Méthodes d'Authentification pour les Communications de Groupes : Taxonomie et Evaluation dans un environnement Ad Hoc

Mohamed Salah Bouassida, Isabelle Chrisment, Olivier Festor

Méthodes d'Authentification pour les Communications de Groupes : Taxonomie et Évaluation dans un environnement Ad Hoc

Mohamed Salah Bouassida et Isabelle Chrisment et Olivier Festor

MADYNES-LORIA, Campus scientifique, B.P. 239, 54506 Vandœuvre-lès-Nancy Cedex - France
tél : +33-3-83-59-30-49 - fax : +33-3-83-41-30-79

Les communications de groupe dans l'Internet offrent un service efficace pour acheminer les données et sont bien adaptées pour de nombreuses applications multimédia. L'utilisation de ces applications à des fins commerciales et privées a nécessité la mise en place de services de sécurité ; ce qui a conduit à de nombreuses propositions de recherche notamment pour le service d'authentification qui représente la clé de voûte de toute architecture de sécurité. Mettre en œuvre les communications de groupe dans un environnement Ad Hoc est un véritable challenge. Dans cet article, nous présentons une classification des approches d'authentification dans les communications de groupe et plus spécifiquement celles de l'authentification de la source et nous étudions leur adéquation dans le cadre des réseaux Ad Hoc en fonction de certains critères que nous avons définis et qui sont liés au contexte Ad Hoc.

Mots-clés: sécurité, multicast, authentification, source, réseaux Ad Hoc

1 Introduction

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

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

Parallèlement le réseau sans fil est devenu une réalité de fait avec l'émergence des nouvelles technologies et standards comme Wi-Fi, Hyperlan,... Ces réseaux sans fil peuvent être utilisés en mode Ad Hoc c'est-à-dire sans infrastructure fixe où chaque nœud est également routeur. Une approche hybride où les réseaux Ad Hoc sont connectés à un passerelle reliée au monde Internet filaire donne la possibilité d'étendre la couverture des réseaux à un faible coût. Cependant la sécurité constitue actuellement l'un des principaux obstacles à un large déploiement des réseaux Ad

[†] http://www.securemulticast.org/msec-index.htm

Hoc. Sécuriser un réseau Ad Hoc revient à instancier les différents services de sécurité, tout en prenant en compte les différentes caractéristiques d'un tel réseau.

Dans le cadre du projet SAFARI [‡], nous travaillons sur l'adaptation des mécanismes de sécurité disponibles pour les services multicast dans le contexte d'un réseau Ad Hoc. Nous nous sommes intéressés au service d'authentification qui représente la clé de voûte de toute architecture de sécurité. En effet, 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. 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.

Dans ce papier, nous identifions les méthodes d'authentification multicast, nous les classifions et nous montrons comment les contraintes d'un environnement Ad Hoc nous permettent de définir des critères qui vont influer sur le choix de la méthode d'authentification à utiliser.

La section 2 présente une classification des approches existantes pour l'authentification multicast. Nous détaillons plus spécifiquement dans la section 3 les méthodes d'authentification de la source car celle des membres s'effectue individuellement et nous ramène à une authentification point à point.

La section 4 présente les contraintes et challenges de l'authentification dans les réseaux Ad Hoc. La section 5 définit les critères permettant d'évaluer l'adéquation des méthodes d'authentification de groupe dans un contexte Ad Hoc. La section 6 conclut.

2 Classification des approches d'authentification pour les communications de groupe

Dans les architectures de sécurité multicast, le contrôle du groupe dicté par des politiques de sécurité préalablement définies, requiert l'authentification des membres ou l'authentification de la source ou les deux en même temps.

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. Plusieurs travaux de recherche ont été menés à ce sujet et ont abouti à deux types d'approches:

- Dépendance par rapport à un tiers de confiance défini:

  L'infrastructure à clé publique (PKI) appartient à cette approche. Le tiers de confiance dans cette méthode appelé CA (Autorité de certification) doit garantir la validité du lien Identité/Clé publique par la simple apposition de sa propre clé privée ; il délivre ainsi un certificat à tout nœud du réseau. Le CA joue aussi le rôle de déploiement à grande échelle des certificats, et de leurs gestion (génération, validation, révocation,...). Le service CA fait appel à d'autres services du réseau pour définir toutes les instances nécessaires à un certificat, tel que le service de nom (certificats X.509) [HFPS99].

  Cependant, ces certificats sont très contraignants et sont difficilement exploitables dans des réseaux n'offrant pas un système centralisé et hiérarchique, capable de fournir une service d'horodatage et un service de nom à grande échelle. SPKI [EFL+99] (Simple Public Key Infrastructure), tout comme SDSI [RL96] (Simple Distributed Security Infrastructure), ont été proposés pour remédier à ce problème, en offrant une souplesse et une dynamicité dans les noms. Ces deux méthodes restent toujours dépendants par rapport à une autorité de nommage.

- Non ou faible dépendance par rapport à un tiers de confiance défini: L'approche PGP (Pretty Good Privacy) [ASZ96] se base sur le principe "les amis de mes amis sont mes amis", et laisse donc la gestion de la confiance sous l'appréciation de l'usager en fonction de critères sociaux et techniques qu'il estime.

  Les CBIDs (Crypto-based Identifiers) offrent aussi l'authentification des membres sans recours à un tiers de confiance. Cette méthode a été élaborée par [MC02] pour résoudre le problème d'identification, en offrant des identificateurs uniques et cryptographiquement vérifiables.

[‡] SAFARI est un projet RNRT qui vise à concevoir, combiner et réaliser une infrastructure protocolaire et logicielle nécessaire à l'accès transparent, la configuration automatique, l'intégration et l'adaptation des services sur un réseau IPv6 en mode Ad Hoc comportant des accès filaires.

L'authentification avec cryptographie à seuil [ZH99] est une méthode qui a été proposée dans le cadre des réseaux Ad Hoc. Elle permet d'émuler un service CA dans le réseau via le partage de sa clé privée en *n* secrets distribués à *n* nœuds spéciaux appelés serveurs.

Selon le type d'application de communications de groupe à sécuriser, et selon les puissances de calculs disponibles côté membres du groupe, on peut distinguer trois niveaux d'authentification de la source [Esk02]:

- Authentification de groupe : Elle fournit l'assurance qu'un paquet a été envoyé par un membre inscrit au groupe (source ou récepteur).
- Authentification des données de la source : Elle fournit l'assurance qu'un paquet a été envoyé par une source du groupe et non par un récepteur.
- Authentification individuelle de la source : Elle fournit l'assurance de l'identité de la source de données et sa non répudiation.

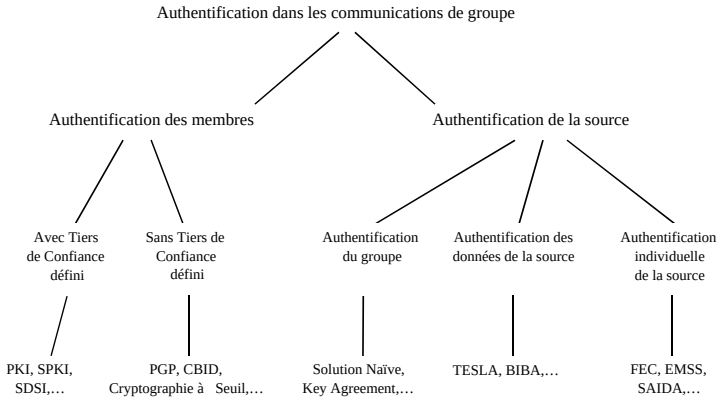

Fig. 1: Classification des approches d'authentification dans les communications de groupe

La figure 1 montre la classification que nous proposons pour les méthodes d'authentification dans les communications de groupe.

Dans ce qui suit, nous nous focalisons sur les approches d'authentification de la source. Nous présentons une méthode par famille et nous discuterons dans la section 5 de leur adéquation dans le cadre des réseaux Ad Hoc.

3 Authentification de la source dans les communications de groupe

3.1 Authentification du groupe

L'authentification du groupe assure que chaque paquet émis en multicast pour les membres du groupe, provient d'une source enregistrée dans le groupe.

La solution la plus intuitive et la plus naïve pour résoudre ce problème consiste à élire un controlleur de groupe centralisé qui génère et distribue une clé de groupe à tous ses membres locaux de nombre *n*. A chaque Join ou Leave, le controlleur met à jour sa clé de groupe et la distribue en unicast à chaque membre. Quand une source, membre du groupe, veut envoyer des données en multicast à tous les membres du groupe, elle authentifie chaque paquet avec la clé du groupe et envoie le paquet ainsi que son MAC en multicast. Les récepteurs vérifient l'authenticité du groupe en vérifiant le MAC avec la clé du groupe. Ce shéma présente l'avantage d'être simple et facile à implémenter. Cependant, il n'assure pas le passage à l'échelle et est impratiquable dans des réseaux de forte dynamicité et de grande taille.

L'approche "Key Agreement" [JV96, AG00] fait partie des protocoles d'établissement de clés (KEP : Key Establishment Protocols). Ces protocoles visent l'établissement d'un canal sûr, favorable à l'échange de clés par la preuve de l'authenticité des interlocuteurs. Le contexte de "Key Agreement" est un petit groupe de personnes dans une conférence, présentes ensemble dans une salle pour une réunion Ad Hoc. Ces personnes veulent s'échanger des données secrètement durant la durée de la réunion.

Le principe de ce protocole consiste, sachant que tous les nœuds présents ont confiance les uns aux autres, à partager un mot de passe faible à partir duquel un mot de passe fort sera généré et sera la clé de session du groupe. Cette clé partagée entre les différents membres du groupe peut être utilisée pour signer tous les paquets envoyés en flux multicast et ainsi assurer l'authentification du groupe, ou bien chiffrer le trafic multicast et ainsi assurer en plus la confidentialité.

Ce protocole tel qu'il est présenté par [AG00] doit avoir les propriétés suivantes:

- Secret : seuls les gens qui connaissent le premier mot de passe doivent être capables de savoir la clé de session résultante.
- Accord contribuant : la clé de session générée doit être composée des contributions des différents participants à la session.
- La tolérance contre les attaques : les attaques prises en compte sont celles qui peuvent insérer des messages mais qui ne peuvent pas modifier ou supprimer des messages envoyés par d'autres personnes.

[AG00] a commencé par présenter un protocole générique d'authentification EKE (Encrypted Key Exchange); les intervenants dans EKE sont deux nœuds A et B dans un réseau Ad Hoc qui veulent se mettre d'accord sur une clé forte de session K, de sorte qu'un attaquant ne puisse pas connaître K et ne puisse non plus attaquer le mot de passe faible de départ p (Dictionary Attack).

[AG00] propose d'étendre le protocole EKE pour qu'il soit un protocole multi-parties, la seule contrainte est qu'un leader doit déclencher les opérations d'authentification et d'échanges de messages ; ce leader constitue ainsi un point de vulnérabilité du réseau Ad Hoc. Ce nouveau protocole tel qu'il a été présenté ne satisfait pas la contrainte d'accord contribuant, car c'est le leader qui calcule la clé de session globale et la diffuse à tous les autres nœuds.

Pour remédier à ce problème, des modifications ont été apportées au protocole de base EKE pour avoir un protocole multi-parties qui permet à tous les intervenants de contribuer à la génération de la clé de session K. Cette modification est très contraignante puisque le leader doit attendre toutes les contributions des autres nœuds pour pouvoir calculer K. Ce protocole doit être accompagné d'un autre protocole additionnel qui pourrait convaincre les nœuds du réseau que la clé diffusée par le leader est bien la clé correcte du système asymétrique de cryptage.
Pour fournir un secret partagé entre les différents participants et rendre plus efficace le nombre de messages échangés, [AG00] présente une amélioration de Diffie-Helmann en arrangeant tous les participants dans un hypercube.

Tous les protocoles de la famille "Key Agreement" résolvent le problème d'authentification pour les communications multicast sans avoir besoin d'autres infrastructures additionnelles ou de canaux physiques de communication sécurisée, ce qui les rend adaptés à la nature des réseaux Ad Hoc [LAU03]. Cependant, la dynamicité des réseaux Ad Hoc n'est pas prise en compte puisque ces protocoles supposent que la composition du groupe ne change pas durant la session. De plus, ces protocoles ne permettent pas le passage à l'échelle.

3.2 Authentification des données de la source

Cette famille de méthodes d'authentification assure qu'un paquet a été envoyé par une source du groupe et non par un récepteur. Toutefois, elle n'assure pas la non répudiation de la source. Plusieurs approches d'authentification appartiennent à cette famille comme par exemple Tesla [HD03] et BIBA [Per01].

Dans la suite de cette section, nous présentons l'approche Tesla [HD03] (Timed Efficient Stream Loss-Tolerant Authentication) qui fournit l'authentification basée sur MAC [MS01] de la source d'un flux multicast tolérant les pertes. Le flux de données dans Tesla est unidirectionnel : les données transitent seulement de la source vers les récepteurs. Ceci implique que le surcoût de l'authentification de la source est indépendant du nombre de récepteurs. L'architecture permet donc le passage à l'échelle.

Les concepts de base de Tesla reposent sur:

1. Les intervalles de temps : chaque paquet P_i est authentifié séparément, avec MAC. Le temps est divisé en t intervalles de durée T_{int} chacun. L'émetteur peut envoyer 0 ou plus paquets par intervalle I_j. A chaque intervalle I_j lui correspond une clé d'authentification k'_j.
2. Clés MAC : l'émetteur génère une chaîne de clés, k_1, k_2, ..., k_t en utilisant une fonction à sens unique. La dernière clé de la chaîne k_t est d'abord générée aléatoirement et les autres sont dérivées en utilisant l'équation : $k_{j-1} = f(k_j)$. Ensuite, l'émetteur génère les clés MAC $k'_j = g(k_j)$ avec g une autre fonction à sens unique.

   De cette architecture découle une propriété importante de TESLA qui est la tolérance aux pertes de paquets. En effet, même si tous les paquets envoyés dans un intervalle donné sont perdues (et par conséquent toutes les clés

révélées à cet intervalle), les récepteurs pourront authentifier les paquets en se basant sur les intervelles futurs. Par exemple, les paquets envoyés pendant l'intervalle I_j peuvent être authentifiés même si tous les paquets émis pendant l'intervalle I_{j+d} (où k_j est révélée) sont perdus. d étant le délai de révélation de clés présenté ci-dessous. Un récepteur peut toujours calculer k_j depuis n'importe quelle clé k_{j+m} avec $m >= d$.

3. Validation des clés de la chaîne: la source peut valider les clés de la chaîne en signant un paquet contenant une clé de la chaîne, ou en incluant cette clé dans un paquet authentifié. Par exemple, pour valider les clés de chaîne k_1, k_2, ..., k_t, la source peut envoyer un paquet authentifié contenant la clé $k_0 = f(k_1)$.

4. Temps de synchronisation entre émetteurs et récepteurs : les récepteurs ont besoin de connaître une limite supérieure du temps de la source. Ainsi, si la différence de temps entre la source et l'émetteur est δt, on suppose que les récepteurs connaissent un Δt tel que $\Delta t >= \delta t$ [HD03].

5. Délai de révélation des clés ou d : ce délai indique le temps (nombre d'intervalles) dont le récepteur a besoin pour être capable d'authentifier un paquet dans un intervalle I_j. Ce délai a une conséquence directe sur l'espace requis de stockage côté récepteur.

Le délai de révélation d est crucial dans TESLA. En effet, le choix d'un d petit rend difficile l'authentification de paquets par des récepteurs assez loins de la source. Par contre, le choix d'un d grand obligerait un espace de stockage assez important côté récepteur. Pour cela, il existe d'autres méthodes d'authentification dans TESLA :

- Authentification immédiate : TESLA propose un mécanisme permettant une authentification immédiate des paquets par les récepteurs. Chaque paquet émis par la source, contient un hachage du futur paquet. Cette méthode implique un surcoût au niveau taille de paquet. En plus, elle n'est plus robuste contre les pertes de paquets.

- Récepteurs hétérogènes : Les récepteurs proches de la source préfèrent un délai de révélation de clé assez court pour pouvoir profiter d'un court délai d'authentification. Par contre, les récepteurs assez loins ne pourront pas opérer avec un délai de révélation court du fait que le temps de transmission des paquets peut dépasser ce délai et par conséquent la condition de sécurité sera violée et les paquets ne pourront pas être authentifiés.

  La solution présentée par [PCST01] consiste donc au fait que le délai de révélation d peut ne pas être le même pour tous les récepteurs. Bien que des membres puissent être capables d'authentifier les paquets avant d'autres, ils ne pourront pas modifier le contenu du flux multicast. Et ainsi les conditions de sécurité restent respectées. Ainsi, plusieurs instances de TESLA peuvent être lancées en même temps, avec une chaîne de clés chacune et un délai de révélation différent. Chaque récepteur peut décider quelle instance de tESLA utiliser et avec quel délai de révélation.

Il existe une nouvelle version de TESLA adaptée aux réseaux de sonde, 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. Les améliorations de cette approche par rapport à TESLA sont détaillées dans [PSW+02].

[Per01] propose le schéma de sécurité BIBA qui fournit une signature plus compacte et plus facile à vérifier, au coût d'une clé publique de vérification qui peut être grande. Cette approche supporte une transmission d'un flux de données temps réel, avec une authentification efficace, et fournit une parfaite robustesse aux pertes et un passage à l'échelle du nombre des récepteurs. Contrairement à TESLA, l'authentification dans BIBA est immédiate et instantanée, et ne dépend pas des erreurs dus aux temps de synchronisation. Cependant, une erreur de synchronisation grande requiert du côté des récepteurs un grand espace mémoire de stockage.

3.3 Authentification individuelle de la source

Cette famille de méthodes d'authentification se propose de fournir l'assurance de l'identité de la source de données et sa non répudiation.

Plusieurs approches appartiennent à cette famille, tels que EMSS [PCTS00], SAIDA [PCS02] et FEC [PM03].

EMSS (Efficient Multi-chained Stream Signature) est basé sur le fait de signer un petit nombre de paquets spéciaux dans un flux de données. Chaque paquet du flux, est lié à un paquet signé via des chaînes de hachage multiples. EMSS amortit ainsi le coût de signature (1 opération de signature par 100 à 1000 paquets). Cette approche tolère les pertes de paquets, détient un faible surcôut de communication et assure la non répudiation de la source.

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 décodeur FEC.

[PM03] combine des techniques de hachage et de FEC pour assurer une authentification efficace des paquets multicast.

La nouvelle approche utilisant FEC présente un faible surcôut par rapport à EMSS, et offre une probabilité plus forte d'effectuer la vérification de l'authentification des paquets avant le temps maximum requis de latence d'authentification.

Il existe plusieurs similarités entre SAIDA et FEC (Schéma ECU). Cependant, le surcoût de communication par paquet est plus faible dans FEC que dans SAIDA. En plus, le schéma utilisant FEC utilise deux niveaux de code d'effacement (FEC) alors que SAIDA utilise un seul code de redondance.

Plusieurs comparaisons ont été réalisées entre ces différentes méthodes et bien d'autres [PM03, CCF03]. Dans ce qui suit et pour les raisons citées ci-dessus, nous détaillons l'authentification utilisant FEC. Cette technique assure que Les mécanismes d'authentification soient robustes contre les pertes potentielles de paquets dans un environnement d'émission multicast hostile, tout en ayant un faible surcoût vu la contrainte temps réel des données émises.

Le nouveau schéma d'authentification multicast proposé par [PM03] assure l'intégrité, l'authentification et la non répudiation de la source. Ce schéma offre aussi l'accessibilité (la capacité des récepteurs à accéder au flux multicast et à authentifier les paquets à n'importe quel point du flux) et la robustesse.

On considère un bloc comme une séquence de b paquets [P_1,...,P_b].
h_1,...,h_b| h_i <− H(P_i) sont les valeurs de hachage de ces paquets. H étant une fonction de hachage cryptographique telle que SHA ou MD5 ; cette fonction produit un hachage de h bits. À partir de ces valeurs de hachage, on va construire une suite de tags d'authentification τ_1,...,τ_b.

Le critère d'authentification est le suivant : un paquet P_i est parfaitement authentifié dans un bloc si, étant donné la suite h_1,...,h_b de paquets du bloc et leurs signatures $\sigma = S(H(h_1||h_2||...||h_p))$, on peut vérifier que :
$V(\sigma, H(h_1||h_2||...||h_p) = true$ et que H(P_i) = h_i.

Le couple (S, V) dénote respectivement la signature digitale et les algorithmes de vérification, associé à la source du flux. Autrement dit, c'est le couple de clés privée et publique de la source de données. On peut utiliser RSA par exemple pour générer le couple (S, V). La taille de la signature est s bits. s peut être égal à 128 ou 1024.

[PM03] a montré dans sa proposition que n'importe quelle suite d'au moins $b(1-p)$ paquets peut être authentifiée par n'importe quelle suite d'au moins $b(1-p)$ tags d'authentification.

Il existe différentes alternatives de schémas d'émissions du flux multicast (ECU, EC1 et EC2).

Considérons un flux de données comme m blocs B_1,...,B_m. Les paquets d'un bloc B_i sont P[i,1],...,P[i,b]. Les tags d'authentifications correspondants sont notés $\tau[i,1],...,\tau[i,b]$. Les paquets P[i,j] sont la combinaison des paquets de données D[i,j] et de tags d'authentification.

* ECU (unbuffered sender sheme). Le bloc $B_{(i+1)}$ est utilisé pour l'émission des tags d'authentification appartenant au bloc B_i. Le $j^{\text{ème}}$ paquet du bloc P_i sera ainsi défini comme : $P[i,j] = D[i,j]||\tau[i-1,j]$. Ceci requiert à la source de créer un bloc en plus (padding bloc) $B_{(m+1)}$ pour permettre au dernier bloc B_m d'être authentifié. Par contre, aucun stockage n'est nécessaire du côté de l'émetteur. Les récepteurs doivent attendre dans le pire des cas la réception de deux blocs pour pouvoir authentifier le premier paquet du premier bloc.

* EC2 (double buffered sheme). Le bloc $B_{(i-1)}$ est utilisé pour l'émission des tags d'authentification appartenant au bloc B_i. Le $j^{\text{ème}}$ paquet du bloc P_i sera ainsi défini comme : $P[i,j] = D[i,j]||\tau[i+1,j]$. Ceci requiert à la source le stockage de deux blocs à la fois. Du côté des récepteurs, l'authentification se fait automatiquement dès la réception d'un bloc.

* EC1 : (single buffered sheme). Dans ce schéma, les tags d'authentification du bloc B_i sont émis au sein du bloc B_i lui même. Le $j^{\text{ème}}$ paquet du bloc P_i sera ainsi défini comme : $P[i,j] = D[i,j]||\tau[i,j]$. Ce schéma représente un compromis entre ECU et EC2. Un avantage de ce schéma est qu'il ne crée pas de dépendances entre blocs.

4 Contraintes et challenges des communications de groupes dans les réseaux Ad Hoc

Assurer le service d'authentification pour les communications multicast dans les réseaux Ad Hoc se révèle un véritable challenge de part la nature même des réseaux Ad Hoc, du niveau de sécurité à instaurer et des caractéristiques et types

des applications à sécuriser. Les défis de l'authentification des communications multicast dans les réseaux Ad Hoc sont les suivants :

- L'utilisation des liens sans fil rend un réseau Ad Hoc susceptible d'être exposé à des attaques malicieuses passives comme des écoutes clandestines, ou actives comme ré-envoyer un message ou le déformer. Ces attaques violent ainsi le service d'authentification [ZH99].

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

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

  * La répartition de la clé privée sur plusieurs nœuds pour l'autorité de confiance mobile : la clé privée d'une autorité de certification ou CA joue un rôle essentiel pour l'établissement de la confiance dans le réseau. En effet, elle assure le mécanisme de signature des certificats. Cependant, l'établissement d'une PKI basée sur une unique entité centralisée suppose que cette entité de confiance ne pourra jamais être compromise, ce qui ne reflète pas la réalité hostile de l'environnement Ad Hoc.
  * La gestion des contextes de confiance des nœuds en cours de mobilité : l'autorité de certification est responsable de la gestion des certificats (émission, renouvellement, révocation, ...). Pour cela, elle héberge ces données confidentielles dans des bases de données spécialisées. Dans le cadre des réseaux Ad Hoc, ce système de stockage de données n'est plus adapté, à cause de la forte mobilité des nœuds Ad Hoc. Une solution plus appropriée aux réseaux Ad Hoc consisterait à ce que les données relatives à un nœud soient hébergées par le nœud lui même grâce à une carte à puce.
  * L'émission des certificats : un certificat X.509, dans une architecture PKI, est signé par la clé privée de la CA, et contient des références comme l'horodatage et le nom X.500. Ce schéma n'est plus possible pour les réseaux Ad Hoc. Plusieurs travaux de recherche ont ainsi cherché à alléger le certificat X.509 pour un certificat plus souple et plus libre utilisant SPKI [EFL+99].

- 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. Ainsi le service d'authentifiation doit faire face à la dynamicité et au passage à l'échelle des groupes multicast dans les réseaux Ad Hoc.

- La mobilité des nœuds Ad Hoc doit aussi être prise en compte pour assurer le service d'authentification. 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 transmis 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 on doit faire confiance pour assurer le service d'authentification, quel niveau de confiance faut il leur donner, ...

Les défis apportées par les réseaux Ad Hoc vont impliquer des critères et des fonctionnalités de sécurité que doivent apporter les architectures de sécurité proposées. Dans la section suivante, nous définissons ces critères d'analyse et nous étudions l'adéquation des différentes méthodes d'authentification de la source dans le cadre des réseaux Ad Hoc selon ces critères.

5 Critères d'analyse et discussions

5.1 Critères

Dans le tableau de la figure 2, nous présentons une récapitulation des différentes méthodes d'authentification de la source, présentées ci dessus. Le tableau définit le contexte et les objectifs de chaque méthode.

	Contexte	Fonctionnalités
Key Agreement	Ensemble restreint de nœuds Ad Hoc	Partager un secret (clé de session) et établir des communications sécurisées en multicast (1 vers n ou n vers n)
TESLA	Un flux de données multicast non fiable émis par une source vers des récepteurs (1 vers n)	Authentification de la source Permettre le passage à l'échelle Faible surcoût de calcul et de communication Robustesse et tolérance aux pertes de paquets
Authentification utilisant FEC	Un flux de données multicast non fiable émis par une source vers des récepteurs (1 vers n)	Authentification et non répudiation de la source Tolérance aux pertes de données Assurer l'intégrité des données Diffusion en temps réel Faible surcoût de communication

Fig. 2: Contextes et fonctionnalités des méthodes d'authentification présentées

Pour pouvoir juger de l'adéquation de ces méthodes dans le cadre des réseaux Ad Hoc, nous avons défini les critères d'analyse suivants :

- Robustesse : la capacité de l'architecture d'authentification à réagir face aux pertes de données.
  - L'approche "Key Agreement" ne prévoit pas de solutions contre les pertes de données. Ces pertes peuvent être problématiques surtout lors de la phase initiale de détermination de la clé de session.
  - Par contre, l'authentification avec TESLA est robuste contre les pertes de paquets (excepté le cas de l'authentification immédiate avec TESLA, où chaque paquet émis contient le hachage du paquet suivant).
  - Tout comme TESLA, l'authentification avec FEC est robuste contre les pertes de paquets. FEC permet aussi la correction des pertes de données.
- Accessibilité : 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.
  - L'accessibilité est fournie par l'authentification avec FEC, plus difficilement par TESLA, à cause de sa phase de synchronisation et d'initialisation.
  - Pour les protocoles de "Key Agreement", l'accessibilité est très faible car la clé de session doit être recalculée pour prendre en compte la contribution du nouveau membre.
- Stockage des données : le nombre de paquets maximum que la source ou les récepteurs doivent stocker.
  - "Key Agreement" ne requiert pas de stockage mémoire, ni côté source, ni côté récepteurs.
  - L'authentification selon TESLA ne nécessite pas de stockage mémoire côté source. Les récepteurs, par contre, doivent stocker des paquets pour d intervalles de temps dans le pire des cas. (d étant le délai de révélation de clé dans TESLA).
  - L'authentification utilisant FEC, selon ses alternatives, requiert ou non un stockage en mémoire des données. Pour ECU (Unbuffered Sender Sheme), aucun stockage mémoire n'est nécessaire du côté de l'émetteur. Par contre, les récepteurs doivent stocker au maximum, deux blocs de données. Pour EC2 (Double Buffered Sheme), du côté des récepteurs, l'authentification se fait automatiquement sans stockage en mémoire. Tandis que la source doit stocker à la fois deux blocs de données. Pour EC1, (Single Buffered Sheme), aucun stockage n'est nécessaire. Le tableau de la figure 3 résume ces comparaisons.

		Stockage des données		Délai d'authentification
		Source	Récepteurs	
Key Agreement		0	0	4
TESLA		0	Paquets reçus en d intervalles	paquets reçus en d intervalles +1
FEC	ECU	0	2b	2b
	EC2	2b	0	0
	EC1	0	0	0

Fig. 3: Stockage des données et délai d'authentification

- Délai d'authentication : le nombre de paquets maximum que les récepteurs doivent recevoir pour pouvoir authentifer le premier paquet.
  - Selon le protocole "Key agreement", un nœud doit attendre la réception de 4 paquets afin que la clé de session soit établie.
  - Pour l'authentification utilisant TESLA, au maximum, les récepteurs doivent attendre la réception du premier paquet d'initialisation, plus le nombre de paquets émis pendant *d* intervalles de temps.
  - L'authentification utilisant FEC requiert, pour son alternative ECU, une latence maximale égale à 2* nombre de paquets dans un bloc. (cf Figure 3)
- Coût en terme de puissance de calcul.
  - "Key Agreement" requiert une opération de cryptage/décryptage de chaque paquet émis par la source, comme la confidentialité des données est également assurée. Les opérations de cryptage et de décryptage se font avec la même clé de session.
  - L'authentification avec TESLA requiert la signature (côté source) et la vérification (côté récepteurs) du premier paquet d'initialisation du protocole. Ensuite, la source calcule une fonction de hachage par paquet. Les performances de MD5 et de HMAC-MD5 selon la taille du bloc, sont détaillés dans le tableau de la figure 4 (Exécution avec Pentium III).

    Pour analyser les performances du schéma d'authentification du flux, on calcule aussi le nombre de paquets par seconde que la source peut créer. Le tableau de la figure 5 présente ce taux de création, selon la taille du paquet.

Taille du Bloc	16	64	256	1024
MD5	256410	169491	72463	22075
HMAC-MD5	75187	65359	39525	17605

Fig. 4: Nombre d'opérations par seconde

Taille de paquet (en octets)	64	256	1024
Nombre de paquets créés par seconde	27677	23009	8148

Fig. 5: Nombre de paquets créés par seconde

  - L'authentification utilisant FEC requiert côté source et pour chaque bloc, *b* opérations de hachage (*b* étant le nombre de paquets par bloc), une signature digitale et 2 opérations de cryptage-décryptage. Du côté des récepteurs, et au minimum, *b* opérations de hachage et une vérification de signature sont aussi requises. Mais dans le cas où il y a eu pertes de données, des opérations additionnelles de décodage seront nécessaires. [Riz97] montre que le coût de décodage dû aux pertes de paquets est en O(m.e.L) avec *m* le nombre de paquets du message initial, *e* le nombre de paquets redondants correspondant aux pertes de données et *L* la taille du paquet.
- Surcoût en terme de bande passante.
  - "Key Agreement" n'a aucun impact sur la bande passante. En effet, l'authentification se fait en dehors de l'envoi des données multicast sécurisées (out of band).

- Pour TESLA, le surcoût en bande passante peut ne pas dépasser 10 octets par paquet. Cependant, selon les fonctions de hachage utilisés, ce nombre peut varier. Par exemple, en utilisant 80 bits HMAC-MD5, on atteint 24 octets par paquets (10 octets pour la clé révélée, 10 octets pour l'information d'authentification MAC et 4 octets pour l'index de l'intervalle) [PCTS00].
- Le surcoût pour la solution utilisant FEC est dépendant de *b* (nombre de paquets par bloc) et *p* (taux de pertes par bloc). Le tableau de la figure 6 présente quelques valeurs de ce surcoût suivant *p* et *b*.

P\b	16	32	64	128	256	512	1024
0.05	10	6	4	2	2	2	1
0.10	12	7	5	3	3	3	2
0.25	16	11	8	7	6	6	6
0.50	32	24	20	18	17	17	17
0.75	80	64	56	56	50	49	49

Fig. 6: Surcoût en octets par paquet pour différentes valeurs de p et b

- Le passage à l'échelle de l'architecture d'authentification : les protocoles de "Key Agreement" dépendent du nombre de récepteurs et ne permettent pas le passage à l'échelle. Les approches utilisant FEC et TESLA sont extensibles car ce sont des approches à flux unidirectionnel et indépendants du nombre de récepteurs.
- Confidentialité : la confidentialité n'est assurée de fait que par les protocoles de "Key Agreement". Elle doit être fournie en plus pour les architectures TESLA et FEC.
- Problématique de la Confiance : les protocoles de "Key Agreement" requièrent pour tous les membres du groupe, la connaissance initiale d'un mot de passe faible pour pouvoir établir la clé de session du groupe. Les approches utilisant TESLA et FEC ont besoin d'une clé privée et publique de la source. Pour cela, on peut ne pas faire appel à une infrastructure à clé publique et utiliser les identificateurs cryptographiques [MC02].
- Synchronisation entre source et récepteurs : Seule l'authentification utilisant TESLA requiert une synchronisation entre source et récepteurs.

5.2 Synthèse et discussions

La définition de ces critères permet de mieux comprendre comment adapter les méthodes d'authentification dans un environnement Ad Hoc et permet de déterminer la méthode d'authentification à choisir en fonction de l'importance de certains critères.

Pour pouvoir utiliser les protocoles de "Key Agreement" dans le cadre des communications de groupes dans les réseaux Ad Hoc à grande échelle, des adaptations s'avèrent nécessaires :

- Pour 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 affects 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 doit être 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. Une solution consisterait à alléger la contrainte de temps limite de synchronisation entre émetteur et récepteurs.

[PM03] propose un schéma d'authentification utilisant FEC, de 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). On peut faire donc appel à l'authentification avec cryptographie à seuil ou plus simplement aux identificateurs cryptographiques.

Ainsi, pour authentifier efficacemment 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 la solution la plus appropriée. 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.

6 Conclusion

Sécuriser des 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 le télé-enseignement présentent plus de vulnérabilité en terme de sécurité que les communications point à point et nécessitent des critères de performances spécifiques quant à la tolérance aux pertes de données, au faible surcoût en communication, au passage à l'échelle de la taille du groupe,...

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

Dans cet article, nous nous sommes focalisés sur le service d'authentification qui permet à un nœud de s'assurer de l'identité des entités avec lesquelles il communique, et plus spécifiquement sur le service d'authentification de la source. Nous avons abouti à une classification des approches d'authentification de la source d'un flux multicast en trois familles : authentification du groupe, authentification des données de la source et authentification individuelle de la source. L'authentification du groupe fournit l'assurance qu'un paquet a été envoyé par un des membres inscrit dans le groupe. L'authentification des données de la source assure qu'un paquet a été envoyé par une source du groupe et non par un récepteur. L'authentification individuelle de la source fournit le niveau le plus élevé d'authentification puisqu'elle assure l'identité de la source de données et sa non répudiation.

Nous avons ensuite évalué les différentes familles de protocoles d'authentification en fonction de critères que nous avons fixés en mettant en évidence les contraintes induites par les communications de groupe dans un environnement Ad Hoc. Cette évaluation nous a permis de déterminer l'adéquation de ces approches dans un contexte Ad Hoc. De cette étude, il apparaît clairement qu'il n'existe pas une solution optimale qui résouderait l'authentification des 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.

Actuellement, nous travaillons sur l'extension des architectures de sécurité proposées pour pouvoir assurer le service de confidentialité. Plus spécifiquement, nous nous focalisons sur l'intégration de mécanisme de gestion de clés afin de sécuriser les flux multicast dans un réseau Ad Hoc.

References

[AG00] N. Asokan and P. Ginzboorg. Key-agreement in ad-hoc networks. *Computer Communications*, 23(17):1627–1637, February 2000.

[ASZ96] D. Atkins, W. Stallings, and P. Zimmermann. RFC 1991 - PGP Message Exchange Formats, August 1996.

[CCF03] T. Cucinotta, G. Cecchetti, and G. Ferraro. Adopting redundancy techniques for multicast stream authentication. In *FTDCS 9th IEEE International Workshop on Future Trends of Distributed Computing Systems*, 2003.

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

[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 Network s, pp. 125-133, Cambridge, MA*, November 2002.

[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.

[LAU03] V. Legrand, F. Abdesselam, and S. Ubéda. Etablissement de la confiance et réseaux ad hoc - un état de l'art. In *SAR'2003*, July 2003.

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

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

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

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

[MRR99] M. Moyer, R. Rao, and P. Rohatgi. A survey of security issues in multicast communications. In *IEEE Network*, pages 12–23, November/December 1999.

[MS01] S. Miner and J. Staddon. Graph-based authentication of digital streams. In *Proceedings of The IEEE Symposium on Research in Security and Privacy*, pages 232–246, May 2001.

[PCS02] J. Park, E. Chong, and H. Siegel. Efficient multicast packet authentication using signature amortization. In *In 2002 IEEE Symposium on Security and Privacy, Berkeley, California*, May 2002.

[PCST01] A. Perrig, R. Canetti, D. Song, and D. Tygar. Efficient and secure source authentication for multicast. In *Network and Distributed System Security Symposium*, San Diego, february 2001.

[PCTS00] A. Perrig, R. Canetti, J. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, pages 56–73, 2000.

[Per01] A. Perrig. The biba one-time signature and broadcast authentication protocol. In *Eighth ACM Conference on Computer and Communication Security, pages 28–37. ACM, November 5–8 2001.*, 2001.

[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.

[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.

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

[RL96] R. Rivest and B. Lampson. SDSI – A simple distributed security infrastructure. Presented at CRYPTO'96 Rumpsession, 1996.

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