Baal : Sécurisation des communications de groupes dynamiques

Ghassan Chaddoud, Isabelle Chrisment, André Schaff

Baal : sécurisation des communications de groupes dynamiques

Ghassan Chaddoud* **– Isabelle Chrisment**** **– André Schaff***

LORIA
**Université de nancy 1*
***Université de Nancy 2*
Campus Scientifique BP 239
54506 Vandœuvre-lès-Nancy
{chaddoud,ichris,schaff}@loria.fr

RÉSUMÉ. *Avec l'émergence de nouvelles applications coopératives multimédia, la communication de groupes apparaît comme le moyen le plus efficace d'envoyer des données à un groupe spécifique composé de plusieurs membres. De plus, l'intérêt croissant dans la communication réseau à travers l'utilisation de l'Internet a rendu nécessaire des services comme l'authentification, l'intégrité et la confidentialité pour transporter des données de manière sûre. Dans ce papier, nous présentons le protocole* `Baal`[1] *comme nouvelle solution au problème de gestion des clés dans une communication de groupe dynamique. Nous montrons comment* `Baal` *résoud le problème de révocation et le problème d'authentification individuelle. Nous expliquons les problèmes d'extensiblité es protocoles de gestion des clés de groupe. Puis nous détaillons les algorithmes de* `Baal`*. Les premiers résultats montrent un gain au niveau de la taille de messages de distribution et de renouvellement des clés.*

ABSTRACT. *With the emergence of new cooperative applications, group communication appears to be the most efficient way to send data to a specific group composed of several members. Moreover, the increasing interest in network communication through the use of the Internet requires services such as authentication, integrity and confidentiality to transport data securely. In this paper, we present the protocol* `Baal` *as a new solution to the problem of group key management. We then show how* `Baal` *resolves a user's revocation problems and a sender's authentication problems. We introduce the scalability issues of group key management. Next, we detail* `Baal`*'s algorithms.The preliminary results show an optimization of the re-keying and distribution message size.*

MOTS-CLÉS : *sécurité point-à-point et multipoint, protocole de sécurité, clé de groupe, clé cryptographique*

KEYWORDS: *security unicast and multicast, security protocol, group key, cryptographic key*

1. 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 la mort.

1. Introduction

L'évolution rapide des technologies vers des réseaux haut-débit, la vitesse de plus en plus rapide des processeurs ont favorisé le développement de nouvelles classes d'applications comme l'audio, la vidéo conférence et le tableau blanc partagé. Pour ces applications coopératives la communication de groupe est devenue un concept non seulement important mais nécessaire. La transmission multipoint apparaît comme le moyen le plus efficace pour envoyer des données vers plusieurs récepteurs en réduisant la bande passante utilisée.

De plus, l'utilisation croissante des réseaux comme l'Internet pour des communications privées ou commerciales accroît l'importance liée à la sensibilité des données et nécessite des services comme l'authentification, l'intégrité et la confidentialité pour transporter les informations de façon sûre.

Beaucoup de recherches ont été effectuées pour protéger la communication point-à-point et des standards ont émergés ([ATK 98, HAR 98, MAU 98],...). L'architecture de sécurité dans le monde IP, IPsec [ATK 98], définit les concepts de base d'une Association de Sécurité ou SA. Une SA point à point est un ensemble de paramètres (clés et algorithmes de cryptage, clés et algorithmes d'authentification, durée de vie de la SA,...) partagé entre deux entités communicantes, connu seulement de ces deux entités et les autorisant à avoir un canal virtuel de communication sécurisé. Cette SA se négocie au moyen d'algorithmes prédéfinis comme ISAKMP/IKE [MAU 98, HAR 98].

La communication de groupe est, cependant, plus complexe à sécuriser [MIT 97]:

– le multicast présente plus d'opportunité pour l'interception du trafic ;

– si une attaque se produit alors un grand nombre de systèmes peut être affecté ;

– l'identité et l'adresse du groupe sont connues à large échelle ce qui aide les intrus à diriger leurs attaques ;

– les attaquants peuvent remplacer des membres principaux (membres légitimes du groupe) par d'autres illégitimes.

De plus, l'association de sécurité pour un groupe ou GSA ne peut pas se négocier, c'est-à-dire que les membres du groupe ne peuvent pas interagir pour créer une GSA car le nombre de participants peut être important. Dans [ATK 98, HAR 99b] les auteurs proposent que seule une entité, par exemple, le gestionnaire de groupe ou le contrôleur de groupe, choisisse une GSA et la distribue ensuite aux membres du groupe par des tunnels sécurisés. Le contrôleur de groupe peut annoncer les paramètres de sécurité du groupe (algorithmes de cryptage , d'authentification,..) par des protocoles de session multicast (*e.g.* SAP [HAN 00], SDP [HAN 98],...).

Le paramètre le plus important dans une GSA est la clé utilisée pour chiffrer les données du groupe. Cette clé K_{grp}, appelée clé multicast ou clé de groupe, est utilisée pour assurer la confidentialité de la communication de groupe. Le contrôleur de groupe crée et distribue cette clé, d'une manière sûre, aux différents membres du groupe. Un message multicast envoyé par un membre du groupe et chiffré (nous parlons ici de la

cryptographie symétrique) avec K_{grp} peut être reçu et déchiffré par tous les membres possédant K_{grp}. De cette manière, en utilisant cette clé, le contrôle d'accès au groupe est assuré car seules les entités ayant K_{grp} peuvent lire le trafic du groupe.

Le partage de la clé de groupe par les membres du groupe peut implicitement assurer l'authentification du groupe (l'émetteur d'une donnée dans un groupe sécurisé doit appartenir au groupe). Mais ce n'est pas suffisant pour assurer l'authentification individuelle de l'émetteur car cette clé ne peut pas être utilisée pour différencier les différents émetteurs entre eux. Par conséquent, la clé de groupe peut être utilisée pour la confidentialité, le contrôle d'accès, l'authentification de groupe et l'intégrité.

Il existe plusieurs propositions pour gérer la clé de groupe et sécuriser les communications de groupe. Cependant, aucune n'est satisfaisante pour les grands groupes ou les groupes à forte dynamicité comme ceux pouvant exister sur l'Internet. Une solution idéale devrait satisfaire les points suivants [CHA 99] :

– temps minimal de configuration de groupe ;

– trafic aussi réduit que possible ;

– groupe dynamique, *i.e.* retrait et ajout d'un membre est possible à tout moment ;

– indépendance des protocoles de routage ;

– indépendance des protocoles de sécurité point-à-point (IKE [HAR 98], SSL [FRE 96]) ;

– confidentialité, intégrité et authentification des données ;

– décentralisation de la gestion du groupe.

Dans la section suivante, nous présentons les problèmes liés à la communication de groupe sécurisée et plus spécifiquement les problèmes de passage à l'échelle. Ensuite, nous décrivons les principaux travaux effectués dans ce domaine. Les Sections 4 et 5 donnent une description de notre protocole. La Section 6 analyse et effectue des comparaisons avec d'autres approches. Nous terminons la Section 7 par une conclusion.

2. Problématiques

En général, les protocoles multicast présentent deux problèmes qui limitent le passage à l'échelle ou l'extensibilité. Ces problèmes ont été résumés par [MIT 97] sous les termes **1 affects n** et **1 does not equal n**.

– **1 affects n:** se produit lorsque une action chez un membre du groupe affecte tous les autres membres. Par exemple, dans le protocole DVMRP [PUS 99], l'ajout d'un système émetteur provoque la construction d'un nouvel arbre basé sur la source et tous les routeurs doivent alors mettre à jour l'état d'information du groupe.

– **1 does not equal n:** apparaît quand un protocole ne peut pas traiter avec tous les membres d'un groupe ; il doit prendre en compte les capacités de chacun. MITTRA

cite les protocole de contrôle de flux quand il y a des récepteurs qui veulent augmenter le rythme de transmission et d'autres qui veulent le diminuer [MIT 97].

Les protocoles de gestion de clés multicast rencontrent le problème de type **1 affects 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} (la clé commune entre les membres du groupe et utilisée pour chiffrer la communication multicast du groupe) par une autre $K_{grp}{}'$ afin d'empêcher un nouvel abonné au groupe d'accéder à l'ancien trafic. La clé actuelle K_{grp} du groupe est utilisée pour distribuer la nouvelle clé $K_{grp}{}'$. Un seul message contenant $K_{grp}{}'$ chiffré avec K_{grp} est diffusé à l'ensemble du groupe. Les membres du groupe, qui reçoivent ce message, remplacent K_{grp} par $K_{grp}{}'$. 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 remplacer aussi la clé du groupe K_{grp} afin d'empêcher le membre supprimé d'accéder aux futures communications du groupe (dans [CAN 98] ceci est connu comme le problème de *révocation de l'utilisateur*). Le gestionnaire de clé 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 multicast chiffré avec l'ancienne clé. Le membre supprimé pourrait déchiffrer ce message et avoir la nouvelle clé. Par conséquent, le gestionnaire de clé est obligé d'utiliser des tunnels sécurisés de communication pour distribuer $K_{grp}{}'$ à chaque membre individuellement. Les deux types de problèmes d'extensibilté se rencontrent : le premier est **1 does not equal n** car le gestionnaire communique la clé à un membre comme s'il était indépendant du groupe. Le deuxième est **1 affects n** car la suppression d'**1** membre oblige les **n** membres à remplacer la clé K_{grp} par une autre.

Lors de la mise à jour de la clé du groupe, d'autres problèmes d'extensibilité se manifestent sous forme de trous de sécurité :

– les membres récepteurs, qui n'arrivent pas à recevoir la nouvelle clé, ne peuvent plus être capables de déchiffrer les communications du groupe. De plus, ils risquent de recevoir des communications multicast envoyées par des membres supprimés.

– les membres émetteurs, qui n'arrivent pas à recevoir la nouvelle clé, continuent á chiffrer les messages émis avec l'ancienne clé ; les autres membres ne sont plus capables de recevoir les messages du groupe. De plus, des membres supprimés peuvent être capables de déchiffrer les messages ; ce qui compromet la sécurité du groupe.

Ces problèmes de synchronisation entre les membres peuvent être résolus en utilisant des protocoles de multicast fiable comme SRM [FLO 95].

Il y a d'autres problèmes d'extensibilité relatifs à l'authentification individuelle de l'émetteur. Dans un groupe de communication où les membres utilisent la cryptographie asymétrique pour assurer l'authentification, la clé publique de l'émetteur peut ne

pas être disponible immédiatement à tous les membres du groupe. Ils peuvent utiliser par exemple SSL [FRE 96] pour obtenir les clés publiques mais cela peut demander un certain délai. De plus des manques d'information au sujet des centres de certification à clés publiques rendent cette méthode peu efficace pour des groupes très dynamiques, où les abonnements et désabonnements sont très nombreux.

3. Les différentes approches de la sécurisation d'un groupe

Les travaux réalisés pour la construction et la distribution de la clé de groupe peuvent être classifiés en deux catégories : théoriques et pragmatiques.

La première catégorie regroupe les approches basées sur la théorie de l'information [BLU 92], les approches hybrides [FIA 93] et les échanges de clés Diffie-Hellmann [STE 96, BUR 97]. Pour ces approches l'espace de stockage, le temps de calcul ainsi que le nombre de messages échangés augmentent linéairement avec la taille du groupe (nombre de participants) pour l'ajout ou la suppression d'un membre. Ces approches sont difficiles voire impossibles à implanter.

Quant à la seconde catégorie, elle regroupe la méthode SKDC (Single Key Distribution Center) [HAR 97a, HAR 97b, WAL 98] et les approches hiérarchiques [WAL 98, WON 98, MCG 98, BAL 99]. La méthode SKDC croît linéairement avec la taille du groupe et souffre d'une gestion centralisée du groupe. Elle ne résoud pas les deux problèmes d'extensiblité présentés dans la section précédente et reste seulement bien adaptée aux petits groupes de discussion. Dans les approches hiérarchiques qui sont logarithmiques par rapport à la taille du groupe, on distingue les approches nécessitant des routeurs *fidèles* comme SMKD (Scalable Multicast Key Distribution) [BAL 96] et les approches sans noeuds intermédiaires comme, par exemple, LKH (Logical Key Hierarchy) [WAL 98, WON 98] et OFT (One-way function Tree) [MCG 98, BAL 99]. En comparant ces trois approches SKDC, LKH et OFT, nous pouvons noter que, durant la phase d'initialisation du groupe, l'approche SKDC est plus efficace que les approches hiérarchiques. De plus, elle exige un espace de stockage plus petit que les autres. En revanche, les approches hiérarchiques sont plus adaptées pour les groupes dynamiques car elles distribuent l'effort du calcul de renouvellement de la clé du groupe sur différents participants du groupe.

Les approches hiérarchiques essaient de résoudre le problème de type **1 does not equal n** en changeant un problème en $O(n)$ en un problème en $O(log(n))$, avec n la taille du groupe. Nous pouvons observer qu'il n'y a aucune des approches pour la construction et la distribution des clés de groupe qui résolvent le problème **1 affects n**. En appliquant une gestion décentralisée [MIT 97, HAR 99a, HAR 99b] par opposition à une gestion centralisée [HAR 97a, WAL 98, WON 98], il devient possible de résoudre le problème **1 affects n** en divisant le groupe en sous-groupes. Chaque sous-groupe, géré par un contrôleur local, a sa propre clé. Les sous-groupes sont liés par des agents intermédiaires pour construire un groupe virtuel. Le rôle des agents intermédiaires est de traduire/chiffrer des données diffusées par un membre dans son

sous-groupe à tous les membres dans le groupe virtuel. En conséquence, cette gestion centralisée prend mieux en compte la dynamicité des groupes. Mais elle est moins effective pour diffuser les données de groupe qui subissent des opérations chiffrement/déchiffrement par les agents intermédiaires alors que la gestion centralisée utilise une seule clé partagée entre les membres du groupe. Il faudrait donc mettre en place un protocole qui assure une gestion décentralisée avec une seule clé ; c'est ce que nous allons proposer dans le protocole décrit dans les deux Sections 4 et 5.

En ce qui concerne l'authentification individuelle, il y a deux solutions connues. La première est basée sur la signature à clé publique. L'inconvénient de cette solution est que la clé publique de l'émetteur du trafic multicast n'est peut-être pas disponible à tous les membres récepteurs. De plus, elle implique un surplus de traitement notamment dans le travail nécessaire pour générer les signatures. La seconde présentée par Canetti dans [CAN 98] repose sur les mécanismes de clés partagées, appelées MACs (Message Authentication Codes) [SCH 97]. En effet, chaque partie *U* connaît un sous-ensemble aléatoire R_u d'un ensemble global de clés. Quand un émetteur *U* envoie un message, il l'authentifie avec toutes les clés de R_u. Chaque récepteur *V* vérifie les MACs réalisés sur les clés appartenant à l'intersection de R_u et R_v, *i.e.* $R_u \cap R_v$.

L'inconvénient de ce schéma est qu'il devient trop compliqué, même impossible à appliquer, sur les grands groupes existant sur Internet. Sa difficulté est induite de la gestion d'un grand nombre de clés distribuées d'une manière aléatoire aux participants. Par contre, elle reste une bonne idée pour les petits groupes.

4. Baal : Description générale du protocole

En considérant les objectifs cités précédemment, nous avons spécifié un nouveau protocole de distribution des clés de groupe qui présente une solution extensible pour le problème de révocation d'un membre d'un groupe et le problème de l'authentification individuelle. Nous améliorons l'extensibilité par la décentralisation de la gestion de la clé de groupe, l'indépendance du protocole de routage et l'indépendance de la gestion de la sécurité point-à-point (IKE [HAR 98], SSL [FRE 96],...).

Nous avons observé que dans les conférences spécialisées, il peut y avoir plusieurs participants venant du même laboratoire de recherche qui travaillent sur le même domaine de recherche. Le nombre de participants du même laboratoire à une même conférence peut donc varier de un (minimum) à plusieurs dizaines et être plus important dans des conférences virtuelles sur l'Internet. Nous avons ainsi défini un coefficient de participation, α, comme étant le nombre de systèmes (participants) qui se trouvent dans le même LAN (Local Area Network). Nous supposons qu'un LAN est relié à l'Internet par un routeur local supportant IGMPv3 [CAI 00].

Dans notre protocole, nous considérons trois types d'entités : le contrôleur de groupe, le contrôleur local et le membre du groupe.

– Le contrôleur du groupe ou GC peut être un organisateur de conférence ou un *chairman* ; il a les droits pour créer un groupe de communication. C'est l'entité qui détient la liste, appelée ACL (Access Control List), des futurs participants, crée la clé

de groupe et la distribue aux membres du groupe par l'intermédiaire des contrôleurs locaux. Nous supposons que le contrôleur du groupe crée la liste ACL à partir d'autres moyens (*e.g.* courrier, fax,...) ;

– Un contrôleur local est délégué par le contrôleur de groupe. Il reçoit la clé de groupe et la distribue aux membres du groupe dans son réseau local. Un contrôleur local peut être un routeur local supportant IGMPv3 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. Nous supposons que l'ensemble des contrôleurs (locaux et de groupe) sont de confiance et qu'ils ont les moyens de générer de manière sûre des clés cryptographiques.

– Un membre du groupe est rattaché à un contrôleur local par une liaison LAN.

La clé de groupe est distribuée en deux étapes :

1. La première consiste à inviter et à connaître les entités, à partir de la liste ACL, qui veulent participater au groupe ainsi que les contrôleurs locaux.

2. La seconde est de distribuer, à proprement dit, la clé de groupe aux participants ayant répondu à l'invitation de la première étape.

Les messages échangés durant la première étape ne sont pas chiffrés mais seulement sont authentifiés car ils ne contiennent pas des éléments cruciaux de sécurité. Les contrôleurs signent simplement les messages et ajoutent leur clé publique à la fin du message pour permettre aux entités réceptrices de vérifier leurs signatures. Par contre, les messages de la seconde étape sont chiffrés, avec les clés publiques des récepteurs, car ils contiennent des paramètres importants de sécurite comme la clé du groupe, l'identité de groupe, l'identité du contrôleur... .

Nous assurons l'authentification individuelle par l'utilisation de la signature à clé publique. À la fin de la première étape le contrôleur du groupe distribue aux contrôleurs locaux leur clé publique. Un émetteur dans le groupe signe un message et le diffuse aux membres du groupe. Son contrôleur local signe le message et le réémet. Dans chaque réseau local, un gestionnaire local recevant ce message peut vérifier la signature de l'émetteur car il a confiance dans l'autre contrôleur local et ce dernier authentifie l'émetteur.

À n'importe quel moment dans la vie du groupe, un contrôleur local peut jouer le rôle d'un contrôleur de groupe ; il peut alors créer et distribuer une nouvelle clé, accepter ou refuser un nouveau membre dans le groupe, et signaler tout changement dans le groupe aux autres contrôleurs.

Nous supposons que, dans un réseau local, chaque entité connaît la clé publique des autres entités et que ces entités font confiance au contrôleur local.

Nous allons maintenant décrire plus en détail les algorithmes utilisés par notre protocole.

5. Baal : Algorithmes

5.1. *Initialisation du groupe*

Cette phase se propose :

– de distribuer de manière sûre une clé unique, la clé de groupe K_{grp}, à tous les éléments de la liste ACL, qui acceptent de participer au groupe ;

– de déléguer des contrôleurs locaux pour garantir l'accès au groupe de manière locale (réseau local, domaine d'administration,...) ;

– de coopérer avec les autres contrôleurs pour gérer la clé de groupe et contrôler l'accès au groupe ;

– et aussi de distribuer à un contrôleur local la liste des autres contrôleurs avec leur clé publique ;

Le contrôleur de groupe commence le processus d'initialisation du groupe en créant la clé de groupe K_{grp}. Nous supposons que le contrôleur de groupe a les moyens de générer la clé cryptographique, sinon il devrait coopérer avec d'autres membres du groupe pour générer la clé du groupe. Ensuite, le contrôleur de groupe communique la clé de groupe K_{grp} aux membres du groupe via les contrôleurs locaux. Ceux-ci communiquent à leur tour leur clé publique au contrôleur de groupe.

L'initialisation de groupe se découpe en deux phases : la phase d'invitation qui est réservée pour inviter les membres de l'ACL et les contrôleurs locaux à participer à la communication de groupe et la phase de distribution de K_{grp} qui a comme objectif de distribuer de manière sûre la clé de groupe aux membres du groupe.

5.1.1. *Phase d'invitation*

Afin de protéger les messages en clair dans cette étape, l'émetteur d'un message inclut dans le message un *token*. Un *token* forme une partie essentielle du processus d'authentification des messages échangés. Il aide un récepteur à vérifier l'originalité du message et l'identité de l'émetteur. Pour définir un *token* nous reprenons la forme définie dans [BAL 96] où un *token* contient l'identité unique du récepteur, une estampille et un nombre pseudo-aléatoire.

Cette phase inclut deux types de messages (Fig. 1): KC_mg1 et KC_mg2.

KC_mg1 est envoyé par le GC à un élément de l'ACL. Ce message, contenant la clé publique du GC et son *token*, est signé par le GC. Le routeur local du destinataire du message authentifie l'émetteur. Si l'authentification réussit, il stocke la clé publique de GC et ré-envoie le message au destinataire qui, à son tour, authentifie l'émetteur. Si l'authentification réussit et si le destinataire accepte de participer alors ce dernier acquitte le message par un `IGMP-report` contenant son *token* signé. Nous supposons que IGMPv3 [CAI 00] définit un type de message qui indique la présence d'un *token* signé.

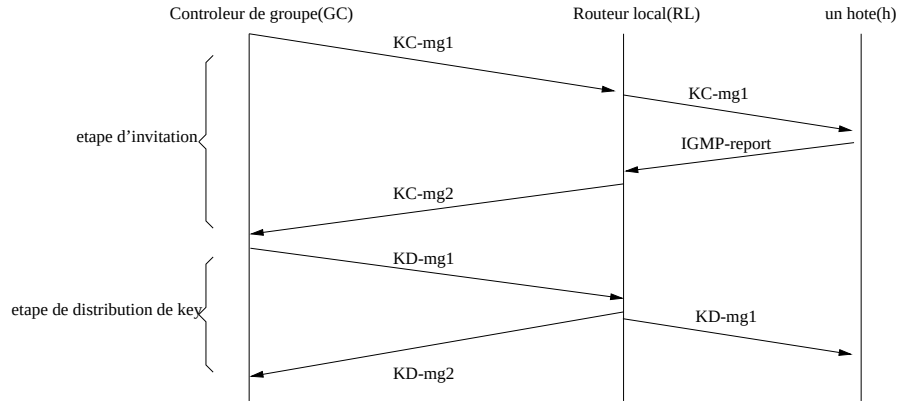

Figure 1. *Initialisation du groupe*

En recevant le `IGMP-report`, le routeur local vérifie la signature. Si la vérification réussit, il envoie au GC un message KC_mg2 signé contenant le *token* de l'entité qui a accepté la participation, le *token* et la clé publique du routeur local qui se présente au GC comme contrôleur local dans le domaine, le réseau local, où l'entité existe. Le GC authentifie l'émetteur. Si l'authentification réussit, il accepte l'émetteur comme contrôleur délégué et stocke sa clé publique et l'ajoute à la liste des contrôleurs locaux ou LCL.

Remarque : Si un contrôleur local reçoit plusieurs messages (un nombre de messages égal au nombre de participants dans le réseau local du contrôleur c'est-à-dire égal à α, le cœfficient de participation) de type KC_mg1, cela implique qu'il y a α participants au groupe qui sont dans le même réseau local. Dans ce cas, le contrôleur local envoie un seul message KC_mg2 contenant le *tokens* d'un membre et la clé publique et le *token* du contrôleur.

À la fin de cette phase, le GC dispose d'une liste de contrôleurs locaux et de leur clé publique, de l'adresse, de l'identité et la clé du groupe.

5.1.2. *Phase de distribution de la clé de groupe*

Dans cette phase, le GC distribue K_{grp}, l'identité du groupe et son identité à tous les autres contrôleurs locaux dans le message KD_mg1 (Fig. 1). Ces messages doivent être chiffrés car ils contiennent des clés cryptographiques et l'identité des entités. Le contrôleur émetteur d'un message utilise la technique de la cryptographie asymétrique à clé publique pour chiffrer ces messages.

En recevant le message KD_mg1, le contrôleur local le déchiffre et l'acquitte par un message KD_mg2 chiffré avec la clé publique de GC ; KD_mg2 contient seulement l'identité du contrôleur et l'identité du groupe. Finalement, la clé de groupe est retransmise aux entités participant au groupe.

Notons que KD_mg1 subit par deux fois des opérations de chiffrement/déchiffrement de cryptographie asymetrique. La première opération est réalisée par le GC. Un contrôleur local décrypte et encrypte 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 symétriques, mais elles ne nécessitent pas la négociation des clés partagées.

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 LCL et la clé publique des autres contrôleurs locaux. La seule entité qui détient toutes ces informations est le GC qui doit les diffuser à tous les contrôleurs locaux par un message multicast.

5.2. *Ajout d'une nouvelle entité*

L'entité qui veut rejoindre le groupe envoie un message IGMP de type `group membership report` à son contrôleur local. Elle inclut dans son message son *token* signé. À la réception de l'`IGMP-report`, le contrôleur local authentifie le *token*.

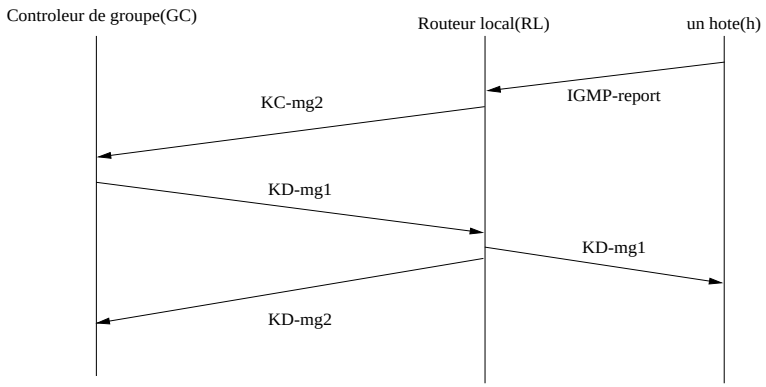

Figure 2. *Ajout d'une nouvelle entité h*

En cas de succès deux cas doivent être distingués :

1. Le routeur local est un contrôleur délégué, c'est-à-dire qu'il possède la clé de groupe K_{grp} et la liste LCL. Il génère une nouvelle clé de groupe K_{grp}' et envoie en point-à-point un message à la nouvelle entité. Ce message contient la nouvelle clé de groupe cryptée avec la clé publique de la nouvelle entité. Le même message crypté avec la clé de groupe K_{grp} est envoyé en multipoint à l'ensemble des autres contrôleurs de groupe.

2. Le routeur local n'est pas un contrôleur délégué. Il doit alors négocier avec le GC pour obtenir la permission de participer à la gestion de la clé de groupe. Le contrôleur local commence la session de négociation par le message KC_mg2 (Fig. 2). Les

messages KC_mg2, KD_mg1 et KD_mg2 sont les mêmes que ceux utilisés lors de l'initialisation de groupe. Si K_{grp} doit être remplacée par une autre clé le GC crée d'abord la nouvelle clé K_{grp}' avant de l'envoyer en multi-point avec l'entité du nouveau contrôleur dans un message crypté avec K_{grp} aux autres contrôleurs de groupe. Ensuite le message KD_mg1 est utilisé pour envoyer la nouvelle clé de groupe et la liste LCL mise à jour au nouveau contrôleur. Finalement les contrôleurs distribuent localement la nouvelle clé à leurs membres du groupe.

5.3. *Retrait d'une entité*

Pour éviter à une entité qui s'est retirée d'accéder aux communications ultérieures du groupe une nouvelle clé de groupe doit être regénérée. Le contrôleur, lié à l'entité, crée la nouvelle clé K_{grp}', la chiffre avec l'ancienne clé de groupe K_{grp} et l'envoie en multipoint aux membres du groupe et aux contrôleurs qui sont capables de la déchiffrer et d'extraire la nouvelle clé du groupe K_{grp}'. Comme le membre qui ne fait plus partie du groupe possède aussi la clé K_{grp}, il sera capable d'obtenir la nouvelle clé. Par conséquent, pour que le message ne parvienne pas à ce membre, nous proposons deux scénarios :

– Le contrôleur local qui crée la nouvelle clé l'envoie aux contrôleurs du groupe et aux membres dans son réseau local chiffré avec leur clé publique. Ensuite les différents contrôleurs locaux distribuent à leur tour la clé aux membres locaux.

– Le contrôleur local diffuse la nouvelle clé chiffrée avec la clé K_{grp}. Puisque le contrôleur local se trouve sur l'arbre de distribution du groupe alors il ne transmet pas ce message multicast aux entités dans son réseau. À la place, il leur envoie, en point-à-point, la nouvelle clé chiffrée avec leur clé publique. Le contrôleur local , qui est un routeur de multicast supportant IGMPv3, peut bénéficier de la nouvelle fonctionnalité de IGMPv3 [CAI 00] " *source filtering*" qui est la capacité d'un système de signaler son intérêt pour recevoir des données, de certaines sources adresses, envoyées à certaines adresses multipoints. Cette fonctionnalité permet à un contrôleur local d'envoyer un message multipoint, contenant la clé de groupe, aux membres du groupe, à l'exclusion de ceux existant dans son réseau local.

Nous constatons que dans le second scénario, avec des routeurs locaux supportant IGMPv3, le problème de révocation se fait par un seul message multipoint inter-domaines.

5.4. *Authentification*

Pour assurer l'authentification individuelle, nous adoptons la solution basée sur la signature à clé publique. Mais le problème est comment rendre disponible la clé publique d'un membre à tous les membres, immédiatement quand ils reçoivent un message multicast signé par le membre émetteur, et si cela est possible comment trouver l'espace nécessaire pour stocker toutes les clés publiques. Nous supposerons donc que chaque membre du groupe et le contrôleur local connaissent leur clé publique : ils peuvent utiliser le protocole SSL (Secure Sockets Layer) [FRE 96] pour échanger

leur clé publique et cela n'affecte pas les problèmes de passage à l'échelle car SSL est utilisé, dans ce cas, entre entités se trouvant dans le même domaine.

D'un autre côté tous les contrôleurs de groupe connaissent leur clé publique, résultat de la phase d'invitation.

Maintenant, soient x, y deux membres du groupe appartenant à deux réseaux locaux différents, et CL1, CL2 les contrôleurs locaux respectivement de x et y. Après avoir signé un message M, y diffuse <M,Sy>aux membres du groupe. CL2 et les membres du groupe dans le même réseau local que y peuvent authentifier y car ils connaissent sa clé publique. x authentifie y de la même manière : CL2 authentifie y car il connait sa clé publique, CL1 croit dans l'authentification de CL2 et x peut authentifier CL1. Par conséquent, x ou n'importe quel autre membre du groupe peut autentifier y.

Notons que, pour assurer l'authentification individuelle dans cette proposition, les membres du groupe subissent des opérations très lourdes en traitement processeur. Pour réduire ce coût, nous pouvons adopter les mécanismes de clés partagées proposés par Canetti [CAN 98] de la manière suivante : chaque contrôleur local gère un ensemble de clés à partager avec les membres de groupe à portée locale. Le GC détient un ensemble de clés à être partagées avec les contrôleurs locaux. Dans ce cas, x authentifie y ainsi :

– CL2 authentifie y car ils partagent les clés d'authentification de CL2 ;
– CL1 authentifie CL2 car ils partagent les clés d'authentification de GC ;
– finalement x authentifie CL1 car ils partagent les clés d'authentification de CL1.

Dans cette proposition, le trafic de groupe subit aussi beaucoup d'opération de signature/vérification mais ces opérations sont symétriques donc moins coûteuses pour l'authentification individuelle.

6. Analyse et comparaison

Dans cette section, nous analysons notre approche et la comparons avec d'autres, surtout celles qui résolvent les problèmes d'extensibilité c'est-à-dire les approches hiérarchiques.

Par la suite, nous notons k la taille d'une clé cryptographique, α le coefficient de participation et n la taille du groupe. Donc, n/α représente le nombre total de contrôleurs.

Lors de l'initialisation de groupe, la taille de transmission 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 alors la taille de transmission reste petite.

La taille de transmission, pour renouveller K_{grp} après une addition ou une révocation d'un membre, est égale à k, *i.e.* un seul message pour renouveller la clé de groupe.

Nous constatons que l'espace de stockage pour un membre est égal :

– à $(\alpha + 1).k$ avec l'authentification individuelle où $\alpha.k$ sont les clés publiques des membres dans un réseau local ;

– à k sans l'authentification individuelle.

Par contre, pour un contrôleur l'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 ; donc l'espace de stockage est de O(1).

En comparant `Baal` avec les approches hiérarchiques [MIT 97, WON 98, MCG 98] qui assurent *forward (backward) secrecy*, nous constatons que ces dernières résolvent le problème d'extensibilité par le biais de la hiérarchie des clés ou des systèmes. 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 révocation d'un membre), le nombre de message échangés est de l'ordre de $log(n)$. Par contre `Baal` résoud le même problème en $O(1)$; en délégant des contrôleur locaux, qui sont des routeurs multicast 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 messages point-à-point. Ces message point-à-point sont intra-domaine, par conséquent, ne posent pas de problèmes d'extensibilité.

Le tableau 1 récapitule le résultat de comparaison de notre approche avec les deux approches hiérarchiques OFT, LKH et SKDC. Dans le tableau h représente la hauteur de l'arbre hiérarchique et est égal à $log(n)$ pour les arbres équilibrés. Nous constatons que `Baal` effectue moins de transmission que les autres approches. Ainsi, `Baal` exige un espace de stockage chez un contrôleur plus petit. En conséquence, tant que α est grand alors `Baal` est efficace.

	Baal	SKDC	LKH	OFT
taille de transmission(init.)	$(n/\alpha).k$	nk	$2nk + h$	$2nk + h$
taille de transmission(rekey)	k	nk	$2hk + h$	$hk + h$
Stockage de manager	$(n/\alpha).k$	nk	$2nk$	$2nk$
Stockage d'un membre	$\alpha.k$	$2k$	hk	hk

Tableau 1. *résultat de comparaison de* `Baal` *avec SKDC, LKH et OFT*

7. Conclusion et perspectives

Nous avons spécifié un nouveau protocole, `Baal`, pour la gestion des clés de groupe. `Baal` présente une solution extensible au problème de révocation et au problème d'authentification individuelle. Il est indépendant des protocoles de routage multipoint, des protocole de gestion de clés point-à-point.

`Baal` améliore l'extensibilité en décentralisant la gestion de clés de groupe. En effet, il permet au contrôleur de groupe de déléguer des contrôleurs locaux, au niveau des réseaux locaux où il existe des membres du groupe, pour gérer l'accès au groupe et la clé de groupe. Un contrôleur local est un routeur multipoint supportant IGMPv3 et bénificiant de la nouvelle fonctionnalité de IGMPv3, *source filtering*, pour envoyer le message de renouvellement de la clé de groupe en multipoint à tous les membres du groupe à l'exception de ceux situés dans son réseau local.

`Baal` assure l'authentification individuelle basée sur la signature à clé publique. L'authentification se fait en trois étapes : un contrôleur authentifie l'émetteur, le contrôleur du récepteur authentifie le contrôleur de l'émetteur et le récepteur authentifie son contrôleur. À signaler que `Baal` est le seul protocole parmi toutes les autres solutions à assurer l'authentification individuelle.

Nous avons montré que `Baal` nécessite une taille de transmission $(n/\alpha).k$, pour distribuer la clé de groupe et la renouveller, plus petite que les autres approches. Nous avons également vu tant que α, le coefficient de participation, est grand alors la taille de transmission reste petite. Par conséquent, en fonction de l'ordre de grandeur du cœfficient de participation, donc du nombre de participants gérés par un contrôleur, nous envisageons, dans la suite de nos travaux, la possibilité d'avoir un contrôleur délégué pour plusieurs réseaux locaux, *i.e.* par système autonome ou domaine de routage. De même, nous nous orientons vers une solution basée sur les clés partagées pour résoudre de façon plus optimale l'authentification individuelle.

8. Bibliographie

[ATK 98] ATKINSON R., KENT S., « Security Architecture for the Internet Protocol », November 1998, Request For Comments rfc-2401: Network Working Group.

[BAL 96] BALLARDIE T., « Scalable Multicast Key Distribution », may 1996, Request For Comments rfc-1949: Network Working Group.

[BAL 99] BALENSON D., MCGREW D., SHERMAN A., « Key Management for Large Dynamic Groups: One-way Function Trees and Amortized Initialization », February 1999, Intenet draft: draft-balenson-groupkeymgmt-oft-00.txt.

[BLU 92] BLUNDO C., SANTIS A., HERZBERG A., KUTTEN S., VACCARO U., YUNG M., « Perfectly-Secure Key Distribution for Dynamic Conferences », Advances in Cryptology: proceedings of Crypto92,E. F. Brickell, Ed., LNCS 740, Springer-Verlag (1992), 471-486, 1992.

[BUR 97] BURMESTER M., DESMEDT Y. G., « Efficient and Secure Conference-Key Distribution », Secure Protocole, M. Lomas, Ed. , LNCS 1189, Springer-Verlag, 119-130., 1997.

[CAI 00] CAIN B., DEERING S., I. K., THYAGARAJAN A., « Internet Group Management Protocol, Version 3 », June 2000, Internet draft.

[CAN 98] CANETTI R., PINKAS B., GARAY J., MICCIIANCIO D., NOAR M., ITKIS G., « Multicast Security: A Taxonomy and Efficient Authentication », rapport, April 1998, IBM, Rapport de recherche.

[CHA 99] CHADDOUD G., CHRISMENT I., SCHAFF A., « Vers Communication de groupe sécurisée : état de l'art », Rapport de recherche, Novembre 99, LORIA.

[FIA 93] FIAT A., NOAR M., « Broadcast Encryption », rapport, 1993.

[FLO 95] FLOYD S., JACOBSON V., LIU C., MCCANNE S., ZHANG L., « A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing. », ACM-SIGCOMM'95, August 1995, p. 352-356.

[FRE 96] FREIER A., KARLTON P., KOCHER P., « The SSL Protocol Version 3.0 », March 1996, ftp://ftp.netscape.com/pub/review/ssl-spec.tar.Z.

[HAN 98] HANDLEY M., JACOBSON V., « Session Description Protocol », APril 1998, Request For Comments rfc-2327: Network Working Group.

[HAN 00] HANDLEY M., PERKINS C., WHELAN E., « Session Announcement Protocol », February 2000, Internet draft: draft-ietf-mmusic-sap-v2-05.txt.

[HAR 97a] HARNEY H., MUCKNHIRN C., « Group Key Management Protocol (GKMP) Architecture », July 1997, Request For Comments rfc-2094: Network Working Group.

[HAR 97b] HARNEY H., MUCKNHIRN C., « Group Key Management Protocol (GKMP) Specification », July 1997, Request For Comments rfc-2093: Network Working Group.

[HAR 98] HARKINS D., CARREL D., « The Internet Key Exchange (IKE) », March 1998, RFC: <draft-ietf-ipsec-isakmp-oakley-07.txt>.

[HAR 99a] HARDJONO T., CAIN B., DORASWAMY N., « A Framework for Group Key Management for Multicast Security », August 1999, Intrenet draft: draft-ietf-ipsec-gkmframework-01.txt.

[HAR 99b] HARDJONO T., CAIN B., MONGA I., « Intra-Domain Group Key Management Protocol », August 1999, Internet draft: draft-ietf-ipsec-intragkm-00.txt.

[MAU 98] MAUGHAN D., SCHERTLER M., SCHNEIDER M., « Internet Security Association and Key Management Protocol (ISAKMP) », March 1998, Internet draft: <draft-ietf-ipsec-isakmp-09.txt>.

[MCG 98] MCGREW D. A., SHERMAN A. T., « Key Establishment in Large Dynamic Groups using One-way Function Trees », TIS Labs at Network Associates, Inc. Gleenwood, Maryland, 1998.

[MIT 97] MITTRA S., « Iolus: A Framework for Scalable Secure Multicasting », ACM-SIGCOMM'97, septembre 1997.

[PUS 99] PUSTERI T., « Distance Vector Multicast Routing Protocol », February 1999, Internet draft: <draft-ietf-idmr-dvmrp-v3-08>.

[SCH 97] SCHNEIER B., *Cryptographie Appliquée*, International Thomson Publishing, 1997, Traduction de L. Viennot.

[STE 96] STEINER M., TSUDIK G., WAINDER M., « Deffie-Hellmen Key Distribution Extended to Group Communication », 3rd ACM conference on Computer and Communication Security, New Delhi, India, 14-16March 1996.

[WAL 98] WALLNER D., HARDER E., AGEE R., « Key Management for Multicast: Issues and Architecture », September 1998, Intenet draft: draft-wallner-key-arch-01.txt.

[WON 98] WONG C., GOUDA M., LAM S., « Secure Group Communications using Key Graphs », ACM-SIGCOMM'98, septembre 1998.