HAL
open science

Maîtrise de la dynamique dans l'Internet – de l'adaptation des protocoles à la sécurité des services –

Isabelle Chrisment

To cite this version:

Isabelle Chrisment. Maîtrise de la dynamique dans l'Internet – de l'adaptation des protocoles à la sécurité des services –. Réseaux et télécommunications [cs.NI]. Université Henri Poincaré - Nancy I, 2005. tel-00010870

HAL Id: tel-00010870

https://theses.hal.science/tel-00010870

Submitted on 4 Nov 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Département de formation doctorale en informatique
UFR STMIA

École doctorale IAEM Lorraine

Maîtrise de la dynamique dans l'Internet – de l'adaptation des protocoles à la sécurité des services –

MÉMOIRE

présenté et soutenu publiquement le 21 Octobre 2005

pour l'obtention de l'

Habilitation à Diriger des Recherches de l'université Henri Poincaré

(spécialité informatique)

par

Isabelle CHRISMENT

Composition du jury

Rapporteurs :	Paul Amer	Professeur à University of Delaware, USA
	Abdelmadjid Bouabdallah	Professeur à l'Université de Technologie de Compiègne
	Jean-Jacques Pansiot	Professeur à l'Université Louis Pasteur, Strasbourg
Examinateurs :	Olivier Festor	Directeur de Recherche à l'INRIA Lorraine
	André Schaff	Professeur à l'ESIAL, Université Henri Poincaré, Nancy
	Stéphane Ubéda	Professeur à l'INSA de Lyon

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

Je remercie Paul AMER, Professeur à University of Delaware (USA), Abdelmadjid BOUABDALLAH, Professeur à l'Université de Technologie de Compiègne et Jean-Jacques PANSIOT, Professeur à l'Université Louis Pasteur à Strasbourg, d'avoir accepté d'être mes rapporteurs et de réaliser ainsi une évaluation de mon travail. Merci également Stéphane UBEDA, Professeur à l'INSA Lyon, d'avoir fait l'honneur de participer à mon jury.

Je tiens également à remercier André SCHAFF de m'avoir parrainé lors de ma candidature à mon habilitation à diriger les recherches et d'avoir également accepté d'être dans le jury. J'ai une profonde reconnaissance pour tout ce qu'il m'a apporté lors de mon parcours. Responsable de la filière DESS que j'ai suivie en 1986-1987, il a ensuite été mon responsable scientifique dans l'équipe RESEDAS lorsque j'ai été recrutée puis mon responsable pédagogique en tant que directeur de l'ESIAL.

Je ne voudrais pas oublier Olivier FESTOR dont j'ai pu apprécier les compétences ainsi que les qualités humaines en tant que responsable scientifique de MADYNES. J'ai appris beaucoup à son contact et ses conseils m'ont toujours été d'une grande aide. Pour moi, il était important qu'il fasse partie de mon jury.

Je souhaiterais remercier également Christian HUITEMA qui a été à l'origine de mon entrée dans le monde de la recherche et grâce à qui j'ai pu réaliser mes travaux de thèse.

Merci individuellement à tous les membres de l'équipe MADYNES du LORIA pour leur soutien, plus particulièrement à tous les étudiants que j'ai encadrés dont Ghassan CHADDOUD, Laurent CIARLETTA et Mohamed Salah BOUASSIDA. L'ensemble des contributions est le résultat d'un travail commun et de nombreuses heures à travailler ensemble. Merci aussi à Radu STATE pour m'avoir supportée et soutenue. Partager le bureau d'un collègue aussi brillant et sympathique est un véritable plaisir.

Je termine par une pensée spéciale pour ma petite famille. Je lui sais gré de tout son support moral pendant les différentes phases de l'écriture du mémoire. Merci surtout à Erik d'avoir accepté de s'occuper des enfants afin de me laisser travailler. Je sais que cela n'a pas toujours été facile...

Je dédie cette habilitation à

mon mari

Érik

nos trois enfants

Aline,
Adrien
et
Augustin

Table des matières

Curriculum Vitæ

1 Introduction 19

1.1 Contexte . . . 19
1.2 Contributions . . . 19
1.3 Organisation du mémoire . . . 22
Bibliographie . . . 22

2 Adaptation des protocoles : ALF et les réseaux actifs 23

2.1 Introduction . . . 23
2.2 Le concept ALF . . . 24
2.2.1 Contexte . . . 24
2.2.2 Motivation . . . 25
2.2.3 Automatisation de l'architecture ALF. . . . 27
2.3 Réseaux Actifs . . . 28
2.3.1 Contexte . . . 28
2.3.2 Technologie active et Supervision . . . 29
2.3.3 Technologie Active et Tests IPv6 . . . 32
2.4 Conclusion . . . 37
Bibliographie . . . 37
Publications . . . 40

3 Sécurité dans les communications de groupe 43

3.1 Introduction . . . 43
3.2 Motivation . . . 45
3.3 Classification des approches de gestion de la clé de groupe . . . 48
3.3.1 Approches centralisées . . . 48
3.3.2 Approches hiérarchiques . . . 50
3.3.3 Les approches hybrides . . . 51

3.3.4 Approches distribuées 52
3.3.5 Synthèse 52
3.4 `Baal` 53
3.4.1 Architecture 53
3.4.2 Protocole 54
3.4.3 Évaluation de `Baal` 57
3.5 Extension d'IGMP Proxying 59
3.6 `S-SSM` 61
3.6.1 Le modèle SSM 61
3.6.2 L'architecture `S-SSM` 63
3.7 `S-SSM` et `DCCP` 66
3.8 Conclusion 68
Bibliographie 69
Publications 72

4 Sécurité dans les réseaux spontanés 75

4.1 Introduction 75
4.2 Sécurité et informatique ambiante 76
4.2.1 Contexte 76
4.2.2 Motivation 76
4.2.3 VPSS (*Virtual Private Smart Space*) 77
4.3 Sécurité et réseaux ad hoc 77
4.3.1 Contexte 77
4.3.2 Motivation 78
4.3.3 Adaptation de `Baal` aux environnements ad hoc 80
4.3.4 `BALADE` 84
4.3.5 Évaluation des méthodes d'authentification des flux multicast 88
4.4 Conclusion 91
Bibliographie 92
Publications 95

5 Synthèse et Perspectives 97

5.1 Résumé des contributions 97
5.2 Projet de recherche 98
5.2.1 Distribution de clés de groupe 98
5.2.2 Aspects autorisation et contrôle d'accès 100
5.2.3 Sécurité et gestion de réseaux/services 101

Bibliographie 102
Publications 103

Acronymes **105**

Annexes **107**

Table des figures

1.1 Résumé des contributions 20

2.1 ILP et non ILP 26
2.2 Classification 28
2.3 Mode de fonctionnement de FLAME 35
2.4 Automate MLD 36
2.5 Approche FLAME Test Seul Routeur Querier Par Lien 36

3.1 Évolution de la vie d'un groupe sécurisé 47
3.2 Graphe de clés : U3 a les clés K3, K34, K1234 49
3.3 Un arbre OFT 50
3.4 Architecture de `Baal` 54
3.5 Initialisation du groupe 55
3.6 Arbre de recouvrement enraciné en (A) 60
3.7 Un hôte s'abonnant à un *canal* enraciné en S 62
3.8 Environnement **S-SSM** 64

4.1 Architecture adaptée 81
4.2 Exemple d'évaluation de mn 82
4.3 Temps de renouvellement de la clé suite à un Leave selon le nombre de membres . 83
4.4 Temps de renouvellement de la clé par fréquence d'évènements 84
4.5 Distribution de TEK 86
4.6 Authentification d'un nouveau et ancien membre 87
4.7 Juke-box dans un réseau ad hoc 87

Curriculum Vitæ

Isabelle CHRISMENT

Date et lieu de naissance :	12 Décembre 1964 à Mirecourt (Vosges)
Nationalité :	française
Situation familiale :	mariée, trois enfants
Adresse professionnelle :	LORIA Campus Scientifique BP 239 54506 Vandœuvre-lès-Nancy Cedex Tél : +33 (0)3 83 59 20 17 Fax : +33 (0)3 83 27 83 19 Mél : ichris@loria.fr URL : http ://www.loria.fr/~ichris
Adresse personnelle :	7, rue de Laxou 54600 Villers-lès-Nancy Tél : +33 (0)3 83 90 42 52 (domicile)
Situation actuelle :	Maître de Conférences, PEDR depuis Octobre 2002.

Titres universitaires français

1996 Thèse de doctorat en informatique ayant pour titre *"Etude et développement d'applications distribuées dans l'architecture ALF"*, soutenue le 11 Juin 1996 à l'Université de Nice/Sophia-Antipolis et dirigée par Christian Huitema. Mention Très Honorable. Jury : M. Banâtre, J.P Courtiat, J. Crowcroft, C. Huitema, J.P Rigault, A. Schaff.

1987 D.E.S.S Informatique à Nancy, spécialisation Réseaux et Télématique, Mention Très Bien. Major de promotion.

1986 Maîtrise MIAGe à Mulhouse, Mention Bien.

1984 D.U.T Informatique à Belfort, Major de promotion.

Expériences Professionnelles

Depuis Fév. 2001	Maître de Conférences à l'ESIAL (École Supérieure d'Informatique et Applications de Lorraine) au sein de l'Université Henri Poincaré. Membre du projet RESEDAS puis MADYNES au LORIA (Laboratoire lOrrain de Recherche en Informatique et ses Applications). Responsable permanent de MADYNES depuis sa création (fin 2002). CRCT (Congé pour Recherches et Conversions Thématiques) d'une durée de 6 mois en 2004-2005.
De Sept. 1997 à Janv. 2001	Maître de Conférences à l'IUT Nancy-Verdun, département informatique, Université Nancy 2 et Membre du projet RESEDAS au LORIA.
De Sept. 1996 à Août 97	A.T.E.R à l'ESSI (École Supérieure en Sciences Informatiques) à Sophia Antipolis.
De Juil. 1996 à Août 1996	Ingénieur Expert à l'INRIA Sophia Antipolis dans le projet RODEO (Réseaux à haut débit).
De Fév. 1993 à Juin 1996	Boursière de recherche dans le projet RODEO dirigé par Christian Huitema à l'INRIA Sophia Antipolis.
De Oct. 1987 à Janv. 1993	Ingénieur chez SEMA GROUP, Société de Services en Informatique : De 1991 à Janvier 1993, j'ai été en assistance technique chez Digital Equipment Corporation à Sophia Antipolis, Division Engineering de Telecom Business Group, centre de compétence mondial de DEC pour le marché des télécommunications. En 1990, j'ai participé à l'étude du système de gestion automatisée du réseau d'assainissement de la Seine Saint-Denis. De 1989 à 1990, j'ai été responsable technique à SEMA GROUP Paris du projet ESPRIT DELTA-4, dont l'objectif était de définir et de concevoir une architecture pour un système distribué sur un réseau local (Token Ring) et tolérant aux pannes. De 1987 à 1988, j'ai participé au développement du produit GENEPX400, système de certification de protocoles en messagerie X400.

Domaine de recherche

L'ensemble des travaux de recherche que j'ai menés sont orientés vers l'évolution des protocoles de l'Internet.

Dans le cadre de ma thèse et de mon année d'ATER que j'ai effectuées dans le projet RODEO (actuellement PLANETE) de l'INRIA Sophia Antipolis, j'ai étudié le développement des applications distribuées dans l'architecture ALF (Application Level Framing). Le travail réa-

lisé a montré qu'il est possible de concevoir et d'implanter des systèmes de communication qui tiennent compte des contraintes imposées par l'application. Il a abouti à un environnement de développement complet et automatisé.

Depuis 1997, j'ai intégré le laboraroire LORIA au sein du projet RESEDAS, puis MADYNES, dont la thématique actuelle est la supervision des réseaux et services dynamiques. Après avoir étudié l'utilisation des réseaux actifs pour la supervision et les tests de protocoles (plus particulièrement IPv6), je me suis focalisée sur une aire fonctionnelle de la gestion, à savoir la sécurité. Je travaille actuellement sur la conception d'infrastructures pour la distribution des clés dans les réseaux à forte dynamicité : communication de groupe et environnement ad hoc. Une architecture et un protocole, `Baal`, pour diffuser les clés de groupe dans les communications multipoints ont été proposés. Nous regardons actuellement, de manière plus approfondie, comment adapter ces propositions pour prendre en compte les contraintes liées à l'ad hoc.

Réalisations logicielles et plates-formes

Les réalisations et prototypes auxquels j'ai personnellement contribué sont les suivants :

Dans le cadre du LORIA :

En 1997, j'ai participé à la mise en place d'une plate-forme IPv6 au LORIA et au raccordement au G6-bone, réseau français expérimental pour IPv6, via le site situé à Strasbourg et via le point d'interconnexion d'organisme (PIO) INRIA situé alors à Montbonnot. Actuellement, je travaille avec des personnes du CIRIL (Centre Interuniversitaire de Ressources Informatiques de Lorraine) et un ingénieur du projet MADYNES pour déployer le service IPv6 sur l'ensemble des universités lorraines avec comme réseau pilote l'ESIAL et la mise en place d'un démonstrateur entre le LORIA et l'Université de Strasbourg intégrant la mobilité et le multicast IPv6. Cette action est réalisée dans un projet (IPV6-ADIRE) initié par la Direction de la Recherche concernant la validation et le déploiement de service IPv6 en vue de prouver notamment les fonctionnalités de mobilité, sécurité et multicast.

Dans le cadre de la thèse :

J'ai spécifié et mis au point le prototype d'un compilateur de protocoles qui génère automatiquement, à partir de spécifications formelles d'une application distribuée, une implantation efficace. Le prototype a servi de base pour la mise au point d'un environnement de développement plus complet.

J'ai développé un serveur d'images JPEG et des mécanismes de protocoles appropriés en vue d'une démonstration pour la communauté européenne dans le cadre du projet HIPPARCH (*H*Igh Performance Protocol ARCHitecture) issu d'une collaboration australo-européenne.

Dans le cadre de SEMA GROUP :

J'ai développé un prototype réalisant la communication entre un OS (Operation System) et un MD (Mediation Device) via l'interface Q3 (Protocole CMIS/CMIP) au dessus de DECmcc, plateforme de développement pour la gestion et l'administration des ressources d'une entreprise.

J'ai participé à l'implantation de l'interface utilisateur de TeMIP (Telecom Management Information Platform), produit d'administration dans le domaine des réseaux publics s'appuyant sur DECmcc.

D'autres réalisations logicielles issues des travaux de recherche ont été effectuées par des étudiants que j'ai encadrés, notamment :

- la mise en œuvre d'un logiciel permettant la gestion des clés dans les communications de groupe avec l'implantation du protocole `Baal` (A. Lahmadi) ;
- l'implantation côté routeur du protocole d'adhésion à un groupe, IGMPv3 (A. Lahmadi) ;
- l'implantation d'un proxy IGMP qui offre une solution pour diffuser les données en multipoint, au sein du même domaine administratif, entre des routeurs ne supportant pas le routage multipoint (A. Ben Hellel).

Recherches contractuelles

Cette section présente les actions contractuelles auxquelles j'ai directement participé.

SAFECAST

Début : 2004 Fin : 2007 (36 mois)

SAFECAST est un projet exploratoire labellisé en Septembre 2003 qui regroupe EADS, LAAS-CNRS, ENST, LORIA/INRIA, Heudyasic UTC Compiègne. Ce projet vise à développer une architecture globale de sécurité permettant la communication multipoint dans un environnement sécurisé où tout membre peut être à la fois récepteur et émetteur. La sécurité des communications de groupe devra être maintenue tout en autorisant une dynamicité au niveau des récepteurs (rejoindre ou quitter le groupe).

J'ai participé à l'élaboration de ce projet et j'en assume actuellement la direction pour l'équipe MADYNES du LORIA. Le travail est réalisé avec la participation active de M.S. Bouassida dans le cadre de sa thèse.

SAFARI

Début : 2003 Fin : 2005 (30 mois)

SAFARI est un projet pré-compétitif labellisé en Septembre 2002 qui regroupe France Telecom R&D, Alcatel, INRIA, LIP6, LRI, LSIIT, LSR-IMAG, SNCF, Telecom Paris. Ce projet propose l'étude, la réalisation et l'expérimentation d'une architecture de réseau intégrée pour déployer des services dynamiques sur un réseau IPv6 hybride ad hoc/filaire.

Dans le cadre de ce projet, j'interviens sur un sous-projet spécifique relatif à l'administration et au contrôle de l'infrastructure. Un domaine fonctionnel que je regarde de plus près est celui de la sécurité. Il s'agit de voir comment adapter les mécanismes de sécurité disponibles pour les services multicast dans le contexte d'un réseau ad hoc avec passerelle au monde fixe.

Une étude de l'adaptation du protocole `Baal` développé en interne aux besoins de l'ad hoc ainsi qu'une application de démonstration sont actuellement en cours d'élaboration.

Fonds France Canada pour la Recherche :

Début : 2001 Fin : 2003 (24 mois)

En 2001, le projet *Formalisation et Tests d'IPv6* a été retenu par le FFCR et a permis de commencer une collaboration entre l'équipe MADYNES et le Professseur Rachida Dssouli tout

d'abord à l'Université de Montréal et actuellement en poste à Concordia University.

L'objectif principal du projet était la génération automatique des suites de tests de conformité et d'interopérabilité des protocoles d'Internet Nouvelle Génération ou IPv6. Les deux équipes ont travaillé de manière complémentaire :

- l'équipe de Rachida Dssouli sur la modélisation des protocoles IPv6 en SDL et la génération automatique de suites de tests ;
- l'équipe de Nancy sur l'utilisation d'un environnement actif pour exécuter les tests IPv6.

J'ai assuré avec André Schaff la co-direction de ce projet et Mohamed Salah Bouassida a mis en place, lors de son stage de fin d'étude, les suites de tests dans l'environnement actif disponible dans MADYNES.

Un article commun, publié à IEEE IWCSE 2002, est issu de ce projet.

ANAIS :

Début : 1998 Fin : 2000 (24 mois)

ANAIS (Active Network Architecture for Internet Service Providers) est un projet soutenu par le programme Télécom du CNRS. Ce projet visait la définition d'une architecture de supervision pour des réseaux actifs afin de permettre un déploiement efficace et sécurisé de services chez un fournisseur de services.

Les travaux réalisés ont permis : l'expérimentation de nombreuses plates-formes actives disponibles, la connexion au ABone, la définition d'une architecture d'accueil pour services actifs adaptée à notre cahier des charges, la réalisation d'une plate-forme de supervision et l'élaboration d'une formation complète sur les réseaux programmables.

J'ai assuré, en collaboration avec Olivier Festor, la réalisation de ce projet.

Encadrement de recherche

Co-encadrement de thèses

Depuis Septembre 2003, je co-encadre avec Olivier Festor la thèse de Mohamed Salah Bouassida relative à la sécurité des communications de groupe dans des infrastructures ad hoc. Ces travaux ont déjà donné lieu à cinq publications : deux publications internationales dans une conférence très sélective (Networking'04 et Networking'05) et trois dans des conférences francophones (CFIP'05, SAR'05 et SAR'04). Le papier de SAR'2005 a été sélectionné pour être publié, sous forme étendue, dans le journal *Annales des Télécommunications*. Cinq rapports de contrats ainsi qu'un rapport de recherche ont été également rédigés.

J'ai également co-encadré sous la responsabilité de André Schaff deux thèses :

- la thèse de Ghassan Chaddoud relative à la définition d'un protocole de sécurité pour les communications dans les groupes dynamiques. Quatre publications dans des conférences internationales, deux dans des conférences francophones et une dans un journal francophone sont issues de ces travaux. G. Chaddoud a soutenu sa thèse en août 2002 et est actuellement chercheur au Département d'Informatique au CEA syrien à Damas.
- la thèse de Laurent Ciarletta dont les travaux, effectués en grande partie au NIST (National Institut of Standards and Technologies) aux Etats-Unis, portent sur l'évaluation

des technologies de l'informatique ambiante. Deux publications dans des conférences internationales et une dans une conférence francophone sont issues de ces travaux. Laurent Ciarletta a soutenu sa thèse en novembre 2002 et est actuellement Maître de Conférences à l'Ecole des Mines de Nancy.

Jury de thèse

J'ai été examinatrice dans les thèses suivantes :

- Van Le sous la direction de Hervé Guyennet. Thèse de l'Unversité de Franche-Comté, Besançon. Titre : *GRIDSEC : une architecture sécurisée pour le GRID Computing.* Jury : L. Philippe (président), T. Ludwig (rapporteur), A. Schaff (rapporteur), I. Chrisment (examinatrice) et F. Desprez (examinateur). Septembre 2003.
- Laurent Ciarletta sous la direction de André Schaff et d'Isabelle Chrisment. Thèse de l'Université Henri Poincaré, Nancy. Titre : *Contribution à l'évaluation des technologies de l'informatique ambiante.* Jury : J-C. Derniame (président), A. Duda (rapporteur), G. Pujolle (rapporteur), I. Chrisment (examinatrice) et A. Schaff (examinateur). Novembre 2002.
- Ghassan Chaddoud sous la direction de André Schaff et d'Isabelle Chrisment. Titre : *Sécurisation de communication de groupes dynamiques.* Jury : F. Alexandre (président), P. Rolin (rapporteur), V. Varadharajan (rapporteur), I. Chrisment (examinatrice), A. Serhrouchni (examinateur) et A. Schaff (examinateur). Août 2002.

Encadrement de stages de DEA

Depuis mon arrivée au LORIA, j'ai encadré et co-encadré dix stages de DEA d'une durée de cinq mois. Lorsque les sujets avaient une intersection avec les thèses en cours, j'ai impliqué mes étudiants en thèse dans l'encadrement du DEA (ce qui est indiqué dans le tableau ci-dessous). De même, j'ai co-encadré avec des industriels des étudiants dont le stage était réalisé dans le laboratoire de recherche et développement de l'entreprise.

2004-2005 : *Adaptation de l'architecture AAA aux réseaux ad hoc.* Stage effectué par Xavier Grandmougin. Le stage consistait à étudier comment adapter et étendre l'architecture AAA pour qu'elle puisse être déployée dans les réseaux ad hoc.

2002-2003 : *Sécurité multicast et réseaux ad hoc.* Stage effectué par Mohamed Salah Bouassida. Le stage consistait à évaluer l'impact d'une infrastructure ad hoc sur la sécurité multicast et à proposer un modèle de sécurité pour les réseaux ad hoc dans un environnement de communication de groupe.

2001-2002 : *Sécurité pour l'informatique ambiante.* Stage effectué par Ahmed Ait Ali et co-encadré avec L. Ciarletta. Le stage consistait à étudier les différentes technologies utilisées pour sécuriser les réseaux de l'informatique ambiante et notamment les technologies sans fil.

2000-2001 : *Sécurité et communication de groupe spécifique à une source.* Stage effectué par Saber Brakta et co-encadré avec G. Chaddoud. Le stage consistait à proposer une architecture pour la sécurisation d'un groupe dynamique ayant une seule source.

2000-2001 : *Architecture AAA et mobilité sur IPv6 adaptées aux applications temps réel.* Stage effectué par Antoine Peyronnel et co-encadré avec O. Charles. Ce stage a été fait en coopération avec France Telecom R&D pour étudier l'autorisation et l'authentification des nœuds.

1999-2000 : *Différenciation de services et réseaux actifs.* Stage effectué par Christopher Scott et co-encadré avec O. Festor. Le stage consistait à définir les apports de la technologie des réseaux actifs dans le cadre de la différenciation de services.

1998-1999 : *Test de validation du protocole de contrôle IPv6.* Stage effectué par Ismaila Ndiaye et co-encadré avec A. Schaff. Le stage consistait à spécifier en SDL certaines fonctionnalités du protocole IPv6.

1997-1998 : *Vers une approche sécurisée des mécanismes d'autoconfiguration dans IPv6.* Stage effectué par Laurent Ciarletta. Le stage consistait à étudier les problèmes posés par la mise en application concurrente d'IPsec et des mécanismes d'autoconfiguration d'IPv6.

1997-1998 : *Plan de Test Internet Nouvelle Génération.* Stage effectué par David Mercier et co-encadré avec F. Pham-Khac. Ce stage a été fait en coopération avec DASSAULT Electronique pour mener une étude sur la conception d'une architecture appliquée aux tests fonctionnels de la souche IPv6.

1997-1998 : *La sécurité dans IPv6 pour des applications multipoints.* Stage effectué par Ghassan Chaddoud. Le stage consistait principalement à étudier les travaux existants pour la gestion des clés multicast.

Encadrement de stages de fin d'études (écoles d'ingénieurs ou universités)

Depuis 1997, j'ai encadré et co-encadré huit étudiants réalisant leur stage de fin d'études d'une durée d'environ quatre mois au sein de l'équipe RESEDAS puis MADYNES :

2001-2002 :	Mohamed Salah Bouassida	*Utilisation de la technologie active pour tester des protocoles multicast IPv6.* Stagiaire ENSI (Tunisie).
2001-2002 :	Anis Ben Hellel	*Une architecture SSM sécurisée utilisant IGMP Proxying.* Stagiaire EPT (Tunisie), co-encadrement avec A. Lahmadi et G. Chaddoud.
2000-2001 :	Abdelkader Lahmadi	*Mise en œuvre d'un outil de gestion des clés de groupes dynamiques.* Stagiaire ENSI (Tunisie), co-encadrement avec G. Chaddoud.
1999-2000 :	Anis Koubaa	*Conception et réalisation d'un outil de simulation pour le protocole de gestion de clé multicast GKMP, en utilisant le simulateur NS.* Stagiaire SUP'COM (Tunisie), co-encadrement avec G. Chaddoud.
1998-99 :	Jalal Zarhoun	*Installation et configuration du logiciel GateD sur la plateforme IPv6.* Stagiaire EMI (Maroc).
1998-99 :	Khadija Boumahdi	*Expérimentation du protocole IPsec sur la plateforme IPv6.* Stagiaire ENSIAS (Maroc).
1997-98 :	Najib Belkhayat	*Étude du protocole RSVP (Resource ReServation Protocol) avec mise en place de tests sur la plateforme IPv6 du LORIA.* Stagiaire ENSIAS (Maroc).

1997-98 :	Amine Boufaied	*Développement d'une interface de programmation socket IPv6.* Stagiaire Université de Tunis (Tunisie).

Encadrement d'ingénieurs ou post-dea

Juil. 2005-Sept. 2005 :	Najah Chridi	*Validation du protocole de gestion de clés défini dans le projet SAFECAST.*
Août 2005-Janv. 2006 :	Vincent Delove	*Mise en place d'un démonstrateur (Jukebox pair-à-pair) intégrant le multicast et le mobilité pour le projet IPV6-ADIRE.*

Accueil de chercheurs étrangers

En Août 2002, le professeur Vijay Varadharajan de l'Université Macquarie de Sydney a effectué un séjour de 1 mois dans le cadre de nos travaux sur la sécurité et les communications de groupe.

En 2003, le professeur Rachida Dssouli de Concordia University, Montréal, Canada, a passé deux mois (Juin, Juillet) dans l'équipe MADYNES. Durant son séjour, elle a travaillé sur les modèles et infrastructures pour les tests des protocoles IPv6.

Animation scientifique

Organisations de conférences

- Organisation en 2003, de la 2ème conférence francophone sur la Sécurité et Architecture Réseaux (SAR'2003) qui s'est tenue à Nancy du 30 juin au 4 juillet 2003. Cette conférence a été co-organisée par le LORIA (Laboratoire lorrain de recherche en informatique et ses applications), Nancy, France et par Concordia University, Montréal, Canada.
- Membre du comité d'organisation de la deuxième rencontre francophone sur les aspects ALGOrithmiques des TELécommunications en 2000 (Algotel 2000).
- Membre du comité d'organisation du Colloque Francophone sur l'Ingéniérie des Protocoles en 1999 (CFIP 1999). Cette conférence a lieu tous les dix huit mois et représente la conférence francophone la plus importante dans le domaine des réseaux et des protocoles.
- Dans le cadre du G6, groupement français des chercheurs et industriels travaillant sur le 6-bone (réseau mondial expérimental du protocol IPv6), j'ai co-organisé au LORIA un connectathon mobilité IPv6 qui s'est tenu à Nancy du 15 au 17 Septembre 1999 et a permis à différents développeurs de tester leur pile mobilité pour le protocole IPv6 et de valider la possibilité d'interopérabilité de leur logiciel. Cette manifestation a regroupé des industriels avec BULL, ERICSSON-TELEBIT, NEC, FRANCE-TELECOM, EUROCONTROL et des académiques avec l'INRIA, les Universités de Strasbourg et Nancy.

Comités de programme

- Membre du comité de programme du Colloque Francophone sur l'Ingéniérie des Protocoles en 2002, 2003, 2005. .

- Membre du comité de programme de la conférence francophone sur la Sécurité et Architecture Réseaux (SAR) depuis 2003. Cette conférence a été créée en 2002 par A. Serhrouchni de l'ENST Paris. Le processus de sélection avec comité de lecture a été mis en place en 2003 pour la première fois. Le succès de cette conférence s'est confirmé en 2004. Depuis Juin 2005, je suis également membre du comité de pilotage.
- Membre du comité de programme de NOTERE 2004, 2005 et 2006 (Les NOuvelles TEchnologies de la REpartition).
- Membre du comité de programme de International Conference on Service Assurance with Partial and Intermittent Resources (SAPIR 2005).
- Membre du comité de programme de Montreal Conference on e-Technologies (MCETECH 2005).
- Membre du comité de programme de International Conference on Advances in Intelligent Systems - Theory and Applications (IASTA 2004).
- Membre du comité de programme du Workshop sur la Sécurité des Technologies de l'Information (WSTI 2003).
- Membre du comité de programme de International Workshop on Communication Software Engineering en (IWCSE 2002).

Réseau d'excellence

- Membre du réseau d'excellence européen MAGIX (Management Solutions for Next Generation) qui regroupe au niveau de l'Europe quatorze équipes académiques travaillant sur la supervision et la gestion de réseaux.

Relectures

- Relectrice pour la conférence International Symposium on Integrated Network Management (IM 2005), pour la conférence IEEE/IFIP Network Operations and Management Symposium (NOMS 2006) et pour toutes les conférences pour lesquelles j'ai été membre du comité de programme.

Commissions et responsabilités administratives

- Depuis septembre 2003, membre de la commission de choix de l'ESIAL.
- De septembre 2001 à septembre 2004, membre élue titulaire de la commission de spécialistes, 27ème section de l'Université Henri Poincaré à Nancy.
- En 2004, membre titulaire de la commission de spécialistes, 27ème section de l'Université Louis Pasteur de Strasbourg.
- En 2004, membre de la commission de choix des enseignants de l'IUT de St Dié.
- De septembre 2001 à fin 2003, membre suppléante de la commission de spécialistes, 27ème section de l'Université Louis Pasteur de Strasbourg.
- En 2000, membre suppléante de la commission de spécialistes, 27ème section de l'Université de Nancy 2.
- Depuis 2002, membre de la commission ingénieurs du comité de recutement INRIA/LORIA des personnels scientifiques contractuels. La comission intervient dans le recrutement de ingénieurs associés INRIA et émet un avis sur le recrutement des ingénieurs experts.

- Responsable, en tant que Maître de Conférences ESIAL/UHP, d'un projet retenu par la Région Lorraine pour 2002-2003 dans le cadre de la mise en place de formations innovantes et intitulé "Formation aux évolutions les plus récentes dans l'Internet (protocole IPv6, mobilité, réseaux sans fil, téléphonie IP, Qualité de services)"
- Responsable de la spécialisation Télécommunications, Réseaux et Services à l'ESIAL/UHP depuis 2002. En plus des charges administratives et pédagogiques inhérentes à cette fonction, j'ai notamment, lors de mon arrivée, réorganisé la 3ème année en l'orientant vers les protocoles et services de l'Internet.
- Responsable permanent du projet MADYNES depuis sa création fin 2002. Dans ce cadre, je travaille en étroite collaboration avec Olivier Festor, directeur de recherche à l'INRIA Lorraine et responsable scientifique du projet MADYNES.

Activités d'enseignement

En qualité de vacataire durant ma thèse (1993-1996), d'Attaché Temporaire d'Enseignement et de Recherche (1996-1997), de Maître de Conférences (depuis 1997), je suis intervenue principalement dans les enseignements de Réseaux et Systèmes.

Enseignement en 3ème cycle

- DEA Informatique de Lorraine : Filière Télécommunication, Réseaux et Services
  Module TRS5 : Les Protocoles de Télécommunications/Évolution Internet.
  Depuis 1999, j'interviens en DEA et suis responsable de ce module :

  > Le module présente les évolutions relatives à l'Internet et les travaux de recherche en cours dans ce domaine : description des algorithmes utilisés dans les protocoles de routage unicast (vecteur de distance, état de liaison), concept de communication de groupe (IP Multicast, Mbone, SSM).
  > L'intégralité de ce module (18H) a été reprise dans une unité d'enseignement d'ossature dans le cadre du master recherche Services Distribués et Réseaux de communication.

  Depuis la rentrée 2005, je suis également co-responsable dans le master recherche d'une unité d'enseignement de différenciation intitulée Sécurité des Réseaux Dynamiques où je présente la sécurité dans les communications de groupe et la sécurité dans les réseaux ad hoc (6h).

- 3ème année ESIAL, Université Henri Poincaré, Nancy 1 :
  Depuis 2001, j'interviens et suis responsable de trois modules.
  Approfondissement et Expérimentation des Protocoles Réseaux

  > Ce module (15h CM, 15 TP) que j'ai entièrement monté aborde les protocoles de routage dynamique utilisés dans l'Internet (RIP, OSPF, BGP), le protocole Internet Nouvelle Génération IPv6 et leur mise en œuvre sur une plate-forme d'expérimentation.

  Evolution et Services dans les Réseaux

  > Dans ce module de 30h (18h CM, 12 TP) dont je suis à l'initiative, j'interviens pour une partie (6h CM, 9h TP) relative aux communications de groupe. Je fais

également intervenir des extérieurs pour présenter des domaines avancés (Pair à Pair, la Mobilité IP, UMTS).

Sécurité des Réseaux et des Systèmes

Dans ce module, j'interviens pour présenter l'étude du protocole de sécurité IPsec et de la distribution des clés avec ISAKMP/IKE (3h CM, 3h TP).

– DESS ISIAL, Université Henri Poincaré, Nancy 1 :

Depuis 2002, j'interviens pour assurer le cours sur le *Protocole IPv6* (3h CM).

– 3ème année E.S.S.I. de 1993 à 1997 :
Cours Synchronisation des Applications Distribuées (3h CM)

Ce cours présente les différentes techniques pour synchroniser des applications distribuées : validation à deux phases, utilisation de sémaphores réseaux, ordonnancement réparti.

Réseaux Couches Hautes : ASN.1, X500, Synchronisation, Sécurité (16h TPs)
Administration Réseaux (8h TP)
Cours Evolution des architectures (1h30 CM)

Ce cours présente les limites du modèle en couches et décrit l'architecture ALF (Application Level Framing).

Enseignement en 2ème cycle

– 2ème année ESIAL, Université Henri Poincaré :
Module Réseaux et Systèmes en 2ème année

Dans ce module, j'interviens en système (12h TD) sur les aspects synchronisation, interblocage, lecteur/rédacteur. Je présente également une introduction aux réseaux à savoir les architectures, le modèle en couches, les réseaux locaux, les protocoles TCP/IP (10h CM, 14h TD/TP).

Module Systèmes, Réseaux et Télécommunications

Dans ce module dont je suis responsable, je présente X25, ATM, Frame Relay, ainsi qu'un approfondissement du protocole TCP et le développement des applications réseaux en C avec l'utilisation des sockets et de la programmation pour des applications de groupe (10h CM, 14h TD/TP).

– 2ème année E.S.S.I. de 1993 à 1997 :
Module Introduction aux Réseaux (4h CM, 6h TPs)

Le cours présente les architectures OSI/ISO et Internet. Il décrit également les différents types d'applications rencontrées sur les réseaux : session à distance, transferts de fichier, WEB, applications multimedia.

Programmation Réseaux sous UNIX (3h CM, 8h TPs)

Ce cours présente le développement des applications réseaux en C avec l'utilisation des sockets.

Système d'exploitation UNIX. (12h TPs)

Travaux pratiques relatifs au système de gestion de fichiers et à la communication inter-processus (signaux, pipes, sémaphores).

Enseignement en 1er cycle

- I.U.T Nancy-Verdun
  De 1997 à 2000, j'ai été responsable du module Réseaux à l'IUT pour les étudiants de deuxième année et ceux d'année spéciale. Je suis intervenue à la fois au niveau du cours magistral et des TD/TPs :
  Module Réseaux (18h CM, 60h TD/TPs)

  > Introduction aux réseaux à savoir les architectures, le modèle en couches, les réseaux locaux, les protocoles TCP/IP.

  Système UNIX (2ème année) (39h TD/TPs)

  > Présentation des outils Unix pour développer en C, du système de gestion de fichiers et de la communication inter-processus (signaux, pipes, sémaphores).

  Module Assembleur (40 TD/TPs)

  > Présentation des concepts pour la programmation via l'assembleur du processeur 8086 : adressage indirect, variables locales, globales,...

Enseignement donné pour des entreprises

- Formation Continue : Dans le cadre de l'IUT, je suis intervenue pour une formation continue en 1998-1999 sur un cours et des TPs d'introduction aux réseaux (10 heures)
- Air France : En 1995, dans les locaux de Air France à Sophia Antipolis, j'ai effectué un cours de programmation réseaux sous Unix (3 h) et des TPs de programmation sockets (6 h)

Interventions ponctuelles

- IUP Réseaux, Université Henri Poincaré, Nancy 1 :
  1999-2000 *Cours Protocole IPv6* (3h CM)
- École des Mines de Nancy :
  1997-1998 *Cours Protocole IPv6* (3h CM)
- École des Mines d'Alès (EMA)
  Fin Janvier 1997 *Cours Introduction aux Applications Réparties* (3h CM)

  > Ce cours présente les besoins d'une application distribuée : gestionnaire de noms, synchronisation, tolérance aux fautes et sécurité.

Supports de Cours

- Polycopiés
  Système UNIX co-écrit avec B. Mangeol, J.F. Mari et D. Roegel. Ce support contient une série d'exercices avec les corrigés (165 pages).
  Les Réseaux co-écrit avec L. Andrey, B. Mangeol, J.F. Mari et E. Nataf. Ce support contient une série d'exercices avec les corrigés (119 pages).
- Transparents
  Cours Introduction Réseaux : 280 pages.
  Cours Réseaux Avancés : 230 pages.
  Cours IPv6 : 70 pages.

Cours Routage (RIP, OSPF, BGP) : 140 pages.
Cours Routage Multicast : 140 pages.
Cours Sécurité et Multicast : 155 pages.

Encadrements de projets

Dans le cadre de mon enseignement, j'encadre chaque année, depuis 2001 un groupe de projet industriel effectué par les étudiants de 3ème année ESIAL. Le projet industriel permet de sensibiliser les étudiants aux problèmes concrets des entreprises en travaillant sur des sujets proposés par des industriels (250h par étudiant).

J'encadre également chaque année un ou deux groupes d'étudiants ESIAL de 2ème année sur des projets interdisciplinaires ou de découverte de la recherche (80h par étudiant).

En tant que Maître de Conférences à l'IUT, j'ai proposé et supervisé trois projets tutorés (50h par étudiant).

Publications

Journaux, livres et chapitres de livres	7
Conférences et workshops internationaux	14
Conférences francophones	10
Cours, Tutoriels	3
Présentation invitée	1
Rapports de recherche, rapports de contrats, drafts	14

Journaux, livres et chapitre de livres

[1] G, Chaddoud, V. Varadharajan, I. Chrisment, and A. Schaff. Gestion efficace de la sécurité des communications de groupe pour le service SSM. *Journal TSI n° spécial Réseaux et Protocoles*, 23(9) :1107-1135, 2004.

[2] L, Andrey, I. Chrisment, O. Festor, and E. Fleury. *Traité IC2, Systèmes multimédia communicants.* Rédaction du chapitre « Infrastructures pour le multimédia : ALF et les réseaux actifs ». Hermès Science, 2001, ISBN 2-7462-0251-4.

[3] I. Chrisment and O. Festor. *Logiciels et réseaux de communication, Observatoire Français des Techniques Avancées.* Rédaction d'un chapitre intitulé « Réseaux programmables et Réseaux actifs ». ARAGO 23, pages 199-211, Paris, Mai 2000.

[4] G. Cizault. *IPv6 : théorie et application, 2ème édition.* Rédaction avec O. Festor du chapitre sur la supervision de piles IPv6. O'Reilly, Paris, 1999, ISBN 2-84177-085-0

[5] I. Chrisment, D. Kaplan, and C. Diot. An ALF Communication Architecture : Design, Automated Implementation. *IEEE Journal of Selected Area in Communications*, 16(3) :332-344, April 1998.

[6] I. Chrisment and C. Huitema. Evaluating the Impact of ALF on Communication Subsystems Design and Performance. *Journal of High Speed Networks*, 5(2) :173-180, 1996.

[7] T. Braun, I. Chrisment, C. Diot, F. Gagnon, and L. Gautier. ALF/ILP Based Automated Implementation of Distributed Applications. *Australian Computer Journal*, 28(2) :48-54, May 1996.

Conférences et workshops internationaux

[8] M.S. Bouassida, I. Chrisment, and O. Festor. Efficient Clustering for Multicast Key Distribution in MANETs. *Fourth IFIP-TC6 Networking (Networking'05)*, LNCS, Springer-Verlag, Vol. 3462, pages 138-153, Waterloo, Canada, May 2005 (acceptance rate 24,7%).

[9] M.S. Bouassida, I. Chrisment, and O. Festor. An Enhanced Hybrid Key Management Protocol for Secure Multicast in Ad hoc Networks. *Third IFIP-TC6 Networking (Networking'04)*, LNCS, Springer-Verlag, Vol. 3042, pages 725-742, Athens, Greece, May 2004 (acceptance rate 19,1%).

[10] R. State, O. Festor, and I. Chrisment. Context Driven Access Control to SNMP MIB Object in Multi-Homed Environment. *IFIP/IEEE Workshop on Distributed Systems Operations and Management (DSOM'03)*, Heidelberg, Germany, October 2003, 12p (acceptance rate 23,3%).

[11] A. Benharref, R. Dssouli, Z. Berbich, and I. Chrisment. Formal Specifications, TTCN and Executable Test Cases for main IPv6 Protocols. *IEEE International Workshop on Communication Software Engineering (IWCSE'02)*, Marrakech, Morocco, December 2002, 4p.

[12] G. Chaddoud, I. Chrisment, and A. Lahmadi. A Secure SSM Architecture. *IEEE International Conference on Network (ICON'02)*, Singapore, August 2002, 13p.

[13] G. Chaddoud, I. Chrisment, and A. Schaff. Dynamic Group Communication Security. *The 6th IEEE Symposium on Computers and Communications (ISCC'01)*, Hammamet, Tunisia, July 2001, 8p.

[14] G Chaddoud, I. Chrisment, and A. Schaff. Dynamic Group Key Management Protocol. *Mathematical Methods, Models and Architectures for Computer Networks Security International Workshop (MMM-ACNS'01)*, St Petersburg, Russia, May 2001, 12 p.

[15] G. Chaddoud, I. Chrisment, and A. Schaff. Secure Multicasting Survey. *IFIP World Computer Congres, (SEC'00)*, Beijing, China, August 2000, 4p.

[16] T. Braun, I. Chrisment, C. Diot, F. Gagnon, and L. Gautier. ALFred, a Protocol Compiler for the Automated Implementation of Distributed Applications. *HPDC 5 Symposium, IEEE press*, Syracuse, 6-9 August 1996.

[17] T. Braun, I. Chrisment, C. Diot, F. Gagnon, and L. Gautier. The HIPPARCH approach to ALF/ILP based Automated Implementation of Distributed Applications. *First Workshop on Compiler Support for Systems Software*, Tucson, Arizona, February 1996.

[18] T. Braun, I. Chrisment, C. Diot, F. Gagnon, L. Gautier, and P. Hoschka. ALFred, an ALF/ILP Protocol Compiler for Distributed Applications. *Second International Workshop on High Performance Protocol Architecture*, Sydney, Australia, December 1995.

[19] C. Diot, I. Chrisment, and A. Richards. Application Level Framing and Automated Implementation. *Sixth International Conference on High Performance Networking*, Palma, Spain, September 1995.

[20] I. Chrisment. Impact of ALF on Communication Subsystems Design and performance. *First International Workshop on High Performance Protocol Architecture*, Sophia Antipolis, France, December 1994.

[21] I. Chrisment and C. Huitema. Remote Operation System Tailored to Application Requirements. *IFIP International Conference (ULPAA'94)*, pages 29-43, Barcelona, Spain, June 1994.

Conférences francophones

[22] M.S. Bouassida, A. Bruneton, A. Lahmadi, I. Chrisment, and O. Festor. Balade : diffusion multicast sécurisée d'un flux multimédia multi-sources séquentielles dans un environnement ad hoc. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'05)*, Bordeaux, France, Mars 2005, 17p (taux d'acceptation 30%).

[23] M.S. Bouassida, I. Chrisment, and O. Festor. Prise en compte de la mobilité dans le protocole de gestion de clé de groupe BALADE. *Sécurité et Architecture Réseaux 2005 (SAR'05)*, Batz sur Mer, France, Juin 2005, 12p.

[24] M.S. Bouassida, I. Chrisment, and O. Festor. Méthodes d'authentification pour les communications de groupes : taxonomie et évaluation dans un environnement ad hoc. *Sécurité et Architecture Réseaux 2004 (SAR'04)*, La Londe, France, Juin 2004, 12p.

[25] H. Sallay, A. Lahmadi, O. Festor, and I. Chrisment. Extension de l'architecture active AMAN pour le support des services de sécurité mulicast. *Colloque Francophone sur la Gestion de Réseaux et de Services (GRES'03)*, Fortaleza, Brésil, 2003, 13p.

[26] L. Ciarletta and I. Chrisment. Outils pour l'expérimentation des technologies de l'informatique ambiante. *16ème Congrès DNAC : De nouvelles architectures pour les communications (DNAC'02)*, Paris, France, Décembre 2002.

[27] G. Chaddoud, I. Chrisment, and A. Lahmadi. S-SSM : sécurisation d'une architecture SSM. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'02)*, Montréal, Canada, Mai 2002, 14p.

[28] G. Chaddoud, I. Chrisment, and A. Schaff. Baal : sécurisation des communications de groupes dynamiques. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'00)*, Toulouse, France, Octobre 2000, 15p.

[29] S. D'Alu, G. Chellius, I. Chrisment, O. Festor, and E. Fleury. Intégration du support IPv6 dans l'environnement de supervision de réseaux actifs ANAIS. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'00)*, Toulouse, France, Octobre 2000, 16p.

[30] L. Andrey, I. Chrisment, O. Festor, et E. Fleury. Supervision et contrôle dans les réseaux actifs ; une nécessité à la mise en œuvre et au déploiement dans les réseaux de télécommunications. *Colloque Francophone sur la Gestion de Réseaux et de Services (GRES 1999)*, Montreal, Canada, 1999, 14p.

[31] I. Chrisment, C. Diot, et C. Huitema. Génération automatique de protocoles de communication pour les applications multimédia. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'96)*, Rabat, Maroc, Octobre 1996.

Tutoriels

[32] A. Bouabdallah and I. Chrisment. IP Multicast Security *Internet Nouvelle Génération (ING'04)*, Obernai, France, Juin 2004.

[33] G. Chaddoud and I. Chrisment. La sécurité et les communications de groupe. *Sécurité et Architecture Réseaux 2002 (SAR'02)*, Marrakech, Maroc, Juillet 2002.

[34] O. Festor, I. Chrisment, and E. Fleury. Les réseaux programmables. *Tutoriel de l'Ecole d'été RHDM'99*, Bretagne, France, Septembre 1999.

Présentation invitée

[35] L. Ciarletta, A. Aït Ali, and I. Chrisment. VPSS : Architecture de Sécurité pour l'Informatique Ambiante. *Sécurité et Architecture Réseaux 2002 (SAR'02)*, Marrakech, Maroc, Juillet 2002.

Rapports de recherche, rapports de Contrats, drafts

[36] M. Adib, A. Bouabdallah, M.S. Bouassida, I. Chrisment, H. Ragab, A. Serhrouchni. Analyse des algorithmes de cryptographie multicast. Rapport de contrat SAFECAST, Mai 2005, 22 p.

[37] A. Bouabdallah, M.S. Bouassida, I. Chrisment, H. Ragab. Etat de l'art des protocoles de gestion de clés dans les communications de groupe Rapport de contrat SAFECAST, Mai 2005, 47 p.

[38] A. Bouabdallah, M.S. Bouassida, I. Chrisment, H. Ragab. Définition d'un protocole de gestion de clés. Rapport de contrat SAFECAST, Mai 2005, 16 p.

[39] A. Bouabdallah, M.S. Bouassida, Y. Challal, and I. Chrisment. Etat de l'art des protocoles d'authentification dans les communications de groupe. Rapport de contrat SAFECAST, LORIA n^o A04-R-250, Octobre 2004, 24 p.

[40] M.S. Bouassida, A. Lahmadi, I. Chrisment, and O. Festor. Diffusion multicast sécurisée dans un environnement ad hoc (1 vers n séquentiel). Rapport de recherche INRIA n° 5310, LORIA n^o A04-R-045, Septembre 2004, 47 p.

[41] M.S. Bouassida, I. Chrisment, V. Guyot, V. Legrand, D. Raffo, and S. Ubeda. Sécurité et réseaux ad hoc. Rapport de contrat SAFARI, LORIA n^o A04-R-045, Avril 2004, 35 p.

[42] A. Benharref, Z. Berbich, M.S. Bouassida, R. Dssouli, and I. Chrisment. Formalisation et test d'IPv6. Rapport de contrat Fonds France Canada, LORIA n^o A03-R-465, 2003, 21p.

[43] G. Chaddoud, I. Chrisment, and A. Schaff. A Secure SSM Architecture. *draft-irtf-gsec-sssm-00.txt*, February 2002.

[44] A. Lahmadi, G. Chaddoud, and I. Chrisment. Implémentation d'un prototype du protocole Baal. Rapport de recherche LORIA n^o A01-R-393, 2001, 31 p.

[45] S. D'Alu, O. Festor, I. Chrisment, and E. Fleury. Active Network Encapsulation Protocol (ANEP) Extension for IPv6. *draft-sdalu-anep-ipv6-00.txt*, November 2000, 3p.

[46] G. Chaddoud, I. Chrisment, and A. Schaff. Vers une communication de groupe sécurisée : état de l'art et perspectives. Rapport de recherche LORIA n° 99-R-244, 1999, 19 p.

[47] O. Festor, I. Chrisment, and E. Fleury. Les réseaux programmables. Rapport de recherche INRIA n° 99-R-100, 1999, 48 p.

[48] T. Braun, I. Chrisment, C. Diot, L. Gautier et P. Hoschka. ALFred, an ALF/ILP Protocol Compiler for Distributed Applications Automated Design. Rapport INRIA n° 2786, January 1996.

[49] C. Castelluccia, I. Chrisment, W. Dabbous, C. Diot, C. Huitema, and R. de Simone. Tailored Protocol Development Using Esterel. Rapport INRIA n° 2374, October 1994.

1

Introduction

1.1 Contexte

L'objectif de ce mémoire est de présenter l'ensemble des travaux que j'ai effectués depuis la rédaction de ma thèse en Juin 1996, thèse réalisée à l'INRIA Sophia Antipolis dans l'équipe RODEO sous la direction de Christian HUITEMA.

Mes travaux de recherche ont commencé en 1993. Il s'agissait de définir et de spécifier des protocoles mieux adaptés aux applications modernes que le protocole RPC (*Remote Procedure Call*) [Nel81] classique, largement utilisé dans le développement des applications distribuées de type client-serveur. L'idée de base avec RPC est relativement simple ; elle vise à étendre la notion d'appels de procédures à distance aux programmes ayant des espaces d'adressage différents et communiquant via un réseau. Cependant si le concept RPC est relativement facile à mettre en place, il n'en présente pas moins des limites tant au niveau des applications cibles qu'au niveau des performances obtenues.

J'ai ainsi été amenée à étudier le paradigme ALF (Application Level Framing) qui remet en cause l'approche classique du modèle en couches en proposant d'adapter le système de communication aux besoins des applications et en permettant d'intégrer la partie contrôle de transmission dans l'application. Le sujet de ma thèse a porté sur l'*"étude et le développement d'applications distribuées dans l'architecture ALF"* [Chr96]. Ces travaux se sont inscrits dans le cadre d'un projet australo-européen HIPPARCH (HIgh Peformance Protocol ARCHitecture) et ont duré jusqu'en 1997, date à laquelle j'ai intégré, en tant qu'enseignant-chercheur, l'équipe RESEDAS dirigée par André SCHAFF.

En 2002, j'ai contribué à la création de l'équipe MADYNES, devenue officiellement projet INRIA en Janvier 2004. J'assure les fonctions de responsable permanent de ce projet dont le responsable scientifique est Olivier FESTOR.

1.2 Contributions

Ce manuscrit décrit les recherches que j'ai menées au sein du LORIA (Laboratoire Lorrain de Recherche en Informatique et ses Applications) à Nancy depuis 1997. L'ensemble des travaux est le résultat d'une collaboration très forte avec les stagiaires, doctorants et ingénieurs que j'ai co-encadrés avec André SCHAFF et Olivier FESTOR. Certaines des contributions scientifiques présentées sont aussi issues d'échanges et de coopérations réalisés avec des équipes de recherche tant au niveau national qu'international.

Mes axes de recherche sont structurés suivant trois thèmes majeurs qui sont synthétisés dans

la figure 1.1 : les réseaux actifs, la sécurité dans les communications de groupe et la sécurité dans les réseaux spontanés.

	Problématique	Thématique	Contributions
93–97	Adaptation des protocoles de transmission aux applications	ALF	Travail de thèse : Etude et développement d'applications distribuées dans l'architecture ALF
98–00	Adaptation des protocoles réseaux aux applications	Réseaux actifs	– Supervision et contrôle (ANAIS) – Tests IPv6
99–02	Dynamicité des services et des réseaux	Sécurité dans les communications de groupe	– Baal – Extension IGMP Proxying – S–SSM – S–SSM et DCCP
Depuis 2002	Evolution vers le monde sans fil	Sécurité dans les réseaux spontanés	– VPSS – Adaptation de Baal aux réseaux ad hoc – BALADE – Evaluation des méthodes d'authentification pour les flux multicast

FIG. 1.1 – Résumé des contributions

Réseaux actifs

Mes connaissances dans l'architecture ALF m'ont conduite à m'intéresser tout naturellement au paradigme des réseaux actifs qui en est une suite logique. Les deux modèles ont d'ailleurs été proposés par TENNENHOUSE à quelques années d'intervalle [CT90] [TW96]. L'approche réseaux actifs pousse à l'extrême le concept d'adaptation. Avec ALF les applications doivent prendre en compte les conditions du réseau et intégrer les éléments protocolaires répondant à leurs besoins. Avec les réseaux actifs, les équipements réseaux, qui étaient jusqu'à présent fermés et où les protocoles étaient jusqu'alors définis une fois pour toute, doivent s'ouvrir afin de prendre en compte les besoins applicatifs. Les réseaux actifs proposent d'introduire de la "programmabilité" dans l'infrastructure réseau elle-même, afin de permettre d'enrichir dynamiquement l'offre des services et le déploiement de nouveaux protocoles [WLG98].

Dans les travaux réalisés, nous avons regardé plus spécifiquement quels pouvaient être les apports des réseaux actifs dans deux domaines : celui de la supervision et celui des tests de protocoles et plus particulièrement du protocole IPv6.

Pour la gestion de réseaux, le concept paraît particulièrement intéressant car il peut servir de support pour faciliter le déploiement et changer la vision traditionnelle client/serveur des approches de gestion.

Le protocole IPv6 représente également un domaine d'application privilégié car il fait partie de ces "nouveaux protocoles" qui intègrent dès la conception des aspects dynamiques avec la mobilité IP, la communication de groupe, l'auto-configuration. Ces aspects dynamiques rendent difficile l'utilisation des méthodes classiques de tests de protocoles plus adaptées à des environnements statiques.

Sécurité des communications de groupe

Les réseaux actifs correspondent à une première réponse à la découverte du caractère dynamique des réseaux et des services offerts aux usagers. Il y a eu, en effet, une montée en puissance de cette dynamicité ces dernières années avec la convergence du monde fixe et du monde mobile, du monde des télécommunications et du monde IP. Il est indéniable que pour faire face à la concurrence, les opérateurs agissent sur les services à valeur ajoutée afin de répondre le plus rapidement aux clients et permettre un déploiement rapide des services orientés vers l'utilisateur.

Se pose alors le problème de gestion de ces réseaux dynamiques. La gestion des réseaux est découpée en cinq aires fonctionnelles : gestion de fautes, configuration, facturation, gestion de performances, sécurité. Un axe crucial sur lequel nous nous sommes focalisés est celui de la sécurité car les mécanismes standards se montrent souvent inadaptés aux environnements contraints et soumis à une forte dynamique. Pourtant, le déploiement et l'acceptation de ces réseaux ne peut se réaliser sans l'offre d'un service de sécurité.

Les communications de groupe constituent un domaine d'application par excellence car, en plus de l'aspect dynamique lié aux membres (les participants peuvent joindre et quitter le groupe à tout moment), elles soulèvent le problème d'extensibilité, c'est-à-dire, de passage à l'échelle dans le cas de grands groupes.

Pour faire face à ces défis et offrir des services de sécurité requis pour les communications de groupe, nous avons proposé des protocoles de gestion des clés cryptographiques adaptés. Ces protocoles de distribution de clés permettent d'assurer la confidentialité des données, l'authentification et le contrôle d'accès aux seuls membres du groupe tout en offrant un temps minimal de configuration et un trafic réduit.

Sécurité dans les réseaux spontanés

Les réseaux spontanés facilitent l'interconnexion d'entités avec peu ou pas d'infrastructure préalable ; leurs principales caractéristiques sont l'auto-configuration et l'auto-organisation. Ils représentent le socle de l'informatique dite ambiante ou ubiquitaire, c'est-à-dire complétement transparente et intégrée dans notre quotidien.

Parmi ce type de réseaux on peut citer les réseaux ad hoc qui sont définis comme une collection de nœuds mobiles communiquant par une technologie sans fil et formant un réseau temporaire sans l'aide de toute administration ou de tout support fixe. Le développement de ces réseaux spontanés s'explique, outre leur facilité d'usage, par l'explosion ces dix dernières années des réseaux sans fil et des terminaux autonomes de plus en plus puissants.

La combinaison d'une infrastructure ad hoc avec des services offrant des communications de groupe soulève de nouveaux challenges de sécurité vers lesquels nous nous sommes

tournés. Parmi ces challenges, l'absence d'infrastructure et de serveur rend inopérants les architectures et protocoles de sécurité retenus dans le monde filaire. Nous avons donc proposé des solutions pour assurer la distribution des clés nécessaires à la sécurité des échanges dans un tel contexte.

1.3 Organisation du mémoire

Ce mémoire se compose de trois principales parties qui correspondent chacune à un ensemble de contributions sur chaque thème présenté dans la section précédente.

La première partie est consacrée principalement à la liaison entre mes travaux de thèse et ceux effectués quand je suis arrivée dans le projet RESEDAS. Elle est relative à l'adaptation des protocoles de transport et de réseaux aux besoins des applications et explique l'évolution de ma recherche de ALF vers les réseaux actifs.

La seconde partie présente mes travaux relatifs à la sécurité dans le cadre des communications de groupe dans l'Internet.

La troisième partie décrit mes travaux concernant la problématique de la sécurité dans les réseaux spontanés et plus spécifiquement la sécurité des communications de groupe dans un environnement ad hoc.

La quatrième partie constitue la conclusion du mémoire et détaille mes perspectives de recherche.

Bibliographie

[Chr96] I. Chrisment. *Étude et développement d'applications distribuées dans l'architecture ALF*. PhD thesis, Université de Nice-Sophia Antipolis, Juin 1996.

[CT90] D.D Clark and D.L. Tennenhouse. Architectural Considerations for a New Generation of Protocols. In *ACM SIGCOMM on Communications Architectures, Protocols and Applications*, pages 200–208, Philadelphie, PA, September 1990.

[Nel81] B.J. Nelson. *Remote Procedure Calls*. PhD thesis, Carnegie Mellon University (CMU), May 1981.

[TW96] D.L. Tennenhouse and D.J. Wetherall. Towards an Active Network Architecture. *Computer Communication Review*, 26(2) :5–17, 1996.

[WLG98] D.J. Wetherall, U. Legedza, and J. Guttag. Introducing New Internet Services : Why and How. *IEEE Network Special Issue : Active and Programmable Networks*, 12(3) :12–19, 1998.

2

Adaptation des protocoles : ALF et les réseaux actifs

2.1 Introduction

Le développement rapide des moyens de télécommunications a favorisé l'essor de nouvelles catégories d'applications distribuées (vidéo conférence, bases de données multimédia,...) ayant des besoins en communication très variés, notamment en termes de débit, de fiabilité, de taux d'erreurs acceptables, de latence et de gigue. Les systèmes de communication en couches tels qu'ils ont été définis dans les dernières décennies (TCP/IP, OSI TP4, RPC [Nel81]) s'avèrent mal adaptés pour faire face à une telle diversité. Ils ont été conçus pour fournir aux applications le maximum de transparence. Ces protocoles sont seuls responsables des aspects de communication et offrent aux applications un ensemble générique de services appelé interface. Du point de vue de l'application, le système de communication est considéré comme une simple boîte noire. Mais, si cacher les détails de communication simplifie l'application, cela se fait malheureusement le plus souvent au détriment des performances et de la flexibilité.

Jusqu'à une période récente, les performances des systèmes d'extrémité (stations de travail) ne constituaient pas un problème. Le goulot d'étranglement se situait au sein du réseau lui-même. Avec l'émergence des réseaux à haut débit, les ordinateurs connectés ne sont plus capables de traiter les données aussi vite que le réseau. Si la vitesse des processeurs double chaque année, il n'en est pas de même des performances de la mémoire et des bus internes. Le goulot d'étranglement s'est donc déplacé initialement du réseau vers les machines terminales et principalement vers les piles de protocoles. La limitation des performances des systèmes de communication s'explique avant tout par leur aspect générique. Ces protocoles souffrent de fonctionnalités excessives parfois redondantes ou sans utilité pour l'application. Généralement, une application ne peut choisir qu'entre un protocole totalement fiable et ordonné et un protocole n'offrant aucune garantie. Ainsi le protocole TCP propose un service fiable et ordonné même si une application peut se satisfaire de données fiables mais arrivant dans le désordre.

Le manque de flexibilité est dû à la rigidité des systèmes traditionnels. Leurs fonctionnalités sont écrites pour s'interfacer avec des applications non connues par avance et ne correspondent donc pas toujours aux besoins réels des applications. L'introduction de contraintes temporelles sur les flux de données à transmettre définit de nouvelles fonctionnalités qui compliquent davantage l'architecture du système de communication et le rendent encore plus délicat à implanter.

L'utilisation d'un protocole de contrôle standard est trop rigide devant la variété des contraintes qui s'appliquent aux informations à acheminer.

Les limites du modèle traditionnel en couches [CWW92] ont conduit, dans le début des années 90, à vouloir considérer de nouvelles approches pour la conception de systèmes de communication performants. Clark et Tennenhouse [CT90] ont défini un nouveau modèle de protocoles appelé ALF (*Application Level Framing*) qui permet d'inclure la sémantique des données de l'application dans la conception du protocole de l'application. Cette philosophie est maintenant utilisée pour des protocoles tels RTP (*Real-Time Protocol*) [SCFJ03] à travers l'Internet.

Mes travaux de thèse [Chr96] ont porté principalement sur cette architecture ALF dont est rappelé ici le concept dans une première partie ainsi que la principale contribution qui en a résulté.

Dans cette continuité, lors de mon arrivée dans l'équipe RESEDAS, nous avons étudié le paradigme d'Active Networks (ActiveNet) ou Réseaux Actifs présenté par Tennenhouse et Wetherall en 96 [TW96], qui représentait une suite logique de ALF, tout en allant plus loin, en autorisant l'application serveur ou client à envoyer du code correspondant à ses besoins dans les nœuds du réseau.

Avec l'idée d'actif, le réseau devient alors partie intégrante du système de communication et n'est plus considéré comme une simple boîte noire par les applications. Les réseaux peuvent être actifs de deux manières :

- les routeurs et les commutateurs agissent sur les données ;
- les utilisateurs programment les éléments du réseau en leur envoyant des bouts de code afin de réaliser des calculs (ex : exécution d'un algorithme de compression durant le traitement des paquets).

L'approche ALF et notre proposition d'un environnement automatisé sont détaillées dans la Section 2.2 de ce chapitre. Dans la Section 2.3 sont présentées nos principales contributions relatives aux réseaux actifs dans le domaine de la supervision et des tests IPv6. Nous terminons ce chapitre par une conclusion.

2.2 Le concept ALF

2.2.1 Contexte

Le concept ALF (*Application Level Framing*) remet en cause l'approche classique du modèle traditionnel en couches ; il est basé sur le principe que l'application est la mieux informée sur le type de données échangées et les services dont ces données ont besoin. Par conséquent, l'application est la mieux à même de définir et de contrôler certains paramètres de transmission comme la fiabilité nécessaire, le type de reséquencement, le contrôle de flux approprié. Par exemple, elle seule est capable de connaître les stratégies à adopter en cas de pertes (faut-il ignorer les pertes ou exiger la retransmission ?) ou de données arrivant dans le désordre (faut-il recevoir en séquence ou non ?).

Mais pour appliquer le concept ALF, les données manipulées par l'application doivent être exprimées en termes significatifs pour le protocole. Ce qui n'est pas possible dans les systèmes de communication traditionnels où une barrière très nette est établie entre les couches application, présentation et transmission. Chaque couche possède sa propre unité de manipulation de données. TCP numérote par exemple les octets par rapport au flux de données et utilise ces numéros pour réaliser le reséquencement et la retransmission. Cette numérotation n'a, cependant, aucun sens pour l'application.

ALF propose une stratégie qui organise les données en trames logiques encore appelées *frames* ou ADU (*Application Data Units* ou *Unités de Données Applicatives*) significatives à la fois pour l'application et le protocole. L'ADU est construite par l'application et est aussi l'unité de contrôle au niveau du protocole transport et l'unité de transmission au niveau du réseau. ALF autorise ainsi une grande flexibilité dans le contrôle des données. L'application peut traiter potentiellement chaque ADU de manière indépendante. L'émetteur doit fournir des informations suffisantes au niveau de l'ADU afin qu'elle puisse être directement et immédiatement utilisée par le récepteur (par exemple, dans le cas d'une transmission d'images, la position relative d'un pixel sur l'écran pourra être précisée).

L'idée maîtresse de ALF est de laisser l'application contrôler elle-même certains mécanismes de transmission, notamment les données perdues ou en désordre. Avec ALF, nous quittons l'architecture classique où tous les aspects de communication sont encapsulés dans une boîte noire. Nous nous orientons vers une architecture où l'application dispose d'informations relatives aux caractéristiques de communication comme les pertes, le ré-ordonnancement, les délais. Mais y a-t-il encore des pertes et du désordre dans les réseaux ? Si dans les réseaux locaux, les taux de pertes sont faibles, à une échelle plus grande, nous pouvons répondre par l'affirmative du fait notamment de l'hétérogénéité de plus en plus grande des réseaux (mobiles, fibres optiques...). Les applications distribuées, s'exécutant au dessus de réseaux à commutation par paquets doivent en effet s'attendre, ou tout du moins le système de communication sur lequel elles reposent, à faire face à des pertes de paquets et à les recevoir en désordre. La perte de paquets s'explique par des risques de congestion au niveau du réseau et des routeurs lorsqu'il y a un accroissement du trafic. La congestion oblige certains paquets à être écartés et à ne pas être délivrés à leur destinataire. À moins d'être prêt à payer le prix requis pour obtenir la bande passante nécessaire, les utilisateurs devront se partager la bande passante existante (politique du « *best-effort* ») et le problème de congestion sera toujours persistant. Le désordre des paquets peut être une conséquence de la perte des paquets. Il est alors causé par les retransmissions effectuées par le protocole de transport. Le désordre peut aussi s'expliquer par l'implantation de techniques de routage comme le routage multi-chemins [Hui94] qui répartit le trafic sur plusieurs routes. Ceci amène à délivrer les paquets dans le désordre à la destination, car les files d'attente des nœuds intermédiaires peuvent avoir des tailles différentes. Le routage multi-chemins repose sur les deux analyses suivantes : plusieurs routes permettent d'atteindre le même destinataire et il est plus efficace de répartir le trafic sur plusieurs chemins [JML92, JMG93].

2.2.2 Motivation

Les principales motivations d'une architecture comme ALF peuvent se résumer en trois points : offrir une plus grande flexibilité à l'application, augmenter les performances dans le cas de pertes et faciliter l'implantation de certaines optimisations.

- Amélioration de la flexibilité. Un aspect intéressant de ALF, et peut-être un des plus importants, est de vouloir être suffisamment flexible pour s'adapter à la diversité des besoins des nouvelles applications. L'architecture ALF doit permettre en effet à l'application de déterminer sa propre réponse lorsque des données sont perdues, en désordre ou arrivent en retard. La caractéristique d'une ADU est de contenir assez d'informations pour être traitée immédiatement par l'application. La stratégie de traitement de cette ADU sera déterminée par l'application elle-même. Ainsi la stratégie adoptée pour une ADU relative à une application vidéo sera différente de celle adoptée pour une ADU relative à une application audio : leurs besoins en termes de communication n'étant pas les mêmes. De plus, à l'intérieur d'une application, il peut s'avérer utile de distinguer plusieurs types

d'ADU pour leur appliquer des contrôles différents.

- Amélioration des performances. Une autre motivation pour ALF est de vouloir améliorer les performances des applications. en les autorisant à contrôler elles-mêmes leurs données perdues ou en désordre. Le fait qu'un protocole ait sa propre unité de manipulation, différente de l'application, peut en effet imposer des délais d'attente inutiles du point de vue de l'application. Ainsi, dans les protocoles ordonnés et fiables comme TCP, les données reçues dans le désordre sont bufferisées. Quand les données manquantes arrivent, l'ensemble du buffer est délivré en rafale à l'application. Il peut en résulter une congestion au niveau du récepteur. Le traitement hors séquence peut améliorer de façon significative l'occupation des buffers ainsi que la gigue [DG99].
  Le choix de la taille des ADUs peut faire varier considérablement les performances. Les meilleurs résultats sont obtenus quand l'ADU est la plus grande unité de manipulation qui peut-être traitée dans le désordre sans fragmentation à travers le réseau [Chr94]. D'où l'intérêt pour les applications de s'adapter au minimum MTU[1] sur le chemin et d'éviter la fragmentation. Dans IP nouvelle génération, IPv6 [Hui97, Ciz02], les machines hôtes doivent apprendre la taille du minimum MTU via une procédure appelée *Path MTU Discovery* [MD90] car les routeurs intermédiaires refusent de fragmenter les données trop grandes.
- Optimisations. L'architecture ALF est un élément clé pour appliquer des techniques d'optimisation comme *Integrated Layer Processsing* ou ILP [CT90]. ILP est une technique d'implantation qui se propose de réaliser, en une seule boucle de traitements (boucle ILP), les différentes fonctions de manipulation de données qui sont généralement effectuées en plusieurs boucles, de manière séquentielle, dans les différentes couches comme illustré dans la figure 2.1.

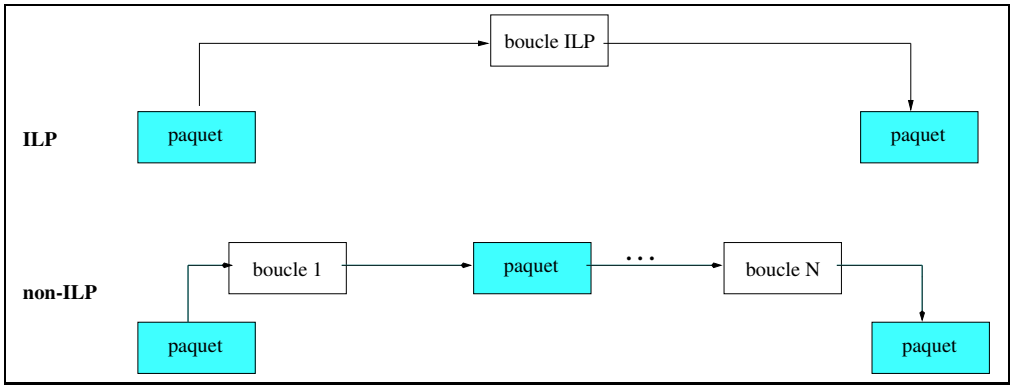

FIG. 2.1 – ILP et non ILP

Le principal avantage de ILP [AP93, BD95] est de permettre la réduction des opérations de lecture/écriture en mémoire qui coûtent cher en performance notamment sur les processeurs RISC. ILP considère qu'il est plus efficace de lire les données une fois et de faire toutes les manipulations possibles quand les données sont dans un cache ou des registres. Pour que ILP puisse s'appliquer, il faut que l'unité de données soit la même dans tous les cas de manipulation (calcul de checksum, chiffrement, présentation...). C'est ce que propose l'architecture ALF en fournissant une unité commune pour les traitements spécifiques au réseau, au transport et à l'application.

[1] Le MTU ou Maximum Transmission Unit correspond à la taille maximum d'une trame sur une liaison de données sans fragmentation.

2.2.3 Automatisation de l'architecture ALF.

Un des principaux inconvénients de ALF et ILP est la complexité du *design* pour réaliser une implantation adaptée. En effet notre idée était d'intégrer les mécanismes de communication dans le code de l'application afin d'obtenir un seul automate ; ce qui permet au système de communication de se synchroniser directement avec l'application et de prendre en compte plus facilement le contrôle de l'application. D'où notre contributions [CKD98] qui reprend et étend mes travaux de thèse. Nous proposons un environnement de développement complet et automatisé, appelé ALFred, en utilisant une approche formelle.

L'environnement de développement ALFred opère en 3 étapes :

- Le compilateur ALF analyse la spécification de l'application décrite dans un langage formel. Le langage qui a été choisi est le langage ESTEREL [BG92] (développé à l'INRIA) qui est un langage synchrone. Les langages synchrones ont été conçus pour implémenter des systèmes réactifs, c'est-à-dire, qui interagissent de manière continue avec leur environnement. Un programme réactif synchrone reçoit des évènements en entrée de son environnement, les traite instantanément (sans interruption) et produit des évènements en sortie. Le style de programmation modulaire, la traduction en automate, la synchronisation via des signaux d'entrée et de sortie, la gestion du temps, le déterminisme sont autant de caractéristiques qui explique le choix de ce langage.
- Un compilateur de talon (stub compiler) combine les fonctions de manipulation des données qui sont utilisées par l'application et le système de communication, en produisant une implantation efficace avec les techniques d'optimisation d'ILP. Ces fonctions sont écrites dans le langage C.
- un compilateur C classique lie l'ensemble des fichiers issus du compilateur ALF et du compilateur de talon pour produire un code exécutable.

Nous avons associé à chaque ADU des paramètres qui permettent de sélectionner les mécanismes de contrôle pour le système de communication ; mécanismes qui seront ensuite intégrés à l'application. Parmi ces paramètres, on peut citer : l'ordre (en séquence ou hors séquence), la fiabilité (non fiable, totalement fiable, partiellement fiable), le trafic (non temps réel, sensible à la gigue et/ou au délai).

Par exemple la description ci-dessous définit un type d'ADU qui doit être transmise de manière fiable, en séquence et qui est sensible au délai.

```
type ADU; %reliable|sequence|delay-sensitive
```

Nous avons prouvé à travers nos travaux que l'intégration automatisée des fonctions de contrôle de transmission dans une application spécifiée de manière formelle est possible. Notre étude a porté sur trois applications : un serveur d'images JPEG où la partie manipulation de données est conséquente, une application "multi-talk" inspirée de IRC (Internet Relay Chat) où la partie contrôle de l'application domine, et une application de senseurs pour collecter des températures où la partie protocole est la plus importante. Les performances au niveau taille du code et efficacité ont montré que l'approche automatisée avec ALFred est pratiquement aussi efficace que les codes ad hoc, écrits à la main, notamment lorsque l'application à développer devient complexe comme le serveur d'images.

2.3 Réseaux Actifs

2.3.1 Contexte

Avec le paradigme des réseaux actifs, se pose le problème de tirer au mieux profit des exigences des applications et des réalités des infrastructures du réseau en intégrant des services applicatifs dans les réseaux et plus particulièrement dans les éléments du réseau : tout usager doit pouvoir déployer un nouveau service de manière dynamique dans le réseau.

Il faut bien séparer les notions de réseaux programmables de celles de réseaux actifs même si elles présentent de nombreux points communs ; comme cela est montré dans le schéma ci-dessous extrait de la classification que nous avons définie dans [FCF99a, FCF99b, CF01].

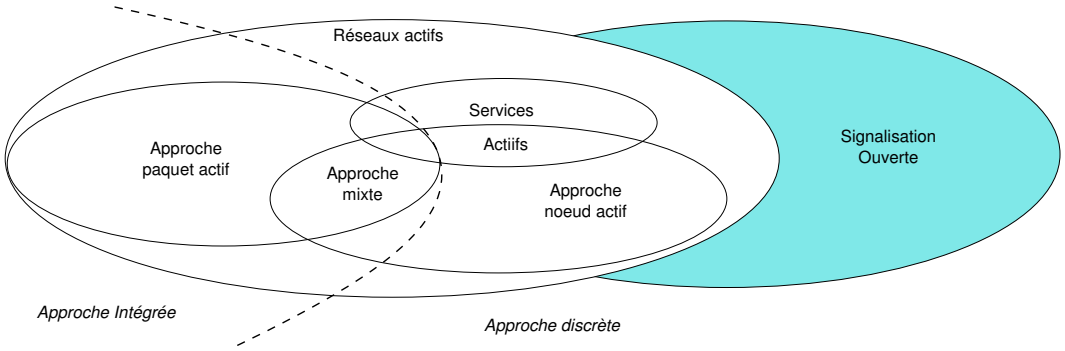

FIG. 2.2 – Classification

Les réseaux programmables correspondent à l'approche signalisation ouverte (OPENSIG) qui est centrée autour de l'ouverture des réseaux actuels via la définition d'interfaces de programmation offrant l'accès au plan de contrôle des équipements. Je ne rentrerai pas dans les détails de cette approche car nous nous sommes orientés, dans nos travaux de recherche, vers la seconde approche connue sous le terme *réseaux actifs* qui est plus ouverte que l'approche *réseaux programmables* dans ses domaines d'applications et dans sa vision de l'architecture de nœuds et de réseaux. Cette approche correspond plus à l'ajout de services applicatifs dans le réseau alors que les réseaux programmables sont surtout issus de travaux relatifs à la signalisation.

Le concept de réseaux actifs est plus généraliste et étudie la mise en place d'architectures permettant aux utilisateurs et aux applications de programmer les éléments du réseau en envoyant des « bouts » de code/programme pour adapter les nœuds à leurs besoins : mise en place d'un nouveau protocole, exécution d'un algorithme de compression,...Une fois le code chargé, les routeurs et les commutateurs agissent sur les données et peuvent même modifier le contenu du paquet avant qu'il ne soit acheminé vers sa prochaine destination. Cette approche doit permettre de s'adapter à des environnements où les besoins des applications changent très rapidement et où les exigences en termes de nouveaux services doivent être rapidement satisfaites. L'approche réseaux actifs peut être affinée en deux sous classes : celle se basant sur une *approche paquet actif* et celle se basant sur une *approche nœud actif*.

- Dans *les approches paquet actif*, le code des services déployés dans les nœuds du réseau est transporté jusqu'aux nœuds dans le même flux que les données qu'il traite. On parle alors d'une approche intégrée.
- Dans les *approches nœud actif*, les services sont déployés dynamiquement sur les nœuds du réseau mais ce déploiement ne se fait pas dans le même flux que celui des données

utilisateurs : les composants de service sont déployés physiquement sur les nœuds du réseau grâce à des techniques de code mobile le plus souvent.

Plusieurs recherches ont été menées [FCF99a] pour concevoir des architectures de réseaux actifs. Dans la suite de nos travaux, nous nous sommes surtout basés sur ANTS. Son principal avantage réside dans son modèle de déploiement de code à la demande. L'architecture ANTS[2] a été définie par le MIT [WGT98] dans la suite des travaux sur Active IP [WT96], prototype de réseaux actifs fondé sur l'utilisation des extensions IP pour le transport du code à appliquer à chaque paquet actif.

Le principe de l'architecture ANTS repose sur la capacité offerte aux applications de déployer dynamiquement dans le réseau les services et les protocoles qu'elles utilisent et ceci sur tous les routeurs actifs que traversent les flux associés. Pour cela, ANTS offre une architecture de nœud permettant un support de protocoles multiples ainsi qu'un mécanisme de déploiement dynamique de ces protocoles au sein du flot de données de l'application. Les principaux composants de l'architecture ANTS sont :

- le protocole qui définit le traitement à effectuer sur un flux de données ;
- la caspule qui représente l'unité de base de la programmation du réseau : elle transporte d'une part les données d'une application entre les nœuds actifs et d'autre part le code d'un protocole à déployer sur les nœuds du réseau ;
- le nœud actif qui représente l'environnement d'exécution et permet la réception, l'exécution et l'envoi de capsules.

Un autre apport d'ANTS est son fonctionnement de nœud qui mélange *l'approche paquet actif* (les paquets transportent le code qui doit leur être appliqué) et *l'approche nœud actif* (le nœud peut accueillir des codes, les exécuter et les maintenir au delà de la durée de vie du paquet). Si conceptuellement, les données et le code transitent dans le même flux, ANTS ne nécessite pas que toute capsule de données inclut le code qui va permettre son traitement dans le nœud grâce à un mécanisme de cache qui permet de conserver du code déjà téléchargé.

2.3.2 Technologie active et Supervision

Motivation

Le domaine de la gestion de réseaux est un thème central dans l'équipe de recherche. La gestion des réseaux et des services est définie comme *regroupant toutes les activités technologiques et organisationnelles mises en œuvre pour offrir des services aux usagers et pour les opérer afin qu'ils respectent les contraintes de qualité et de coût* ou de manière plus précise *gérer un système c'est le surveiller et le contrôler afin qu'il satisfasse les demandes des utilisateurs et les contraintes du propriétaire* [Fes01].

Le concept de réseaux actifs tel qu'il était présenté nous a semblé une approche novatrice pour le fonctionnement et l'exploitation des réseaux, ressources de communications et services. Les applications de gestion de réseaux sont traditionnellement représentées par le modèle centralisé gestionnaire/agents. Le gestionnaire interroge les agents, collecte les alarmes, envoie des ordres de configuration, construit une vue du réseau et avertit l'administrateur lorsqu'un problème est détecté. L'augmentation du nombre de nœuds à gérer a conduit la recherche à s'intéresser à d'autres modèles comme celui de délégation des agents [GY95] en vue de réduire le trafic de gestion et de déplacer les fonctions de management vers les données (c'est-à-dire vers les agents) au lieu de déplacer les données vers les fonctions à appliquer. La technologie des réseaux actifs fournit un mécanisme simple pour faire de la gestion par délégation et offre une architecture

[2] Active Node Transfer System

adaptée à la fonction de gestion de configuration comme le déploiement dynamique de nouvelles versions de protocoles. De nombreux travaux de recherche lient la supervision et les réseaux actifs. Parmi ces travaux nous pouvons citer SMART PACKETS et ceux présentés dans [RS01].

SMART PACKETS [SZJ+99] est une approche intégrée motivée par la croissance de la capacité de traitement disponible sur les nœuds du réseau. Elle utilise le paradigme des réseaux actifs pour décentraliser la gestion. Le principe adopté est un principe sans état afin d'alléger au maximum la charge des routeurs. De ce fait, les programmes de supervision doivent être contenus dans un seul paquet actif dont la taille maximale est fixée à 1Koctets afin que ceux-ci tiennent dans une trame Ethernet. L'architecture SMART PACKETS est composée de quatre parties :

- une spécification des formats de paquets et leur encapsulation dans le standard de description de paquets actifs ANEP (*Active Network Encapsulation Protocol*) [ABG+97] ;
- une spécification de deux langages, un langage de haut niveau (SPROCKET) proche du langage C et un langage d'assembleur SPANNER, associés à une syntaxe de transfert compressée. Ces langages ont été spécifiés pour éviter qu'ils ne contiennent des caractéristiques dangereuses comme l'accès au système de fichiers ou à la gestion mémoire ;
- une machine virtuelle sur chaque élément du réseau pour permettre l'accueil et l'exécution des fonctions de gestion ;
- une architecture de sécurité qui permet d'authentifier un paquet (être certain que le paquet vient bien d'un utilisateur autorisé). L'authentification est réalisée par la présence d'une signature générée à partir des champs non mutables du paquet actif et est liée à l'utilisation d'un système clé publique/clé privée.

Dans [RS01], les auteurs proposent un schéma de gestion de réseau où les mécanismes de contrôle sont complétement distribués laissant peu de tâches à la station centrale qui fournit essentiellement la visualisation et l'interface opérateur. Contrairement à SMART PACKETS, les programmes ne doivent pas forcément correspondre à un paquet ; ceci grâce à la notion de session. Une tâche distribuée est identifiée par un numéro global unique : un numéro de session. Le système est composé de deux entités : un routeur IP et un moteur actif ou *Active Engine* (AE). Le routeur IP réalise les tâches de base du protocole IP. Le moteur actif est un environnement dans lequel du code contenant des paquets actifs peut être exécuté. Un exemple est donné avec la fonction `traceroute` qui permet à un utilisateur d'obtenir la liste de tous les routeurs traversés jusqu'à une destination. L'environnement d'exécution permet de définir un `traceroute` avec l'option `collect-en-route` qui envoie un seul paquet traversant les différents routeurs et collectant les informations. Lorsque le paquet arrive à la destination, celle-ci retourne l'ensemble des données collectées à la source (2 paquets transitent dans le réseau au lieu de $2*(n+1)$, n étant le nombre de routeurs traversés).

Les réseaux actifs peuvent être appréhendés de deux manières face à la gestion de réseaux. Certes, ils peuvent servir de support pour faciliter le déploiement et la réalisation de solutions de gestion et changer la vision client/serveur traditionnelle des approches de gestion. Mais en permettant plus de flexibilité, ils introduisent plus de complexité et, s'ils veulent eux même être déployés, ils nécessitent la mise en place de mécanismes pour leur propre supervision [BP98] :

- dans [Bru02], il est proposé une architecture générique pour permettre à chaque client de créer, exécuter et gérer ses propres services actifs et d'isoler les clients pour éviter une interférence entre eux ;
- le projet ANCORS (*Adaptable Network COntrol and Reporting System*) [PRS98] de son côté est un ensemble d'outils logiciels pour la conception, le déploiement et la supervision des réseaux actifs. Il propose d'enrichir la gestion et le monitorage des réseaux dynamiques en étendant la gestion de réseaux.

Le projet ANAIS

Nous avons pu aborder les aspects supervision et technologie active dans le cadre du projet ANAIS[3], projet soutenu par le programme Télécom du CNRS de 1998 à 2000 et dont j'ai assuré avec Olivier FESTOR la coordination. Ont également collaboré Stéphane D'ALU, Ingénieur Expert, et des élèves ingénieurs de l'École des Mines de Nancy [CCC+98].

Les objectifs d'ANAIS étaient d'expérimenter l'utilisation de paquets actifs pour la supervision d'environnements d'exécution, de protocoles actifs et de réseaux traditionnels et de réaliser un environnement d'exécution ouvert permettant le développement, le déploiement et l'exploitation de protocoles de signalisation pour la supervision des protocoles et services de l'Internet. Il s'agissait de montrer qu'il était possible de superviser un réseau actif avec ses propres méthodes actives. Notre proposition était basée sur ANTS qui offre une approche simple et compréhensible du concept de réseaux actifs. De plus, il existe un environnement ANTS écrit en Java[4] qui permet de développer aisément des prototypes dans l'espace utilisateur.

L'idée de base du projet ANAIS est partie d'un prototype appelé UMANTS [ACF+99] (*User-based Management of the Active Network Transfer System*), prototype de supervision d'une architecture de réseaux actifs, utilisant les mécanismes d'échange de capsules pour déployer à la volée des fonctions de gestion sur les équipements désirés. Pour réaliser ceci, nous avons défini l'architecture suivante :

- à chaque nœud actif du réseau est associée une extension de supervision. Cette extension est un protocole sur lequel une application et/ou une console de gestion va pouvoir effectuer des opération de gestion et/ou télécharger de nouvelles fonctions de gestion via le mécanisme de capsule défini dans ANTS ;
- une fonction de gestion est une classe Java sous-classe de la classe Capsule définie dans ANTS ;
- une capsule de gestion supplémentaire permet à une application de gestion d'invoquer une opération de gestion préalablement chargée sur un nœud.

La plate-forme ANAIS a étendu cet environnement pour en faire une plateforme de supervision d'environnements d'exécution. Elle offre des fonctions de découverte et de rafraîchisssement de la topologie du réseau actif supervisé, une fonction de traçage du déploiement d'un protocole actif ainsi que des fonctions de comptabilité paramétrables et téléchargeables sur les nœuds du réseau (nombre de paquets traités, reçus, envoyés,...). Le superviseur ANAIS, qui est également une application active au-dessus d'ANTS, offre aussi un mécanisme de chargement de fonctions de supervision additionnelles fournies par l'opérateur du réseau.

Le protocole que nous avons retenu pour le transport des paquets actifs est le protocole IPv6 [Ciz02] [DH98]. Au paradigme des nouvelles architectures, nous nous devions d'associer le protocole IP Nouvelle Génération, pour que notre environnement puisse à terme servir à superviser un backbone IPv6. Cependant, la plupart des travaux de recherche relatifs aux réseaux actifs proposent des prototypes au-dessus de UDP, alors qu'il s'agit d'agir directement au niveau réseau. Pour opérer directement au-dessus de IPv6, une extension du protocole ANEP a été proposée [AFC+00, ACC+00] pour permettre l'encapsulation de paquets actifs au format ANEP directement dans IPv6. Le document spécifie comment prendre en compte les jumbogrammes des paquets IPv6 et comment appliquer le checksum tel qu'il est défini avec IPv6. Concernant les jumbogrammes, tous les champs d'ANEP exprimant des longueurs peuvent s'avérer trop petit lors de l'utilisation de l'option jumbogramme qui permet de transporter des paquets $>$ à 2^{16} octets. La sémantique du champ ANEP longueur du paquet ANEP devient une extension du

[3] Active Network Architecture for Internet Service Provider

[4] `http ://www.cs.washington.edu/research/networking/ants`

champ longueur de l'en-tête. Pour le checksum, le draft ANEP précise qu'il doit être calculé sur l'ensemble du paquet ANEP ; ce qui ne permet pas de s'assurer de l'intégrité des adresses source et destination car IPv6 ne calcule plus de checksum sur l'en-tête IP et laisse le soin de le faire au protocole de niveau supérieur. Il a été proposé de prendre en compte un pseudo en-tête dans le calcul du protocole ANEPv6 comme avec UDP.

ANAIS concerne essentiellement la gestion de la technologie active. La supervison des réseaux IP à l'aide de la technologie active a été abordée dans un autre projet de l'équipe : FLAME [DF02] dont nous avons utilisé l'environnement résultant dans le cadre des tests actifs IPv6.

2.3.3 Technologie Active et Tests IPv6

Motivation

Actuellement les protocoles de l'Internet sont décrits de manière informelle dans des documents de travail appelés RFC (*Requests For Comments*) et ils ne respectent pas les étapes d'une conception formelle des protocoles [Dia92]. Toute mauvaise interprétation dans les RFCs due à des ambiguités ou des manques de précision peut conduire à des implantations erronées ou en tout cas non interopérables. La fiabilité des implantations doit alors être mesurée à travers des tests de conformité et d'interopérabilité :

- le test de conformité est une technique qui consiste à réaliser des expérimentations sur une implantation d'un protocole donné ou d'un système réactif afin d'en déduire la correction vis à vis d'une application de référence. Un test de conformité vérifie qu'une implantation peut être considérée comme "conforme" à une spécification donnée ; mais il ne garantit pas l'absence d'erreurs ;
- le test d'interopérabilité permet à des développeurs de tester leur souche et d'effectuer différents scénarii afin de valider l'interopérabilité de leur logiciel. Ils évaluent comment une implantation peut communiquer avec une autre (ou plusieurs) implantation(s) du même vendeur ou d'un autre vendeur sur des systèmes d'exploitation identiques ou différents.

Plusieurs évènements internationaux ont été mis en place pour permettre ces tests de conformité et surtout d'interopérabilité. Dans le monde IPv6, nous pouvons citer les connectathons[5], les rencontres organisées par le projet TAHI[6], le projet TIPI de l'IRISA[7], le laboratoire d'interopérabilité de l'Université du New Hampshire[8], ou par l'ETSI[9]. Afin d'homogénéiser l'ensemble et d'éviter des confusions au niveau des développeurs de logiciels, l'IPv6 Forum[10] a défini un programme logo ou une étiquette de qualité appelée IPv6 Ready[11] qui permet de dire qu'un produit passe un certain niveau de conformité et d'interopérabilité. Pour réaliser ces tests, ces organismes doivent fournir une suite de tests qui consiste en un ensemble hiérarchique de cas de tests, chacun ayant un objectif précis et fournissant un verdict PASS si le résultat du test est conforme à la spécification et aux objectifs, FAIL si le résultat est contradictoire par rapport à la spécification et INCONCLUSIF s'il est impossible de conclure car l'objectif du test n'a pas pu être couvert.

Le projet TAHI a été un des premiers à vouloir développer et offrir des technologies de vérification pour IPv6. Il fournit des outils pour des tests de conformité et pour des tests d'in-

[5] `http ://www.connectathons.org/`
[6] `http ://www.tahi.org/inop`
[7] `http ://www.irisa.org/tipi`
[8] `http ://www.iol.unh.edu`
[9] `http ://www.etsi.org/plugtests/IPv6.htm`
[10] `http ://www.ipv6forum.org`
[11] `http ://www.ipv6ready.org`

teropérabilité [AGK+01]. L'implantation de référence est celle des versions de FreeBSD ; cela ne signifie pas que cette implantation est exempte d'erreurs mais que les deux groupes de travail, le projet TAHI et le projet KAME (développeur de la souche FreeBSD), ont travaillé ensemble pour la définition des tests. Les outils de tests sont basés sur des scripts écrits en langage Perl[12]. Le Laboratoire du New Hampshire produit aussi des services de tests pour des vendeurs afin de vérifier la conformité et l'interopérabilité des produits. Pour IPv6, il offre dans un format informel une liste de suites de tests. De nouvelles suites de tests ont commencé à être développées par l'ETSI et le projet IRISA/TAHI en utilisant des langages plus standardisés comme TTCN-3 (dans les versions 1 et 2, TTCN signifiait *Tree and Tabular Combined Notation*, la version 3 a été renommée en *Testing and Test Control Notation*) [GHR+03].

Étant impliqués dans le groupe de travail G6 (groupe français regroupant des industriels et des académiques testant les protocoles IPv6), nous avions, dans le cadre du LORIA, organisé en 1999 un connectathon relatif à la souche Mobile IPv6. La mobilité IP permet notamment de rester connecter même si les usagers se déplacent : cela signifie la maintenance par exemple des connexions TCP. Nous nous sommes rendus compte à ce moment que tester des protocoles qui impliquent beaucoup de dynamicité était loin d'être trivial et que cela nécessitait des mécanismes permettant d'envoyer des évènements de tests à différents endroits et ensuite de synchroniser les résultats de ces évènements (cas de tests avec plusieurs correspondants et/ou plusieurs mobiles). Cette exigence de dynamique des infrastructures de tests nous a naturellement conduits à l'apport de la technologie active pour la mise en place d'une architecture décentralisée.

L'approche non active ou classique est généralement basée sur des réseaux traditionnels qui comportent un nombre restreint et fixe de services implantés dans les équipements et qui offrent peu de moyens pour déployer de nouveaux services. Que l'architecture de test soit centralisée ou distribuée, elle ne permet pas de modifier dynamiquement le comportement global du réseau.

L'approche active se base sur le fait que tout ou partie des composants d'un réseau actif dans les différents plans (signalisation, supervision et données) sont programmables dynamiquement par des entités tierces (opérateur, fournisseur de services, applications, usagers). Avec la contribution d'une infrastructure active, un test a la forme d'une sonde programmable qui exécute le scénario de test chargé d'une machine de contrôle, obtient les résultats du test, les affiche sur l'écran du même hôte de contrôle et enfin s'achève. Ceci présente des atouts non négligeables pour la mise en place des tests et nous permet d'avoir une architecture hybride : centralisée tout en autorisant la distribution des tests. Parmi les apports et les atouts, nous pouvons encore citer :

- le déploiement dynamique du code. En effet, le code des différents tests peut être téléchargé d'une machine console et exécuté sans aucune installation manuelle ;
- la capacité de télécharger les scénarii de tests vers n'importe quel endroit du réseau ;
- la souplesse et facilité de réalisation des tests du protocole pour l'administrateur en charge des tests. En effet, à partir d'une machine console, il peut superviser tous les scénarii de tests qu'il a mis en place sur des testeurs répartis à travers le réseau de test.

Tests MLD dans l'environnement FLAME

Nous avons mené une activité de recherche sur le test dès 1998 dans le cadre du DEA de David Mercier [Mer98] dont l'objectif était d'étudier la conception d'une architecture de test appliquée aux protocoles IPv6. Le DEA a porté principalement sur une approche dite « informelle » qui se base directement sur les spécifications des RFCs pour tester les protocoles. Dans la continuité de ces travaux, nous avons proposé en 1999 un sujet de DEA [Ndi98] relatif au test de validation

[12] `http ://www.perl.com/perl`

du protocole de contrôle IPv6 (ICMPv6) avec une approche plus « formelle » en partant des spécifications des protocoles décrits en SDL (*Specification and Description Language*).

La coopération que nous avons ensuite mise en place entre Concordia University, Montréal, Canada et notre équipe de recherche en 2001, nous a permis d'approfondir ce domaine d'activité en travaillant sur la génération automatique des suites de tests de conformité et d'interopérabilité des protocoles IPv6 [BDB+02, BBB+03]. Le projet comportait plusieurs étapes :

1. l'utilisation de SDL [CSW84] comme langage de spécification formelle. Le choix de SDL, très largement plébiscité par les industriels, peut s'expliquer de par sa simplicité de mise en œuvre grâce à des environnements de développement et de validation comme Object-Geode[13].
2. la génération des suites de test spécifiées en langage TTCN pour toutes les composantes d'interface (réseau et application) pour l'hôte et le routeur et pour une sélection limitée de configuration d'environnements. La notation TTCN a été conçue à l'origine pour la définition de suite de tests dans le cadre des protocoles définis par l'ISO, organisation internationale de normalisation. Elle permet à un utilisateur de décrire facilement et naturellement tous les scénarii possibles des stimulis en entrée et des réactions diverses entre le testeur et l'implantation sous test.
3. l'expérimentation qui consistait à la définition de l'architecture de test avec la mise en place de tests manuels puis à la comparaison avec les cas de tests obtenus via la génération automatique à partir des spécifications développées par l'équipe de Montréal.

Au niveau de notre activité, nous nous sommes plus particulièrement intéressés à la troisième étape et avons investigué comment utiliser un environnement actif pour effectuer des tests de conformité.

Parmi l'ensemble de la suite de protocoles IPv6, nous avons choisi de nous focaliser sur le protocole MLD (*Multicast Listeners Discovery*) [DFH99], qui offre des caractéristiques de dynamicité appréhendables dans le cas d'un prototype. MLD remplit les mêmes fonctionnalités que le protocole IGMP (*Internet Group Management Protocol*) dans le cadre de IPv4 ; c'est un protocole de gestion de communications de groupe ou multicast de niveau lien local qui permet à un routeur IPv6 de découvrir la présence de nœuds sur l'ensemble de ses liens désirant recevoir des paquets appartenant à un groupe multicast. Ces nœuds sont désignés par le terme *listeners* ou auditeurs à l'adresse de ce groupe. Le protocole MLD permet aussi de tenir à jour la liste des adresses multicast pour lesquelles il existe des participants. Cette information est utilisée par les protocoles de routage multicast pour pouvoir faire acheminer les paquets vers les nœuds intéressés. MLD spécifie des comportements différents pour les hôtes où se trouvent des récepteurs et pour les routeurs. Si un routeur doit lui même être à l'écoute de messages multicast, il implante les deux parties du protocole.

L'environnement d'exécution actif qui a été utilisé pour mettre en place les scénarii de tests MLD est l'environnement FLAME développé dans l'équipe RESEDAS/MADYNES en 2001 par Stéphane D'Alu[DF02]. L'environnement FLAME est un environnement d'exécution développé pour mettre en œuvre des solutions actives pour la supervision. Il est basé sur les composants du nœud actif ASP (*Active Signaling Protocol Execution Environnment*)[14]. De cette architecture initiale, seul le composant d'administration d'application active a été maintenu. Un nœud FLAME comporte deux éléments de base : le gestionnaire du nœud et les instances d'applications actives. Le gestionnaire est le point central et a en charge le téléchargement du code, la

[13] `http ://www.telelogic.com/products/additional/objectgeode/`
[14] `http ://www.isi.edu/active-signal/ARP/DOCUMENTS/ASP_EE.pdf`

réception des paquets, les connexions vers les agents d'administration, le lancement et l'arrêt des applications actives. Les applications quant à elles sont des processus qui font partie de l'environnement d'exécution et dont elles exploitent les services tout en hébergeant leur propre code. Comme présenté dans la figure 2.3, lorsque l'environnement d'exécution actif FLAME reçoit un paquet actif (1), il va identifier l'application active appropriée qui va recevoir ce paquet, et va la charger depuis le serveur de code (2 et 3). Une fois l'application active chargée, FLAME va lui acheminer le paquet actif correspondant (4). Si FLAME reçoit le même paquet actif après un certain temps, il va l'acheminer directement à l'application active déjà chargée en mémoire (5).

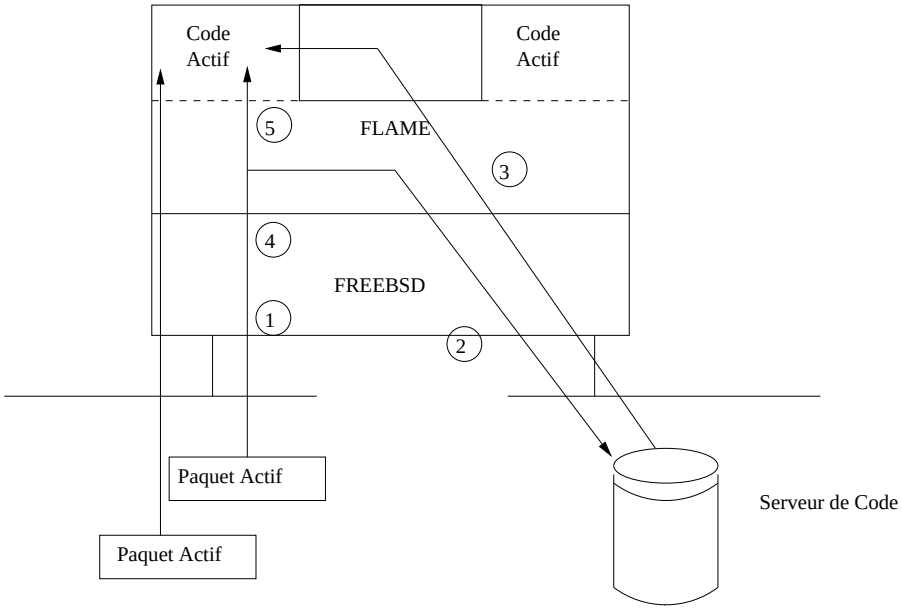

FIG. 2.3 – Mode de fonctionnement de FLAME

Dans le cadre de son stage de fin d'études, Mohamed Salah BOUASSIDA a implanté un certain nombre de tests MLD dans l'environnement actif FLAME [Bou02], en se basant sur les tests MLD décrits dans [Ouz01]. Nous décrivons dans cette section un test qui vérifie la propriété suivante : avec les implantations MLD un seul routeur par lien, à un moment donné, est sélectionné comme celui qui envoie des requêtes multicast sur un lien local ; ce routeur est dit être dans l'état *Querier*.

Comme le montre l'automate extrait du standard (cf. figure 2.4), lors de son initialisation, un routeur commence par envoyer des messages appelés *General Query*, initialise un temporisateur T1 et passe à l'état *Querier*. Quand le temporisateur T1 expire, le routeur renvoie le message. Si le routeur reçoit le message *General Query* d'un autre routeur avec une adresse IP inférieure à la sienne, il passe à l'état *Non Querier*, initialise un temporisateur T2 et arrête toute émission de messages. Si ce routeur ne reçoit plus rien pendant l'intervalle de temps T2, il repasse à l'état *Querier*.

Le scénario équivalent s'exécutant dans l'environnement FLAME nécessite un Routeur Sous Test (RUT) et deux testeurs actifs : un Routeur Testeur (TR) et un Nœud Testeur (NT). La procédure de test consiste à :

- vérifier que le Routeur Sous Test est à l'état *Querier* ;
- amener le Routeur Sous Test dans l'état *Non Querier* en lui envoyant un *General Query* en ayant une adresse IP inférieure à la sienne.

Si le Routeur Testeur reçoit un *General Query* du Routeur Sous Test, alors le test a échoué car il

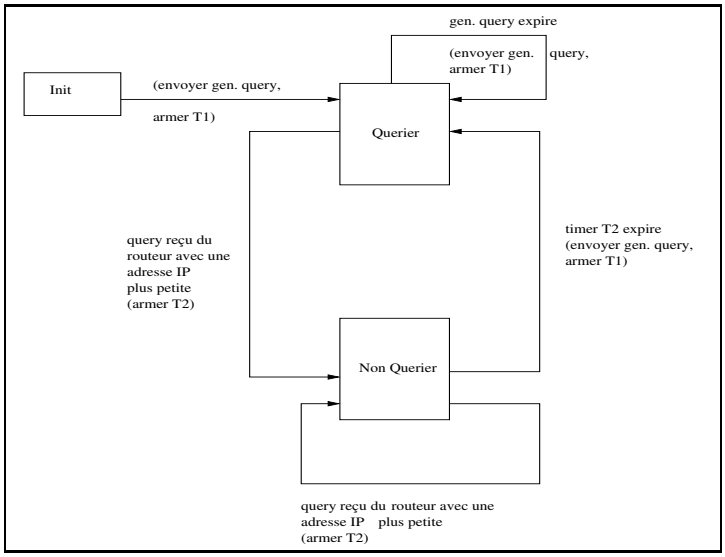

FIG. 2.4 – Automate MLD

y a deux testeurs dans le même état *Querier*. Les étapes pour réaliser la procédure de test seront

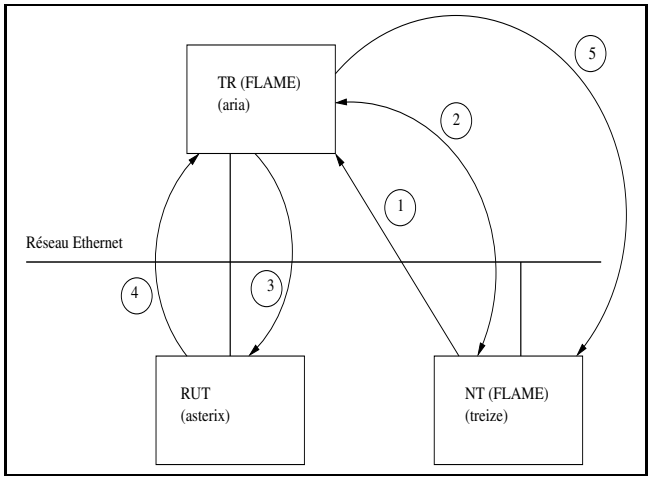

FIG. 2.5 – Approche FLAME Test Seul Routeur Querier Par Lien

les suivantes (cf figure 2.5) :

1. Telnet sur le Routeur Testeur.
2. Le Routeur Testeur charge le code de l'application active correspondante depuis la machine *treize*.
3. Le Routeur Testeur envoie un *General Query* au Routeur Sous Test et déclenche le temporisateur T2, puis envoie un autre *General Query* après une période T1.
4. Si le Routeur Testeur reçoit un *General Query* du Routeur Sous Test après l'expiration du timer T2, c'est l'échec du test, sinon, c'est un succès.
5. Affichage du résultat du test.

Une quinzaine de tests ont ainsi été réalisés et sont décrits dans [Bou02].

2.4 Conclusion

L'approche ALF/ILP dans la conception des réseaux se veut adaptative et évolutive dans un contexte où les besoins des applications changent très rapidement ainsi que leurs exigences en termes de nouveaux services. L'évolution normale a été de pousser l'idée d'adaptation au cœur des éléments du réseau avec le concept des réseaux actifs [ACF+01].

Si l'on observe avec un certain recul l'évolution de ces deux approches :

- L'approche ALF/ILP, outre les influences sur le développement de protocoles comme RTP [SCFJ03] ou sur le Path MTU Discovery [MD90], redevient d'actualité sous un autre nom dans le contexte des environnements mobiles où les implantations basées sur le modèle classique en couches ne fonctionnent pas de manière efficace [XP99]. Cette remise en cause du modèle défini par l'ISO tant au niveau des fonctionnalités que du traitement a ainsi été récemment repensée pour les réseaux sans fil sous le nom de *cross-layer integration/optimisation* où l'idée est d'offrir une gestion plus performante de la mobilité en partageant les connaissances des couches physiques et des couches MAC avec celles des couches plus hautes [SRK03, No3].
- L'approche réseaux actifs où il est possible aux applications de déployer du code au centre même de l'infrastructure du réseau demeure peut-être encore *futuriste* mais il reste indéniable que pour les opérateurs, leur différence, face à une concurrence de plus en plus difficile, va passer par l'offre de nouveaux services à valeur ajoutée et par un déploiement rapide de ces services pour les usagers ; ce qui va conduire d'une manière ou d'une autre à utiliser des infrastructures de réseaux programmables par ces mêmes opérateurs. Par contre, qu'une application soit amenée à configurer ou adapter des éléments de réseaux de proximité pour répondre à ses propres besoins est maintenant devenue une réalité, même s'il s'agit pour l'instant de simple configuration comme par exemple des applications de messagerie instantanée qui utilisent un protocole de découverte de services pour ouvrir, sur un pare-feu, les ports nécessaires à leur fonctionnement[15] .

Travailler sur les réseaux actifs, nous a aussi permis d'aborder la problématique de la dynamicité dans les réseaux et du déploiement de services dans de tels réseaux. Comme service à déployer, nous nous sommes orientés vers celui de la sécurité. L'environnement multicast est un excellent domaine d'application : pour son aspect dynamique tout d'abord, car le modèle défini par Deering permet à tout membre de quitter ou de joindre le groupe à tout moment et ensuite pour son aspect inhérent de grande taille. Ces deux points restent des défis intéressants pour la sécurité dans les communications de groupe, sécurité qui, si elle n'est pas mise en œuvre, reste aussi un frein pour le déploiement du multicast. Le chapitre suivant présente nos principales contributions dans ce domaine.

La liaison entre les réseaux actifs et la sécurité multicast a été concrétisée dans la thèse de Hassen Sallay encadrée par Olivier Festor et André Schaff relative à la supervision des services multicast dans les réseaux IP [Sal04] : une composante sécurité issue des travaux présentés dans le chapitre suivant a été intégrée dans l'architecture active de supervision proposée [HLF+03].

Bibliographie

[ABG+97] D.S. Alexander, B. Braden, C.A. Gunter, A.W. Jackson, A.D. Keromytis, G.J. Minden, and D.J. Wetherall. Active Network Encapsulation Protocol (ANEP). Draft, IETF, July 1997.

[15] `http ://www.microsoft.com/technet/prodtechnol/winxppro/deploy/worki01.mspx`

[AP93] M.B. Abbott and L.L. Peterson. Increasing Network Throughput by Integrating Protocol Layers. *IEEE/ACM Transactions on Networking*, 1(5) :600–610, October 1993.

[BD95] T. Braun and C. Diot. Protocol Implementation using Integrated Layer Processing. In *ACM SIGCOMM on Communications Architectures, Protocols and Applications*, pages 151–161, 1995.

[BG92] G. Berry and G. Gonthier. The ESTEREL synchronous programming language : Design, Semantics, Implementation. *Sciences of Computer Programming*, 12(2) :87–152, 1992.

[BP98] M. Brunner and B. Plattner. Management of Active Networks. In *ICC Workshop on Active Networking and Programmable Networks*, 1998.

[Bru02] M. Brunner. Tutorial on Active Networks and its Management. *Annals of Telecommunications*, 57(5-6) :480–498, May-June 2002.

[Chr94] I. Chrisment. Impact of ALF on Communication Subsystems Design and Performance. In *First International Workshop on High Performance Protocol Architectures*, Sophia Antipolis, December 1994.

[Chr96] I. Chrisment. *Étude et développement d'applications distribuées dans l'architecture ALF*. PhD thesis, Université de Nice-Sophia Antipolis, France, Juin 1996.

[Ciz02] G. Cizault. *IPv6, théorie et pratique. 3ème édition.* O'Reilly, 2002. ISBN 2-84177-139-3.

[CSW84] CCITT-SGX1-WP3-1. SDL, Specification and Description Language. CCITT Recommendations z101-z104. Technical report, CCITT, 1984.

[CT90] D.D. Clark and D.L. Tennenhouse. Architectural Considerations for a New Generation of Protocols. In *ACM SIGCOMM on Communications Architectures, Protocols and Applications*, pages 200–208, Philadelphia, PA, September 1990.

[CWW92] J. Crowcroft, I. Wakeman, and Z. Wang. Layering Considered Harmful. *IEEE Network*, 6(1), January 1992.

[DF02] S. D'Alu and O. Festor. FLAME : une plate-forme active dédiée à la supervision des services de l'Internet. In *Conférence Francophone sur l'Ingéniérie des Protocoles, (CFIP'02)*, pages 99–112, Montreal, Canada, Mai 2002. Hermes.

[DFH99] S. Deering, W. Fenner, and B. Haberman. Multicast Listener Discovery (MLD) for IPv6. RFC 2710, IETF, October 1999.

[DG99] C. Diot and F. Gagnon. Impact of Out-of-Sequence Processing on Data Transmission Performance. *Computer Networks and ISDN Systems*, 31 :475–492, 1999.

[DH98] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF, December 1998.

[Dia92] M. Diaz. Conception formelle des protocoles et des services dans les systèmes distribués. Technical Report 91411, LAAS, Juin 1992.

[Fes01] O. Festor. *Ingénierie de la gestion de réseaux et de services : du modèle OSI à la technologie active.* Habilitation à diriger les recherches, Université Henri Poincaré, Nancy, France, Décembre 2001.

[GHR+03] G. Grabowski, D. Hogrefe, G. Réthy, I. Schieferdecker, A. Wiles, and C. Willcock. An Introduction to the Testing and Test Control Notation (TTCN-3). *Computer Networks*, 42(3) :375–403, June 2003.

[GY95] G. Goldszmidt and Y. Yemini. Distributed Management by Delegation. In *15th International Conference on Distributed Computing Systems*. IEEE Computer Society, 1995.

[Hui94] C. Huitema. *Le routage dans l'Internet*. Eyrolles, 1994. ISBN 2-212-08902-3.

[Hui97] C. Huitema. *IPv6 : the New Internet Protocol. 2nd Edition*. Prentice-Hall, October 1997. ISBN 0-13-850505-5.

[JMG93] A. Jean-Marie and L. Gün. Parallel Queues with Resequencing. *Journal of ACM*, 40(5) :1188–1208, November 1993.

[JML92] A. Jean-Marie and Z. Liu. Stochastic. comparison for queuing models via random sums and intervals. *Journal of Advanced Applied Probabilities*, 24 :960–985, 1992.

[MD90] J. Mogul and S. Deering. Path MTU Discovery. RFC 1191, IETF, November 1990.

[Nel81] B.J. Nelson. *Remote Procedure Calls*. PhD thesis, University of Carnegie Mellon (CMU), USA, May 1981.

[No3] T. Noel. *Contribution à la gestion de la mobilité dans l'Internet*. Habilitation à diriger les recherches, Université Louis Pasteur, Strasbourg, November 2003.

[Ouz01] M. Ouzzif. Tests du protocole MLD. Communication interne non publiée, 2001.

[PRS98] P. Porras, L. Ricciulli, and N. Shacham. ANCORS : Adaptable Network Control and Reporting System. Technical Report SRI-CSL-9801, SRI International's Computer Science Laboratory, 1998.

[RS01] D. Raz and Y. Shavitt. Towards Efficient Distributed Network Management. *Journal of Network and Systems Management*, 9(3) :347–361, September 2001.

[Sal04] H. Sallay. *Architecture de supervision et modèle de comptabilité pour le Multicast IP*. PhD thesis, Université Henri Poincaré, Nancy, Février 2004.

[SCFJ03] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP : A Transport Protocol for Real-Time Applications. RFC 3550, IETF, July 2003.

[SRK03] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. Cross-Layer Design for Wireless Networks. *IEEE Communications Magazine*, (10) :74–80, 2003.

[SZJ+99] B. Schwartz, W. Zhou, A. Jackson, W. Strayer, D. Rockwell, and C. Partridge. Smart Packets for Active Networks. In *Proceedings of InfoComm*, New York, 1999.

[TW96] D.L. Tennenhouse and D.J. Wetherall. Towards an Active Network Architecture. *Computer Communication Review*, 26(2), 1996.

[WGT98] D.J. Wetherall, J. Guttag, and D. Tennenhouse. ANTS : A toolkit for building and dynamically deploying network protocols. In *IEEE OPENARCH'98*, SF, CA, April 1998. `http ://www.sds.lcs.mit.edu/activeware/ants`.

[WT96] D.J. Wetherall and D.L Tennenhouse. The Active IP option. In *Proceedings of the 7th workshop on ACM SIGOPS European Workshop*, pages 33–40. ACM Press, 1996.

[XP99] G. Xylomenos and G.C. Polyzos. Internet Protocol Performance over Networks with Wireless Links. *IEEE Network - Magazine of Global Information Exchange*, 13(4) :55–63, 1999.

Publications

Journaux, livres et chapitre de livres

[ACF+01] L. Andrey, I. Chrisment, O. Festor, and E. Fleury. *Traité IC2, Systèmes multimédia communicants.* Rédaction du chapitre « Infrastructures pour le multimédia : ALF et les réseaux actifs ». Hermès Science, 2001, ISBN 2-7462-0251-4.

[CF01] I. Chrisment and O. Festor. *Logiciels et réseaux de communication, Observatoire Français des Techniques Avancées.* Rédaction d'un chapitre intitulé « Réseaux programmables et Réseaux actifs ». ARAGO 23, pages 199-211, Paris, Mai 2000.

[CKD98] I. Chrisment, D. Kaplan, and C. Diot, An ALF Communication Architecture : Design and Automated Implementation. *IEEE Journal of Selected Area in Communications*, 16(3) :332-344, April 1998.

Conférences

[HLF+03] H. Sallay, A. Lahmadi, O. Festor, and I. Chrisment. Extension de l'architecture active AMAN pour le support des services de sécurité mulicast. *Colloque Francophone sur la Gestion de Réseaux et de Services (GRES'03)*, Fortaleza, Brésil, 2003, 13p.

[BDB+02] A. Benharref, R. Dssouli, Z. Berbich, and I. Chrisment. Formal Specifications, TTCN and Executable Test Cases for main IPv6 Protocols. *IEEE International Workshop on Communication Software Engineering (IWCSE'02)*, Marrakech (Morocco), December 2002, 4p.

[ACC+00] S. D'Alu, G. Chellius, I. Chrisment, O. Festor, and E. Fleury. Intégration du support IPv6 dans l'environnement de supervision de réseaux actifs ANAIS. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'00)*, Toulouse, France, Octobre 2000, 16p.

[ACF+99] L. Andrey, I. Chrisment, O. Festor, and E. Fleury. Supervision et contrôle dans les réseaux actifs ; une nécessité à la mise en œuvre et au déploiement dans les réseaux de télécommunications. *Colloque Francophone sur la Gestion de Réseaux et de Services (GRES 1999)*, Montreal, Canada, 1999, 14p.

Tutoriel

[FCF99b] O. Festor, I. Chrisment, and E. Fleury. Les réseaux programmables. *Tutoriel de l'Ecole d'été RHDM'99*, Bretagne, France. Septembre 1999.

Rapports de recherche, drafts

[BBB+03] A. Benharref, Z. Berbich, M.S. Bouassida, R. Dssouli, and I. Chrisment. Formalisation et test d'IPv6. Rapport de fin de contrat, LORIA n° 03-R-465, 2003, 21 p.

[AFC+00] S. D'Alu, O. Festor, I. Chrisment, and E. Fleury. Active Network Encapsulation Protocol (ANEP) Extension for IPv6. *draft-sdalu-anep-ipv6-00.txt*, November 2000,3 p.

[FCF99a] O. Festor, I. Chrisment, and E. Fleury. Les réseaux programmables. Rapport de recherche INRIA n° 99-R-100, 1999, 48 p.

Travaux encadrés ou co-encadrés

[Bou02] M.S. Bouassida. Utilisation de la technologie active pour tester les protocoles multicast IPv6. Rapport de Stage Ingénieur (ENSI, Tunisie), LORIA n° 02-R-445, 1998, 75p.

[AGK+01] J. Aubé, M.Gardumi, V. Kocher, M. Katya, J. Larnaud, F. Macq, B. Renard, and S. Pesme-Cansart. Environnement de tests pour le protocole IPv6. Rapport Projet 2ème année ESIAL, 2001, 72p.

[Ndi98] I. Ndiaye. Test de validation du protocole de contrôle IPv6. Rapport Stage de DEA, Université Henri Poincaré, Nancy, 1999.

[CCC+98] Y. Carlinet, S. Chaabouni, V. Cremet, and F. Villard. IPv6 et les réseaux actifs. Rapport d'axe de l'Ecole des Mines de Nancy, LORIA n° 98-R-289, 1998, 67p.

[Mer98] D. Mercier. Plan de test Internet nouvelle génération. Rapport Stage de DEA, INPL, Nancy, LORIA n° 98-R-320, 1998, 78p.

3

Sécurité dans les communications de groupe

3.1 Introduction

L'utilisation de l'Internet à des fins commerciales et le nombre croissant d'utilisateurs ne respectant pas la « déontologie Internet » rendent nécessaires des mécanismes de protection et nécessitent des services de sécurité pour transporter les informations de façon sûre. Beaucoup de recherches ont été effectuées dans ce domaine pour protéger les communications réseaux et des standards ont émergé. Différentes approches ont été proposées ; chacune s'appuyant sur une des couches du modèle OSI :

- au niveau applicatif, des solutions permettent de sécuriser la messagerie électronique[16], les protocoles[17] et les documents du web[18], les transactions bancaires avec des cartes de paiement[19] ;
- au niveau transport et session, les protocoles SSL (*Secure Socket Layer*)[20] et TLS (*Transport Layer Security*) [DA99] ajoutent des services de sécurité en agissant séparément de l'application. Il s'agit d'une approche plus générique et indépendante des besoins applicatifs ;
- au niveau réseau, l'architecture IPsec (*IP Security*) fournit une transparence non seulement aux applications mais aussi aux utilisateurs terminaux. Une des principales utilisations de IPsec réside dans la mise en place de réseaux privés virtuels, c'est-à-dire de tunnels sécurisés entre des machines hôtes et/ou entre des passerelles *via* l'Internet.
- aux niveaux physique et liaison de données, des standards comme IEEE 802.11[21], offrent de leur côté des niveaux minimum de protection des données, avec par exemple le protocole WEP (*Wired Equivalent Privacy*) qui assure la confidentialité des données mais présente des vulnérabilités [SIR04].

L'ensemble de ces travaux concerne principalement les transmissions classiques point à point dans des environnements peu dynamiques.

[16] PGP (*Pretty Good Privacy*) : `http://www.pgpi.org`. offre ce service

[17] *via* S-HTTP (Secure HTTP) qui repose sur une extension du protocole HTTP.

[18] `http://www.w3c.org`

[19] à l'aide du protocole SET (Secure Electronic Transaction : `http://www.setco.org`) notamment.

[20] `http://wp.netscape.com/eng/ssl3/draft302.txt`

[21] `http://grouper.ieee.org/groups/802/11`

Parallèlement, avec l'évolution des réseaux et principalement des applications, les communications de groupe ou *transmission multipoint* ou *multicast* (ces termes seront employés de manière indifférenciée dans la suite du mémoire) ont suscité beaucoup d'intérêt au cours de la dernière décennie. Elles correspondent à des modèles adéquats pour des applications coopératives comme l'audio, la vidéo conférence, les communications des forces civiles ou militaires.

De fait, la transmission multipoint apparaît comme un des services de communication le plus efficace pour l'acheminement de données entre de multiples parties. Dans le monde d'Internet, ce service de communication, appelé service de diffusion, repose sur l'utilisation d'une extension du protocole IP : IP multipoint [Dee91]. Cette extension multicast du modèle IP est appelée aussi modèle ASM (*Any Source Multicast*). Le processus de diffusion multipoint sur un réseau local est relativement simple. La machine émettrice spécifie une adresse de destination IP multicast, puis après une conversion de cette adresse en une adresse physique (par exemple de type Ethernet), le système d'exploitation diffuse les paquets de données. Les machines réceptrices doivent notifier à leur couche réseau (en l'occurrence IP) qu'elles veulent recevoir des datagrammes destinés à une adresse multipoint donnée. On appelle cette procédure : « adhésion à un groupe » et est réalisée par un protocole de gestion des adhésions au groupe au niveau local nommé IGMP [Dee89] pour IPv4 ou MLD [VC99] pour IPv6 comme nous avons pu le voir dans le chapitre précédent. Lorsque l'utilisation du multicast est étendue au delà d'un seul réseau physique, et que l'on désire propager des paquets multipoint à travers des routeurs, il est nécessaire d'utiliser un protocole spécifique pour que les routeurs déterminent quelles machines appartiennent à un groupe multipoint donné et qu'un arbre de diffusion multicast puisse être élaboré. L'arbre peut être enraciné à la source et spécifique à une source s'il est mis en place par des protocoles comme DVMRP [Pus03], PIM-DM [ANS05], ou être partagé entre plusieurs sources s'il est mis en place par CBT [Bal97]. Le protocole PIM-SM [EFH+98] construit initialement un arbre partagé et converge vers un arbre enraciné vers la source, s'il y a un fort trafic en provenance de la source.

Avec les communications de groupe, les besoins en sécurité se sont complexifiés et les solutions classiques ne sont plus adaptées. Les applications induisent des problèmes spécifiques qui peuvent influer sur l'architecture et les modèles de sécurité :

Passage à l'échelle. La taille du groupe peut varier d'une dizaine de participants dans les petits groupes de discussion à plusieurs centaines voire plusieurs milliers.

Caractéristiques des membres. Les conditions matérielles (type de machines) ou d'infrastructures réseaux peuvent être hétérogènes entre les membres impliquant des besoins en communication différents.

Dynamisme. La taille du groupe peut évoluer durant une session. Selon le modèle de Deering, tout membre peut joindre ou quitter le groupe à tout moment.

Contrôle du groupe. Il n'y a pas toujours un système central bien informé de l'état du groupe.

Durée de vie. Le groupe peut exister de manière temporaire ou permanente.

Type de membre. Le fait d'être émetteur ou récepteur ou les deux à la fois peut influer sur les mécanismes de sécurité à mettre en place.

Assurer un certain niveau de sécurité représente pourtant un besoin crucial pour un large déploiement des applications de diffusion. Dans le cas des communications de groupe, le potentiel des attaques est beaucoup plus significatif que lors des transmissions point à point :

- 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 malicieux.

La prise en compte de ces problèmes passe par une gestion efficace de la clé de groupe partagée par les différents membres du groupe, permettant que seuls les membres possédant cette clé puissent recevoir et déchiffrer le flux multicast. Dans le cadre de nos travaux de recherche, nous nous sommes principalement intéressés à la distribution de cette clé.

Dans ce chapitre, nous allons tout d'abord, dans la section 3.2, présenter nos motivations en identifiant les services de sécurité à assurer pour un groupe dynamique, ainsi que la problématique de la sécurité dans les communications de groupe et plus particulièrement de la gestion des clés cryptographiques partagées par les membres du groupe. Ensuite nous décrivons nos principales contributions qui sont au nombre de cinq ; chacune faisant l'objet d'une section spécifique. La première contribution a permis de définir une classification des approches de gestion de clés de groupe. La seconde contribution est relative à la sécurité dans le modèle ASM avec la proposition de `Baal` pour assurer la distribution de la clé de groupe. Nous avons ensuite proposé une extension du proxy IGMP afin d'améliorer les performances de l'architecture initiale de `Baal`. En avançant dans le cadre de nos recherches, nous avons orienté nos travaux vers la sécurisation des architectures spécifiques à une source qui correspondent à des applications comme la TV Internet. Le modèle SSM (*Source Specific Multicast*), dérivé du modèle ASM, est en effet apparu pour résoudre les problèmes de déploiement du multicast IP, lié au routage multipoint à large échelle, à l'allocation d'adresses de groupe et au contrôle d'accès aux données du groupe. Cependant le succès de SSM nous semble lié au niveau de service de sécurité qui peut être offert. Ceci a conduit à notre quatrième contribution S-SSM, basée sur `Baal`, pour assurer la sécurité de SSM. Enfin, une collaboration avec l'Université de Sydney nous a permis d'intégrer un système distribué de chiffrement à clés publiques dans l'architecture S-SSM pour améliorer le système de gestion de clés.

3.2 Motivation

Comme la communication point à point, la communication de groupe nécessite des services de sécurité [HM97a] tels que l'authentification, la confidentialité des données et/ou du trafic, l'intégrité mais dont les exigences sont différentes et plus complexes.

Authentification. L'authentification vérifie l'identité d'une entité. Elle constitue une partie essentielle du contrôle d'accès aux groupes sécurisés. Appliquer des mécanismes d'authentification aux processus d'adhésion permet d'assurer que seules les entités autorisées ont le droit de rejoindre les groupes sécurisés. En cas d'utilisation de techniques cryptographiques comme le chiffrement, l'authentification permet de restreindre l'accès aux clés utilisées pour sécuriser les communications de groupe. L'adhésion aux groupes est essentiellement définie par l'accès à ces clés, dont la disponibilité est restreinte aux membres autorisés du groupe. Dans les communications de groupe, le service d'authentification comprend : l'authentification des membres et l'authentification de la source.
D'une façon générale, pour toute application multicast, la source commence par authentifier individuellement tous les membres et contrôler leur accès au groupe, puis lors de la diffusion des données, ce sont les membres qui authentifient la source. L'authentification des membres est réalisée *via* des méthodes utilisant des listes de contrôle d'accès et des certificats capables

d'authentifier mutuellement et individuellement l'émetteur et le récepteur. Ceci nous ramène au cadre de l'authentification point à point, visant à assurer à un nœud l'identité réelle de son interlocuteur.
Concernant l'authentification de la source, on peut distinguer trois niveaux [Esk02] :

- l'authentification de groupe qui fournit l'assurance qu'un paquet a été envoyé par un membre inscrit dans le groupe. Un message chiffré avec la clé de groupe et reçu par un membre du groupe possédant la clé de groupe, garantit que l'émetteur est un membre du groupe ;
- L'authentification des données de la source qui fournit l'assurance qu'un paquet a bien été envoyé par une source du groupe.
- L'authentification individuelle de la source : elle fournit l'assurance de l'identité de la source de données et de sa non répudiation, c'est-à-dire, qu'une tierce partie peut prouver que l'émetteur est bien la source.

Confidentialité. La confidentialité est un service essentiel pour créer des sessions multipoint privées. Le chiffrement peut être appliqué à différents niveaux des couches de protocoles. Par exemple, au niveau de la couche réseau, le protocole ESP (*Encapsulating Security Payload*) [Atk98]) assure la confidentialité aux datagrammes IP par le chiffrement. En effet, avant d'envoyer les données aux membres du groupe, l'émetteur les chiffre avec la clé de groupe, *i.e,* la clé partagée entre les membres du groupe. Ainsi, seuls les membres qui ont cette clé peuvent recevoir et déchiffrer les données de groupe.
En outre, la confidentialité doit être appliquée pendant l'échange des clés dans le groupe. La confidentialité peut être également appliquée aux annonces des sessions multipoint afin de les annoncer publiquement tout en gardant des détails sur les sessions privées.
Assurer la confidentialité, c'est donc permettre un établissement sûr et efficace de la clé de groupe.

Intégrité. Ce service assure que le trafic multipoint n'a pas été altéré en transmission. Elle résulte immédiatement de la confidentialité ou de l'authentification.

Par conséquent, la clé de groupe est le point central de la sécurité dans les communications de groupe car elle permet d'assurer la confidentialité, l'intégrité, l'authentification de groupe et d'une certaine façon le contrôle d'accès aux données du groupe. La gestion de cette clé de groupe représente l'ensemble des techniques et fonctions permettant d'assurer l'établissement et la mise à jour des informations cryptographiques entre les différentes parties :

- tout d'abord, l'inscription à un groupe sécurisé doit fournir un accès à la clé de groupe. Pour les sessions privées, tous les participants devraient être authentifiés individuellement pendant l'étape d'inscription ;
- si l'inscription réussit, la distribution ou la génération de la clé peut avoir lieu.

Ainsi, la participation à une session sécurisée, *i.e.* groupe sécurisé, est définie, non seulement par l'obtention d'une adresse IP multipoint, mais aussi par une clé du groupe destinée à chiffrer/déchiffrer le trafic multipoint du groupe. Cette clé est appelée TEK (*Traffic Encryption Key*). La sécurisation des communications de groupe et la plupart des problèmes de sécurité sont liés à la gestion de cette clé de groupe.
Pour illustrer la problématique liée à la sécurité multipoint, en particulier à la gestion des clés de groupe, nous commençons par présenter schématiquement une session sécurisée de communication d'un groupe dynamique (cf. figure 3.1). Une session est constituée d'intervalles de temps où une entité peut participer à la vie de la session. Un intervalle est défini par un changement sur l'état de groupe, *i.e.* une arrivée ou un départ d'une entité, membre du groupe. La tâche

fondamentale d'un protocole de sécurité multipoint est de permettre aux seules entités autorisées d'accéder au trafic multipoint de la session. Afin d'assurer la confidentialité des données du groupe, il peut être nécessaire de renouveler la clé de chiffrement après chaque arrivée (*backward secrecy*) ou départ (*forward secrecy*). Après chaque arrivée, pour qu'un nouveau membre ne soit pas capable d'accéder à l'ancien trafic du groupe. Après chaque départ, afin d'assurer qu'un membre quittant le groupe ne soit plus capable d'accéder au futur trafic du groupe. Cette exigence en terme de renouvellement de la clé de groupe est fonction de la politique de sécurité à appliquer.

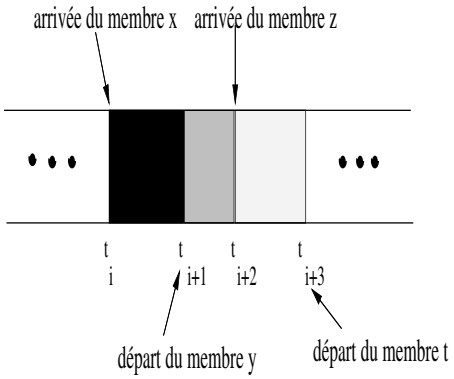

FIG. 3.1 – Évolution de la vie d'un groupe sécurisé

En général, les protocoles multipoint présentent deux problèmes qui limitent le passage à l'échelle ou l'extensibilité. Ces problèmes ont été résumés dans [Mit97] sous les termes 1 **affecte** n et 1 **n'égale pas** n.

- 1 **affecte** n **:** se produit lorsqu'une action chez un membre du groupe affecte tous les autres membres.
- 1 **n'égale pas** n **:** apparaît quand un protocole ne peut pas traiter avec tous les membres d'un groupe ; il doit prendre en compte la capacité de chacun.

Les protocoles de gestion de clés multipoint rencontrent le problème de type 1 **affecte** n lors de l'ajout d'un nouveau membre au groupe et les deux types de problèmes lors de la suppression d'un membre.

Quand un nouveau membre se joint au groupe, l'entité responsable de la gestion de clés doit remplacer la clé du groupe K_{grp} par une autre K'_{grp} afin d'empêcher un nouvel abonné d'accéder à l'ancien trafic du groupe. L'ajout d'un seul membre oblige donc tous les autres membres à remplacer la clé du groupe. L'ajout d'**1** seule entité affecte les **n** (taille du groupe) autres entités.

Quand un membre quitte le groupe, l'entité responsable de la gestion de clés doit également remplacer la clé du groupe K_{grp} afin d'empêcher le membre supprimé d'accéder aux futures communications du groupe. Le gestionnaire de clés crée une nouvelle clé K'_{grp} comme dans le cas précédent, mais cette fois il ne peut pas distribuer la nouvelle clé par un seul message multipoint chiffré avec l'ancienne clé. Les deux types de problèmes d'extensibilité se rencontrent : le premier est 1 **n'égale pas** n car le gestionnaire communique la clé à un membre comme s'il était indépendant du groupe. Le deuxième est 1 **affecte** n car la suppression d'un seul membre oblige les n membres à remplacer la clé K_{grp} par une autre.

3.3 Classification des approches de gestion de la clé de groupe

La classification des approches de gestion de la clé de groupe est importante car elle permet de comprendre, non seulement l'ensemble des travaux qui ont été réalisés dans ce domaine, mais aussi le positionnement de nos contributions ultérieures. Nos premiers travaux présentant cette classification ont été publiés dans [CCS99, CCS00a] et sont à la base de deux tutoriels [CC02b, BC04]. Cette classification a évolué au cours des cinq dernières années, du fait de l'activité de recherche importante dans ce domaine. Les groupes de travail GSEC et MSEC[22] ont été ainsi créés respectivement à l'IRTF et à l'IETF en 2000 ; le groupe GSEC ayant remplacé le SMuG (*S*ecureMulticast Research Group) dont l'origine datait de 1998. D'autres états de l'art ont été réalisés autour de la sécurité des communications de groupe dont ceux, également très complets, de [HD03, SBB+03]

L'établissement de la clé de groupe peut être effectué de deux manières :

- par le transport ou *key transport* : une partie crée un secret et le transporte, le distribue, *via* un tunnel sécurisé vers les autres participants. Ces approches de distribution se décomposent en trois grandes familles : les approches centralisées, les approches hiérarchiques et les approches hybrides ;
- par accord ou *key agreement* : un secret partagé est dérivé par plusieurs parties comme une fonction des contributions de ces parties. C'est ce que nous appelons les approches distribuées.

3.3.1 Approches centralisées

La gestion centralisée est définie par le fait qu'une seule entité contrôle la sécurité du groupe [HM97a, WHA99]. Cette entité s'appelle le contrôleur du groupe, en abrégé CG. Le CG contrôle l'accès au groupe en distribuant la clé du groupe aux participants autorisés (*i.e.* qui ont une permission certifiée par une autorité).

Structure linéaire. Les premières propositions sont des extensions naturelles de la communication point à point. D'un point de vue algorithmique, elles sont extensibles linéairement avec le nombre de participants. C'est le cas, par exemple, de GKMP (*Group Key Management Protocol* [HM97a, HM97b]) où le centre de distribution des clés appelé SKDC (*Single Key Distributor Center*), est un participant du groupe, et non un tiers, qui coopère avec le premier membre pour créer la clé du groupe. Un simple hôte pourrait créer les clés mais l'objectif recherché en impliquant plusieurs hôtes est d'augmenter la probabilité que la clé résultante ait des propriétés cryptographiques adaptées. La distribution aux autres participants est réalisée par des tunnels point à point sécurisés, *via* l'utilisation d'une clé de session. Pour obtenir un grand niveau de sécurité, la clé doit donc être changée après chaque ajout (*Join*) ou retrait (*Leave*). Pour l'ajout d'un nouveau membre, il suffit d'envoyer en point à point la nouvelle clé au nouveau membre et en multicast cette nouvelle clé aux autres membres. Pour le retrait, il faut effectuer n (n étant la taille du groupe) tunnels sécurisés pour distribuer la nouvelle clé ; ce qui revient à créer un nouveau groupe. De ce fait, le nombre de messages échangés et le nombre d'opérations de calcul effectuées par le contrôleur du groupe sont d'ordre n. Malgré la complexité linéaire qui peut être associée à une dynamique importante au niveau des retraits et des ajouts, cette approche reste la solution la plus simple et la plus intuitive pour les petits groupes.

Structure d'arbres. Pour améliorer ces problèmes d'extensibilité, d'autres travaux proposent de stocker les clés dans un arbre de clés.

[22] `http://www.securemulticast.org`

Dans l'approche **LKH** (*Logical Key Hierarchy*), l'arbre est défini de la manière suivante : les membres du groupe se trouvent aux feuilles de l'arbre et les nœuds intermédiaires sont formés de clés logiques. Chaque membre possède les clés se trouvant sur le chemin entre sa feuille et la racine. La racine de l'arbre correspond à la clé de groupe ou TEK (*Traffic Encryption Key*), clé de chiffrement de trafic. Les autres clés représentent des KEKs (*Key Encryption Keys*), clés permettant de chiffrer d'autres clés et non pas les données. Toutes les clés sont gérées par un contrôleur central. Cette stratégie permet à un sous-groupe de participants, qui ont une clé partagée, de communiquer entre eux pour s'échanger des clés d'une manière sécurisée. Cette approche propose un compromis entre le coût temporel, l'espace de stockage et le nombre de message transmis, en utilisant un système hiérarchique de clés auxiliaires, nœuds intermédiaires logiques, pour faciliter la distribution de la clé du groupe. Nous pouvons citer comme exemple de ce type les travaux décrits dans [WHA99, WGL98]. Dans [WGL98], les auteurs se basent sur la notion de graphe de clés et de groupe sécurisé. Un groupe sécurisé est un triplet (U,K,R) : U est l'ensemble des usagers, K l'ensemble des clés et R un sous-ensemble de $U \times K$ tel (u,k)$\in$ R $\Leftrightarrow$ l'usager u possède la clé k. Un graphe de clés est un graphe orienté acyclique avec deux types de nœuds : u-nœud représentant un usager et k-nœud représentant une clé. Les feuilles sont les usagers et le k-nœud à la racine est la clé du groupe. Un graphe de clés ayant un seul k-nœud dont aucun arc ne sort est appelé un arbre. La figure 3.2 représente un arbre de clés dont la racine est K1234.

L'ajout (la suppression) d'un membre exige le changement des clés existant sur le chemin liant le nœud ajouté (supprimé) et la racine, puis la distribution de ces clés aux membres concernés.

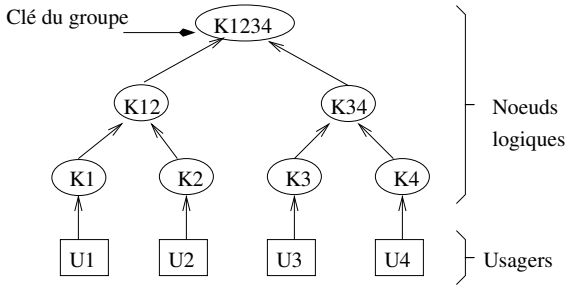

FIG. 3.2 – Graphe de clés : U3 a les clés K3, K34, K1234

Dans l'approche **OFT** (*One-way Function Tree*), le manager du groupe maintient un arbre binaire [MS98, BMS99]. Une feuille représente un membre du groupe. Le manager attribue une clé à chaque membre qui sera la clé du nœud représentant le membre. À chaque nœud de l'arbre OFT sont associées deux clés : la clé du nœud k_x et sa clé aveugle $k'_x = g(k_x)$. La clé aveugle d'un nœud x est calculée à partir de la clé k_x en utilisant une *one-way function* g ; elle est aveugle dans le sens où un adversaire connaissant k'_x ne peut pas apprendre k_x. Le manager communique les clés aux membres par le biais des tunnels de sécurité extérieurs. Les clés des nœuds sont définies par la règle :

$$k_x = f(g(k_{gauche(x)}), g(k_{droit(x)}));$$

tel que *gauche*(x) (*droit*(x)) est le fils *gauche* (*droit*) du nœud x, k_x est la clé de x, g une *one-way function* et f une fonction de mélange (e.g XOR). Une fois qu'un membre a sa propre clé et les clés aveugles des nœuds frères liés directement à son chemin à la racine, il peut calculer toutes les clés des nœuds sur son chemin vers la racine. Lors de l'opération de retrait (ajout) d'un membre, toutes les clés des nœuds (en noir sur la figure 3.3) qui sont sur le chemin entre le nœud du

membre supprimé (ajouté) et la racine doivent être recalculées. De plus, les clés non aveugles associées seront distribuées aux membres concernés (en gris sur la figure 3.3).

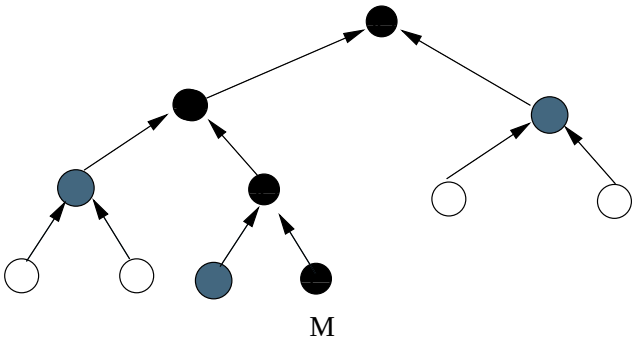

FIG. 3.3 – Un arbre OFT

3.3.2 Approches hiérarchiques

L'approche centralisée nécessite d'avoir un serveur qui peut représenter d'une part un goulot d'étranglement si le nombre de participants devient élevé et d'autre part une cible privilégiée pour les attaques. De plus la vision centralisée ne résout pas le problème 1 **affecte** n car si un membre rejoint ou quitte le groupe, une nouvelle clé doit être générée pour l'ensemble des participants.

Une autre approche consiste à décentraliser la gestion du groupe [Mit97, HCM00, HCD01] et à le diviser en sous-groupes organisés sous forme d'un arbre. Chaque sous-groupe, géré par un contrôleur local, possède sa propre clé. Les sous-groupes sont liés par l'intermédiaire d'agents pour construire un groupe virtuel. Le rôle des agents intermédiaires est de traduire les données multipoint diffusées par un membre dans un sous-groupe à tous les membres du groupe virtuel. Par conséquent, l'ajout ou la suppression d'un membre n'affecte que le sous-groupe auquel il appartient. Dans [Mit97], le GSC (*Group Security Controller*) gère le groupe principal/racine et est responsable de la sécurité entière du groupe. Les GSIs (*Group Security Intermediary*) sont affectés aux autres sous-groupes et forment un pont entre les différents sous-groupes. Pour joindre un groupe sécurisé, un émetteur ou récepteur localise son agent de sécurité (GSC ou GSI) et communique par une demande d'ajout en utilisant un canal sécurisé point à point. Sur la réception de cette requête, si elle est acceptée, l'agent génère un secret partagé avec le membre et communique la nouvelle clé du sous-groupe au nouveau membre chiffrée avec le secret partagé et aux autres membres chiffrée avec l'ancienne clé du groupe.

Quand un membre veut quitter le groupe, il envoie un message à son agent de sécurité qui génère une nouvelle clé et la distribue aux membres du sous-groupe *via* des tunnels sécurisés. Pour la transmission des données, [Mit97] préconise deux méthodes :

- l'émetteur envoie les données chiffrées avec la clé de son sous-groupe à l'ensemble du sous-groupe en multicast. L'agent de sécurité à la réception du message doit le déchiffrer et, le recrypter avec la clé du groupe parent ou enfant avant de l'émettre en multicast et ainsi de suite de sous-groupe en sous-groupe.
- L'émetteur envoie le messsage en point à point à son agent de sécurité qui se charge de le réémettre vers les autres membres du sous-groupe et des autres sous-groupes.

On peut rapprocher ces deux méthodes des arbres bi-directionnels et unidirectionnels rencontrés dans le routage multicast.

3.3.3 Les approches hybrides

Un inconvénient des deux approches centralisées et hiérarchiques est le manque de flexibilité par rapport au dynamisme du groupe. Pour les groupes avec peu de changement dans l'ajout et le retrait des membres une approche centralisée est suffisante, alors que dans le cas d'une dynamicité plus importante l'approche dite hiérarchique ou décentralisée atténue le phénomène 1 **affecte** n. D'où l'idée présentée dans AKMP (*Adaptive Key Management Protocol*) [BBC02], de définir une approche hybride qui consiste à diviser le groupe en sous-groupes selon des critères de dynamicité. Dans chaque routeur AKMP est implantée une fonction d'évaluation qui modifie l'état du routeur à actif ou inactif selon la dynamicité du groupe. Si le nombre de changements (ajout/retrait) par unité de temps excède un certain seuil, alors la dynamicité est dite élevée. Par contre, si le seuil n'est plus atteint au bout d'un certain temps, la dynamicité est à nouveau dite faible. Le protocole commence avec un seul groupe qui partage une clé TEK unique. Ce groupe est initialement géré par un routeur AKMP. Durant la session multicast, si le routeur AKMP détecte une dynamicité locale, il crée un sous-groupe avec une clé locale indépendante. Pour cela, le routeur AKMP génère et distribue la clé locale aux membres dans le sous-groupe construit. Cette clé est appelée clé descendante *Downstream Key* (DK). Ensuite, le routeur décrypte les paquets reçus en utilisant la clé de son routeur parent, appelée clé montante ou *Upstream Key* (UK) et rechiffre les paquets reçus en utilisant la clé DK. Le routeur AKMP commute ainsi d'un état inactif à un état actif. Par conséquent, AKMP réduit le surcoût de chiffrement/re-chiffrement au minimum tout en attenuant le phénomène 1 **affecte** n.

La fonction d'évaluation de AKMP est statique. Une version plus dynamique ainsi qu'un modèle analytique ont été proposés dans SAKMP (*Scalable Adaptive Key Management Protocol*) [CBB04a, CBB04b]. L'objectif de SAKMP est de trouver le partitionnement qui minimise le surcoût engendré par la création de sous-groupes. Pour cela des agents SAKM adjacents s'échangent périodiquement leurs paramètres de dynamisme (fréquence d'arrivée des membres dans le sous-groupe, temps moyen de séjour d'un membre dans le sous-groupe). Sont également pris en compte, le coût du chiffrement/déchiffrement si un fils décide d'être la racine d'un sous-groupe et le nombre moyen de messages en cas de renouvellement de clé si un fils décide de réintégrer un groupe.

Une autre forme d'approche hybride est proposée par le protocole DEP (*Dual Encryption Protocol*) [DMS99] qui utilise une seule clé TEK comme les approches centralisées mais subdivise, comme les approches hiérarchiques, le groupe en sous-groupes. Chaque sous-groupe est géré par un SGM (*Sub Group Manager*) qui ne fait pas partie nécessairement du groupe et donc à qui on ne peut pas faire confiance. DEP gère 3 types de KEKs :

- KEK_{i1} : partagée par un SGM_i et ses membres locaux ;
- KEK_{i2} : partagée par le contrôleur global CG et les membres locaux du sous-groupe i si SGM_i est fils de la racine ;
- KEK_{i3} partagée par le CG et un SGM_i.

La clé TEK est d'abord chiffrée avec la clé KEK_{i2}. Puis elle est chiffrée et décryptée entre chaque sous-groupe (d'où le terme de double chiffrement) : tout d'abord avec la clé KEK_{i3} puis avec la clé KEK_{i1}. Dans cette approche, le flux est crypté avec la même clé TEK et les secrets passé et futur ne sont pas assurés quand un membre quitte ou s'abonne à un groupe ; en effet seule la clé KEK_{i1} est alors renouvelée.

3.3.4 Approches distribuées

Par approches distribuées, nous entendons les solutions où chaque membre contribue à la génération de la clé. Ces approches se basent principalement sur l'algorithme de Diffie-Hellman [DH76] qui permet à deux utilisateurs de calculer une clé commune à partir d'une clé secrète et de l'information échangée publiquement.

Des travaux comme [STW96, BD96] ont regardé comment étendre Diffie-Hellman dans le cas de la communication de groupe. Chaque membre i du groupe contribue à la construction de la clé du groupe par un nombre aléatoire N_i. Le protocole se base sur le calcul distribué du sous-ensemble $\{q^{\Pi(S)}, S \subset \{N_1, ..., N_n\}\}$. À partir de $q^{N_1,...,N_{i-1},N_{i+1},...,N_n}$, le membre M_i calcule facilement $((q^{N_1,...,N_{i-1},N_{i+1},...,N_n})^{N_i}) \bmod p$ qui représente la clé partagée ; q et p sont deux grands nombres premiers entre eux et connus par tous les participants. [STW96] fait une étude comparative de plusieurs extensions de Diffie Hellman et montre que le nombre de messages échangés entre les différents membres et le nombre de tours nécessaire pour construire la clé du groupe sont chacun en $O(n)$.

3.3.5 Synthèse

Nous pouvons considérer les approches centralisées comme étant les plus simples. Mais elles ne résolvent pas, de manière générale, les deux problèmes d'extensibilité : 1 **affecte** n et 1 **n'égale pas** n. Elles conviennent principalement aux petits groupes.

Les besoins en calculs et la taille des données transmises lors de l'initialisation diffèrent selon que l'on utilise des approches basées sur des arbres de clés ou non. Dans [BMS99], les auteurs comparent les trois approches centralisées que sont GKMP, LKH et OFT. Ils montrent que la taille des messages diffusés pour LKH et OFT est double de celle des messages diffusés pour SKDC. Ceci résulte du fait que chaque clé d'un arbre binaire à n feuilles, doit être diffusée aux membres. En conséquence, l'initialisation de SKDC est plus rapide et moins coûteuse que les approches centralisées à structure d'arbres. Par contre, pour les besoins en calcul effectués par le manager et les membres, et la taille des données transmises du manager lors de chaque ajout et retrait d'un membre, on constate que dans le cas de SKDC, la taille des données transmises du manager est nk (n correspond à la taille du groupe et k à la taille de la clé), tandis que celle-ci est $2hk + h$ pour LKH et $hk + h$ pour OFT (h correspond à la hauteur de l'arbre). Par conséquent OFT effectue moins de transmissions, ce qui est normal car les clés du membre à la racine peuvent être calculées de manière récursive. LKH nécessite un espace de stockage plus important car toutes les clés de l'arbre doivent être stockées (soit une taille de $2nk$) contre nk pour OFT (seulement les clés des feuilles de l'arbre) et nk pour SKDC (seulement les clés des membres).

Si on regarde les approches hiérarchiques, elles résolvent en partie le problème d'extensibilité ; puisque seul un sous-groupe est concerné par l'ajout ou le retrait d'un membre. Elles sont plus efficaces pour le dynamisme du groupe ; parce qu'elles distribuent l'effort du calcul de renouvellement de la clé du groupe sur différents participants du groupe. Mais cela se fait au détriment des coûts en chiffrement et déchiffrement lors du changement de sous-groupe ; ce qui peut être pénalisant.

Les appproches hybrides sont parmi les approches les plus intéressantes car elles permettent d'associer les avantages relatifs aux solutions centralisées à ceux des solutions hiérarchiques. Le paramètrage reste cependant difficile à réaliser. De plus, bien que le coût engendré par le chiffrement et le déchiffrement du flux ait été atténué, il peut rester encore contraignant dans certains cas d'applications.

Les approches DIFFIE-HELLMAN de groupe offrent une fonctionnalité distribuée de calcul : il n'y a aucun contrôleur de groupe et tous les membres contribuent à la génération de la clé. Cependant, le nombre de messages émis suit une croissance linéaire et les opérations de calcul sont coûteuses en temps. En effet, les temps de traitement et de communication augmentent avec le nombre de membres. De plus ces solutions ne sont pas tolérantes aux pannes, car si un des membres tombe en panne, la chaîne de calcul est arrêtée.

À la fin des années 1990, après une étude approfondie des approches de la gestion de la sécurité des communications de groupe, nous avions constaté qu'il n'existait pas une solution satisfaisante. Un protocole de sécurité multipoint doit permettre aux seules entités autorisées de participer aux communications du groupe tout en satisfaisant les conditions suivantes dont les seuils dépendent du type d'applications [Chad98] :

- un temps minimal de configuration de groupe ;
- un trafic aussi réduit que possible ;
- un groupe dynamique, *i.e.* retrait et ajout d'un membre possibles à tout moment ;
- une indépendance des protocoles de routage ;
- une confidentialité, intégrité et authentification des messages de configuration ;
- une décentralisation de la gestion du groupe.

Nous avons travaillé dans cette direction dans le cadre de la thèse G. CHADDOUD [Chad02] qui a commencé en Octobre 1998. Ces travaux ont donné naissance à la proposition `Baal` dont nous présentons les principaux éléments dans la section suivante.

3.4 `Baal`

L'objectif de `Baal`[23] est d'offrir une solution au problème d'extensibilité de la gestion des clés dans les groupes dynamiques étendus à large échelle sur Internet, *i.e.* les groupes du modèle ASM. Une telle gestion doit assurer le contrôle d'accès aux données multipoint du groupe en distribuant la clé du groupe seulement aux membres de confiance.

3.4.1 Architecture

La spécification de `Baal` [CCS00b] a été motivée par le constat suivant : dans les conférences spécialisées, il peut y avoir plusieurs participants venant de la même organisation (laboratoire, université, entreprise, ...) et travaillant dans le même axe de recherche que celui de la conférence. Le nombre de participants venant de la même organisation peut varier de un à plusieurs dizaines voire centaines ; le groupe est alors constitué de peu d'organisations mais de beaucoup de membres dans chaque organisation. De ce fait, l'idée d'avoir des entités partiellement déléguées à la gestion de la sécurité de groupe au niveau d'un établissement, d'un réseau local, d'un domaine ou d'un système autonome, s'avère intéressante. Le but principal de cette délégation est de restreindre l'échange de messages de signalisation de contrôle du groupe lors d'un changement sur l'état du groupe aux domaines concernés. L'architecture de `Baal` définit trois entités (cf. figure 3.4) [CCS00b] :

- Un **contrôleur global**, ou CG, qui peut être un organisateur de conférence ou un *chairman*. Il détient une liste `Participant_List`, des futurs participants aux communications du groupe. Il crée la clé de groupe et la distribue aux membres du groupe par l'intermédiaire de contrôleurs locaux. En outre, il effectue le renouvellement périodique et, parfois, occasionnel

[23] Dans la mythologie cananéenne, `Baal` nom générique sémitique signifiant Maître. Il est le dieu de l'orage chez les cananéens et vainqueur sur Môt, dieu de Mort.

de la clé de groupe. Il contrôle toute action concernant la sécurité de groupe effectuée par les contrôleurs locaux.

– Un **contrôleur local**, ou CL, qui est délégué par le contrôleur de groupe. Il reçoit la clé du groupe et la distribue aux membres du groupe dans son réseau lors de la configuration initiale du groupe. Un contrôleur local peut jouer le rôle du contrôleur global : *i.e.* effectuer le renouvellement occasionnel, créer et distribuer une nouvelle clé de groupe, accepter ou refuser un membre, et notifier tout changement dans le groupe aux autres contrôleurs. Un contrôleur local peut être un routeur local, supportant un protocole d'adhésion au groupe comme IGMP, directement lié aux membres du groupe, un agent mère dans le cadre de la mobilité ou un *border router* dans un domaine de routage ou *gateway* dans un système autonome.

– Un **membre de groupe**, ou MG, qui est un membre de la liste `Participant_List` ou tout membre qui rejoint le groupe ultérieurement.

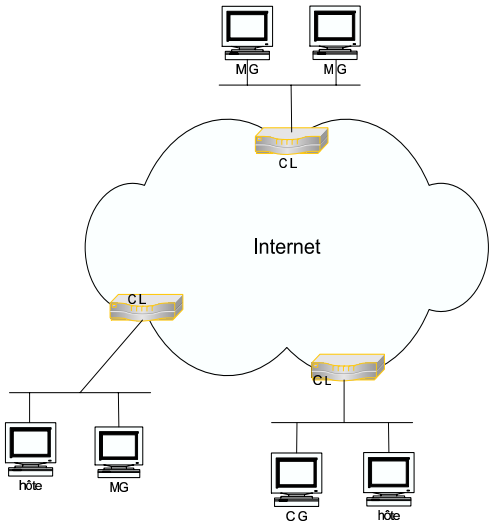

FIG. 3.4 – Architecture de `Baal`

L'ensemble des contrôleurs locaux ou global est supposé être de confiance et avoir les moyens pour générer de manière sûre des clés cryptographiques. L'entité responsable de la coordination entre les contrôleurs locaux est le contrôleur global.

Dans une première version de la spécification de notre architecture, nous sommes partis d'un schéma de base avec comme hypothèse le fait que le réseau d'un établissement est constitué d'un seul réseau local relié à Internet *via* un routeur multipoint supportant la version 3 d'IGMP. Une extension de l'architecture est décrite dans la section 3.5.

3.4.2 Protocole

`Baal` réalise quatre opérations principales : intialisation du groupe, ajout d'une entité, retrait d'un membre et renouvellement périodique.

Initialisation du groupe. L'initialisation de groupe, effectuée par le contrôleur du groupe, se découpe en deux phases : la phase d'invitation qui est réservée à l'invitation des membres de la liste `Participant_List` à participer à la communication de groupe et la phase de distribution de K_{grp} qui a, entre autre, pour objectif de distribuer de manière sûre la clé de groupe aux membres du groupe et d'authentifier les contrôleurs délégués.

Lors de la phase d'invitation, afin de protéger les messages en clair dans cette étape, l'émetteur inclut dans le message un *token* signé. Le *token* signé forme une partie essentielle du processus d'authentification des messages échangés. Il aide un récepteur à vérifier l'origine du message et l'identité de l'émetteur. Pour définir un *token* nous avons repris la forme définie dans [Bal96] où un *token* contient :

- l'identité unique du récepteur, par exemple, l'adresse IP du récepteur ;
- une estampille ;
- un nombre pseudo-aléatoire dont le rôle est de protéger le récepteur contre le rejeu du message.

Cette phase inclut deux types de messages (cf. figure 3.5) : `KC_mg1` et `KC_mg2` en point à point. `KC_mg1` est envoyé par le CG à un élément h de la liste `Participant_List`. Ce message contient la clé publique du CG, son *token* signé et l'adresse IP multipoint du groupe.
Le contrôleur local du destinataire du message authentifie l'émetteur, stocke la clé publique de CG et renvoie le message au destinataire qui, à son tour, authentifie l'émetteur. Si le destinataire accepte de participer, il acquitte le message par un `IGMP-report` contenant son *token* signé et l'adresse IP multipoint du groupe[24]. L'utilisation d'un `IGMP-report` comme un message d'acquittement au message `KC_mg1` permet à l'application multipoint sécurisée du membre invité de se joindre à l'arbre de diffusion multipoint.

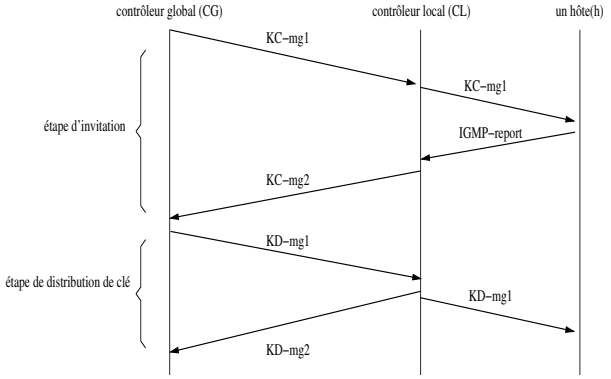

FIG. 3.5 – Initialisation du groupe

En recevant le message `IGMP-report`, le contrôleur local, après authentification, ajoute l'émetteur à la liste `Local_Participant_List` et envoie au CG un message `KC_mg2` signé contenant le *token* signé et la clé publique de l'entité qui a accepté la participation, le *token* et la clé publique du contrôleur local qui se présente au CG comme contrôleur local candidat. Si le CG accepte l'émetteur comme contrôleur local délégué, il stocke sa clé publique et l'ajoute à la liste des contrôleurs locaux. Si un contrôleur local reçoit plusieurs messages, alors le contrôleur local envoie un seul message `KC_mg2` contenant le *token* du premier membre aquittant un message `KC_mg1` et la clé publique et le *token* signé du contrôleur. Dans le cas où une infrastructure de clés publiques, PKI (*Public Key Infrastructure*), est déployée, l'envoi des clés publiques n'est pas obligatoire.

À la fin de cette phase, le CG dispose d'une liste de contrôleurs locaux et de leur clé publique, de l'adresse, de l'identité et de la clé du groupe. Ensuite la phase de distribution de la clé de groupe peut commencer. Le CG construit un paquet constitué de deux clés K_{grp} et KEK (une clé de chiffrement des données et une clé de chiffrement des clés), de l'identité du groupe et de

[24]Nous supposons que IGMPv3 [CDK+02] définit un type de message qui indique la présence d'un *token* signé.

sa propre identité. Ce paquet est envoyé, en point à point à chaque contrôleur dans un message KD_mg1 (cf. figure 3.5) qui doit être chiffré car il contient des clés cryptographiques et l'identité des entités. Le CG chiffre les messages KD_mg1 avec la clé publique du récepteur de message. En recevant le message KD_mg1, le contrôleur local le déchiffre et l'acquitte par un message KD_mg2 chiffré avec la clé publique du CG ; KD_mg2 contient seulement l'identité du contrôleur et l'identité du groupe.
Finalement, le contrôleur local envoie le paquet contenant les clés à tous les membres de la liste `Local_Participant_List` chiffré avec leur clé publique respective.

Jusqu'à présent, un contrôleur local ne possède pas toutes les informations requises pour participer à la gestion de la clé de groupe, notamment la liste et la clé publique des autres contrôleurs locaux. La seule entité qui détient toutes ces informations est le CG qui doit les diffuser à tous les contrôleurs locaux par un message multipoint chiffré avec la clé de groupe.

Ajout d'une nouvelle entité. Le modèle IP multipoint de Deering [Dee91] est un modèle ouvert, c'est-à-dire, n'importe quel hôte peut se joindre à n'importe quel groupe. La jonction à un groupe peut avoir lieu dès qu'un hôte envoie un `IGMP-report`. Dans notre architecture, devenir membre d'un groupe, en dehors de la phase d'invitation, n'est pas systématique. Un hôte doit, selon la politique de sécurité du groupe, remplir certaines conditions. Par exemple, il ne doit pas avoir été expulsé du groupe ; il doit, s'il le faut, avoir payé des frais de participation aux communications de groupe,...etc.
L'entité désirant rejoindre le groupe envoie un message `IGMP-report` à son contrôleur local. Elle inclut dans son message son *token* signé et l'adresse IP multipoint du groupe. A la réception de ce message, le contrôleur local authentifie le *token* signé. Si l'authentification réussit, le contrôleur local donne suite à la demande en vérifiant que l'hôte n'apparaît pas dans la liste `Recovery_List`, *i.e.* l'hôte n'a pas déjà été expulsé du groupe. Deux cas doivent être distingués :

1. Le contrôleur local est un contrôleur délégué possédant les clés K_{grp} et KEK. Si la clé de groupe doit être remplacée suite à l'ajout d'une nouvelle entité, il génère deux nouvelles clés K'_{grp} et KEK'. Puis, il les envoie en multipoint à tous les membres du groupe et aux contrôleurs du groupe, cryptées avec la clé KEK et en point à point à la nouvelle entité, cryptées avec la clé publique de cette dernière.

2. Le contrôleur local n'est pas encore un contrôleur délégué. Il doit alors, avant d'envoyer les nouvelles clés, négocier avec le CG pour obtenir la permission de participer à la gestion de la clé de groupe. Les messages `KC_mg2`, `KD_mg1` et `KD_mg2` sont les mêmes que ceux utilisés lors de l'initialisation de groupe.

Il est à noter que, dans les deux cas, le nombre de messages requis au renouvellement occasionnel après un ajout d'un membre est égal à deux : un multipoint et un autre point à point.

Retrait d'une entité. Nous distinguons deux cas : retrait volontaire et retrait obligatoire ou expulsion. Le premier n'a aucune implication sur la sécurité de groupe ; il est réalisé quand un membre veut quitter le groupe et envoie un message `IGMP-leave` dans le but de stopper le flux du trafic du groupe. Par contre, lors de l'expulsion d'un membre, la clé de groupe doit être remplacée. Le contrôleur local du membre à expulser crée deux clés K'_{grp} et KEK' et construit un message de renouvellement occasionnel avec l'identité du groupe, son identité et l'identité du membre à expulser. Ensuite, il pourrait chiffrer ce message avec l'ancienne KEK et l'envoyer en multipoint aux membres du groupe et aux contrôleurs. Mais comme le membre qui ne fait plus partie du groupe possède aussi la clé KEK, il serait capable d'obtenir la nouvelle clé. Par

conséquent, pour que le message ne parvienne pas à ce membre, nous avons proposé la solution qui suit.
Le contrôleur local se trouvant sur l'arbre de distribution du groupe ne renvoie pas ce message multipoint aux entités de son réseau. Il leur envoie, en point à point, la nouvelle clé chiffrée respectivement avec leur clé publique sauf à l'entité expulsée. Le contrôleur local, qui est un routeur multipoint supportant IGMPv3, bénéficie de la fonctionnalité de *source filtering* proposée par IGMPv3 [CDK+02], qui est « la capacité d'un système de signaler son intérêt pour recevoir ou ne pas recevoir des données destinées à un groupe et émises par certaines adresses sources ». Cette fonctionnalité permet à un contrôleur local d'envoyer un message multipoint, contenant la clé de groupe, à l'ensemble des membres du groupe, à l'exception de ceux existant dans son réseau local.

Renouvellement périodique. Généralement les clés cryptographiques ont une durée de vie limitée et doivent être périodiquement remplacées par des nouvelle clés. La période de renouvellement des clés est déterminée par des crypto-analystes. Elle est fonction de la longueur de clé et de l'algorithme de génération avec lequel la clé a été créée. Le renouvellement de la clé peut être effectué soit par le contrôleur de groupe, soit par un contrôleur local. Le renouvellement se produit en deux étapes. Premièrement, le contrôleur de groupe génère deux nouvelles clés de groupe, K'_{grp} et KEK'. Deuxièmement, ces nouvelles clés sont envoyées en multipoint aux membres du groupe et aux contrôleurs du groupe.

3.4.3 Évaluation de `Baal`

Nous avons repris les critères de performance que nous avions fixés dans [CCS01a, CCS01b] pour analyser les architectures de sécurité pour les communications de groupe et les avons appliqués à l'architecture `Baal` afin de voir comment ces critères étaient respectés.

Sécurité et confidentialité. Concernant le critère sécurité de la clé de groupe, nous avons montré que la distribution se faisait de manière sécurisée et ne permettait pas aux attaquants de récupérer la clé de groupe. Ansi, les deux messages de la phase d'invitation sont munis d'un *token* signé contre le rejeu. Les messages de la phase de distribution de la clé contiennent l'identité, les clés de groupe et sont chiffrés avec la clé publique du récepteur.

La confidentialité du trafic de groupe, quant à elle, est assurée. En effet, seules les entités, c'est-à-dire les membres du groupe, possédant la clé de groupe peuvent déchiffrer les messages envoyés dans le groupe. Pour gérer le fait que des membres puissent recevoir plusieurs clés en provenance de différents contrôleurs, des mécanismes de priorité et de *points d'arrêt* ont été mis en place ; ce qui permet à tous les membres du groupe d'utiliser la même clé en même temps.

Analyse et comparaison. En comparant `Baal` avec les approches centralisées à structure d'arbres [WGL98, MS98] qui assurent la propriété de *forward (backward) secrecy*, nous constatons que ces dernières résolvent le problème d'extensibilité par le biais de la hiérarchie des clés. En effet, elles changent le problème $O(n)$ par un autre en $O(log(n))$, *i.e.* lors du renouvellement de la clé de groupe (à cause d'un ajout ou de la révocation d'un membre), le nombre de messages envoyés est de l'ordre de $log(n)$. Par contre `Baal` résout le même problème en $O(1)$ en délégant des contrôleur locaux, qui sont des routeurs multipoint supportant IGMPv3, au niveau des réseaux locaux où il existe des membres de groupe : un contrôleur nécessite un seul message multipoint pour distribuer la nouvelle clé aux membres n'existant pas dans son réseau et α (α représente le coefficient de participation ou le nombre moyen de membres par contrôleur local) messages point

à point. Ces messages point à point sont intra-domaine, et par conséquent ne représentent pas un problème d'extensibilité.

Nous avons mis également en évidence le fait que `Baal` effectue moins de transmissions pour configurer un groupe sécurisé que les autres approches, surtout SKDC. Lors de l'initialisation de groupe, la taille des données transmises nécessaire pour distribuer K_{grp} est égale à $(n/\alpha).k$. La distribution de K_{grp} au niveau des réseaux locaux n'est pas un souci d'extensibilité ; tant que α est grand, la taille des données transmises reste petite.

Enfin, `Baal` exige un espace de stockage plus petit au niveau d'un contrôleur. Cet espace est égal :

- à $(n/\alpha + 1).k$ avec l'authentification individuelle où $(n/\alpha).k$ sont les clés publiques des autres contrôleurs ;
- à k sans l'authentification individuelle ; soit un espace de stockage en $O(1)$.

Comme avec les approches hiérarchiques [Mit97], dans `Baal`, un contrôleur local décrypte et crypte à nouveau le message pour être transmis à une entité qui, à son tour, le déchiffre pour extraire la clé de groupe. Ces opérations de cryptographie ralentissent cette phase car elles sont moins efficaces que les opérations de cryptographie symétrique, mais elles ne nécessitent pas la négociation des clés partagées qui est la solution alternative dans cette phase. Cependant seule la clé de groupe subit ces opérations, car pour les données il y a une unique TEK.

Implantation. Nous avons mis en œuvre l'architecture et le protocole `Baal` pour tester les fonctionnalités dans un environnement de diffusion multipoint. Le prototype, développé par A. Lahmadi est décrit en détail dans [LGC01]. Il a été testé sur plate-forme de tests composée de cinq machines dont trois routeurs interconnectés par des liaisons à 100Mbit/s. Cela nous a permis d'étudier l'impact du dynamisme d'un groupe sur cette architecture de sécurité. Nous avons ainsi mesuré le temps d'abonnement d'une nouvelle entité dans trois cas :

- abonnement non sécurisé ;
- abonnement sécurisé avec contrôleurs locaux délégués ;
- abonnement sécurisé sans contrôleurs délégués.

Nous avons remarqué que la latence, dans le premier cas, est minime ($<$ à 0.09ms) par rapport aux deux autres cas. Ce résultat est sans surprise car la requête d'abonnement émise sous la forme d'un message `IGMPv3-report`, ne subit aucun traitement supplémentaire. De plus, le routeur local fait partie de l'arbre PIM-SM. Par conséquent, cette latence représente uniquement le temps de propagation et de traitement du message `IGMPv3-report` par le routeur local.

Quant au deuxième cas, nous avons constaté que le temps de latence est beaucoup plus important (de l'ordre de 3ms). Cette différence s'explique par la mise en œuvre du mécanisme de contrôle d'accès. En effet, le délai obtenu représente le temps de transmission et de traitement du message `IGMPv3-report` plus le temps nécessaire pour demander la permission auprès du contrôleur global. Par conséquent, le délai d'abonnement dépend fortement de la capacité de la liaison et de la distance entre le contrôleur local et le contrôleur global, ainsi que de la vitesse de traitement des requêtes au niveau du contrôleur global.

La latence dans le troisième cas est encore plus importante (de l'ordre de 6ms) car si le contrôleur local n'est pas encore sur l'arbre de diffusion sécurisée, il doit se greffer à cet arbre.

Dans la seconde suite de test, nous avons évalué le délai de renouvellement de la clé en fonction de la taille du groupe, *i.e.* du nombre d'abonnés. Le délai de renouvellement est le temps qui s'écoule entre le moment où le contrôleur global décide de renouveler la clé après un nouvel abonnement ou une expulsion et le moment où tous les abonnés reçoivent la nouvelle clé. Le temps de renouvellement de la clé s'est avéré, dans le cadre de nos tests, indépendant de

la taille de groupe ; ce qui peut s'expliquer car le contrôleur de groupe envoie un seul message multipoint sur le canal de contrôle pour effectuer le renouvellement de la clé.

3.5 Extension d'IGMP Proxying

Le but de cette proposition était d'augmenter le nombre de participants derrière un contrôleur afin de réduire le nombre de messages de distribution de clés et de réduire ainsi le nombre de contrôleurs effectuant des opérations d'ajout et d'expulsion dans un groupe sécurisé. L'extension que nous avons proposée est basée sur IGMP Proxying [FHHS04] qui donne une solution pour les routeurs ne supportant pas le routage multipoint, au sein du même domaine administratif. Cette extension a été réalisée par A. Lahmadi et A. Ben Hellel, lors de leur stage de fin d'étude [Lah01, Hel02].

IGMP proxying. C'est un mécanisme de distribution de données multipoint dans un environnement dépourvu de protocoles de routage multipoint. La diffusion du trafic est basée sur des informations de participation aux groupes. Le routage multipoint d'arbre de recouvrement [Dee91] est appliqué à un environnement IGMP. Dans cet environnement, la topologie est limitée à un arbre. La racine de l'arbre est supposée être reliée à une infrastructure multipoint plus large.

L'arbre de recouvrement est construit de la manière suivante : un routeur est sélectionné comme racine de l'arbre. Puis, pour chaque sous-réseau, un routeur relié est nommé routeur désigné. Pour les sous-réseaux reliés à la racine, la racine est leur routeur désigné. Quant aux autres, le routeur qui a le moins de sauts vers la racine est nommé routeur désigné. Une fois les nominations faites, chaque routeur est capable de classifier chacun de ses sous-réseaux reliés à une des catégories suivantes :

- sous-réseau parent : le réseau le plus proche de la racine. Seule la racine n'a pas de sous-réseau parent ;
- sous-réseau fils : tous les sous-réseaux pour lesquels le routeur est un routeur désigné ;
- sous-réseau ignoré : n'importe quel autre réseau.

Le routeur IGMP-proxy possède une unique interface haute, direction parent, et une ou plusieurs interfaces basses, direction fils. Il tourne la partie routeur d'IGMP sur ses interfaces basses et la partie hôte sur son interface haute. Le résultat de ce protocole est un ensemble de *souscription*s[25] pour chaque interface basse.

Un routeur envoie un `IGMP-report` sur l'interface haute lorsqu'il est interrogé, et envoie un `IGMP-leave` lors d'un changement dans sa base de données. Quand un routeur reçoit un message `IGMP-leave` ou un `IGMP-report` sur une de ses interfaces basses, il le renvoie sur son interface haute si le message reçu génère un changement sur l'état de l'ensemble de ses *souscription*s.

Un routeur relaie un paquet reçu sur son interface haute aux interfaces basses figurant dans la liste des *souscription*s. Un routeur relaie les paquets reçus sur une interface basse à son interface haute et aux interfaces basses figurant dans la liste des *souscription*s à l'exception de l'interface sur laquelle le paquet a été reçu.

La figure 3.6 représente un ensemble de quatre LANs. La racine est le routeur A. Les routeurs A, B et C sont des routeurs proxys.

Notre proposition pour l'extension d'IGMP proxying Le but de cette extension était de sélectionner un contrôleur local pour plusieurs LANs.

[25] Une souscription est une entrée d'état IGMPv3, *i.e.* (une adresse multipoint, timer de groupe, mode filtre, liste de source).

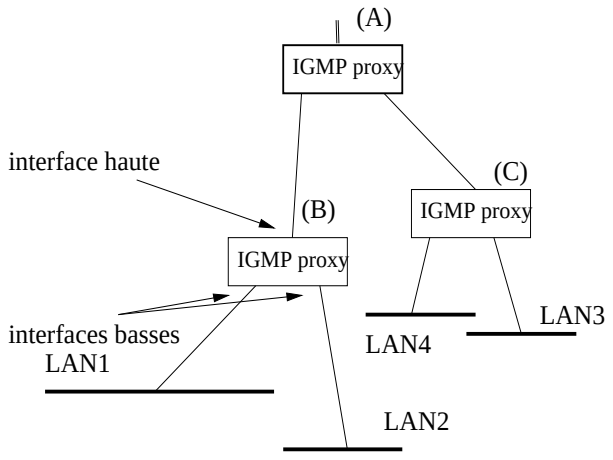

FIG. 3.6 – Arbre de recouvrement enraciné en (A)

Dans un arbre de recouvrement de routage multipoint basé sur IGMP proxying, la racine de l'arbre connaît toutes les relations de participation ou de souscription des membres dans ses sous-réseaux qui forment les feuilles de son arbre. La racine de l'arbre peut alors jouer le rôle du contrôleur local. Un hôte qui reçoit un message d'invitation dans la phase d'initialisation du groupe, confirme sa participation au groupe en envoyant un `IGMPv3-report` accompagné de son *token* signé. En recevant l'`IGMP-report`, le routeur local met à jour sa base de données et relaie l'`IGMP-report` qui est propagé d'un IGMP-proxy à un autre jusqu'à l'arrivée à la racine, qui est le CL du domaine. Ce dernier met à jour sa base de données et la liste de participants locaux après avoir authentifié l'émetteur du message `IGMP-report`. Le contrôleur détient à la fin de cette phase la liste de tous les participants dans son arbre.

La racine peut jouer le rôle du contrôleur local comme cela était spécifié dans le schéma de base. Par contre, pour les IGMP-proxys intermédiaires, une modification du comportement est nécessaire quand ils reçoivent un `IGMP-report` avec un token signé. En effet, lorsque un IGMP-proxy reçoit un `IGMP-report`, il existe deux cas à traiter :

- Un `IGMP-report` sans *token* signé ; il s'agit d'une requête de jonction à un groupe non-sécurisé. Le IGMP-proxy fonctionne comme normalement ;
- Un `IGMP-report` avec un *token* signé ; il s'agit soit d'une réponse à une lettre d'invitation soit d'une requête de jonction à un groupe sécurisé. Dans les deux cas le `IGMP-report` fait partie du protocole `Baal`. En recevant ce type de `IGMP-report`, le IGMP-proxy met à jour sa table IGMP. Puis, il doit impérativement faire suivre ce *report* dans la direction de la racine, pour que l'émetteur du *report* soit authentifié auprès du CL.

Le fait d'utiliser un arbre de recouvrement IGMP proxying, permet d'élargir le nombre de membres gérés par un contrôleur local. Ce qui implique une réduction du nombre de contrôleurs intervenant dans la gestion de la sécurité du groupe et par suite, le nombre de messages utilisés dans la phase d'initialisation du groupe. Supposons que la moyenne du nombre de réseaux locaux gérés par un CL est β et que le nombre de membres par réseau local est α, alors la moyenne du nombre de membres gérés par un CL est $\alpha \times \beta$. Donc, le nombre de messages nécessaires à la distribution de la clé du groupe dans la phase d'initialisation est $n/(\alpha.\beta)$ avec une amélioration d'un facteur de β. Parallèlement, le nombre de messages point à point échangés à l'intérieur d'un domaine entre un contrôleur local et ses membres augmente. Mais, nous avons considéré que cette augmentation ne posait pas de problèmes d'extensibilité.

3.6 S-SSM

Cette approche vise à protéger les communications des groupes spécifiques à une source, *i.e.* les communications du modèle SSM. Elle reprend les principes de `Baal`, mais dans le cas particulier des applications 1-à-N où la source approuve et connaît les récepteurs.

Notre architecture `S-SSM` [CVC+04, CCL02b, CCL02a] offre deux mécanismes de sécurité que nous détaillons un peu plus loin : le contrôle d'accès et la protection du contenu. Le mécanisme de contrôle d'accès est une variante de `Baal`. La protection du contenu est réalisée *via* l'authenfication de l'émetteur et le chiffrement des données. Ce mécanisme nécessite la gestion d'une clé unique appelée la clé de canal, K_{ch}, partagée entre l'émetteur et les abonnés.

3.6.1 Le modèle SSM

Le modèle SSM est apparu comme solution pour résoudre les problèmes de déploiement du modèle ASM tels que l'allocation d'adresses de groupe, le contrôle d'accès et la construction non appropriée des arbres multipoint. Par analogie entre les deux modèles, le groupe ASM est un canal SSM ; un membre correspond à un abonné d'un canal. Les actions d'abonnement (désabonnement) sont équivalentes à l'adhésion (retrait). En plus, dans ASM, nous trouvons PIM-SM, DVMRP et CBT comme protocoles de routage multipoint contre un seul protocole dans SSM qui est PIM-SSM. Et le protocole d'abonnement à un canal est une version modifiée de IGMPv3 [HCH04]. .

Avant d'être adopté au sein de l'IETF, le modèle SSM a été proposé dans EXPRESS [HC99] qui propose des solutions aux problèmes d'accès aux groupes, d'allocation d'adresses et de comptabilité.

EXPRESS : *EXPlicit REquestes Single Source.* À l'opposé du modèle IP multicast qui permet à n'importe quel hôte d'envoyer des données aux groupes, sans avis préalable, EXPRESS est basé sur le fait qu'un seul hôte bien connu, comme la source de données, peut envoyer de données aux abonnés d'un *canal*. EXPRESS ne nécessite pas un mécanisme d'allocation d'adresses multipoint. Il présente un mécanisme de contrôle d'accès au groupe.

En effet, EXPRESS définit un *canal*, au lieu d'un groupe, identifié par un tuple (S, E) où S est l'adresse de l'unique émetteur et E l'adresse EXPRESS de destination. Un abonné, équivalent d'un membre d'un groupe dans le modèle IP multicast, demande la réception de données envoyées au *canal* (S, E) en spécifiant explicitement dans une demande d'abonnement S et E. La source S envoie les données au *canal* en transmettant simplement des datagrammes adressés à E. Le routage EXPRESS assure que les données émises par S à destination de E, seront délivrées aux abonnés du *canal* (S, E) avec un délai et une fiabilité similaires aux datagrammes point à point provenant de S. Les deux *canaux* (S, E) et (S', E) sont indépendants. Un abonné au canal (S, E) ne reçoit pas les paquets envoyés au canal (S', E).

Le processus d'abonnement et de désabonnement dans EXPRESS est équivalent au *join* et *leave* de CBT respectivement. Il existe cependant quelques différences car les messages d'abonnement sont propagés dans la direction de la source, et non pas dans celle du *core* et l'entrée dans la table de retransmission dans un routeur sur l'arbre de diffusion contient le tuple (S, E). La figure 3.7 montre un exemple d'un hôte h qui s'abonne au *canal* enraciné en S. Le message de jonction envoyé vers S ne se propage pas au delà de R qui fait partie du *canal*.

EXPRESS propose l'utilisation de l'authentification des demandes d'abonnement. Un hôte qui désire s'abonner à un *canal* a besoin, en plus des adresses S et E, d'une clé $K_{(S,E)}$. Le premier message d'abonnement se propage le long du chemin vers la source, permettant au

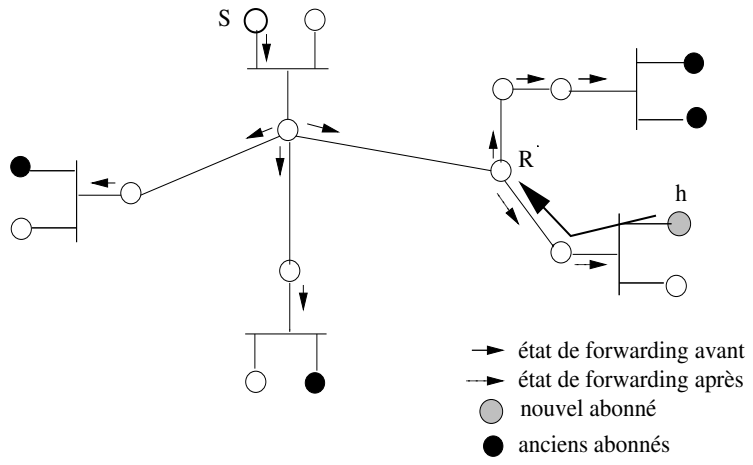

FIG. 3.7 – Un hôte s'abonnant à un *canal* enraciné en S

premier routeur recevant ce message et se trouvant sur l'arbre de diffusion d'authentifier le nouvel abonné car le routeur stocke la clé $K_{(S,E)}$. Un premier routeur qui se trouve sur l'arbre de diffusion et possède la clé d'authentification, doit donc authentifier le nouvel abonné. Cette clé est vue par les routeurs comme un paramètre facultatif permettant de limiter l'accès au canal. En outre, la gestion de la clé de canal n'est pas explicitée.

SSM de l'IETF. L'architecture SSM [HC05] de l'IETF propose le protocole PIM-SSM comme protocole de routage SSM et une version modifiée [HCH04] de IGMPv3 [CDK+02] comme protocole de gestion d'appartenance aux *canaux*.

- **Le routage PIM-SSM.**
  La version 2 de PIM-SM, PIM-SM v2 [FHHK04], supporte la création de deux types d'arbre : un arbre partagé enraciné en un point de rendez-vous (RP) et un arbre basé-source. De plus, elle définit un nouveau type de message *join* spécifique-source. Ce message permet l'abonnement à un arbre basé-source ou à un *canal*. PIM-SM v2 doit intégrer plusieurs changements pour supporter SSM [Bha03] dont :
  * lorsqu'un routeur délégué directement rattaché à l'hôte, reçoit un message *join* (S,G) avec une adresse G dans la plage des adresse SSM[26], il DOIT déclencher un message *join* (S,G), *i.e.* spécifique-source, et JAMAIS un *join* (*,G) ;
  * les routeurs de backbone (*i.e.* qui ne sont pas directement rattachés à des hôtes) ne DOIVENT PAS propager des messages *join* (*,G) pour des adresses G dans la plage d'adresse SSM ;
  * les points de rendez-vous ne DOIVENT PAS accepter de messages *PIM Register* ou *join* (*,G) pour des adresses G dans la plage d'adresse SSM.
- **IGMP avec SSM.**
  Afin de pouvoir implanter un modèle de service SSM, un hôte doit spécifier une adresse source point à point et une adresse de destination SSM. Ceci est fourni par le protocole IGMPv3 [CDK+02] qui définit une nouvelle fonctionnalité *filtrage de source* permettant, entre autre, à un hôte d'exprimer son intérêt pour la réception des données émises par des sources spécifiques. Cette fonctionnalité est nécessaire pour réaliser un modèle de service

[26] L'IANA a réservé la plage d'adresses de classe D $(232, *, *, *)/8$ pour une utilisation expérimentale par le modèle SSM.

SSM. [HCH04] propose des modifications sur IGMPv3 pour l'utilisation avec SSM. Parmi ces modifications nous pouvons citer les restrictions concernant l'utilisation des modes de filtrage, notamment :
* Le mode *EXCLUDE*, permettant de recevoir toutes les sources sauf celles définies dans la liste donnée par *EXCLUDE*, ne doit pas être utilisé avec des adresses SSM ;
* Un hôte ne doit pas envoyer des messages demandant le changement du filtrage de *INCLUDE* à *EXCLUDE* ;
* Si un routeur reçoit l'un de ces types de *reports*, il doit les ignorer.

SSM présente une solution aux problèmes d'adressage et de contrôle d'accès. Le contrôle d'accès est partiel car il permet aux récepteurs de choisir la source, et à la source de définir son propre canal :
- une source ne peut pas émettre sur le canal appartenant à d'autres sources. Contrairement au modèle IP classique, les deux canaux (S, E) et (S', E) qui ont la même adresse destination E, sont indépendants l'un de l'autre. Ceci est assuré par le routage PIM-SSM ;
- un abonné à (S, E) ne reçoit pas automatiquement les données émises à (S', E). Le filtrage par source permet qu'un abonné ne reçoive pas toutes les données émises avec une même adresse destination E.

Le modèle SSM apparaît attractif, mais il montre ses limites lors de son utilisation par les fournisseurs d'accès Internet surtout pour les prestataires de services de diffusion multimédia, comme la TV Internet, qui exigent des frais de souscription[27] à leurs services. Un modèle de sécurité du modèle SSM doit donc permettre aux seules entités légitimes d'accéder au canal de données pour lequel elles ont souscrit.

3.6.2 L'architecture S-SSM

Nous avons proposé S-SSM, une architecture de sécurité plus complète, qui permet d'offrir deux mécanismes de sécurité : le contrôle d'accès et la protection du contenu.
- Le contrôle d'accès. Le but de ce mécanisme est, entre autres, de protéger les ressources réseaux contre les requêtes d'abonnement intempestives et malicieuses.
- La protection de contenu. Elle est effectuée *via* le chiffrement des données. Elle nécessite la gestion d'une clé partagée entre les abonnés et la source. La clé est appelée clé de canal, ou K_{ch}. La source du canal chiffre les données avant diffusion avec K_{ch}. Seuls les abonnés ayant la clé K_{ch} sont capables de les déchiffrer. Ainsi, outre la confidentialité du contenu, l'authentification de la source est assurée, car le routage SSM n'achemine que le trafic en provenance de la source.

Le premier mécanisme peut être vu comme une extension de SSM et devenir une partie intégrante inséparable de l'abonnement à un canal. Le deuxième mécanisme se veut indépendant de SSM et peut être utilisé, selon la politique de sécurité mise en œuvre, par les prestataires qui exigent une confidentialité pour leur contenu.

L'architecture S-SSM propose que les acteurs de l'environnement de diffusion, et notamment les routeurs multicast locaux supportant IGMPv3, soient responsables de la sécurité du canal. Deux raisons justifient ce choix :
- La première est de permettre le contrôle de certains événements qui pourraient compromettre les ressources du réseau, en particulier, les routeurs multicast. À partir du moment

[27] Nous utilisons le mot souscription pour le distinguer du mot abonnement qui signifie l'envoi d'une requête pour la réception des données d'un canal

où une entité envoie une demande d'abonnement, *i.e.* un `IGMPv3-report`, le premier routeur recevant la requête doit être capable de décider du sort de la requête. En effet, si l'entité remplit certaines conditions imposées par la sécurité du canal, le routeur peut donner suite à la requête. Dans le cas contraire, la requête ne sera pas prise en compte. Cela permet de limiter autant que possible le gaspillage des ressources réseau et la protection de l'infrastructure multicast. De plus, les entités malicieuses ne pourront pas ou plus accèder au flux du canal.

– La seconde est de favoriser le passage à l'échelle. Une gestion décentralisée de la sécurité du canal offre plus de flexibilité. En d'autres termes, au lieu d'avoir une seule entité contrôlant toutes les opérations de sécurité dans le canal, certaines entités distribuées peuvent être déléguées pour réaliser des tâches de sécurité. Il y a plusieurs avantages à cette délégation. La propagation des messages de contrôle est limitée aux seuls domaines d'où les demandes de souscription sont issues. De plus, le temps de latence est aussi minimisé et la concentration de trafic au niveau d'une seule entité responsable de la gestion de sécurité est évitée.

Les tâches de sécurité sont déléguées par le *gestionnaire de canal* à des acteurs locaux de l'environnement de diffusion. Les acteurs coopèrent à travers un groupe de communication, ou un canal (CG, E) où CG est l'adresse du gestionnaire de canal et E est une adresse SSM. Ce canal est le *canal de contrôle* ou de signalisation. Ce canal est responsable du contrôle d'accès au canal de diffusion et de la gestion de la clé de canal K_{ch}.

L'architecture `S-SSM` (cf. figure 3.8) ainsi définie repose sur l'architecture de `Baal`. On y retrouve les trois entités que sont le contrôleur global (CG), le contrôleur local (CL) et le membre/abonné (MG). Une entité supplémentaire a été ajoutée. Il s'agit du Serveur ou `DS` (*Directory Server*).

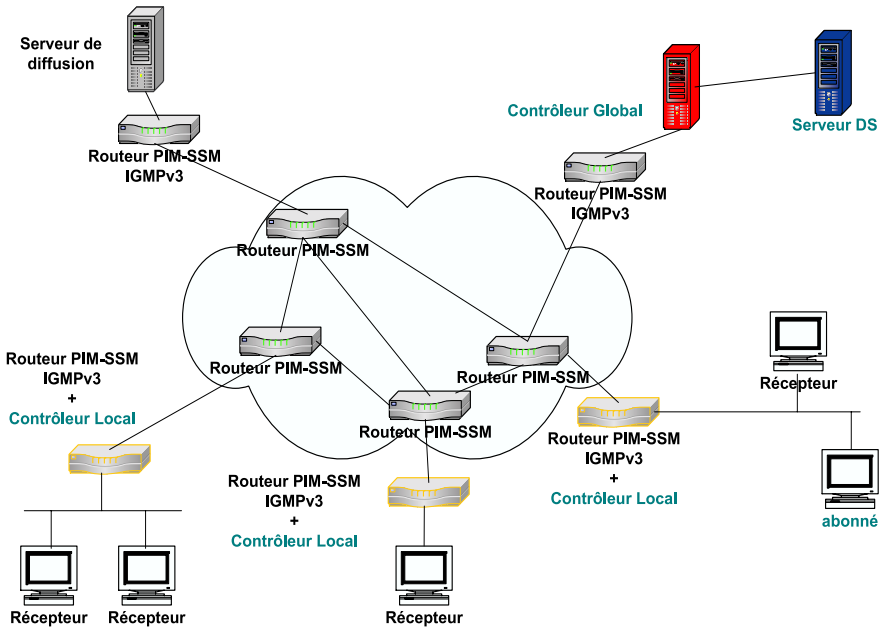

FIG. 3.8 – Environnement **S-SSM**

Le fournisseur de services stocke toutes les informations relatives à ses clients ou abonnés et nécessaires à la gestion du canal dans un serveur, appelé *Directory Server* ou DS. Le serveur DS contient un registre ou une entrée par abonné. Une entrée peut être formée de plusieurs champs comme, *Validité* pour prouver que l'entrée est valide, *date_ début* and *date_ fin* pour le début et la fin de la souscription et un nombre aléatoire, attribué à l'abonné. Un exemple d'un tel serveur pourrait être un serveur LDAP [YHK95].

En accord avec le serveur DS, le CG et les contrôleurs locaux créent une liste globale appelée Recovery_List (*Recovery List*), composée des entités qui ont compromis ou essayer de compromettre la sécurité du canal. Les membres de cette liste n'ont pas ou plus le droit d'accéder au canal. En outre, chaque contrôleur local maintient une liste des abonnés locaux qui est mise à jour après chaque ajout ou éviction d'un abonné. Un contrôleur local distribue la clé K_{ch} seulement à ses membres locaux.

Dans le reste de cette section, nous montrons comment nous avons réalisé les mécanismes de contrôle d'accès et de gestion de la clé de canal.

Contrôle d'accès. L'un des objectifs du modèle SSM visait au départ à combler l'une des plus importantes lacunes du modèle ASM, qui est le contrôle d'accès. Mais, si le modèle SSM a permis à un récepteur de choisir ou de connaître la source, il n'informe pas la source de l'identité des récepteurs. Pour cette raison, nous avons proposé un mécanisme pour renforcer l'autorité de la source, *i.e.* du propriétaire ou du prestataire du service de diffusion, sur le contrôle d'accès des récepteurs à son canal.

Pour le contrôle d'accès aux canaux, nous avons utilisé le concept du *token* signé déjà utilisé dans Baal mais, avec une légère différence dans sa définition. En effet, son utilisation dans Baal permettait d'authentifier l'émetteur au niveau du routeur local avant d'accéder à l'infrastructure multipoint. Par contre, dans S-SSM, en plus de l'authentification et le non-rejeu, le *token* permet de prouver l'appartenance ou la souscription au service de diffusion du canal. Le *token* signé forme une partie essentielle du processus d'authentification des messages échangés lors de l'abonnement. Un *token* est composé de :

- Ns : nombre spécifique attribué par le prestataire de services lors de la souscription au service de diffusion du canal. Il identifie le service auquel le nouvel abonné a souscrit ;
- le tuple (S, ch), avec S l'adresse de la source et ch adresse du canal ;
- *nonce* : un nombre aléatoire dont le rôle est de protéger la requête d'abonnement contre le rejeu ;
- une estampille ;
- l'adresse IP du récepteur.

La phase d'abonnement est réalisée en deux étapes :

- l'authentification qui est réalisée par un contrôleur local. L'authentification forme la première partie du mécanisme du contrôle d'accès. Elle vise à authentifier un nouvel abonné lorsqu'il envoie une requête d'abonnement à un canal. Si l'authentification échoue ou si le nouveau membre est dans la liste Recovery_List, le message sera tout simplement ignoré. Sinon une étape de vérification de validité de la demande d'abonnement est déclenchée.
- La vérification de validité. Le but de cette étape est de vérifier avec le CG si le nouvel abonné a une entrée valide au niveau du serveur DS pour le canal demandé.

Protection du contenu *via* la gestion de la clé de canal. Le mécanisme de contrôle d'accès ne protège pas complètement les récepteurs d'un canal. En effet, n'importe quel hôte malveillant qui se trouve sur l'arbre de diffusion d'une source, peut se faire passer pour cette source. Il suffit

pour cela qu'il mette l'adresse point à point de la vraie source à la place de son adresse dans le paquet IP. Pour cette raison l'utilisation de la clé de canal assure, en plus de la confidentialité des données et de l'authentification de la source, le contrôle d'accès au canal de diffusion.

Le changement de la clé peut prendre place après :

- la réception d'une nouvelle demande d'abonnement pour ne pas permettre au nouvel abonné d'accéder à l'ancien trafic du canal ;
- l'expiration de la permission d'accès d'un abonné ou l'expulsion d'un abonné compromettant pour leur éviter d'accéder au trafic transmis ultérieurement sur le canal ;
- la périodicité du renouvellement de la clé de canal pour éviter que la découverte d'une clé de groupe compromette toute la diffusion du trafic sur le canal.

Dans tous les cas, le renouvellement de la clé est effectué par le CG[28]. Il crée une nouvelle clé, K'_{ch} et forme un message de renouvellement, *msg_ rekey*. Ce message est composé de l'adresse du canal (S, ch), de l'identité du CG et de la nouvelle clé K'_{ch}, ainsi que de deux champs *type* et *INFO*. Le champ *type* est utilisé pour signaler le type de renouvellement (renouvellement périodique, abonnement, expiration ou expulsion). Dans le cas d'expiration de validité ou d'expulsion d'un abonné, le champ *INFO* contient des informations sur l'abonné expulsé. Le message *msg_ rekey* sera chiffré avec la clé du canal de contrôle, K_{ctl}, puis diffusé sur le canal de contrôle aux différents CLs.

La distribution de la clé du canal au nouvel abonné suit directement l'étape de contrôle d'accès. La nouvelle clé est chiffrée avec la clé du canal de contrôle, puis diffusée sur ce canal aux CLs qui assurent la sécurité de la clé dans leur domaine.

De même, la clé de canal est remplacée par une autre après l'expulsion d'un membre. La clé est diffusée à tous les abonnés sauf à celui qui vient d'être expulsé.

Donc comme nous le remarquons, la sécurité de la clé de canal dépend fortement de la confiance des contrôleurs locaux et du canal de contrôle. Cela implique que la sécurité du canal de contrôle doit être bien assurée. D'où l'importance du renouvellement périodique des clés de ce canal.

L'approche que nous avons proposée pour la gestion de la clé de canal n'est pas la seule qui pourrait être utilisée ; plusieurs approches pour la gestion de la clé de groupe comme GKMP [HM97a, HM97b], LKH [WGL98, WHA99], ou OFT [MS98, BMS99] peuvent intégrer notre solution pour la gestion de la clé de canal.

3.7 S-SSM et DCCP

Ce travail est le résultat d'une collaboration avec le Professeur Varadharajan lors de son séjour au LORIA en Août 2002. Nous avons ainsi pu apporter une amélioration dans la gestion de la clé de canal.

Système DCCP. Le schéma distribué de chiffrement à clés publiques proposé par Yi et Varadharajan [YV01] que nous nommons, par simplication, dans la suite du document schéma DCCP consiste en un gestionnaire et plusieurs utilisateurs ou membres formant un groupe. Le gestionnaire du groupe a comme tâches :

- la construction de la seule clé publique du système et des clés privées correspondantes ;
- la distribution des clés privées aux membres ;

[28] **Remarque** : Mais toutefois le contrôleur global peut déléguer complètement ou partiellement cette tâche aux CLs

– le renouvellement de la clé de canal après la révocation d'un membre de groupe, l'ajout, le retrait d'un membre ou lors d'un renouvellement de manière périodique.

La clé publique est la clé de chiffrement et les clés privées sont les clés de déchiffrement. Dans le système DCCP, la clé publique est gardée par le gestionnaire. Chaque membre du groupe possède une clé privée. Un message chiffré par le gestionnaire avec la clé publique de groupe, peut être déchiffré avec n'importe quelle clé privée détenue par un membre. Dans un système DCCP, la qualification publique/privée des clés ne veut pas dire que la clé publique est connue publiquement. Au contraire, cette clé est une clé secrète et connue seulement par le gestionnaire.

La tâche principale de la construction du système DCCP est de trouver un algorithme qui construit plusieurs clés privées de déchiffrement pour une clé publique de chiffrement. Pour plus d'informations sur ce système les lecteurs peuvent se référer à [YV01].

Le schéma DCCP définit toutes les opérations nécessaires à la gestion d'une clé de groupe. En outre, il permet d'expulser, ou d'ajouter des membres d'une manière transparente vis-à-vis d'autres membres. En effet, seul le gestionnaire du groupe effectue le calcul supplémentaire. Ce système permet donc une gestion efficace de groupes sécurisés et dynamiques.

– Chiffrement/déchiffrement. Le but est d'utiliser la clé publique pour chiffrer le message M avant de le diffuser aux membres du groupe. À la réception du message chiffré, le membre j utilise sa clé privée pour déchiffrer et récupérer M.
– Expulsion de membre. Le but est de permettre au gestionnaire d'expulser un membre γ du groupe sans affecter les autre membres. Après l'expulsion, le gestionnaire recalcule la valeur s qui, avec la clé publique, fait partie de l'opération de chiffrement, de manière à ce que la valeur s_γ correspondant au membre expulsé soit enlevée du calcul, c'est-à-dire, $s = \prod_{i=1, i\neq\gamma}^{n} s_i$. Sous ce schéma, le protocole de chiffement donné ci-dessus peut encore être utilisé sans aucune modification. Le membre expulsé γ ne peut pas déchiffrer le message M tandis que les autres membres peuvent encore continuer à le déchiffrer comme avant.
– Ajout de membre. Le gestionnaire est capable d'ajouter un nouveau membre au groupe sans affecter les autre membres. Pour un nouveau membre r, le gestionnaire a besoin de générer un nouveau « s_r » et calcule un nouveau « s » tel que $s = s_1 s_2 \cdots s_r \cdots s_n$.

DCCP dans S-SSM. Nous avons proposé d'intégrer le système DCCP dans notre architecture S-SSM pour gérer la clé de canal K_{ch} [CVC+04].

Pour réaliser cette intégration, nous avons proposé que le CG et les CLs de S-SSM forment un groupe dont le gestionnaire est le CG et les membres sont les CLs. Dans la phase d'initialisation et avant la diffusion de données du canal, le CG crée la clé de chiffrement et les clés de déchiffrement des futurs CLs. Au départ le groupe est vide. Puis, à chaque fois qu'un nouveau CL se joint au canal de diffusion pour la première fois, le CG lui envoie sa clé de déchiffrement. Pour envoyer cette clé d'une manière sécurisée, le CG peut soit utiliser :

– la clé publique du CL pour chiffrer la clé. Dans ce cas, nous supposons que si le CG ne dispose pas de cette clé, alors le CL doit joindre son certificat au message d'adhésion. Cela impose, cependant, l'existence d'une infrastructure de gestion de clés ;
– établir un tunnel sécurisé *via* l'établissement d'un secret partagé en utilisant par exemple un mécanisme d'échange de clé comme dans IKE [HC98].

Un contrôleur local utilisera sa clé privée pour déchiffrer les messages en provenance du CG. Ces messages peuvent être des messages de notification (par exemple, éviction d'abonnés) ou de renouvellement de la clé de canal K_{ch}. S'il s'agit d'un message de renouvellement de la clé, le contrôleur local doit procéder à la distribution de cette clé aux abonnés de son domaine, c'est-à-dire, aux membres de la liste. Cette distribution peut se faire de deux manières :

- si le nombre d'abonnés dans un domaine n'est pas grand, le CL peut utiliser les clés publiques d'abonnés pour chiffrer la clé de canal. Nous avons supposé que les CLs connaissent les clés publiques des abonnés dans leur domaine.
- si le nombre d'abonnés est important, le CL peut, à l'instar du CG, former un groupe DCCP dont il est le gestionnaire et dont les membres de groupe sont les abonnés dans son domaine. Dans ce cas, le système DCCP est utilisé sur deux niveaux. Le premier niveau est le niveau racine qui contient un groupe formé et géré par le CG et qui a comme membres les CLs. Ce niveau est utilisé pour sécuriser tous les messages, y compris celui du renouvellement de la clé de canal, envoyés par le CG à destination des CLs. Quant au deuxième niveau, il est utilisé par le CL pour distribuer la clé de canal aux abonnés locaux.

Apport à `S-SSM`. Le système DCCP est efficace lors de la révocation de membres dans le groupe. En effet, pour renouveler la clé de canal, le CG génère une nouvelle clé de canal K'_{ch}, la chiffre avec la clé publique avant de l'envoyer en un seul message multipoint à tous les CLs. Les CLs font le relais dans leur domaine, c'est-à-dire qu'ils distribuent la clé aux abonnés dans leur domaine. Comme nous l'avons déjà dit, la distribution dans le domaine se fait soit en formant un groupe DCCP, soit en utilisant les clés publiques des abonnés. Comme nous pouvons le constater, le problème de la révocation, qui est un problème d'extensibilité, se résume à un renouvellement périodique. La majeure partie des calculs est effectuée à la configuration du système. Par conséquent, l'intégration du système DCCP dans notre `S-SSM` présente une solution au problème d'extensiblité 1 **n'égale pas** n.

3.8 Conclusion

Le modèle de communications de groupe dynamique sur Internet ou IP multicast est apparu comme un moyen pour optimiser les ressources utilisées par les applications multi-parties (conférence audio, serveur vidéo...). Mais aujourd'hui, nous pouvons constater que le déploiement commercial de ce modèle n'est pas encore une réalité et ne pourra devenir effectif que s'il offre des services de sécurité conséquents. La communication de groupe présente en effet plus d'opportunités que les communications point à point pour les attaques et les écoutes passives. De plus, aucune solution n'était vraiment satisfaisante au moment où nous démarrions nos travaux.

Dans ce cadre, nous avons regardé comment assurer la sécurité des communications pour les groupes dynamiques et avons proposé une approche de gestion de la clé de groupe qui offre, entre autre, la confidentialité des données et l'authentification du groupe. Notre première contribution a été la proposition de l'architecture et du protocole `Baal` pour assurer la sécurité du modèle ASM. Une extension du proxy IGMP a permis d'optimiser notre architecture.

De nombreuses critiques du modèle ASM face aux problèmes de routage et d'allocation d'adresse avaient conduit la communauté scientifique à définir un modèle simplifié du multicast, de 1 vers N, le modèle SSM, spécifique à une source. Nous avons proposé une variante de `Baal` appelée `S-SSM` pour la sécurisation d'un environnement de diffusion multimédia SSM et y avons intégré un schéma distribué de chiffrement à clés publiques. Les travaux que nous avons réalisés pour la sécurisation des communications SSM ont fait l'objet d'une rédaction d'un *Internet Draft* [CCS02] publié au sein du groupe GSEC[29].

Nous avons montré que notre approche `Baal` présente une solution au problème d'extensiblité de l'ordre 0(1) en utilisant, après modification, la nouvelle fonctionnalité du filtrage de la source pour permettre au contrôleur local de ne pas renvoyer des messages multipoint de distribution

[29] Le groupe de recherche sur la sécurité des communications de groupes de l'IRTF

de la clé de groupe dont il est la source. Cependant la problématique de 1 **affecte** n n'est pas encore résolue : `Baal` utilise une seule clé TEK et une seule KEK qui sont renouvelées auprès des membres après chaque ajout ou départ. Avec l'utilisation du système distribué de chiffrement à clés publiques dans S-SSM, on évite de changer la clé KEK (clé du canal de contrôle) à chaque changement au niveau des abonnés. Il faut remarquer que le problème d'extensiblité 1 **affecte** n n'est pas résolu même si un seul message multipoint est utilisé pour renouveler la clé de canal. De plus, le nombre maximal de contrôleurs locaux et/ou de membres doit être fixé à l'avance.

Dans la suite de nos travaux, nous avons cherché à adapter l'architecture et le protocole `Baal` à des environnements plus dynamiques comme les réseaux spontanés et plus particulièrement les réseaux ad hoc. Se posait alors le problème de la prise en compte de différents facteurs que nous n'avions pas envisagés initialement et qui nous ont amené à envisager une évolution de nos contributions. C'est cette évolution que nous présentons dans le chapitre suivant.

Bibliographie

[ANS05] A. Adams, J. Nicholas, and W. Siadak. Protocol Independent Multicast - Dense Mode (PIM-DM) : Protocol Specification (revised). RFC 3973, IETF, January 2005.

[Atk98] R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2406, IETF, November 1998. Standards Track.

[Bal96] T. Ballardie. Scalable Multicast Key Distribution. RFC 1949, IETF, May 1996. Experimental.

[Bal97] A. Ballardie. Core Based Trees (CBT version 2) Multicast Routing. RFC 2189, IETF, September 1997. Experimental.

[BBC02] H. Bettahar, A Bouabdallah, and Y. Challal. An Adaptive Key Management Protocol for Secure Multicast. In *Proceedings of the 11th IEEE-International Conference on Computer Communications and Networks ICCCN'02*, pages 125–133, Miami, Florida, USA, October 2002.

[BD96] M. Burmester and Y. G. Desmedt. Efficient and Secure Conference-Key Distribution. In Springer-Verlag LNCS 1189, editor, *Proceedings of the International Workshop on Security Protocols*, pages 119–129, 1996.

[Bha03] S. Bhattacharyya. An Overview of Source-Specific Multicast (SSM). RFC 3569, IETF, July 2003.

[BMS99] D. Balenson, D. McGrew, and A Sherman. Key Management for Large Dynamic Groups : One-way Function Trees and Amortized Initialization. draft-balenson-groupkeymgmt-oft-00.txt, February 1999. http ://www.securemulticast.org/draft-balenson-groupkeymgmt-oft-00.txt.

[CBB04a] Y. Challal, H Bettahar, and Bouabdallah. An Adaptive Key Management Protocol for Secure Multicast. In *WWIC'04, Lecture Notes in Computer Sience (2957)*, pages 260–271, Frankfurt(Oder)-Germany, February 2004.

[CBB04b] Y. Challal, H Bettahar, and Bouabdallah. SAKM : A Scalable and Adaptive Key Management Approach for Multicast. *ACM SIGCOMM Computer Communications Review*, 34(2), April 2004.

[CDK+02] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagaragan. Internet Group Management Protocol, version 3. RFC 3376, IETF, October 2002.

[DA99] T. Dierks and C. Allen. The TLS Protocol, Version 1.0. RFC 2246, IETF, January 1999.

[Dee89] S. Deering. Host Extensions for IP Multicasting. RFC 1112, IETF, August 1989.

[Dee91] S. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, December 1991. Ph.D. Thesis.

[DH76] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6) :644–654, 1976.

[DMS99] L. Dondeti, S. Mukherjee, and A. Samal. Scalable Secure One-to-Many Group Communication using Dual Encryption. In *Proceedings of the Fourth IEEE Symposium on Computer Communication*, Red Sea, Egypt, July 1999.

[EFH+98] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast Sparse Mode (PIM-SM) : Protocol specification. RFC 2362, IETF, June 1998. Experimental.

[Esk02] A. Eskicioglu. Multimedia Security in Group Communications : Recent Progress in Wired and Wireless Networks. In *Proceedings of the IASTED International Conference on Communications and Computer Networks*, pages 125–133, Cambridge,MA, USA, November 2002.

[FHHK04] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. Protocol Independent Multicast Sparse Mode (PIM-SM) : Protocol Specification (revised). Internet-Draft draft-ietf-pim-sm-v2-new-11.txt, IETF, October 2004.

[FHHS04] B. Fenner, H. He, B. Haberman, and H. Sandick. IGMP/MLD-based multicast Forwarding (IGMP/MLD proxying). Internet-Draft draft-ietf-magma-igmp-proxy-06.txt, IETF, April 2004.

[HC98] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, IETF, November 1998. Standards Track.

[HC99] H. Holbrook and D. Cheriton. IP Multicast Channels : EXPRESS Support for Large-scale Single-source Applications. In *ACM SIGCOMM on Communication Architecture and Protocols (SIGCOMM'99)*, (Cambridge, Massachusetts, USA), August 1999.

[HC05] H. Holbrook and B. Cain. Source-Specific Multicast for IP. Internet-Draft draft-ietf-ssm-arch-07.txt, IETF, October 2005.

[HCD01] T. Hardjono, B. Cain, and N Doraswamy. A Framework for Group Key Management for Multicast Security. Draft draft-ietf-ipsec-gkmframework-03.txt, IETF, August 2001. http ://archive.dante.net/mbone/refs/draft-ietf-ipsec-gkmframework-03.txt.

[HCH04] H Holbrook, B Cain, and B. Haberman. Using IGMPv3 and MLDv2 For Source-Specific Multicast. Internet-Draft draft-holbrook-idmr-igmpv3-ssm-08.txt, IETF, October 2004.

[HCM00] T. Hardjono, B. Cain, and I Monga. Intra-Domain Group Key Management Protocol. draft Internet draft : draft-ietf-ipsec-intragkm-02.txt, February 2000. http ://www.ietf.org/proceedings/00jul/I-D/ipsec-intragkm-02.txt.

[HD03] T. Hardjono and L.R. Dondeti. *Multicast and Group Security*. Artech House Computer Security Series, 2003. ISBN 1-58053-342-6.

[HM97a] H. Harney and C. Mucknhirn. Group Key Management Protocol (GKMP) Architecture. RFC 2094, IETF, July 1997. Experimental.

[HM97b] H. Harney and C. Mucknhirn. Group Key Management Protocol (GKMP) Specification. RFC 2093, IETF, July 1997. Experimental.

[Mit97] S. Mittra. Iolus : A Framework for Scalable Secure Multicasting. In *ACM SIGCOMM on Communication Architecture and Protocols (SIGCOMM'97)*, September 1997.

[MS98] D. A. McGrew and A. T. Sherman. Key Establishment in Large Dynamic Groups using One-way Function Trees. Technical Report TIS Report N° 0755, TIS Labs at Network Associates, Inc., May 1998.

[Pus03] T. Pusateri. Distance Vector Multicast Routing Protocol. Internet-Draft draft-ietf-idmr-dvmrp-v3-11.txt, IETF, October 2003.

[SBB+03] H. Seba, A. Bouabdallah, N. Badache, H. Bettahar, and Tandjaoui D. Gestion de clés et sécurité multipoint : étude et perspectives. *Annals of Telecommunications*, 58(7-8) :1090–1129, 2003.

[SIR04] A. Stubblefield, J. Ioannidis, and A.D. Rubin. A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP). *ACM Transactions on Information and System Security (TISSEC)*, 7(2) :319–332, 2004.

[STW96] M. Steiner, G. Tsudik, and M. Wainder. Diffie-Hellman Key Distribution Extended to Group Communication. In *3rd ACM conference on Computer and Communication Security,*. 3rd ACM conference on Computer and Communication Security, New Delhi, India, March 1996.

[VC99] R. Vida and L. Costa. Multicast Listener Discovery, Version 2 (MLDv2) for IPv6. RFC 2004, IETF, June 1999.

[WGL98] C. Wong, M. Gouda, and S. Lam. Secure Group Communications using Key Graphs. In *ACM SIGCOMM on Communication Architecture and Protocols (SIGCOMM'98)*, September 1998.

[WHA99] D. Wallner, E. Harder, and R Agee. Key Management for Multicast : Issues and Architecture. RFC 2627, IETF, June 1999.

[YHK95] W. Yeong, T. Howes, and S. Kille. Lightweight Directory Access Protocol. RFC 1777, IETF, March 1995.

[YV01] M. Yi and V. Varadharajan. Robust and Secure Broadcasting. In *International Conference on Cryptology in India (INDOCRYPT 2001)*, Chennai, India, décembre 2001.

Publications

Journaux, livres et chapitre de livres

[CVC+04] G, Chaddoud, V. Varadharajan, I. Chrisment and A. Schaff. Gestion efficace de la sécurité des communications de groupe pour le Service SSM. *Journal TSI n° spécial Réseaux et Protocoles*, 23(9) :1107-1135, 2004.

Conférences

[CCL02b] G. Chaddoud, I. Chrisment, A. Lahmadi. A Secure SSM Architecture. *IEEE International Conference on Networks (ICON'02)*, Singapore, 27-30 August 2002, 13p.

[CCL02a] G. Chaddoud, I. Chrisment, A. Lahmadi. S-SSM : Sécurisation d'une architecture SSM. *Colloque Francophone sur l'Ingénierie des Protocoles (CFIP'02)*, Montréal, Canada, Mai 2002, pages 295-408

[CCS01b] G. Chaddoud, I. Chrisment, A. Schaff. Dynamic Group Communication Security. *The 6th IEEE Symposium on Computers and Communications (ISCC'01)*, Hammamet, Tunisia, July 2001, 8p.

[CCS01a] G Chaddoud, I. Chrisment, A. Schaff. Dynamic Group Key Management Protocol. *Mathematical Methods, Models and Architectures for Computer Networks Security International Workshop (MMM-ACNS'01)*, St Petersburg, Russia, May 2001, 12 p.

[CCS00b] G. Chaddoud, I. Chrisment and A. Schaff. Baal : sécurisation des communications de groupes dynamiques. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'00)*, Toulouse, France, Octobre 2000, 15p.

[CCS00a] G. Chaddoud, I. Chrisment, A. Schaff. Secure Multicasting Survey. *IFIP World Computer Congres, (SEC'00)*, Beijing, Chine, August 2000, 4p.

Tutoriels

[BC04] A. Bouabdallah and I. Chrisment. IP Multicast Security. *Internet Nouvelle Génération (ING'04)*, Obernai, France, Juin 2004.

[CC02b] G. Chaddoud and I. Chrisment. La sécurité et les communications de groupe. *Sécurité et Architecture Réseaux 2002 (SAR'02)*, Marrakech, Maroc, Juillet 2002.

Rapports de recherche, drafts

[CCS02] G. Chaddoud, I. Chrisment, A. Schaff. A Secure SSM Architecture. *draft-irtf-gsec-sssm-00.txt*, IRTF, February 2002.

[CCS99] G. Chaddoud, I. Chrisment and A. Schaff. Vers une communication de groupe sécurisée : état de l'art et perspectives. Rapport de recherche LORIA n° 99-R-244, Novembre 1999, 19 p.

[LGC01] A. Lahmadi, G. Chaddoud, I. Chrisment. Implémentation d'un prototype du protocole Baal. Rapport de recherche LORIA n^o A01-R-393, 2001, 31 p.

Travaux encadrés ou co-encadrés

[Chad02] G. Chaddoud. *Sécurisation de communication de groupes dynamiques.* Thèse d'Université, Université Henri Poincaré, Nancy, LORIA n° 02-T-130, Août 2002, 121p.

[Hel02] A. Ben Hellel. Une architecture SSM sécurisée utilisant IGMP Proxying. Rapport Stage Ingénieur (EPT, Tunisie), LORIA n° 02-R-442, 2002, 67p.

[Lah01] A. Lahmadi. Mise en œuvre d'un outil de gestion de clés dans les groupes dynamiques. Rapport Stage Ingénieur (ENSI, Tunisie), 2001, 62p.

[Chad98] G. Chaddoud. La sécurité dans IPv6 pour des applications multipoint. Rapport Stage DEA, Université Henri Poincaré, Nancy, LORIA n° 98-R-346, Juillet 1998.

4

Sécurité dans les réseaux spontanés

4.1 Introduction

Les réseaux spontanés permettent à un ensemble de machines hôtes d'être connectées facilement et rapidement entre elles avec un minimum d'infrastructure préalable, voire sans infrastructure. Les réseaux ad hoc, les réseaux de senseurs, les réseaux pair à pair sont une illustration de ce concept de spontanéité où chaque nœud contribue activement à la vie du réseau soit en collaborant pour acheminer les données à destination, soit en acceptant d'être à la fois client et fournisseur de contenu. L'apparition des réseaux spontanés est aujourd'hui possible d'une part avec le déploiement exponentiel des réseaux sans fil grâce à l'émergence des nouvelles technologies et standards (e.g. les réseaux 802.11[30] Hiperlan [ETS96], ...) et d'autre part avec la mise à disposition croissante des terminaux évolués autonomes (téléphone, PDA, ordinateurs, ...).

Les réseaux spontanés constituent un pilier de l'informatique ambiante ou ubiquitaire qui peut être définie comme une informatique mobile et embarquée offrant des ressources partout et à tout moment et permettant aux objets de se coordonner et d'interagir. La définition initiale de l'informatique ambiante a été donnée par Marc Weiser en 1991 [Wei91] qui prévoyait une omniprésence de l'informatique dans notre quotidien à un point où elle deviendrait transparente, invisible. Cette idée considérée à l'époque comme relativement futuriste est devenue quasi-réalité : on peut citer des exemples avec le vêtement intelligent ou l'informatique vestimentaire[31] qui intègre des écrans et des téléphones portables, ou les voitures devenues maintenant communicantes qui peuvent founir de nouveaux services : alerte des secours, localisation, informations touristiques, téléphonie intégrée, accès Internet,...

L'ensemble de ces réseaux est dynamique dans l'espace et dans le temps et offre une grande flexibilité. Cependant cette flexibilité associée avec la vulnérabilité des connexions sans fil nécessite un besoin croissant dans la sécurisation des données et des utilisateurs. Utiliser des réseaux sans fil rend les réseaux spontanés vulnérables aux attaques des écoutes passives, aux usurpations d'identité, au rejeu et à la distortion de messages. Les écoutes passives permettent à un adversaire d'avoir accès à des informations secrètes ; les attaques actives permettent, quant à elles, de détruire des messages, d'injecter des messages erronés, de modifer des messages et d'ursurper l'identité d'un nœud et par conséquent de violer la disponibilité, l'intégrité, l'authentification et la répudiation qui sont les éléments de base de la sécurité des réseaux.

[30] http ://http ://grouper.ieee.org/groups/802/11/

[31] `http://www.media.mit.edu/wearables/`

Dans le cadre de nos travaux nous avons regardé les problèmes de sécurité rencontrés dans l'informatique ambiante en général, puis nous nous sommes davantage intéressés aux réseaux ad hoc qui sont un élément important de cette informatique ubiquitaire. Les réseaux ad hoc ne bénéficient d'aucune infrastructure fixe même s'ils peuvent être combinés avec des passerelles vers le monde filaire afin d'offrir une couverture plus large et un accès à l'Internet ; il est impossible de mettre un composant centralisé dans la solution proposée. Les mécanismes de sécurité doivent donc aussi être dynamiques et efficaces pour s'adapter eux-mêmes à la nature des réseaux ad hoc. Les nœuds dans un réseau ad hoc sont hétérogènes et peuvent avoir une protection physique très faible ; par conséquent les attaques internes doivent aussi être prises en compte.

Dans la suite du chapitre, nous détaillons nos contributions relatives à la sécurité des réseaux spontanés. La section 4.2 présente une architecture de sécurité dans les environnements d'informatique ambiante. La section 4.3 décrit nos travaux actuels sur la sécurité dans les réseaux ad hoc et plus particulièrement pour les communications de groupe. La section 4.4 conclut.

4.2 Sécurité et informatique ambiante

4.2.1 Contexte

Nous avons travaillé sur l'informatique ambiante lors des travaux de thèse de L. Ciarletta [Ciar02] au NIST[32], Institut Américain de Standardisation, sur les aspects d'architecture, de configuration automatisée et de sécurité des réseaux dans le contexte des Espaces Intelligents (appelés Smart Spaces). La thèse de L. Ciarletta s'est en effet intégrée dans les projets AirJava [Mil99] et Aroma[33] du NIST.

Le but de `AirJava` était de combiner le protocole de découverte de services Java Jini[34] avec la technologie sans fil pour permettre à des systèmes de se découvrir mutuellement pour changer des programmes et communiquer entre eux. Ces travaux s'inscrivaient dans la continuité du stage de DEA de L. Ciarletta [Ciar98] que j'ai également co-encadré et où était étudiée l'approche sécurisée des mécanismes d'autoconfiguration IPv6 en intégrant au maximun l'architecture de sécurité IPSec.

Le projet `AirJava` a évolué vers le projet `Aroma` moins focalisé sur les aspects technologiques et se proposant d'identifier la problématique de standardisation, de mesure et d'interopérabilité induite par l'informatique ambiante.

Les études du NIST se sont ainsi orientées vers la définition de modèles et d'architectures pour concevoir et évaluer les technologies de l'informatique ambiante avec l'évaluateur EXiST [CID01]. Nous avons, de notre côté, insisté sur les aspects sécurité en mettant en avant une architecture de sécurité, appelée `VPSS` [CCA02, CC02], pour l'accès aux ressources dans les réseaux ambiants.

4.2.2 Motivation

L'informatique ambiante correspond à de l'informatique fortement intégrée dans notre quotidien. Par conséquent, la sécurité, le respect de la vie privée et la sûreté de fonctionnement constituent des points forts pour l'acceptation de l'omniprésence de l'informatique.

On retrouve les principaux services de sécurité à assurer comme l'authentification et la confidentialité des données. Le contrôle d'accès est également un élément clé ; que ce soit un contrôle basé sur l'identité des individus mais aussi sur leurs compétences ou leurs rôles. Ceci

[32] National Institute of Standards and Technology
[33] `http://www.nist.gov/aroma`
[34] `http://java.sun.com/developer/products/jini/`

explique pourquoi nous avons regardé de plus près la notion de RBAC (*Role Based Access Control*) [FSG+01] qui est une méthode de contrôle d'accès aux ressources informatiques dans des environnements multi-utilisateurs. Les rôles sont définis après une analyse des mécanismes de fonctionnement de l'organisation. Les droits d'accès sont regroupés par rôle et seuls ceux qui peuvent jouer ce rôle accèdent à la ressource. Parmi les fonctionnalités qui en font une architecture puissante on trouve la hiérarchie des rôles. Un utilisateur peut avoir plusieurs rôles qui ne sont pas mutuellement exclusifs. Ainsi un utilisateur qui a le rôle X exerce de manière implicite le rôle Y. La hiérarchie des rôles est un moyen d'organiser le rôle pour refléter l'autorité, la responsabilité et la compétence. Le modèle RBAC peut être mis en œuvre dans plusieurs domaines qui présentent des environnements sensibles aux questions de sécurité (santé, secours, finances,...) ; le domaine de la santé est souvent cité en exemple.

4.2.3 VPSS (*Virtual Private Smart Space*)

Les VPSS ou espaces virtuels privés intelligents [CCA02] représentent une architecture de contrôle et de sécurisation de l'accès aux ressources des environnements d'informatique ambiante. Ils s'appuient sur la philosohie de RBAC qui exprime que l'accès aux ressources doit être autorisé ou refusé selon le rôle de l'utilisateur. Ils utilisent de plus la notion de réseau privé virtuel VPN (*Virtual Private Network*) pour permettre de créer des environnements où chaque VPN correspond à un rôle.

L'architecture VPSS ne résout pas les problèmes de sécurité particuliers de bas-niveau liés aux technologies sans fil mais correspond plutôt à une méthodologie de haut-niveau, indépendante du médium de communication utilisé, qui s'appuie sur les rôles et les politiques de sécurité associés. L Ciarletta, dans le cadre de sa thèse, a développé un prototype de VPSS sous Linux. Le protocole IPsec a été utilisé pour fournir les mécanismes de sécurité sous-jacents afin de sécuriser les canaux de communication. Le mécanisme d'alias IP permet de donner à chaque nœud plusieurs adresses IP par interface. Chaque rôle correspond à un VPN et chaque VPN est un sous-réseau virtuel protégé par IPsec.

Le premier rôle/sous-réseau est particulier puisqu'il s'agit du point d'entrée où tout utilisateur doit s'enregistrer avec le serveur de contrôle d'accès. Celui-ci renvoie les informations concernant les configurations physique et logique des autres rôles/sous-réseaux accessibles (clés des SA IPsec, configuration réseau,...). Le premier sous-réseau est donc générique et est un "rôle" limité que nous appelons le rôle invité ou rôle 0. Ce rôle n'est qu'un point d'entrée au réseau et au serveur de politiques de sécurité et de droits d'accès qui fournit les informations de configuration correspondant aux rôles qui peuvent être assumés par les utilisateurs à un moment donné. Sur un sous-réseau donné, les utilisateurs ne peuvent accéder qu'aux services enregistrés dans le rôle correspondant. Un service peut être disponible pour plusieurs rôles en même temps.

4.3 Sécurité et réseaux ad hoc

4.3.1 Contexte

Les réseaux ambiants sont fondés sur les systèmes embarqués, la continuité de la communication avec les réseaux sans fil et les interfaces utilisateurs intelligents. Nos travaux ultérieurs ont porté sur la sécurité des réseaux mobiles que nous avions étudiés dans le cadre du DEA de A. Peyronnel [Pey01] et de celui de A. Ait Ali [Ait02]. Nous nous sommes focalisés sur les réseaux ad hoc pour leurs propriétés de réseaux spontanés et de dynamicité élevée.

Nos contributions ont essentiellement porté sur l'adaptation des mécanismes de sécurité disponibles pour les services multicast dans le contexte d'un réseau ad hoc. Nous nous sommes intéressés d'une part aux aspects distribution de clé avec l'adaptation du protocole `Baal` aux environnements ad hoc et la diffusion sécurisée d'un flux multicast multi-sources séquentielles réalisée avec le protocole `BALADE`, d'autre part au service d'authentification qui représente la clé de voûte de toute architecture de sécurité. Ces trois contributions ont été apportées dans le cadre de la thèse de M.S Bouassida et sont détaillées après la présentation de nos motivations.

4.3.2 Motivation

De par la nature même des réseaux ad hoc, du niveau de sécurité à instaurer et des caractéristiques et types des applications à sécuriser, le déploiement des réseaux ad hoc associé à la disponibilité des services multicast fait émerger de nouveaux défis [BCG04].

Caractéristiques des réseaux ad hoc

Les principales contraintes induites par l'environnement ad hoc sont les suivantes :

- l'utilisation des liens sans fil rend un réseau ad hoc facilement exposé à des attaques malicieuses passives comme des écoutes clandestines, ou actives comme le renvoi d'un message ou sa déformation ;
- l'absence d'infrastructure fixe est l'une des principales caractéristiques des réseaux ad hoc. Elle élimine toute possibilité de pouvoir établir une référence centralisée afin de concentrer les accès au réseau en un point unique capable d'administrer les différents services indispensables pour le bon fonctionnement du réseau. De cette absence d'infrastructure, il découle que les modèles classiques centralisés ou hiérarchiques d'authentification ou de distribution de clés peuvent difficilement s'appliquer. C'est le cas du modèle de confiance des Infrastructures à Clés Publiques ou PKI [HFPS99] ;
- 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 ;
- 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 groupe et par conséquent ne doit pas être obligé à chaque fois de s'authentifier auprès de la source du groupe auquel il appartient. En plus, ce service d'authentification doit être assez efficace et nécessiter le moins de messages possibles ;
- 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] ;
- finalement, on doit instaurer un modèle de confiance capable de répondre à ces différentes questions : à quelles entités du réseau doit-on faire confiance pour assurer le service d'authentification, quel niveau de confiance faut-il leur donner,...

Travaux liés

Peu de résultats ont été publiés concernant la sécurité des communications de groupe associée aux réseaux ad hoc, même si on assiste depuis peu à une augmentation des travaux dans ce domaine tant pour la distribution des clés de groupe que pour l'authentification :

Pour assurer la gestion sécurisée de la clé de groupe, la proposition de [VHS01] se situe dans le cadre d'une architecture NTDR (*Near Term Digital Radio*) composée d'un ensemble de *clusters*

ou petits groupes ; chacun contenant une tête de cluster qui, interconnectée aux autres têtes contribue, quand elles sont reliées ensemble, à l'ossature de routage. La génération et la distribution de la clé sont assurées en ayant deux types de clés : la clé de groupe du cluster utilisée pour chiffrer tout le trafic à l'intérieur du cluster et la clé de chiffrement de clé qui est un secret partagé entre la tête du cluster et un nœud. La communication inter-clusters est restreinte aux têtes de clusters. Si la structure de contrôle basée sur les clusters permet une utilisation plus efficace des ressources, elle génère plus de calculs à cause du découpage en sous-groupes, de l'élection du leader dans chacun des groupes. De plus, la gestion de la clé de groupe est vulnérable car elle est établie entre les têtes de clusters qui peuvent être compromises. Un surcoût de calcul est aussi généré à cause des mouvements possibles des têtes de clusters.

Dans la proposition de [CH03], la clé de groupe est gérée via le nombre de Prüfer [Prü18] et le protocole d'échange de clés GDH (*Group Diffie Hellman*) [STW96]. Le nombre de Prüfer permet de coder l'arbre multicast afin de diffuser la clé de groupe en se basant sur la topologie du réseau et sur le routage multicast. Chaque nœud dans le réseau peut ainsi facilement retrouver l'arbre multicast et déterminer s'il doit ou non relayer le paquet. L'échange de la clé de groupe aux seuls membres du groupe et non pas aux nœuds relais est réalisé avec le protocole GDH. Le modèle de distribution de la clé de groupe est efficace mais ne permet pas le passage à l'échelle. L'algorithme de Prüfer ainsi que le protocole GDH génèrent aussi un surcoût important pour les nœuds du réseau qui ont une faible puissance de calcul. Le facteur de mobilité est adressé à travers l'utilisation du GPS (*Global Positioning System*). Chaque nœud diffuse à tous les autres ses mesures ; ce qui permet de mettre à jour la topologie du réseau. Cependant cette inondation est également très coûteuse.

Le protocole GKMPAN [ZSXJ04] présente une architecture de gestion de clés pour les réseaux ad hoc qui se base essentiellement sur une pré distribution de listes de clés aux membres du groupe à partir desquelles la clé du groupe sera chiffrée et propagée. La validation des listes de clés est assurée via le mécanisme d'authentification TESLA [HD03, Arc02] présenté dans la section 4.3.5. Un serveur de clés génère et distribue la nouvelle clé de groupe aux voisins immédiats qui la rediffusent à leurs propres voisins de manière sécurisée grâce aux clés pré distribuées. Une densité élevée dans le groupe est requise pour obtenir de meilleures performances. Le protocole GKMPAN offre de plus la caractéristique d'être « *partial statelessness* » car un nœud peut obtenir la clé de groupe même s'il manque certains événements de renouvellement de clés, mais le nombre d'évènements pouvant ne pas être reçus reste limité d'où la notion de « *partial* ».

Dans [LN03], les auteurs proposent de réduire les charges de calcul et de communication sur la source en ayant des membres de groupe actifs qui participent à la sécurité du groupe. Cette approche repose sur IOLUS [Mit97] mais en prenant en compte les contraintes des réseaux ad hoc. Les nouveaux membres se rattachent au nœud de l'arbre le plus proche en utilisant des informations GPS. Tous les membres du groupe peuvent agir comme agent intermédiaire afin de distribuer la clé. La fiabilité peut-être accrue en autorisant un nœud à maintenir plus qu'une liaison et la sécurité augmentée en demandant à un nouveau membre de s'authentifier auprès d'au moins k membres du groupe. Le problème de renouvellement de clés après un *leave* n'est cependant pas réellement abordé.

Pour assurer l'authentification, [VHS01] utilise des systèmes à clés publiques, des certificats et autorités de certification. Des PKIs sont supposées être dans des réseaux fixes auxquels les participants ont accès avant d'être impliqués dans un réseau mobile ad hoc. Faire l'hypothèse d'une PKI établie dans un réseau ad hoc sans infrastructure reste encore une question ouverte.

L'authentification dans [LN03] est réalisée également à travers des certificats ; seuls les nœuds avec un certificat valide peuvent accèder aux données. Cependant, le mécanisme de révocation

tel qu'il est présenté reste un point faible. Périodiquement une liste de révocation de certificats (CRL ou *Certificat Revocation List*) est envoyée en multicast depuis par exemple la source. Chaque nœud intermédiaire la rediffuse à leurs fils et suspend le flux multicast en aval jusqu'à réception d'un acquittement ; cette approche n'est pas très efficace à mettre en œuvre dans un environnement très dynamique et sensible aux pertes de paquets.

Dans la proposition décrite dans [BT03], un cadre est défini pour les réseaux de senseurs hiérarchiques. Le réseau de senseurs est composé d'une architecture trois-tiers avec des niveaux variables de possibilités de calcul et de communication : des points d'accès très puissants qui routent les paquets via des liaisons radio vers une infrastructure filaire, des mobiles de capacité moyenne qui peuvent servir de relais entre les nœuds de senseurs et le points d'accès et les nœuds de senseurs mobiles avec une faible puissance de calcul incapables de réaliser de la cryptographie à clé publique. [BT03] présente un nouveau type de certificat, appelé certificat TESLA. Ce certificat peut être utilisé par ces nœuds de senseur pour réaliser de l'authentification d'entité. Chaque nœud de relais et point d'accès sont supposés disposer d'une paire de clés RSA et de son certificat. Comme dans [VHS01], cette hypothèse peut être dans certains cas contraignante.

Orientations de recherche

Pour aborder le problème de la sécurité du multicast dans les réseaux ad hoc, nous avions le choix entre plusieurs orientations :

- reconcevoir complètement une nouvelle solution ;
- adapter des approches proposées dans le cadre de l'ad hoc pour des services point-à-point à des services multicast ;
- adapter des approches proposées dans le cadre de services multicast à des environnements ad hoc.

Nous avons choisi la dernière orientation, et ce, pour deux raisons :

1. Nos travaux précédents avaient abouti à l'architecture `Baal` de distribution de clés pour des communications de groupe dans des environnements classiques filaires et où la dynamicité réside dans l'abonnement ou le désabonnement à un groupe. Nous voulions étendre cette architecture afin qu'elle puisse être utilisée dans des réseaux plus dynamiques. Cette extension nous a ensuite amenés à proposer `BALADE`, un protocole de distribution de clés adapté à la diffusion multicast sécurisée d'un flux multi-sources séquentielles.
2. L'authentification, notamment de la source des données, est un composant important dans les communications de groupe. De nombreuses solutions dans le monde filaire ont déjà été proposées pour permettre le passage à l'échelle, la prise en compte de l'hétérogénéité des récepteurs, la dynamicité liée aux ajouts et retraits successifs à l'intérieur d'un groupe [CBA04]. L'authentification représente en effet un élément important de toute architecture de sécurité de groupe : il faut être sûr de l'origine du message qui va inonder de multiples récepteurs. L'authentification permet à un nœud de s'assurer de l'identité des nœuds avec lesquels il communique. Sans authentification, un adversaire peut communiquer avec des nœuds du réseau et ainsi bénéficier de ressources auxquelles il n'a pas le droit d'accéder. Nous avons voulu montrer comment les contraintes liées à un environnement ad hoc nous permettaient de définir des critères qui influent sur le choix de telle ou telle méthode d'authentification utilisée dans les communications de groupe.

4.3.3 Adaptation de `Baal` aux environnements ad hoc

Notre proposition [BCF04a] se veut indépendante des protocoles de routage utilisés. Pour assurer les services de sécurité des communications de groupe dans les réseaux ad hoc : authen-

tification, confidentialité, integrité et non répudiation, nous avions besoin d'une architecture de gestion des clés de groupe. Nous avons repris l'architecture fonctionnelle de `Baal` [CCS00] que nous avons étendue par deux nouveaux blocs (cf Figure 4.1) :

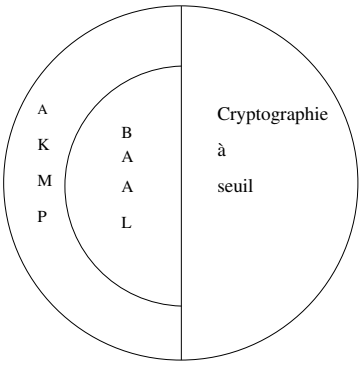

FIG. 4.1 – Architecture adaptée

- Le support hybride du protocole AKMP [BBC02] pour réduire le problème de « 1 affecte n » tout en limitant le surcoût de traitement dû au processus de chiffrement/déchiffrement. L'utilisation de ce bloc est présentée, ci-après, dans le paragraphe relatif à l'architecture de notre approche.
- la cryptographie issue de la cryptographie à seuil [ZH99] pour assurer l'authentification dans notre réseau au lieu d'établir une infrastructure PKI gérée par une autorité de certification centralisée. Le service de confiance est distribué à n nœuds spéciaux appelés serveurs. Chaque serveur possède une clé publique et privée (K_i , k_i) ainsi que les clés publiques de toutes les entités du réseau. La clé publique du service K est connue de tous ; la clé privée est divisée en n secrets : chaque secret étant détenu par un serveur. Il suffit que $t+1$ secrets parmi n avec $n >= 3t + 1$ soient récupérés pour arriver à générer la clé. La génération de la clé de groupe est également réalisée via la cryptographie à seuil à travers la combinaison de secrets envoyés par les serveurs.

Architecture

Comme dans `Baal`, les trois principaux acteurs sont le contrôleur, les contrôleurs locaux et les membres du groupe.

Le contrôleur global (CG) est la source du groupe. Initialement, cette entité détient la liste des participants ou des membres du groupe (MG) appelée `Participant_List`. Le CG est responsable de la génération et de la distribution de la clé de groupe, il assure aussi périodiquement le renouvellement de cette clé et la gestion de la sécurité du groupe (contrôle du comportement des contrôleurs locaux et des membres du groupe).

Le contrôleur local (CL). Un nœud mobile appartenant à l'arbre multicast, comme membre du groupe ou comme simple participant à l'arbre multicast, et ayant des nœuds fils vers lequels il envoie du flux multicast est considéré comme un contrôleur local passif. Si un CL passif est également membre du groupe, il détient la même clé cryptographique que son nœud parent. Quand le taux de dynamicité local atteint un certain seuil, le CL décide de commuter vers l'état actif. Par conséquent il génére une nouvelle clé locale, la distribue à tous ses membres et un processus de déchiffrement/rechiffrement peut commencer.

Nous disons que le CL actif forme avec ses membres locaux un nouveau cluster. Pour décider de son état, chaque CL détient une fonction d'évaluation de dynamicité (cf. algorithme 1) qui diffère de la fonction d'évaluation de AKMP en prenant en compte non seulement la fréquence de changement des membres mais aussi leur nombre. Ceci est nécessaire dans le cadre des réseaux ad hoc. Premièrement, tous les membres du groupe sont aussi des routeurs et peuvent donc être considérés comme des contrôleurs locaux (s'ils ont des nœuds fils). Deuxièmement, quand un membre quitte le groupe, le contrôleur actif local est obligé de renouveler la clé locale et de la distribuer en point à point à tous les membres de son cluster. Par conséquent, le temps nécessaire pour le renouvellement est proportionnel au nombre de membres dans le cluster.

Algorithme 1 Fonction d'évaluation

```
if (mcf > d1 or mn > d2) then
   //commutation processus déchiffrement/rechiffrement
   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,
// d2 : seuil de nombre de membres locaux.
```

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 4.2 illustre un exemple de calcul de mn.

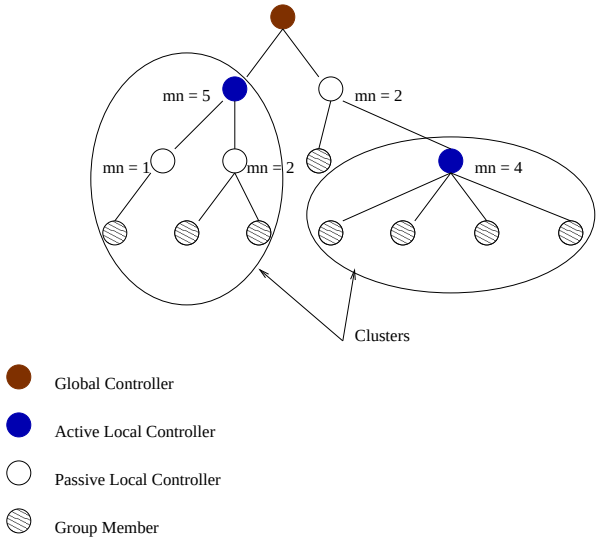

FIG. 4.2 – Exemple d'évaluation de mn

La description détaillée des opérations d'initialisation de groupe, d'ajout d'une nouvelle entité, de retrait d'un membre et de renouvellement périodique de la clé est de groupe est donnée dans [BCF04a].

Simulations et Résultats

Nous avons réalisé des simulations sur notre approche pour définir des seuils de fréquence et de nombres de membres. Nous avons ainsi mesuré le temps nécessaire pour le renouvellement de la clé de groupe après un message *Join* (Ajout) ou *Leave* (Retrait), selon la fréquence des événements et le nombre de membres dans le groupe. Ces deux seuils nous ont donné une première évaluation de notre solution.

Pour effectuer la simulation, nous avons utilisé le simulateur NS version ns2.1b9a [Lin03]. Le réseau simulé dans un réseau ad hoc est composé de 100 nœuds avec l'utilisation du protocole de routage multicast MAODV [RP00] disponible sous NS. Le déplacement des nœuds est généré aléatoirement afin de prendre en compte le facteur de mobilité dans les réseaux ad hoc. Pour générer des sessions multicast, nous avons utilisé le modèle présenté par Almeroth [AA96], qui suggère que l'arrivée des membres suit un processus de Poisson ($\lambda = 10$ arrivées par unité de temps) et que la durée d'adhésion est une distribution exponentielle (en moyenne $\mu = 145$ unités de temps). Ce modèle est déduit de sessions réelles multicast observées sur le Mbone en 1995.

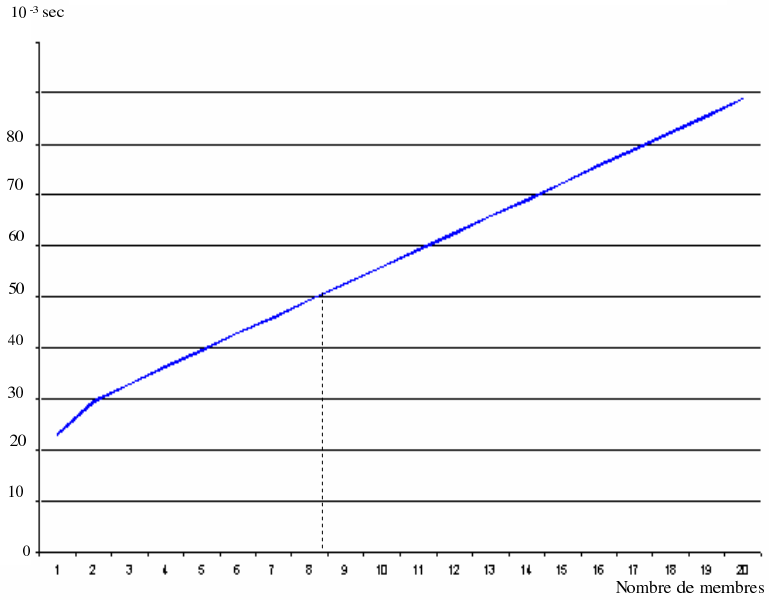

Fig. 4.3 – Temps de renouvellement de la clé suite à un Leave selon le nombre de membres

Avec le *Join* le coût pour renouveller la clé est constant en termes de messages envoyés et d'opérations de chiffrement/déchiffrement nécessaires. Cependant lors de nos simulations, nous avions observé des variations dans le temps de renouvellement de la clé qui sont notamment à imputer au protocole de routage multicast sous-jacent.

Dans le cas du *Leave*, le coût est, lui, proportionnel au nombre n de membres dans le groupe comme le montre la Figure 4.3. Ceci nous permet de définir le seuil du nombre de membres dans un cluster. Si nous prenons, par exemple, comme contrainte que le temps nécessaire ne peut excéder 0.05s, le seuil devra être de 8 membres par cluster.

Nous avons aussi calculé le temps moyen nécessaire pour le processus de renouvellement de la clé suivant une procédure de *Join* ou de *Leave* par fréquence d'événements calculés dans des intervalles de temps égaux. Si nous prenons comme contrainte le fait que le temps pour le

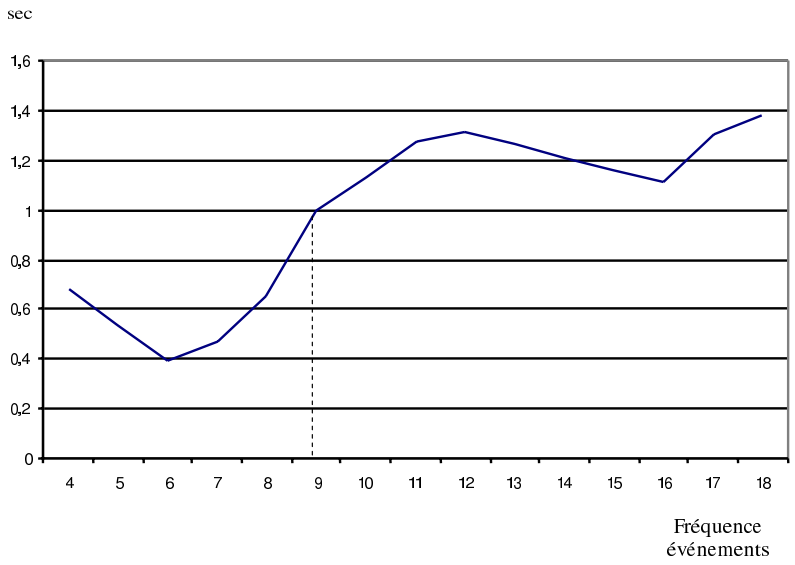

FIG. 4.4 – Temps de renouvellement de la clé par fréquence d'évènements

processus de renouvellement ne peut excéder 1s, la courbe de la Figure 4.4 montre que le seuil de fréquence ne doit pas dépasser 9 événements par unité de temps. L'allure globale de la courbe s'explique par le protocole de routage sous-jacent qui met en place des mécanismes de découverte et de maintenance de routes ; ce qui provoque certaines périodes d'instabilité.

4.3.4 BALADE

L'ensemble des travaux que nous avons réalisés sur `Baal` et sur l'adaptation de `Baal` ont mis en évidence que la définition d'une architecture globale ne peut s'avèrer efficace pour tout type d'application et que les besoins de l'application ainsi que ceux de l'environnement doivent être pris en compte. Dans ce contexte, nous sommes partis d'une classe d'application pour laquelle nous avons défini un protocole de gestion de clés adapté. Nous avons proposé `BALADE` [BBL+05, BLC04], une approche de gestion de clés du groupe dans un environnement ad hoc pour des applications de diffusion de flux multimédia (streaming) dans le cadre précis de communications de groupe multi-sources séquentielles (à tout instant t, une et une seule source émet et, une fois qu'elle termine, une autre source peut prendre le relais). Ce modèle de communication est très répandu dans l'Internet et peut être utilisé par plusieurs applications comme la vidéo à la demande, la radio internet.

L'idée de base de `BALADE` est de conserver une approche hybride en subdivisant dynamiquement le groupe multicast. Le flux multicast est chiffré par la source avec la clé de cryptage du trafic TEK et envoyé en multicast à tous les membres du groupe. La source envoie la clé TEK chiffrée à tous les contrôleurs locaux ; ce chiffrement se fait à l'aide d'une clé KEK. Les contrôleurs locaux, qui partagent avec leurs membres locaux une clé de sous-groupe, leur transmettent alors la TEK chiffrée avec cette clé de sous-groupe. Un des points forts de `BALADE` est que seule la clé TEK est chiffrée et déchiffrée et non plus les données du flux multicast.

Pour assurer l'intégrité et la confidentialité des données un processus de renouvellement de la TEK doit être lancé côté source à chaque unité sémantique de données dépendant de l'application en question. Ainsi une source qui diffuse un flux MP3 va changer la TEK à chaque titre alors qu'une source qui diffuse un flux vidéo va renouveler sa TEK à chaque film, ou à chaque chapitre

d'un film. Cette solution est réaliste dans le cadre de notre application et n'oblige pas à changer la clé à chaque évènement de *Join* ou de *Leave* provoqué par les récepteurs ce qui est très contraignant dans un environnement ad hoc. Le changement de clé est orienté émetteur et par conséquent suit la faible dynamicité de la source et non pas la forte dynamicité des membres du groupe dans les approches orientées récepteur.

Pour assurer l'authentification des membres et des sources dans le cadre de notre approche, nous avons choisi d'utiliser les identificateurs cryptographiques ou CBIDs (*Crypto-Based Identifiers*) [MC02, BCK96] afin d'obtenir le couple clé privée/clé publique et d'éviter ainsi la mise en place d'une infrastructure PKI. Les identifiants cryptographiques sont statistiquement uniques et cryptographiquement vérifiables, 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. Ces identificateurs permettent d'avoir une forte liaison cryptographique avec leurs composants (clés privée et publique) ; en effet les identificateurs sont dérivées de la clé publique et le nœud peut prouver sa possession du CBID en signant les paquets avec sa clé privée correspondante. C'est exactement le but des certificats. Ainsi, chaque membre du groupe génère préalablement une clé publique et une clé privée et calcule son CBID.

Dans ce qui suit va être présenté le protocole `BALADE` en insistant plus spécialement sur les différences par rapport au protocole défini dans la section précédente.

Gestion des membres du groupe : clusterisation dynamique

Le contrôleur global (CG) correspond à la source du groupe. Ainsi, avec une architecture multi-sources séquentielles, à un instant donné il existe un seul CG au sein du groupe. Le CG assure aussi le renouvellement de la clé de chiffrement de trafic (TEK) à chaque unité de données, selon la sémantique du flux.

Le contrôleur local (CL) doit être un nœud mobile membre du groupe. Formant avec ses membres locaux un sous-groupe ou cluster, il doit générer et distribuer une clé à ses différents membres locaux. Cette clé est appelée KEK_{CSG} : clé du sous-groupe. Le CL doit acheminer la clé de chiffrement de trafic, envoyée par la source, et chiffrée avec KEK_{CSG} à tous ses membres locaux. C'est pour cette raison qu'un contrôleur local doit impérativement être un membre du groupe. La fonction d'évaluation utilisée pour qu'un membre décide de passer à l'état CL est la même que celle définie précédemment dans la section 4.3.3.

Gestion et distribution de clés

À l'initialisation de l'application, tous les membres du groupe reçoivent en point à point de la part de la source, la clé de session notée KEK_{CSG-0} (clé du sous-groupe 0), chiffrée avec leurs clés publiques respectives ; 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 KEK_{CSG-i}, générée par le CL_i. Pour envoyer la clé TEK à tous les membres du groupe, la source chiffre cette clé avec KEK_{CSG-0} et l'envoie en multicast à tous les membres de son sous-groupe. Puis elle envoie cette clé TEK au groupe formé par les contrôleurs locaux, chiffrée avec la clé KEK_{CCL}. Les contrôleurs locaux appartenant à ce sous-groupe vont décrypter le message, extraire la TEK, la recrypter avec leur clé de sous-groupe et l'envoyer à tous leurs membres locaux. L'avantage de cette solution est de minimiser 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. Chaque nouvelle source doit joindre le groupe des contrôleurs locaux, pour pouvoir envoyer la clé de chiffrement du trafic chiffrée à tous les autres CLs. Une illustration de la distribution de TEK

est présentée dans la figure 4.5. Le détail des opérations d'ajout et de retrait d'une entité du groupe est donné dans [BLC04, BBL+05].

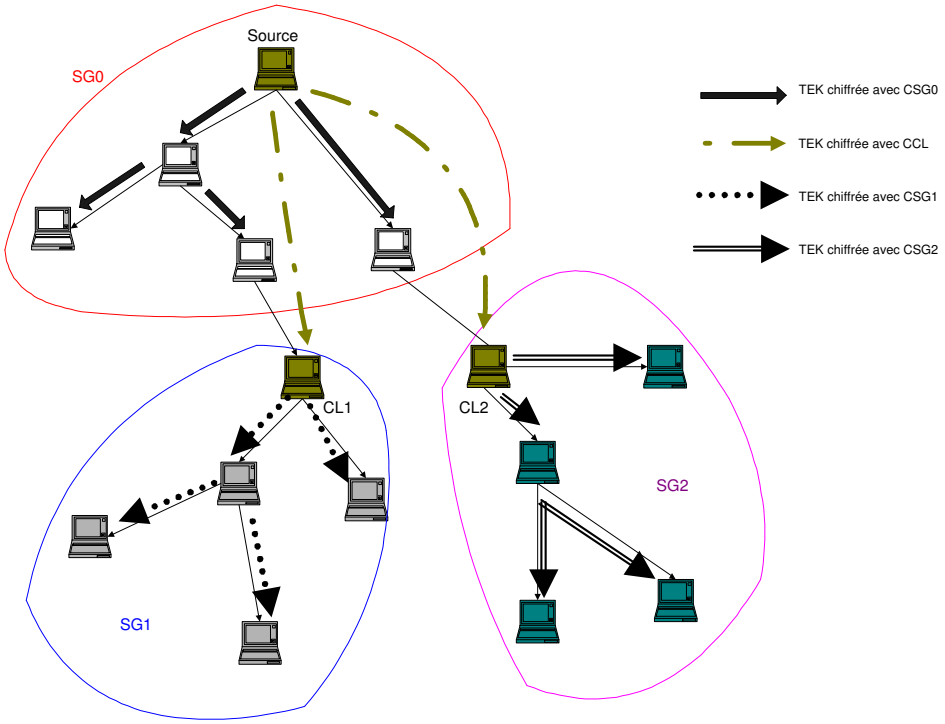

Fig. 4.5 – Distribution de TEK

Authentification et contrôle d'accès

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. Pour l'authentification et le contrôle d'accès, nous distinguons deux cas :

Dans le premier cas, un nouveau nœud demande à un membre du groupe choisi selon le protocole de routage multicast sous-jacent, à joindre le groupe multicast. Pour cela, il lui envoie un message de demande d'adhésion au groupe, contenant son CBID. Le membre du groupe envoie un message de vérification de contrôle d'accès à son contrôleur local. Si l'authentification et le contrôle d'accès réussissent, le membre parent se charge d'activer la route de l'arbre multicast vers le nouveau nœud et lui envoie en plus, un message d'acceptation d'adhésion, chiffré avec la clé publique du nœud et contenant la clé TEK courante et un mot de passe. Ce mot de passe que nous appelons ticket, sera utilisé lors de la ré-authentification du nouveau membre.

Dans le deuxième cas, il s'agit d'un membre du groupe déjà authentifié mais qui perd la connectivité, cas très probable dans le cadre des réseau ad hoc : nœud qui se déplace d'un sous-groupe à un autre, ou qui s'arrête brusquement suite à problème de batterie. Quand le nœud est à nouveau connecté, il envoie à un membre du groupe un message de demande d'adhésion au groupe contenant le 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. Le ticket, commun à tous les membres du groupe, est chiffré avec la clé de chiffrement de données. Ainsi, ce ticket n'est plus valable après un renouvellement de la TEK.

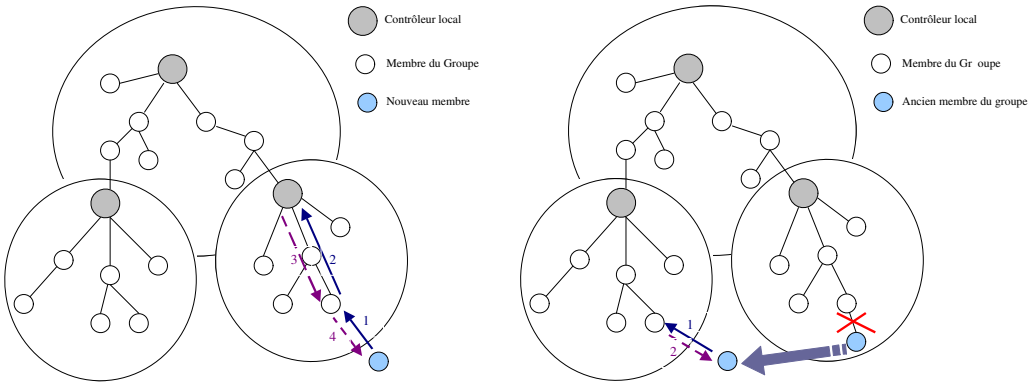

FIG. 4.6 – Authentification d'un nouveau et ancien membre

Implantation

Le modèle proposé est en cours d'implantation dans une application de *juke-box* collaboratif permettant la diffusion et l'écoute de flux MP3 sur un réseau ad hoc. L'application a été réalisée par A. BRUNETON dans le cadre de son stage d'école d'ingénieurs et par A. LAHMADI en tant qu'ingénieur expert. Les différents nœuds sont rassemblés en groupes qu'un utilisateur peut choisir de rejoindre ou de créer. Chacun de ces groupes propose une *playlist* commune, dans laquelle l'utilisateur peut choisir d'ajouter, à la manière d'un *juke-box* classique, la ou les chansons qu'il aimerait voir jouer. Cette *playlist* est synchronisée sur l'ensemble du groupe tout en étant

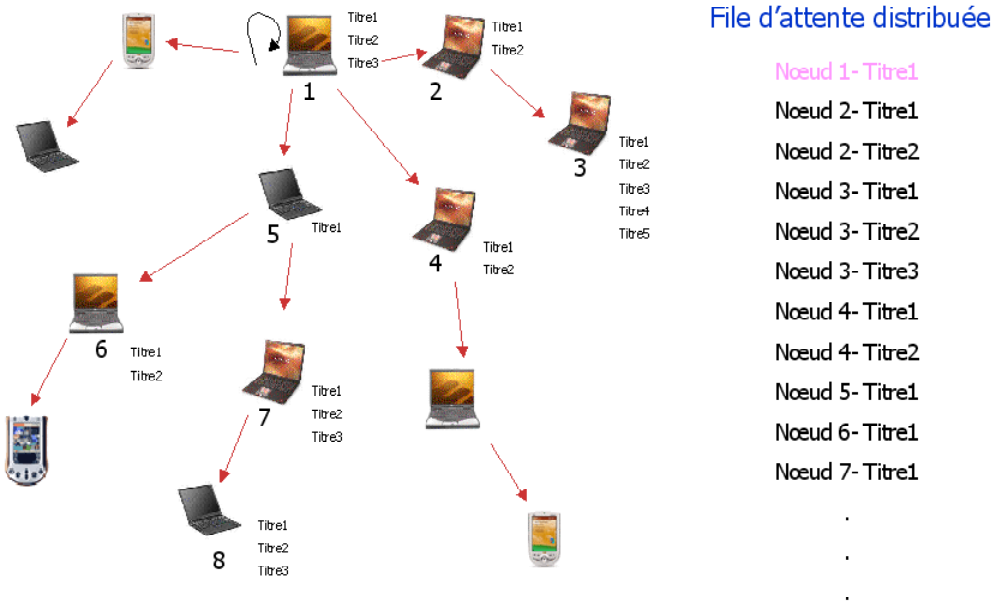

FIG. 4.7 – Juke-box dans un réseau ad hoc

totalement distribuée (tous les nœuds sont équivalents) (cf Figure 4.7). Tout nœud implante trois

modules principaux :

- le module de *streaming des fichiers MP3* (sur IP multicast), via RTP, avec une adresse IPv6 multicast spécifique. Il gère aussi le (dé-)cryptage des informations ainsi que la reconstitution des flux reçus.
- le module de *synchronisation des données* qui s'appuie sur le modèle pair à pair JXTA[35] qui offre, via les annonces, des mécanismes de diffusion de groupe, et la possibilité d'établir une liaison point-à-point entre deux pairs ne disposant initialement d'aucune information l'un sur l'autre. La synchronisation des données à proprement parler se fait grâce à l'algorithme SOCT 2 [SCJF98] et à l'utilisation d'horloges vectorielles de taille variable. Les opérations concurrentes, c'est-à-dire émises « simultanément » par deux pairs, sont réordonnées (« transposées »), et les conflits sont correctement gérés.
- le module de *sécurité* constitué de deux composants : 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. Dans sa version initiale, `BALADE` utilise comme protocole de routage multicast MAODV [RP00], en raison de la disponibilité de son implémentation. L'algorithme de cryptage que nous avons retenu est AES [Fed01, SKW+99]. Lors de l'adhésion à un groupe multicast, une application émet une demande d'adhésion en spécifiant l'adresse de groupe au module de gestion multicast qui envoie un message *REPORT* en utilisant le protocole *IGMP* ou *MLD* suivant la valeur de l'adresse du groupe. Le protocole *MAODV* filtre ce message *REPORT* et génère un message *RREQ (Route Request)* qui permet de faire la demande d'adhésion. Pour assurer l'authentification des nœuds, les données d'authentification sont ajoutées sous forme d'une extension dans les messages de MAODV.

4.3.5 Évaluation des méthodes d'authentification des flux multicast

Nous avons travaillé sur le service d'authentification qui permet à un nœud de s'assurer de l'identité des entités avec lesquelles il communique. Nous nous sommes intéressés plus spécifiquement au service d'authentification de la source d'un flux multicast ; l'authentification des membres relevant davantage du contrôle d'accès. Nous avons proposé une taxonomie des approches d'authentification de la source dans les communications de groupe et étudié leur adéquation dans le cadre de l'ad hoc en fonction de critères que nous avons définis [BCF04b, BBC04].

Classification

Nous sommes partis de la classification de Eskicioglu [Esk02] que nous avons présentée dans le chapitre précédent et qui considère trois niveaux pour l'authentification de la source à savoir l'authentification de groupe, l'authentification des données de la source et l'authentification individuelle. Pour chacun de ces niveaux, nous avons retenu l'approche la plus représentative.

L'approche « Key Agreement » [JV96, AG00] fait partie des protocoles d'établissement de clés pour l'authentification de groupe. Le contexte de « Key Agreement » est un petit groupe de personnes dans une conférence, présentes ensemble dans une salle, et voulant s'échanger des données secrètement durant la durée de la réunion. Les nœuds ont confiance les uns aux autres et partagent un mot de passe faible à partir duquel un mot de passe fort sera généré. Ce mot de passe fort, composée des contributions des différents participants, sera la clé de session du groupe utilisée pour signer les paquets envoyés en multicast.

[35] http ://www.jxta.org

L'approche TESLA [HD03] (*Timed Efficient Stream Loss-Tolerant Authentication*) permet d'authentifier les données de la source et fournit une authentification basée sur le calcul d'un MAC (*Message Authentication Code*) pour chaque paquet émis par la source. Le principe général consiste à dévoiler dans un paquet envoyé à l'instant j une clé utilisée dans les paquets émis à l'instant $j-d$, d étant le délai de révélation de la clé. Le flux de données dans TESLA est tolérant aux pertes et unidirectionnel : les données transitent seulement de la source vers les récepteurs. Le surcoût de l'authentification de la source est indépendant du nombre de récepteurs.

L'approche FEC (Forward Error Correction) est une technique de recouvrement de pertes de paquets. Son principe est le suivant : pour transmettre k paquets de données, on transmet en plus h paquets redondants. Pour générer les paquets redondants, on utilise un codeur et un décodeur FEC. [PM03] combine des techniques de hachage et de FEC pour assurer une authentification efficace des paquets multicast ; cette solution appartient aux méthodes permettant l'authentification individuelle de la source car la non répudiation est assurée. Les paquets d'un bloc sont la combinaison de données et de tags d'authentifications pour le bloc courant ou le bloc suivant/précédent.

Critères

Pour pouvoir juger de l'adéquation de ces méthodes d'authentification dans le cadre des réseaux ad hoc, nous avons défini un certain nombre de critères d'analyse et de comparaison pertinents dans le monde ad hoc à savoir :

- la *robustesse* qui est la capacité de l'architecture d'authentification à réagir face aux pertes de données ;
- l'*accessibilité* qui représente la capacité des récepteurs à accéder au service de réception du flux multicast et à authentifier les paquets depuis n'importe quel point du flux ;
- le *stockage des données* ou le nombre de paquets maximum que la source ou les récepteurs doivent stocker ;
- le *délai d'authentification* ou le nombre de paquets maximum que les récepteurs doivent recevoir pour pouvoir authentifier le premier paquet ;
- le *coût en terme de puissance de calcul*, c'est-à-dire le temps passé à effectuer des opérations de chiffrement/déchiffrement et/ou de hachage ;
- le *surcoût en terme de bande passante* qui correspond au nombre d'octets supplémentaires dédiés à l'authentification ;
- le *passage à l'échelle* de l'architecture d'authentification ou la capacité des approches à pouvoir augmenter le nombre de récepteur potentiels.

Évaluation des critères dans le contexte de la classification

Nous avons ensuite comparé les trois approches selon ces critères. La synthèse de cette comparaison est présentée dans le tableau 4.1 où d est le délai de révélation des clés, b le nombre de paquets par blocs et p le taux de pertes par bloc. Une présentation plus détaillée et quantitative se trouve dans [BCF04b].

De l'étude que nous avons effectuée, il apparaît clairement qu'il n'existe pas une solution optimale qui résoudrait l'authentification de la source pour les communications de groupe dans les réseaux ad hoc, mais que l'importance de certains critères permet d'influer sur le choix de telle ou telle méthode d'authentification.

Ainsi pour pouvoir utiliser les protocoles de « Key Agreement » des adaptations s'avèrent nécessaires. Cette approche ne prévoit pas de solutions contre les pertes de données. De plus, afin d'assurer le même niveau d'authentification, il faut recalculer la clé de session à chaque

changement de membre du groupe (*Join* ou *Leave*). On se retrouve devant la problématique « 1 affecte n » car la clé de session est l'ensemble de contributions de tous les participants du groupe. La solution serait donc de limiter le calcul de la clé du groupe à un nombre restreint de membres. Ce nombre ne pouvant pas être égal à 1 pour ne pas créer un point de vulnérabilité au réseau. Limiter le calcul de la clé à k contributions de nœuds qu'on appelle serveurs, permettrait aussi le passage à l'échelle. Un nœud participant à la première phase de détermination de la clé de session du groupe, peut disparaître à tout moment (de par sa mobilité ou suite à un problème de batterie) ; ce nœud peut aussi être le leader du groupe. Le protocole doit donc prendre en compte cette possibilité et prévoir des solutions dynamiques appropriées (élection d'un nouveau leader, élimination de la contribution d'un nœud injoignable, ...).

De par ses propriétés, l'approche TESLA s'adapte au contexte des réseaux ad hoc car elle offre une authentification efficace de la source, une forte robustesse contre les pertes de paquets, une forte extensibilité et un surcoût minimal ; au détriment d'une perte de temps pour la synchronisation initiale et une authentification différée des paquets. Cependant, l'initialisation de TESLA est effectuée par l'envoi d'un paquet multicast à tous les membres du groupe, signé avec la clé privée de la source. Ceci nécessite que les récepteurs connaissent la clé publique de la source et donc qu'une PKI soit établie au sein du réseau. Pour cela, on pourrait mettre au point une authentification avec cryptographie à seuil qui permettrait d'émuler une autorité de certification. Une solution plus simple consisterait à utiliser les identificateurs cryptographiques [MC02] pour assurer la non répudiation de la source lors de l'envoi du premier paquet d'initialisation. La synchronisation de la source et des membres dans un milieu ad hoc peut s'avérer beaucoup plus compliquée que les réseaux filaires. Il existe une version de TESLA adaptée aux réseaux de sondes, qui s'appelle μTESLA [PSW+02]. Cette approche présente l'avantage d'être adaptée à des environnements de nœuds à ressources limitées, ce qui la rend plus adéquate pour les réseaux ad hoc. Elle utilise des mécanismes symétriques pour l'authentification du paquet initial et ne dévoile la clé que par période et non à chaque paquet. Les améliorations de cette approche par rapport à TESLA sont détaillées dans [PSW+02].

	Robust.	**Access.**	**Stockage des données**	**Délai d'authentification**	**Surcoût Puissance calcul**	**Surcoût bande passante**	**Passage à l'échelle**
Key Agreement	Non	Non	Non	4 paquets	Cryptage/ Décryptage	Authentification "hors bande"	Non
TESLA	Oui	Difficile	Oui (max : paquets reçus en d unités de temps	1 + paquets reçus en d unités de temps	1 opération de hachage par paquet	Varie selon les fonctions de hachage (env. 24 octets par paquet)	Oui
Approche FEC	Oui	Oui	Oui (max 2 blocs)	$MAX = 2 * b$	b opérations de hachage + vérification de signature par bloc de données	Dépend de b et de p	Oui

TAB. 4.1 – Critères pour la sécurité multipoint

Dans [PM03], les auteurs proposent un schéma d'authentification utilisant FEC pour des flux

multicast en temps réel, à un nombre illimité de récepteurs. Cette architecture offre l'authentification, la non répudiation de la source, et l'intégrité des données. L'aspect temps réel de cette architecture est bien réalisé grâce au faible surcoût de communication, et à la forte tolérance aux pertes de paquets. Comme TESLA, la source a besoin d'avoir une paire de clés (publique et privée).

Ainsi, pour authentifier efficacement un flux de données multicast dans un groupe de nœuds ad hoc, et n'ayant pas besoin d'assurer la confidentialité, l'authentification avec TESLA paraît une des solutions les plus appropriées même si le délai d'authentification et le stockage des données demeurent des points contraignants. Cependant, si on a besoin d'assurer la non répudiation et l'authenticité de la source tout en tolérant les pertes de paquets, l'authentification avec FEC devient la solution la plus adéquate.

Les résultats de cette étude ont été utilisés dans le cadre du projet RNRT SAFECAST[36] où l'un des objectifs est de définir un protocole d'authentification pour des applications de groupe de type PMR (*Professional Mobile Radiocommunication*) utilisées par des forces civiles ou militaires, présentant des caractéristiques communes avec les réseaux ad hoc [BBC04].

4.4 Conclusion

Les réseaux spontanés sont devenus une réalité mais leur déploiement à grande échelle reste conditionné à la prise en compte des aspects sécurité dans ces environnements pour la plupart sans fil. Notre contribution relative à l'informatique ambiante a permis de concilier de manière originale la notion de rôles à celle de VPN. Ce travail est poursuivi actuellement par L. Ciarletta au sein de Madynes sur les mécanismes de configuration dynamique des VPSS avec des technologies actives XML dans le cadre du projet RNRT SWAN[37] (*Self aWare mANagement*) qui se propose de développer et d'expérimenter des méthodes de « gestion autonome ».

Sécuriser les communications de groupe dans un environnement ad hoc est un véritable défi. Tout d'abord, les applications multicast comme les conférences virtuelles sur Internet et la diffusion de flux multimédia présentent plus de vulnérabilité en terme de sécurité que les communications point à point et ont 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,...De plus la taille et la dynamicité des groupes peuvent être plus importantes dans les réseaux ad hoc et le nombre de membres ainsi que la fréquence d'adhésion au groupe sont plus difficilement contrôlables. La mobilité des nœuds doit également être prise en compte.

Notre proposition initiale `Baal` utilise une seule clé de chiffrement de trafic ou TEK (*Traffic Encryption Key*). La gestion du groupe selon `Baal` est à la charge du contrôleur global et peut être déléguée aux contrôleurs locaux des sous-groupes. Cependant `Baal` souffre encore, dans une certaine mesure, du phénomène « 1 affecte n » car, à chaque évènement dans le groupe correspondant à un *Join* ou *Leave*, la KEK et la TEK sont renouvelées pour chaque membre du groupe. Même si le changement de la clé est réalisé via un message multicast, tous les membres restent concernés.

L'adaptation de `Baal` aux environnements ad hoc a pris en compte ce fait en utilisant le principe des approches hybrides qui divisent le groupe multicast dynamiquement en des sous-groupes selon des critères de dynamicité de manière à réduire le surcoût du processus de cryptage/décryptage. Le nombre de clés TEK varie de 1 à p selon le nombre de sous-groupes créés.

[36] `http:www.telecom.gouv.fr/rnrt/rnrt/projets/safecast.htm`
[37] `http://www.telecom.gouv.fr/rnrt/rnrt/projets/SWAN.htm`

Cependant, si le surcoût lié au cryptage/décryptage est diminué, il n'est pas complètement supprimé.

La définition d'une architecture de sécurité générique ne peut être efficace pour toutes les classes d'application au vue des différentes contraintes inhérentes aux applications sur le délai d'authentification, la bande passante, le temps d'accessibilité,.... C'est ce que nous avons montré, dans le cadre de l'authentification de la source.

Pour la poursuite de nos travaux relatifs à la distribution de clés, nous nous sommes focalisés sur le service de gestion de clés pour la sécurisation des communications de groupes dans le contexte précis de multi-sources séquentielles avec l'approche appelée `BALADE` qui reste une approche hybride, en divisant dynamiquement le groupe multicast en clusters mais dont le changement de clés n'est plus orienté récepteurs, mais déterminé par l'émetteur.

Ces études relatives au multicast et aux réseaux ad hoc sont poursuivies dans le cadre des projets RNRT SAFARI[38] et SAFECAST[39].

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.

[AG00] N. Asokan and P. Ginzboorg. Key-Agreement in Ad-Hoc Networks. *Computer Communications*, 23(17) :1627–1637, February 2000.

[Arc02] M. Archer. Proving Correctness of the Basic TESLA Multicast Stream Authentication Protocol with TAME. In *Workshop on Issues in the Theory of Security, 2002*, 2002.

[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.

[BCK96] M. Bellare, R. Canetti, and H. Krawczyk. Keying Hash Functions for Message Authentication. *RSA CryptoBytes*, 2(1), 1996.

[BT03] M. Bohge and W. Trappe. An Authentication Framework for Hierarchical Ad Hoc Sensor Networks. In *WISE'03, San Diego, California, USA*, September 2003.

[CBA04] Y. Challal, H. Bettahar, and Bouabdallah A. A taxonomy of multicast data origin authentication : Issues and solutions. *IEEE Communications Surveys and Tutorials*, 6(3) :12–25, July 2004.

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

[CH03] T. Chiang and Y. Huang. Group Keys and the Multicast Security in Ad Hoc Networks. In *Proceedings of the 2003 International Conference on Parallel Processing Workshops (ICPP 2003 Workshops)*, 2003.

[CID01] Laurent Ciarletta, Vassil Iordanov, and Alden Dima. Using Intelligent Agents to assess Pervasive Computing Technologies. In *International Conference on Intelligent Agents, Web Technology and Internet Commerce - IAWTIC 2001, Las Vegas, USA*, Jul 2001.

[38] `http://www.telecom.gouv.fr/rnrt/rnrt/projets/res_02_04.htm`

[39] `http:www.telecom.gouv.fr/rnrt/rnrt/projets/safecast.htm`

[Esk02] A. Eskicioglu. Multimedia security in group communications : Recent progress in wired and wireless networks. In *Proceedings of the IASTED International Conference on Communications and Computer Networks, pp. 125-133*, pages 125–133, Cambridge,MA, November 2002.

[ETS96] ETSI. High Performance Radio Local Area Network (Hiperlan), draft standard ETS 300652, March 1996.

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

[FSG+01] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R Chandramouli. Proposed nist standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3) :224–274, 2001.

[HD03] T. Hardjono and L. Dondeti. *Multicast and Group Security.* Computer Security Series. Artech House, Librarie Eyrolles, 2003.

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

[JV96] M. Just and S. Vaudenay. Authenticated Multi-Party Key Agreement. In *ASIACRYPT : Advances in Cryptology – ASIACRYPT : International Conference on the Theory and Application of Cryptology.* LNCS, Springer-Verlag, 1996.

[Len02] J. Leneutre. Authentification dans les Réseaux Ad Hoc : Problématique et Etat de l'Art. In *SAR'2002*, Telecom Paris, July 2002.

[Lin03] C. Lindemann. Wireless multicast extensions to AODV/DSR in ns-2.1b9a, 2003. http ://rul-www.cs.uni-dortmund.de/MobileP2P/mainE.html.

[LN03] G. Lin and G. Noubir. Secure Multicast over Multihop Wireless Ad Hoc Networks. In *Workshop on Mobile Ad Hoc Networking and Computing*, March 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.

[Mil99] K.L. Mills. AirJava : Networking for Smart Spaces. In *Usenix : Proceedings of Embedded Systems Workshop*, Cambridge, Massachussets, USA, March 1999.

[Mit97] S. Mittra. Iolus : A Framework for Scalable Secure Multicasting. In *SIGCOMM*, pages 277–288, 1997.

[PM03] A. Pannetrat and R. Molva. Efficient Multicast Packet Authentication. In *The 10th Annual Network and Distributed System Security Symposium*, San Diego, California, February 2003.

[Prü18] H. Prüfer. Neuer Beweis eines Satzes über Permutationen Archiv für Mathematik and Physik. *Archiv für Mathematik und Physik*, 27 :742–744, September 1918.

[PSW+02] A. Perrig, R. Szewczyk, V. Wen, D. Tygar, and D. Culler. Spins : Security protocols for sensor networks. In *Wireless Networks*, volume 8, pages 521–534. Kluwer Academic Publishers, 2002.

[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.

[SCJF98] M. Suleiman, M. M. Cart, and J. J. Ferrie. Concurrent Operations in a Distributed and Mobile Collaborative Environnement. In *International Conference on Data Engineering (Orlando, Floride, USA)*, 1998.

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

[STW96] M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman Key Distribution Extended to Group Communication. In *ACM Conference on Computer and Communications Security*, pages 31–37, 1996.

[VHS01] V. Varadharajan, M. Hitchens, and R. Shankaran. Securing NTDR Ad-Hoc Networks. In *IASTED International Conference on Parallel and Distributed Computing and Systems 2001*, pages 593–598, Anaheim California, August 2001.

[Wei91] M. Weiser. The Computer for the Twenty-First Century. *Scientific American*, 265(3) :94–104, September 1991.

[ZH99] L. Zhou and J. Haas. Securing Ad Hoc Networks. *IEEE Network*, 13(6) :24–30, 1999.

[ZSXJ04] S. Zhu, S. Setia, S. Xu, and S. Jajodia. GKMPAN : An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks . In *Proc. of the 1st ACM International Conference on Mobile and Ubiquitous Systems (Mobiquitous*, Boston, Massachussets, USA, August 2004.

Publications

Conférences

[BBL+05] M.S. Bouassida, A. Bruneton, A. Lahmadi, I. Chrisment, and O. Festor. Balade : diffusion multicast sécurisée d'un flux multimédia multi-sources séquentielles dans un environnement ad hoc. *Colloque Francophone sur l'Ingéniérie des Protocoles (CFIP'05)*, Bordeaux, France, Mars 2005, 17p (taux d'acceptation 30%).

[BCF04b] M.S. Bouassida, I. Chrisment, and O. Festor. Méthodes d'authentification pour les communications de groupes : taxonomie et évaluation dans un environnement ad hoc. *Sécurité et Architecture Réseaux 2004 (SAR'04)*, La Londe, France, Juin 2004, 12p.

[BCF04a] M.S. Bouassida, I. Chrisment, and O. Festor. An Enhanced Hybrid Key Management Protocol for Secure Multicast in Ad-hoc Networks. *Third IFIP-TC6 Networking 2004 (Networking'04)*, LNCS, Springer-Verlag, Vol. 3042, pp. 725-742, Athens, Greece, May 2004 (acceptance rate 19,1%).

[CC02] L. Ciarletta and I. Chrisment. Outils pour l'expérimentation des technologies de l'informatique ambiante. *16ème Congrès DNAC : De nouvelles architectures pour les communications (DNAC'02).* Paris, France, Décembre 2002.

Présentation invitée

[CCA02] L. Ciarletta, A. Aït Ali, and I. Chrisment. VPSS : Architecture de Sécurité pour l'Informatique Ambiante *Sécurité et Architecture Réseaux 2002 (SAR'02).* Marrakech, Maroc, Juillet 2002.

Rapports de recherche, drafts

[BBC04] A. Bouabdallah, and M.S Bouassida, and Y. Challal, and I. Chrisment. État de l'art de l'authentification dans les communications de groupe. Rapport de contrat SAFECAST, LORIA n^o A04-R-250, Octobre 2004, 24 p.

[BLC04] M.S Bouassida, A.Lahmadi, I. Chrisment and O. Festor. Diffusion multicast sécurisée dans un environnement Ad-Hoc (1 vers n séquentiel). Rapport de recherche INRIA n° 5310, LORIA n^o A04-R-045, Septembre 2004, 47 p.

[BCG04] M.S Bouassida, I. Chrisment, V. Guyot, V. Legrand, D. Raffo, and S. Ubeda, Sécurité et Réseaux Ad Hoc. Rapport de contrat SAFARI, LORIA n^o A04-R-045, Avril 2004, 35 p.

Travaux encadrés ou co-encadrés

[Ciar02] L. Ciarletta. *Contribution à l'évaluation des technologies de l'informatique ambiante* Thèse d'université, Université Henri Poincaré, Nancy, Novembre 2002.

[Ait02] A. Ait Ali. Sécurité pour l'informatique ambiante. Rapport Stage de DEA, LORIA n^o A02-R-463, 2002, 68p.

[Pey01] A. Peyronnel. Architecture AAA et mobilité sur IPv6 adaptées aux applications temps réel. Rapport Stage de DEA, Septembre 2001, 60p.

[Ciar98] L. Ciarletta. Vers une approche sécurisée des mécanismes d'autoconfiguration dans IPv6. Rapport Stage de DEA, LORIA n° 98-R-332, 1998, 60p.

5

Synthèse et Perspectives

5.1 Résumé des contributions

Dans ce mémoire, nous avons présenté l'évolution de nos recherches au cours de la dernière décennie. Trois grandes périodes ont marqué ce parcours, centré sur la dynamique dans les réseaux et services offerts dans l'Internet :

1. les réseaux actifs ;
2. la sécurité dans les communications de groupe ;
3. la sécurité dans les réseaux spontanés.

Notre travail de classification des différentes approches de réseaux actifs et/ou programmables a montré l'intérêt de tels paradigmes et a mis en évidence les limites d'une vision statique du réseau dans un environnement où les offres de nouveaux services évoluent très rapidement. Les protocoles transport et réseaux doivent pouvoir s'adapter aux besoins des applications. Cependant le déploiement de codes dans les infrastructures réseaux, conséquence de la mise en place des réseaux actifs, soulève des problèmes importants de supervision. Dans le cadre de notre recherche, nous avons proposé une plate-forme de supervision d'environnements d'exécution et de protocoles actifs. Cette plate-forme est également une application active qui permet de découvrir la topologie du réseau supervisé et de suivre le déploiement du code. De plus, qui dit réseaux actifs dit chargement dynamique de nouveaux protocoles, d'où un besoin de tester ces protocoles. Nous avons montré que la technologie active est une très bonne approche pour mettre en place des tests de conformité et d'interopérabilité dans une architecture décentralisée où les protocoles ont des exigences fortes de dynamique.

La communication de groupe dans l'Internet représente le moyen le plus efficace pour envoyer des données vers plusieurs récepteurs en réduisant la bande passante utilisée. Elle est très adaptée aux applications de diffusion multimédia utilisées à travers l'Internet. La problématique de la sécurité constitue un défi important à résoudre pour que le déploiement du multicast soit effectif. La communication de groupe est, cependant, plus complexe à sécuriser que la communication point à point notamment pour les groupes à forte dynamicité où le retrait et ajout d'un membre est possible à tout moment ; ce qui implique des renouvellements de clés fréquents. Nos travaux

ont abouti à la définition de protocoles de distribution de clé adaptés aux environnements dynamiques comme la communication de groupe. Deux contributions majeures sont issues de ces travaux :

- `Baal` qui est à la fois une architecture et un protocole de distribution de clés de groupe dans le modèle générique ASM (Any Source Multicast). Une extension du protocole IGMP Proxying a été proposée dans `Baal` afin d'optimiser cette architecture.
- `S-SSM` qui reprend les bases de `Baal` mais dans le contexte des applications spécifiques à une source. Un schéma distribué de chiffrement à clés publiques a été également intégré dans le modèle `S-SSM` pour rendre plus efficace le renouvellement de la clé de groupe.

Les réseaux spontanés font maintenant partie de notre quotidien ; l'utilisation d'une informatique mobile et nomade est devenue incontournable. Les réseaux ad hoc sont par nature des réseaux spontanés puisqu'ils permettent de faire communiquer ensemble des entités (qui peuvent être mobiles) et cela sans aucune infrastructure préalable. La sécurité des communications de groupe présente dans le cadre des réseaux ad hoc de nouveaux challenges liés à la nature dynamique et flexible de ce type de réseaux, par rapport aux réseaux filaires. Tout d'abord, l'utilisation de liens sans fil rend un réseau ad hoc facilement exposé à des attaques. Ensuite, la taille et le caractère dynamique propres aux groupes multicast peuvent être beaucoup plus importants dans les réseaux ad hoc où le nombre de membres et la fréquence d'adhésion au groupe sont plus difficilement contrôlables. La mobilité des nœuds présente également un problème pour la sécurité des communications et doit aussi être prise en compte. En effet, certains nœuds peuvent devenir temporairement inacessibles, sans pour autant quitter le groupe auquel ils appartiennent ; la mobilité implique un changement rapide et imprévisible de la topologie du réseau. L'absence d'infrastructure fixe dans les réseaux ad hoc élimine toute possibilité de pouvoir établir une référence centralisée au sein du groupe. Tout cela implique que les architectures classiques de sécurité des groupes, centralisées ou hiérarchiques, ne sont pas applicables dans de tels réseaux. Nos travaux ont principalement donné lieu à la proposition d'une architecture adaptant `Baal` à l'environnement ad hoc ainsi qu'à proposition d'une nouvelle approche de gestion de la clé du groupe dans un environnement ad hoc pour des applications de diffusion de flux multimédia (streaming).

5.2 Projet de recherche

Nous proposons de poursuivre nos travaux de recherche selon trois orientations qui sont détaillées dans cette section :

- les protocoles de distribution de clés ;
- le contrôle d'accès et l'authentification ;
- la sécurité et la gestion de réseaux/services.

La prise en compte de la sécurité dans les réseaux dynamiques représente la ligne directrice que nous nous fixons pour les prochaines années.

5.2.1 Distribution de clés de groupe

Nous allons approfondir notre recherche sur la distribution de clés pour les communications de groupe dans les réseaux ad hoc et étendre le domaine d'application à d'autres environnements dynamiques notamment les réseaux pair-à-pair dont les caratéristiques sont proches des réseaux ad hoc.

Travaux dans les domaines des réseaux ad hoc

La fonction d'évaluation que nous avons proposée pour constituer des sous groupes partageant la même clé ne prend en compte que la fréquence d'adhésion au groupe et le nombre de membres locaux. Nous voulons intégrer le facteur mobilité dans la constitution d'un cluster et évaluer son impact au niveau de la performance de notre protocole [BCF05b]. Les travaux de [HGPC99] montrent que le modèle de mobilité choisi influence fortement le comportement des protocoles de routage utilisés dans l'ad hoc. Les protocoles de distribution de clé doivent être également affectés. Même si le modèle de mobilité aléatoire est souvent retenu pour les simulations dans le cadre des réseaux ad hoc, le déplacement en groupe est une caractéristique de ces réseaux spontanés. Cela est d'autant plus vrai pour les communications de groupe. Ainsi il est facile de se représenter un groupe de pompiers qui se déplace vers une même direction selon les ordres donnés par leur supérieur hiérarchique. Dans ce cas, tous les membres du cluster peuvent être amenés à suivre leur contrôleur local.

La réduction de la consommation d'énergie dans un réseau ad hoc constitute également un véritable challenge en raison de la limitation des batteries. En proposant un algorithme de clusterisation pour la distribution des clés qui exploite l'avantage de la diffusion dans un environnement sans fil et optimise le temps de transmission, nous pouvons minimiser le nombre de relais contribuant à la diffusion des messages émis de la source du groupe vers le récepteur et par la même la consommation à la fois de l'énergie et de la bande passante [BCF05a].

La fiabilité de la distribution de la clé est également à prendre en compte. Il est en effet important que la clé de groupe distribuée arrive à tous les récepteurs concernés et ce de manière synchonisée. Sinon le membre du groupe possédant uniquement l'ancienne clé ne sera pas capable de déchiffrer le flux chiffré avec la nouvelle clé. Ce problème de fiabilité est d'autant plus critique dans les réseaux ad hoc et les réseaux sans fil en général plus sensibles aux pertes.

Un autre point concerne la validation du protocole de distribution de clé. Comment en effet valider que le protocole que nous proposons est sûr et ne décèle pas des failles de sécurité qui le rendent attaquable. Nous espérons développer une coopération avec l'équipe CASSIS au LORIA, experte dans la vérification des protocoles cryptographiques. Un travail en commun est en cours dans le cadre du projet RNRT SAFECAST[40].

Évolution vers les applications pair-à-pair

L'évolution des réseaux et notamment des infrastructures haut débit comme l'ADSL a favorisé le développement des applications fondées sur des réseaux pair-à-pair dans lesquels chaque usager peut être à la fois client et serveur (eMule, eDonkey, Kazaa,...). Non seulement, les réseaux pair-à-pair constituent une plafe-forme idéale pour le multicast applicatif [CDKR02, RHKS01], mais présentent une grande similitude avec les réseaux ad hoc.

Les deux modèles correspondent à des réseaux spontanés où les frontières entre les rôles sont de plus en plus floues mais où la notion d'appartenance à un groupe ou une communauté est très forte. Dans les réseaux ad hoc, toute machine est à la fois nœud terminal et nœud de transit permettant d'assurer la connectivité. Dans les réseaux pair-à-pair, toute machine est à la fois client et serveur permettant d'assurer la distribution de données et donc la connectivité au

[40] `http:www.telecom.gouv.fr/rnrt/rnrt/projets/safecast.htm`

niveau applicatif. Comme dans les réseaux ad hoc, les réseaux pair-à-pair mettent en relation des machines complétement hétérogènes tant en vitesse de traitement, quantité de stockage possible. Les deux modèles sont certes très proches mais peuvent également se compléter. Rien n'empèche de fournir un service pair-à-pair au dessus d'un réseau ad hoc ou en tout cas hybride offrant une interface avec le monde filaire.

Comme perspectives, nous souhaitons d'une part utiliser notre protocole de distribution de clés pour offrir des services pair-à-pair privés. D'autre part, de manière duale, nous voulons regarder si l'utilisation des réseaux pair-à-pair pour effectuer la distribution des clés peut être envisagée. Dans le cas de l'application de juke-box, développée au sein de Madynes, le module de synchronisation de la liste des fichiers audio s'appuie sur le modèle des réseaux pair-à-pair, avec l'infrastructure JXTA. L'un des points que nous souhaiterons poursuivre est d'appliquer le même modèle pour la distribution des clés.

5.2.2 Aspects autorisation et contrôle d'accès

Dans le cadre de nos travaux, nous avons regardé essentiellement le problème de la gestion et de la distribution de la clé de groupe. Afin d'étendre l'architecture de sécurité, il nous paraît nécessaire d'y intégrer les aspects contrôle d'accès, autorisation et authentification des membres.

L'architecture AAA (Authentication, Authorization, Accounting) [dLGG+00, VCF+00] est aujourd'hui déployée chez de nombreux fournisseurs de services ainsi que dans de nombreux réseaux d'entreprise. Quand un utilisateur veut accéder à un réseau, il émet une requête au client AAA qui l'envoie au serveur AAA. Le serveur AAA authentifie l'utilisateur en appliquant une politique de sécurité reliant l'utilisateur et le service demandé. Le serveur AAA ensuite répond à l'utilisateur (via le client AAA) pour l'avertir que le service a été mis en place. Les protocoles RADIUS [RWS00] ou DIAMETER [CLG+03] implantent cette architecture et permettent d'établir une configuration entre, par exemple, un NAS (serveur d'accès au réseau) et un serveur partagé AAA.

Le développement des réseaux sans fil a augmenté considérablement la complexité de ces protocoles avec la prise en compte de la mobilité (WiFi ou IP). Des propositions ont ainsi émergé qui définissent ce que doit supporter un serveur AAA pour aider à fournir des services de Mobilité IP [GHJP00]. La norme IEEE 802.1X[41] définit un contrôle d'accès basé sur les ports. Elle présente un cadre pour authentifier et autoriser des stations dans les réseaux locaux et notamment dans les réseaux 802.11. L'accès au réseau (obtention d'une adresse IP) est interdit tant qu'une authentification auprès d'un serveur n'a pas été obtenue. Le protocole 802.1X s'applique entre l'utilisateur et le client AAA qui en général est un point d'accès sans fil. Le protocole souvent utilisé entre la base et un serveur d'authentification est le protocole RADIUS.

Par contre, l'ensemble des solutions précédentes font l'hypothèse d'un serveur dédié. Les réseaux ad hoc ont la particularité d'être dynamiques dans l'espace et le temps. Souvent éphémères, ils ne nécessitent aucune infrastructure fixe. De cette absence d'infrastructure, il découle que l'architecture AAA classique ne peut pas s'appliquer. Nous voulons adapter le modèle actuel client/serveur de l'architecture AAA dans le cadre des réseaux ad hoc. Notre principal objectif est de proposer des modèles d'usage de cette architecture dans les contextes d'authentification et d'autorisation des terminaux, des applications et des usagers pour des communications de groupe.

[41] `http://standards.ieee.org/getieee802/download/802.1X-2001.pdf`

5.2.3 Sécurité et gestion de réseaux/services

La gestion de la sécurité dans les réseaux dynamiques constitue à elle seule un objectif de recherche. Cependant, la sécurité ne doit pas être vue comme un service isolé mais comme faisant partie d'une politique d'administration plus générale. Dans cette optique, nous envisageons la relation entre la sécurité et la gestion des réseaux et des services sous trois angles qui sont présentés dans les paragraphes ci-après.

Supervision de la gestion des clés

Comme nous l'avons vu précédemment la distribution des clés est un mécanisme important de configuration des membres dans un groupe. De ce fait, elle doit pouvoir être supervisée et contrôlée par des entités administratives. Un modèle d'information doit être défini permettant de déterminer les données à gérer. Il sera alors possible pour un gestionnaire de savoir, à un instant donné, qui sont les contrôleurs (global/local) et d'arriver à découvrir la topologie de l'arbre de distribution des clés. Des statistiques pourront permettre d'établir si tous les membres ont bien reçu leur nouvelle clé et quel est le taux de pertes effectif.

Sécurité du plan de gestion

Les informations de supervision liées à la gestion des clés sont très sensibles et les protocoles de gestion pourraient devenir des portes d'accès aux données faciles à franchir. La sécurisation du plan de gestion est donc un élément vital pour assurer la sécurité du système dans son ensemble. Actuellement, Vincent Cridlig doctorant dans le projet MADYNES, travaille sur la sécurité du plan de gestion avec comme objectif de proposer un modèle générique de sécurité dans un contexte d'interactions de gestion multi-parties. Un exemple de tels contextes peut être une configuration distribuée d'un pare-feu ou un environnement *multi-homé*. Pour permettre que seules les entités autorisées soient capables d'accéder à la gestion des données et par conséquent avoir les droits pour accéder aux données, des clés doivent être distribuées aux agents et aux gestionnaires. En proposant d'associer les mêmes clés pour les entités partageant les mêmes rôles, les auteurs dans [CSF05] considèrent un rôle comme l'équivalent d'un groupe ; ils utilisent un algorithme basé sur les arbres de clés [WHA99, WGL98] pour distribuer les clés aux agents et aux gestionnaires. Mais cela implique une solution centralisée pour le serveur de clés. Nous voulons montrer que les protocoles de distribution de clé que nous avons définis dans le cadre de la communication de groupe peuvent être appliqués au cas particulier des applications de gestion de réseaux.

Configuration des règles de sécurité

Nous pouvons appeler cette proposition ALS (Application Level Security) en référence à ALF. L'objectif est d'arriver à permettre à une application de définir ses propres besoins en sécurité via un langage de spécification, comme nous l'avions réalisé avec le langage ESTEREL pour l'expression des besoins de communication.

À partir des besoins exprimés au niveau sécurité par l'application et de l'environnement dans lequel évolue cette application (WiFi, présence de pare-feux), nous souhaitons en dériver des politiques de configuration transparentes et spécifiques. Cela permettra d'éviter des empilements de couches de sécurité qui ne sont pas toujours nécessaires : ainsi il est aujourd'hui possible

d'effectuer une session https vers un serveur avec lequel un VPN sécurisé a été mis en place ; et ce depuis un portable utilisant un chiffrement WEP dynamique avec sa station de base WiFi.

Il est indéniable que des mécanismes de vérification de cohérence devront être mis en place afin d'être certain que certaines règles ne sont pas incompatibles.

L'approche ALS est certes un challenge ambitieux. Mais elle permet d'avoir une vision plus générale de l'architecture de sécurité et non plus couche par couche comme cela est le cas actuellement.

Bibliographie

[CDKR02] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron. SCRIBE : A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in communications (JSAC)*, 20(8) :1489–1499, october 2002.

[CLG+03] P. Calhoun, J. Loughney, E. Guttnan, Z. Zorn, and J. Arkko. Diameter Base Protocol. RFC 3588, September 2003.

[CSF05] V. Cridlig, R. State, and O. Festor. An Integrated Security framework for XML based Management. In *The 9th IFIP/IEEE International Symposium on Integrated Network Management - IM'2005, Nice, France*, May 2005.

[dLGG+00] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence. Generic AAA Architecture. RFC 2903, August 2000.

[GHJP00] S. Glass, T. Hiller, S. Jacobs, and C. Perkins. Mobile IP Authentication, Authorization, and Accounting Requirements. RFC 2977, October 2000.

[HGPC99] X. Hong, M. Gerla, G. Pei, and C. Chiang. A Group Mobility Model for Ad Hoc Wireless Networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and Simulation of Wireless and Mobile systems, MSWiM'99*, pages 53–60, Seattle, Washington, USA, 1999.

[RHKS01] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Application-Level Multicast Using Content-Addressable Networks. In *Networked Group Communication, Third International COST264 Workshop, NGC 2001*, volume 2233 of *Lecture Notes in Computer Science*, pages 30–43. Springer, November 2001.

[RWS00] C. Rigney, A. Willens, S.and Rubens, and W. Simpson. Remote Authentication Dial in User Services (RADIUS). RFC 2865, June 2000.

[VCF+00] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. AAA Authorization Framework. RFC 2904, August 2000.

[WGL98] C. Wong, M. Gouda, and S. Lam. Secure Group Communications using Key Graphs. In *ACM SIGCOMM on Communication Architecture and Protocols (SIGCOMM'98)*, September 1998.

[WHA99] D. Wallner, E. Harder, and R Agee. Key Management for Multicast : Issues and Architecture. RFC 2627, IETF, June 1999.

Publications

Conférences

[BCF05b] M.S. Bouassida, I. Chrisment, and O. Festor. Prise en compte de la mobilité dans le protocole de gestion de clé de groupe BALADE. *Sécurité et Architecture Réseaux 2005 (SAR'05)*, Batz sur Mer, France, Juin 2005, 12p.

[BCF05a] M.S. Bouassida, I. Chrisment, and O. Festor. Efficient Clustering for Multicast Key Distribution in MANETs. *Fourth IFIP-TC6 Networking (Networking'05)*. Volume 3462 of Lectures Notes in Computer Science, Springer, pages 138-153, Waterloo, Canada, May 2005 (acceptance rate 24,7%).

Acronymes

AAA	Authentication, Authorization and Accounting
ADU	Application Data Unit
AKMP	Adaptive Key Management Protocol
ALF	Application Level Framing
ANAIS	Active Network Architecture for Internet Service Provider
ANCORS	Adapatble Network COntrol and Reporting System
ANEP	Active Network Encapsulation Protocol
ANTS	Active Node Transfer System
ASM	Any Source Multicast
ASP	Active Signaling Protocol
CBID	Crypto-Based IDentifier
CBT	Core Based Tree
CG	Contrôleur de Groupe
CL	Contrôleur Local
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
DCCP	schéma Distribué de Chiffrement à Clés Publiques
DEP	Dual Encryption Protocol
DK	Downstream Key
DS	Directory Server
DVMRP	Distance Vector Multicast Routing Protocol
ESP	Encapsulating Security Payload
ETSI	European Telecommunications Standards Institute
GDH	Group Diffie Hellman
GKMP	Group Key Management Protocol
GSC	Group Security Controller
GSEC	Group SECurity Research Group
GSI	Group Security Intermediary
GPS	Global Positioning System
HIPPARCH	High Performance Protocol ARCHchitecture
ILP	Integrated Layer Processing
IP	Internet Protocol
IPv6	Internet Protocol version 6
IRTF	Internet Research Task Force
IETF	Internet Engineering Research Task Force
IGMP	Internet Group Management Protocol
IPsec	IP Security

IRC	Internet Relay Chat
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
KEK	Key Encryption Key
LDAP	Lightweight Directory Access Protocol
LKH	Logical Key Hierarchy
MLD	Multicast Listener Discovery
MSEC	Multicast SECurity Working Group
MTU	Maximum Transmission Unit
OFT	One-way Function Tree
OSI	Open Systems Interconnection
PIM-DM	Protocol Independent Multicast-Dense Mode
PIM-SM	Protocol Independent Multicast-Sparse Mode
PKI	Public Key Infrastructure
PMR	Professional Mobile Radiocommunication
RFC	Request For Comments
RPC	Remote Procedure Call
RNRT	Réseau National de Recherche en Télécommunications
RTP	Real Time Protocol
SAKMP	Scalable Adaptive Key Management Protocol
SDL	Specification and Description Language
SGM	Sub Group Manager
SKDC	Single Key Distributor Center
SMuG	Secure Multicast Research Group
SNMP	Somple Network Management Protocol
SSL	Secure Socket Layer
SSM	Source Specific Multicast
TCP	Transmission Control Protocol
TEK	Traffic Encryption Key
TLS	Transport Layer Security
TTCN	Tree and Tabular Combined Notation
	Testing and Test Control Notation
UDP	User Datagram Protocol
UK	Upstream Key
UMANTS	User-based Management of the Active Network Transfer System
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy

Annexes

Les publications fournies en annexe complètent les contributions présentées dans les différents chapitres du manuscrit. Ces publications sont dans l'ordre de correspondance avec les chapitres :

[CKD98] I. Chrisment, D. Kaplan, and C. Diot, An ALF Communication Architecture : Design and Automated Implementation. *IEEE Journal of Selected Area in Communications.* 16(3) :332-344, April 1998.

[ACC+00] S. D'Alu, G. Chellius, I. Chrisment, O. Festor, and E. Fleury. Intégration du support IPv6 dans l'environnement de supervision de réseaux actifs ANAIS *Conférence Francophone sur l'Ingéniérie des Protocoles (CFIP'00).* Toulouse, France, Octobre 2000, 16p. Hermes.

[CCS01b] G. Chaddoud, I. Chrisment, A. Schaff. Dynamic Group Communication Security. *The 6th IEEE Symposium on Computers and Communications (ISCC'01).* Hammamet, Tunisia, July 2001, 8p.

[CVCS04] G, Chaddoud, V. Varadharajan, I. Chrisment and A. Schaff. Gestion efficace de la sécurité des communications de groupe pour le Service SSM *Journal TSI n° spécial Réseaux et Protocoles.* 23(9) :1107-1135, 2004.

[BCF04b] M.S. Bouassida, I. Chrisment, and O. Festor. An Enhanced Hybrid Key Management Protocol for Secure Multicast in Ad-hoc Networks *Third IFIP-TC6 Networking 2004 (Networking'04).* LNCS, Springer-Verlag, Vol. 3042, pp. 725-742. Athens, Greece, May 2004 (acceptance rate 19,1%).

Résumé

De nouveaux modèles d'organisation de réseaux et de routage ainsi que de nouveaux services apparaissent face à une dynamique de plus en plus croissante au niveau de l'Internet : services multicast, services pair à pair, réseaux ad hoc et réseaux de capteurs. Ceci s'explique en partie par la convergence du monde fixe et du monde mobile, du monde des télécommunications et du monde IP. L'ensemble des contributions présentées dans ce mémoire met en évidence comment nous avons pris en compte l'aspect dynamique offert par les réseaux et les services rencontrés actuellement dans l'Internet. Nous nous sommes orientés vers la gestion de ces réseaux dynamiques avec une focalisation sur un axe crucial pour leur développement, celui de la sécurité. Dans une première partie, nous étudions comment nous pouvons adapter les protocoles aux besoins des applications tant au niveau transport que réseau. Nous montrons comment nos travaux relatifs à ALF (Application Layer Framing) nous ont conduits à nous intéresser au paradigme des réseaux actifs et plus particulièrement à leur gestion. La deuxième partie est concentrée autour d'un service de gestion spécifique ; celui de la sécurité dans le contexte de la communication de groupe. L'environnement multicast représente un excellent domaine d'application grâce à son aspect dynamique : tout membre peut quitter ou joindre le groupe à tout moment. Ce qui implique des renouvellements de clés fréquents et pose des problèmes de passage à l'échelle. Nos travaux ont abouti à la définition de protocoles de distribution de clés adaptés à la communication de groupe. Avec le déploiement des réseaux sans fil, le besoin de créer et d'interconnecter des réseaux autonomes et spontanés, appelés aussi réseaux ad hoc, va en augmentant. Le support du multicast dans ce type de réseau est important notamment pour des applications militaires ou des opérations de sécurité civile qui requièrent des communications de groupe pour l'échange d'informations confidentielles. Ces applications sont très sensibles et demandent un niveau de sécurité relativement élevé. Dans ce contexte, nous présentons comment nous avons adapté nos protocoles de distribution de clés de groupe à l'environnement ad hoc. Nous proposons aussi une nouvelle approche de gestion de clés dans les réseaux MANETs (Mobile Ad hoc NETworks) pour des applications spécifiques de diffusion de flux multimédia de 1 vers n séquentiel.

Mots-clés: réseaux actifs, IPv6, gestion de réseaux, sécurité, multicast, ad hoc.

Abstract

New organization models for networks and routing and as well as new services appear due to increasing dynamics within Internet : multicast services, peer-to-peer services, ad hoc and sensor networks. This is partly explained by the convergence of fix and mobile networks, of telecommunications and Internet worlds. The set of contributions presented in the manuscript points out how we have taken into account the dynamic aspect provided by networks and services that we are currently used within Internet. We have worked on management of dynamics networks and we are focused on an important axis of their development, namely the security functional area. In the first part, we study how to adapt protocols to the requirements of the applications at the transport and network level. We show how our work related to ALF (Application Layer Framing) has led us to investigate the active networks paradigm and, more specifically, the management of these active networks. The second part is dedicated to a specific management service, i.e. the security service in the context of group communication. The multicast environment is an excellent application domain, which offers high of dynamics : every member may leave or join its group at every time. This involves frequent rekeying and leads to extensibility problems. Our research work has resulted in defining new key distribution protocols suitable for group communication. Aside with the deployment of wireless networks, the need for creating and interconnecting autonomous and spontaneous networks, so-called ad hoc networks, is increasing. Integrating multicast in such networks is important, in particular for military applications or public security operations where group communications are formed in order to exchange confidential data information. These applications are very sensitive and require a high level of security. In this context, we present how we have adapted our group key management protocols to the ad hoc environment. We also propose a new key distribution approach in MANETs (Mobile Ad hoc NETworks) dedicated to secure flow multicast communications according to the sequential multi-source models.

Keywords: active networks, IPv6, networks management, security, multicast, ad hoc.