Diffusion multicast sécurisée dans un environnement Ad-Hoc (1 vers n séquentiel)

Mohamed Salah Bouassida, Abdelkader Lahmadi, Isabelle Chrisment, Olivier Festor

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Diffusion multicast sécurisée dans un environnement Ad-Hoc (1 vers n séquentiel)

Mohamed Salah Bouassida, Abdelkader Lahmadi, Isabelle Chrisment et Olivier Festor

N° 5310

Septembre 2004

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--5310--FR

INRIA
LORRAINE

Diffusion multicast sécurisée dans un environnement Ad-Hoc (1 vers n séquentiel)

Mohamed Salah Bouassida, Abdelkader Lahmadi, Isabelle Chrisment et Olivier Festor

Thème 1 — Réseaux et systèmes
Project Madyne

Research Report n° 5310 — Septembre 2004 — 47 pages

Résumé : Un réseau Ad Hoc est une collection de nœuds mobiles, formant un réseau temporaire, sans l'aide de toute infrastructure fixe ni administration centralisée. Cette flexibilité en temps et en espace induit de nouveaux défis envers l'architecture de sécurité à mettre en œuvre pour assurer des communications unicast et multicast sécurisées. Les approches de gestion de clé de groupe réalisées pour les réseaux filaires ne sont plus appropriées dans un tel environnement, essentiellement en raison de la forte dynamicité et de la mobilité des nœuds.

Dans ce rapport, nous présentons un nouveau protocole de gestion de clé de groupe dans le cadre des réseaux Ad Hoc. Ce protocole s'appelle BALADE. Il réalise une gestion de clé de groupe efficace, rapide et dynamique pour un service de communications multicast, 1 vers n séquentiel.

Mots-clés : Ad Hoc, Multicast, Sécurité, Gestion de clé

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Secure Multicast Streaming in an Ad-Hoc Environment (Sequential 1 to n)

Abstract: An Ad Hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. This flexibility in space and time induces new challenges towards the security infrastructure needed to support secure unicast and multicast communications. Traditional group key management architectures proposed for wired networks are not appropriate in such environments, mainly due to both high dynamicity and mobility of nodes.

In this report, we present a new key management protocol for secure multicast communications, dedicated to operate in Ad Hoc networks, called BALADE. Our approach delivers a fast, efficient and mobility aware key distribution in a sequential 1 to n multicast service.

Key-words: Ad Hoc, Multicast, Security, Key Management

Table des matières

Table des figures

Chapitre 1

Introduction Générale

Dans ce rapport, nous présentons une nouvelle approche de gestion de clé du groupe, dans un environnement Ad Hoc, pour des applications de diffusion de flux multimédia (streaming), dans le cadre précis de communications de groupe 1 vers n séquentiel (à tout instant t, il y a une et une seule source qui émet et une fois qu'elle termine, une autre source prend le relais). Au lieu de trouver une solution générique pour tout type d'applications, nous avons voulu nous focaliser sur ce modèle de communications de groupe qui est très répandu dans Internet, et que plusieurs applications utilisent, comme par exemple des audio/vidéo conférences, des diffusion MP3, ou dans des contextes moins ludiques et plus tactiques, comme les communications entre des groupes de forces de sécurité civiles ou militaires voulant s'échanger des informations confidentielles. Nous travaillons sur ce type d'applications cibles dans le cadre du projet SAFECAST [1].

Cette approche est appelée BALADE. Elle se propose d'assurer l'authentification et le contrôle d'accès des membres du groupe, la confidentialité des données, tout en prenant en compte la dynamicité et la mobilité des nœuds dans un tel environnement. La gestion de clé dans BALADE doit se faire de manière dynamique, efficace et nécessiter le minimum de puissance de calcul tout en atténuant le phénomène *1 affecte n*, n étant le nombre de membres affectés par une opération de renouvellement de clé de groupe.

1.1 Motivations

L'essor des technologies sans fil offre aujourd'hui de nouvelles perspectives dans le domaine des télécommunications. L'évolution récente des moyens de communication sans fil a permis le traitement de l'information à travers des unités de calcul portables qui ont des caractéristiques particulières et accèdent au réseau à travers une interface de communication sans fil. Cet environnement permet aux unités de calcul une forte liberté de mobilité et ne

1. SAFECAST est un projet RNRT qui vise à développer une architecture globale de sécurité permettant la communication multipoint dans un environnement sécurisé

pose aucune restriction sur la localisation des usagers. Les environnements mobiles offrent une grande flexibilité d'emploi. En particulier, ils permettent la mise en réseau des sites dont le câblage serait trop onéreux à réaliser, voire impossible.

Les réseaux mobiles sans fil peuvent être classés en deux catégories : les réseaux avec infrastructure qui utilisent généralement le modèle de la communication cellulaire, et les réseaux sans infrastructure ou les réseaux Ad Hoc. Un réseau Ad Hoc peut être défini comme une collection d'entités mobiles interconnectées par une technologie sans fil formant un réseau temporaire sans l'aide de toute administration ou de tout support fixe. Les nœuds dans un réseau Ad Hoc sont caractérisés par des liens sans fil, des ressources limitées en terme de bande passante, de CPU et de mémoire. Aucune hypothèse n'est faite ni contrainte établie sur la taille des réseaux. Ceux-ci peuvent donc être de taille très variable.

Les réseaux Ad Hoc sont actuellement au cœur d'une activité de recherche soutenue qui vise leur bon déploiement et mise en œuvre. Particulièrement, la sécurité dans les réseaux Ad Hoc constitue l'un des principaux obstacles à un large déploiement de ces réseaux.

1.2 La sécurité dans les réseaux Ad Hoc

Sécuriser un réseau Ad Hoc revient à instaurer les différents services de sécurité au sein du réseau, tout en prenant en compte les différentes caractéristiques d'un tel réseau. Les défis de la sécurité dans les réseaux Ad Hoc sont les suivants :

1- L'utilisation des liens sans fil rend un réseau Ad Hoc plus facilement exposé à des attaques malicieuses passives comme des écoutes clandestines, ou actives comme le renvoi ou la déformation des messages. Ces attaques violent ainsi le service d'authentification [ZH99].

2- L'absence d'infrastructure fixe élimine toute possibilité de pouvoir établir une référence centralisée afin de concentrer les accès au réseau en un seul point unique capable d'administrer les différents services indispensables au bon fonctionnement du réseau. De cette absence d'infrastructure, il en découle que les modèles classiques centralisés ou hiérarchiques d'authentification peuvent difficilement s'appliquer. C'est notamment le cas du modèle de confiance des Infrastructures à Clés Publiques ou PKI [HFPS99]. Paradoxalement, la plupart des solutions existantes pour la sécurisation des réseaux Ad Hoc supposent l'existence de clés privées/clés publiques et donc d'une infrastructure de gestion de clé. Hors dans certains scénarii, typiquement lorsque le réseau Ad Hoc n'est pas connecté à l'Internet, les nœuds n'ont pas accès à une infrastructure de sécurité de ce type. Le développement de solutions ne reposant pas sur une autorité de certification ou PKI est un vrai verrou scientifique. Pour y remédier, plusieurs travaux de recherche portent sur l'émulation d'un service de confiance qui pourrait jouer le rôle d'autorité de certification centralisée.

Ainsi [Leg03] extrait trois problématiques principales pour l'émulation d'un tiers de confiance dans un milieu Ad Hoc :

* La répartition de la clé privée de l'autorité de certification sur plusieurs nœuds du réseau : la clé privée d'une autorité de certification ou CA joue un rôle essentiel pour l'établissement de la confiance dans le réseau. En effet, elle assure le mécanisme de signature des certificats. Cependant, l'établissement d'une PKI basée sur une unique entité centralisée suppose que cette entité de confiance ne pourra jamais être compromise, ce qui n'est pas réalisable dans un environnement Ad Hoc. Pour cette raison, la clé privée de l'autorité de certification doit être répartie sur plusieurs nœuds qui partageront la capacité de générer et signer des certificats pour les autres nœuds du réseau.
* La gestion des contextes de confiance des nœuds en cours de mobilité : l'autorité de certification est responsable de la gestion des certificats (émission, renouvellement, révocation, ...). Pour cela, elle héberge ces données confidentielles dans des bases de données spécialisées. Dans le cadre des réseaux Ad Hoc, ce système de stockage de données n'est plus adapté, à cause de la forte mobilité des nœuds Ad Hoc. Une solution plus appropriée aux réseaux Ad Hoc consisterait à ce que les données relatives à un nœud soient hébergées par le nœud lui même grâce à une carte à puce.
* L'émission des certificats : un certificat X.509, dans une architecture PKI, est signé par la clé privée de la CA, et contient des références comme l'horodatage et le nom X.509. Ce schéma n'est plus possible pour les réseaux Ad Hoc. Plusieurs travaux ont ainsi cherché à alléger le certificat X.509 pour un certificat plus souple et plus libre utilisant SPKI [EFL+99].

3- La taille et la dynamicité du réseau Ad Hoc peuvent être très importantes. En effet, on ne peut pas contrôler le nombre de membres ni la fréquence d'adhésion au réseau. Ainsi le service d'authentifiation doit faire face à la dynamicité et au passage à l'échelle dans les réseaux Ad Hoc.

4- La mobilité des nœuds Ad Hoc doit aussi être prise en compte pour assurer le service d'authentification. En effet, quand un nœud se déplace dans le réseau, il ne quitte pas nécessairement le réseau et par conséquent ne doit pas être obligé à chaque fois de s'authentifier auprès des autres nœuds. En plus, ce service d'authentification doit être efficace et nécessiter le moins de messages transmis possible.

5- Finalement, le service d'authentification dans les réseaux Ad Hoc est primordial, et s'il est compromis tous les autres services ne pourront plus être assurés (attaques sur les mécanismes de sécurité eux mêmes) [Len02].

Le problème de sécurité s'est amplifié avec l'arrivée des applications multicast dans les réseaux Ad Hoc. Dans la section suivante, nous présentons les problèmes de sécurité pour les communications multicast et en particulier dans le cadre des réseaux Ad Hoc.

1.3 La sécurité dans les communications multipoint

Les communications de groupe dans l'Internet ont suscité beaucoup d'intérêt au cours de cette dernière décennie. Elles correspondent à des modèles appropriés pour des applications comme l'audio/vidéo conférence, la mise à jour de logiciels, la télévision par Internet. L'utilisation de ces applications pour des objectifs commerciaux a orienté de nombreux travaux de recherche sur la nécessité d'offrir des services de sécurité tels que l'authentification, l'intégrité et la confidentialité des données et a donné naissance à un groupe de travail à l'IETF nommé MSEC [2]. Dans le cas des communications de groupe, le potentiel des attaques est beaucoup plus significatif que lors des transmissions point-à-point [Mit97] :

- Les communications de groupe présentent plus d'opportunités pour l'interception de données, car elles mettent en relation plusieurs participants.
- Quand une attaque se produit, un grand nombre de systèmes peut être affecté.
- L'identité et l'adresse du groupe sont connues à large échelle et aident les intrus à diriger leurs attaques.
- Les attaquants peuvent remplacer des membres principaux (membres légitimes du groupe) par d'autres membres illégitimes.

Les communications multipoints dans le cadre des réseaux Ad Hoc présentent de nouveaux challenges liés à la nature dynamique et flexible de ce type de réseaux. En effet, la taille et la dynamicité propres aux groupes multicast peuvent être très importantes dans les réseaux Ad Hoc. En effet, on ne peut pas contrôler le nombre de membres ni la fréquence d'adhésion au groupe. Le service d'authentifiation doit faire face à la dynamicité et au passage à l'échelle des groupes multicast dans les réseaux Ad Hoc.

La mobilité des nœuds Ad Hoc doit aussi être prise en compte pour assurer le service d'authentification des membres du groupe multicast. En effet, quand un membre se déplace dans le réseau, il ne quitte pas nécessairement le groupe multicast et par conséquent ne doit pas être obligé à chaque fois de s'authentifier auprès de la source du groupe auquel il appartient. Il est à noter que, généralement c'est l'arbre multicast qui change et non pas le groupe multicast.

1.4 Plan du rapport

Pour faire face à ces défis, et pour assurer les services de sécurité requis pour le bon fonctionnement d'un groupe multicast dans un environnement Ad Hoc, la solution la plus appropriée est l'établissement d'un protocole de gestion de clé. Ce protocole vise la génération et la distribution de la clé de chiffrement de données à tous les membres du groupe pour assurer la confidentialité des données, et par la même, à authentifier et contrôler l'accès des membres au groupe de sorte que seuls les membres authentifiés et autorisés soient capables d'accéder au flux multicast émis par la source.

2. http://www.securemulticast.org/msec-index.htm

Afin de détailler notre proposition, le rapport est organisé comme suit. Dans le deuxième chapitre sont décrites les différentes approches de gestion de clé de groupe multicast et nous situons notre nouvelle approche par rapport à l'existant. Le troisième chapitre donne une description détaillée de l'approche BALADE. La mise en œuvre et l'implantation de BALADE ainsi que les simulations sont présentées dans le quatrième chapitre. Le cinquième chapitre conclut ce rapport.

Chapitre 2

Gestion de clé et sécurité multicast : un état de l'art

2.1 Introduction

La sécurité d'un groupe multicast requiert que seuls les membres du groupe puissent accéder aux données émises par la source, même si ces données sont diffusées dans tout le réseau. Pour assurer cette confidentialité des données, une clé symétrique est utilisée par la source pour crypter les données et par les membres pour les décrypter. Cette clé est appelée TEK (Traffic Encryption Key). La vie d'une session d'un groupe sécurisé est schématisée par un ensemble d'intervalles de temps ; chaque intervalle est défini par un changement de l'état du groupe, c'est à dire une arrivée ou un départ d'une entité membre du groupe (cf Figure 2.1).

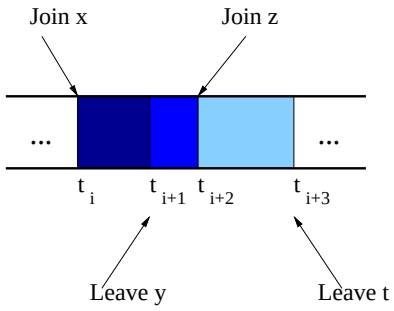

FIG. 2.1 – *Évolution de la vie d'un groupe sécurisé*

Afin de préserver la confidentialité des données du groupe, il est nécessaire de renouveler la clé de chiffrement de données après chaque arrivée (secret passé) ou départ (secret futur) d'un membre du groupe. En effet, le secret futur (*forward secrecy*) consiste à ce qu'un membre ayant quitté le groupe, ne doit plus être capable de décrypter le trafic après son départ. De même, le secret passé (*backward secrecy*) consiste à ce qu'un nouvel utilisateur, membre du groupe, ne doit pas accéder au trafic envoyé avant son arrivée.

Les secrets futur et passé sont réalisés selon la politique de sécurité mise en place. En effet, une politique de sécurité peut exiger une assurance immédiate du secret futur et passé (dès le *Join* ou le *Leave* d'un nœud du groupe) ou différée (périodiquement ou à chaque unité de données selon la sémantique du flux multicast).

2.2 Classification des approches de gestion de clé de groupe

Plusieurs architectures de gestion de clé de groupe dans les réseaux filaires ont été proposées et élaborées, nous pouvons les classifier en trois approches selon le nombre de TEKs utilisés (cf Figure 2.2).

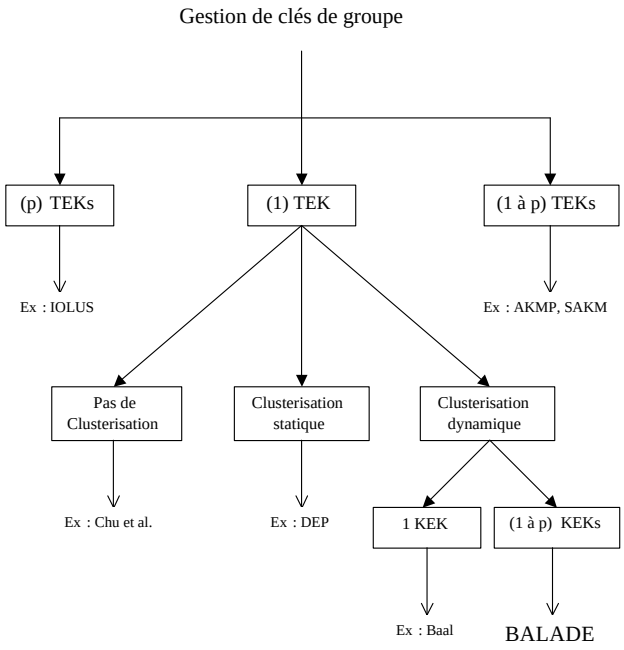

FIG. 2.2 – *Taxonomie des approches de gestion de clé*

1- Approche (p TEK)
Une première approche consiste à subdiviser le groupe multicast en plusieurs sous-groupes. Chaque sous-groupe partage une TEK locale gérée par un contrôleur local ; p étant le nombre de sous-groupes composant le groupe multicast. Cette approche se veut hiérarchique pour remédier au problème *1 affecte n*, qui consiste à ce que le nombre de messages requis pour renouveler la clé du groupe soit égal au nombre des membres du groupe. En effet, lors d'une arrivée ou d'un départ dans le groupe, seul le sous-groupe concerné par ce changement changera sa TEK locale. Ainsi, cette approche est plus robuste au facteur d'échelle, mais en contre partie, elle présente l'inconvénient de requérir des opérations de cryptage/décryptage lors de la transmission de données d'un sous-groupe à un autre. Les protocoles IOLUS [Mit97] et AMAM [SLFC03] appartiennent à cette approche.

2- Approche (1 à p TEKs)
L'inconvénient majeur de l'approche précédente est qu'elle n'est pas flexible par rapport à la dynamicité du groupe, et nécessite un surcoût important en terme de puissance de calcul.
Pour remédier à ces problèmes, l'idée de base de l'approche (1 à p TEKs) pour réaliser la gestion d'un groupe multicast est de diviser le groupe multicast dynamiquement en des sous-groupes, selon des critères de dynamicité. Ainsi, les protocoles appartenant à cette approche commencent par une approche centralisée *1 affecte n* et dynamiquement commutent vers une approche de subdivision du groupe en sous-groupes (p TEK). De cette manière, l'approche réduit le surcoût du processus de cryptage/décryptage tout en atténuant le phénomène *1 affecte n*. Les protocoles AKMP [BBC02] et SAKM [CBB04] appartiennent à cette approche.

3- Approche (1 TEK)
La troisième approche consiste à partager une seule clé de chiffrement de données TEK, utilisée par la source pour crypter les données multicast et par les membres pour les décrypter ; cette approche peut être utilisée au sein d'une architecture centralisée (pas de clusterisation) ou hiérarchique (à clusterisation statique ou dynamique).
L'approche centralisée utilise un seul serveur responsable de la génération, du renouvellement et de la distribution de la clé du groupe. Cette approche présente un inconvénient majeur ; elle ne permet pas le passage à l'échelle car le nombre de messages requis pour renouveler la clé du groupe est égal au nombre des membres du groupe (problème *1 affecte n*). En plus, l'utilisation d'un seul serveur aboutit à un goulot d'étranglement lors de la phase de renouvellement de clé du groupe, et présente un point de vulnérabilité pour les attaques malicieuses. Le protocole Chu et al. [CQN+02] appartient à cette approche.
L'approche hiérarchique utilisant une seule TEK peut être à clusterisation statique ou dynamique.

* Clusterisation statique :
Bien que l'approche (1 à p TEKs) ait atténué le surcoût important dû aux opérations de chiffrement / déchiffrement du flux multicast vu dans l'approche (p

TEKs), ce surcoût demeure très contraignant pour des nœuds à capacité de calcul limitée. DEP (Dual Encryption Key) [DMS99] propose une solution à ce problème. DEP consiste à subdiviser hiérarchiquement et statiquement le groupe multicast en des sous-groupes, géré chacun par un SGM (Sub Group Manager). DEP ne fait pas confiance aux nœuds non membres du groupe (Trusting Third Party Problem), et par conséquent les SGMs qui ne sont pas membres ne doivent pas être capables de déchiffrer le flux multicast émis par la source. Pour cela, DEP gère trois types de clés KEKs (Key Encryption Key) :

- KEK_{i1} : partagée par un SGM_i et ses membres locaux.
- KEK_{i2} : partagée par le contrôleur global (GC) et les membres locaux du sous-groupe i, à l'exception du SGM_i (s'il n'est pas membre du groupe).
- KEK_{i3} : partagée par le GC et les SGM_i.

Pour acheminer la TEK aux membres, le GC transmet un paquet contenant la TEK chiffrée avec KEK_{i2}, puis avec KEK_{i3}. En recevant ce paquet, les SGM_i le déchiffrent en utilisant KEK_{i3}, le ré-enchiffre de nouveau avec KEK_{i1} et le distribuent à tous leurs membres locaux respectifs. Chaque membre du groupe recevant ce paquet, le déchiffre avec KEK_{i1} puis KEK_{i2} et obtient la clé de chiffrement du trafic TEK (cf 2.3).

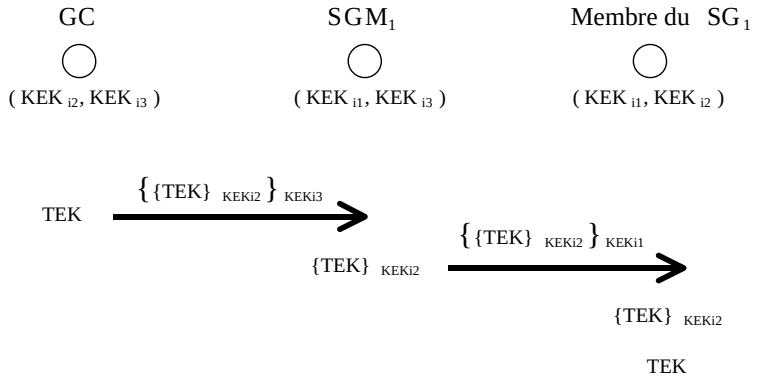

FIG. 2.3 – *Distribution de la TEK dans DEP*

À chaque évènement de *Join* ou de *Leave* au sein d'un sous-groupe, le SGM_i renouvelle la clé KEK_{i1} et ainsi tous les membres ne seront capables d'accéder aux futures TEK que s'ils obtiennent la nouvelle KEK_{i1}. Cependant, le secret futur et passé ne seront plus assurés. En effet, tout nœud ayant quitté le groupe pourrait continuer à déchiffrer le flux multicast jusqu'au prochain renouvellement périodique de la TEK. Pareillement, un nœud joignant le groupe, peut accéder au flux multicast antérieur, chiffrée avec la clé TEK qu'il reçoit à son *Join*.

Le protocole DEP propose ainsi une solution au problème de chiffrement / déchiffrement des données, en contre partie, il a recours à un double chiffrement de la clé TEK et donc utilise plus de KEKs. La clusterisation du groupe multicast dans DEP se fait également de manière statique, ce qui ne peut pas être adapté à la dynamicité et à la mobilité des nœuds.
SEMSOMM [Wei01] est une approche similaire à DEP. Cependant, SEMSOMM utilise la technique de double chiffrement pour chiffrer la totalité des données multicast et non seulement la TEK. Ceci assure le secret futur et passé des données en contre partie d'un coût beaucoup plus important dû à des doubles opérations de chiffrement / déchiffrement des données. De ce fait, SEMSOMM fait partie de l'approche hiérarchique et en hérite ses limites.

* Clusterisation dynamique :
L'approche à clusterisation dynamique utilisant une seule clé de chiffrement de données (1 TEK) peut être divisée en deux sous classes, selon le nombre de clés de chiffrement de clés (KEK) utilisées.
BAAL [CCS00] utilise une seule KEK pour tout les membres du groupe multicast. Ainsi, à chaque évènement dans le groupe correspondant à un *Join* ou un *Leave*, la KEK et la TEK sont renouvelés. La gestion du groupe selon BAAL est à la charge du contrôleur global du groupe, et peut être déléguée aux contrôleurs locaux des sous-groupes. Cependant, BAAL souffre du phénomène *1 affecte n*, et n'assure pas le passage à l'échelle.
BALADE est une nouvelle approche de gestion de clé de groupe multicast, pour des applications de diffusion de flux multimédia, 1 vers n séquentiel, qui propose une amélioration et adaptation de l'approche DEP au contexte des réseaux Ad Hoc. Notre approche BALADE fait partie de l'approche à clusterisation dynamique à une seule TEK et (1 à p KEK), et élimine totalement le surcôut dû aux opérations de cryptage / décryptage des données, tout en atténuant le phénomène *1 affecte n*.

Le tableau de la figure 2.4 résume les différentes caractéristiques des protocoles de gestion de clé pour les communications de groupe, présentés ci-dessus.

	Nombre de TEK	Clusterisation	Nombre de KEK
IOLUS	p	statique	p
A M A M	p	statique	p
AKMP	1 à p	dynamique	1 à p
SAKM	1 à p	dynamique	1 à p
Chu et al.	1	pas de clusterisation	1
DEP	1	statique	p
SEMSOMM	1 à p	statique	p
BAAL	1	dynamique	1
BALADE	1	dynam ique	1 à p

FIG. 2.4 – *Comparaison des protocles présentés*

Chapitre 3

Spécification de BALADE

3.1 Introduction

L'idée de base de BALADE est de subdiviser le groupe multicast dynamiquement. Chaque sous-groupe sera géré et supervisé par un contrôleur local qui partagera avec ses membres locaux une clé de sous-groupe.

Le flux multicast sera chiffré par la source avec la clé TEK, et envoyé en multicast à tous les membres du groupe. La source envoie la clé TEK chiffrée à tous les contrôleurs locaux, chiffrée avec une clé KEK. Ces contrôleurs locaux transmettent alors la TEK à leurs membres locaux, chiffrée avec leurs clés de sous-groupes respectifs. Ainsi, comme dans DEP, l'avantage de BALADE est que seule la clé TEK est chiffrée et déchiffrée et non plus les données du flux multicast. La clé TEK est renouvelée à chaque unité de données émise par la source, c'est à dire selon la sémantique du flux multicast. En plus, BALADE propose une gestion de la dynamicité et de la mobilité des nœuds dans le réseau, adaptée à la nature des réseaux Ad Hoc.

Le modèle de diffusion multicast de cette approche est 1 vers n séquentiel. Selon ce modèle, à tout instant t, il y a une et une seule source qui émet, et une fois qu'elle termine, une autre source prend le relais. L'établissement des sources du groupe et de l'ordre d'émissions des différents flux correspondants à chaque source n'entre pas dans le cadre de ce rapport.

3.2 Motivations

BALADE est une approche de gestion de clé de groupe, adaptée à un environnement Ad Hoc. Les services que se propose de réaliser BALADE sont conformes à ce type d'environnement, tout en assurant la sécurité requise aux communications multicast, 1 vers n

séquentiel. Dans ce qui suit, nous citons les services [BCF04b] que se proposent de réaliser BALADE.

- Confidentialité : seuls les membres du groupe doivent être capables d'accéder au flux multicast.
- Contrôle d'accès : l'accès au groupe est réservé aux membres qui appartiennent à une liste ACL (Access Control List) et qui n'ont pas été exclus du groupe. Le contrôle d'accès est effectué à chaque fois qu'un nœud veut joindre le groupe.
- Authentification : permet à un nœud de s'assurer de l'identité des nœuds avec lesquels il communique. Pour assurer ce service, et n'ayant pas la possibilité d'utiliser une PKI au sein de notre réseau Ad Hoc, la solution que nous allons adopter se base sur les identificateurs cryptographiques [MC02].
- Accessibilité : capacité des récepteurs à accéder au service de réception du flux multicast depuis n'importe quel point du flux
- Dynamicité : les membres du groupe peuvent à tout moment joindre ou quitter le groupe. En plus, selon la mobilité du groupe, il est possible que plusieurs nœuds se déplacent en même temps et veulent joindre le groupe au même instant. La solution proposée doit donc s'adapter à cette forte dynamicité.
- Stockage des données minimal : le nombre de paquets maximum que la source ou les récepteurs doivent stocker.
- Délai d'accessibilité minimal : le nombre maximum de messages qu'un nouveau membre doit échanger pour accéder au flux multicast.
- Coûts minimaux en terme de puissance de calcul et de bande passante.
- Gestion de la mobilité : lorsqu'une source quitte le groupe, lorsqu'un contrôleur quitte le groupe ou losqu'un membre quitte le groupe et le rejoint.
- Passage à l'échelle : en terme de coût de communication, de calcul, de stockage et de bande passante.

3.3 Contraintes et choix

Vu les caractéristiques des réseaux Ad Hoc, plusieurs contraintes sont à prendre en compte lors de la réalisation d'une architecture de sécurité. Dans ce qui suit, nous traitons les problèmes de choix du protocole de routage multicast à utiliser (source spécifique ou arbre partagé), de la méthode d'authentification à mettre en œuvre (à cause de l'absence d'infrastructure fixe), et de la nature des contrôleurs locaux (pour assurer la confidentialité des données et pour faire face aux ressources limitées des nœuds).

1- Le choix du protocole de routage multicast adéquat est très important pour notre approche. En effet, utiliser un protocole basé source spécifique (comme MOLSR [LJM+03]) implique n arbres multicast construits autour des n sources séquentielles. Par contre, un protocole multicast basé arbre partagé (comme MAODV [RP00]) construit un seul

arbre multicast pour toutes les sources. Et en plus du surcoût engendré par la création d'un nouvel arbre multicast pour chaque source dans le cas d'un protocole basé source spécifique, le temps de latence entre la diffusion de deux sources consécutives peut être très important. Pour ces raisons, nous avons choisi de bâtir BALADE en utilisant le protocole de routage arbre partagé MAODV.

2- Pour assurer l'authentification des membres et des sources dans le cadre de notre approche, nous avons choisi d'utiliser les identificateurs cryptographiques [MC02, BCK96]. Les identifiants cryptographiques sont statistiquement uniques et cryptographiquement vérifiables (SUCV), ce qui signifie que de par leur nature, il est très peu probable que deux entités aient le même identifiant, et qu'il est possible de vérifier la validité de l'identifiant présenté par une entité grâce à des techniques cryptographiques.
L'identité cryptographique notée CBID (*Crypto Based IDentifier*) est définie par :

$$\text{CBID} = \text{hmac\_sha1\_128}(\text{sha1}(\text{imprint}),\text{sha1}(\text{PK}))$$

Où : PK est la clé publique du créateur de l'identifiant et inprint est un entier à 64 bits qui peut être une quantité dépendante de la location du nœud (préfixe du réseau local) ou simplement une valeur aléatoire.
L'idée de base de ces identificateurs est d'avoir une forte liaison cryptographique avec leurs composants (clés privée et publique). C'est exactement le but des certificats.
Cette technique d'authentification ne doit pas compter sur une tierce partie, à savoir une PKI globale, ou un serveur central de distribution de clés. Cette contrainte implique que deux entités qui ne se connaissent pas ne peuvent pas communiquer. L'authentification d'un nouveau nœud n'est donc pas possible. Seuls les nœuds qui se connaissent au préalable peuvent s'identifier et communiquer.
Dans le cadre de notre approche, l'identification et le contrôle d'accès se font grâce à une liste ACL qui contient tous les CBIDs des nœuds autorisés à joindre le groupe.

3- Les contrôleurs locaux, supervisant tous les sous-groupes du groupe multicast, peuvent accéder à la clé de chiffrement de données TEK. C'est pour cette raison qu'un contrôleur local doit impérativement être un membre du groupe. De plus, tous les contrôleurs locaux sont des nœuds Ad Hoc et non des routeurs comme dans le cas des réseaux filaires, et il n'est pas raisonnable qu'un nœud Ad Hoc non adhérent au groupe se propose volontaire pour assurer le rôle d'un contrôleur local. Un contrôleur local doit obtenir la permission de son contrôleur local parent pour former et gérer son propre sous-groupe. C'est donc le contrôleur local parent qui vérifie l'authenticité du contrôleur local en question et son appartenance au groupe multicast.

3.4 Description de la solution

BALADE est une architecture de sécurisation de diffusion multicast dans un environnement Ad Hoc. Dans ce qui suit, nous décrivons les différentes opérations de gestion et de sécurité qu'elle réalise.

3.4.1 Gestion des membres du groupe : clusterisation dynamique

Les acteurs principaux de l'architecture BALADE sont :

- Contrôleur Global (CG) : à chaque source du groupe est associé un contrôleur global. Ainsi, et avec une architecture de diffusion séquentielle 1 vers n, à un instant donné il existe un seul CG au sein du groupe.
  Cette entité est responsable de la génération de la clé de chiffrement (TEK), du chiffrement des données et de la distribution des données chiffrées à tous les membres du groupe. La clé de chiffrement du trafic sera distribuée séparément via les contrôleurs locaux. Le CG assure aussi le renouvellement de la clé de chiffrement de trafic (TEK) à chaque unité de données, selon la sémantique du flux.
  Tous les contrôleurs (global et locaux) détiennent les mêmes listes ACL et RL (liste de contrôle d'accès et liste des membres banis). Ces listes seront utilisés lors du contrôle d'accès d'un nouveau membre au groupe. La génération et la cohérence des ces listes n'entre pas dans le cadre de ce rapport. Un projet en cours, au sein de l'équipe MADYNE, traite de ce problème.
- Contrôleur Local (CL) : tout nœud mobile membre du groupe, formant avec ses membres locaux un sous-groupe ou cluster doit générer et distribuer une clé à ses différents membres locaux. Cette clé est appelée CSG : clé du sous-groupe. Chaque CL détient sa liste de membres locaux (LPL). Il doit acheminer la clé de chiffrement de trafic, envoyée par la source, à tous les membres de cette liste.
  Un simple membre du goupe décide de passer à l'état CL si le taux de dynamicité locale (nombre de *Join* et de *Leave* par unité de temps) et le nombre de membres locaux dépassent un certain seuil.
  Les données qu'un CL actif doit transmettre à ses membres locaux sont la clé de chiffrement du trafic, émise par la source du flux. Le flux chiffré étant diffusé séparément.
  Pour décider de son état, chaque membre détient une fonction d'évaluation [BCF04a], implantant l'algorithme 1.

Algorithm 1 Fonction d'évaluation

if (mcf > d1 or mn > d2) **then**
  //switch to dec/rec process
  f_i = true;
else
  f_i = false;
end if
//avec mcf : fréquence de dynamicité , d1 : seuil de fréquence de dynamicité, mn : nombre
//de membres locaux et d2 : seuil de nombre de membres locaux

Cette fonction d'évaluation ne prend en compte que la fréquence d'adhésion au groupe et le nombre de membres locaux. Une étude en cours consiste à améliorer cette fonction d'évaluation en introduisant le facteur de mobilité.
Pour obtenir la valeur mn, un CL compte ses membres locaux passifs avec leurs membres fils, et ses membres locaux actifs sans leurs membres fils. Le schéma de la Figure 3.1 illustre un exemple de calcul de mn.

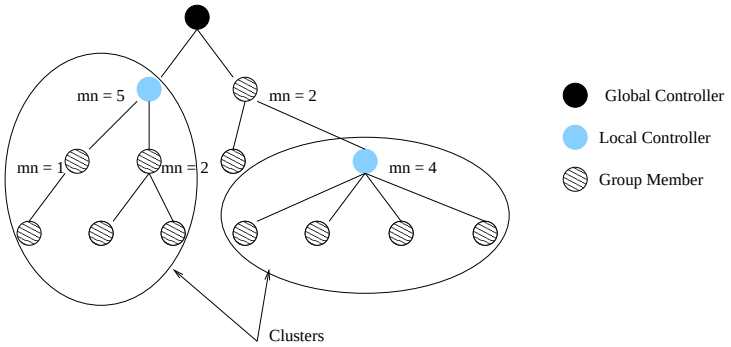

FIG. 3.1 – *Exemple d'évaluation de mn*

Il est à noter qu'un membre doit demander l'autorisation auprès de son CL parent pour pouvoir passer à l'état CL. Le CL parent authentifie le membre en question, s'assure qu'il est bien un membre du groupe (pour assurer la confidentialité de la TEK) et en cas de succès, l'autorise à joindre le groupe des contrôleurs locaux et lui envoie les listes ACL et RL.

– Membre du groupe (GM) : est un membre de la liste ACL (Access Control List).

3.4.2 Diffusion des données sécurisées

La source chiffre les données avec une clé TEK (Traffic Encryption Key). L'algorithme de cryptage qu'on va adopter est AES [Fed01, SKW+99] (Authenticated Encryption Scheme).

La source achemine les données sécurisées via l'arbre multicast de diffusion de données. Les récepteurs ayant reçu la clé de chiffrement TEK par le processus de distribution de clés (présenté ci-dessous), pourront déchiffrer le flux.

3.4.3 Gestion et distribution des clés

Comme présenté ci-dessus, la source commence par chiffrer les données avec la clé TEK puis les envoie à tous les membres du groupe suivant l'arbre de diffusion de données.

À l'initialisation de l'application, tous les membres du groupe reçoivent de la part de la source la clé de session qu'on va appeler CSG_0 (clé du sous-groupe 0), puis dynamiquement,

des nouveaux sous-groupes vont se créer. Chaque sous-groupe i aura un contrôleur local CL_i et partagera une clé de sous-groupe CSG_i.

Pour envoyer la clé TEK à tous les membres du groupe, la source chiffre cette clé avec CSG_0 et l'envoie à tous les membres de son sous-groupe. Puis elle envoie cette clé TEK au groupe formé par les contrôleurs locaux (ce groupe partage une clé de groupe appelée CCL : clé des contrôleurs locaux), chiffrée avec la CCL.

Les contrôleurs locaux appartenant à ce sous-groupe vont décrypter le message, extraire TEK, la réencrypter avec leurs clés de sous-groupe respectifs et l'envoyer à tous leurs membres locaux.

L'avantage de cette solution est de faciliter le processus de cryptage/décryptage aux contrôleurs locaux, qui n'ont qu'à crypter et décrypter la clé de chiffrement et non plus tout le flux multicast.

On note que chaque nouvelle source doit joindre le groupe des contrôleurs locaux, pour pouvoir envoyer la clé de chiffrement du trafic chiffrée avec la CCL à tous les autres CLs. Lorsqu'on passe d'une source à une autre séquentiellement, l'arbre de distribution de clés reste inchangé. Une source, à la fin de sa diffusion des données, doit vérifier si elle a des membres fils qui ne sont pas des CLs. Si elle n'en a pas, elle doit quitter le groupe des CLs.

Une illustration de la distribution de TEK est présentée dans la figure 3.2.

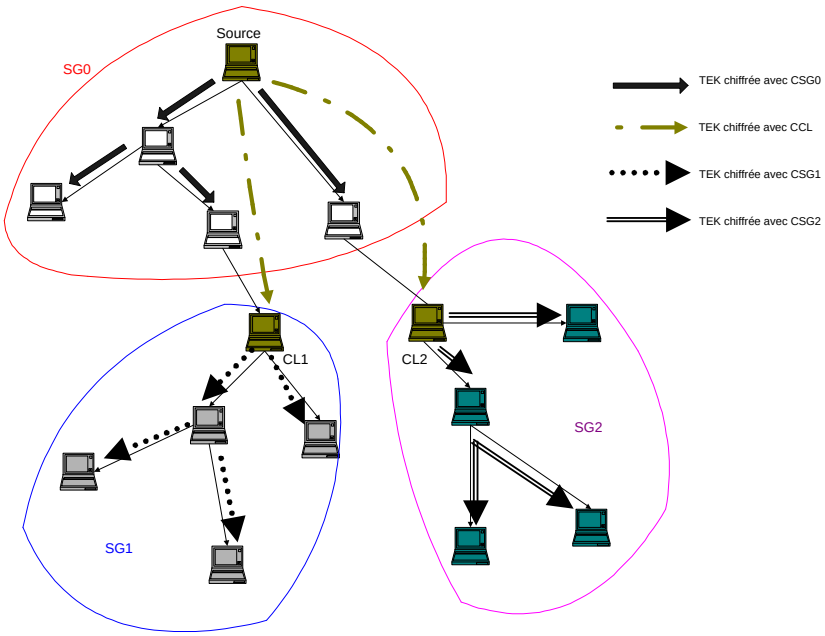

FIG. 3.2 – *Distribution de TEK*

3.4.4 Initialisation de BALADE

À l'initialisation de l'application, l'authentification et le contrôle d'accès sont réalisés par la première source du groupe. Ensuite, dynamiquement, la source délègue cette tâche aux CLs qui doivent être capables d'autoriser ou de refuser un *Join* d'un nouveau membre au groupe.

L'intialisation de la liste de contrôle d'accès se fait avant le lancement de BALADE (*off-line*) et contient tous les CBIDs des nœuds pouvant joindre le groupe dès son initialisation ou ultérieurement.

L'établissement de la liste des sources du groupe est à la charge d'une application[1] externe qui s'exécute à l'initialisation de l'application en question.

Le contrôleur global CG, qui initialement est la première source du groupe, initialise les deux listes LPL et RL, LPL contiendra tous les membres du groupe appartenant à l'arbre multicast, ayant joint le groupe off-line. RL sera initialement vide. Tous les membres initialisent à leur tour leurs listes LPL à vide.

Le CG entame ensuite la phase de distribution de la clé du groupe à tous les membres n_i de la liste LPL, il procède alors comme suit :

CG $->n_i : \{TEK, CSG_0, IDG, ID_{CG}, CBID-CG\}`Pub\_n_i$

avec TEK : Clé de chiffrement de données, SG_0 : clé du sous-groupe 0 géré par la source, IDG : identité du groupe, ID_{CG} : identité du CG, CBID-CG : CBID du CG.

3.4.5 Ajout d'une nouvelle entité

Tout nœud n_i doit s'authentifier auprès de la source ou d'un contrôleur local. Si l'authentification et le contrôle d'accès réussissent, il pourra joindre le groupe. L'authentification et le contrôle d'accès sont détaillés dans le paragraphe suivant.

Pour cela, chaque nœud n_i génère et détient une paire de clé publique et privée (Pub$_n_i$, Pri$_n_i$). À partir de cette paire de clés, chaque nœud calcule son identificateur cryptographique unique (CBID) qui lui sera nécesssaire pour s'authentifier auprès de la source.

La source et tous les contrôleurs locaux (CLs) détiennent une ACL (Access Control List) contenant tous les CBIDs des nœuds autorisés à joindre le groupe. Ils détiennent aussi une liste de révocation contenant tous les CBIDs des nœuds exclus du groupe et ne pouvant plus le joindre. Un CL autorise ou non un nœud à joindre le groupe selon ces deux listes.

À ce stade, et en cas de succès de l'authentification et du contrôle d'accès du nouveau membre, le membre parent UP_i calcule sa fonction d'évaluation, deux cas se présentent :

- si $f_i = true$, le membre passe à l'état CL. Il doit donc générer une nouvelle clé et la distribuer à ses membres locaux. Pour cela, il envoie la nouvelle clé locale CSG_i en multicast à tous les anciens membres locaux, cryptée avec l'ancienne clé old$_CSG_i$, et envoie la même clé CSG_i au nouvel abonné, cryptée avec sa clé publique.

1. par exemple un application "peer to peer" qui se charge de créer la liste des sources du groupe et de la diffuser à tous les membres du groupe

En plus, si le membre vient de passer à l'état CL, et donc son ancienne clé locale est égale à la clé de son nœud parent CL_{parent}, il doit envoyer son ancienne clé locale à son CL pour que ce nœud change sa clé locale pour tous ses membres locaux (c'est à dire au cas où old_$CSG_i = CSG_{parent}$).
On note qu'un membre ne peut passer à l'état CL que s'il reçoit l'autorisation de son contrôleur local pour le faire. Ce dernier doit l'authentifier, et vérifier qu'il est bien un membre du groupe. Au cas où l'authentification réussit, le contrôleur actif lui envoie la liste de contrôle d'accès ACL, et la clé du groupe des contrôleurs locaux (CCL), cryptés avec sa clé publique.
Le CL procèdera donc de la façon suivante :

for j: 1..nb_old_attached_members
CL $->n_j : \{CSG_i\}`old\_CSG_i$

CL $->n_i : \{CSG_i\}`Pub\_n_i$ avec n_i : nouvel abonné

CL $->CL_{parent} : \{old\_CSG_i\}`Pub\_CL_{parent}$
avec Pub_CL_{parent} : clé publique du nœud parent CL_{parent}.

- si $f_i = false$, le membre parent reste à l'état passif, et donc il doit envoyer une demande de renouvellement de clé à son CL, qui entame la génération d'une nouvelle clé et sa distribution à tous ses membres locaux.

3.4.6 Authentification et contrôle d'accès

Pour l'authentification et le contrôle d'accès, nous distinguons deux cas : authentifier et contrôler l'accès d'un nouveau membre qui est à la charge du contrôleur local du sous-groupe en question, et authentifier et contrôler l'accès à un membre du groupe qui a perdu son lien avec l'arbre multicast à cause de sa mobilité ou d'un problème de ressources.

Authentification et contrôle d'accès d'un nouveau membre du groupe

La figure 3.3 illustre un exemple d'authentification d'un nouveau membre au groupe.

Le nouveau nœud demande à un membre du groupe de joindre le groupe multicast. Pour cela, il lui envoie un message de demande d'adhésion au groupe, contenant son CBID. Le membre du groupe ne pouvant pas contrôler son accès au groupe, envoie un message de vérification de contrôle d'accès à son contrôleur local. Le contrôleur local qui détient la liste ACL, peut accepter ou non la demande d'adhésion du nouveau nœud. Au cas où l'authentification et le contrôle d'accès réussissent, le contrôleur local renvoie un message d'acceptation d'adhésion au groupe au membre parent, qui se charge d'activer la route de l'arbre multicast vers le nouveau nœud. Le membre parent envoie en plus, un message d'acceptation d'adhésion au nouveau nœud, contenant un mot de passe chiffré avec la clé de

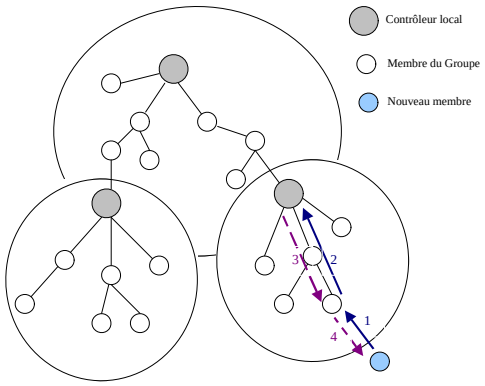

FIG. 3.3 – *Authentification et contrôle d'accès d'un nouveau membre au groupe*

chiffrement de données TEK. Ce mot de passe qu'on va appeler ticket, sera utilisé lors de la ré-authentification du nouveau membre (cf. section suivante).

Authentification et contrôle d'accès d'un membre du groupe

La figure 3.4 illustre un exemple d'authentification d'un ancien membre au groupe.

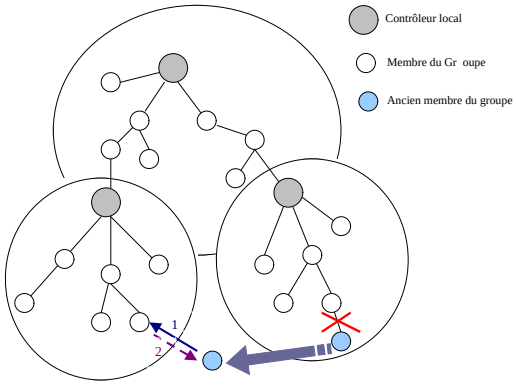

FIG. 3.4 – *Authentification et contrôle d'accès d'un ancien membre du groupe*

Ce cas d'utilisation est très probable dans le cadre des réseaux Ad Hoc et correspond à un nœud qui se déplace d'un sous-groupe à un autre, ou bien qui s'éteint brusquement à cause d'un problème de batterie. Le nœud en question demande à un membre du groupe de l'au-

thentifier et de contrôler son accès. Pour cela, il lui envoie un message de demande d'adhésion au groupe contenant son ticket qu'il avait reçu lors de sa première authentification.

Le membre de l'arbre pourra ainsi déchiffrer le ticket (mot de passe) avec la clé TEK, vérifier si le nœud était bien un membre du groupe et en cas de succès accepter la demande d'adhésion du nouveau nœud.

On note que le ticket est commun à tous les membres du groupe, et est chiffré avec la clé de chiffrement de données. Ainsi, ce ticket n'est plus valable après un renouvellement de la TEK. Ceci est nécessaire et efficace dans le cas d'une révocation d'un membre du groupe qui ne doit pas être capable de joindre le groupe de nouveau, depuis n'importe quel cluster.

Pour évaluer le gain en nombre de messages envoyés, engendré par l'authentification d'un ancien membre du groupe par rapport à celle d'un nouveau membre, on considère :

- n : nombre de membres du groupe.
- c : nombre moyen de membres par sous-groupe.

Le nombre de messages nécessaires pour l'authentification d'un nouveau membre est au minimum 2 messages, au maximum $2*c$ messages ; soit en moyenne $c+1$ messages nécessaires. Cependant, le nombre de messages nécessaires pour l'authentification d'un ancien membre du groupe est 2 messages.

3.4.7 Retrait d'une entité du groupe

Nous distinguons deux cas : retrait volontaire et retrait obligatoire ou expulsion. Le premier cas est réalisé quand un membre veut quitter le groupe et envoie un message *Leave* incluant son CBID, à son contrôleur local dans le but de stopper le flux de trafic multicast du groupe. Dans ce cas, le CL supprime ce membre de sa liste de membres locaux LPL, et entame une phase de renouvellement de clé.

Le deuxième cas (expulsion) est connu sous le nom de révocation du membre, et a lieu quand un membre peut mettre la sécurité du groupe en péril. Dans ce cas, le contrôleur local ajoute le nom de ce membre à la liste de révocation RL.

Le membre parent UP du nœud quittant le groupe le supprime de sa liste de membres locaux LPL. Ensuite il entame une phase de renouvellement de clé.

La phase de renouvellement de clé commence par évaluer la fonction d'état, afin de décider du nouvel état que doit adopter le membre parent UP_i du nœud quittant le groupe. Comme dans le cas de l'ajout d'une entité au groupe, deux cas se présentent :

- si $f_i = true$, le membre passe à l'état CL. Il doit donc générer une nouvelle clé et la distribuer à ses membres locaux.
  Pour cela, il envoie la nouvelle clé locale CSG_i en unicast à tous les anciens membres locaux, en excluant le membre qui a quitté le groupe, cryptée avec leurs clés publiques respectives.
  En plus, dans le cas où le membre vient de passer à l'état CL, et donc sa clé locale ancienne est égale à la clé de son nœud parent CL_{parent}, il doit envoyer sa clé locale

ancienne à son CL pour que ce nœud change sa clé locale pour tous ses membres locaux (c'est à dire au cas où old_$CSG_i = CSG_{parent}$).

n_s est le membre qui a quitté le groupe
for j: 1..nb_old_attached_members, j différent de s
CL $->n_j : \{CSG_i\}`Pub\_n_j$

CL $->CL_{parent} : \{old\_CSG_i\}`Pub\_CL_{parent}$
avec Pub_CL_{parent} : clé publique du nœud parent CL_{parent}.

- si $f_i = false$, le membre parent reste à l'état passif, et donc il doit envoyer une demande de renouvellement de clé à son CL, qui entame la génération d'une nouvelle clé et sa distribution à tous ses fils, en excluant le membre qui a quitté le groupe.

Le cas de révocation d'un membre du groupe nécessite, en plus du renouvellement de la clé du sous-groupe, le renouvellement de la clé de chiffrement de données TEK. Pour cela, le contrôleur local du sous-groupe en question envoie un message à la source lui demandant de changer la TEK et de la distribuer à tous les membres du groupe.

3.4.8 Renouvellement des clés

Pour assurer la confidentialité des données, un processus de changement de la TEK doit être lancé côté source, selon plusieurs méthodes :

- La source change la TEK à chaque bloc fixe de données :
  Si le bloc est assez petit, le secret futur est vérifié dès le *Leave* d'un nœud, à un coût très contraignant côté source. Par contre, si le bloc de données est grand, le surcoût est nettement moins contraignant avec l'inconvénient qu'un nœud peut continuer à déchiffrer le bloc de données courant même lorsqu'il est expulsé du groupe.
- La source change la TEK à chaque unité sémantique de données dépendant de l'application en question :
  Cette solution est beaucoup plus adaptée au type de l'application en question. Ainsi, une source qui diffuse un flux MP3 va changer la TEK à chaque titre MP3, alors qu'une source qui diffuse un flux vidéo va renouveller sa TEK à chaque film ou à chaque chapitre d'un film. Cette solution est plus réaliste du fait qu'elle tient compte du côté pratique de l'application. C'est pour cette raison que nous l'avons adopté dans BALADE.

Le renouvellement de la TEK se fait de la façon suivante :

La source (CG) génère une nouvelle TEK et la distribue à tous les membres locaux de son sous-groupe, chiffrée avec la clé locale du sous-groupe :
$\forall n_i \in LPL\_CG$:
CG $->n_i : \{TEK\}`CSG_{CG}$

La source (CG) envoie la nouvelle TEK au groupe de CLs, chiffrée avec la CCL :
$\forall CL_i \in GCL$:
CG $->CL_i : \{TEK\}`CCL$

Tout CL_i envoie la nouvelle TEK à ses membres locaux, chiffrée avec la clé CSG_i :
$\forall CL_i \in GCL, \forall n_j \in LPL\_CL_i$:
$CL_i ->nj : \{TEK\}`CSG_{CL_i}$

Les clés des sous-groupes servant à chiffrer la TEK doivent aussi être changées périodiquement, à la charge des contrôleurs locaux des sous-groupes.

3.5 Conclusion

Dans ce chapitre, nous avons présenté les différents services de gestion de clé que se propose de réaliser BALADE. Nous avons commencé par décrire la clusterisation dynamique du groupe multicast. Puis nous avons présenté la gestion et distribution de clés ainsi que la diffusion des données sécurisées selon BALADE. Enfin, nous avons détaillé les opérations d'authentification et d'ajout d'un nouveau membre au groupe, de retrait d'un membre du groupe et de renouvellement des clés mise en œuvre dans BALADE.

Les états que peut avoir un membre du groupe multicast (simple feuille de l'arbre, membre ayant des fils, CL ou CG) et les traitements à sa charge sont résumés dans l'annexe.

Dans le chapitre suivant, nous présenterons la mise en œuvre de BALADE dans le cadre d'une application spécifique que nous détaillerons.

Chapitre 4

Mise en œuvre de BALADE

4.1 Introduction

Dans ce chapitre, nous présentons l'application cible que nous avons choisie pour déployer BALADE. Ensuite, nous décrivons les implantations des opérations de *Join* et de *Leave*. Finalement, et afin de comparer BALADE avec les autres approches de gestion de clé de groupe, nous présenterons les résultats des simulations réalisées.

4.2 Application cible

Le cadre de déploiement de BALADE est celui des communications de groupes multicast, intéractives et coopératives, 1 vers n séquentiel. Ce contexte consiste à avoir plusieurs sources, au sein d'un groupe multicast, qui prennent le relais séquentiellement pour diffuser un flux de données. Ainsi, à un instant donnée, une et une seule source émet au sein du groupe.

Plusieurs applications cibles peuvent être utilisées pour mettre en œuvre BALADE. nous pouvons citer : conférences multimédia (audio / vidéo / tableau blanc partagé), ressources synchronisées (SGBB partagés), édition de fichier partagé, enseignement à distance, traitement parallèle distribué, ... Dans ce qui suit, nous détaillons l'application cible de BALADE que nous avons retenu dans MADYNE : une application jukebox.

La première application cible de cette architecture est un ensemble de nœuds Ad Hoc, voulant réaliser une application de diffusion multicast de titres MP3 (JukeBox). Chaque nœud ayant une liste de titres MP3 la déclare au sein du réseau ; une file d'attente de titres MP3 est ainsi distribuée à tous les nœuds. Plusieurs nœuds du réseau peuvent jouer le rôle de source du groupe multicast, nous avons donc une émission multicast 1 vers n séquentiel. L'émission des titres MP3 doit être sécurisée, ainsi seuls les membres du groupe seront capables de déchiffrer les données. Cette première architecture est schématisée à la figure 4.1.

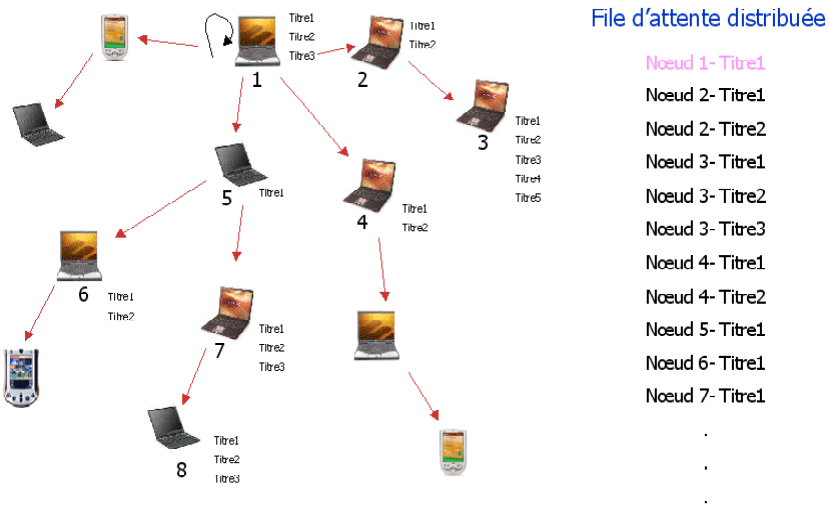

FIG. 4.1 – *Jukebox dans un réseau Ad Hoc*

On peut adapter l'application JukeBox à une approche hybride de réseaux Ad Hoc connectés à une passerelle (ou point d'accès) reliée au monde Internet filaire. L'architecture fonctionnelle dont nous disposons au sein de notre équipe est schématisée à la figure 4.2. La source du groupe peut être dans le réseau Ad Hoc (S1 ou S2) ou dans le réseau filaire (S3).

Pour l'implantation et les tests de BALADE, nous avons choisi de mettre en place cette application jukebox. Pour cela, un module jukebox est réalisé. Il s'agit d'une application de diffusion des morceaux MP3 en multicast; elle possède trois composants : une partie Client permettant la réception du flux multicast et son décodage en utilisant un lecteur MP3 comme xmms ou winamp ; une partie Serveur responsable de la diffusion des morceaux MP3 pour un group multicast et une partie Manager qui s'appuie sur des interfaces pour communiquer avec d'autres modules comme celui de la sécurité (BALADE) pour récuperer la clé courante du groupe lors d'une session multicast. Pour assurer une correction des pertes de paquets, nous utilisons la technique FEC (Forward Error Correction) [Riz97]. Des services de contrôle de flux sont aussi mis en place grâce aux protocoles RTP (Real-time Transfer Protocol) [SCFJ96] et RTCP (Real-time Transfer Control Protocol) [SCFJ96].

4.3 Implantation de BALADE

Le module BALADE à réaliser supervise et sécurise toutes les diffusions multicast de l'application jukebox (voir section 4.2). Le module BALADE est constitué de deux compo-

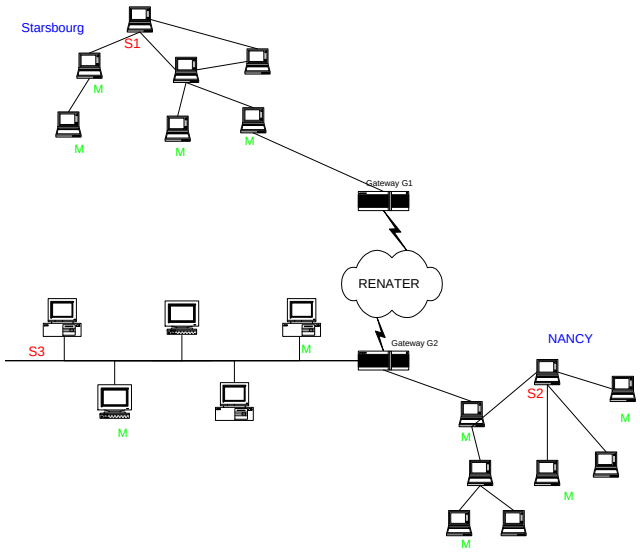

FIG. 4.2 – *Jukebox dans un réseau hybride*

sants : un composant du côté client qui gère le comportement d'un nœud joignant le groupe ; et un composant du côté serveur qui gère le comportement des sources et des contrôleurs de groupe. L'implantation de BALADE est réalisée sous Linux version Mandrake 9.2 Noyau 2.4.x. Le protocole de routage multicast utilisé est MAODV v6.

4.3.1 Protocole relatif à l'adhésion d'un membre au groupe

Lors de l'adhésion à un groupe multicast, une application multicast sollicite une demande d'adhésion en spécifiant l'adresse de groupe au module de gestion des groupes multicast implémenté dans le noyau Linux.

Ce module envoie un message *REPORT* en utilisant le protocole *IGMP* dans le cas d'une adresse de groupe IPv4 ou via le protocole *MLD* pour une adresse de groupe IPv6. Cependant, le protocole *MAODV* n'utilise pas ces protocoles pour effectuer la demande d'adhésion mais c'est le message *RREQ* qui permet de faire cette demande.

Pour assurer l'authentification des nœuds, nos données d'authentification sont encapsulées directement dans le message *RREQ*.

Afin d'intercepter les messages d'adhésion aux groupes, BALADE utilise l'interface netfilter [KWM+]. Cette interface offre des *hooks* (points d'accochage) d'interception de messages depuis l'espace noyau vers l'espace utilisateur où tourne notre démon de sécurité BALADE, qui redirige les paquets vers un code défini par l'utilisateur pour être examiné, supprimé,

ignoré, modifié ou mis en attente par un démon utilisateur. Netfilter est ainsi utilisé pour identifier les évènements qui déclenchent les actions associées au protocole MAODV.

L'avantage de cette solution est qu'elle est portable et facile à installer et ne nécessite pas des modifications dans le noyau Linux. En effet, installer un module dans le noyau Linux est plus facile que de le modifier. En plus, un module peut être installé et désinstallé facilement et à tout moment.

L'architecture Netfilter définit 4 *hook*s (Figure 4.3). Nous trouvons deux *hook*s principaux qui sont NF_IP_LOCAL_IN et NF_IP_LOCAL_OUT, ces *hook*s traitent les paquets entrant et sortant des processus utilisateurs. D'autres *hook*s comme NF_IP_PRE_ROUTING et NF_IP_POST_ROUTING traitent les paquets envoyés et reçus d'autres machines sur le réseau. Le démon BALADE utilise le deux premières *hook*s (NF_IP_LOCAL_IN et NF_IP_LOCAL_OUT) pour intercepter les messages du démon MAODV.

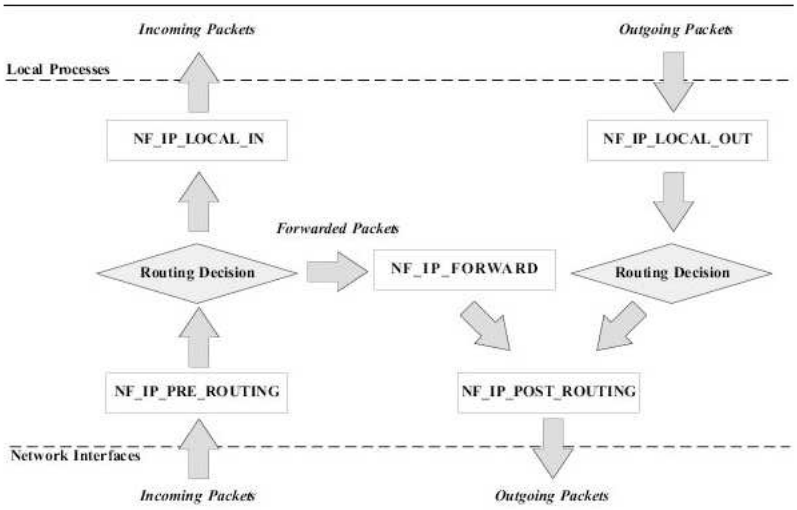

FIG. 4.3 – *Architecture de Netfilter*

Des travaux ont été faits pour sécuriser le protocole MAODV. [Zap02] se base sur les signatures numériques avec des clés privées des émetteurs pour assurer l'authentification et l'intégrité des messages MAODV.

Cependant, notre approche se base sur les CBIDs qui seront rajoutés respectivement dans les messages MAODV *RREQ* et *MACT* pour assurer un service d'authentification des nœuds participant à la construction de l'arbre multicast.

Ces données d'authentification seront ajoutées sous forme d'une extension (Figure 4.4) dans les messages de MAODV.

Cette extension comporte 4 champs :

- Type : ce champ spécifie le type d'authentification à effectuer. Nous avons défini 2 types d'authentification qui sont les suivants : authentification d'un nouveau membre et authentification d'un ancien membre.
- Length : c'est la taille de données d'authentification
- Multicast Group Address : c'est l'adresse du groupe multicast à joindre
- Authentication Data : Ce champ contient les données d'authentification. Dans le cas d'une authentification d'un nouveau membre, ce champ contient le CBID du nouveau membre. Il contient un ticket d'authentification d'un ancien membre du groupe dans le cas d'une authentification d'un ancien membre.

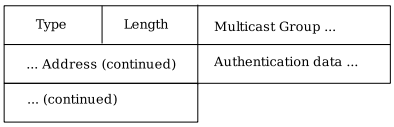

FIG. 4.4 – *L'extension d'authentification de BALADE dans les messages MADOV*

Dans la suite, nous présentons les étapes nécessaires pour authentifier un nouveau membre d'un groupe multicast et les nœuds intermédiaires participants à son adhésion.

La figure 4.5 montre le comportement d'un nouveau nœud qui veut joindre le groupe.

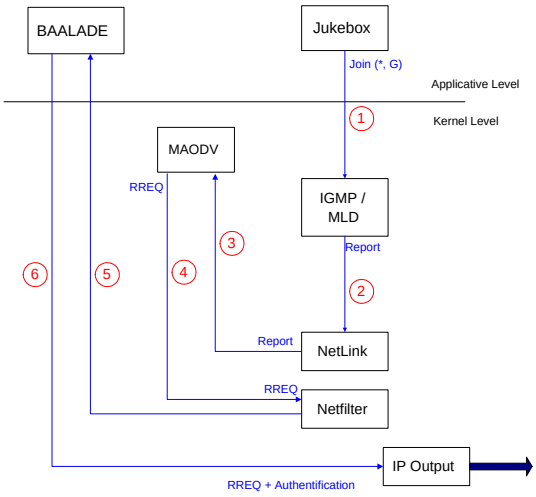

FIG. 4.5 – *Join côté nouveau membre*

(1) L'application Jukebox envoie un message *Join* (*, G) pour joindre le groupe de diffusion multicast des titres MP3.

(2) Le module IGMP/MLD dans le noyau génère un message *REPORT* pour demander l'adhésion à ce groupe multicast.

(3-4) Le démon MAODV intercepte le *REPORT* depuis le module Netlink, génère et envoie un message *RREQ* avec le flag *Join* mis à 1 pour spécifier qu'il s'agit d'une découverte de route multicast.

(5) Après réception des messages *RREP* des nœuds membres du groupe ou possédant une route vers ce groupe, le démon MAODV choisit son nœud parent et lui envoie un message *MACT*.

(6) Le démon BALADE intercepte le message *MACT* et rajoute les données d'authentification du membre. Seul le message *MACT* avec le flag *Join* mis à 1 sera intercepté par BALADE.

Dans ce qui suit, nous décrivons le comportement des nœuds parents qui reçoivent la demande de *Join* du nouveau nœud (Figure 4.6).

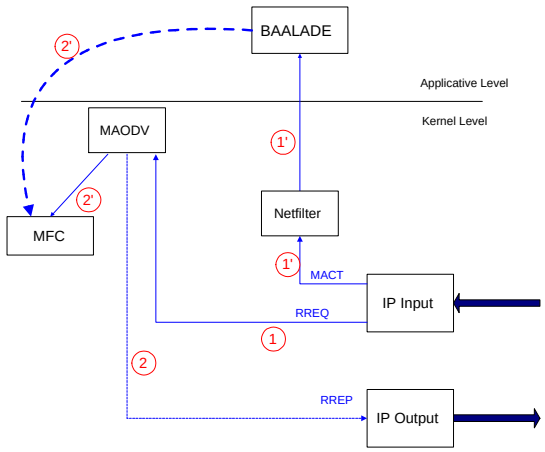

FIG. 4.6 – *Join côté Nœud Parent*

(1) Le nœud parent reçoit un message *RREQ* d'un nœud voulant joindre le groupe.

(1') Le nœud parent reçoit un message *MACT* authentifié provenant d'un nœud lui confirmant qu'il est son son nœud parent dans l'arbre multicast. Le message *MACT* contient le CBID du nouveau membre de groupe et le flag *Join* est mis a 1.

(2) Si le nœud parent n'est pas un membre du groupe ou ne connaît pas de route vers ce groupe, il achemine le message *RREQ*, sinon il répond avec un message *RREP*.

(2') À la réception d'un message MACT, le nœud parent attache le nouveau membre à l'arbre multicast du groupe et il ajoute une entrée dans la *MFC (Multicast Forwarding Cache)* afin de lui acheminer le flux multicast. Une copie de ce message sera interceptée par le démon BALADE qui essayera d'authentifier et de contrôler l'accès de ce membre, localement si le nœud est un contrôleur local, sinon il contactera son CL en lui demandant de contrôler l'accès du nouveau membre. En cas de réussite de ces deux opérations, le démon BALADE envoie un message unicast au nouveau membre contenant la clé du groupe. En cas d'échec, le démon BALADE supprime l'entrée du membre dans la *MFC*.

4.3.2 Protocole relatif au Leave d'un membre du groupe

Dans ce cas, nous nous intéressons aux membres feuilles de l'arbre multicast. Seuls ces nœuds peuvent quitter l'arbre sans avoir de l'impact sur l'acheminement du trafic sur l'arbre multicast.

Lorsque un nœud quitte un groupe, il envoie un message $MACT$ avec le flag P mis à 1. Dans ce cas, le démon BALADE intercepte ce message sur le nœud parent.

Les différents messages de gestion des clés TEK et CSG_i, après un *Leave*, sont détaillés dans le chapitre 3.

4.4 Simulations

Dans cette section, nous comparons des approches de gestion de clés (IOLUS - AKMP - approche centralisée) avec notre protocole BALADE. Les métriques de comparaisons sont le nombre d'opérations de chiffrement / ré-enchiffrement que réalisent les différents CL pour router le flux multicast à tous les membres du groupe, et le nombre de membres affecté par un évènement (*Join* ou *Leave*) afin d'évaluer le phénomène "*1 affecte n*".

On considère des sessions de groupe multicast réelles [AA96], caractérisées par une inter-arrivée de membres qui suit une loi poissonième de paramètre λ, et une durée d'appartenance au groupe qui suit une loi exponentielle de paramètre μ.

Pour générer les temps d'adhésion et de départ du groupe pour chaque membre, nous utilisons un générateur de nombres aléatoires, nous obtenons ainsi deux suites de nombres aléatoires $(x_1, x_2, ..., x_n)$ et $(y_1, y_2, ..., y_n)$.

Les temps d'adhésion au groupe qu'on note λ'_i seront alors calculés de la façon suivante [Eti04]:

$\lambda_i = 1/\lambda * log(1 - x_i)$

$\lambda'_i = \lambda'_{i-1} + \lambda_i$

Pareillement, les temps de départ μ'_i des différents membres du groupe seront calculés de la façon suivante (on ne prend pas en compte le cas de la révocation des membres du groupe) :

$\mu_i = 1/\mu * log(1 - y_i)$

$\mu'_i = \lambda'_i + \mu_i$

La figure 4.7 montre la moyenne du nombre de membres affectés pour l'approche centralisée, IOLUS, AKMP et BALADE. Le nombre moyen de membres affectés pour notre approche BALADE est donc pratiquement égal à celui de AKMP, sauf pour des instants bien définis (300, 650, 1100 ... unités de temps) qui correpondent au renouvellement de la clé de chiffrement du flux TEK. À ces instants, le renouvellement de la clé TEK affecte tous les membres du groupe et correspond donc à une approche centralisée. Le renouvellement de la clé TEK est à la charge de la source, et selon l'unité des données émises.

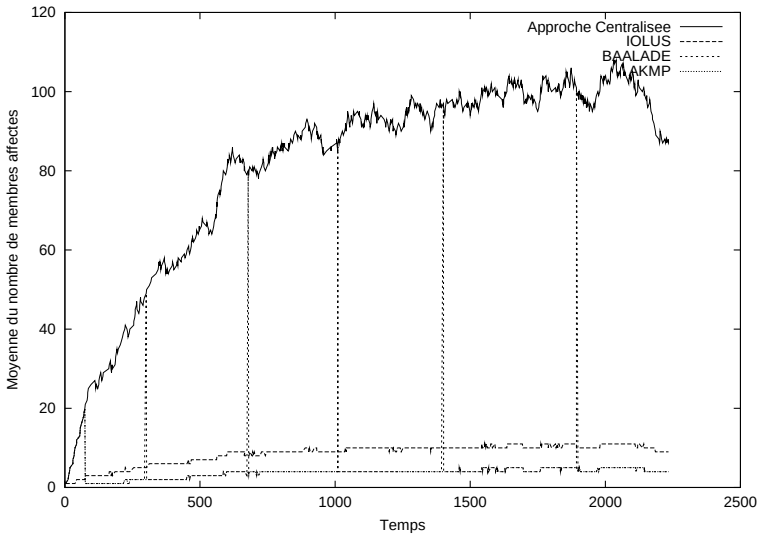

FIG. 4.7 – *Moyenne du nombre de membres affectés*

La figure 4.8 montre le nombre d'opérations de chiffrement / déchiffrement des données, pour l'approche centralisée, IOLUS, BALADE et AKMP. Ceci correspond donc au nombre de sous-groupes ou clusters existant au sein du groupe multicast. Pour l'approche IOLUS, nous avons choisi de subdiviser le groupe en 10 sous-groupes ce qui correspond à 10 opérations de déchiffrement / ré enchiffrement du flux. Pour l'approche centralisée et BALADE, aucune opération supplémentaire de chiffrement / déchiffrement est requise pour les données, les données sont chiffrées par la source et déchiffrés par les récepteurs, avec la même clé TEK. Pour AKMP, le nombre d'opérations de chiffrement / déchiffrement dépend du nombre de sous-groupes crées dynamiquement selon la dynamicité de l'adhésion au groupe.

Les résultats présentés confirment ainsi que BALADE élimine le surcôut dû aux opérations de chiffrement / déchiffrement, au coût d'une affectation complète de tous les membres du groupe lors du renouvellement de la TEK (Trafic Encryption Key) selon la politique de sécurité adaptée par la source en cours de diffusion.

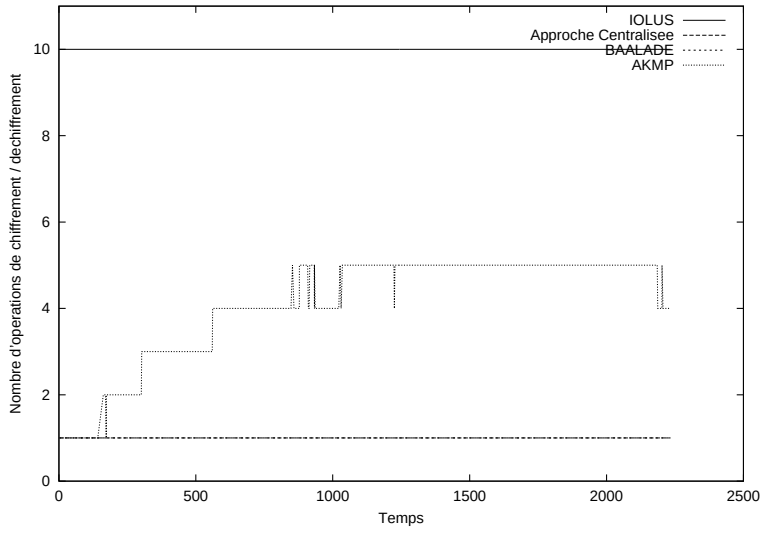

FIG. 4.8 – *Nombre d'opérations de chiffrement / déchiffrement*

Chapitre 5

Conclusion

Sécuriser des communications de groupes dans un environnement Ad Hoc est un véritable défi. Tout d'abord, les applications multicast comme les conférences virtuelles sur Internet et le télé-enseignement présentent plus de vulnérabilité en terme de sécurité que les communications point à point et nécessitent des critères de performances spécifiques quant à la tolérance aux pertes de données, au faible surcoût en communication, au passage à l'échelle de la taille du groupe,...

Les réseaux Ad Hoc sont aussi très sensibles en terme de sécurité, à cause de leurs caractéristiques (absence d'infrastructure, topologie dynamique, bande passante limitée, liens sans fil,...).

Dans ce rapport, nous nous sommes focalisés sur le service de gestion de clé pour la sécurisation des communications de groupes dans le contexte précis de diffusion 1 vers n en séquentiel. Nous avons présenté notre nouvelle approche, appelée BALADE. BALADE fait partie de l'approche hybride, en divisant dynamiquement le groupe multicast en clusters, tout en éliminant le surcoût dû aux opérations de chiffrement / déchiffrement et en atténuant le problème *1 affecte n*.

Actuellement, BALADE est en cours d'implantation et de tests. Nous prévoyons dans la suite de travailler sur l'optimisation de la fonction d'évaluation utilisée pour la clusterisation dynamique du groupe en lui intégrant le facteur de mobilité.

Nous prévoyons également d'adapter la mise en œuvre de BALADE dans le cadre des réseaux hybrides.

Bibliographie

[AA96] K. Almeroth and M. Ammar. Collecting and modelling the join-leave behaviour of multicast group members in the mbone. In *The Symposium on High Performance Distributed Computing*, Syracuse NY, 1996.

[BBC02] H. Bettahar, A. Bouabdallah, and Y. Challal. An adaptive key management protocol for secure multicast. In *11th International Conference on Computer Communications and Networks ICCCN*, Florida USA, October 2002.

[BCF04a] M.S. Bouassida, I. Chrisment, and O. Festor. An Enhanced Hybrid Key Management Protocol for Secure Multicast in Ad Hoc Networks. In N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, and L. Merakos, editors, *Networking 2004, Third International IFIP TC6 Networking Conference*, volume 3042 of *Lecture Notes in Computer Science LNCS*, pages 725–742, Athens, Greece, May 9-14 2004. Springer.

[BCF04b] M.S. Bouassida, I. Chrisment, and O. Festor. Méthodes d'Authentification pour les Communications de Groupe: Taxonomie et Èvaluation dans un Environnement Ad Hoc. In A. Bouabdallah and A. Serchouni, editors, *SAR'2004, Sécurité et Architectures Réseaux, Third Conference on Security and Network Architecture*, pages 197–208, La Londe, Cote d'Azur, France, June 21-25 2004.

[BCK96] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. *RSA CryptoBytes*, 2(1), 1996.

[CBB04] Y. Challal, H. Bettahar, and A. Bouabdallah. SAKM: A Scalable and Adaptive Key Management Approach for Multicast Communications. In *ACM SIGCOMM Computer Communications Review, Volume 34, number 2*, April 2004.

[CCS00] G. Chaddoud, I. Chrisment, and A. Schaff. Baal : Sécurisation des communications de groupes dynamiques. In *The Proceedings of the 8th Colloque Francophone sur l'Ingénierie des Protocoles CFIP'2000*, Toulouse, France, October 2000.

[CQN+02] H. Chu, L. Qiao, K. Nahrstedt, H. Wang, and R. Jain. A Secure Multicast Protocol with Copyright Protection. *SIGCOMM Comput. Commun. Rev.*, 32(2):42–60, 2002.

[DMS99] L. Dondeti, S. Mukherjee, and A. Samal. Secure one-to-many group communication sing dual encryption. In *Comput. Com-mun.23, 17 (Nov.)*, 1999.

[EFL+99] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. RFC 2693 - SPKI Certificate Theory, September 1999.

[Eti04] H. Etiévant. http://cyberzoide.developpez.com/java/random/random.pdf, 2004.

[Fed01] Federal Information Processing Standard FIPS. Specification for the Advanced Encryption Standard (AES), November 2001.

[HFPS99] R. Housley, W. Ford, W. Polk, and D. Solo. RFC 2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile, January 1999.

[KWM+] J. Kadlecsik, H. Welete, J. Morris, M. Boucher, and R. Russell. The netfilter/iptables Project http://www.netfilter.org.

[Leg03] V. Legrand. Rapport de DEA, Etablissement de la Confiance et Réseaux Ad Hoc - Le Germe de Confiance, EDIIS, Laboratoire CITI, INRIA ARES, Juillet 2003.

[Len02] J. Leneutre. Authentification dans les réseaux ad hoc : Problématique et état de l'art. In *SAR'2002*, Telecom Paris, July 2002.

[LJM+03] A. Laouiti, P. Jacquet, P. Minet, L. Viennot, T. Clausen, and C. Adjih. Multicast optimized link state routing. Research Report 4721, INRIA, February 2003.

[MC02] G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Verifiable Identifiers and Addresses. In *ISOC Network and Distributed System Security Symposium (NDSS)*, February 2002.

[Mit97] S. Mittra. Iolus: A framework for scalable secure multicasting. In *SIGCOMM*, pages 277–288, 1997.

[Riz97] L. Rizzo. Effective erasure codes for reliable computer communication protocols. *ACM Computer Communication Review*, 27(2):24–36, April 1997.

[RP00] E. Royer and C. Perkins. Multicast Ad hoc On-Demand Distance Vector (MAODV) routing, IETF Internet Draft: draft-ietf-manet-maodv-00.txt, 2000.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889 - RTP: A Transport Protocol for Real-Time Applications, January 1996.

[SKW+99] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson. Performance of the AES Candidate Algorithms in Java. In *2 nd AES conference*, Rome, Italy, march 1999.

[SLFC03] H. Sallay, A. Lahmadi, O. Festor, and I. Chrisment. Extension de l'architecture active amam pour le support des services de sécurité multicast. In *GRES'2003 Colloque Francophone sur la Gestion de Réseaux et des Services*, February 2003.

[Wei01] N. Weiler. SEMSOMM : A scalable multiple encryption scheme for one-to-many multi-cast. In *In Proceedings of the 10th IEEE International WETICE Enterprises Security Workshop, (Cambridge, Mass., June). IEEE Computer Society Press, Los Alamitos, Calif*, 2001.

[Zap02] M.G. Zapata. Secure ad hoc on-demand distance vector routing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6:106–107, Jully 2002.

[ZH99] L. Zhou and J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

Annexe

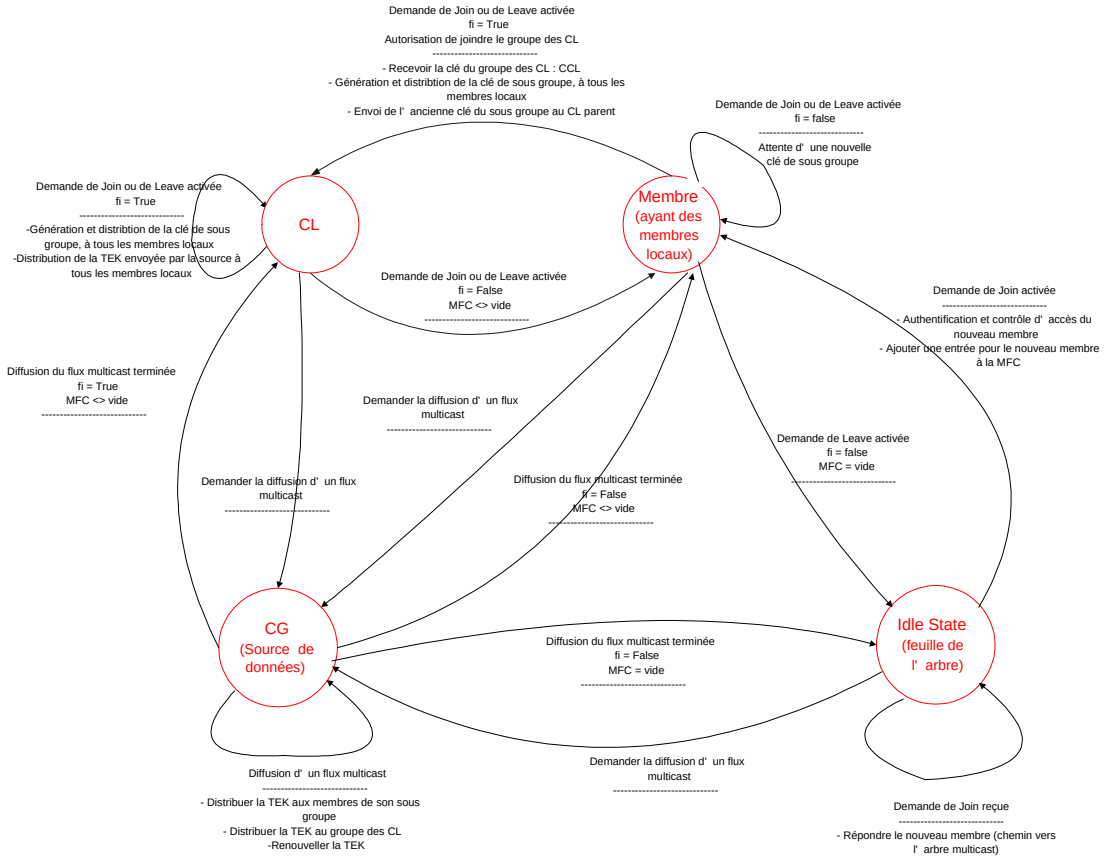

FIG. 5.1 – *Automate d'états d'un Contrôleur du groupe*

Glossaire

– [ACL :] (Access Control List) : Liste des membres autorisés à joindre le groupe multicast
– [CBID :] (Crypto Based Identifier) : Identificateur cryptographique
– [CCL :] Clé du groupe des Contrôleurs Locaux
– [CG :] Contrôleur global du groupe
– [CL_i :] Contôleur Local du i^{eme} sous-groupe
– [CSG_i :] Clé du sous-groupe i
– [DW :] Membres fils du nœud en question
– [GCL :] Groupe des Contrôleurs Locaux
– [GM :] Membre du groupe
– [KEK :] Key Encryption Key
– [LPL :] (Local Participant List) : liste des membres locaux d'un contrôleur
– [SG_i :] sous-groupe i
– [Pri_n :] Clé privée du nœud n
– [Pub_n :] Clé publique du nœud n
– [RL :] Liste des membres expulsés ou banis du groupe
– [TEK :] Traffic Encryption Key
– [UP :] (Upstream) : membre parent du nœud en question

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399