tels systèmes analysent le réseau social qui représente les relations existantes dans chaque communauté dans le but de tirer des conclusions sur les niveaux de confiance à accorder aux autres nœuds, Ils reposent sur des mécanismes de réputation et d'honnêteté.

3.2.4.2 Approche sociale de la confiance

La confiance sociale porte sur les relations sociales entre les individus (on ne peut parler de confiance que par rapport à une personne), ces relations peuvent être familiales ou amicales dans le sens de la connaissance préalable des deux cotés. Dans le cadre des connaissances sociales, l'échange de données peut être plus sûr car :

Si A et B se connaissent mutuellement alors :
- A et B se font confiance (échangistes).
- A fait confiance en la qualité de ce qu'il peut recevoir de B et réciproquement.
- A fait confiance au procédés qui permet d'échanger avec Y (moyen d'échange) et réciproquement. [36][12]

Dans notre cas, on parle de nœuds transporteurs, car ils sont les responsables des transmissions des données des clients, ainsi si un nœud transporteur réussit la transmission des données (même en acheminant les messages entre plusieurs nœuds transporteurs) alors, son degré de confiance devient plus important.

3.2.4.3 Modèle de confiance

Notre modèle de confiance met en avant les relations réciproques de confiance entre les nœuds, il est clair que la construction des relations de confiance entre deux entités autonome en l'absence d'une autorité (tiers) centrale est un enjeu très complexe, c'est pour cette raison qu'on a pris en considération les relations sociales entre les nœuds.

Ce modèle est la brique de base sur laquelle notre stratégie est bâtie, il constitue un réseau dont chaque nœud est relié à un autre par un lien de confiance, chaque nœud est

considéré comme une autorité de certification, qui délivre des certificats pour certifier la clé publique d'un autre nœud, et qui servira pour l'authentification lors de l'ouverture d'une session de transfert.

3.2.4.4 Description du modèle

Les nœuds du réseau sont classés en deux catégories :

Les nœuds clients : se sont des nœuds fixes représentent les utilisateurs de la région pour demander des services sur internet (qui nécessite une connexion).

Les nœuds transporteurs : se sont des nœuds mobiles représentent les moyens de transports pour assurer l'opération de transfert (transport) des données des utilisateurs (les nœuds clients) de/vers la partie du réseau avec infrastructure (internet).

D'un coté, Chaque nœud transporteur ne peut ouvrir une session de transfert qu'avec soit un nœud client ou un nœud transporteur, à travers un seul saut.
D'un autre côté, chaque nœud transporteur va choisir uniquement ceux envers lesquels il fait confiance pour transférer leurs données.

Ceci permet d'élaborer un graphe de confiance particulier (figure 3.3), qui relie d'une manière réciproque chaque nœud client et chaque nœud transporteur dans le réseau.

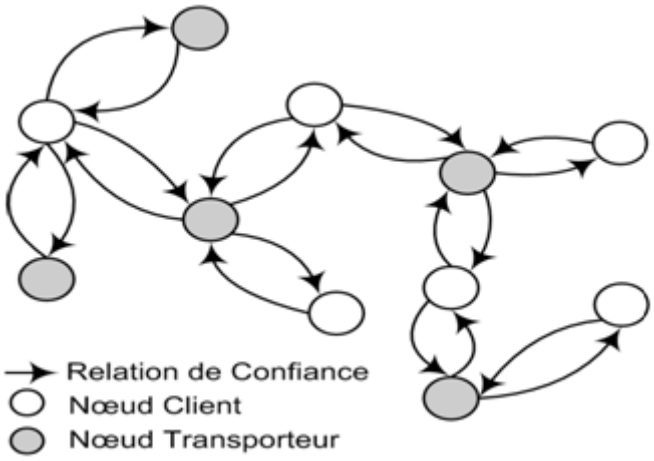


FIGURE 3.3 – Le graphe de confiance établi

3.2.4.5 Mise en œuvre de la confiance entre les nœuds

Pour ouvrir une session de transfert entre un nœud client et un nœud transporteur, il est nécessaire qu'il y ait une relation de confiance réciproque (réputation) entre les deux nœuds.

Pour mettre en œuvre cette confiance mutuelle, chaque nœud délivre à l'autre un certificat à clés publiques afin de pouvoir s'authentifier et sécuriser les sessions de transferts

qui seront établies ultérieurement.

Lorsqu'un nœud client et un nœud transporteur veulent communiquer, ils doivent s'échanger leurs listes de certificats à clés publiques, comme ils peuvent essayer d'établir une chaine de confiance (figure 3.4) entre eux, selon les relations de confiance transitives.

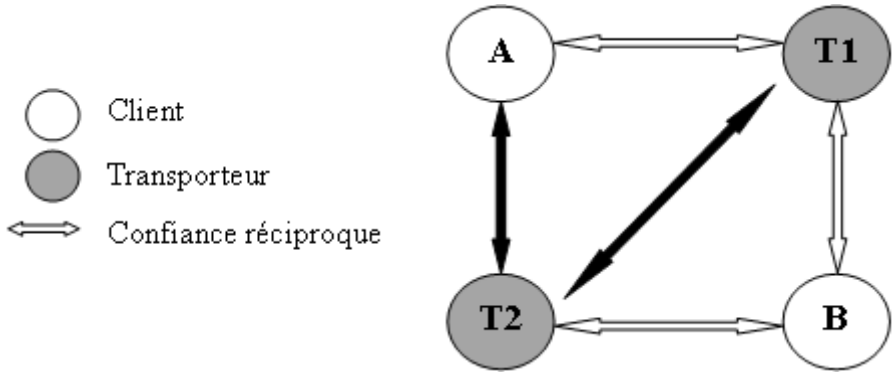


FIGURE 3.4 – Chaine de confiance

– Le transporteur T1 fait confiance aux clients A et B, il dispose de leurs certificats à clé publique.

– Le transporteur T2 fait confiance au client B, il dispose de son certificat à clé publique.

– Transitivement, le transporteur T1 fait confiance au transporteur T2, ce qui conduit le client A à faire confiance au transporteur T2.

Une chaine de confiance est construite entre le client A et le transporteur T2, aussi entre le transporteur T1 et le transporteur T2.

Dans le cas d'un transfert de données entre deux nœuds transporteurs, chacun d'eux doit prouver à l'autre qu'il est certifié par le nœud dont ses données font l'objet du transfert.

3.2.5 Traitement des clés

3.2.5.1 Génération des clés

Chaque nœud génère une paire de clés (publique et privé), la clé privée reste secrète et connue que par son propriétaire, elle servira pour la signature des certificats numériques d'autres nœuds et le déchiffrement des messages. En revanche la clé publique est supposée connue par tous les nœuds dont il fait confiance, elle est utilisée pour le déchiffrement des certificats numériques et le chiffrement des messages.

Une bonne clé est une chaine aléatoire de bits générée par un processus automatique, dans le but à assurer un haut niveau de sécurité, si un nœud utilise un processus cryptographiquement faible pour engendrer des clés, alors tout son système est faible.

3.2.5.2 Longévité des clés

Une clé générée par un nœud doit expirer automatiquement au bout d'une période prédéfinie, à cause des raisons suivantes :

- Plus la clé est utilisée longtemps, elle peut être détecté par n'importe qui (par l'attaque exhaustive).
- Plus la clé est utilisée longtemps, elle peut être compromise.
- Si la clé est compromise, elle peut être perdue.
- Il est plus facile en générale d'effectuer une cryptanalyse, quand on a plus de texte chiffrés avec une même clé [33].

3.2.5.3 Stockage des paires de clés

Les clés générées sont stockées dans un endroit sûr où un attaquant ne peut pas les introduire. Chaque nœud dispose d'un trousseau de clés pour sauvegarder l'ensemble des clés valides et un autre trousseau pour sauvegarder l'ensemble des clés révoquées.

3.2.5.4 Duplication des clés

Chaque nœud doit dupliquer ces clés pour éviter les accidents de tout genre. La duplication des clés est recommandée dans un environnement sensible aux défaillances consistant en la création d'un double pour qu'il servira dans le cas de perte. Par exemple, une clé sauvegardée dans un disque dur peut être perdu si ce dernier tombe en panne, ainsi tout les messages chiffrés avec cette clé seront perdus (indéchiffrables) donc, il ne y'aura aucune chance pour les récupérer.

3.2.5.5 Échange des clés publiques

L'échange des clés publiques se fait manuellement (présence physique est obligatoire) entre un nœud client/transporteur et un nœud transporteur, en se basant sur les relations de confiances sociales entre ces derniers, l'échange peut se faire aussi par téléphone où la reconnaissance de la voix est un très bon moyen pour l'identification des nœuds communicants.

3.2.6 Délivrance des certificats

La décision de délivrance des certificats est basée sur le rapport social entre les nœuds du réseau. Si un nœud client (respectivement transporteur) croit qu'un nœud transporteur (respectivement client) est digne de confiance dans le sens où il va assurer correctement le transfert de ses données, alors le nœud client peut lui délivrer un certificat à clé publique signé avec sa propre clé privée.

Les certificats sont délivrés selon une période de validité décidée par le nœud émetteur. Quand un certificat expire et son émetteur croit que le nœud est toujours digne de confiance, l'émetteur peut lui délivrer un nouveau certificat avec une nouvelle période de validité.

Les protocoles de délivrance des certificats :

1. **Entre un client et un transporteur :** l'échange des certificats suit la procédure suivante :

- Le nœud client génère une paire de clé : clé privée SK_c, et une clé publique PK_c.
- Le nœud transporteur génère une paire de clé : clé privée SK_t, et une clé publique PK_t.
- $CERT_{client}$: Certificat à clé publique du client.
- $CERT_{transporteur}$: Certificat à clé publique du transporteur.

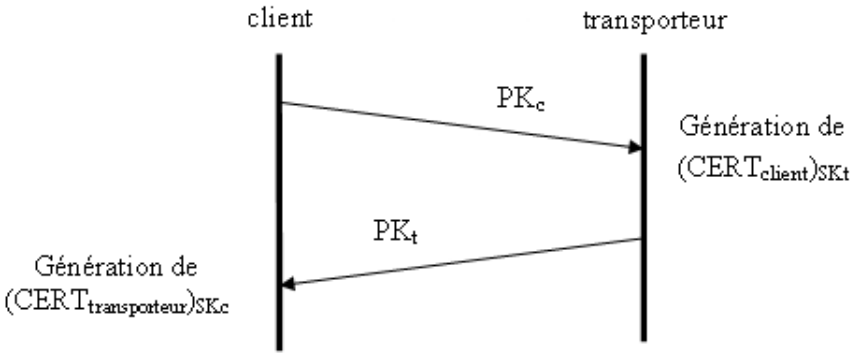


FIGURE 3.5 – Délivrance des certificats client / transporteur

2. **Entre transporteur et transporteur :** l'échange des certificats suit la procédure suivante :

- Le nœud transporteur T1 génère une paire de clé : clé privée SK_{t1}, et une clé publique PK_{t1}.
- Le nœud transporteur T2 génère une paire de clé : clé privée SK_{t2}, et une clé publique PK_{t2}.
- $CERT_{t1}$: Certificat à clé publique du transporteur T1.
- $CERT_{t2}$: Certificat à clé publique du transporteur T2.

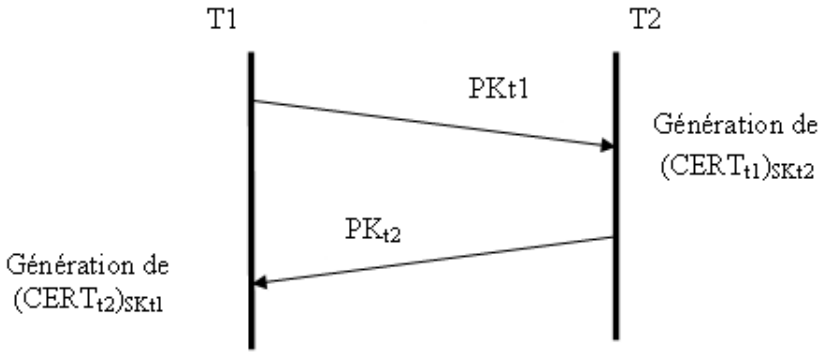


FIGURE 3.6 – Délivrance des certificats transporteur / transporteur

3.2.6.1 Validité des certificats

Chaque nœud client/transporteur doit contrôler la validité d'un certificat avant de le signer, ce certificat est considéré valide s'il appartient réellement à son propriétaire supposé et si sa période de validité n'a pas été expirée.

Le contrôle de validité des certificats est crucial, car on doit à chaque instant être en mesure d'établir l'authenticité d'un certificat donné. Chaque nœud détient dans son trousseau de clé publique la liste des clés publiques valides des autres nœuds (figure 3.7). Un nœud peut valider le certificat d'un autre nœud si et seulement s'il a entièrement confiance en lui.

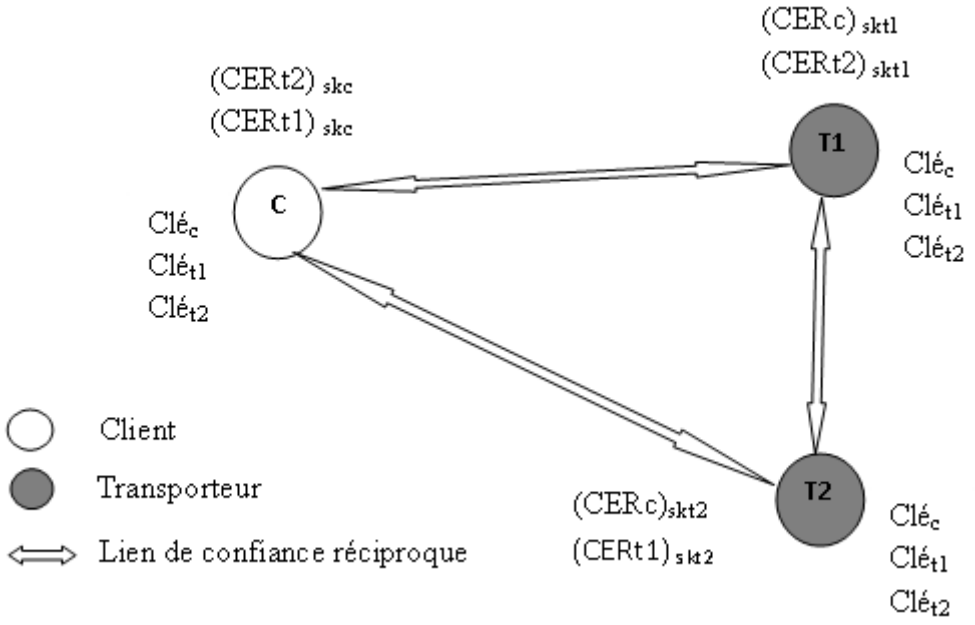


FIGURE 3.7 – Validité des certificats

Chaque nœud a confiance en les autres nœuds, chaque nœud délivre à l'autre nœud un certificat qui certifie sa clé publique, avec une durée de validité. chaque nœuds détient dans son trousseau de clés publiques la liste des clés publiques valides.

3.2.7 Authentification lors d'une session de transfert

L'authentification est un point extrêmement difficile, notamment pour appréhender les mouvements des nœuds communicants et respecter les délais de propagations des messages. Un bon protocole doit prendre en considération les erreurs de transmission des données, la mobilité des nœuds, la topologie arbitraire du réseau et les délais de réponse, car dans un système d'authentification centralisé, les nœuds s'authentifient auprès d'une autorité de certification centrale, cette dernière peut être un point de vulnérabilité pour plusieurs attaquants, comme elle peut être surchargée à un moment donné par les requêtes d'authentifications.

Les sessions de transferts de données sont ouvertes seulement à travers un seul saut de communication, c'est-à-dire que la communication sera directe entre le nœud client et le nœud transporteur sans l'aide de nœuds intermédiaires. Ceci est nécessaire dans le sens où le processus d'authentification doit s'achever instantanément pour pouvoir commencer le transfert. Par contre, si on adopte une communication à multi-sauts, le processus d'authentification va être perturbé par le délai de réponse qui est une caractéristique particulière du réseau DTN. Également, avec cette façon, l'authentification sera binaire seulement entre le nœud client et le ntransporteur. Par ailleurs, adoptant une session de transfert à multi-sauts complique davantage le processus d'authentification qui va devoir être établi dans ce cas de bout en bout avec un délai de réponse imprévisible.

3.2.7.1 Authentification entre un nœud client et un nœud transporteur

- Le nœud client dispose d'une paire de clé : clé privée SK_c , et une clé publique PK_c.
- Le nœud transporteur dispose d'une paire de clé : clé privée SK_t , et une clé publique PK_t.
- Le nœud client génère un certificat à clé publique CER_{tr} pour le nœud transporteur signé par sa clé privé (SK_c).
- Le nœud transporteur génère un certificat à clé publique CER_{c1} pour le nœud client signé par sa clé privé (SK_t).

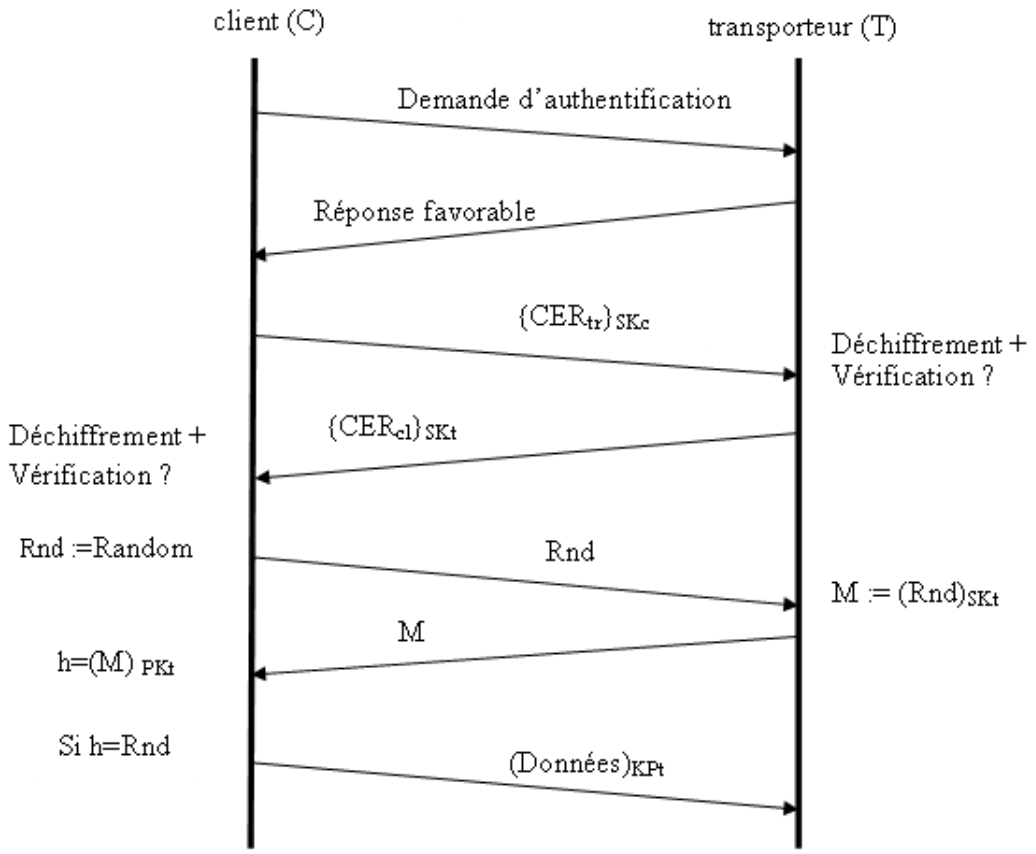


FIGURE 3.8 – Authentification entre un client et un transporteur

Initialement, le nœud client demande une authentification au nœud transporteur, ce dernier répond par un avis favorable, donc ils peuvent s'échanger leurs certificats qui doivent être signés antérieurement par son interlocuteur même.

Lors de la réception du certificat signé, chaque nœud le déchiffre en utilisant la clé publique de l'autre nœud, si les certificats sont corrects par rapport à la validité des signatures et à la période de validité alors le nœud client chiffre ses données avec la clé publique du nœud transporteur avant de les envoyer pour les protéger contre l'écoute clandestine.

Le nœud client génère un nombre aléatoire Rnd, et met le nœud transporteur en défi pour le signer avec la clé privé du transporteur. Le transporteur construit puis envoie un message M contenant un Rnd signé au nœud client. Â la réception de M, le client déchiffre M avec la clé publique du transporteur, si elle trouve le nombre Rnd alors le nœud transporteur est bien celui supposé être sinon le nœud transporteur est un nœud malveillant.

3.2.7.2 Authentification entre deux nœuds transporteurs

◦ Le nœud transporteur T1 dispose d'un certificat de croisement $CCROI_{t1}$.
◦ Le nœud transporteur T2 dispose d'un certificat de croisement $CCROI_{t2}$.

◦ Le nœud transporteur T1 dispose d'un certificat de réception $CRECEPT_{t1}$.
◦ Le nœud transporteur T2 dispose d'un certificat de réception $CRECEPT_{t2}$.
◦ Le nœud transporteur T1 dispose d'un certificat de réplication $CREPL_{t1}$.
◦ Le nœud transporteur T2 dispose d'un certificat de croisement $CREPL_{t2}$.

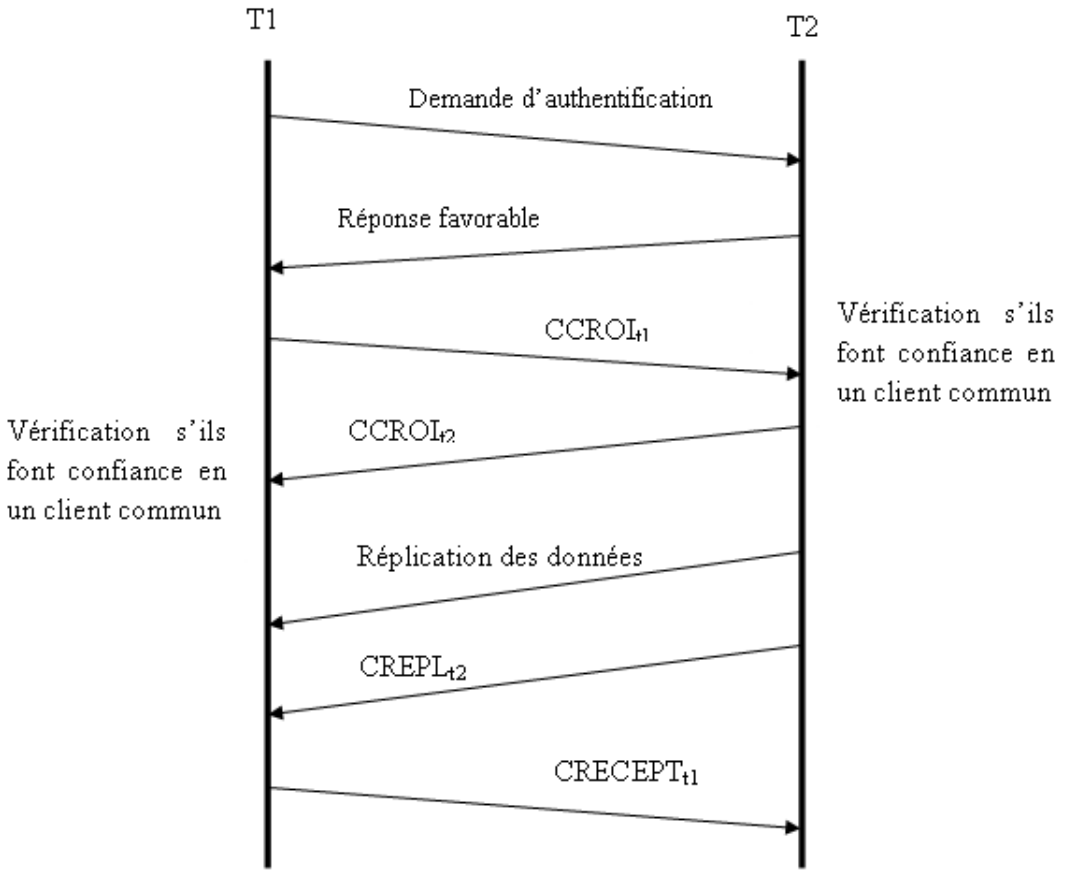


FIGURE 3.9 – Authentification entre deux transporteurs

Dans ce cas un nœud transporteur initie le processus d'authentification, en evoyant une demande d'authentification à l'autre nœud transporteur, suite à une réponse favorable,chacun d'eux doit délivrer à l'autre un certificat de croisement ; avec ce dernier un nœud transporteur peut prouver à un nœud client d'être entré en contact avec un autre nœud transporteur en qui ils font confiance. Chaque nœud transporteur identifie les identités des utilisateurs qui détiennent leurs données.

Les deux nœuds transporteurs doivent s'échanger seules les données des utilisateurs en qui les deux font confiance. Ceci sera vérifié seulement à travers les certificats qu'aura délivrés le nœud client pour les deux nœuds transporteurs. Une fois l'échange est fait, le nœud transporteur qui a répliqué les données, délivre à l'autre un certificat de réplication. En réponse, le deuxième nœud transporteur lui délivre un certificat de réception.

Â chaque opération de transfert accomplie vers le réseau avec infrastructure, le nœud transporteur doit répondre, lors de son prochain contact avec le nœud client, avec un

accusé de réception du destinataire et l'ensemble des certificats de réplication, de réception et de croisement qu'ont lui délivré les nœuds transporteurs avec lesquels ses données ont été échangées. Ceci est nécessaire pour le nœud client afin qu'il puisse mettre à jour le degré de la confiance de ses nœuds transporteurs.

3.2.8 Échange sécurisé des données (protocole de confidentialité)

Une fois les relations de confiance établies , les clés générées, les certificats délivrés et l'authentification mise en œuvre, l'échange de données entre un client et un transporteur peut être établi. Puisque le nœud client détient la clé publique du transporteur (respectivement le nœud transporteur celle du client), il peut l'utiliser pour chiffrer les messages. Comme le nœud transporteur est le seul qui connaitre sa clé privée donc, lui seule peut les déchiffrer.

Protocole d'échange sécurisé :

- DATA : données à transmettre.
- PK_i : clé publique d'un nœud client/transporteur i.
- SK_i : clé privée d'un nœud client/transporteur i.

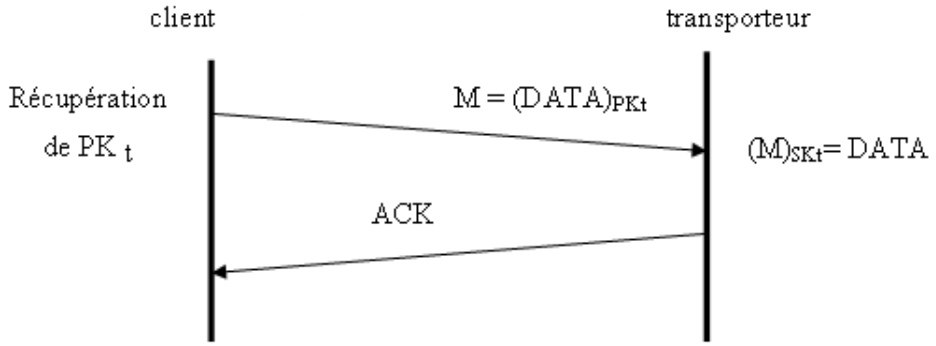


FIGURE 3.10 – Échange sécurisé des données

1. Le nœud client chiffre les données avec la clé publique du nœud transporteur digne de confiance, M= $(DATA)_{PKt}$.
2. Il envoie le message M au nœud transporteur.
3. Lors de la réception, le nœud transporteur déchiffre le message M avec sa clé privée SK_t.
4. Le nœud transporteur répond par un accusé de réception (ACK).

3.2.9 Réplication des données

Une fois les données échangées entre un client et un transporteur, elles peuvent être transportées directement vers internet comme elles peuvent être répliquées entre plu-

sieurs nœuds transporteurs intermédiaires dont le client fait confiance.

La réplication des données consiste à la création d'un double pour assurer la continuité de service même en cas de la présence des pannes, cette méthode permet aussi de gagner du temps, tout en acheminant le message vers le nœud transporteur près de la destination (internet).

Protocole de réplication des données (figure 3.11) :

◦ DATA : données du client à répliquer.
◦ SK_i : clé privée du transporteur i.
◦ PK_i : clé publique du transporteur i.

Avant le lancement de la procédure de réplication, les deux nœuds transporteurs doivent s'authentifier mutuellement, en s'échangeant leurs certificats à clés publiques.

Une fois les deux nœuds authentifiés, ils vérifient si la clé publique du client dont les données vont être répliquées figure dans leurs listes de clés publiques valide. Si la clé publique du client figure dans chaque liste alors le client fait confiance aux deux transporteurs, ce qui donne la possibilité de réplication, sinon la réplication ne peut pas être effectuée.

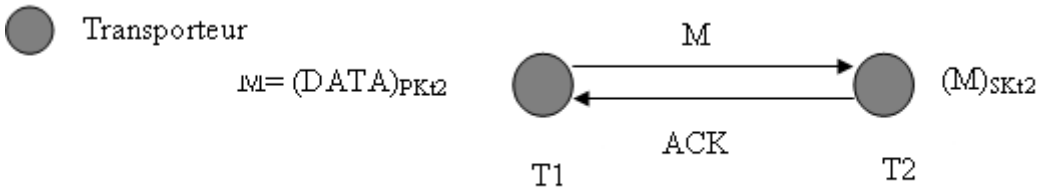


FIGURE 3.11 – Réplication des données

1. Le nœud transporteur T1 chiffre les données (DATA) à répliquer avec la clé publique de transporteur T2 (M= $(DATA)_{PKt2}$).
2. Il envoie le message M au transporteur T2.
3. Lors de la réception de M, le transporteur T2 déchiffre le message en utilisant sa clé privée SK_{t2}.
4. le transporteur T2 répond par un acquittement.

3.2.10 Mise à jour de la confiance des nœuds transporteurs

Chaque nœud client évalue régulièrement le degré de la confiance Ci (initialisé à 1) de chaque nœud transporteur i par rapport à l'ensemble des différents certificats qu'il détient. Les règles d'évaluation sont faites selon trois cas possibles :

1. Le nœud transporteur i est rentré en contact avec le nœud client pour un nouveau transfert de données, alors qu'il n'a pas remis l'accusé de réception du transfert précédent.

Dans ce cas, le nœud client constate que le nœud transporteur i a quitté puis rejoint la région sans faire suivre ses données vers le destinataire dans le réseau avec infrastructure, ce qui lui permet de dégrader son degré de confiance (Ci<-Ci-1). Autrement, à chaque réception d'un accusé de réception, le nœud client augmente le degré de confiance du ntransporteur i (Ci<-Ci+1).

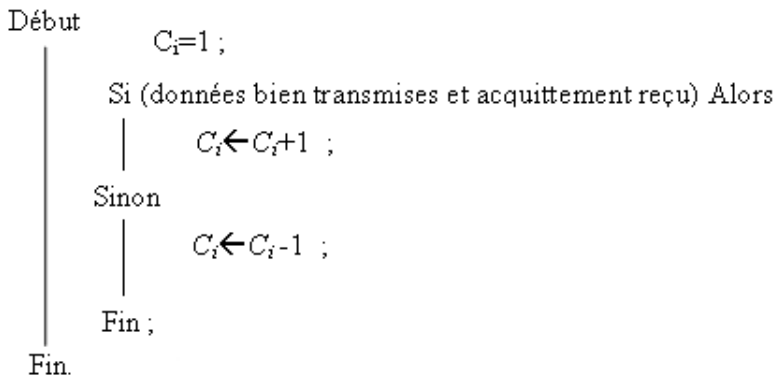
2. Le nœud client détient un certificat de croisement de deux nœuds transporteurs i et j, alors que chacun d'eux n'a délivré à l'autre un certificat de réplication et un certificat de réception.

Dans ce cas, le nœud client constate qu'il y a eu un contact entre deux de ses nœuds transporteurs alors qu'il n'a eu aucune réplication de ses données, ce qui lui permet de dégrader le degré de la confiance du nœud transporteur qui détient les données.

3. Le nœud client détient un certificat de croisement de deux nœuds transporteurs i et j, dans lequel j a reçu un certificat de réplication et i a reçu un certificat de réception.

Pour ce cas, le nœud constate que les deux nœuds transporteurs ont effectué la réplication de ses données, ce qui lui permet d'augmenter le degré de la confiance des deux nœuds.

Le degré de confiance peut être mis à jour selon l'algorithme suivant :



3.2.11 Révocation des certificats

Dans les réseaux DTN, chaque nœud participe à la communication seulement pendant la période de validité du certificat. Une fois le certificat expiré, il doit être révoqué puis renouveler pour une nouvelle utilisation.

Le problème le plus défficile à résoudre dans une PKI est la révocation, la révocation d'un certificat se fait à travers la création d'un *certificat de révocation*. Un certificat peut être révoqué pour les raisons suivantes :

(1) Si le degré de la confiance du nœud transporteur est inférieur à 1.

(2) Si la période de validité d'un certificat a expiré.

(3) Si la clé privée d'un client a été divulguée.

(4) Si un nœud décide de quitter le réseau, c'est-à-dire qu'il n'est plus client.

Dans notre cas d'application, nous avons opté pour un système de révocation distribué, il permet à chaque nœud de révoquer son propre certificat numérique ou celui d'un autre nœud, contrairement à ce principe, une autorité de certification centrale est la seule responsable de la révocation des certificats dans un système centralisé. Pour résoudre les problèmes de révocation, il existe deux solutions variables : les listes de révocation de certificats (LRC) et l'expiration rapide.

- **Liste de révocation de certificats (LRC)**

Chaque nœud client/transporteur doit disposer d'une base de données (LRC), qui contient la liste des certificats révoqués (en plus des informations uniques sur les certificats révoqués, afin de déterminer si un certificat est révoqué ou non).

Quiconque veut vérifier la validité d'un certificat, doit examiner la base de données de la LRC pour s'assurer qu'il n'a pas été révoqué. Dès qu'un certificat est ajouté à la LRC, aucune transaction n'est plus autorisée, la clé publique correspondante ne sera plus utilisée.[4]

- **Expiration rapide**

Avant toute transaction de messages, chaque nœud client/transporteur échange des certificats possédant une durée de validité, l'expiration rapide des certificats permet d'annuler un certificat le plus rapidement possible, te que chaque nœud délivre à l'autre un certificat avec une durée de validité très courte égale au temps nécessaire pour l'échange des messages, c'est-à-dire qu'il ne serra utilisé que pendant une durée calculée. Â Chaque fois que deux nœuds veulent communiquer, ils s'échangent des certificats avec une durée d'expiration précise.[33]

L'expiration rapide est peu couteuse, contrairement à une LRC, cependant l'utilisation d'une LRC est plus souhaitable dans notre cas d'application, car les délais de réponse trop longs dans les réseaux DTN peuvent perturber le processus de l'expiration rapide.

Dans un réseau DTN, la période de validité d'un certificat doit être assez grande pour que les retards impliqués dans la propagation, le renouvellement et la réponse n'affluent pas sur le système de sécurisation.

- **Échange sécurisé de l'identifiant d'une clé publique entre deux nœuds**

◦ PKID : identifiant d'une clé.
◦ H : fonction de hachage, les deux nœuds doivent se mettre d'accord sur elle.
◦ h : haché (empreinte digitale).
◦ PK_i : clé publique d'un nœud client/transporteur i
◦ SK_i : clé privée d'un nœud client/transporteur i.

- **Protocole d'échange sécurisé**

1. Le nœud client calcule H(PKID)=h.
2. Le nœud client signe avec sa clé privée le message M = $(h, PKID)_{SKc}$.
3. Le nœud client envoie M au nœud transporteur.
4. Lors de la réception, le nœud transporteur déchiffre le message avec la clé publique du nœud client.
5. Le nœud transporteur calcule le haché de PKID, H(PKID)=h1.
6. Le nœud transporteur vérifié si h=h1, si oui il répond par un acquittement positif, sinon il répond par un acquittement négatif car l'identifiant a été corrompu.

- **Procédure de révocation des certificats**

1. Chaque nœud dispose de l'identifiant de la clé publique à révoqué PKID.
2. Le nœud client dispose d'une liste des clés révoquées LRC_{client}.
3. Le nœud transporteur dispose d'une liste des clés révoquées $LRC_{transporteur}$.

- **Révocation d'une clé d'un transporteur**

Dans un réseau DTN, un nœud client peut faire confiance à plusieurs nœuds transporteurs, tel que chaque nœud transporteur a une connaissance préalable de cette confiance mutuelle. Ce dernier peut être indigne de confiance à un certain moment, en adoptant un comportement malveillant (par exemple : il peut ne pas répliquer les données d'un client), donc il est indispensable de prévenir et d'informer le reste de l'ensemble de ses nœuds transporteurs de ne plus répliquer ses données vers ce transporteur suspect.

- **Protocole de révocation**

Dans le protocole qui suit, on suppose que le nœud client fasse confiance à seulement deux nœuds transporteurs, dont l'un est suspecté. L'échange du PKID suit le protocole d'échange sécurisé décrit ci-dessus :

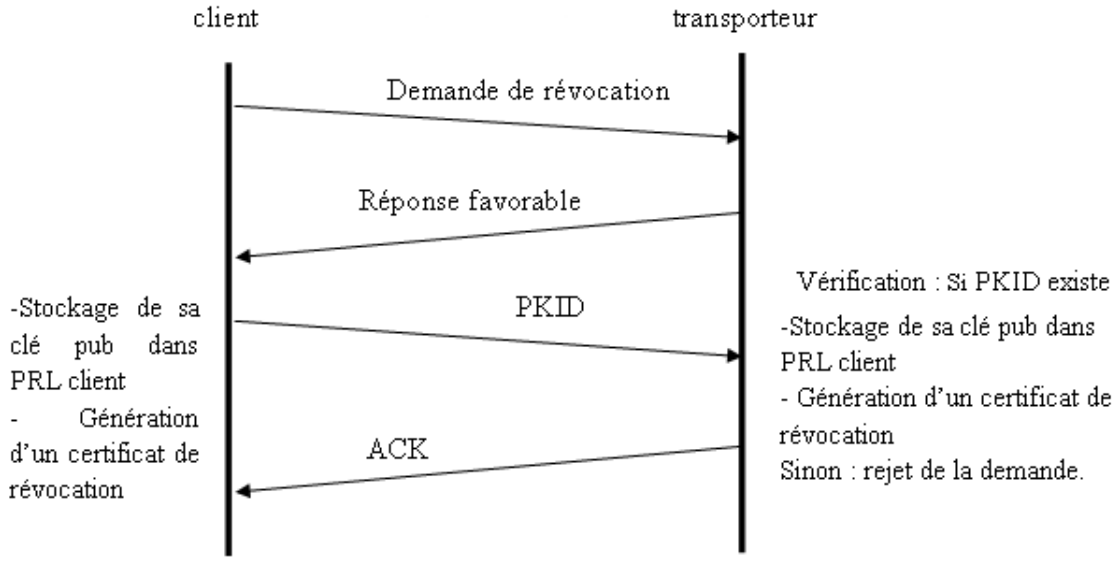


FIGURE 3.12 – Révocation d'une clé d'un transporteur

1. Le nœud client envoie une demande de révocation pour le nœud transporteur.
2. Le nœud transporteur répond par une réponse favorable.
3. Le nœud client stocke la clé publique du transporteur suspect dans sa liste des clés révoquées LRC_{client} , puis lui génère un certificat de révocation .
4. Le nœud client émet PKID (de la clé de nœud transporteur suspect) au transporteur digne de confiance.
5. Lors de la réception de PKID, le nœud transporteur vérifie s'il est correct (s'il appartient à son trousseau de clés publiques).

Si le PKID appartient à son trousseau de clés publiques alors la clé publique correspondante sera stockée dans sa $LRC_{transporteur}$, par la suite il lui génère un certificat de révocation et informe le client par un acquittement. Sinon (la clé ne fait pas parti de son trousseau) la demande sera rejetée.

• **Révocation de la clé publique d'un propriétaire (client/transporteur)**

Un nœud donné peut à un certain moment décider d'arrêter de travailler avec sa propre clé publique, soit par ce que sa clé privée correspondante a été divulguée ou par ce que sa période de validité a expiré, dans les deux cas, la révocation de cette clé est nécessaire, cette dernière peut suivre la procédure suivante :

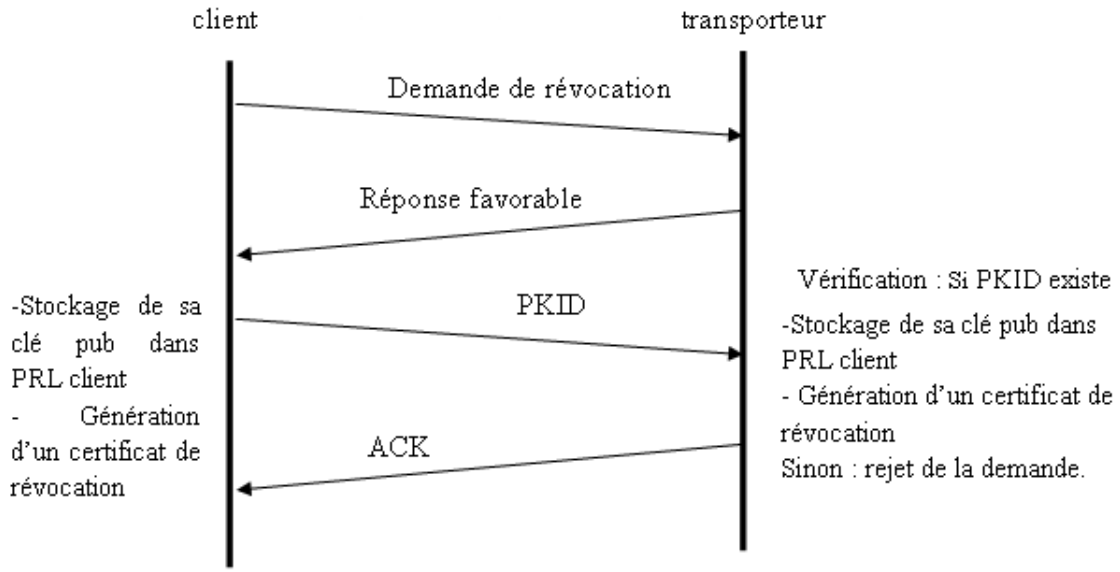


FIGURE 3.13 – Révocation de la clé d'un propriétaire (Client)

1. Le nœud client envoie une demande de révocation au nœud transporteur.
2. Le nœud transporteur répond par un avis favorable.
3. Le nœud client/transporteur envoie l'identifiant de sa clé publique PKID au nœud transporteur/client.
4. Lors de sa réception, le transporteur/client vérifie si PKID appartient à son trousseau de clés alors le transporteur/client lui génère un certificat de révocation et une place dans sa $LRC_{transporteur}$ et informe le client/transporteur, sinon la demande sera rejetée.
5. Le nœud client/transporteur lui génère aussi un certificat de révocation et une place dans sa LRC_{client}.

3.2.11.1 Destruction des clés

Les clés révoquées doivent être détruites d'une manière sûre,ces veilles clés ont de la valeur même si elles ne sont plus utilisées, pour vérifier des signatures faites au passé et pour déchiffrer des messages faits au passé, avec ces dernières un attaquant (adversaire) peut lire les anciens messages chiffrés.

3.2.11.2 Mise à jour des clés

La clé privée d'un nœud peut être divulguée ou modifiée par un attaquant voulant toucher à l'intégrité du réseau. Une fois la clé est modifiée, un nœud ne peut continuer de l'utiliser, donc il est nécessaire de générer une autre paire de clés.

Il est souvent fastidieux de distribuer une nouvelle clé chaque jour. La parade trouvée est de générer une nouvelle clé à partir de l'ancienne (mettre à jour les clés). Cette nouvelle clé n'est pas aussi sûre que l'ancienne, si cette dernière a été divulguée alors la nouvelle clé

peut aussi être divulguée. De même, la mise à jour des trousseaux de clés est indispensable une fois les clés publiques révoquées.

Conclusion

Les réseaux DTN constituent, de leurs nature, un challenge pour la sécurité informatique. Plusieurs contraintes concourent à rendre la sécurité des réseaux DTN difficile et complexe à appréhender. De ce fait, ce chapitre permet d'approcher à ce domaine et de pouvoir tester la mise en place d'une nouvelle architecture de sécurisation permettant d'apporter une meilleure gestion de sécurité dans ces réseaux.

Ce chapitre a apporté une stratégie de gestion des clés publiques pour le réseau DTN, qu'elle repose sur un modèle de confiance hiérarchique basé sur la certification de clé publique où chaque nœud joue le rôle d'une autorité de certification, ainsi La délivrance des certificats, l'authentification des communicants, la mise à jour de la confiance des nœuds transporteurs et la révocation des certificats.

Chapitre 4

Modélisation de notre solution

Introduction

Dans le chapitre précédent, nous avons proposé un graphe de confiance pour une architecture spécifique du réseau DTN. Le réseau est composé de plusieurs sous-réseaux se trouvant dans des régions géographiquement isolées et qui ont un accès intermittent au réseau avec infrastructure de communication. Chaque sous-réseau comporte des nœuds clients et des nœuds transporteurs qui assurent l'acheminement des données vers le réseau avec infrastructure.

La théorie des graphes est un outil très puissant pour la modélisation, elle ouvre un grand champ de modélisation conduisant à des solutions efficaces. Dans ce chapitre nous allons introduire les notions de base de la théorie des graphes pour modéliser notre graphe de confiance.

4.1 Graphe de confiance

Le graphe de confiance permet de relier les différents nœuds du réseau, ce dernier est un graphe G (V , E) non orienté, représenté comme suit :

G : est le nom du graphe.
V = {V_1,...,V_n} : représente l'ensemble des sommets du graphe (les nœuds clients et les nœuds transporteurs).
E = {{V_i , V_j } | V_i, $V_j \in V$ } représente l'ensemble des arêtes, ces dernières sont des arcs reliant deux nœuds (client et transporteur) par une relation de confiance.

4.2 Coloration de graphe

Colorer un graphe consiste à affecter une couleur à chacun de ses sommets de sorte que deux sommets adjacents ne portent pas la même couleur, autrement dit : une k-coloration (k entier positif qui représente le nombre de couleurs) est une fonction c : $V \rightarrow \{1..k\}$ telle que pour toute arête {V_i , V_j } de E, | c(V_i)- c(V_j) | <> 0.

Notre graphe de confiance est composé de deux types de nœuds : clients et transporteurs, on parlera donc d'une 2-coloration :

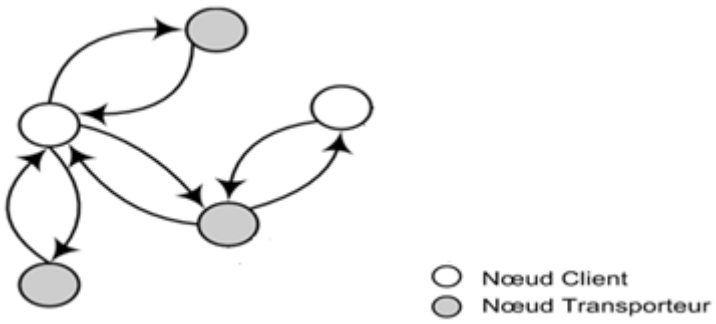


FIGURE 4.1 – Coloration le graphe de confiance

Le nombre chromatique d'un graphe est le plus petit nombre de couleurs permettant de le colorer. Dans notre cas, il est égal à 2.

4.3 Modélisation

Le modèle que nous proposons s'adresse en particulier à l'architecture DackNet du réseau DTN, notre réseau est modélisé par un graphe coloré, cependant il ne prend pas en considération le problème de coloration des graphes.

4.3.1 Propriétés du réseau

La modélisation du réseau est fortement conditionnée par des propriétés, elles ont pour objet de décrire les caractéristiques du réseau à prendre en considération avant tout :

- Le réseau est situé dans une zone géographique délimitée.
- Les nœuds clients sont fixes, se sont eux qui réclament des services sur Internet.
- Les nœuds transporteurs sont mobiles, ils se chargent du transfert (transport) des requêtes des clients vers le réseau avec infrastructure.
- Le réseau est fermé c'est-à-dire qu'aucun ajout de nœuds n'est pris en considération.
- Les données sont échangées entre les nœuds à un saut.
- La taille du réseau (nombre de nœuds) est prévue à l'avance.

4.3.2 Les données du problème

Nous allons tenir compte des paramètres suivants :

- **N** : taille du réseau.
- **D** : données destinée à la transmission.
- **X** : taille d'une donnée.
- **Y** : capacité de stockage d'un nœud transporteur.

- α : entier positif ($\alpha > 0$).
- **Nc** : nombre de nœuds clients.
- **Nt** : nombre de nœuds transporteurs.
- X_{total} : taille total des données.
- Y_{total} : espace de stockage total.
- **T** : durée de transfert.
- **Np** : nombre de nœuds transporteurs tombés en panne.

4.3.3 Supposition

- L'échange de données se fait entre deux nœuds de couleurs différentes (coloration propre).
- La réplication de données se fait entre deux nœuds de même couleur (coloration impropre).

4.3.4 Protocole de transfert de données

Supposons qu'un nœud client C1 veut transmettre une donnée D de taille X vers un nœud transporteur T1 disposant d'un espace de stockage de taille Y.
Le nœud transporteur peut véhiculer la donnée du nœud client, si et seulement si la taille de la donnée est inférieur ou égale à son espace de stockage (X < = Y).

Si (X < = Y) alors le nœud transporteur accepte la requête et transporte le message, dans le cas contraire le nœud transporteur la rejette.

Le même scénario est décrit dans le cas de la réplication des données entre deux nœuds transporteurs.

4.3.5 Optimisation du nombre des nœuds transporteurs dans le réseau

Le but principal de cette sous-section est le calcul de nombre de nœuds transporteurs optimal dans le graphe, pour résoudre cette problématique nous avons opté pour un modèle mathématique, illustré en deux étapes :

Étape 1

Dans la première étape, on considère que la taille des données X est la même pour tous les nœuds clients, et que l'espace de stockage Y est le même pour tout les nœuds transporteurs.

1. La taille totale des données est définie par : X_{total} = X+ X+ ... + X.
$$X_{total} = \text{Nc} * \text{X}.$$
2. L'espace de stockage total est défini par : Y_{total} = Y+ Y+ ... +Y.
$$Y_{total} = \text{Nt} * \text{Y}.$$

3. Nous allons considérer que la taille totale des données est inférieur ou égale à l'espace de stockage total c'est à dire : $X_{total} <= Y_{total}$.

Modèle mathématique
Nous allons prendre en considération la taille totale des données, l'espace de stockage total des nœuds transporteurs et la taille totale du réseau pour former le système suivant :

$$(*) = \begin{cases} Nc + Nt = N \ldots\ldots\ldots\ldots\ldots\ldots (1) \\ \\ Nc * X - Nt * Y <= 0 \ldots\ldots\ldots\ldots\ldots\ldots (2) \end{cases}$$

Pour résoudre (*), on procède ainsi :
De l'équation (1), le nombre de nœuds clients peut être calculé comme suit :

$$Nc = N - Nt.$$

En remplaçant Nc par sa valeur dans l'équation (2) on obtient :

$$(N - Nt) * X - Nt * Y < = 0.$$

Le nombre de nœuds transporteurs peut être calculé à partir de la formule précédente par la résolution mathématique.

$X * N - X * Nt - Y * Nt < = 0.$
$X * N - Nt * (X + Y) < = 0.$
$X * N < = Nt * (X + Y).$
$Nt * (X + Y) > = X * N.$

$$\boxed{Nt >= (X * N) / (X + Y).}$$

- **Cas (X < =Y) :** Le nombre de nœuds transporteurs optimal dans le réseau est :

$$Nt = (X * N) / (X + Y).$$

- **Cas (X >Y) :**
  Le nombre de nœuds transporteurs n'est pas défini, car ils n'ont pas l'espace nécessaire pour le transport des données.

Étape 2

Dans la deuxième étape, on considère que la taille des données est égale à α X et qu'elle est identique pour tous les nœuds clients, et que l'espace de stockage est de taille Y est identique pour tous les nœuds transporteurs.

1. Le nombre de nœuds transporteurs optimal dans le réseau peut-être déterminé de la même façon que cité dans la première étape.

2. La taille totale des données est définie par : X_{total} =α X+ α X+ ... + α X.

$$X_{total} = \text{Nc} * \alpha \text{ X}.$$

Dans ce cas la taille totale des données est considérée aussi inférieur ou égale à la taille totale de l'espace de stockage, c'est à dire que : $\alpha <= $ Y / X.
En remplaçant X par α X dans le modèle mathématique précédent,on trouve que, le nombre de nœuds transporteurs est : Nt > = (α X * N) / (α X + Y).

- **Cas ($\alpha <= $ Y / X) :**
  Le nombre de nœuds transporteurs optimal dans le réseau est :

$$\text{Nt} = (\alpha \text{ X} * \text{N}) / (\alpha \text{ X} + \text{Y}).$$

- **Cas ($\alpha >$ Y / X) :**
  Le nombre de noeuds transporteurs est indéfini.

Si les tailles des données sont différentes pour tous les nœuds, la taille totale des données est :

$$X_{total} = \sum_{(i=1)}^{Nc} Xi.$$

Si les tailles des espaces de stockages sont différentes pour tous les nœuds, la taille totale des données est :

$$Y_{total} = \sum_{(j=1)}^{Nt} Yj.$$

Pour que les nœuds transporteurs puissent répondre à toutes les requêtes des nœuds clients, en acheminant leurs données vers le réseau avec infrastructure. Il est indispensable que l'espace de stockage soit suffisamment grand que la taille des données :

$$\sum_{(j=1)}^{Nt} Yj >= \sum_{(i=1)}^{Nc} Xi.$$

Dans ce cas on cherche à :
- Maximiser l'espace de stockage : Max $\sum_{(j=1)}^{Nt} Yj$.
- Minimiser la taille des données : Min $\sum_{(i=1)}^{Nc} Xi$.

4.3.6 Impact de la défaillance des nœuds transporteurs

Un nœud transporteur peut tomber en panne à n'importe quel moment, ce qui perturbe le bon acheminement des données.
Supposons qu'une donnée D de taille X, nécessite Nt nœuds transporteurs pour son acheminement, pendant une durée T. Si Np est le nombre de nœuds transporteurs tombés en panne, alors X nécessitera $\text{N}t'$ nœuds transporteurs, pendant une durée T'. Tel que :

$$\begin{cases} Nt' = Nt - Np. \\ T' = T * Nt' \ / Nt. \end{cases}$$

Quelques résultats

Dans cette partie nous allons étudier l'impacte de la défaillance des nœuds transporteurs, en changeant à chaque fois un des paramètres initiaux, les résultats sont illustrés par des graphiques sous Excel. On suppose que T reste fixe dans tout les cas (par exemple : T=6).

- **Cas Nt augmente et Np fixe**

Nt'	2	3	5	7
Nt	3	4	6	8
T'	4	4.5	5	5.25

Remarque : D'après le graphique suivant, on remarque que dans ce cas la durée de transfert après défaillance augmente.

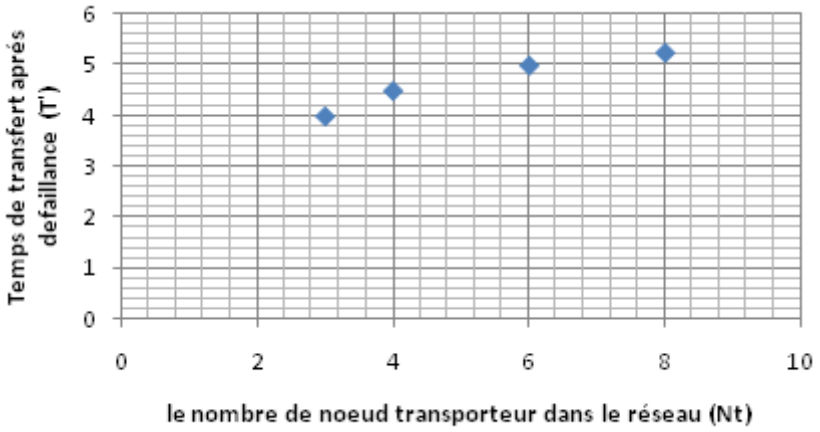


FIGURE 4.2 – Cas Nt augmente et Np fixe

- **Cas Nt augmente et Np augmente**

Nt'	2	2	2	2
Nt	3	4	6	8
T'	4	3	2	1.5

Remarque : Dans ce cas la durée T', décrémente comme illustré dans le graphique suivant.

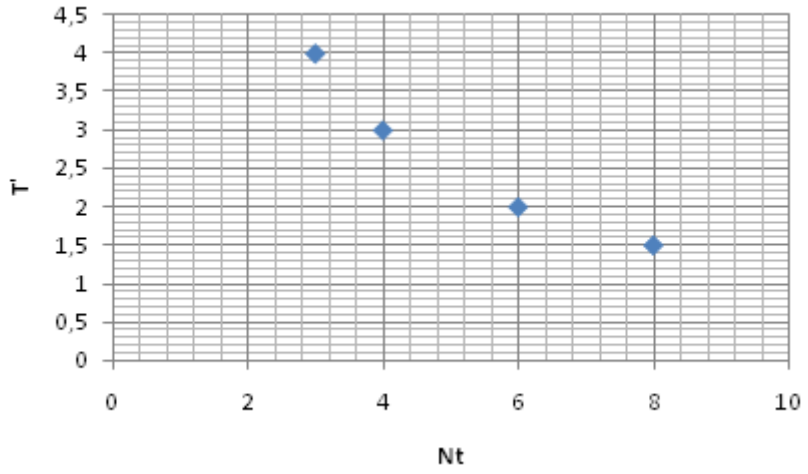


FIGURE 4.3 – Cas Nt augmente et Np augmente

- **Cas Nt décrémente et Np décrémente**

Nt'	2	2	2	2
Nt	8	6	4	3
T'	1.5	2	3	4

Remarque : Dans ce cas la durée T' décrémente aussi .

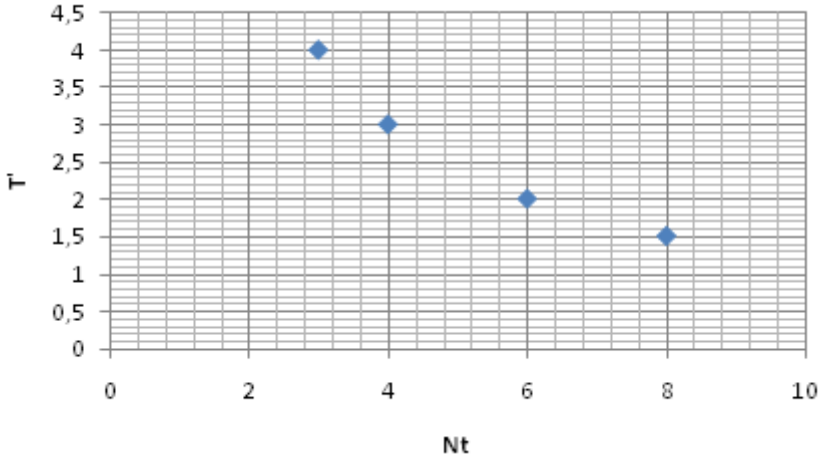


FIGURE 4.4 – Cas Nt décrémente et Np décrémente

- **Cas Nt décrémente et Np fixe**

Nt'	6	4	2	1
Nt	8	6	4	3
T'	4.5	4	3	2

Remarque : Dans ce cas la durée $T^{'}$ augmente, comme illustré dans le graphique qui suit. .

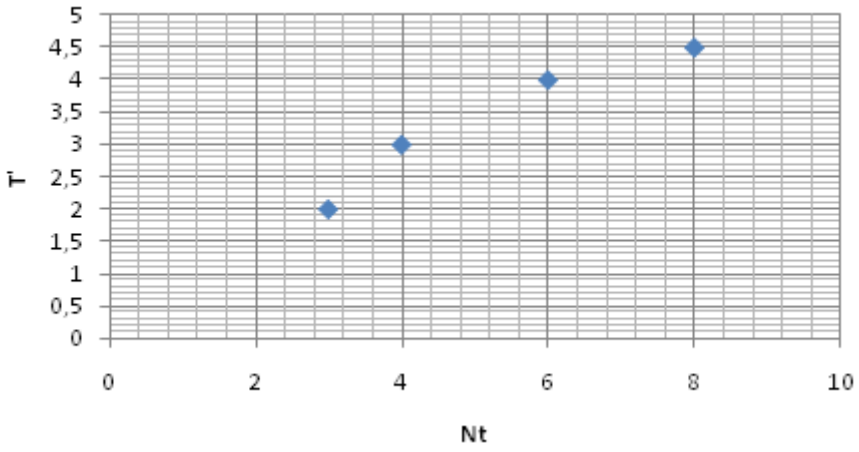


FIGURE 4.5 – Cas Nt décrémente et Np fixe

Conclusion

Modéliser un problème est une étude très importante dans tout les travaux de recherches, dans ce chapitre nous nous somme intéressés aux concepts de bases de la théorie des graphes et à quelque notions mathématiques. Pour modéliser notre graphe de confiance, nous avons calculé le nombre de nœuds transporteurs optimal dans le réseau, pour plusieurs scénarios, nous avons aussi étudié l'impact des nœuds transporteurs défaillants sur le temps de transfert des données.

Conclusion générale et perspectives

Nous avons présenté à travers ce mémoire la sécurité des réseaux tolérants aux délais. La connaissance d'un tel type de réseaux est très important de nos jours, car il offre un nouveau développement dans le monde de la recherche sur les réseaux, et redonne la possibilité de connecter des entités implantées dans des environnements isolés géographiquement, qui empêche leurs communications.

Nous avons connu à travers ce dernier, le mode de fonctionnement des DTN et tout ce qui s'y réfère, à savoir leurs architectures, les raisons qui ont amenées à leurs apparitions, nous avons détaillé la couche protocolaire, et expliqué le principe sur lequel est fondé le protocole bundle, comme nous avons expliqué le principe des deux techniques de commutation de messages : "Store And Forward" et le "transfert de garde".

Notre centre d'intérêt se localise sur un système de gestion des clés publiques pour ces derniers, notre stratégie met en avant les relations sociales de confiances, et se base sur un modèle de certification distribué. Pour cela, nous avons aussi présenté les notions de bases sur la sécurité informatique.

Les réseaux DTN ont le potentiel de connecter des environnements hétérogènes dont une connexion bout en bout ne peut jamais exister. Pour que la communication soit possible, les nœuds intermédiaires prennent la garde des données « store » et les retransmettent « forward » quand l'opportunité se présente.

Les mécanismes de sécurité traditionnels ne répondent pas aux exigences de tels réseaux. Il s'agit donc de concevoir de nouveaux mécanismes afin de garantir leurs sécurité. Actuellement, il n'existe pas un modèle de gestion pour les clés publiques adaptées pour de tels réseaux.

Pour cela, à l'issu de ce travail, nous avons défini une architecture de gestion des clés publiques adaptée à ce type de réseaux. L'architecture proposée n'a pas la prétention de correspondre à toutes les situations d'usage de tels environnements et pose encore des questions. En effet, elle aborde le problème de la gestion des clés publiques pour une architecture spécifique de ces réseaux, non pas pour l'architecture globale. Cette dernière se focalise sur un réseau DackNet fermé, l'ajout de nouveaux nœuds ou leurs suppressions

ne sont pas prises en considération.

Il est impossible de déterminer quelle approche est la plus adéquate quand il n'y a aucun réseau DTN étant activement employé. Peut-être les travaux futurs les plus importants seront de déployer des applications DTN, et voient alors ce que les problèmes de sécurité se produisent dans la pratique.

En guise de perspectives, nous prévoyons dans un premier temps de compléter notre travail en implémentant notre proposition et la comparer par simulation avec d'autres solutions si elles existent, nous envisageons aussi de faire une gestion des clés publiques pour l'architecture globale du réseau.

Pour conclure, ce thème de recherche se situe dans le cadre d'un domaine plus général, il nous a permis de toucher à la réalité du travail dans un environnement orienté recherche. On a appris à considérer les notions d'un œil critique afin de vouloir toujours améliorer, toujours essayer de changer et d'innover. L'avenir des solutions plus efficaces pour la sécurité des DTN pourrait devenir possible.

Bibliographie

[1] M.J.B. Robshaw, RSA Laboratories "On Recent Results for MD2, MD4 and MD5" Number 4 , November 12, 1996.

[2] Raymond G. Kammer and William Mehuron, Information Technology Laboratory " Data Encryption Standard " , October 25, 1999 .

[3] Joel Weise , $SunPS^{SM}$ Global Security Practice Sun $BluePrints_{TM}$ OnLine "Public Key Infrastructure", August 2001.

[4] bruce schneier, Livre "CRYPTOGRAPHIE APPLIQUÉE algorithmes, protocoles et code en C" , traduction de laurent viennot ; vuibert , paris, 2001 .

[5] Federal Information, Processing Standards Publication "Advanced Encryption Standard (AES)" November 26, 2001.

[6] Stéphane Natkin, Dunod , Livre "les protocole de sécurité D'internet" ,Paris, 2002.

[7] Asma BENMESSAOUD, "Classification des protocoles de routage dans les réseaux DTN" , École nationale supérieure d'informatique, Alger, Novembre 2002.

[8] Groupe CGI, étude technique " Cryptographie à clé publique et signature numérique", les principes de fonctionnement, Septembre 2002.

[9] Vinton Cerf, Scott Burleigh, Adrian Hooke, Leigh Torgerson, Robert Durst, Keith Scott, Kevin Fall, Howard Weiss, Tutorial : "Delay Tolerant Network Architecture", DTN Research Group Internet Draft, Mars 2003.

[10] F.Warthman , "Delay Tolerant Networks (DTN) Atutoriel". Warthman Associates, version 1.1, Mai 2003.

[11] Niels ferguson ; bruce schneier, vuibert(edition) Livre : "Cryptographie En Pratique", Pris, 2004.

[12] Véronique Legrand et Stéphane Ubéda, Joël Morêt-Bailly et Agnes Rabagny, Laurent Guihéry et Jean-Philippe Neuville, Article : "Vers un modèle de confiance pour les objets communicants : une approche sociale" , 1 mars 2004.

[13] Brent A. Cottom, CS 6520 Cryptography " Blowfish" , 18 Août 2004.

[14] José R. Beuret et Gwenol Grandperrin , " Le protocole Diffie-Hellman " Cryptographie - EPFL , Juin 2006.

[15] M.Mehdi, A.Anou, S.Zair, M.Bensebti and M.Djebari « la Sécurité dans les Réseaux Ad Hoc », March 25-29, 2007.

[16] Granville Barne and Luca Del Tongo, Annotated Reference with Examples "Data Structures and Algorithms : DSA" , 2008.

[17] E-watching.net, sécurité informatique "Explications sur la cryptographie" Version 1.0, 22 fevrier 2008.

[18] Patra R, Surana S, Nedevschi S. Hierarchical identity based cryptography for end-to-end security in DTNs. In : 4th international conference on intelligent computer communication and processing (ICCP) ; August 2008.

[19] Michel Asencio, Chercheur associé dans la Sécurité des Systèmes d'Information : "L'infrastructure de gestion des clefs publiques à la portée de tous...", 15 septembre 2008.

[20] Abdesselem BEGHRICHE, Mémoire de magistère en informatique, De la sécurité a la e-confiance basée sur la cryptographie à seuil dans les réseaux sans fils ad hoc, Université L'Hadj Lakhdar- Batna, 2008/2009

[21] Asma Benmessaoud, Mémoire de magister, "Classification des protocoles de routage dans les Réseaux Tolérants aux Délais (DTN)", ESI (Ecole nationale Supérieure d'Informatique),promotion 2008/2009.

[22] Farrell S, Symington SF, Weiss H, Lovell P. "Delay-tolerant networking security overview" , Work in process as an internet-draft, draft-irtf-dtnrg-sec-overview- 06, March 2009.

[23] Bhutta N, Ansa G, Johnson E, Ahmad N, Alsiyabi M, Cruickshank H. Security analysis for delay/disruption tolerant satellite and sensor networks. In : International workshop on satellite and space communications (IWSSC), September 2009.

[24] Symington S, Farrell S, Weiss H, Lovell P. Bundle security protocol specification. draft-irtf-dtnrg-bundle-security-15, February 2010.

[25] Lu R, Lin X, Shen X. Spring : a social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In : The 29th IEEE inter- national conference on computer communications (INFOCOM) ; March 2010.

[26] Stéphane GRATON, Projet de recherche présenté comme exigence partielle de la maitrise en génie logiciel sur " Gestion des clés publiques pour les entreprises de l'économie sociale et solidaire " , université du Québec à Montréal, Hiver 2010.

[27] Zhongtian Jia, XiaodongLin, Seng-HuaTan, LixiangLi, YixianYang. Public key distribution scheme for delay tolerant networks based on two-channel cryptography, Journal of Network and Computer Applications, 9 March 2011.

[28] BENHAMIOUD Housseyn et KOUIRI Khaled, mémoire "La sécurité dans les réseaux tolérants aux délais (DTN)", juin 2011.

[29] E. Bresson, " Cryptographie : Chiffrement par flot" , SGDN/DCSSI Laboratoire de cryptographie.

[30] Paul C. Kocher , Cryptography Research, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems ".

[31] Olfa Samet and Francine Krief, "Apport des satellites et de l'architecture des DTN dans les réseaux ad hoc."(satellite.pdf).

[32] Kevin Fall, Wei Hong and Samuel Madden, Article : "Custody Transfer for Reliable Delivery in Delay Tolerant Networks".

[33] Fida Ullah Khattak, "Delay Tolerant Networks in Rural Areas", Department of Communication and Networking, Aalto University.

[34] O.Samet, "Apport des satellites et de l'architecture DTN dans les réseaux ad Hoc", SUP-COM-Tunisie Francine Krief Université Bordon1.

[35] Valérie Gayraud , Loutfi Nuaym2, Francis Dupont, Sylvain Gombault, and Bruno Tharon Article : "La Sécurité dans les Réseaux Sans Fil Ad Hoc".

[36] Abdesselem Beghriche, Azeddine Bilami, Article : "Modélisation et Gestion de la Confiance dans les Réseaux Mobiles Ad hoc" ; Département d'informatique, Université de Batna-Algérie.

[37] Aaditeshwar Seth and Srinivasan Keshav, "Practical Security for Disconnected Nodes", School of Computer Science University of Waterloo, Canada.

[38] Robert A. Nichols, A. Roger Hammons Jr, Daniel J. Tebben and Anurag Dwivedi, Article : "Delay Tolerant Networking for Free-Space Optical Communication Systems.