PARTICIPANT ACCESS CONTROL IN IP MULTICASTING

Salekul Islam

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Ph.D. in Computer Science

Concordia University

Montréal, Québec, Canada

July 2008

Your file Votre référence
ISBN: 978-0-494-42547-3
Our file Notre référence
ISBN: 978-0-494-42547-3

Abstract

Participant Access Control in IP Multicasting

Salekul Islam, Ph.D.

Concordia University, 2008

IP multicast is best-known for its bandwidth conservation and lower resource utilization. The classical multicast model makes it impossible to restrict access to authorized End Users (EU) or paying receivers and to forward data originated by an authorized sender(s) only. Without an effective participant (i.e., receivers and sender(s)) access control, an adversary may exploit the existing IP multicast model, where a host can join or send any multicast group without prior authentication and authorization. The Authentication, Authorization and Accounting (AAA) protocols are being used successfully, in unicast communication, to control access to network resources. AAA protocols can be used for multicast applications in a similar way. In this thesis, a novel architecture is presented for the use of AAA protocols to manage IP multicast group access control, which enforces authentication, authorization and accounting of group participants. The AAA framework has been deployed by implementing the Network Access Server (NAS) functionalities inside the Access Router (AR). The proposed architecture relates access control with e-commerce communications and policy enforcement.

The Internet Group Management Protocol with Access Control (IGMP-AC), an extended version of the IGMPv3, has been developed for receiver access control. The IGMP-AC, which encapsulates Extensible Authentication Protocol (EAP) packets,

has been modeled in PROMELA, and has also been verified using SPIN. Finally, the security properties of an EAP method, EAP Internet Key Exchange, have been validated using AVISPA. Protocol for Carrying Authentication for Network Access, a link-layer agnostic protocol that encapsulates EAP packets, has been deployed to authenticate a sender that establishes an IPsec Security Association between the sender and the AR to cryptographically authenticate each packet. Next, a policy framework has been designed for specifying and enforcing the access control policy for multicast group participants.

The access control architecture has been extended to support inter-domain multicast groups by deploying Diameter agents that discover network entities located in remote domains and securely transport inter-domain AAA information. Furthermore, the inter-domain data distribution tree has been protected from several attacks generated by a compromised network entity (e.g., router, host) by deploying a Multicast Security Association. Finally, the scope of receiver access control architecture and IGMP-AC has been broadened by demonstrating the usability of IGMP-AC in wireless networks for mobile receiver (or EU) access control. In addition, using the EAP Re-authentication Protocol (ERP), a secured and fast handoff procedure of mobile EUs in wireless networks has been developed.

Acknowledgements

First and foremost, I am expressing my solemn gratitude to Almighty Allah, the most gracious and merciful. Without His help and kindness, I could not have finished my thesis.

I am forever grateful to my supervisor Dr. J. William Atwood, who has given me support from the beginning. His vast experience, immense knowledge and friendly attitude directed me to achieve my research goal. I want to express my heartfelt gratitude for the patience he showed when I was a novice researcher. Specially, I am thankful for the freedom he has provided me in deciding my goals and experimenting with the approaches towards achieving them. He has taught me and demonstrated what it means to be a professor.

I would like to thank the other members of my examination committee, Drs. Maryline Laurent-Maknavicius, Mourad Debbabi, Jaroslav Opatrny and Peter Grogono, for reviewing my thesis, and providing valuable comments and insight into my research. Thanks also to the faculty members, the staff, and my fellow graduate students of the High Speed Protocols Lab of the Computer Science and Software Engineering Department. Special thanks are due to the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) for granting me the Doctoral Research Scholarship, to Concordia University for awarding me several scholarships and to Dr. Atwood for providing me financial assistance.

My deepest gratitude goes to my parents for their unflagging love and support throughout my life. I have no suitable word that can fully describe their everlasting love to me. I am also grateful to my father-in-law, sister, brother and other family members for their appreciation. I must mention my wife, Sarah, for her spontaneous assistance and encouragement. She helped me not only organizing my thoughts, but also improving the text of the thesis. I am fortunate to have her at my side and thank her for everything that she is. Finally, my acknowledgement would never be completed without mentioning our son, Ridwan, who was born in the first year of my PhD program. I promise, in the coming days, I will compensate everything what he missed from his father in the last three years. His ever smiling face is the inspiration and motivation of my life.

The thesis is dedicated to my

Parents, ***Abbu-Ammu***

Wife, ***Sarah*** *&* *Son,* ***Ridwan***

Contents

List of Figures

List of Tables

List of Acronyms

AAA .....Authentication, Authorization and Accounting

AAAS .....AAA Server

AR .....Access Router

ASM .....Any Source Multicast

AVISPA .....Automated Validation Internet Security Protocols and Applications

BR .....Border Router

CP .....Content Provider

CR .....Core Router

DDT .....Data Distribution Tree

DoS .....Denial of Service

DR .....Designated Router

DSRK .....Domain Specific Root Key

EAP .....Extensible Authentication Protocol

EMSK .....Extended Master Session Key

EP .....Enforcement Point

ERP .....EAP Re-Authentication

EU .....End User

GCKS .....Group Controller and Key Server

GKM .....Group Key Management

GO ..... Group Owner

HAAAS ..... Home AAA Server

IETF ..... Internet Engineering Task Force

IGMP ..... Internet Group Management Protocol

IGMP-AC ..... Internet Group Management Protocol with Access Control

IKE ..... Internet Key Exchange

IPsec ..... IP Security

LAAAS ..... Local AAA Server

MitM ..... Man-in-the-Middle

MN ..... Mobile Node

MSA ..... Multicast Security Association

MSEC ..... Multicast Security

MSK ..... Master Session Key (MSK)

NAI ..... Network Access Identifier

NAS ..... Network Access Server

NSP ..... Network Service Provider

PAA ..... PANA Authentication Agent

PaC ..... PANA Client

PANA ..... Protocol for Carrying Authentication for Network Access

PIM-SM ..... Protocol Independent Multicast - Sparse Mode

RFC ..... Request For Comment

SA ..... Security Association

SPIN ..... Simple PROMELA INterpreter

SSM ..... Source-Specific Multicast

XACML ..... eXtensible Access Control Markup Language (XACML)

Chapter 1

Introduction

IP multicast, with all its advantages, is not widely deployed yet, although it was standardized by the IETF [25] many years ago. A number of applications, from different categories, can benefit from the use of multicast. In the Internet, the number of users and various applications are growing rapidly. Many applications, previously available only to a limited number of power users with high-end workstations, are starting to become mainstream applications in the PC world. In last few years, the Internet speed has increased tremendously. Even a regular PC user is enjoying Internet speed of several megabits by paying only $30/month. The price is reducing day by day and becoming affordable for more users. The increased Internet speed has shifted the End Users' (EU) interest. They are no longer satisfied by browsing different sites, exchanging emails and communicating through text chat. A human likes to interact with other humans, and the Internet has turned into the best meeting place. It has created the always "connected" platform and changed the world to a small place. Therefore, sites like YouTube and MySpace are attracting users more than any other site.

Bandwidth-intensive applications including audio and video streaming, Voice over IP (VoIP), online games with high-resolution graphics and video-conference are be-

coming popular and mainstream applications. Many of the new applications (e.g., Internet TV, multi-player games, distance learning, video-conferencing) rely on one-to-many or many-to-many communications [94], where one or more sources are sending data to multiple receivers. The Network Service Providers (NSP) could save a significant amount of bandwidth by using IP multicast for these types of applications. It is an efficient technology to deliver data to many recipients who may be geographically scattered in a wide area. For each recipient, multicast replicates data at a point as close to that recipient as possible. Thus, it is considered as a bandwidth conservation technique with respect to unicast, specially, when there are many recipients and a large amount of data (e.g., streaming video) is being sent. However, multicast is not widely deployed yet due to its lack of security and control over multicast groups. The present IP multicast architecture has created a circular cause and consequences phenomena: the NSPs are not supporting multicast as there are no applications, and multicast applications are not being developed due to lack of support. Revenue generation from the EUs is the only way to break this cycle. The Authentication, Authorization and Accounting (AAA) protocols [85] are being used very successfully to ensure revenue generation by controlling access to network resources in unicast communication. AAA protocols can be used for multicast based applications to authenticate the EU, and to establish their authority to participate in the group.

1.1 IP Multicasting

The extensions required for a host implementation of the Internet Protocol (IP) to support multicasting were first specified in RFC1112 [25]. Unlike simple peer-to-peer unicast communication, m-to-n communication should be considered in multicasting. To implement multicasting successfully, we can identify four major steps or processes responsible for the whole complex tasks. Figure 1 describes the schematic diagram

of these four processes.

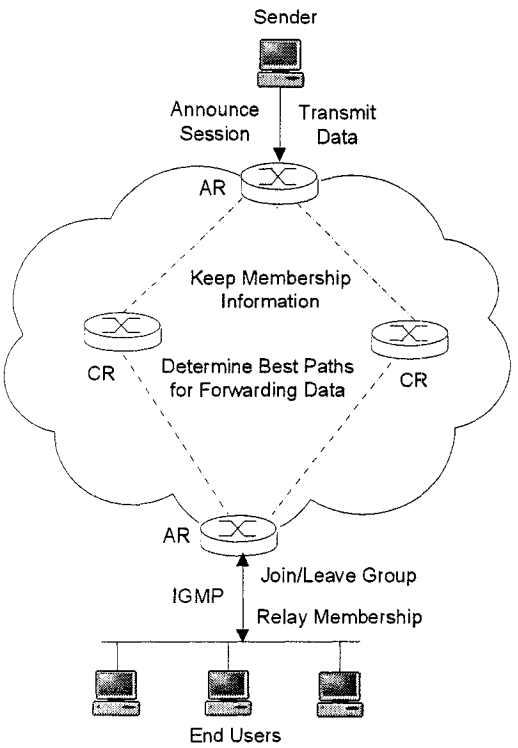

Figure 1: Basic Components of IP Multicasting

1. At first, a multicast host group having a class D multicast address is created by the group owner and this group address is announced to the potential receivers or EUs.

2. An EU or host will communicate with its local Access Router (AR) to join/leave a multicast group, using Internet Group Management Protocol (IGMPv3) [17] in IPv4 or Multicast Listener Discovery (MLDv2) protocol [110] in IPv6.

3. The third component is the multicast routing protocol (e.g., Protocol Independent Multicast-Sparse Mode (PIM-SM) [32]). Several such protocols have been specially formulated to support multicasting. The Core Routers (CR) will use one of the routing protocols to build the Data Distribution Tree (DDT).

4. Finally, there are application protocols for creating and managing the multicast data that are distributed in a multicast session.

1.2 Multicast-based Applications

An application is classified as a "multicast application" when it sends to and/or receives from an IP multicast address. Then again, depending on the number of sender(s) or source(s) there are two types of multicast service models:

Source-Specific Multicast (SSM) service model delivers a datagram originated by a source, `S` with destination address, `G` only to those end hosts who have explicitly joined an SSM group by sending a source-specific or `(S, G)` join message [47].

Any Source Multicast (ASM) service model delivers a datagram originated by any source with destination address, `G` only to those end hosts who have joined an ASM group by sending a `(*, G)` join message.

From the IP multicast address block (224.0.0.0 through 239.255.255.255), the Internet Assigned Numbers Authority (IANA) [53] has designated IPv4 addresses in the 232/8 (232.0.0.0 to 232.255.255.255) range as Source-Specific Multicast (SSM) destination addresses and has reserved them for use by source-specific applications and protocols only.

There are three general categories of multicast applications depending on the number of sender(s) and receiver (s) [94]:

One-to-Many (1toM) applications have a single sender and multiple receivers. A 1toM application typically uses the SSM service model and is suitable for audio/video streaming applications (e.g., Internet TV).

Many-to-Many (MtoM) applications have more than one senders and receivers. An ASM service model should be deployed for an MtoM application. These

types of applications efficiently provide simultaneous communication among more than one users.

Many-to-One (Mto1) applications have multiple senders and one (or a few) receiver(s). These types of applications are not very common, however, we have listed some of them in the following.

In Table 1, we have listed some popular applications, which could benefit by the deployment of IP multicast.

Table 1: Taxonomy of Multicast-based Applications

Number of participants	**Applications**
One-to-many	• Scheduled audio/video distribution • Push media: news headlines, weather updates • File distribution and caching • Announcements: multicast session, key updates • Monitoring: stock prices, sensor equipment
Many-to-many	• Multimedia conferencing • Synchronized resources • Distance learning with input from receivers • Multi-player games • Chat groups
Many-to-one	• Resource discovery • Auctions • Polling • Jukebox

1.3 Secure Multicasting

Multicasting has suffered from various security breaches from the very beginning of its evolution. We cannot expect large-scale commercial deployment of multicast applications without ensuring security of multicasting in every aspect. Secured multicast is a complex issue to achieve. Depending on the application, it may not be worth the cost. Nevertheless, there is a large set of applications where establishing the security is essential. There are two orthogonal criteria to satisfy for making multicast communication concrete against all sorts of attacks. These are as follows:

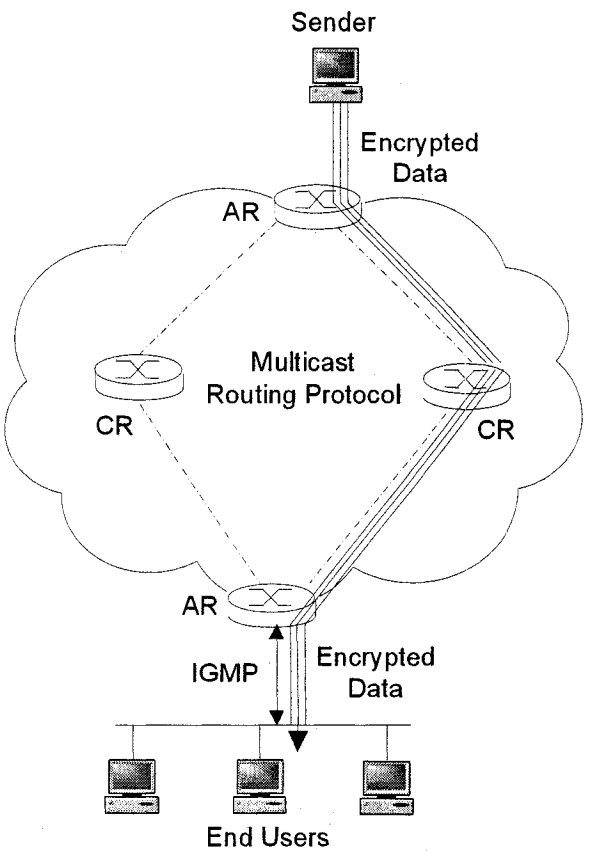

Figure 2: Control Messages for Multicast Communication

1. **Protecting Control Messages:** As shown in Figure 2, a host computer (sender or receiver) communicates with its nearest router, which is also known as the Designated Router (DR) of that domain. A host uses the Internet Group

Management Protocol (IGMP) [17] to join or leave a multicast group. The routers use one of the multicast routing protocols to maintain or update their own routing tables. In this way, the DDT is built. PIM-SM, DVMRP, MOSPF, and CBT are some examples of multicast routing protocols. There are two types of control messages: the messages exchanged between a host and the DR and the messages exchanged by the multicast enabled routers. All these control messages must be protected. Security issues of a multicast routing protocol are local issues and they depend on the protocol itself. We cannot expect any general solution for this. One such example is the security issue of PIM-SM link-local messages [8].

2. **Protecting Multicast Data:** Once the group has been set up, the sender will start sending data using the distribution tree. In a secured group, only the legitimate group members will be able to join the group. The sender will send data in encrypted format and the receivers should have the appropriate key to decrypt it. To protect multicast data, different issues such as registration of group members, key generation, key distribution and key management should be considered. The multicast Security (MSEC) [89] Working Group at the IETF is doing extensive research in this area. They have already developed the Security Architecture [40] for multicast communication and are now concentrating on other issues.

1.4 Motivation for the Research: Creation of a Revenue Generation Architecture

It is evident that in spite of significant progress in secure multicasting, large scale deployment of multicast has not materialized yet. This fact suggests the existence of a hole in the security of multicasting. The key to establishing a revenue stream from the

multicast clients lies in authenticating the end users, and establishing their authority to participate in the group. Prior to delivering any service to a client, the client's ability to pay for it should be checked. This implies that there has to be a strong relationship to AAA functions, and various e-commerce interactions. The solution must be fully distributed to handle hundreds or thousands of participants, who may be geographically scattered over a wide area. An overall architecture for secure and accountable multicasting [9] should address not only protection of multicast data and control messages but also distributed access control and necessary e-commerce components. Therefore a revenue generation architecture should satisfy the following two issues in addition to secure multicasting:

1. **Participant Access Control:** Access control encompasses all the AAA [85] functionalities: authentication, authorization and accounting. Therefore, a receiver (sender) must be authenticated before being allowed to receive (send) any data. Authorization defines the rights and services the authenticated receiver or sender is allowed to have. This implies that successful authentication is a prerequisite for authorization. The lack of accounting capability is one of the major reasons that multicast based applications are well behind in comparison with those based on unicast.

2. **E-commerce Technologies:** Establishing an e-commerce transaction for a multicast data stream is inherently more difficult than establishing an e-commerce transaction for unicast delivery. This implies that considerable adaption of unicast e-commerce protocols will be necessary to meet the accounting and revenue generation goals.

In this thesis, we have devoted our research to developing an architecture to extend the existing IP multicast service model with participant (both sender and receiver) access control. The architecture we have developed will facilitate all three AAA functionalities—Authentication, Authorization and Accounting—for both sender(s)

and receivers of a multicast group. The e-commerce issues are not an integral part of our research. However, we acknowledge it as an open issue for future research.

1.5 Thesis Organization

The thesis is composed of eleven chapters. In the following we have briefly explained the contents of each chapter.

Chapter 1 introduces the basic service model of IP multicasting, lists some common applications that use multicast and then describes secure multicasting. The motivation of our research has been also illustrated.

Chapter 2 helps the reader to develop background knowledge in basic security features of multicasting, IGMP protocol, AAA protocol, Extensible Authentication Protocol (EAP) and Protocol for carrying Authentication for Network Access (PANA).

Chapter 3 starts with the definition of the problem. Next, the access control architecture that we have designed is presented with the role of involved entities and the different events of the architecture.

Chapter 4 presents the Internet Group Management Protocol with Access Control (IGMP-AC) that we have designed by extending the IGMPv3 protocol to provide receiver access control. IGMP-AC is explained using state-diagrams and by explaining the roles of different entities. Furthermore, the IGMP-AC has been modeled in PROMELA and the model has been verified using SPIN.

Chapter 5 illustrates the encapsulation procedure of EAP packets over IGMP-AC to support a wide range of methods to authenticate EU, such as password, shared secret, Public Key Infrastructure, etc. The encapsulation technique is demonstrated by an example EAP method, EAP Internet Key Exchange (EAP-IKEv2). In addition, the security properties of the EAP-IKEv2 method have been validated using AVISPA.

Chapter 6 presents the proposed sender access control architecture, which deploys PANA for sender authentication and setting up a Security Association (SA) to cryptographically authenticate each multicast packet at the AR.

Chapter 7 describes the policy framework that has been designed to facilitate the access control architecture with a flexible and scalable solution to enforce access control policy. The policy framework recommends the use of eXtensible Access Control Markup Language (XACML) for policy specification, and Security Assertion Markup Language (SAML) for policy transportation.

Chapter 8 presents the inter-domain access control architecture. The receiver and sender access control architectures have been developed independently. Next, two different methods—centralized and distributed—that deploy Multicast Security Association (MSA) for inter-domain data distribution control have been explained.

Chapter 9 extends the receiver access control architecture to mobile wireless networks, and presents a secured handoff technique with low join latency for mobile EUs by encapsulating EAP Re-authentication (ERP) packets over IGMP-AC.

Chapter 10 compares different aspects of the proposed architecture with the state of the art.

Chapter 11 summarizes the contributions of the research, briefly discusses the impacts of the research, and concludes the thesis with a list of work that has been planned for the future.

Chapter 2

Useful Background Knowledge

The problem we are dealing in our research is complex and distributed in different protocols and Working Groups' materials. In this chapter, we will briefly explain the frameworks and the protocols that we have used in our research. First, we need a clear understanding of the MSEC [89] Working Group's goals and milestones, what they have done so far, and the direction in which they are going. Specifically, from their reference framework it will be clear that they are not dealing with authentication, authorization or accounting of receivers/sender(s). The Internet Group Management Protocol (IGMP) [17] plays an important role in the communication between the EU and the network AR. Therefore, we have also summarized the IGMPv3 protocol here. Our goal is to use the AAA [85] framework for authentication and authorization of secured groups' participants. We are interested in the accounting issue also. Next, we are interested in incorporating a generic authentication framework in our proposed framework to enhance our framework with a variety of authentication schemes. The IETF has developed the Extensible Authentication Protocol (EAP) [2], a universal authentication framework, which is being used for EU authentication in wireless networks, PPP and wired LAN. Finally, our proposed framework would deploy Protocol for carrying Authentication for Network Access (PANA) [62] for sender access control.

2.1 Group Security Architecture for Multicast

The Group Security Architecture [40] has been designed for multicast communication and is independent of any multicast routing protocol (e.g., PIM-SM, MOSPF) or admission control protocol (e.g., IGMP, MLD). It is an architectural overview that outlines the security services required to secure a large multicast group. More specifically, by deploying this architecture, we can protect the data of a large multicast group. Although the target of this framework is to protect multicast data, this does not preclude the use of the framework for a specific multicast routing protocol to protect control messages, if possible.

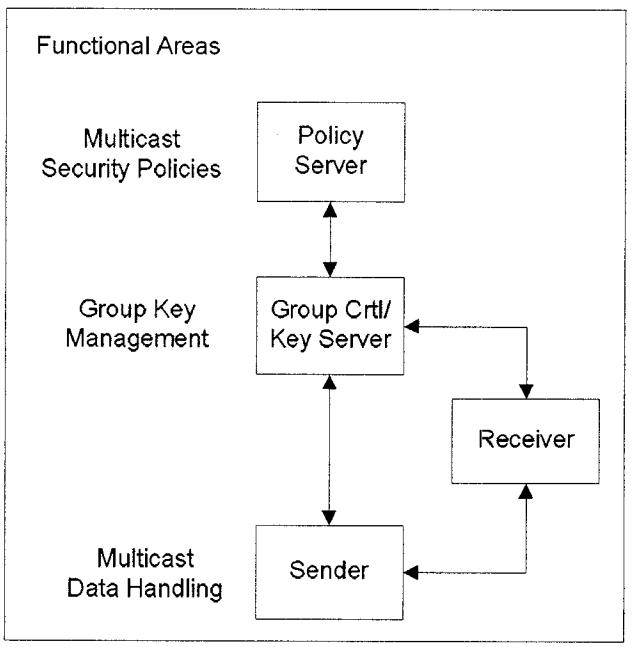

Figure 3: Centralized Multicast Security Reference Framework

The framework provides a Reference Framework (shown in Figure 3) for organizing the various elements within the architecture, and explains the elements of the Reference Framework. The Reference Framework organizes the elements of the architecture into three Functional Areas pertaining to security. These elements cover the treatment of data when they are to be sent to a group, the management of keying material used to protect the data, and the policies governing a group. Another im-

portant item in this document is the definition and explanation of a Group Security Association (GSA), which is the multicast counterpart of the unicast Security Association (SA) [69]. The GSA is specific to multicast security, and is the foundation of the work on group key management.

2.1.1 Reference Framework

The Reference Framework depicts the main entities and functions relating to multicast security, and identifies the inter-relations among them. In Figure 3, a "box" should be interpreted loosely as a given function related to a functional area. The protocols to be standardized are represented in the Reference Framework diagrams by the arrows that connect the various boxes. These arrows are the interfaces between the boxes.

The **Group Controller and Key Server (GCKS)** represent both the entity and the functions relating to the issuance and management of cryptographic keys used by a multicast group.

The **Sender** and the **Receivers** will interact with the GCKS for authentication, authorization, and key management purposes. Key management includes obtaining the new keys and the key updating materials.

The **Policy Server** represents both the entity and the functions used to create and manage security policies specific to a multicast group.

2.1.2 Functional Areas

The Reference Framework identifies three functional areas.

Multicast Data Handling covers the security-related treatments of multicast data

by the sender and the receiver. In a secure multicast group, the data should be:

1. Encrypted using the group key for access control and confidentiality.
2. Authenticated, for verifying the source and integrity of the data. There are two types of authentications: source authentication and group authentication.

Group Key Management is concerned with the secure distribution and refreshment of keying materials. It includes the cryptographic keys belonging to a group, the states associated with the keys, and the other security parameters related to the keys. Hence, the management of the cryptographic keys belonging to a group necessarily requires the management of their associated states and parameters.

Multicast Security Policies must provide the rules to operate the other elements of the Reference Framework. Security policies may be distributed in ad-hoc fashion in some instances. However, better coordination and a higher level of assurance are achieved if a Policy Controller distributes security policies to the group.

2.1.3 Remarks

We can add the following remarks on the MSEC Framework:

- The framework has been designed to protect multicast data from many forms of attack. It is a complete solution for the multicast data plane and has not mentioned anything about the multicast control plane or how to protect multicast routing protocol messages.
- Participant access control (i.e., authentication, authorization and accounting of sender and receivers) is absent in this framework. An immense amount of work needs to be done by the research community to solve this issue.

- The e-commerce issues are missing in this framework. If we expect large-scale deployment of multicast based applications, the NSPs and the Group Owners (GO) will interact with the users. The users' credits should be verified before granting them access to a service.

- Finally, the functionalities of the Policy Server are not standardized yet. For example, authorization of a user should be explicitly specified through the policy server, but how it will be specified is not well defined.

2.2 Internet Group Management Protocol (IGMPv3)

Internet Group Management Protocol version 3 (IGMPv3) [17] has been standardized by the IETF for IPv4 end systems to inform the neighboring router(s) about the multicast group memberships of these systems. IGMPv3 supports "source filtering" through which a system can request to receive multicast packets *only* from a specific list of sources, or from *all but* specific sources, sent to a particular multicast group. Three main operations performed by IGMP are:

- A system (host or router) sends a *join message* when it wants to join a multicast group or some specific sources of a group.

- A system sends a *leave message* when it wants to unsubscribe from a multicast group.

- A router sends a *query message* periodically to check which multicast groups are of interest to the hosts that are directly connected to that router.

2.2.1 Multicast Reception States Maintained by Systems

There are two types of reception states maintained by the systems: socket state and interface state.

Socket state is used to keep a record of the desired multicast reception state for each socket on which the upper layer Application Program Interface (API) has been invoked. This state consists of a set of records of the form:

(interface, multicast-address, filter-mode, source-list)

Interface state is maintained in addition to socket state for each of the interfaces of a system. This state consists of a set of records of the form:

(multicast-address, filter-mode, source-list)

For a given interface, at most one record per multicast address is stored.

2.2.2 Message Formats

There are two types of IGMPv3 messages: Membership Query Message and Version 3 Membership Report Message. These messages are encapsulated in IPv4 datagrams and are always sent with Time To Live (TTL) = 1.

2.2.2.1 Membership Query Message

A query is sent by the IP multicast routers to update the multicast reception states of neighboring interfaces. It has the format as shown in Figure 4.

There are three types of queries:

1. A **General Query** will be sent with both the Group Address and the Number

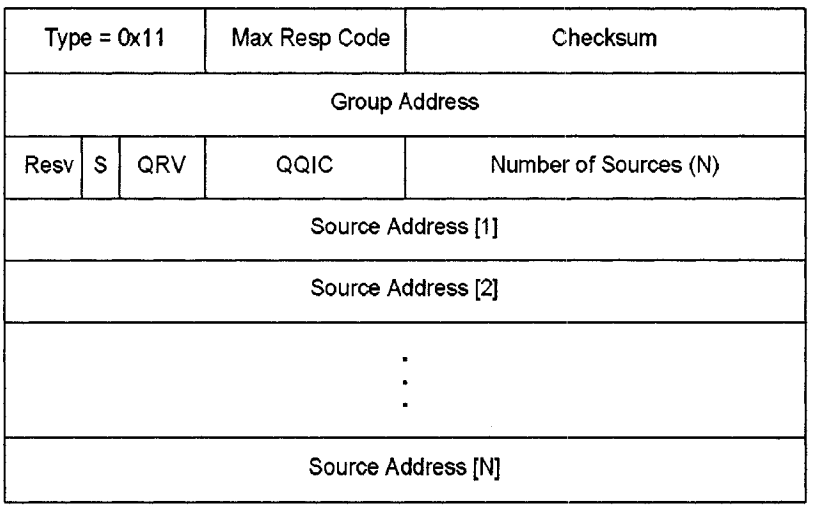

Figure 4: Membership Query Message Format

of Sources (N) fields equal to zero. It will give the complete multicast reception state to the multicast router.

2. A **Group-Specific Query** will inform the router of the reception state with respect to a *single* multicast address. The Group Address field will carry the multicast address of interest and the Number of Sources (N) field will be zero.

3. If the router wants to learn the reception state with respect to a list of sources of a specific multicast group, it will send a **Group-and-Source-Specific Query**, where the Group Address field will carry the multicast address of interest and the Source Address [i] fields will contain the source address(es) of interest.

2.2.2.2 Version 3 Membership Report Message

A system will send a Membership Report Message to the neighboring routers to inform them of its present member reception state or changes in the multicast reception state. Its format is shown in Figure 5, where each Group record has the format as shown in Figure 6.

There are three types of Group Records: Current State Record, Filter-Mode-

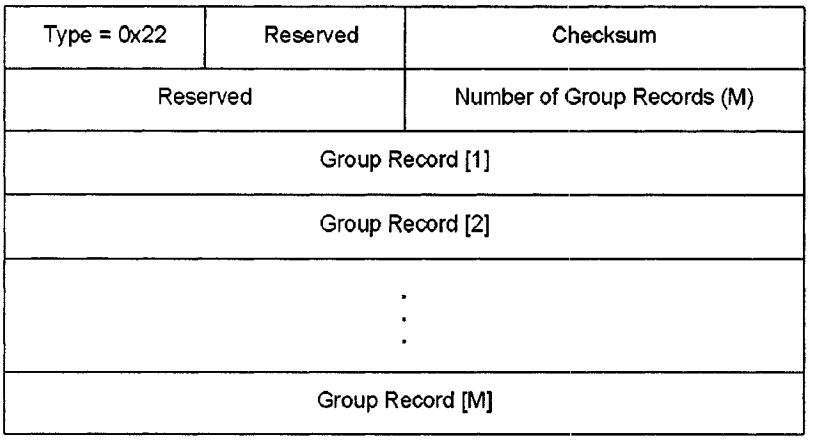

Figure 5: Membership Report Message Format

Change Record and Source-List-Change Record. In a Group Record, the Aux Data Len field contains the length of the Auxiliary Data field in 32-bit words. IGMPv3 does not use any Auxiliary Data and always sends zero for the Aux Data Len.

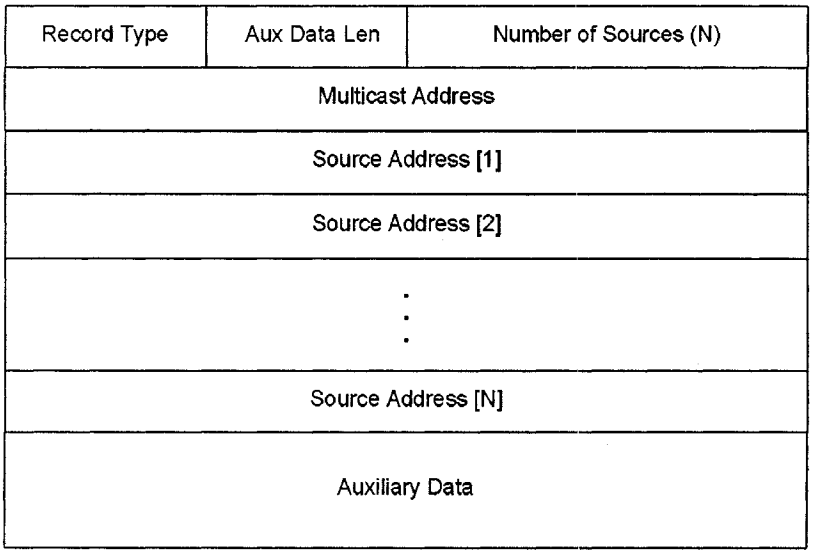

Figure 6: Group Record Format

It should be noted that the Multicast Listener Discovery (MLDv2) protocol [110] provides similar functionalities for IPv6 systems that IGMPv3 does for IPv4. Our research is presented in terms of IPv4 operation, and the IGMP with Access Control (IGMP-AC) protocol (see Chapter 4), an extended version of IGMPv3 to provide

receiver access control that we have designed for IPv4 systems. However, with little effort, a similar extension to MLD could be done to provide receiver access control for IPv6 systems.

2.3 Authentication Authorization and Accounting (AAA)

The term "AAA (Triple-A)" stands for Authentication, Authorization and Accounting [85]. AAA is a client-server based protocol, where a Network Access Server (NAS) acts as a client and communicates with the AAA Server (AAAS). The server maintains the database of user profiles and configuration data, and provides distributed services to NASs. The AAA framework provides primarily the following three services:

1. **Authentication** means validating the end user identity before permitting his access to network. The End User (EU) must possess a unique piece of identity, such as username/password combination, a secret key or may be biometric data. The same identity must be shared with the AAAS so that the server can verify the identity of the EU.

2. **Authorization** defines the rights and services the EU is allowed to have once network access is granted. Not all the authenticated EUs are authorized for every service. Some examples of authorization are IP addresses and invoking a filter to determine which applications or protocols are supported.

3. **Accounting** provides the methodology for collecting information about the EU's resource consumption. This is processed for billing, auditing, and capacity-planning purposes.

2.3.1 Components of the AAA Framework

Figure 7 illustrates the components of the AAA Framework. The AAAS—multiple servers can be used for resiliency—is attached to the network and it serves as a central repository for storing and distributing AAA information. The device acting as the point of entry into the network is typically a NAS (although it could also be a router, a terminal server, or perhaps another host) that contains an AAA client function. AAA processing can be summarized in the following steps:

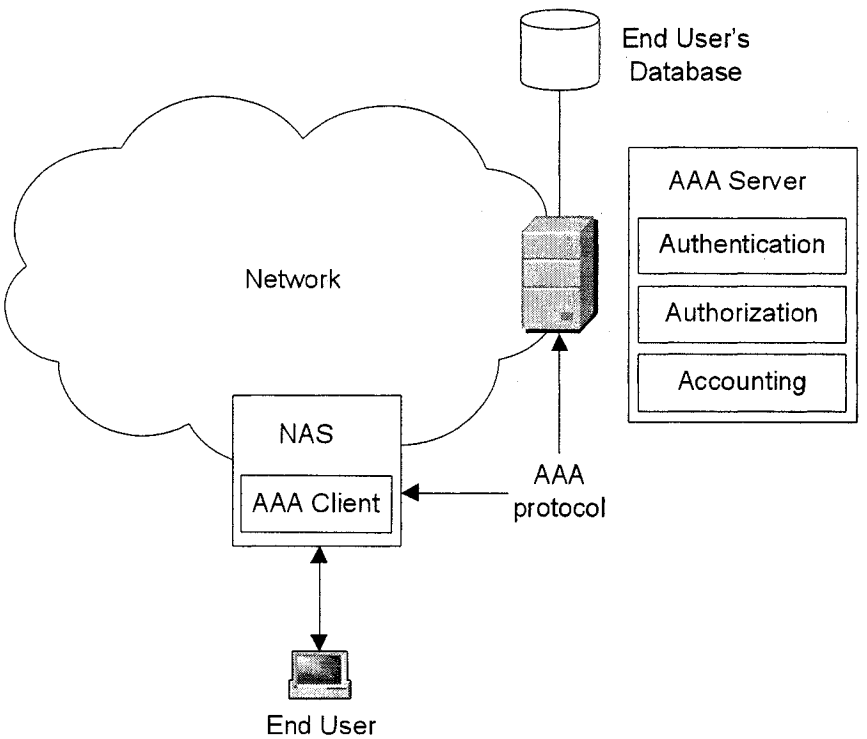

Figure 7: AAA Architecture

1. The EU connects to the point-of-entry device and requests access to the network.
2. Inside the NAS, the AAA client function collects and forwards the EU's credentials to the AAAS.
3. The AAAS processes the data and returns an "accept" or "reject" response and

other relevant data (e.g., authorization information for successful authentication) to the AAA client.

4. The AAA client on the NAS notifies the EU that access is granted or denied for the specified resources.

The NAS may also send an accounting message to the AAAS during connection setup and termination for record collection and storage.

2.3.2 RADIUS

The most widely deployed AAA protocol is Remote Authentication Dial in User Service (RADIUS) [96]. It was developed in the mid '90s by Livingston Enterprises to provide authentication and accounting services to their NAS devices. The IETF formalized that effort in 1996 with the RADIUS Working Group. Key features of this protocol are:

Client/Server Model: A NAS operates as a RADIUS client, which is responsible for passing user information to a designated RADIUS server, and then acts on the response. RADIUS servers are responsible for receiving user connection requests, authenticating the user, and then returning all configuration information necessary for the client to deliver service to the user.

Network Security: Transactions between the client and the RADIUS server are authenticated through the use of a shared secret, which is never sent over the network. In addition, any user's password is sent in encrypted format between the client and the RADIUS server to eliminate the possibility that someone snooping on an insecure network could determine a user's password.

Flexible Authentication Mechanisms: The RADIUS server can support a variety of methods to authenticate a user. When it is provided with the user name

and the original password given by the user, it can support PPP (Point to Point Protocol), PAP or CHAP, UNIX login, and other authentication mechanisms.

Extensible Protocol: All transactions are comprised of variable length Attribute-Length-Value 3-tuples. New attribute values can be added without disrupting existing implementations of the protocol.

2.3.3 Diameter

RADIUS was initially developed for dial-up PPP and terminal server access. Over the years, the Internet has moved forward and many advanced technologies, including wireless, DSL, Mobile IP and Ethernet have evolved. As a result, the task of the NAS has become very complex and dense, and RADIUS is not capable enough to deal with all these issues. The next generation AAA protocol is called Diameter [18]. The architecture of this protocol is shown in Figure 8. The Diameter base protocol provides the minimum requirements needed for a AAA protocol. The base protocol is mandatory to implement, and all other extensions and applications (i.e., NASREQ [19], Mobile IP [20] and Diameter-EAP [29]) will be implemented on top of it. In practice, the base protocol is used with a Diameter application, however, it might be used alone only for accounting purposes. Diameter is a peer-to-peer protocol (any node can initiate a request) with three types of nodes: NAS, agents and AAAS. The use of the Diameter agents will be discussed in detail later when we will present our inter-domain access control architecture in Chapter 8.

2.3.3.1 Diameter AVPs

The Diameter messages consist of a Diameter header followed by one or more Attribute-Value-Pairs (AVPs). An AVP message has its own header and data part. Diameter AVPs carry specific authentication, authorization, accounting, routing and security

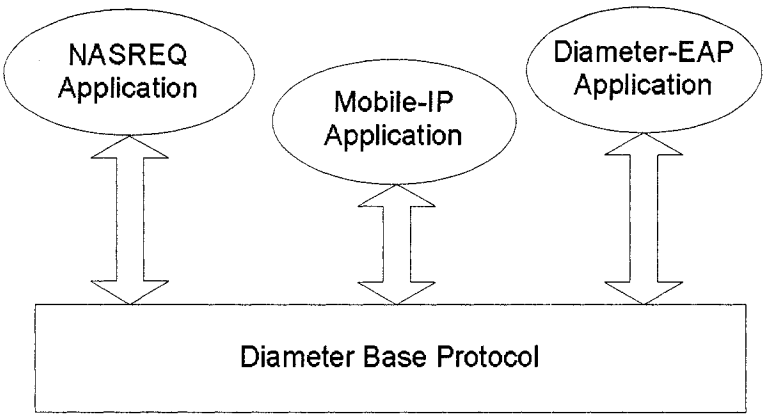

Figure 8: Diameter Protocol Architecture

information, and configuration details. Diameter is considered an easily extensible protocol due to its provision of defining new applications and AVPs. However, in the Diameter specification [18], if possible, it is strongly recommended to reuse the existing AVPs. In order to allocate a new AVP code, a request has to be sent to IANA [53] to assign the requesting value of that AVP code.

2.3.3.2 Key Features of Diameter

Diameter has been developed to meet various demands of the existing heterogeneous networks. It has been designed carefully not to face the difficulties its predecessor, RADIUS [96], faced after being used many years. Moreover, it is expected that the use of AAA protocols would not be limited to network access only. The following are some important key features that demonstrate the advantages of Diameter over RADIUS:

Large AVP space: RADIUS allocates one byte for the length field of its attribute header, which allows for 256 different AVPs only. Diameter resolves this by allocating 4 bytes to the AVP space, which allows for 2^{32} AVPs per vendor.

Failover mechanism: RADIUS dose not specify any failover mechanism and thus,

the failover mechanism differs among implementations. However, Diameter supports application-layer acknowledgements, and defines a failover algorithm and the associated state machines.

Reliability: RADIUS runs over UDP, and does not specify any retransmission behavior. In Diameter, reliability is provided by the underlying transport protocol, such as Transmission Control Protocol (TCP) or Stream Control Transport Protocol (SCTP).

Security features: Diameter has specified security as a mandatory requirement. The Diameter protocol employs either IPsec [69] or Transport Layer Security (TLS) [26] for hop-by-hop integrity and confidentiality between two Diameter peers. End-to-end security is provided via the end-to-end security extension, described in [21].

Agent support: RADIUS does not support different agents, including Proxies, Relays and Redirects. Diameter agents provide a great variety of routing and forwarding facilities that help in deploying inter-domain AAA functionalities.

Server-initiated messages: This feature is optional for a RADIUS implementation, however, it is mandatory in Diameter. It adds some extra functionalities, such as unsolicited disconnect, re-authentication on demand, etc.

2.4 Extensible Authentication Protocol (EAP)

The Extensible Authentication Protocol (EAP) supports multiple authentication mechanisms. It has been implemented between a host and a router connected via switched circuit or dial-up line that uses PPP [103]. It has also been implemented between a switch and an access point that uses IEEE 802 [52]. The EAP runs between an authenticator and a peer, where the authenticator can act as a pass-through, and

a back-end authentication server (or EAP server) may be connected with the authenticator. In this case, the actual authentication will be performed by the back-end server. The pass-through authenticator is nothing but a NAS and the back-end server is an AAAS. In such an environment, the EAP will be used by the EU (or host) and the NAS, and one of the AAA protocols will be used by the NAS and the AAAS. The EAP packets that arrive at the NAS are sent to the AAAS by encapsulating them inside the AAA packets, and the NAS will decapsulate the AAA packets received from the AAAS and forward the EAP packets to the EU. The Diameter EAP application that carries the EAP packets inside the Diameter packets between a NAS (EAP authenticator) and an AAAS (back-end authentication server) is already standardized by the IETF in RFC 4072 [29].

2.4.1 EAP Messages

There are four types of EAP messages: Request, Response, Success and Failure. The Request is always sent by the authenticator to the peer, and the peer replies to it by sending a Response to the authenticator. The sequence of different messages between the NAS and the EU, and between the NAS and the AAAS is shown in Figure 9. The EAP Request and Response messages are sent inside the Diameter messages. Depending on the authentication method used, more than one round-trip may be required. In this scenario, after N round-trips, the AAAS authenticates the EU and sends an EAP Success message inside an EAP-Payload. If authorization is requested appropriate authorization AVPs are sent also.

2.4.2 EAP Methods

EAP has not been developed for a specific authentication mechanism. It is an authentication framework to provide some common functions and a negotiation of the

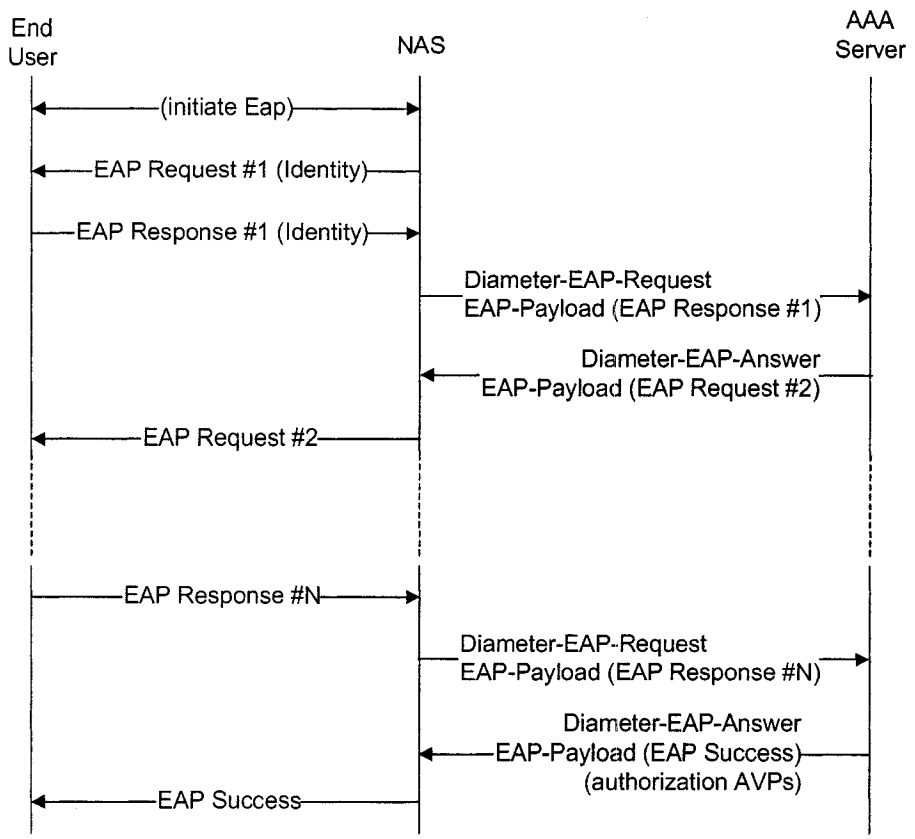

Figure 9: EAP and Diameter Messages

desired authentication mechanism. Such a mechanism is called an EAP method, and there are currently about 40 different EAP methods. Although EAP is standardized in RFC 3748 [2], the defined specification only supports MD5-Challenge, One-Time Password (OTP) and Generic Token Card (GTC) based authentication. All these methods are commonly known as "legacy methods". These methods have some shortcomings: support one way authentication only, no key derivation is possible, vulnerable to some known attacks, etc. To attain the present needs, a number of EAP methods have been developed, which have better security properties and provide more flexibility. The IETF has already standardized some of the methods, such as EAP-IKEv2 [107], EAP-POTP [91], EAP-PSK [15], EAP-AKA [5], EAP-SIM [42], etc. Among these methods, the last two have been specially developed to be used in wireless networks. Each EAP method defines its own messages, which are sent inside EAP packets. However, EAP does not run directly over the IP layer, and an EAP

lower-layer is required. In the following section, we will describe a protocol that has been developed to be used as an EAP lower-layer protocol.

2.5 Protocol for carrying Authentication for Network Access (PANA)

Protocol for Carrying Authentication for Network Access (PANA) [33], a network access authentication protocol, is standardized by the IETF to be deployed as an EAP lower-layer. PANA carries EAP authentication methods encapsulated inside EAP packets between a client node and a server in the access network.

2.5.1 PANA Architecture

The PANA framework [62], comprised of four functional entities, is shown in Figure 10. The PANA Client (PaC) resides on a requesting node, such as an end host, laptop, PDA, desktop PC, etc. It is connected to a network via a wired or wireless interface. It interacts with the PANA Authentication Agent (PAA) in the authentication process using the PANA protocol [33]. The server implementation of PANA is the PAA, which consults an Authentication Server (AS) for authentication and authorization of a PaC. The AS may reside in the same node as the PAA, or it may be a separate entity, in which case, one of the AAA protocols (e.g., Diameter) will be used for their communication. The PAA resides on a node that is typically called a NAS in the access network. The AS is a conventional backend AAAS that terminates the EAP and the EAP methods.

The Enforcement Point (EP) is the access control implementation that allows (blocks) data traffic of authorized (unauthorized) clients. An EP is updated with

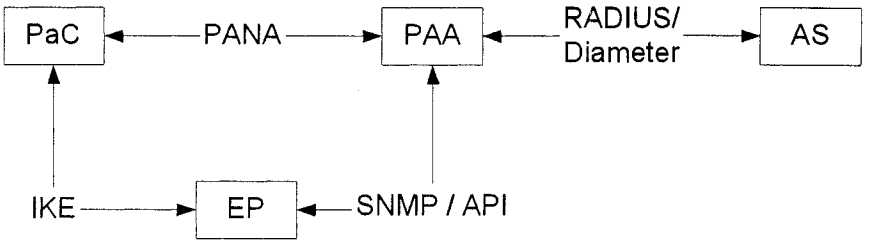

Figure 10: PANA Framework

the attributes of the authorized clients from the PAA. When the PAA and EP reside on the same node, they use an API for communication, otherwise, a protocol (e.g., SNMP) is required.

2.5.2 Access Control using PANA

When the EP uses cryptographic (may use non-cryptographic also) filters for allowing or blocking data packets from a client, a secure association protocol (e.g., IKEv2 [66]) is required to run between the PaC and the EP. If the desired security is to be established at the IP-layer, an IPsec Security Association (SA) [69] is established between these two entities to enforce per-packet cryptographic protection [92]. Depending on the type of SA (i.e., Authentication Header (AH) [70] or Encapsulating Security Payload (ESP) [71]) established, the cryptographic protection may provide integrity protection, data origin authentication, replay protection and optional confidentiality protection.

Chapter 3

Access Control Architecture

We will present our research objectives and the detailed description of the proposed architecture in this chapter. First, the classification of multicast groups that we have introduced will be presented. Next, the security threats that the existing multicast architecture is facing in the absence of access control will be explained. There are two fronts of attacks: through forged IGMP report messages and from an adversary playing the role of a legitimate sender. How these attacks will be solved by building the access control architecture will be explained. We will also illustrate why group key management protocols fail to provide necessary access control for multicast communication. Moreover, the relation between access control and e-commerce communication will be presented. Next, the proposed access control architecture with a set of requirements will be presented. The role of related entities and different events of the architecture will also be elaborately explained. Finally, we will present the direction of the thesis by outlining the scope of our research.

The architecture we have presented in this chapter is limited to a single domain or network, which is a restrictive assumption. The participants (e.g., sender and EUs) may be distributed over different domains in reality. We will discuss this issue further in Chapter 8, where we have extended the proposed architecture to multi-

domain multicast groups, where the participants are distributed over more than one domain.

3.1 Classification of Multicast Groups

The classical IP multicast service model [25] allows any end host to join a multicast group by sending an IGMP message. This message does not carry the identity of the end host/EU. Given that there exists no sender join mechanism, anyone can send multicast data to a multicast group. Hence, the sender(s) and the receivers remain anonymous. The only information that is divulged to the receivers is the sender address in SSM group communication. It is reasonably established that the existing IP multicast service model must be extended to provide participant access control. However, the access control mechanism must be optional in addition to the existing service model. Hence, we have classified multicast groups into two categories: *Open group* and *Secured group*. An Open group is analogous to the exiting multicast service model to which a host can join/leave any time by sending regular IGMPv3 messages and any end system can send multicast data without explicitly joining the group. A Secured group will be restricted to the authenticated and authorized group participants (i.e., sender(s) and receivers) only. The access control mechanisms presented in this thesis will be enforced for the Secured group participants. For successful implementation and co-existence of such a classification of Open and Secured groups, a simple mechanism is required for the ARs and the CRs to distinguish between two types of groups. The easiest and most efficient way would be to delegate a range of class D multicast group addresses (which should be assigned by the IANA [53]) for Secured group operations only. However, our work does not depend on this assumption.

3.2 Problem Definition

The multicast Open group is susceptible to numerous attacks from different adversaries. Although, researchers have been devoting much effort to the development of secured group communication (using group key management protocols) and secured routing protocols, their efforts fail to provide necessary functionalities for participant access control.

3.2.1 Effects of Forged IGMP Report Message

A multicast routing protocol builds the DDT among the CRs, while the ARs extend that tree up to the edges of the network. Securing routing protocol messages will not prevent an illegal host from sending an IGMP report (either join or leave) message. Similarly, deploying a group key management protocol and applying an end-to-end data encryption technique will not eradicate these problems since a host can always send an IGMP report message. The extent of possible damage due to a forged report message depends on the type of the message accepted by the AR. There are three types of IGMP report messages.

1. **State-change join message** is an IGMP report message that will add one or more reception states maintained by the ARs. Hence, a forged state-change join message will pull the multicast DDT all the way from the branch point to the subnet. Thus, useless routing or forwarding states will be added to the CR's routing tables even if there is no traffic. Furthermore, if a large number of reception states is added for multicast groups that really exist, the Quality of Service (QoS) inside the subnet will be degraded, and this may create the risk of Denial of Service (DoS) attack also. The attack will be severe for secured groups, where EUs might had paid before for a desired level of QoS. This attack will leak multicast traffic (even when data are sent in encrypted format) to some

illegitimate members, and if a large number of bogus members had joined, the QoS will be reduced.

2. **State-change leave message** is a report message that will delete one or more reception states maintained by the ARs. A forged state-change leave message will prune the DDT and will preclude a legitimate EU from receiving multicast data for which he/she might had paid earlier. Therefore, this will create another form of DoS attack.

3. **Current-state message** is a report message that a host sends periodically to keep alive the existing receptions states maintained by the ARs. A forged current-state message may cause the ARs to think there are group members of a group on a network when there are not. The effect of this message will be similar to state-change join message if the EU left silently (without sending state-change leave message), and the attacker continues sending current-state messages.

A forged report might be sent targeting an intra-domain or inter-domain multicast group. In an intra-domain attack, the receivers and senders are in the same domain, and it is easy to prevent the attack by enforcing access control. In an inter-domain attack, receivers and senders are not in the same domain and the multicast DDT may cross the receivers' and/or the senders' domain(s). Therefore, access control is not easy and may require inter-domain communication.

In the IGMPv3 specification [17], a list of attacks that might be generated due to forged IGMP message has been presented. The consequences of these attacks are wastage of the local subnet's bandwidth and of the resources of the hosts and the routers. IGMPv3 recommends the use of IPsec [69] with Authentication Header [70] to authenticate IGMP messages. IGMPv3 has been developed to be deployed for Open groups only. Hence, the suggested security features of IGMPv3 are unable to enforce the access control for the IGMP report messages. It should be noted that

the attacks mentioned earlier could be prevented by implementing access control and not by implementing message authentication only [23]. Therefore, the NSPs should deploy an access control mechanism at the ARs, which will enforce a mandatory authentication and authorization of EUs before joining or leaving a secured group through IGMP. Therefore, each time an EU sends a state-change join/leave message, it will trigger an authentication session between the EU and the AR. Finally, if a current-state message (targeting a secured group) comes from an EU who had not authenticated and authorized before, the message will be dropped by the AR.

3.2.2 Effects of Forged Sender

Without an effective sender access control, an adversary may exploit the existing IP multicast model, where a sender can send multicast data without prior authentication and authorization. The following three attacks are easy to mount by a forged sender:

1. Replay attack: This is the simplest attack that a forged sender may generate. An adversary will keep a copy of a previously sent multicast packet, and will replay it after a period of time. A secured group communication (that uses group key management protocol) implements an anti-replay mechanism and allows the receivers to detect a replayed data packet. However, a receiver will detect this at the end of the DDT, while the replayed packet had already come through all the intermediate CRs. This is obviously a wastage of bandwidth and router resources.

2. Sender address spoofing attack: This attack is more dangerous than a replay attack, where an adversary will impersonate a legitimate sender address. Thus, it will get freedom to generate any packet as a multicast data packet. Again, a secure group communication fails to prevent this attack at the intermediate routers, and can only detect this attack (if data origin authentication is imple-

mented) at the receiver end. Therefore, the effects of sender address spoofing are similar to those of a replay attack.

3. Denial of Service (DoS) attack: An adversary may generate very large sized packets and may flood the DDT by either replay attack or sender address spoofing attack. This can create an efficient DoS attack. DoS itself is not an attack here. This is a consequence of the replay and the sender address spoofing attacks. Multicast suffers from a DoS attack more severely than any other attack due to its amplification of data packets enroute.

The best way to prevent an attack from a forged sender is to establish a checkpoint at the entry point of a DDT, which is the one-hop AR of the sender. A forged sender may target an intra-domain or an inter-domain group. As we have mentioned in the previous section, for an intra-domain group, it is easy to prevent a forged sender by enforcing access control. In an inter-domain group, it is difficult to prevent a forged sender as the adversary might be located in any intermediate domain. Hence, protection of the DDT is required.

3.2.3 Scope of the Thesis: Access Control Architecture

The present architecture of IP multicast needs to be extended to achieve receiver and sender access control. Here "access control" encompasses all three functionalities of the AAA framework: authentication, authorization and accounting. The access control architecture we are developing could be divided into three exclusive and loosely-coupled sub-problems. The boundaries of the sub-problems are shown in Figure 11.

Receiver Access Control

Receiver or EU access control must be implemented at the receivers' end, and

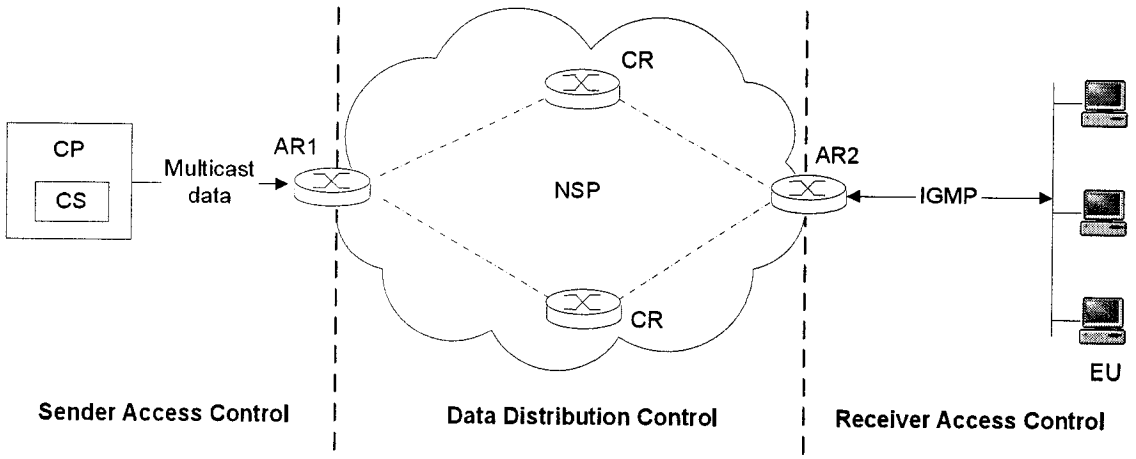

Figure 11: Access Control Architecture

between the EUs and the directly connected AR. In Figure 11, AR2 is an IGMP router, which will protect a secured multicast group from various attacks generated by the forged IGMP messages from the EUs' subnet. This protection will be achieved by EU authentication through the IGMP protocol and implementing AAA functionalities at AR2. Therefore, EU's authentication information must be carried inside IGMP messages, and a suitable extension of the IGMPv3 protocol is needed.

Sender Access Control

Only an authenticated and authorized sender would be allowed to send data to a secured group, and each data packet should be authenticated before forwarding to the DDT. This will be achieved by implementing AAA functionalities at AR1, which is directly connected with the sender. Hence, AR1 will protect the DDT from any attack originating from a forged sender.

Data Distribution Control

This type of access control is part of sender access control is essential for inter-domain groups. However, data distribution control may be useful for intra-domain groups also. The DDT must be protected from attacks generated by a forged sender or compromised router.

3.2.4 Group Key Management vs. Access Control

In the literature, there exist many Group Key Management (GKM) protocols (e.g., GOTHIC [63]) that provide access control for multicast groups. It should be noted that their definition of "access control" is a subset of the access control (i.e., AAA functionalities) we have addressed in this thesis. Most of the GKM protocols only provide indirect EU authentication and a crude authorization.

In a GKM system, an EU has to register with the Group Controller/Key Server (GCKS) to receive the Data Encryption Key (DEK), which is used to encrypt/decrypt multicast data. Hence, the possession of the right DEK implies an authenticated member. Next, an authenticated member joins the secured group by sending an ordinary IGMP message. Therefore, there is no direct authentication while joining a multicast group, and an adversary can always join a secured group.

By deploying a GKM protocol no flexible or sophisticated authorization could be achieved. The only way to authorize a member is to manipulate the DEK. For example, consider a group member who is authorized to receive data during a specific time slot of the lifetime of the group. If we want to achieve this using a GKM protocol, the GCKS should modify the DEK at the starting and the ending of the time slot. Now, if this has be to implemented for hundreds of group members, it will drastically affect the scalability of the GKM protocol. Even worse, some of the authorization features, such as, a group member is allowed to receive a specific amount of data, could not be implemented using the GKM protocol alone.

The third issue, accounting, could never be achieved by a GKM protocol. However, accounting is an essential element for a revenue generating application. IP multicast deployment has been slow due to several factors. One of the important reasons is lack of support for accounting and billing services for multicast based applications.

From the above discussion, it is reasonably established that a GKM could never be

a solution for multicast group participant access control. It is worth while to note that we are not precluding the use of GKM protocols for integrity and confidentiality of multicast data. We are clearly distinguishing the functionalities of a GKM protocol and the access control architecture that we have developed. Therefore, our access control architecture should be seen as orthogonal to the goals of a GKM.

3.2.5 Participant Access Control and End-to-End Multicast Security

Participant access control is composed of three independent pieces: receiver access control, sender access control and data distribution control. Any of these pieces can be implemented independently, although data distribution control without any effective sender access control will always be vulnerable to attacks from the sender's subnet. However, the access control architecture we have developed should not be implemented stand alone. It is worth while to note that although we have solved an important piece of multicast communication, the complete and end-to-end security of multicast communication depends on many other factors.

We have already mentioned that secure multicasting is composed of control plane security and data plane security. Control plane security, which is achieved by securing multicast routing protocol and host-router signaling protocol, protects the distribution tree from unwanted modification. Data plane security provides confidentiality, integrity and source authentication using one of the GKM protocols. The access control architecture developed in this thesis should implement secure multicasting techniques to provide end-to-end security. Moreover, security of all other protocols and communication (e.g., AAA, PANA, EAP, ERP, etc.) that we have deployed in the participant access control architecture must be assured following the guidelines provided in the corresponding specifications.

3.3 Requirements of Our Design

Our goal is to develop a generic architecture with minimum overhead. We have defined the following requirements that our proposed architecture should meet:

- Only an authenticated and authorized receiver (sender) would be permitted to receive (send) data from (to) a Secured group. Depending on the business model accounting may also be required.
- Authentication of a receiver/sender should be optional so that the architecture will be implemented in addition to the service model of Open groups that operate following the classical IP multicast.
- Security never comes free of cost, and is not even required sometimes. The required level of security depends on the specific application, the value of the information being delivered, available hardware and software resources, etc. The proposed model should be able to provide a wide range of authentication mechanisms from simple password to asymmetric keys (digital signature).
- An IGMP report message that has a secured group address, will be processed by the AR if it had been originated from an authenticated and authorized EU. If a report message changes the reception state, the EU originating this message must be authenticated and authorized. Otherwise, the report message must be dropped.
- Per-packet cryptographic protection should be implemented at the entry point of the DDT (i.e., at the AR) to develop a solution that would be free from replay, sender address spoofing and DoS attacks.
- The most appealing features of multicast are scalability and bandwidth conservation. Given that access control will introduce additional tasks for the ARs, hosts and some of the CRs (while protecting distribution tree), the proposed

architecture should not overload the DDT routers or the hosts, and the key concern must be to deliver a packet as fast as possible in a secured way.

- The architecture will be independent of the underlying routing protocol. For example, the way the DDT is constructed by the routing protocol will not affect our solution, multicast data may be distributed through either a unidirectional (e.g, in case of PIM-SM [32]) or a bi-directional (e.g., in case of CBT [13]) tree.

- The proposed model should not be restricted to intra-domain groups, and should be practicable for inter-domain groups as well. An inter-domain access control will require communication and transportation of AAA information between more than one domain.

- Both Source-Specific Multicast (SSM) [47] and Any-Source Multicast (ASM) groups should work in our architecture. An SSM group has only one sender while an ASM group may have more than one senders.

- The proposed architecture will recognize the requirements of necessary e-commerce communications and should be compatible with any standard e-commerce framework.

3.4 Proposed Architecture

The problem of adding AAA functionalities is distributed in nature, and also composed of different protocols and entities. In the literature, we have found only a few efforts to define the domain of the problem or to identify the functionalities of different sub-modules. The architecture shown in Figure 12 was first proposed in [56], and further illustrated in [9].

3.4.1 Where to Implement AAA Functionalities

During unicast communication, the entry point of a network is the obvious choice to implement AAA client behavior. In case of IP multicast, the point of implementation of AAA client behavior is difficult to specify. In unicast, there is a one-to-one relation between the sender and the receiver(s), whereas in multicast, intermediate routers replicate multicast data packets in the middle of the transmission, if needed. Thus, the sender has no idea about the presence of the receivers, and who the actual receivers are. In Figure 12, the receivers use IGMP/MLD to inform AR2 of their desire to join a group. As a result, only AR2 has some sort of way to learn of the presence of a receiver in the network to which it is attached. This is clear that the only suitable point to implement AAA client behavior is AR2. Similarly the AR of the network to which the sender is directly connected (AR1 in Figure 12) will be the best place for sender authentication and other AAA functions (i.e., authorization and accounting).

3.4.2 Participants and their Roles

The proposed architecture has a number of participants: Content Provider (CP), Content Server (CS), CRs, ARs, EUs, AAAS, Merchant (MR), Financial Institution (FI) and Group Owner (GO). To give the EU a single point of contact for a variety of services, we will assume the existence of the Merchant. Finally, we will assume that the ability of the EU to pay for services will be certified by an FI. The NAS or the AAA client will reside inside the ARs. The CP offers the product to be delivered, and makes use of a CS to send to the multicast group. It will send multicast packets to AR1, and AR1 will forward the packets through the multicast DDT, which has already been constructed by the CRs. AR2 will perform dual tasks: it will receive and process the IGMP messages and also act as a NAS by communicating with the AAAS. The policy server will either push the access control policies to the AAAS, or the AAAS triggered by EU requests to join a group or sender request to send data

will pull the policies. The GO is responsible for the creation and overall activities of the group. We have assumed that the GO and the Merchant are co-located and reside in the same entity.

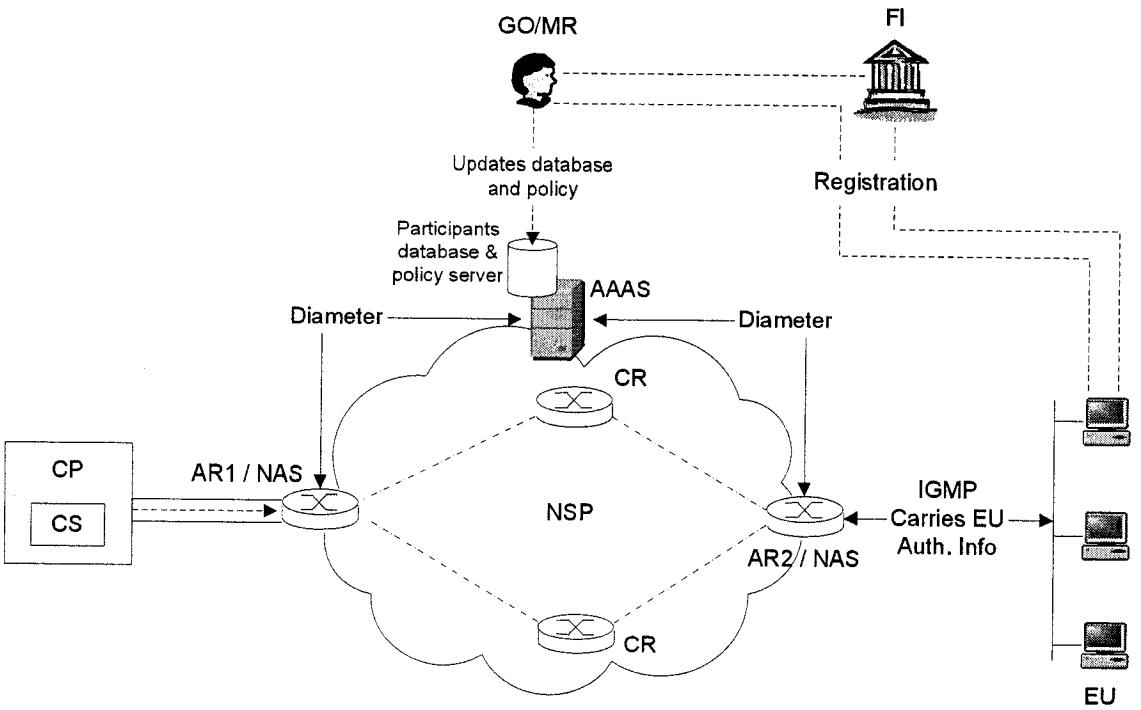

Figure 12: Different Components of Proposed Architecture

Although the CP, the GO and the NSP are assumed to be different entities and to reside in the same network (i.e., inside the same Autonomous System or AS) in Figure 12, the architecture is dependent on the specific business model and the trust relation between these three entities. They may be co-located and maintained by the same authority. If they are different entities, they must trust each other and should have agreed to share the revenue on some basis. Again, in Figure 12, it is assumed that the GO has outsourced the participant (the sender and the EUs) authentication and delegated the NSP to perform the authentications on its behalf. Thus, the participants' database that resides in the AAAS is accessible by the GO, and the GO can update that database. This is a restrictive assumption and any other variation is possible depending on the architecture.

3.4.3 Different Events of the Architecture

We have listed different steps from group creation to user joining that should take place. Though we have presented different steps one-by-one, some of them may take place in parallel and the order may be violated.

Session announcement: The GO will announce the schedule of the multicast session. For example, *CrickInfo* will live multicast the upcoming Test cricket match of Bangladesh vs. England from London, UK for five days from 1st to 5th of July 2008. *CrickInfo* will advertise this live telecast on their websites and other cricket related website to spread the information to the potential EUs.

EU Registration: An EU must register with the GO/Merchant prior to the start of the session. The registration may be online using the Internet or may be offline (e.g., by phone, fax, etc.). During registration, the EU's credit will be checked by communicating with his/her Financial Institution (FI), and the user will indicate the type of service she is requesting. For example, one EU is willing to watch the cricket match for the last two days, whereas the other may want to watch for all five days.

EU account creation: At the end of the registration process, the GO will create the EU's account, and will forward the authentication token to the EU. For example, it may be the user's name and the password or the keys of a digital signature. Depending on the nature of the service, authorization information will be created for the newly created account. In our cricket match example, if the user pays for the last two days, she will be allowed (authorized) to view (receive multicast data) for that period only. Next, the GO will add the new user account and authorization information of the EU to the participants' database of the AAAS.

Sender account creation: If the GO and the CP are different entities, the GO will

create an account (e.g., sender address, sender identity, authentication information, etc.) for each legitimate sender and will update the participants' database. This information will also be forwarded to the sender for whom the account has been created.

Update policy server: The GO will update the access control policy that resides in the policy server. If the policy server is outside the AAAS, the AAAS will communicate with the policy server using a proper policy protocol (e.g., COPS [28]).

Sender authentication and authorization: The sender will inform AR1 to start an authentication protocol session through which AR1 will authenticate and authorize (with the help of AAAS) the sender. AR1 will communicate with the AAAS using the Diameter [18] protocol. To validate the authentication and authorization information of the sender, the AAAS will check the participants' database and may have to consult the access control policy that is stored in the policy server.

Cryptographic protection of multicast data: Only an authenticated and authorized sender will be allowed to send data to a secured multicast group. Moreover, the sender will add a piece of authentication information with each data packet so that AR1 would be able to cryptographically authenticate a packet before forwarding it to the DDT.

EU joining/leaving a secured group: An EU will send a state-change IGMP join/leave message to AR2 to join/leave a secured group. In response, AR2 will open an authentication protocol session with the EU through which AR2 will authenticate and authorize (with the help of AAAS) the EU. The EU authentication token (which was previously received during registration) will be forwarded to AR2 using an extended IGMP protocol. AR2 will communicate with the AAAS using the Diameter [18] protocol. The AAAS will check the participants' database, communicate with the policy server if required and inform AR2 of

authentication success or failure. It should be noted that depending on the method of authentication, multiple round-trips might be required between the EU and the AAAS through AR2. For a successful authentication, the AAAS will send authorization information for that specific EU to AR2.

Accounting: AR2 (AR1) informs the AAAS of the start and end of accounting. AR2 (AR1) will gather accounting information for a session of an EU (a sender), and at the end of the session will forward the gathered data to the AAAS. The AAAS will update the participants' database to store the accounting data. The GO can access this database at any time.

3.5 Relation between Access Control and E-commerce Communications

The access control architecture presented in the previous section has two main features: participant access control and e-commerce communications. A revenue generating access control architecture should be addressed in the presence of e-commerce communications. Participant access control, specially receiver access control, will be performed at the end of successful e-commerce communications. Hence, in our proposed architecture, we have accommodated e-commerce interactions among the related entities. It should be noted that e-commerce problems are out of the scope of this thesis. Therefore, we have not included them in the problem definition and the requirements of our design. However, while discussing different events of the proposed architecture, we have mentioned the events related to e-commerce communications to explain the flow and dependency of different events.

The e-commerce communications including session announcement, registration, account creation and update participants' database are out side the scope of our research. During registration, an EU will prove her identity and credit (e.g., by using

a valid credit card) to the GO, and confirm the service or content she is intending to purchase. The GO will validate the credit by communicating with the Financial Institution (FI) of the EU. Through a proper e-payment protocol an e-transaction will take place. Different e-commerce issues, such as authentication, data confidentiality, non-repudiation, money atomicity, etc., should be addressed for the registration process. The GO and the AAAS will authenticate mutually and should trust each other. Moreover, a security policy should be enforced through which the GO will be able to update the participants' database that resides in the AAAS. For more information, interested readers can look at [109].

The rest of the thesis will focus on the participant access control. We have solved receiver access control and sender access control separately to make them independent of each other. In addition, a data distribution control mechanism has been developed for inter-domain groups. We have also designed a policy framework to enforce the access control policies.

Chapter 4

Receiver Access Control

In the previous chapter, we have presented the proposed architecture to provide access control (i.e., authentication, authorization and accounting) for both receiver (or EU) and sender for a secured multicast group. However, the present multicast architecture deploying Internet Group Management Protocol (IGMPv3) [17] fails to provide necessary functionalities for receiver access control. IGMPv3 does not carry any identity or authentication information of the EUs. Hence, IGMP needs to be extended to be used in our proposed architecture.

In the literature, a few extended/modified versions of IGMP exist to provide receiver access control. However, none of the attempts was successful in attracting the attention of the researchers and the IETF community due to their poor design and because they suffered from different attacks. Therefore, when we have developed Internet Group Management Protocol with Access Control (IGMP-AC) by extending IGMPv3, we were very cautious not to repeat the same mistakes that the other researchers made.

In the following, we will start with the related work that different researchers have accomplished to deploy receiver access control in IP multicast. We will also present a

summary, and list various limitations of the existing methods. These limitations have guided us to set the requirements for IGMP-AC. Next, we will present the IGMP-AC in detail with state diagrams of different entities, list of newly created messages and description of the required reception states maintained by the involved entities. Finally, we have modeled the IGMP-AC protocol in PROMELA [48] and verified this model using SPIN [49].

4.1 Related Work

AAA issues are not addressed well in IP multicast by researchers. We have found only a few attempts that meet our requirements. A number of researchers have proposed EU authentication in conjunction with group key management, where authentication takes place during group key distribution. Some of these methods have been designed for "group communication" instead of multicast communication, when the underlying communication technique is not necessarily IP multicast. Moreover, a number of group communication protocols do not use IGMP at all for joining/leaving a group. It should be noted that group key distribution is happening in the application layer whereas IGMP runs between the network layer and the transport layer. Therefore, the two authentications—authentication through IGMP to join a secured group and authentication to get the group keys—should not be coupled and performed in the same layer. Hence, in the following, we have excluded group key management mechanisms in our list of related work.

4.1.1 End User Identification and Accounting (EUIA)

The End User Identification and Accounting (EUIA) [105] system deploys AAA protocols and a Group Policy Server for EU authentication, authorization, user-based accounting, and host identification. The EUIA system is composed of a AAA frame-

work, a Group Policy Server, a multicast edge router or AR, and hosts directly connected to the AR in the same subnet. In this model, an AR acting as a NAS, authenticates an EU by communicating with the AAAS, and the Group Policy Server confirms the authorization status of the EU. EUIA extends IGMPv3 by adding two new messages to accommodate user authentication and host identification. Although EUIA supports different types of authentications, it fails to provide a flexible authentication framework (e.g., EAP [2]).

4.1.2 Internet Group Membership Authentication Protocol (IGAP)

The Internet Group Membership Authentication Protocol (IGAP) [43] provides all the functionalities of IGMPv2 [30] with the addition of EU authentication and accounting by deploying RADIUS [96] protocols. The user authentication information in IGAP enables a provider to control the distribution of the multicast traffic as well as to collect real time user accounting information. Instead of using static access control (e.g., physical ports, host IP or MAC address) IGAP uses user based access control policy. It supports simple password authentication and challenge-response or CHAP authentication. It suffers from a serious threat due to its use of plain-text password for authentication. Moreover, it does not provide the "source filtering" feature.

4.1.3 Upload Authentication Info using IGMPv3

An Internet Draft to modify IGMPv3 to authenticate an IGMP Report message when needed [45] is a work in progress. As we have mentioned earlier, a multicast enabled AR has to maintain IGMP reception states. Depending on these states, a router will trigger the multicast routing protocol and multicast traffic starts to flow. The goal of this ID is to protect IGMP reception states from any malicious host or end user

by authenticating IGMP messages. It added two messages to the IGMPv3 protocol: Authentication Query and Authentication Report. To minimize the number of times authentication information would be exchanged, if an IGMP request changes any reception state, only then are these authentication messages exchanged. Otherwise, standard IGMPv3 messages are exchanged between a host and a router. A host sends authentication information inside an Authentication Report message in the format of 32-bits words only once.

4.1.4 Other Methods for Receiver Access Control

In this section, we will discuss four other methods for receiver access control. All of them have one common feature, they have proposed different modifications of the IGMP protocol.

A multicast architecture has been presented in [46], where a centralized Multicast Manager (MM) for each domain has been introduced. When a Leaf Network Router (LNR) or AR receives an IGMPv3 Report message, the LNR relays the message to the MM, which authenticates a host or receiver using its IP address only. The MM uses the "source filtering" option of IGMPv3 to authorize or control the access of the receivers. This model clearly fails to provide a secured access control mechanism because the relayed messages are not even integrity protected and IP address based authentication is always vulnerable to the address spoofing attack.

To authenticate each participant (sender and receiver) an architecture has been presented in [55], which deploys AAA protocol. In this model, a challenge-response protocol with shared secret for EU authentication has been proposed. A host willing to receive data should communicate with the nearest AR using an extended IGMPv2 protocol, and a challenge-response session between the egress router and the host will be initiated. The egress router collects authentication information and forwards it to

a back-end AAAS to perform the actual authentication. This approach and all other subsequent ones that we are going to discuss do not provide any authorization and accounting features.

Multicast receiver access control based on a one-time token is presented in [39]. This token is digitally signed, and distributed by the Key Server (KS) to the users. This token contains a symmetric key called the IGMP-key, which is used by the receiver to authenticate the IGMP Report message. The KS also distributes the same token to the multicast routers.

In [36], an IGMPv3-based method has been designed to prevent DoS attacks by authenticating end users (receivers). This method dose not change the packet format of IGMPv3 and uses the *Auxiliary Data* field inside the *Group Record* of the IGMP Report message to carry user information and authentication data. For authentication, MD5 hash value of the concatenation of IGMP packet and a shared secret (e.g., password) is sent by the receiver. The router possesses the same secret, and on receiving IGMP Report message, it goes through the same process. A major drawback of this method is that the AR performs the authentication by itself, which will add extra workload to the AR.

An up to date (published in 2005) survey on multicast receiver and sender access control, and its applicability to the mobile IP environment has been presented in [68]. The existing approaches have been classified into three classes: digital signature-based solutions, shared secret-based solutions, and hybrid solutions. A set of requirements has been used to determine their efficiency and limitations in stationary and mobile networks. The contribution of this paper lies in finding four major limitations of the existing methods: lack of sender access control, vulnerable to DoS attacks, no efficient user exclusion scheme and not suitable in mobile IP environments.

Different security issues of multicast content distribution have been investigated in [64]. Four areas of research for security of content distribution have been identified:

receiver access control, group key management, multicast source authentication, and multicast fingerprinting. The authors have explained the issues and vulnerabilities that exist and have discussed the research that has been accomplished in each area. Finally, some unsolved issues have been highlighted for future research that must be addressed to deploy multicast enabled applications.

4.1.5 Summary of Different Approaches

A summary of the different approaches regarding AAA issues is presented in Table 2. We have listed only the methods that are similar to our research goals. Comprehensive lists of receiver and sender access control mechanisms including group key management protocols can be found in [68, 64]. By analyzing the existing methods we have identified the following essential findings:

1. Only a few models deal with authorization and accounting issues in IP multicast.
2. Sender authentication is overlooked most of the time.
3. Some of the methods are based on IGMPv2, which has already been outdated by IGMPv3.
4. Most of the architectures are confined to a specific authentication mechanism, and they do not support a wide range of authentication schemes.
5. The unique advantage of IGMPv3 over IGMPv2 is "source filtering", which is absent in most proposals.
6. Only one method [55] has successfully used a AAA protocol (RADIUS). Two others, EUIA and IGAP, intend to use AAA protocols, but have not presented any detail at the architectural level.

Table 2: Summary of Different Approaches Regarding AAA Issues

Approach	IGMP Version	Auth-entication	Autho-rization	Acc-ounting	Remarks
EUIA [105]	IGMPv3	Flexible authenti-cation	Yes	Yes	Provides host identification. No pro-tocol exists for communication be-tween NAS and Group Policy Server.
IGAP [43]	IGMPv2	Password (manda-tory) or CHAP	Yes	Yes	Sends plain-text password. Does not support any secured authentication.
Upload authenti-cation info by IGMPv3 [45]	IGMPv3	No specific scheme	No	No	Supports source filtering feature. Not applicable for an authentication (e.g., IKEv2 [66]) that requires mul-tiple round-trips.
IGMPv3 to multicast access [46]	IGMPv3	Using host IP address	By IGMPv3 source filtering	No	Suffers from address spoofing attack. Does not support any advanced au-thentication scheme. Single point of failure.
Authentica-tion using RADIUS [55]	IGMPv2	CHAP	No	No	Provides sender authentication also. Suffers from replay attack. Does not support any secured authentication.
IGMP key establish-ment [39]	IGMPv2	Access to-ken signed by digital signature	No	No	A Group Key Management protocol must be in place beforehand. Adds overhead to end host and multicast routers.
Shared secret [36]	IGMPv3	Hash of shared secret	No	No	AR must be updated with shared se-cret. Extra work for AR. May suffer from DoS attack.

4.1.6 Internet Drafts from the MBONED Working Group

The MBONE Deployment (MBONED) Working Group [81] at the IETF is develop-ing the functional requirements for accounting and access control for multicasting. A multicast network will be identified as "fully AAA enabled" when it fulfills the require-ments defined in [44]. They have classified different business models with respect to Content Delivery Services. Moreover, a framework to satisfy these requirements has been presented in [99]. However, they have designed a hypothetical framework, and

they have not presented any detail or identified different modules of the framework yet. While the intent of this framework is to provide a mechanism for user tracking and billing, and the associated access control, it does not address the interaction among the EU, his/her Financial Institutions (FI), and the collection of Merchants who might offer the content for purchase. This is in keeping with the areas of interest of the IETF. However, it is our belief that any proposed solution must mandate a simple control interface between the NSP and the e-commerce world. This will allow the EU to purchase individual "items" from a variety of MRs, for delivery via the facilities of the NSP.

4.2 Internet Group Management Protocol with Access Control (IGMP-AC)

The Internet Group Management Protocol with Access Control (IGMP-AC) performs access control of the EUs (only if required for a specific application) in addition to the IGMPv3 functionalities. Here, "access control" is used to address all the AAA functionalities: to authenticate successfully, to verify authorization to receive group data for which an IGMP-AC Report message has been sent, and also to keep accounting information for each EU. IGMP-AC was first proposed in [57] with the example of a simple password based authentication, which was an extension of the IGMPv3 protocol that added new messages and reception states. However, the EAP encapsulation mechanism was absent, and therefore the initial version did not support all sorts of authentication. In the next version [60], we have generalized the IGMP-AC protocol by redesigning it to encapsulate the EAP packets.

4.2.1 Requirements

The following essential properties have been identified for the extension of the IGMPv3 protocol to perform access control of EUs:

1. The IGMP-AC will provide a generic client-server authentication protocol, where the EU will act as a client, the AAAS will act as a server and the AR will perform the forwarding task. Thus, any suitable authentication protocol (e.g., EAP [2]) having client-server entities can be incorporated with the IGMP-AC architecture.

2. The EU authentication process should support a variety of authentications—from simple password based authentication to certificate based authentication. Hence, depending on the required level of security, the NSP and the GO will have freedom to pick a specific authentication mechanism.

3. IGMP version 3 is the current standard designed by the IETF. The extension will be based on IGMPv3 and must support the "source filtering" property as well.

4. IGMP-AC will not disrupt the usual function of the IGMPv3 and EU access control must be performed only if required. Thus, IGMP-AC will support the classical multicast model.

5. The least functionality and the minimal workload should be added to the ARs and the hosts.

6. Authentication is a costly process in terms of CPU cycles and bandwidth. Therefore, an EU should send the authentication information to the AR as few times as possible.

7. The IGMP-AC will not be inclined to any business model, and will not depend on the architecture of the network or the relation between the NSP and the CP.

8. The proposed IGMP-AC protocol should not be restricted to intra-domain groups, and should be practicable for inter-domain groups as well.

9. IGMP-AC protocol should be designed by extending or reusing standard protocols. Hence, it would be easily deployable in the existing IP multicast model.

4.2.2 Protocol Descriptions

It is important to clarify when the IGMP-AC protocol will perform access control. Previously, we have classified multicast groups into two categories: Open group and Secured group. The access control mechanism of the IGMP-AC will take place to join/leave to/from the Secured group only. While the IGMP-AC supports "source filtering" a Secured group will be either Group-Specific, (*, G) or Group-and-Source-Specific, (S*, G). Here, "*" means absence of any specific multicast source address, and "S*" means one or more multicast source address(es).

Three entities are involved in the IGMP-AC architecture: hosts, AR and AAAS. A host and the AR communicate with each other using the IGMP-AC while the AR and the AAAS will use one of the AAA protocols. We have explained the IGMP-AC protocol by using flow charts. For each entity (i.e., host, AR and AAAS), one flow chart has been drawn to explain the state diagram of the corresponding entity. In the flow charts, two-dimensional labeled arrows are used to represent sending or receiving of messages to or from the entity labeled inside the arrow. An incoming arrow represents receiving of a message and an outgoing arrow represents sending of a message. We have adopted the phrase `g_or_gs` to indicate Group-Specific or Group-and-Source-Specific. Thus, `query(g_or_gs)` is to mean that this is either a Group-Specific Query or a Group-and-Source-Specific Query. The circled state, labeled as "S", stands for the "Start" state. Thus, an arrow towards "S" means it is going back to the "Start" state.

In the state diagrams, `query` and `report` messages are standard IGMPv3 messages; `auquery` (*Authentication Unicast Query*), `areport` (*Authentication Report*) and `aresult` (*Authentication Result*) have been created for the IGMP-AC protocol. IGMP-AC has been designed to encapsulate EAP [2] packets: `eap_reqs` (`Request`), `eap_resp` (`Response`) and `eap_result` (`Success` or `Failure`). EAP encapsulation will be explained in detail in Chapter 5.

Finally, we are assuming that the security mechanisms recommended in the IGMPv3 specification to authenticate the IGMPv3 messages have been deployed to authenticate the IGMP-AC messages also. This issue will be further explained in section 4.2.5.

4.2.2.1 Host Behavior

The state diagram for a host that communicates with an AR using the IGMP-AC protocol has been shown in Figure 13. Here, `API` means Application Program Interface or any upper layer protocol interface. The `leave` and the `join` messages are not part of the IGMP-AC protocol, they are used to express that one of the applications, running in the host, is interested in joining or leaving from a multicast group. The filter mode (`rstate`) of the reception states that a host maintains, may have any of the three values: `add`, `del` or `current`. When a host receives a `join` message, it sends a `report` message to the AR with `rstate` equal to `add` for `g_or_gs`. Then, it creates a new reception state and assigns the filter mode with the value `add`. When it receives a `leave` message it sends a `report` message to the AR with `rstate` equal to `del` for `g_or_gs`. The filter mode is also changed from `current` to `del`. The filter mode is assigned with the value `current` for the period of time the host maintains membership for a group. It is to be noted that the `report` message will not carry any authentication information.

In response to a `query(g_or_gs)`, the host will send a `report` message with

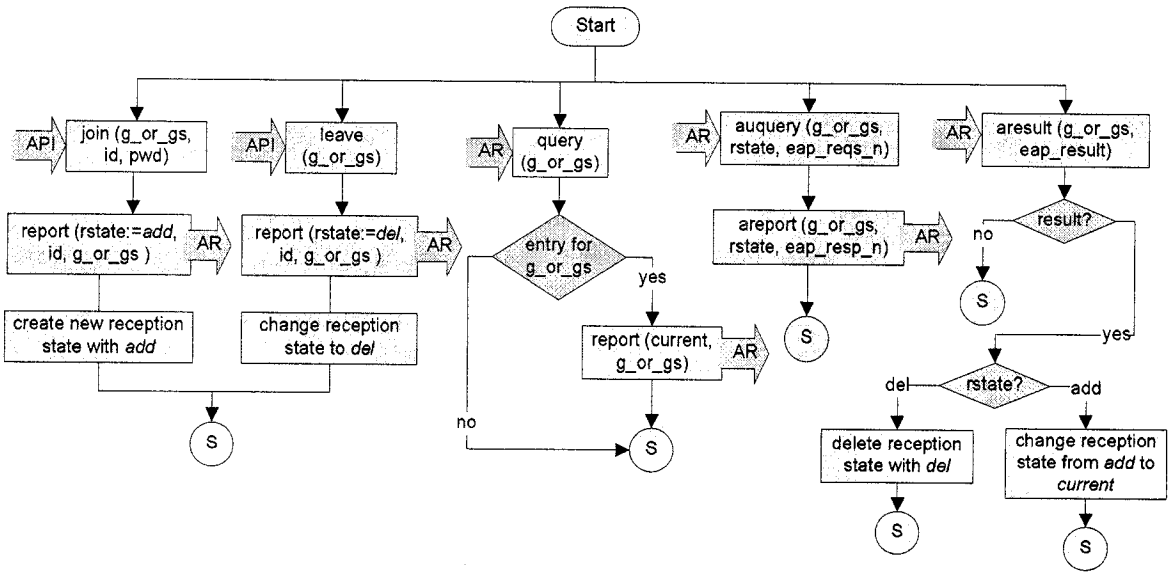

Figure 13: State Diagram for Host

`current` status if it has an entry for `g_or_gs`.

To minimize the number of times authentication data has to be sent on the wire, the host will send an `areport` to the AR carrying an `eap_resp` only in response to an `auquery` that encapsulates an `eap_reqs`. EAP `Request` and `Response` are shown with sequence numbers to indicate that the EAP server and the peer maintain sequence numbers to protect from anti-replay attack. Moreover, some EAP methods require more than one round-trip for successful authentication.

The AR will inform the host of the result of the authentication process by sending an `aresult` message that encapsulates the `eap_result` packet. On successful authentication, if the value of `rstate` was `add`, the host will add a new reception state for `(g_or_gs, id, auth_info)` by changing the filter mode from `add` to `current`. If the value of `rstate` was `del`, the host will delete the reception state `(g_or_gs, id, auth_info)` for which the filter mode was assigned `del` before.

4.2.2.2 Role of AAA Server (AAAS)

The state diagram for the AAAS that communicates with the AR using the AAA protocols Diameter [18], has been presented in Figure 14. We have assumed that `request`, `answer` and `account` are messages of the Diameter protocol. Previously, we have mentioned that it is possible to extend the Diameter protocol by defining new AVPs while any data delivered by the protocol is in the form of AVPs.

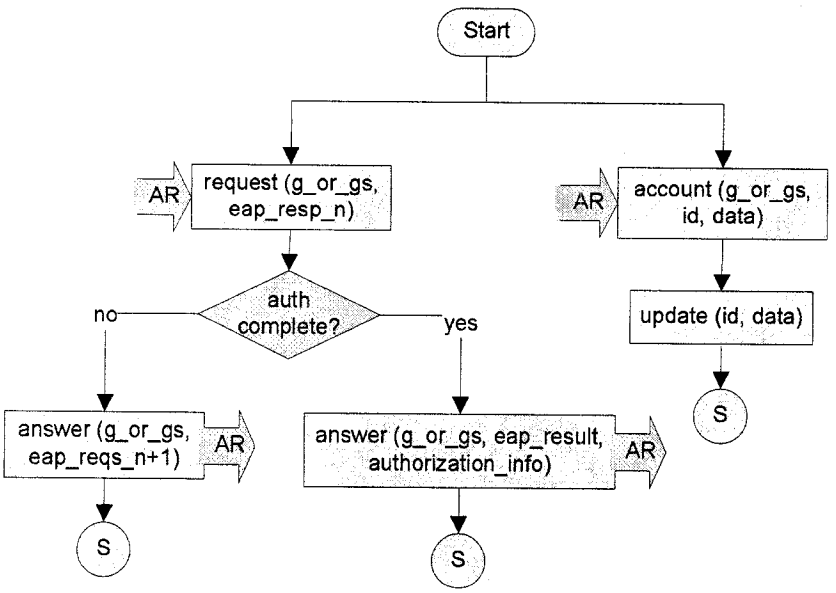

Figure 14: State Diagram for AAA Server

If the AR receives an `eap_resp` packet, it will send the packet inside a `request` message to the AAAS. The purpose of sending a `request` is to verify the identity and the authentication information of the EU, and also if she is authorized to join the group, `g_or_gs`. If the EU is successfully authenticated and also authorized for the group `g_or_gs`, the AAAS will send an `answer` that encapsulates the `eap_result` packet. If authentication or authorization fails, the `eap_result` packet will reflect it. Authorization information will be sent only for successful authentication. If it requires more information to complete authentication (either successful or failed), the AAAS will send an `eap_reqs` packet inside an `answer` message. This is the case when more than one round trip is required (e.g., CHAP). The sequence number of

eap_reqs has been incremented from n to n+1 to indicate that this is the next EAP Request message.

The AAAS will update the user database indexed by the identity of EU, id on receiving an account message. This database can be accessed later by the GO (see Figure 12).

4.2.2.3 Role of Access Router (AR)

The state diagram for the AR is presented in Figure 15. We have already explained all the messages of this diagram. The IGMP-AC has been developed to be used for Secured groups only, while the IGMPv3 [17] protocol should be deployed to be used for Open groups. An AR will be able to distinguish a Secured group from an Open group just checking the group address (provided that IANA [53] had delegated a range of class D group addresses for Secured group operations). The functionalities we have summarized for the AR in the following are only to deal with Secured groups.

- If the AR receives a report with rstate as current, the AR will check if the report had originated from an authenticated and authorized EU (or host). The identity of the originator could be verified only if an asymmetric signature algorithm had been deployed to protect the IGMP messages (see section 4.2.5). If the report had been sent by a legitimate EU the querytimer will be refreshed to keep alive the reception state.
- If the AR receives a report with rstate as add/del, it will create a new reception state with add filter mode in case of rstate equal to add or it will change the filter mode of the reception state (with g_or_gs) from current to del. The AR will also send an auquery to the host. Thus, authentication is only necessary during joining or leaving of an EU. The auquery will carry the first EAP Request packet (eap_reqs) to trigger an EAP authentication method.

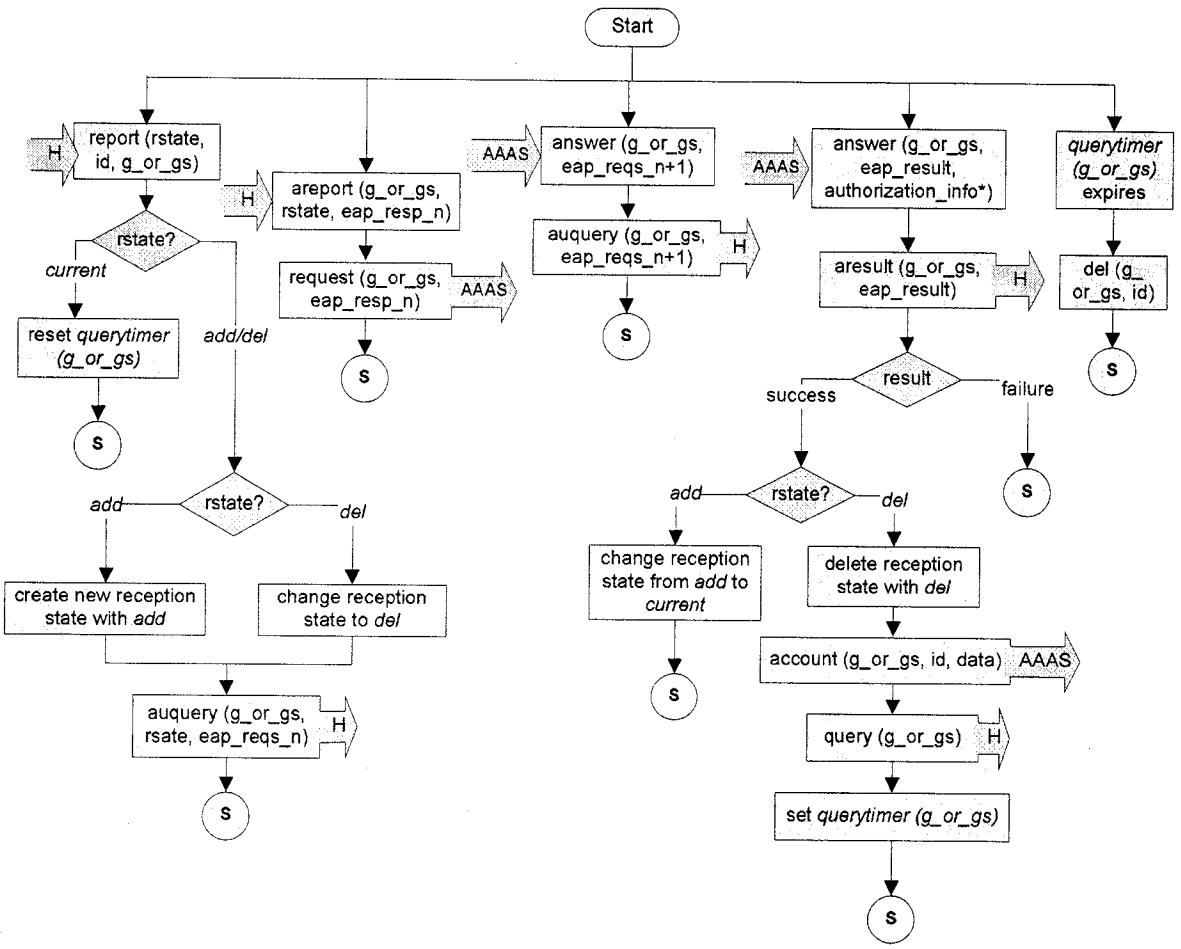

Figure 15: State Diagram for Access Router

- The AR will act as a pass-through device. When it receives an `areport` from the host, it will extract the `eap_resp` packet and forward it to the AAAS inside a `request` message. If the AR receives an `answer` from the AAAS, it will extract the `eap_reqs` (or `eap_result`) packet and forward it to the host inside an `auquery` (or `aresult`) message.
- If the authentication is successful and the `rstate` was equal to `add`, the AR will add a new reception state for `(g_or_gs, id)` by changing the reception state from `add` to `current`.
- For successful authentication, with the `rstate` value as `del`, the AR will delete the reception state only for the EU. Next, an `account` message will be sent to the AAAS and an IGMPv3 `query` will be sent to all the hosts inside the subnet.

Finally, the querytimer for the group g_or_gs will be set.

- In case the querytimer for the group g_or_gs is expired, all the entries with g_or_gs in reception states will be deleted. This will allow the AR to detect a silent departure of an EU, who had not left explicitly (by sending a report message with rstate equals del) from a secured group.

4.2.3 Additional Messages

The host membership report suppression was specified in IGMPv2 [30] and was removed in IGMPv3 [17]. Hence, an EU will always send a report message in response to an IGMP/IGMP-AC query. Given that IGMP-AC follows the IGMPv3 specification, IGMP-AC routers, by exploiting this property, will be able to maintain host-specific or EU-specific reception states for Secured group. In addition, an IGMP-AC router depends on this property to get the host IP address.

The IGMP-AC has added the following three messages to the existing ones of the IGMP protocol:

1. *Authentication Unicast Query*: In IGMPv3, all types of query messages are sent with an IP multicast destination address. A General Query is sent with an IP destination address of 224.0.0.1, the all-system multicast address. A Group-Specific or Group-and-Source-Specific Query is sent with an IP destination address equal to the multicast address of interest. An Authentication Unicast Query, denoted as auquery in the state diagrams, is sent with a unicast destination address. Whenever the AR receives a report from a host with the value of filter mode as either add or del for a Secured group, the AR gets the IP address of the host from that report, and uses that IP address as the IP destination address of the auquery message. This query encapsulates the eap_reqs packet and is sent with g_or_gs, hence, it is either Group-Specific or

Group-and-Source-Specific type.

2. *Authentication Report*: In the above state diagrams, Authentication Report is presented by `areport`. It carries EU authentication information through an `eap_resp` packet. This report is sent by a host in response to an `auquery` only.
3. *Authentication Result*: The AR forwards authentication information of the EU to the AAAS and gets back the `eap_result` packet from the AAAS. This `eap_result` is relayed to the host by an Authentication Result or `aresult` message. This is a unicast message and is sent with the IP address of the host.

4.2.4 Required Reception States

To operate the IGMP-AC successfully a set of extra reception states must be maintained by the AR and the hosts in addition to the IGMP reception states. The AR and the hosts will have to maintain different sets of reception states.

4.2.4.1 Reception States Maintained by the Host

A host will be informed through the Application Program Interface (API) or upper-layer protocol interface when it should perform a join operation to a multicast group. The host will be able to differentiate between an Open group and a Secured group by checking the group address received from the API or upper-layer protocol interface. If it is an Open group the host will follow the IGMPv3 standard [17]. In case of a Secured group, the host will have to maintain a list of:

Group address: The secured group address to which an EU has joined or sent an IGMP-AC `report` message to join from the host.

Source address: For the Group-Specific membership the `source address` field will be empty and for the Group-and-Source-Specific membership this field will con-

tain one or more source address(es).

Identity of EU: An EU identity will be in the format of Network Access Identifier (NAI) (e.g., `bob@example.com`) [1]. A NAI, consists of an user name (e.g., `bob`) and a domain name (e.g., `example.com`). This will be further discussed in Chapter 8, when we will present our inter-domain access control architecture.

Authentication information: Content of this information will depend on the EAP method used for EU authentication. It might be a simple password, a shared secret key or a digital certificate.

Filter mode: It will be one of the values of `add`, `del` or `current`.

4.2.4.2 Reception States Maintained by the AR

The AR will never maintain authentication information from an EU. It will maintain an entry for each authenticated and authorized EU. For each successful authentication, the AAAS will send the `authorization information` to the AR. Moreover, the AR should collect accounting information for each EU and when an EU leaves a multicast group, the AR should forward accounting information to the AAAS by sending an `account` message. To meet all these requirements the AR should maintain the reception states of:

Group address: The secured group address to which an EU has joined or sent an IGMP-AC `report` message to join.

Source address: For the Group-Specific membership the `source address` field will be empty and for the Group-and-Source-Specific membership this field will contain one or more source address(es).

Identity of EU: An EU identity will be in the format of Network Access Identifier (NAI) (e.g., `bob@example.com`). A NAI, consists of an user name (e.g., `bob`)

and a domain name (e.g., `example.com`).

Authorization information: The AR will receive this information from the AAAS for a successful authentication. Depending on the specific application, it might be quality of services, period of time, length of time, amount of data allowed to be received, etc.

Accounting information: The content of the information depends on the specific application. It might be period of time, length of time, amount of data allowed to be received, etc.

Filter mode: It will be one of the values of `add`, `del` or `current`.

4.2.5 Securing IGMP-AC Messages

In RFC 3376 [17], IPsec [69] with Authentication Header (AH) protocol [70] has been suggested to use for the IGMPv3 messages to provide connectionless integrity, data origin authentication and replay protection. Thus, an AR will be certain that a received IGMPv3 message was originated from a system (or, more specifically, a system with the proper key) on the LAN to which the AR is directly connected. Two types of authentication mechanisms are possible, a symmetric signature algorithm with a single key for the LAN or an asymmetric signature algorithm, where a sender can be authenticated individually. We are adopting the same concept of using IPsec with AH for the messages of the IGMP-AC protocol. However, the AH protocol provides limited security services as the IGMP messages are not encrypted. Moreover, in case of manual key configuration, anti-replay is not supported either. When we will present the EAP authentication method, EAP-IKEv2 [107] in section 5.2, we will explain an enhanced security scheme to authenticate and encrypt the IGMP-AC messages.

4.3 Verification of IGMP-AC using SPIN

We have used the formal verification language, PROcess MEta LAnguage (PROMELA) [48] to specify the verification model, and used the tool, Simple PROMELA INterpreter (SPIN) [49] to verify our model.

In PROMELA, procedure rules are used as formal programs to model distributed systems. The model should be as simple as possible yet sufficiently powerful to represent all types of coordination problems that can occur in a distributed system. A verification model is defined in terms of three specific types of objects: variable, process and message channel. PROMELA allows for the dynamic creation of concurrent processes. It supports two types of communications via message channels: synchronous (i.e., rendezvous) and asynchronous (i.e., buffered).

SPIN is a generic verification system, which can simulate the execution of a verification model by interpreting PROMELA statements on the fly. It detects many types of logical design error in distributed systems and checks the logical consistency of a specification. During simulation and verification it checks for unspecified receptions and unexecutable code. It also reports on deadlock, livelock, improper termination, etc.

4.3.1 Model Description

We have developed the PROMELA model from the state diagrams of the IGMP-AC protocol presented in section 4.2. We have designed the model in such a way that it should be as simple as possible, but at the same time, satisfy all the states and transitions of the diagrams. Our model consists of four processes: `host` for a host, `ar` for the AR, `aaas` for the AAAS and `init` to start the first three processes. The processes communicate with each other according to the architecture of Figure 12.

Thus, the hosts send messages to the AR using a common channel and the AR sends messages to the three hosts using three unique channels (to simulate sending of the unicast message, `auquery`). The AR and the AAAS communicate using separate channels. To model the communication that a host may receive messages from any upper layer protocol interface (see Figure 13), another channel has been created.

Before the starting of the processes, `aaas` is initialized with the AAAS database, which consists of the membership information (e.g., group and source addresses, user id, authentication information, etc.). At the beginning, the reception state of the `ar` is empty. We have invoked multiple instances of `host`, through which EUs can join/leave dynamically to/from different multicast groups. In the runtime, the `ar` will build up its reception states according to the behavior of the `hosts`. We have added joining/leaving instances inside the `host` processes in such a way that all the scenarios of the state diagrams are covered at least once. Finally, all the channels are defined as asynchronous type with a fixed buffer size.

4.3.2 Verification Results

SPIN can be used for either simulation or verification. Once we are sure that our model is free from syntax errors, different random simulation runs are performed with various options, and no errors were reported by the simulator. All these simulation runs have helped to build our confidence that our model is behaving as expected. The simulator also produces a Message Sequence Chart (MSC) to demonstrate the interactions of different processes. A partial MSC of our model generated by the SPIN simulator is shown in Figure 16.

Next, we have generated the verifier (a program written in C) from the PROMELA model using SPIN. This C program is compiled several times to produce the executable verifiers with different search techniques:

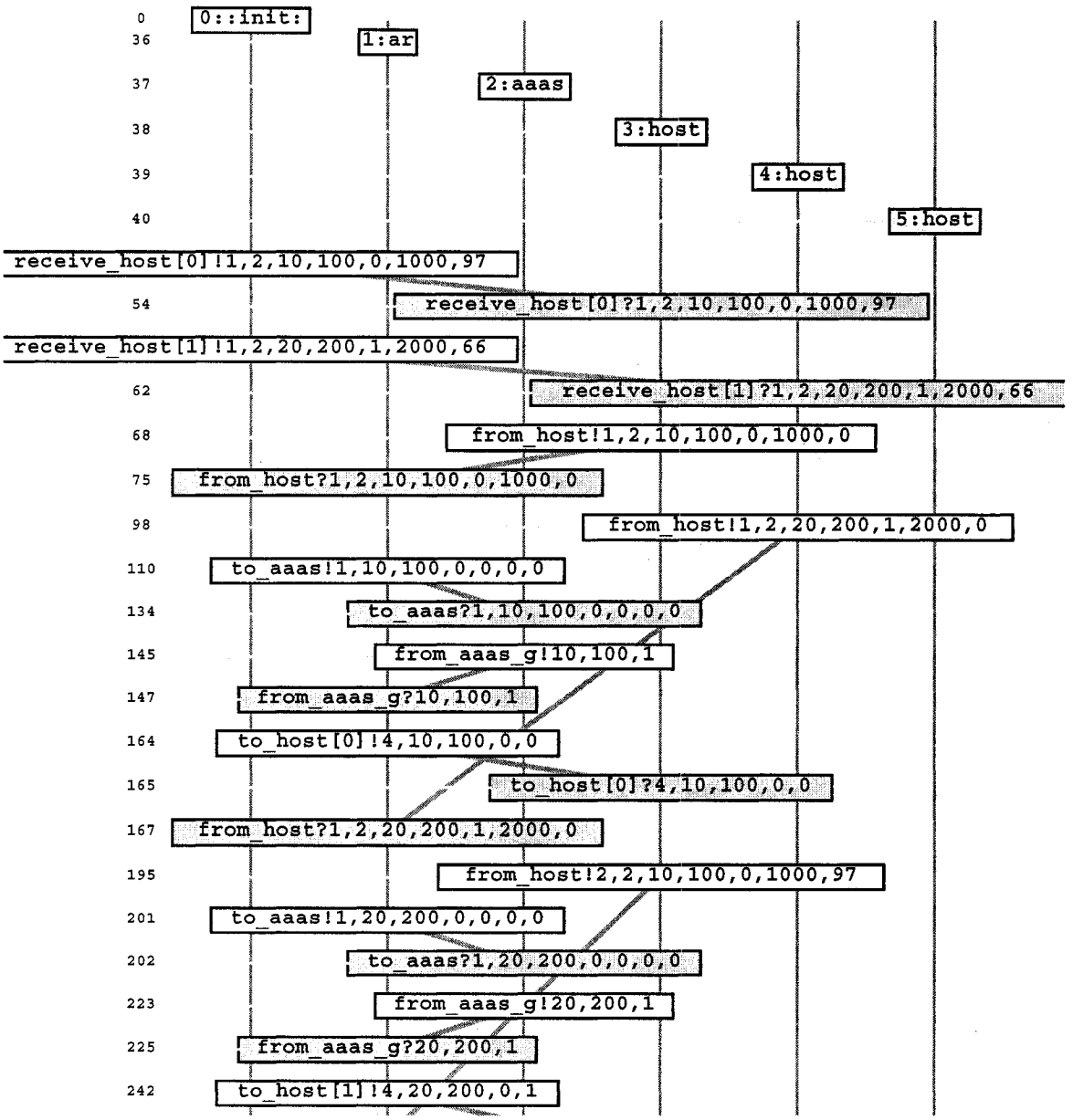

Figure 16: Partial Message Sequence Chart of IGMP-AC Model

Exhaustive search It is the default search technique that uses full state exploration search.

Depth-first search It is the default search technique.

Breadth-first search Using the compile time option, `DBFS`, the breadth-first search technique can be used. It is useful for finding the shortest path of an error.

Bitstate storage algorithm Using the compile time option, `DBITSTATE` the bit-state storage algorithm can be used. It is an efficient way of minimizing search time and memory requirements.

Hash compact Using the complie time option, `DHC`, the hash compact storage method can be used. In this method, the state descriptor is replaced with a 64-bit hash value that is stored in a conventional hash table.

Collapse compression The compile time option, `DCOLLAPSE`, collapses the state descriptors using an indexing method, which increases runtime but can significantly reduce the memory requirements.

When the verifier is executed it looks for the following errors:

Unreachable code Checking for unexecutable or unreachable code is the basic safety property that the SPIN verifier looks for by default. If there is any unreachable code, the verifier reports the name of the process and the line number(s) that are unreachable.

Invalid end state By default the the verifier checks for invalid end sate. This is characterized as a *safety* property. There are two types of end states: the last statement of each process and the states that are explicitly declared as valid end states using `end`-state label in the PROMELA code. At the end of the execution of the verifier, if any process terminates at any state that is not an end state, the verifier reports it as an invalid end state.

Non-progress cycle The verifier reports the existence of a non-progress cycle if it finds any infinite cycle during the execution. This is characterized as a *liveness* property. The compile time option, `DNP` and the runtime option, `1` should be used to check for non-progress cycles. Using the `DNP` option during compile time, necessary codes are included in the verifier for non-progress cycle detection.

Never claim A never claim is used to specify either finite or infinite system behavior that should *never* occur. Never claims are usually user-defined, however, a claim can also be generated by SPIN internally to support a predefined check (see p. 80 of [48]). In our model, the claim was generated by SPIN and automatically inserted into the verifier to define a check for the non-progress property.

Table 3: Verification Results of IGMP-AC Model using SPIN

Search technique	**Compile option**	**Never claim**	**Assertion violation**	**Acceptance cycle**	**Invalid end states**	**Unreached code**	**Depth reached**
Bitstate storage algorithm	DBITSTATE	–	No	–	No	No	438
Breadth first search	DBFS DHC	–	No	–	No	No	437
Exhaustive & Collapse compression	DCOLLAPSE	–	No	–	No	No	438
Nonprogress cycle check	DNP DCOLLAPSE	No	No	No	–	No	873
Hash Compact	DHC	–	No	–	No	No	438

In Table 3, the results of different verifications using different search techniques are listed. Here, "No" means absence of a specific error and "–" means that specific error was not looked for. Various depth levels reached by the verifier in search of errors are also presented.

The outputs confirm that our model is free from the errors that SPIN has searched for. From the outputs, it is also established that there is no unreachable state in our design. Thus, it can be concluded that the working of the `ar`, the `aaas` and the `host` processes are observed to function correctly.

Chapter 5

End User Authentication using EAP

One of the design goals of IGMP-AC was to provide the maximum flexibility by allowing any authentication scheme. The IETF has standardized the Extensible Authentication Protocol (EAP) [2] to support all sorts of authentication through different EAP methods in the wired or wireless networks. Therefore, the IGMP-AC protocol has been developed to act as the carrier of different EAP methods. However, the IGMP-AC does not preclude the use of any other authentication protocol given that it is based on the client-server model.

In this chapter, first, we have justified the operability of IGMP-AC as the EAP lower layer protocol. By applying the keys established through EAP method, the IGMP messages that are exchanged after the EAP session, might be secured with confidentiality and authenticity. This way, the IGMP messages could be provided with enhanced security in addition to the security features recommended in the RFC 3376 [17]. Next, we have demonstrated the encapsulation of EAP packets inside the IGMP-AC messages using the EAP-IKEv2 [107] method, an EAP method that uses

the same packet format as the Internet Key Exchange (IKEv2) [66] protocol does. Any other EAP methods can be deployed following this example.

To analyze the security claims of the EAP-IKEv2 method, we have built a model in High-Level Protocol Specification Language (HLPSL) and validated it using Automated Validation of Internet Security Protocols and Applications (AVISPA) [11]. We have also found a Man-in-the-Middle (MitM) attack in the peer-to-peer mode in absence of the authenticator. By adding an optional message, the attack has been prevented. Finally, the model has been extended for pass-through mode, and AVISPA has reported it free from attack.

5.1 EAP Encapsulation over IGMP-AC

IGMP-AC has been designed to carry EAP packets, and thus, it will support any EAP method. It should be implemented in pass-through mode, although it can be used in peer-to-peer mode also. The protocol stacks implementing an EAP method on top of IGMP-AC are shown in Figure 17. Thus, it will add an extra layer on the peer side between the EAP layer and the lower layer. IGMP-AC will act as the immediate lower layer of the EAP layer. A similar implementation can be found in [52], where the EAPOL (EAP encapsulation over LAN) messages encapsulate the EAP packets.

In RFC 3748 [2], six requirements have been identified for a layer to serve as the lower layer for EAP. We have extended IGMP-AC from IGMPv3 [17] by defining new messages and reception states only, and we are not going to modify the packet format or other functionalities of IGMPv3. Thus, in the following we have justified how IGMPv3 satisfies the requirements of the EAP lower layer protocol:

- EAP runs over an unreliable transport and thus lower layer reliability is not

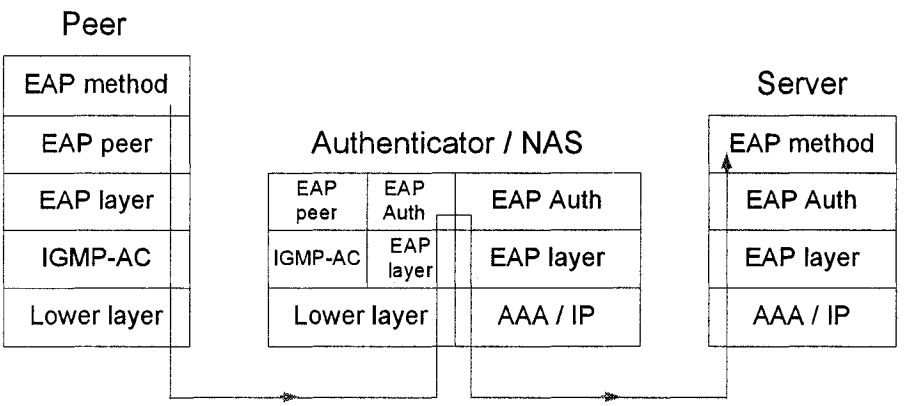

Figure 17: Protocol Layers for EAP Encapsulation over IGMP-AC

required. IGMPv3 does not offer any reliability. There is retransmission of messages in IGMPv3, but that is performed due to a specific state change (e.g., reception state or filter state).

- EAP relies on the lower layer error detection (e.g., CRC, Checksum, etc.). IGMPv3 computes and transmits a Checksum for every message, which is the 16-bit one's complement of the one's complement sum of the whole IGMPv3 message. This is recalculated in the receiving end to provide error detection.

- EAP does not depend on the lower layer security services such as confidentiality, integrity, replay protection, etc., of the lower layer. However, the lower layer may have its own security services. IGMPv3 provides optional limited security services by implementing IPsec [69] with Authentication Header (AH) [70] protocol.

- The EAP lower layer must provide an MTU size of at least 1020 octets. IGMPv3 does not define its own MTU size. It depends on the network over which it runs. If it runs over the Ethernet, the MTU of the Ethernet is 1500 octets, which is enough to accommodate 1020 octets even after leaving some octets for the headers of the IGMPv3 and the EAP protocols.

- If the lower layer is reliable, it may provide non-duplication stream to EAP. IGMPv3 is not able to remove duplicate packets. However, EAP with the help

of its Identifier field, can detect duplicate packets. Thus, this is not a mandatory requirement for successful deployment of EAP.

- The lower layer of EAP must guarantee the ordered delivery of the EAP packets. Although nothing is given in the IGMP specification regarding ordering of packets, the IGMP packets are sent with Time-to-Live (TTL) = 1 and are never relayed beyond the subnet. A packet will experience reordering only if it goes into a queue and/or multiple paths exist from the source to the destination. An IGMP packet is always going to be picked up by the Designated Router or AR on a single hop. Thus, re-ordering will never happen for an IGMP packet.

We have explored two other alternatives of using EAP by deploying Protocol for Carrying Authentication for Network Access (PANA) [33]. A short description of PANA has been given in section 2.5.

The first alternative we considered will trigger a PANA authentication each time an EU wants to join a Secured group. We do not have to extend IGMPv3 that much as IGMP-AC will not carry any EAP packet. The AR will act as the PAA and the EU's host will act as the PaC. When the AR will understand that an EU is asking to join a Secured group, it will start a PANA session with the host (PaC). However, this will not improve the model in terms of traffic and complexity. Moreover, the EU authentication will be decoupled from joining a Secured group. Thus, without proper channel or identity binding, it will be vulnerable to the Man-in-the-Middle (MitM) attack. Therefore, we have to bind the identity of the requester (that sent the IGMP join message) with the identity of the EU acting as the PaC.

The second option we explored and finally discarded is encapsulating the PANA packets inside IGMP-AC the same way IGMP-AC is carrying the EAP packets in our present model. It will create an extra PANA layer over IGMP and thus, increase overall packet size, traffic, and per packet processing time in the sender and the receiver ends. Moreover, from the above discussion we have reasonably established

that the IGMP-AC protocol is fully capable of carrying EAP packets without any modification.

5.2 Enhanced Security for IGMP-AC Messages

A standard EAP method provides mutual authentication and secret key derivation between the peer (host) and the server (AAAS) after a successful conversation. In pass-through mode, the authenticator (AR) sits between these two parties and relays messages from one to another. Thus, the peer and the authenticator do not authenticate or exchange their identities. However, after a successful EAP communication, if further information is exchanged between the authenticator and the peer, that information can be protected using the keys established between the peer and the server. In case of IGMP-AC, a host will establish keys with the AAAS, while a host and the AR will communicate (send IGMP-AC messages) after a successful EAP run. We can protect any subsequent IGMP-AC messages, which are sent after the mutual authentication and key establishment of the host and the AAAS. It should be noted that the initial IGMP-AC messages that are exchanged to complete an EAP authentication could not be protected using the keys established at the end of EAP session. To provide enhanced security for the IGMP-AC messages, the appropriate keying materials must be transported from the AAAS to the AR using an AAA protocol. This way we can provide confidentiality and other security services that are absent in the present IGMPv3 specification [17] as explained previously in section 4.2.5.

The details are outside the scope of this thesis, and the IETF is still working on the vulnerabilities of key transportation using AAA protocol [3, 51]. The major weakness of this key transportation lies in the inappropriate binding of keys or channel bindings. The AR is not directly authenticated by a host, rather a host determines that the authenticator has been authorized by the AAAS by confirming that the AR has the same AAAS provided keying materials. Thus, this mechanism suffers from

possible MitM attack as reported in [7]. However, following the guidelines of [3], [51] and [2], the MitM attack can be prevented and secured key transportation using AAA protocol can be achieved.

5.3 EAP-IKEv2 Protocol

In this thesis, we are considering EAP-IKEv2 [107], an EAP method that is based on the Internet Key Exchange Protocol version 2 (IKEv2) [66]. It provides mutual authentication and session key establishment between an EAP peer and an EAP server. We have selected the EAP-IKEv2 method to demonstrate the encapsulation over IGMP-AC, due to its support of a variety of authentication techniques: asymmetric key pairs, passwords and symmetric keys. Most of the EAP methods are confined to a specific authentication mechanism. Moreover, EAP-IKEv2 follows the header format of IKEv2, which is a widely used authentication protocol. Thus, it will ease the deployment of EAP-IKEv2 method for the network administrators and the security personnel.

```
1. P <- S: EAP-Request/Identity
2. P -> S: EAP-Response/Identity(Id)
3. P <- S: EAP-Req (HDR, SAs, KEs, Ns)
4. P -> S: EAP-Res (HDR, SAp, KEp, Np, [SK{IDp}])
5. P <- S: EAP-Req (HDR, SK{IDs, AUTH})
6. P -> S: EAP-Res (HDR, SK{IDp, AUTH})
7. P <- S: EAP-Success
```

Figure 18: EAP-IKEv2 Protocol

The peer-to-peer version of the EAP-IKEv2 protocol is shown in Figure 18, where `P` is the peer (EU in IGMP-AC) and `S` is the server (AAAS in IGMP-AC). The authenticator (AR or NAS) and some optional exchanges are not shown here. The first two messages are defined in [2], and are standard EAP Identity `Request` and

Response messages. HDR is the EAP-IKEv2 header, which contains the Security Parameter Index (SPI). Two separate SPIs are chosen by S and P for each direction on a per protocol run basis. In message 3, S informs P about the set of cryptographic algorithms it prefers (SAs), Diffie-Hellman payload (KEs) and its Nonce (Ns). Similar information is sent by P in message 4. The identity of P, IDp is sent in encrypted format in message 4 in the case of symmetric key authentication only. This payload is optional to send in the EAP-IKEv2 specification.

At this moment, a set of keys (e.g., SK_e, SK_a, etc.) is computed by both S and P according to [66]. These keys are used to generate the AUTH payload and the key, SK is used in the 5th and the 6th messages. Here, SK{x} means x is both encrypted (using SK_e) and integrity protected (using SK_a). Two sets of SKs are calculated and are to be used in the opposite directions. IDs and IDp are the identities of the server and the peer respectively. AUTH is the Authentication payload and is defined in [66].

When P receives message 5, it authenticates S. The actual authentication mechanism depends on the type of authentication used (asymmetric key pairs, passwords or symmetric keys) and local policies. If all checks succeed, P will send message 6. On receiving this message, S will authenticate P. Finally, an EAP-Success message is sent to P.

At the end of a successful EAP-IKEv2 run, S and P will have mutually authenticated each other, and will be able to generate a Master Session Key (MSK) and an Extended Master Session Key (EMSK). These keys can be used to secure any further communication between these two parties.

In Figure 19, we have shown the complete message sequence of a join scenario, where an EU joins a Secured group using IGMP-AC. We can draw a similar message sequence when an EU is authenticated using EAP-IKEv2 method in pass-through mode. At that time, the first EAP-IKEv2 message, EAP-Request/Identity should be sent by the AR, instead of the AAAS. In Figure 19, the eap_req1 message

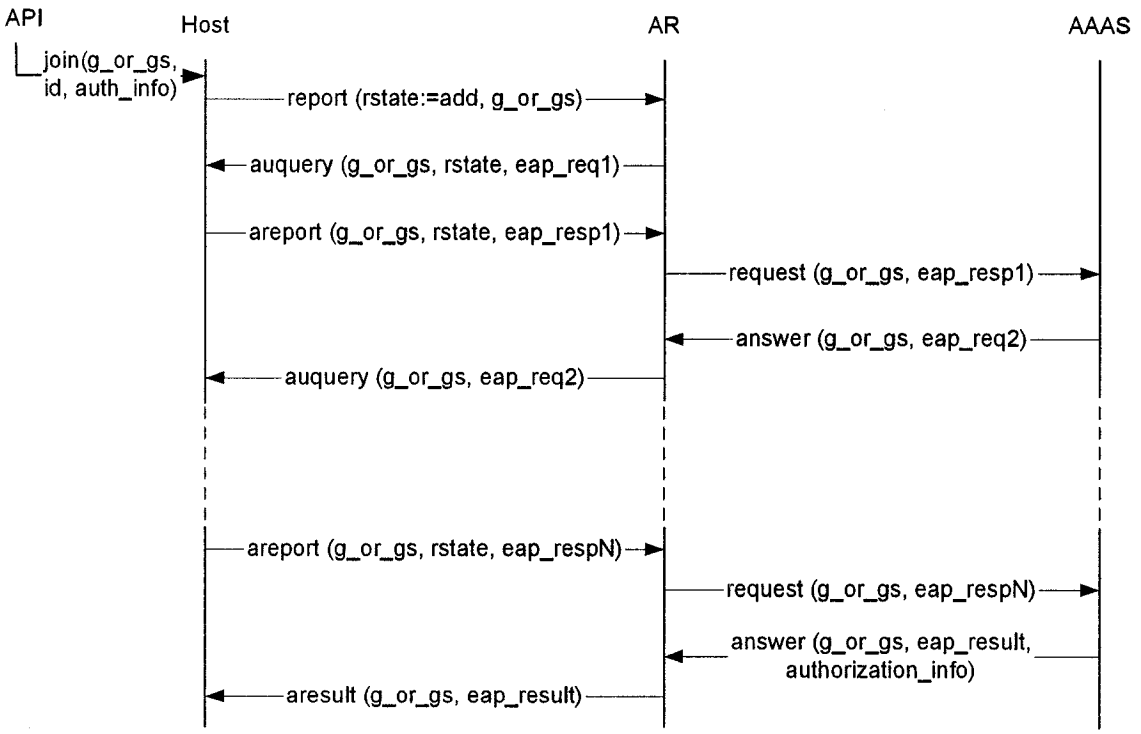

Figure 19: Message Sequence for End User Join using IGMP-AC

(inside the first `auquery`), which is sent from the AR to the host, should be replaced by the `EAP-Request` message. Similarly, the second EAP-IKEv2 message, `EAP-Response/Identity(Id)` will replace the `eap_resp1` inside the first `areport` message (from the host to the AR) and also the `eap_resp1` inside the first `request` message (from the AR to the AAAS). In this way, three pairs of EAP `Request` and `Response` messages will be required for a successful authentication. Finally, after three complete round trips, if the EU is authenticated and authorized, an `EAP-Success`, which replaces the `eap_result` in the last `answer` message of Figure 19, will be sent.

5.4 Validation of EAP-IKEv2 Method using AVISPA

We have modeled the EAP-IKEv2 method using AVISPA [11] to validate different security properties that EAP-IKEv2 is designed for. Previously, we have used SPIN [48] to verify the logical consistency of the IGMP-AC protocol. However, SPIN is not designed for verification of security protocols, and hence, it is not applicable for validating the security properties of the EAP-IKEv2 method. On the contrary, AVISPA is a push-button tool with industrial-strength technology for the analysis of different Internet security protocols and applications. It is being used by the developers of different security protocols and by academic researchers also. The security protocols standardized by the IETF have been analyzed by the AVISPA community, and some of the protocols have even been found to be flawed.

An AVISPA model of the EAP-IKEv2 method is available in [104], which is a peer-to-peer model (without the pass-through authenticator). We have started our validation with this model. We have found an attack on it and extended it to remove the flaw by adding some messages. Finally, we have modified the peer-to-peer model to a pass-through model by introducing the authenticator. The validations of the peer-to-peer model and the pass-through model are presented in section 5.4.2 and 5.4.3 respectively.

5.4.1 Security Properties of the EAP-IKEv2 Method

A number of security properties that are met by the EAP-IKEv2 method have been listed in [107]. We are not using all the functionalities (e.g., fast reconnect) of the EAP-IKEv2 method, and thus, not all those properties are relevant to our model. Here, we are mentioning some of the important properties:

Mutual Authentication: EAP-IKEv2 provides mutual authentication by the AUTH payload, which is either signed (for asymmetric key authentication) or hashed (for symmetric key or password authentication) by P and S. The construction of the AUTH payload is described in [66].

Integrity Protection: This protection is performed after authentication is completed and is accomplished by two Checksum Data fields. At the end of each EAP packet a Checksum is appended and the other one is calculated during the generation of SK{x}.

Replay Protection: This is achieved by two nonces, Ns and Np. These nonces are assumed to be fresh and unpredictable, and are used in calculating the AUTH payload.

Confidentiality: EAP-IKEv2 provides confidentiality to certain fields, which are included in SK{x}. It also provides identity confidentiality with active adversary, if SK{IDp} is not included in message 4. Otherwise, an active adversary will get P's identity.

Key Strength: EAP-IKEv2 establishes two session keys, MSK and EMSK with a variety of key strengths (maximum 512 bits per key). Effective key strength depends on a number of factors: the authentication credentials used, the negotiated ciphersuite, the Diffie-Hellman group used, etc.

Dictionary Attack Resistance: These types of attacks are common in password based authentication. EAP-IKEv2 resists offline dictionary attacks, since P uses his/her password only after S is authenticated successfully using any other authentication mechanism. To prevent online dictionary attack, S should log failed peer authentication events and should take action in case of consecutive failed peer authentication attempts within a short period of time.

Session Independence: Even if a passive adversary stores all the conversations of an EAP-IKEv2 session, she will not be able to compute the MSK or the EMSK.

5.4.2 The Peer-to-Peer Model

The EAP-IKEv2 protocol shown in Figure 18 is presented in Alice and Bob notation in Figure 20 to validate its security properties using AVISPA. The name of the messages (i.e., `EAP-Req` and `EAP-Resp`) and the entire header payload (i.e., `HDR`) have been removed in this version as these portions of a message do not carry any security information. However, it is still possible to capture all the important security properties using the simplified version and validate these properties with AVISPA. In Figure 20, "." means concatenation, {x}_K means x is signed by the key, K, and inv{K} stands for the private key corresponding to K.

5.4.2.1 Limitations of the Peer-to-Peer Model

Due to the simplifications, some limitations will be present. It is expected that these limitations will not affect the finding of a possible flaw or attack in the original protocol. Here, we have listed the limitations of the peer-to-peer AVISPA model of the EAP-IKEv2 protocol:

1. In this peer-to-peer model, the authenticator (NAS) and the back-end server (AAAS) are assumed to exist inside the same entity. This issue will be addressed in our pass-through model.

2. EAP-IKEv2 provides different use cases in terms of authentication mechanism (asymmetric key, symmetric key, password based). In this model, only asymmetric key or digital signature based authentication has been validated.

3. The entire `HDR` payload (which contains the `SPI` and the `Message ID`) has been omitted as it does not carry any sensitive information that is related to security measures.

```
1. P <- S: request_id
2. P -> S: respond_id.P
3. P <- S: SA.KEs.Ns
4. P -> S: SA.KEp.Np.[{IDp}_SKp]
5. P <- S: {S.{AUTHs}_inv(Ks)}_SKs
6. P -> S: {P.{AUTHp}_inv(Kp)}_SKp
7. P <- S: success

IDp   : Identity of P
SA    : Cryptographic algorithms from S
SA    : P's selected cryptographic algorithms

KEs   : exp(G, DHs), S's Diffie-Hellman value
DHs   : S's Diffie-Hellman exponent (nonce of S)

KEp   : exp(G, DHp), P's Diffie-Hellman value
DHp   : P's Diffie-Hellman exponent (nonce of P)

Ns    : nonce of S
Np    : nonce of P

SKp   : hash(Ns.Np.exp(exp(G,DHs),DHp))
SKs   : hash(Ns.Np.exp(exp(G,DHp),DHs)), here SKp = SKs

AUTHs : SA.KEs.Ns.Np
AUTHp : SA.KEp.Np.Ns
```

Figure 20: Alice and Bob Notation of EAP-IKEv2 Protocol

4. There is no choice of cryptographic algorithm, and it is assumed that `SA = SAs = SAp`. This will eliminate the possibility of finding any downgrade or chosen cipher suite attack. This simplification will not affect the validation as this attack is already eliminated by the EAP-IKEv2 specification, where `S` will first propose a set of cryptographic algorithms (by `SAs`) it supports and `P` is bound to chose one (by `SAp`) from the set.

5. In [66], a key seed, named `SKEYSEED` has been calculated first using the following equation:

```
SKEYSEED = prf (Ns | Np | exp(exp(G, DHs), DHp))
```

Here, prf is a pseudo random function. A set of seven keys is also calculated using SKEYSEED, Ns, Np, and SPI values. Among these keys, SK_es, SK_ep, SK_as, SK_ap are used in generating SK{x}. In this model, the generation of keys and SK{x} has been simplified. Instead of using two sets of keys (total four keys) in both directions, only one key (SKs = SKp) is used for encryption, which is exactly equal to the SKEYSEED. However, this simplification will only affect the key strength property, which is not falsifiable by AVISPA in any case.

5.4.2.2 Security Goals

From the discussion of the previous section, it can be stated that AVISPA is not able to capture all the security properties of the EAP-IKEv2 protocol. However, the properties it captures and validates are the important ones. Thus, the following security properties are targeted for validation by the peer-to-peer model:

- Mutual authentication: P and S strongly authenticate each other on Ns and Np respectively by checking the value of the AUTH payload.
- Key establishment: At the end of a successful protocol run, P and S will be able to derive the keys SKp and SKs respectively, which have equal values.
- Confidentiality: Throughout the protocol run, the keys, SKp and SKs must be secret from any adversary.
- Replay protection: The model should be resistant to replay attack. To find the existence of any such attack parallel sessions have been invoked in our model.

The HLPSL specification of the peer-to-peer model is constructed by following the simplified protocol of Figure 20. This specification has two `roles`, one for the server and the other for the peer. The built in function, `hash_func` of AVISPA is used as the function, `prf`. An active intruder, who can play both roles of the server and the peer simultaneously has been introduced. Two parallel `sessions` of the protocol have been executed to find the replay attack. For validation the `HLPSL` code is transformed to `IF` format using the translator `HLPSL2IF`. Next, we have used all four back-ends of AVISPA. The first two, `OFMC` and `CL-AtSe` have produced similar MitM attacks. The other two, `SATMC` and `TA4SP` have reported `NOT_SUPPORTED` and produced `INCONCLUSIVE` results.

5.4.2.3 Finding the Attack

The attack reported by the `OFMC` back-end is shown in Figure 21. For space consideration and similarity of the attacks, the one produced by the `CL-AtSe` is not shown here. In this attack, the intruder is able to generate a MitM attack by opening two parallel sessions with `S` and `P`. The intruder convinced `P` that he was talking with `S`, when in fact `S` did not participate in the same session. Rather, the intruder relayed messages to `P` from the session she opened with `S`. The intruder is not able to gain anything directly by performing this attack, as she has not accessed the secret key that `P` believes himself to have established with `S`. Thus, she cannot exploit the flaw by intercepting any message that `P` sends to `S`. However, still this is a flaw of the protocol and it must be fixed with care. Sometimes, a flaw appears harmless and remains so for a while. However, it may shift to an unsafe one due to further extension or modification of the flawed protocol. The attack we have found is analogous to the one reported in [83] and originates from the use of IKEv2 for authentication.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /nfs/thesis/s/salek_is/avispa-1.1/testsuite/results/ike.if
GOAL
  authentication_on_ns
BACKEND
  OFMC
COMMENTS STATISTICS
  parseTime: 0.00s
  searchTime: 0.88s
  visitedNodes: 278 nodes
  depth: 5 plies
ATTACK TRACE
  i -> (s,10): start
  (s,10) -> i: request_id
  i -> (p,3) : request_id
  (p,3) -> i : respond_id.p
  i -> (s,10): respond_id.i
  (s,10) -> i: SA(3).exp(g,DHs(3)).Ns(3)
  i -> (p,3) : SA(3).exp(g,DHs(3)).Ns(3)
  (p,3) -> i : SA(3).exp(g,DHp(4)).Np(4)
  i -> (s,10): SA(3).exp(g,DHp(4)).Np(4)
  (s,10) -> i: {s.{SA(3).exp(g,DHs(3)).Ns(3).Np(4)}_inv(ks)}
               _(f(Ns(3).Np(4).exp(exp(g,DHp(4)),DHs(3))))
  i -> (p,3) : {s.{SA(3).exp(g,DHs(3)).Ns(3).Np(4)}_inv(ks)}
               _(f(Ns(3).Np(4).exp(exp(g,DHs(3)),DHp(4))))
  (p,3) -> i : {p.{SA(3).exp(g,DHp(4)).Np(4).Ns(3)}_inv(kp)}
              _(f(Ns(3).Np(4).exp(exp(g,DHs(3)),DHp(4))))
```

Figure 21: Attack Reported by the OFMC Back-end

5.4.2.4 Securing the Peer-to-Peer Model

The first modification we have made is addition of Message ID (MID) in the 5th and the 6th messages. As per [107], MID is sent as a field of the HDR payload in the 3rd to the 6th messages. However, in the 3rd and the 4th messages, HDR is not cryptographically protected. Thus, we have not added MID in these messages. In the 5th and the 6th messages, the integrity of MID is protected using the keys derived from SKEYSEED. We have modified these two messages accordingly:

```
5. P <- S: hash{MID.SKs}.{S.{AUTHs}_inv(Ks)}_SKs
6. P -> S: hash{MID.SKp}.{P.{AUTHp}_inv(Kp)}_SKp
```

Here, P sends the 6th message with the same MID he has received in the 5th message. P and S always send a Response with the same MID they receive in the preceding Request message. Thus, a specific value is assigned to MID for a pair of Request and Response messages. In our model, MID is equal to one.

In spite of this modification, AVISPA reported the same MitM attack. This is quite understandable, as the function of MID is to prevent replay attack and the attack we want to eradicate is a MitM attack. To solve this problem, we have used the optional {IDp}_SKp payload in message 4 of Figure 20. According to [107], this payload must be included in case of symmetric key authentication only. The reason for this is an active adversary is able to get the identity of P, IDp if this payload is included. It should be clarified that the EAP identity (P), sent in the 2nd message, is not an authenticated identity, and is supposed to be different from IDp. The EAP identity (P) is known to everyone and not subjected to the identity confidentiality attack. We are suggesting to use the EAP identity (P) instead of IDp in the 4th message in the case of asymmetric key and password authentication. In case of symmetric key authentication, IDp must be used following the specification [107].

We have modified the AVISPA model by adding {P}_SKp in the 4th message. We have used the two back-ends, OFMC and CL-AtSe to find the attack. This time, we have received the expected results as both back-ends reported SAFE. If we look carefully at the ATTACK TRACE of Figure 21, we will get the answer. The intruder is communicating with S using its own identity, i. She has not produced any encrypted or authenticated message. She receives a message from P, and just relays it to S. To convince S of her identity, she must send {i}_SKi in message 4. She does not have access to any secret key nor she can generate it. As a result, the intruder is not able to supply the appropriate 4th message to S. Thus, it is not possible for the intruder to mount any MitM attack.

5.4.3 The Pass-through Model

We are now prepared to develop our final model by extending the peer-to-peer one to a pass-through model. This will be accomplished by replacing the eap_reqs and the eap_resp messages with the simplified EAP-IKEv2 messages of Figure 20 (with appropriate modifications explained in the previous section to preclude MitM attack). The replacement procedure is explained in section 5.3 with the example of joining an EU to a Secured group (see Figure 19). We have made the appropriate changes in our HLPSL code by adding a new role of the NAS (or authenticator), which will do nothing but relay messages from S to P and vice versa. The intruder ability is extended also to perform the role of the NAS. Thus, an active intruder can play any role of server, peer or NAS, and even play multiple roles simultaneously. Two parallel sessions of the protocol have been executed to discover any replay attack.

First, we have verified our HLPSL specification using Security Protocol ANimator (SPAN) [100], a tool that interactively builds Message Sequence Charts (MSC) from HLPSL code. SPAN also provides an intruder mode that interactively builds attack traces from the HLPSL specification. Thus, by executing the Protocol Sim-

ulation option of SPAN we are able to generate a full trace or MSC of our specification. It confirms that the entities are communicating properly and the model is working as expected. Next, we have used all four back-ends of AVISPA to find an attack if any exists. The first two, `OFMC` and `CL-AtSe` have reported `SAFE` for `BOUNDED_NUMBER_OF_SESSIONS`. The other two, `SATMC` and `TA4SP` have reported `NOT_SUPPORTED` and produced `INCONCLUSIVE` results. Due to the complexity of the model, we have to run `OFMC` with bounded depth. Finally, we can conclude that the pass-through model that we have developed is free from the attacks that AVISPA is able to find. Hence, the security goals we have mentioned in section 5.4.2.2 have been validated.

Chapter 6

Sender Access Control

Our research goal—adding AAA functionalities to the present multicast model—is divided into receiver access control and sender access control. We have already presented our receiver access control mechanism in detail in the previous chapters. This chapter will explain how sender access control would be deployed with necessary extensions.

Sender access control is different from source authentication or data origin authentication. Source authentication allows a receiver to validate the identity of the source, i.e., to confirm that a received packet is coming from a source that belongs to the same multicast group to which the receiver has joined. A number of group key management protocols (e.g., SIM-KM [88]) that provide source authentication with great variety, have been developed for multicast applications. However, we cannot prevent an adversary from spoofing a legitimate sender address and sending bogus packets to flood the DDT. Thus, the DoS attack exists even in the presence of a group key management protocol. A DoS attack is one of the severe attacks that is not possible to completely eliminate. However, we can take different prevention steps to minimize this attack. Multicast suffers from DoS attack more severely than any other attack due to its amplification of data packets enroute. The best way to prevent

such an attack is to establish a checkpoint at the entry point or Access Router (AR) of the DDT if the Core Routers (CR) are being trusted.

First, we will briefly present some related work and will compare these methods. The proposed architecture will be explained with the underlying threat model and the role of different entities. We have deployed Protocol for Carrying Authentication for Network Access (PANA) [33], a link-layer agnostic protocol that encapsulates EAP packets to authenticate a sender. Finally, different benefits of our architecture will be listed.

6.1 Related Work

In the literature, only a few attempts have been found to provide sender access control ([12], [55], [102] and [111]). The first two methods ([12] and [55]) had design goals similar to the architecture we have presented in Figure 12. The later two methods ([102] and [111]) attempted to enforce sender access control by controlling the construction of the DDT of a multicast group only for the authenticated/authorized senders.

6.1.1 Authentication Stamp

In [12], Ballardie and Crowcroft first defined the sender (and receiver) access control mechanism for Core Based Tree (CBT) [13], where an Authentication Server (AuthS) authenticates each member and supplies an Authorization Stamp for successful authentication. A sender includes this stamp in each multicast packet and digitally signs the packet. A control router from DDT will forward these signed packet to the AuthS to check the sender identity. The AuthS also adds a lifetime to each Authorization Stamp to prevent replay attack.

6.1.2 Challenge-Response Method

Ishikawa *et al.* [55] deployed AAA architecture to authenticate each user (sender and receiver). They proposed a challenge-response protocol with shared secret for sender authentication. In this model, a host willing to send data should communicate with the nearest ingress router, which will start a challenge-response session with the sender. The ingress router collects authentication information and forwards it to a back-end AAA Server (AAAS) to perform the actual authentication.

6.1.3 Keyed Hierarchical Multicast Routing Protocol (KHIP)

Shiels and Garcia-Luna-Aceves [102] designed the Keyed Hierarchical Multicast Routing Protocol (KHIP), an extension of the Hierarchical Multicast Routing Protocol, which protects the multicast routing protocol from untrusted routers and provides sender and receiver access control as well. KHIP authenticates and authorizes hosts and trusted routers, and distributes encryption keys. In each sub-branch of the DDT, a packet is processed by the root router of the sub-branch, which will introduce delay in delivering data and high deployment cost. The model burdens the Designated Routers (DR) and the CRs with multiple processing of digital signatures for hosts' subscriptions, and hence, is vulnerable to possible DoS attack [68].

6.1.4 Sender Access Control List (SACL)

Wang and Pavlou [111] developed a scalable sender access control policy mechanism for bi-directional shared trees to check and discard unauthorized data when they arrive at an on-tree router. All senders must register with the core first. When an on-tree router forwards a registration message to the core, it adds the downstream sender address to its local Sender Access Control List (SACL). On receiving a registration

message, the core contacts a Source Authorization Server (SAS) for deciding whether to accept the new sender. If the join is approved by the SAS, the core will send back to the sender an *activating packet*, which will activate the previously added sender address in the SACL of each on-tree router. A data packet will be forwarded to the upstream interface if it comes from a sender address having an entry in the SACL and if the packet comes from the same interface as the one recorded in the entry. However, a data packet coming from the upstream interface will always be forwarded to the downstream interfaces with group state without performing any verification.

6.1.5 Summary of Different Methods

In Table 4, we have summarized how these four methods fulfill different requirements that we mentioned earlier (see section 3.3) a sender access control architecture should provide. None of them provides accounting and meets all the requirements. They have other limitations such as being confined to a specific authentication mechanism, dependency on the routing protocol, not free from different attacks and slow packet delivery due to processing of authentication information in many places. Hence, it is clear that a sender access control framework is needed, which will satisfy the requirements we have identified.

6.2 Proposed Architecture

We have listed a set of requirements for multicast access control architecture in section 3.3. While developing the sender access control architecture we have focused on meeting those requirements. The architecture we have developed was first presented in [59], which is shown in Figure 22. The Content Server (CS) will reside in the CP,

Table 4: Summary of Existing Methods with respect to Required Properties

Method	**AAA functions**	**Authentication mechanism**	**Possible attacks**	**Overhead**	**Routing protocol**	**Intra or inter domain**
Auth. Stamp [12]	Authentication Authorization	Digital signature	DoS attack at control router	High—control router forwards each packet to AuthS	CBT	Both
Challenge response [55]	Authentication Authorization	Challenge-response using password	Dictionary attack, source address spoofing	Low—between host and ingress router	Any protocol	Intra-domain
KHIP [102]	Authentication Authorization	Digital signature + encryption	DoS attack at DRs and core	Medium—each sub-branch root decrypts and re-encrypts data	CBT, OCBT	Both
SACL [111]	Authorization	No explicit authentication	Replay attack, source address spoofing	Medium—multiple routers on DDT check SACL	Any bi-directional	Both

which will send multicast data through its nearest AR, AR1. The CRs will construct the DDT using one of the multicast routing protocols. A receiver will join a secured group using the IGMP-AC [60] protocol and his/her nearest AR will extend the DDT up to that AR. Thus, multicast data will reach from a sender to multiple receivers by traveling multiple hops of the DDT.

6.2.1 Threat Model and Assumption

In our model, we are assuming that a host can be compromised while a router is a trusted entity. Thus, a compromised host may capture, modify and replay any data packet inside a subnet. The proposed architecture must defend against such attacks. A router is usually less vulnerable to attack and expected to behave correctly. This is a reasonable assumption for an intra-domain group, given that the network adminis-

trators have better control over the routers and will be able to detect a compromised router more quickly than a compromised host. However, for an inter-domain group, there will be routers in the path that are not under the control of the local administrator; therefore, it is harder to believe that the upstream routers in an adjacent domain are to be trusted. When we present the inter-domain architecture in Chapter 8, we will show a solution that considers compromised routers and hosts as well. This solution will be applicable to the single-domain case, thus removing the assumption that local routers are to be trusted.

6.2.2 Role of Different Entities

A successful sender access control requires interactions of three entities: a host or sender, the closest AR to the sender and the AAAS. This AR is commonly known as Designated Router (DR) and will act as the NAS. With the help of a AAAS, it will authenticate and authorize a sender. It is important to clarify that the AR will perform access control only if the sender wants to send data to a secured group. As we have mentioned earlier, a secured group could be easily identified using the group address, if the IANA [53] assigns a range of addresses from the Class D IP address (224.0.0.0 to 239.255.255.255) block to be used by the secured multicast groups. Hence, the routers, the AAA nodes and the end hosts will easily distinguish a secured group from an open group.

The architecture shown in Figure 22 works for intra-domain multicast groups only. This is a restrictive assumption, and in reality, the receivers may be distributed in different domains or ASes. An architecture is always dependent on the specific business model and the trust relation between the CP and the NSP. We will extend the intra-domain architecture in section 8.3 for inter-domain multicast groups.

When a sender wants to send data to a secured multicast group, she will com-

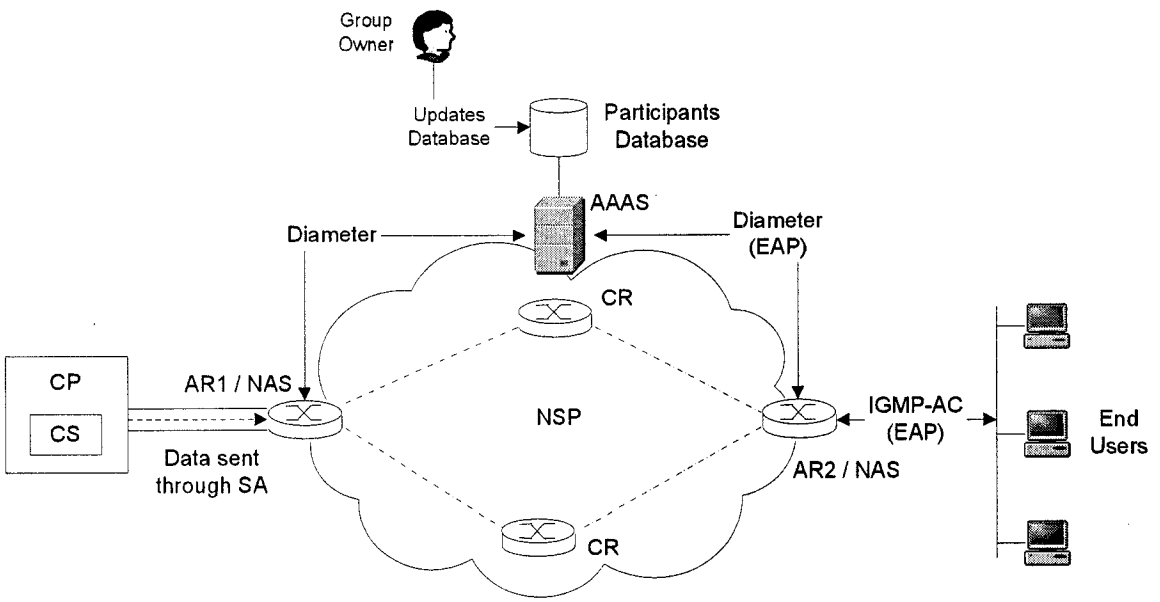

Figure 22: Sender Access Control Architecture

municate with the one-hop AR (with AR1 in Figure 22) to start an authentication protocol session. We have presented two different ways for this initial communication between the sender and the one-hop AR in the following section. Through the authentication protocol session, the sender will send her authentication information to AR1, which will be forwarded to the AAAS. The AAAS will authenticate the sender and for successful authentication, relevant authorization and accounting information will be sent to AR1 along with the authentication result. The content of the authorization information depends on a specific application. For example, it may be the duration of time up to which the sender is allowed to send data to a multicast group. The accounting information will provide direction for AR1, how it will keep track of the sender behavior (e.g., the duration of time or the amount of data). AR1 will communicate with the AAAS using the Diameter [18] protocol. The exact construction of the Diameter messages is out of the scope of this research. However, we have already mentioned that all data in Diameter messages are delivered in the form of Attribute Value Pairs (AVP), and it is possible to extend the Diameter protocol by defining new Attribute Value Pairs.

If the group is an open group, then given that there is no restriction on sending data to an open group, a sender will send data to such a group as soon as a data packet is generated. On receiving a data packet for an open group, AR1 will follow its local policy and thus, either forward the packet or discard it (e.g., in case there is no entry in the routing table to reach the destination group address).

If AR1 receives a non-empty multicast data packet (destined to a secured group) from a sender, who is not authenticated yet, AR1 can either silently discard the packet or start a security protocol session to authenticate the sender.

Once the sender is authenticated by the AAAS, an authentication key will be established between the sender and AR1 to accomplish per-packet cryptographic protection. The sender must send authentication information with each multicast packet, which will aid AR1 to authenticate each packet before forwarding it to the DDT. This authentication key will be maintained by the sender and AR1 as long as the sender has authorization to send data.

6.2.3 Initiating an Authentication Protocol

The sender will send AR1 her identity in the format of a Network Access Identifier (NAI) [1] (e.g., `bob@example.com`) and the destination address of the secured group that she intends to send data to. This will trigger an authentication protocol. It should be noted that the sender identity consists of two parts: a user name (i.e., `bob`) and a domain name (i.e., `example.com`). The sender identity will be required for inter-domain sender access control, and therefore, it will be discussed again in Chapter 8.

Given that there exists no standard protocol using which the sender may send this information, we have formulated two different ways to solve the problem:

- Unlike the case for receivers, where IGMP is used by the receivers to explicitly join a multicast group, a sender does not go through any joining process. However, if we had any protocol like IGMP for multicast senders, we could have better control over the senders. Unfortunately, the IETF has not standardized any such protocol yet. An Internet Draft [114] in the PIM Working Group has been written to separate the control-plane and the data-plane of the PIM-SM protocol. The Internet Draft has also introduced different messages including `Sender-Request` (to be sent by a sender to the one-hop AR to show her interest in sending multicast data) and `DR-ACK` (to be sent by the AR to a sender to notify that the sender can send multicast packets now). It is to be noted that the ID has been written to improve the register mechanism of the existing PIM protocol rather extending the general multicast architecture. Therefore, it would not be deployable directly in our architecture. However, we may exploit the idea of this ID and may develop a general solution of sender join in multicasting. Hence, using that protocol, the sender would be able to send necessary information (i.e., sender identity, secured group address) to the AR.

- The second option we have studied depends on the authentication protocol that would be used to authenticate the sender. In general, an authentication protocol is designed in the client-server communication pattern (in case of a one-way authentication) where the server authenticates the client, and the protocol might be initiated either by the client or by the server. We have deployed PANA [33] for sender authentication in our architecture and it could be initiated by the sender also. Hence, the sender would be able to send necessary information to the AR inside the initial message of a PANA session. This may need minor modification of the `PANA-Client-Initiation` message.

6.2.4 Authentication using PANA

One of our design goals is to develop a flexible authentication framework to support various authentication mechanisms. Hence, the administrator of the network will have freedom to pick the appropriate authentication mechanism. The Extensible Authentication Protocol (EAP) [2] will be a good choice for this purpose.

EAP provides a wide range of authentication mechanisms through different EAP methods. An EAP method defines its own messages, which are sent inside EAP packets. However, EAP does not run directly over the IP layer, and a EAP lower-layer is required. To overcome this limitation, the IETF has standardized PANA [33], a network access authentication protocol that works as an EAP lower-layer. PANA is a client-server protocol that transports EAP authentication methods encapsulated inside EAP packets between a client node and a server in the access network. We have discussed the PANA framework [33] briefly in section 2.5. The PANA framework consists of four entities: PANA Client (PaC), PANA Authentication Agent (PAA), Authentication Server (AS) and Enforcement Point (EP).

PaC is the client entity of PANA that interacts with the PAA in the authentication process using the PANA protocol [33]. The server implementation of PANA is the PAA, which consults an AS for authentication and authorization of a PaC. The PAA resides on a node that is typically called a NAS in the access network while the AS is a conventional backend AAAS. The EP, which is updated with the attributes of the authorized PaCs from the PAA, is the access control implementation that allows (blocks) data traffic of authorized (unauthorized) PaCs.

The PANA framework exactly resembles our proposed architecture, and the functionalities PANA provides match the requirements we have identified in our design goal. How PANA will be accommodated in our architecture is shown in Figure 23. The host from which a sender is sending a multicast packet will be the PaC and the

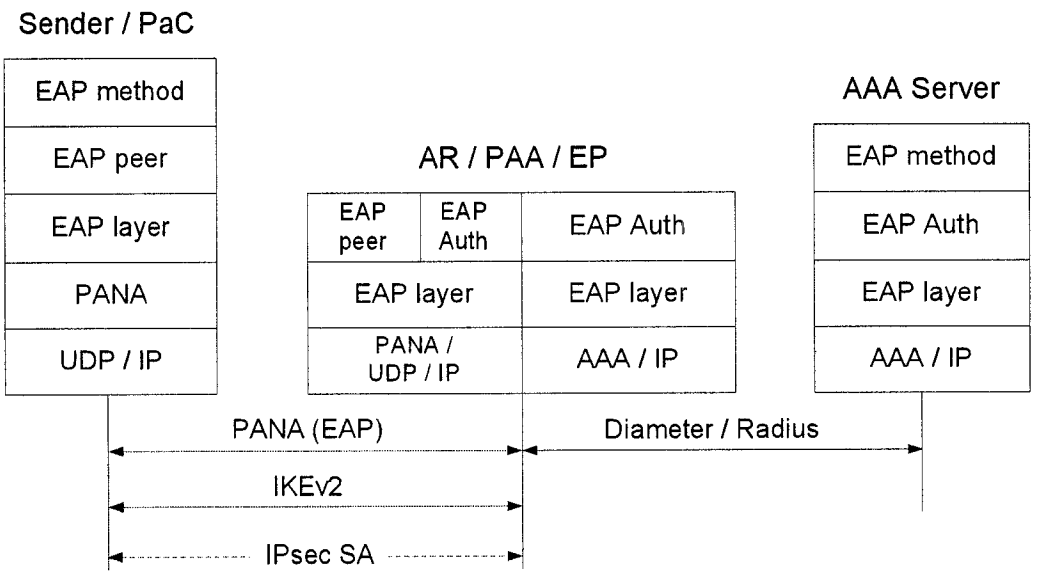

Figure 23: PANA Framework within Proposed Architecture

nearest AR to it will be the PAA. The AAAS will be the AS. Hence, when a sender will send an initialization message with her identity and the destination address of a secured group, a PANA session will start between the sender and the AR. PANA will add an extra layer between the UDP and EAP layers in the sender and the AR entities. There will be no PANA layer in the AAAS side, as it will communicate with the PAA using the Diameter-EAP protocol. Thus, PANA will be terminated at the PAA and EAP will be terminated at the AAAS, and for authentication any EAP method can be used. We are also assuming that the EP is co-located with the PAA (or AR), and thus, an API is sufficient for their communication.

A PANA session consists of five phases. In the following we have explained how the entities will behave in our architecture during these phases:

1. Handshake phase: The sender will send an initialization message (carrying her identity and the address of the secured group), which will trigger a new PANA session.

2. Authentication and authorization phase: Immediately following the handshake phase, EAP packets, carrying EAP method messages, will be exchanged be-

tween the sender and the AR. The AR conveys the result of authentication and authorization to the sender at the end of this phase.

3. Access phase: After a successful authentication and authorization the sender is allowed to send multicast data through the EP or AR. The cryptographic protection will be performed by the EP in this phase.

4. Re-authentication phase: If re-authentication is required, this phase will trigger an EAP re-authentication, which turns back to access phase again upon successful re-authentication.

5. Termination phase: The sender or the AR may choose to discontinue the access service at any time. At the end of this phase, the AR will gather accounting data for the sender and communicate these data to the AAAS (using Diameter) to update the sender's database.

6.2.5 Per-packet Cryptographic Protection by IPsec SA

A number of EAP methods provide mutual authentication and session key establishment between an EAP peer (sender) and an EAP server (AAAS). This key is known as `AAA-Key` or Master Secret Key (MSK) and can be exported to the PAA (or AR) using a AAA protocol [3]. In Figure 24, how different keys are generated using `AAA-Key` and how the IPsec SA is established between the sender and the AR are explained. In our model, the EP and the PAA reside inside the same entity, AR. A PAA may be connected with multiple EPs. On receiving the `AAA-Key` the PAA generates the `PaC-EP-Master-Key` for each EP. One way of calculating this key is,

```
PaC-EP-Master-Key = The first 64 octets of prf+(AAA-Key,"PaC-EP master
key"|Session ID|Key-ID|EP-Device-Id).
```

Here, "|" means concatenation of different fields and `prf+` is a pseudo-random

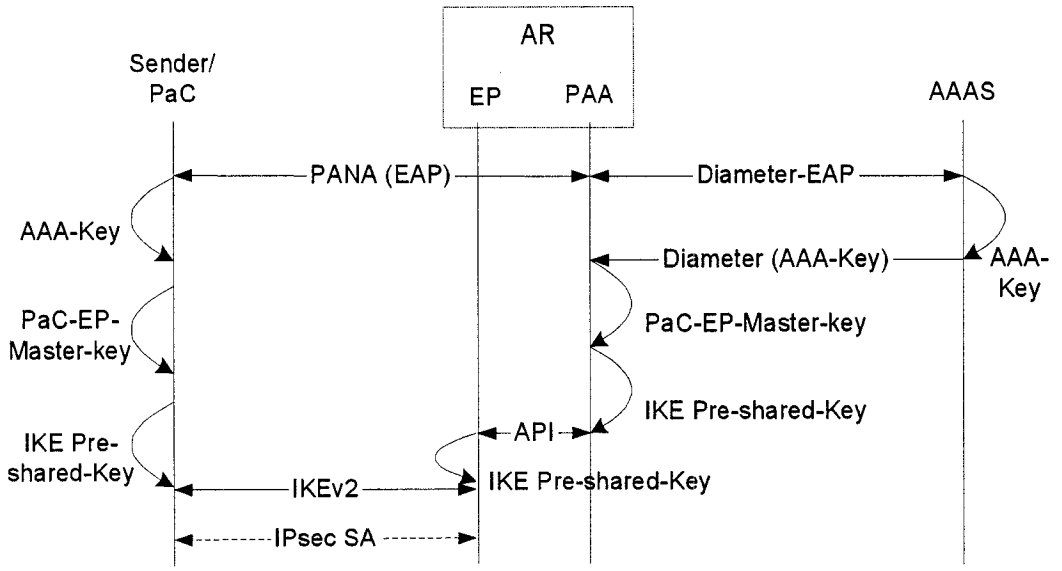

Figure 24: IPsec SA Establishment after PANA Session

function defined in [66]. `Session ID` must be globally and eternally unique. It identifies a PANA session. `Key-ID` is an identifier of a specific `AAA-Key`. `EP-Device-Id` is an identifier of the EP, and it can be an IP address, a MAC address, or an identifier that may not be carried in data packets but has local significance in identifying a connected device.

Next, the PAA generates the `IKE-Pre-shared-Key` using the following method,

```
IKE-Pre-shared Key = HMAC-SHA-1 (PaC-EP-Master-Key,"IKE-Pre-shared
key"| Session ID|Key-ID|EP-address).
```

Here, `Key-ID` identifies the `PaC-EP-Master-Key` within a given session and `EP-address` is the address of the EP that will participate in the IKE exchange. If the PAA controls multiple EPs, this provides a different pre-shared key for each of the EP.

The PAA will communicate the `IKE-Pre-shared-Key`, `Key-Id`, the device identifier of the PaC, and the `Session-Id` to the EP using an API before the IKE exchange begins. The sender will also calculate the `IKE-Pre-shared-Key` in a similar way. At

this moment, the sender and the EP will open an IKEv2 session for mutual authentication, and will eventually establish an IPsec SA, which will be used for per-packet cryptographic protection. Depending on the required security services (i.e., secrecy, integrity, authentication, replay protection, etc.) the appropriate SA protocol (i.e., Encapsulating Security Payload [71] or Authentication Header (AH) [70]) will be determined by the network administrator.

It should be noted that none of the protocols—IPsec, EAP and PANA—is defined for multicast communication. However, although we have used them in sender access control and in protecting multicast data, they have been used only for unicast communications between two entities.

6.3 Benefits of the Architecture

Our goal was to develop a generic framework in terms of authentication mechanism, routing protocol, multicast group types, etc. The existing IP multicast model will benefit by adopting the proposed architecture by a number of ways. It fulfills all the requirements of our design goal, and also provides some additional advantages. We have summarized the contributions of our proposal in the following:

Provides AAA functionalities

Our sender access control architecture provides all three AAA functionalities, specially, this is the first model (to our knowledge) to support accounting for sender behavior in IP multicast.

Per-packet cryptographic protection

We have decoupled sender authentication and per-packet protection. However, only after successful sender authentication, per-packet protection will be implemented and a strong trust relation exists (through `AAA-Key` exportation) be-

tween sender authentication (EAP authentication) and IKEv2 authentication that establishes the IPsec SA.

Minimum overhead and fast packet processing

Our model will add only a minimum workload to the AR of the sender side. The remaining routers of the DDT will still process a multicast packet without extra effort. We can further leverage the packet processing at the AR. When data are sent in encrypted format due to the deployment of a group key management protocol, an IPsec SA with AH instead of Encapsulating Security Payload (ESP) can be used.

Independent of routing protocol

On successful authentication of the first data packet, the AR will take necessary action (e.g., send the first unicast-encapsulated data packet to the RP in case of PIM-SM [32]) it would have taken without the presence of sender access control. Thus, the whole mechanism will remain transparent to the underlying routing protocol.

Serves both SSM and ASM groups

Although the list maintained by the AR and the query sent to the AAAS are `(S, G)` or source-specific type, there is no restriction for the address range of `G`. Thus, the model is applicable to both SSM and ASM groups.

Security services

Sender access control will prevent a number of attacks against the classical multicast model.

- A forged packet produced by an adversary will be detected (through integrity protection of SA) and discarded at the AR, and will not be forwarded to the DDT. This will stop a non-member from sending data to a secured group, and therefore, DoS attack will be reduced significantly.

- IPsec SA provides optional anti-replay by sliding window protocol. If an attacker stores a valid packet and replays it after a while, the anti-replay mechanism will detect and discard it.
- IPsec AH considers all immutable fields of IP header (including the sender address) to calculate the cryptographic hash value, and thus, provides source address authentication. This will prevent the source address spoofing attack.

Chapter 7

Policy Framework for Participant Access Control

In our research objective, access control is distributed over three functionalities: authentication, authorization and accounting. In our proposed architecture (see Figure 12 in section 3.4), these three functions will be enforced by the ARs. However, authentication and authorization decisions will be determined by the AAAS. The task of deciding the authenticity of an EU or a sender is relatively straightforward—if the requesting user sends the proper authentication token (e.g., password, hashed value, certificate) she would be deemed to be authenticated for the requested group activity. However, defining authorization and accounting depends on many factors: specific application, business model used, EU agreement with the CP or Group Owner (GO), etc. A simplified access control architecture has been shown in Figure 25, which presents the interactions of GO, Policy Server and AAAS. To deploy such an access control architecture, a policy framework is needed where the GO will update the multicast security policy in the Policy Server. The multicast security policy will be transported to the AAAS and consequently will be enforced at the NAS or AR. This chapter will present a policy framework [58] for specifying and enforcing the access

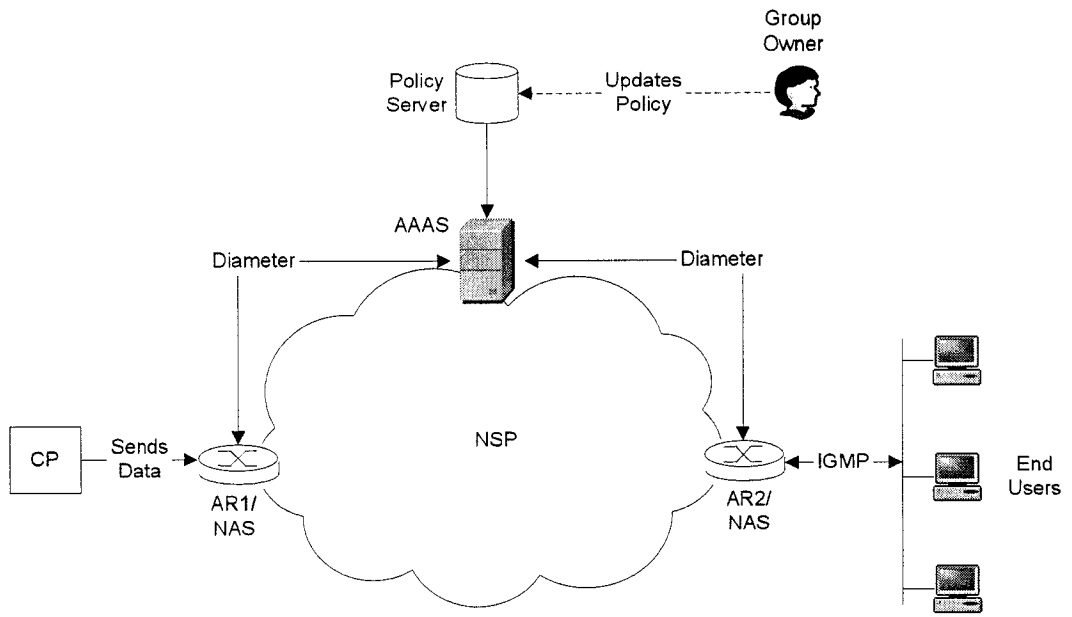

Figure 25: Simplified Proposed Architecture

control policy for multicast group participants.

A policy is a set of rules that dictates the behavior and implementation of a system or service. A security policy specifies the conditions to be fulfilled before allowing access to some restricted information or resources. This type of policy expresses formally the rules that must be enforced to meet the security requirements of a system. The multicast group security policy provides the rules for the operations of the elements of a group. This policy can be divided into access control policy and data control policy [87]. Access control policy deals with group participants' authentication, joining, leaving, billing, role delegation, etc. Data control policy specifies the rules that protect multicast data such as integrity, authentication, key management, etc.

In this chapter, first we will briefly present the IETF policy framework that has broad acceptance and use in the research community. We will present a summary with comparisons among the existing methods that we have found in the literature. Next, access control policies for some popular multicast-based applications will be

discussed. We have also defined a set of requirements that our policy framework should meet. The proposed policy framework will be discussed in detail. We are recommending the use of eXtensible Access Control Markup Language (XACML) [35] for policy specification, and Security Assertion Markup Language (SAML) [22] for policy transportation in our framework. Finally, we have developed an access control policy for an on-line course, which will be multicast for the registered students.

7.1 IETF Policy Framework

In the literature, different policy frameworks exist, among them, the IETF Policy Framework [115] is widely accepted by the research community. The framework is shown in Figure 26. It has mainly the following three components:

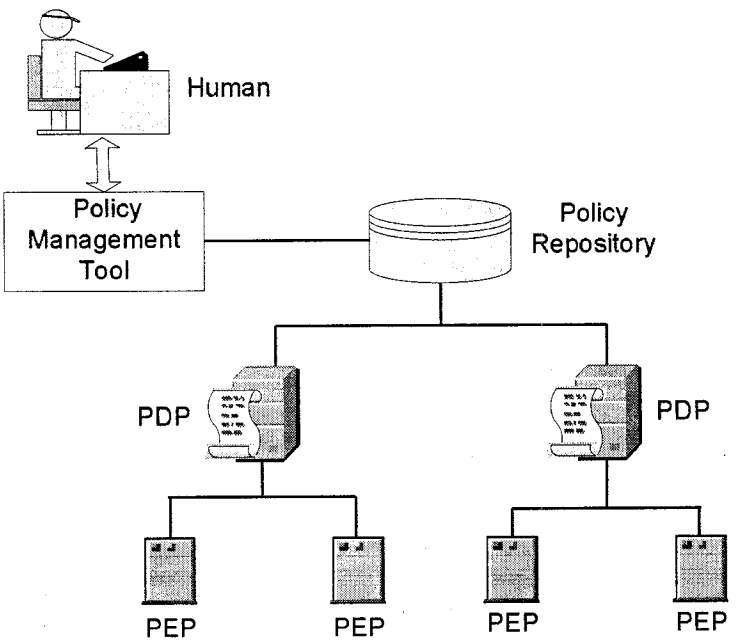

Figure 26: The IETF Policy Framework

Policy enforcement point (PEP): A PEP, co-located with a packet forwarding component (e.g., an AR), is responsible for enforcement of policy by executing

necessary actions when it encounters a packet. A PEP always runs on the policy aware node. It is the point at which policy decisions are actually enforced.

Policy decision point (PDP): Policy decisions are made primarily at the PDP, which produces policy decisions for itself or for other network elements that request such decisions. A PDP itself may make use of additional mechanisms and protocols to achieve additional functionality such as user authentication, accounting, policy information storage, etc.

Policy repository: A policy repository is the location where the policies defined for a domain are stored. When a PDP asserts a decision it will need to gather information. The policy repository stores this information. This can be any database, and/or AAA system.

The policy management tool is an interface for the network manager to update policies located in policy repository. The IETF also standardized a query-response signaling protocol, Common Open Policy Service (COPS) [28], for the exchange of policy information between a PEP and a PDP. When COPS is being used, a PEP acts as a client and sends a request to a PDP, which acts as a server.

The basic interaction between the components begins with the PEP. If the PEP receives any notification or request that requires a policy decision, the PEP creates a query and sends it to the PDP. A query carries one or more policy elements in addition to the admission control information (such as a flowspec or amount of bandwidth requested) in the original message or event that triggered the policy decision request. A policy element contains single units of information necessary for the evaluation of policy rules. The PDP returns the response (policy decision) and the PEP then enforces the policy decision by appropriately accepting or denying the request (that triggered the query).

7.2 Related Work

We have carried out a survey on security policy for multicast groups. A number of frameworks and policy specification languages have been proposed for network access control, while only a few of them addressed access control for multicast groups. In the following we have briefly discussed five existing methods that deal with data control policy and/or access control policy for multicast.

7.2.1 Group Secure Association Key Management Protocol (GSAKMP)

Multicast Security Policy is one of the functional areas of the Reference Framework [40] developed by the MSEC (Multicast Security) Working Group of the IETF. The Group Secure Association Key Management Protocol (GSAKMP) [41] further extends this framework by providing policy distribution, policy enforcement, key distribution, and key management for cryptographic groups. To define group policy, GSAKMP uses policy tokens [24] extensively, which are issued and signed by the GO.

7.2.2 XML Policy Representation

In [87], XML is used to represent multicast data control policies that fit within the MSEC Framework. How XML documents modeled on the same structure as X.509 certificates can be created, edited and verified using available XML editors are also described. Finally, some critical data control policies are identified, and as an example, an XML representation of policy that uses Scalable Infrastructure for Multicast Key Management (SIM-KM) [88] (a secure key management scheme using proxy encryptions) to provide security for multicast data transmission has been shown.

7.2.3 Antigone

Antigone [82] provides a framework with flexible interface for the definition and implementation of a wide range of secure group policies for secure group communication. It has a set of *mechanisms* through which the developers may select a policy that best addresses their security criteria and needs. Policies are implemented by the composition and configuration of these mechanisms, which are comprised of a number of micro-protocols. Antigone classified group security policies in different dimensions: session rekeying policy, application message policy, membership awareness policy, and failure policy. Finally, Antigone provides high-level mechanisms to implement security policy instead of dictating to the developers, and they may configure security policy from a set of standard policies that have been implemented using the Antigone mechanisms.

7.2.4 Dynamic Cryptographic Context Management (DCCM)

The Dynamic Cryptographic Context Management (DCCM) [27] project has been developed to provide security for a very large and dynamically changing group. In the DCCM system, policy plays a key role, and group security policies are represented, negotiated, managed and configured. It also deploys One-way Function Tree (OFT) to provide scalable key management that can handle very large dynamic group (i.e., sizes up to 100,000 members). In DCCM, secure group policies are represented using Cryptographic Context Negotiation Template (CCNT) and are negotiated via Cryptographic Context Negotiation Protocol (CCNP).

Table 5: Summary of Different Policy Frameworks

Method	Data control policy	Access control policy	Specification language	Policy protocol	Follow IETF Policy FW	Fits with MSEC FW
GSAKMP [41]	Yes	Partly	Token based	Specified	No	Yes
XML [87]	Yes	No	XML	Not specified	No	Yes
Antigone [82]	Yes	Yes	Not specified	Specified	No	No
DCCM [27]	Yes	No	Using CCNT	Via CCNP	No	No
MGMS [84]	Yes	No	GML	Not specified	No	No

7.2.5 Multicast Group Management System (MGMS)

The Multicast Group Management System (MGMS) [84] with the aid of Group Management Language (GML) enforces *group integrity* by specifying group policy. The MGMS deals with membership set and member attribute, and does not consider integrity of multicast data. It can be used on top of IP multicast. By sending data in encrypted format, the authors assume that it will be possible to prevent unauthorized users from using IGMP to join the group. However, encryption of traffic can only establish the secrecy of group data. Denial of Service (DoS) and other attacks are still possible by replaying IGMP messages.

7.2.6 Summary of Different Methods

A brief summary of different methods is shown in Table 5. The key entities of the IETF Policy Framework [115] are PEP, PDP and policy repository, and the framework is based on the communication between the PEP and the PDP. It is evident that none of the methods has been constructed conforming to this framework [115], and there is no notion of PEP and PDP in these methods. Only Antigone has fully addressed

access control policy for multicasting. The last three systems (i.e., Antigone, DCCM and MGMS) are basically "secure group communication" frameworks, and are independent of underlying IP protocol (i.e., unicast or multicast). None of them complies with the MSEC Framework for secure multicasting either. It can be concluded that a flexible, scalable, easily adaptable multicast access control policy architecture that follows the IETF Policy Framework is needed.

7.3 Access Control Policy for Multicast Applications

In this section, we outline the functions of access control policies for some of the well-known multicast applications. The authentication policy is very straight-forward, whereas the authorization and accounting policies are hard to specify. Access control policy is composed of all these three types of policies, and is determined according to the needs of a specific application. A comprehensive list of multicast applications can be found in [94], in which, multicast applications are broadly divided into two categories: One-to-Many and Many-to-Many.

7.3.1 One-to-Many (1toM) Applications

1toM applications consist of a single sender and more than one simultaneous receivers. These applications are the most popular and well-suited for IP multicast.

Audio/video streaming: A number of applications (e.g., Internet TV, distance learning) will benefit in audio/video streaming by using IP multicast. In general, authentication of an EU takes place during the joining process and authentication of a sender takes place before she starts sending data.

For Internet TV, an EU may be authorized to view all the channels at any time of the day, or an EU may subscribe to a specific program that will be multicast on a specific day and time. Accounting policy will be specified according to the subscriptions.

In distance learning, class lectures can be multicast during a previously announced time. Only the registered students will be able to receive such lectures. For such applications, authorization policy will be simple, and accounting policy is not even required.

Push media: News headlines, weather updates, sports scores are non-essential applications and require low bandwidth. For these applications, strict authentication of the EU is not necessary, and usually, there will be no authorization and accounting policies. However, a sender must be authenticated and authorized before allowing her to multicast any data.

7.3.2 Many-to-Many (MtoM) Applications

In MtoM applications, more than one GMs will simultaneously send multicast data, while all of them may receive data. Multimedia conferencing, chat groups, multi-player games are some of the examples of MtoM applications.

Multimedia conferencing: A conference may be comprised of audio, video and white-board, and different participants have different roles to play. For a corporate meeting, each of the GMs must authenticate her identity. A role-based authorization policy should be specified to determine the speaker on a time slot. There may be no significance to accounting policy for a corporate meeting.

Multi-player game: A multi-player game is a distributed and interactive application, which may have chat group capability. If subscription is free of cost, authoriza-

tion and accounting policies are not required. Otherwise, an end-user has to prove his/her credit, and relevant authorization and accounting policy should be specified.

7.4 Proposed Policy Framework

In this section, we have presented a set of requirements for the proposed policy framework, its different components, and the policy specification language and the policy protocol it will use.

7.4.1 Design Requirements

We have identified the following requirements that our framework should satisfy:

1. The proposed policy framework will extend the architecture shown in Figure 25. Therefore, it should have provision for the entities—GO, Policy Server, AAAS and AR—and it will allow the transportation of policy information from the Policy Server to the AAAS.

2. The entities defined in the MSEC Framework, i.e., the GO, the Group Controller/Key Server (GCKS) and GMs should be present in the policy framework.

3. The IETF Policy Framework will be the basis of our extended architecture. Hence, the three key entites—PDP, PEP and Policy repository—should be present in our design, and will perform the same tasks as they do in the IETF Policy Framework.

4. Dividing multicast policies into data and access control policies will help us to develop policies independently. Thus, our architecture will be able to deploy any group key management protocol.

5. No single policy specification language and protocol is suitable for every policy framework. A policy framework should not depend on any specification language or policy protocol.

7.4.2 Policy Framework

The proposed policy framework is shown in Figure 27, where the AR will act as the PEP and the GCKS will be the PDP. A host will send an EU's authentication information using the IGMP-AC [60] protocol to the AR. Similarly, a sender will send her authentication information to the nearest AR using PANA. The AR, acting as a NAS (AAA client), will collect the authentication information, and will send this information to the AAA server (AAAS) using one of the AAA protocols. For a failed authentication, a participant (i.e., either receiver and sender) will not be allowed to receive (or send) any multicast data. On successful authentication only, the AAAS will send the AR the authorization and accounting information of the participant. The authorization information of an authenticated EU will depend on the application or content for which the EU had registered earlier. For example, an EU is subscribed for a football match that will be held on 1st January 2008 from 17:00 to 19:00. That EU will be allowed to receive multicast data for that period of time, on that specific date only. Similar authorization information will be needed for an authenticated sender also. The accounting information will be the direction for the AR, how it will keep track of the participant (e.g., the duration of time or the amount of data). Thus, multicast access control policy will be communicated from the AAAS to the AR, where the policy will be enforced.

The role of the GC/KC has been explained in the MSEC Framework [40] and in [41]. It is responsible for the issuance and management of cryptographic keys by building, maintaining and distributing the keys. In a distributed architecture, more than one GC/KSs may exist. In Figure 27, a PDP will be co-located with a

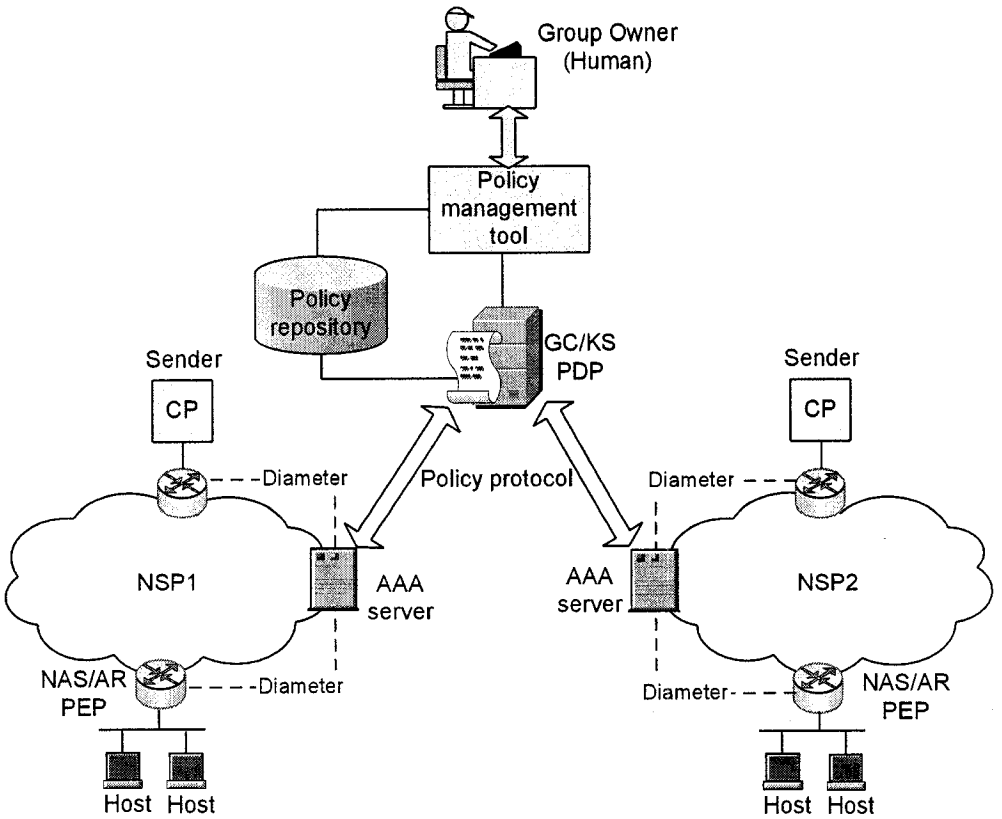

Figure 27: Proposed Policy Framework

GCKS. The AAAS will communicate with the PDP using one of the policy protocols (e.g., COPS [28]) that carries policy information. In the "pull" mode, the AAAS will request the PDP to assert a decision on a participant's access control. However, in the "push" mode, the PDP may send unsolicited policy information (e.g., to ostracize a specific participant) to the AAAS. The push mode might be useful in the case, where the GCKS is notified by the GO that many customers or EUs have originated from a certain Internet Service Provider (ISP). The GCKS will push the policies for a certain group activity to that ISP. In both cases, the AAAS will convey the received policy information to the AR or NAS. In our design, the policy repository will store multicast security policies. It serves the same purposes as the policy server that is defined in the MSEC Framework. Our policy framework will assign some extra tasks to the AR, the AAAS and the GCKS. Moreover, a number of messages should be communicated among these three entities. However, the whole load will be distributed

in these entities, and the presence of more than one GC/KSs and AAASs will further minimize the overhead.

7.4.3 Policy Specification and Protocol

Our architecture is not dependent on any specific policy specification language or policy information transportation protocol. However, we are recommending the use of XACML [35] for policy specification, and Security Assertion Markup Language (SAML) [22] for policy information communication between a PEP and a PDP. We are not precluding the use of any other suitable policy specification language or policy transportation protocol. XACML is an XML-based language for access control, which describes a policy specification language. It is also a query-response language to expresses a query (produced by a PEP) if a particular access should be allowed and the response (provided by a PDP) to that query. An OASIS standard [4] specifies how XACML queries and responses will be carried in SAML.

7.5 Policy Specification in XACML

XACML, along with SAML, is being used widely for access control in different types of networks. It is flexible enough to express most of the needs of access control policy. It is also extensible to support new requirements. Hence, the developers can reuse their existing code to support policy language. For example, Sun Microsystems, Inc. has already implemented XACML using Java [106].

In this section, first, we discuss briefly the construction of XACML policy and the XACML policy framework. Then, an example policy for access control of an on-line course is presented.

```
<Policy >
  <Rule>
    <Target>
       <Subjects>
          <Subject>
             <!--an actor with specific attributes-->
          </Subject>
       </Subjects>
       <Resources>
          <Resource>
             <!--it may be data, service or system component-->
          </Resource>
       </Resources>
       <Actions>
          <Action>
             <!—an operation on a resource-->
          </Action>
        </Actions>
    </Target>
    <Condition>
        <--an expression that can be evaluated true or false-->
    </Condition>
  </Rule>
</Policy>
```

Figure 28: Component of XACML Policy

7.5.1 XACML Policy Framework

The skeleton of an XACML policy is shown in Figure 28. A `rule`, the most elementary unit of a `policy`, consists of a `target`, an `effect` and a `condition`. A `target` is a composition of `resources`, `subjects`, `actions` and `environments` to which a `rule` is applied. `Effect` and `environment` are not shown in Figure 28. `Subjects` may have more than one `subject`, which are actors with specific attributes. Similarly, `resources` and `actions` may have multiple `resources` and `actions`, respectively. A `resource` may be data, a service or a system component. A `condition` further refines the applicability established by a `target`.

A simplified overview of the XACML framework [80] is shown in Figure 29. In a typical XACML usage scenario, if a `subject` (e.g., multicast participant, work station) wants to take an `action` on a particular `resource`, it submits its query to the PEP (e.g., a host sends an IGMP join message to the AR). The PEP forms an

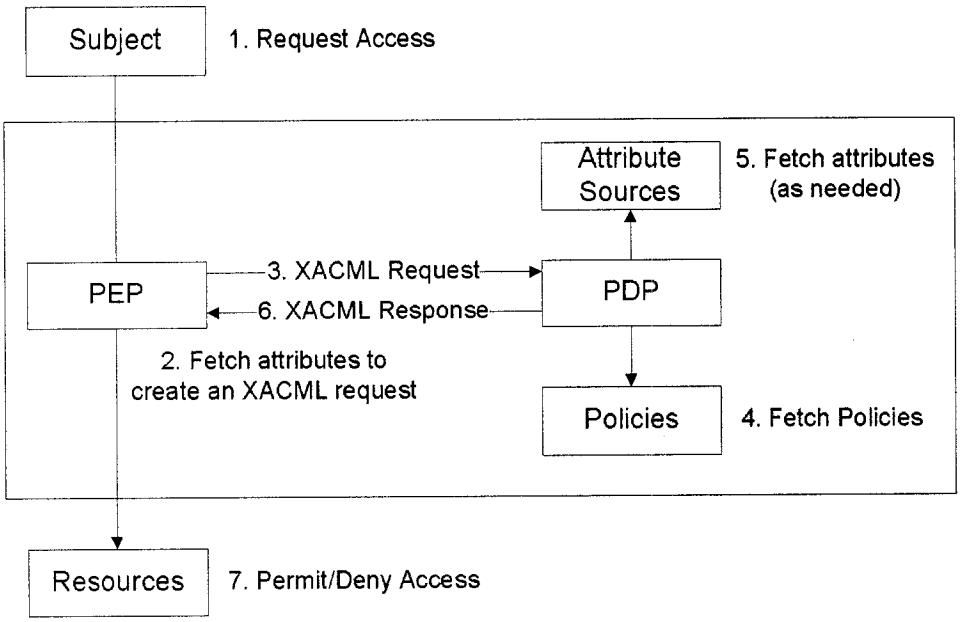

Figure 29: XACML Policy Framework

XACML request based on the attributes of the `subject`, `action` and `resource`, and sends this request to a PDP. The PDP examines the request by retrieving policies (written in XACML) that are applicable to the received request. It determines the access control decision by evaluating the XACML rules and sends an XACML answer (any one of `Permit`, `Deny`, `NotApplicable` or `Indeterminate`) to the PEP, which can then allow or deny access by the requester. Although in first impression, an XACML encoded policy seems to have many redundant lines, the PDP applies an efficient parsing for evaluating XACML rules. Moreover, there exists a number of tools for developing XACML policies (e.g., UMU-XACML-Editor [108]).

7.5.2 Policy for an On-line Course

Here, we have shown how XACML could be used to develop access control policy for a real-life application. This is an access control policy for the on-line course, INTE290 (Computer Usage and Document Design) offered by Concordia University. We have made the following assumptions:

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:cd"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:cd
        http://docs.oasis-open.org/xacml/access_control-xacml-2.0- policy-schema-cd-01.xsd"
        PolicyId="StudentAccessControlPolicy"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description> An access control policy for the on-line course, INTE290 offered by Concordia University
   during September to December, 2006. The rule defines that a student, successfully authenticated with his/
   her email address and password, is authorized to  receive class lectures.</Description>
  <Target />
  <Rule Effect="Permit" RuleId="StudentAccessControlRule">
    <Description> If a request is successfully matched against this rule an evaluating PDP will return Permit
     </Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                    user-id@econcordia.ca
            </AttributeValue>
            <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                    DataType="http://www.w3.org/2001/XMLSchema#string"/>
           </SubjectMatch>
           <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                    Password
            </AttributeValue>
          <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:authentication-method"
                    DataType="http://www.w3.org/2001/XMLSchema#string" />
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                    http://www.econcordia.ca/INTE290/lectures
            </AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                    DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
            </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                    Read
            </AttributeValue>
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                    DataType="http://www.w3.org/2001/XMLSchema#string" />
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-greater-than">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
            <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
                    DataType="http://www.w3.org/2001/XMLSchema#date" />
          </Apply>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">
                    2006-09-01
          </AttributeValue>
        </Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than">
          <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
            <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
                    DataType="http://www.w3.org/2001/XMLSchema#date" />
          </Apply>
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">
                    2006-12-31
          </AttributeValue>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

Figure 30: XACML Access Control Policy for an On-line Course

- The course will be offered from September to December, 2006.
- Each student will be provided with an email address (in the format of `user-id@econcordia.ca`) and password that he/she will use for authentication.
- The multimedia presentation of the lecture will be multicast during the class hours.
- The students will be able to receive a lecture through `http://www.econcordia.ca/INTE290/lectures` and are allowed to send any question in textual or voice format during the class hours.

We need one 1toM multicast group for sending the multimedia data of the class lecture and another MtoM group for voice or text chat. The 1toM group should be allocated higher bandwidth and higher priority compared to the MtoM group. The policy shown in Figure 30 is only for the 1toM group that the students will join during class hours. This specification in XACML would be stored in the Policy Repository that the PDP would access to reply any query.

The policy specified in Figure 30 has single `Subject` with two attributes—`user-id@econcordia.ca` and `Password`, which must be matched to authenticate a student. Moreover, it is stated that the `Resource` will be available at the website, `http://www.econcordia.ca/INTE290/lectures`. The only `Action` an authenticated student is permitted to is `Read`, which means to receive data. The policy is further refined with a date constraint, which is specified using `Condition` that the policy is valid from `2006-09-01` to `2006-12-31`.

Chapter 8

Inter-Domain Access Control

An inter-domain multicast group is distributed over more than one domain, which makes it quite difficult to design a scalable access control solution. In this chapter, we have presented an inter-domain access control architecture that we have developed by extending the intra-domain architecture (see Figure 12). We have developed the receiver and sender access control mechanisms independently. Inter-domain access control depends on facilities provided by the Diameter protocol, such as discovery of network entities that are located in remote domains and secure inter-domain transportation of AAA information. In this chapter, after a brief discussion on Diameter agents that participate in inter-domain communication, the receiver and sender access control architectures are presented. To maximize the distribution of the multicast group members we have assumed that the sender, the EUs or receivers and the GO are located in different domains.

An inter-domain Data Distribution Tree (DDT) is distributed over different domains. Hence, protecting only the communication between the sender and the one-hop AR without protecting the whole DDT, the sender access control will be meaningless. Therefore, we have focused in protecting the DDT from different attacks generated by a compromised or forged network entity (e.g., router and host). We have devel-

oped two alternate solutions that detect and stop forwarding of any forged packet. The first method deploys a centralized Multicast Security Association (MSA) for the whole DDT while the second method deploys a number of small sized MSAs. Finally, we have compared the two methods with respect to different features, such as establishment and maintenance costs, delivery time, security, etc.

8.1 Diameter Agents

Diameter is a peer-to-peer protocol (any node can initiate a Diameter `Request`) with three types of nodes: NAS, agents and AAAS. Diameter agents—Proxy, Relay and Redirect—are Diameter nodes that do not authenticate and/or authorize messages by themselves. Diameter messages (i.e, `Request` and `Answer`) are always originated by the NAS or the AAAS. An agent sits between the NAS and the AAAS, and forwards a Diameter message to the appropriate node. In the Diameter specification [18], the term "realm" is used to address administrative domain, and hence, the words *realm* and *domain* will be used interchangeably in this chapter. Given that our proposed inter-domain access control architecture deploys Redirect and Relay agents, the roles of these two agents have been explained briefly in the following.

A Relay agent maintains a Realm Routing Table and on receiving a `Request` message, routes the message to another Diameter node based on the information found in the `Request` message and the Realm Routing Table. A Redirect agent does not forward any message. However, it notifies the requesting node which route is to be used next. Figure 31 illustrates how a NAS communicates with a Home Diameter Server (HMS) (a remote AAAS, which is deployed outside the local realm) with the aid of the Redirect and the Relay agents. In this example, the NAS is an access control device for the user with identity, `bob@example.com` in the format of NAI [1]. As we have explained before, `bob` is the user name and `example.com` is the domain name of this user identity. It is worth while to note that, a user identity must be

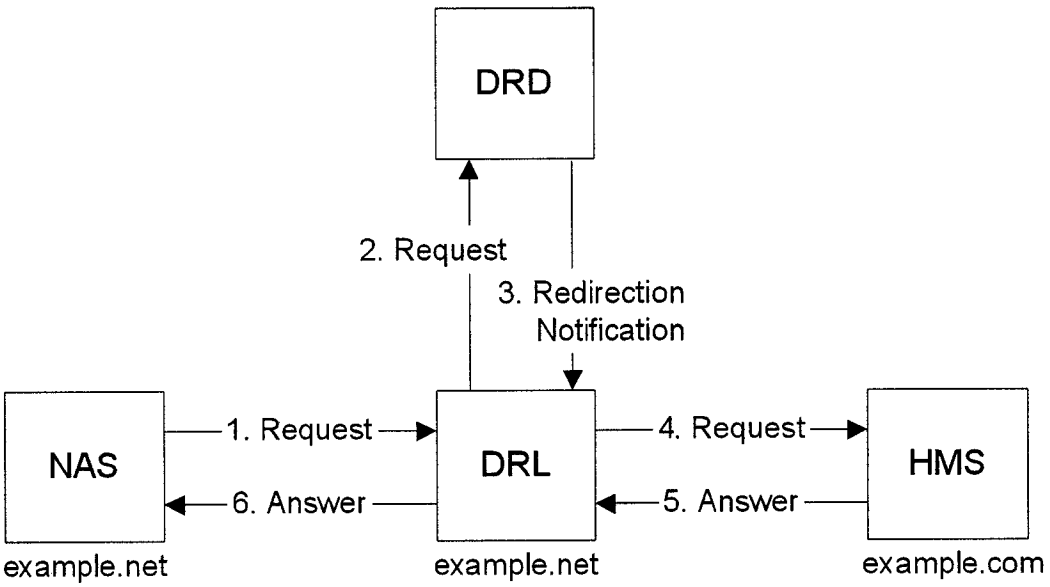

Figure 31: Diameter Agents: Redirect and Relay

globally unique and could be verified only by the HMS that maintains the user's database.

The NAS, triggered by the user's access request that contains the user identity, starts the communication. First, the NAS performs a Diameter route lookup, using `example.com` as the key, and determines that the `Request` is to be relayed to the Diameter Relay agent (DRL). The DRL performs the same route lookup as the NAS has done, and fails to find any routing entry for `example.com`. Therefore, the DRL forwards the `Request` to the default route, which is configured towards the Diameter Redirect agent (DRD). The DRD returns the `Redirection Notification` to the DRL, which contains the route to reach the HMS. Next, the DRL relays the `Request` to the HMS. In the last step, the HMS returns the `Answer` to the DRL, which forwards it to the NAS. Finally, depending on the `Answer` received from the HMS, the NAS allows/denies the user's access to the network.

8.2 Receiver Access Control

In Figure 32, we have shown the inter-domain receiver access control architecture spanning three domains (i.e., NW1, NW2 and NW3). This architecture has been designed by extending the access control architecture shown in Figure 12. Given that this section is devoted in receiver access control only, the other features of the architecture, such as sender access control and e-commerce communications, are not shown here. Furthermore, to avoid complexity the Diameter nodes (i.e., AAAS, Relay agent and Redirect agent) are not shown for each domain. However, each domain has all three Diameter nodes, and the inter-domain links between two Diameter nodes are secured following the Diameter specification. The sender, `S`, that sends data to the group, `G`, is connected with NW1, while the EUs are connected with NW2. These two domains are linked by two Border Routers (BR): BR1 and BR2. The Multiprotocol Extensions for Border Gateway Protocol v4 (M-BGP) [14] will be used by the BRs to build the forwarding table. The DDT, which is being built by the CRs and the BRs, will be distributed over the two domains. The shape of the tree will depend on the underlying routing protocol. For example, in an SSM group [47], the DDT will be a shortest-path tree rooted at the source and the BRs will be the internal nodes of the DDT. We are also assuming that the GO is connected through NW3. The participants maintain an account relationship with the AAAS of NW3, which is referred as Home AAAS (HAAAS). Hence, HAAAS has direct access to the participants' database and is able to verify the participants' identities.

8.2.1 IGMP-AC Behavior

At first, an EU will send AR2 an IGMP-AC report message, which will carry the EU's identity (in the format of `bob@example.com`) and the secured group address to which the EU is willing to join. On receiving the IGMP-AC report, AR2 will compose a Diameter `Request` message. Next, AR2 will have to decide to which Diameter node

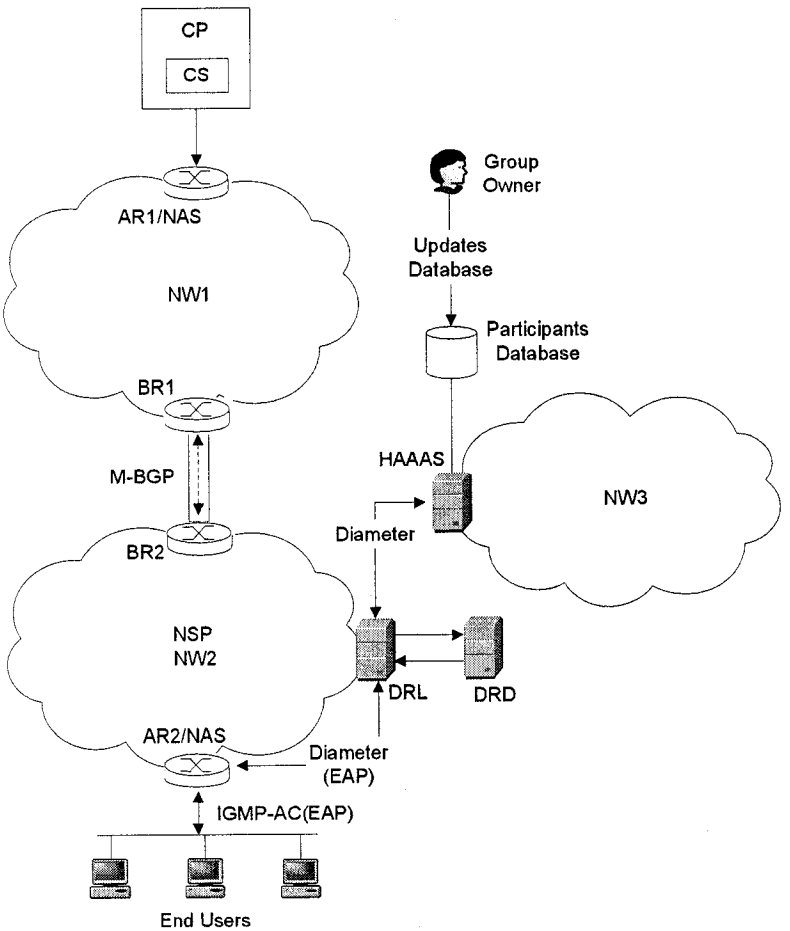

Figure 32: Inter-domain Receiver Access Control Architecture

the `Request` would be sent. Given that the EUs and the HAAAS are located in different domains in the example scenario shown in Figure 32, Diameter agents of NW2 would have to be involved in inter-domain communication. By extracting the domain name from the EU's identity (e.g., `example.com`) AR2 will understand that it would not be possible for the local AAAS (i.e., AAAS of NW2) to `Answer` the `Request`. Hence, AR2 will forward the `Request` to the DRL of NW2. The DRL will follow the procedure explained in the previous section, and will communicate with the DRD (if required) to get the route of the HAAAS. Finally, the DRL will relay the `Request` to the HAAAS.

8.2.2 Distributed vs. Centralized Database

There are two ways of maintaining the database of participants for inter-domain access control. In the centralized database method, the participants' database would be maintained only by the HAAAS. A local Diameter node will act as a pass-through device and will never maintain any AAA information. Therefore, in our receiver access control architecture, if an EAP method requires n round-trips to be completed, n pairs of Diameter messages (`Request` and `Answer`) will be exchanged between AR2 and HAAAS. These messages will always be relayed through the DRL.

In the distributed database method, the participants' database might be distributed over a number of local AAASes. When the HAAAS will receive the first Diameter `Request` originated by AR2, the HAAAS will search the participants' database using the EU's username (e.g., using `bob` in case the identity is `bob@example.com`). If the AAA information of the EU is found, it will be forwarded to the local AAAS (i.e., AAAS of NW2). From then on, AR2 will communicate with the local AAAS directly to authenticate and authorize the EU.

Once an EU has left a secured group or his/her session time has been expired, AR2 will gather accounting information. If the EU's AAA information is maintained by the local AAAS, AR2 will send the accounting information to it. Otherwise, this information should be forwarded to the HAAAS.

Each of these two methods of maintaining participants' database has its own advantages and disadvantages. Network administrators will decide the specific implementation depending on the requirements and available resources.

8.3 Sender Access Control

In Figure 33, we have shown an inter-domain sender access control architecture spanning four domains: NW1, NW2, NW3 and NW4. The sender, S, that sends data to the group, G, is connected with NW1, while the EUs are connected with NW2 and NW3. These three domains are linked by three Border Routers (BR): BR1, BR2 and BR3. The Multiprotocol Extensions for Border Gateway Protocol v4 (M-BGP) [14] will be used by the BRs to construct the DDT. The CRs and the BRs will build the DDT, which will be distributed over the three domains. We are also assuming that the GO is connected through NW4. The participants maintain an account relationship with the AAAS of NW4, which is referred as Home AAAS (HAAAS). The AAAS of the network to which the sender is located in known as Local AAAS (LAAAS). Hence, the AAAS of NW1 (not shown in this figure) is named as LAAAS in Figure 33.

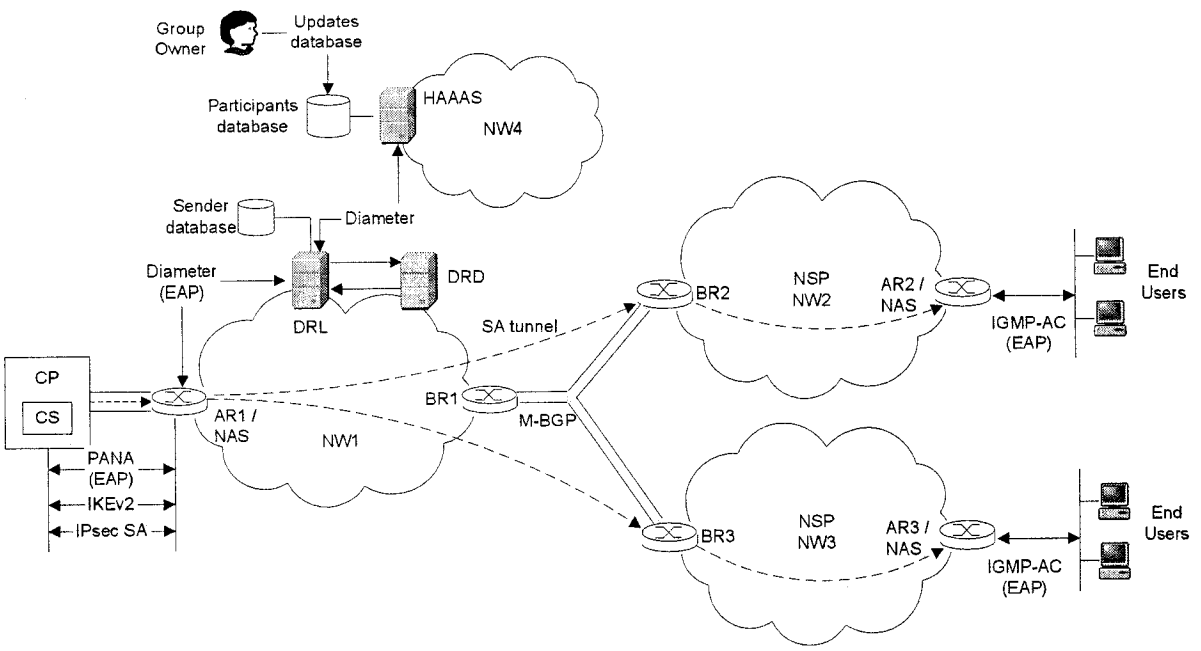

Figure 33: Inter-domain Sender Access Control Architecture

The inter-domain sender access control architecture, shown in Figure 33, will be explained in two steps. In the first step, the sender will be authenticated and

authorized, and an IPsec SA will be established between the sender and AR1 to control access of data to NW1. Next, a number of SAs, which enforce checkpoints every time a packet reaches a new domain, will be established to protect the distribution tree.

8.3.1 Sender Authentication and SA Establishment

The inter-domain sender access control architecture has been developed by extending the intra-domain architecture shown in Figure 22. Therefore, the procedure explained in section 6.2.3 to initiate sender authentication will be followed, and the sender will send AR1 her identity (in the format of an NAI, e.g., `bob@example.com`) and the destination address of the secured group to which she intends to send data. By parsing the domain name (i.e., `example.com`) of the sender's identity, AR1 will understand that the Diameter `Request` it generates must be sent to the HAAAS located in the `example.com` domain. Thus, AR1 will forward the `Request` to the DRL of NW1. The communication process explained in the previous section will be followed to reach the HAAAS.

It should be noted that the distributed database method explained in the previous section will be a good design choice for sender access control. A multicast group is expected to have only a few senders and most of the groups have only one sender. Therefore, in distributed database method we do have to maintain only few copies of the sender database. In distributed method, on receiving the first `Request` message, the HAAAS will forward AAA information of the sender to the LAAAS. The following steps will be same as explained in section 6.2.4. Hence, the `AAA-Key` will be established between the sender and the LAAAS, and the `IKE Pre-shared-Key` will be established between the sender and AR1. Consequently, the sender and AR1 will authenticate each other and will establish an IPsec SA to provide per-packet cryptographic protection to each packet before forwarding to the DDT.

8.4 Data Distribution Control

In our intra-domain sender access control architecture, we have assumed that a router is a trusted entity (see section 6.2.1). However, this is a restrictive assumption. Specially, for an inter-domain group, we have to trust the upstream routers of an adjacent domain, which are maintained by other administrators. In this section, we will overcome that limiting assumption. It should be noted that the data distribution control mechanism described in this section is applicable only for SSM [47] groups.

The key concept for controlling data distribution lies in protecting the DDT by establishing a number of IPsec SAs. However, the number of SAs to be established and the underlying communications to establish these SAs may vary. In section 3.2.2, we have listed a number of attacks that a multicast group may face without any sender access control. We have noticed that a multicast group suffers from replay and sender address spoofing attacks, which may cause DoS attack also. Among these attacks, we can prevent replay and sender address spoofing by deploying an IPsec SA. On the other hand, a DoS attack could not be resisted directly, however, it would definitely be minimized by preventing the other two attacks.

The goal of our inter-domain sender access control architecture is to implement a checkpoint at the entry point of each domain. Therefore, in the example scenario shown in Figure 33 two SAs (in tunnel mode) will be established between the ingress routers of NW1 (i.e., AR1) and NW2 (i.e., BR2), and between AR1 and the ingress router of NW3 (i.e., BR3). An "ingress router" is located at the edge of a network and might be designated for access control and traffic filtering.

In addition to establishing an SA at the entry point of a domain, another SA will be established at the last domain, where an EU is located. Hence, in Figure 33, two SAs will be established in NW2 and NW3. These SAs will be established between the ingress router and the AR, i.e., between BR2 and AR2, and between BR3 and AR3.

By establishing these SAs our architecture would be able to prevent attacks from any entity (e.g., routers and end hosts) even from inside the domain of the EUs. For example, if a CR of NW1 replays some previously sent packets, the replayed packets will be detected and dropped at BR2 and BR3. Therefore, the attack will be detected very quickly and never be propagated or flooded throughout the DDT.

Following this pattern, a series of SAs will be established until the destination domain is reached. Hence, the number of SAs to be established (from S to a specific EU) will be equal to the number of domains on the path from S to the EU. Thus, the longer this distance is (in terms of number of domains) the longer the packet delivery time would be. Furthermore, in case an intermediate domain has no receiver for a multicast group, an SA will be established only if the domain had a prior Service Level Agreement with its adjacent domain to forward data of that group.

8.4.1 Multicast Security Association (MSA)

Our data distribution control mechanism is heavily dependent on IPsec SA. An SA is a simplex "connection" that affords security services to the traffic carried by it. IPsec SA has been primarily designed to protect unicast traffic, however, it could be used for multicast communication with limited security services. In RFC 4301 [69], an SA with multicast address as a destination address has been classified as Multicast SA (MSA). An extension of RFC 4310 has been proposed in [112] to broaden the functionalities of an MSA. In this thesis, we have adopted the extended definition of MSA to make use of the facilities it provides.

In Figure 34, a generic composition of an MSA for SSM groups has been shown. As we are dealing with SSM groups only, only one sender, S, has been shown here. However, for ASM groups, more than one sender may use the same MSA for secure data distribution. An MSA has one or more receivers (R1, R2, ... , Rn) that receive

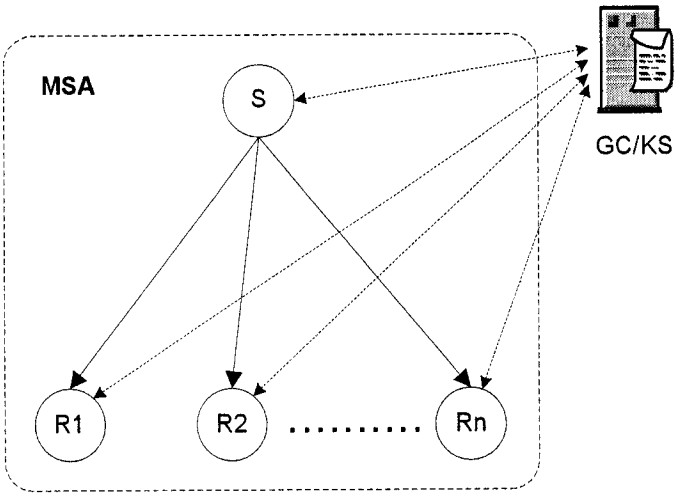

Figure 34: Multicast Security Association (MSA)

multicast data from S through the DDT. The presence of a Group Controller/Key Server (GCKS) is required to construct the MSA, and all the members of the MSA (i.e., the sender and the receivers) should communicate with the GCKS to get the MSA parameters (e.g., encryption and authentication keys, Security Parameter Index (SPI), etc.). The members of the MSA will use one of the Group Key Management (GKM) protocols (e.g., Group Domain of Interpretation (GDOI) [113], SIM-KM [88], etc.) to communicate with the GCKS. It should be noted that the deployment of a GKM protocol is not mandatory to implement. However, without an automated GKM protocol, an MSA would not be able to provide some important functionalities, such as anti-replay.

The definition of the GCKS has been presented in the MSEC Reference Framework [40] that we have briefly discussed in section 2.1.1. In a distributed design, where a multicast group is distributed over multiple domains, a GCKS entity has to be present in each domain. Moreover, a GCKS entity will interact with other GCKS entities to achieve scalability in the key management related services. However, to simplify our design, we are assuming the presence of a centralized and global GCKS entity that will serve all the group members of an MSA. The deployment of scalable and distributed GCKS entities is a ongoing research topic, and outside the scope of

this thesis. We have designed two different data distribution control mechanisms for SSM groups, which are described in the following after explaining the SSM service model.

8.4.2 Source-Specific Multicast (SSM)

IP Multicast group addresses in the 232/8 (232.0.0.0 to 232.255.255.255) range are designated as Source-Specific Multicast (SSM). A group address, which is not in this range, should never be considered as an SSM group. On the other hand, Any-Source Multicast (ASM) is the IP multicast service model defined in RFC 1112 [25]. An ASM group address is identified by a single IP destination address from the range 224/4 (224.0.0.0 through 239.255.255.255 for IPv4) excluding 232/8. This model supports multicast groups with arbitrarily many senders.

SSM uses a subset of PIM [32], which is also known as PIM-SSM to build the DDT. However, it is much simpler than PIM, and there is no use of Rendezvous Point (RP) and shared-tree. The BRs use Multiprotocol Border Gateway Protocol v4 (M-BGP) [14] to advertise the reachability beyond one domain. In ASM inter-domain group (that uses PIM), a receiver will join its local RP first. The local RP will learn the location of the sender's domain RP using Multicast Source Discovery Protocol (MSDP) [31]. Then, the local RP will join the sender's domain RP. This whole complex mechanism is absent in SSM.

The service model of SSM is shown in Figure 35, where the sender and the receivers are located in different domains or networks: NW1 and NW2. Here, we have assumed that both of these networks are PIM domains, and hence, they have deployed PIM as their multicast routing protocol. A receiver (e.g., Host2) will know the source address of an SSM group using some out of band mechanisms (e.g., Session Announcement Protocol (SAP) [37]), and will send an IGMP (S, G) Join message to the one-hop AR

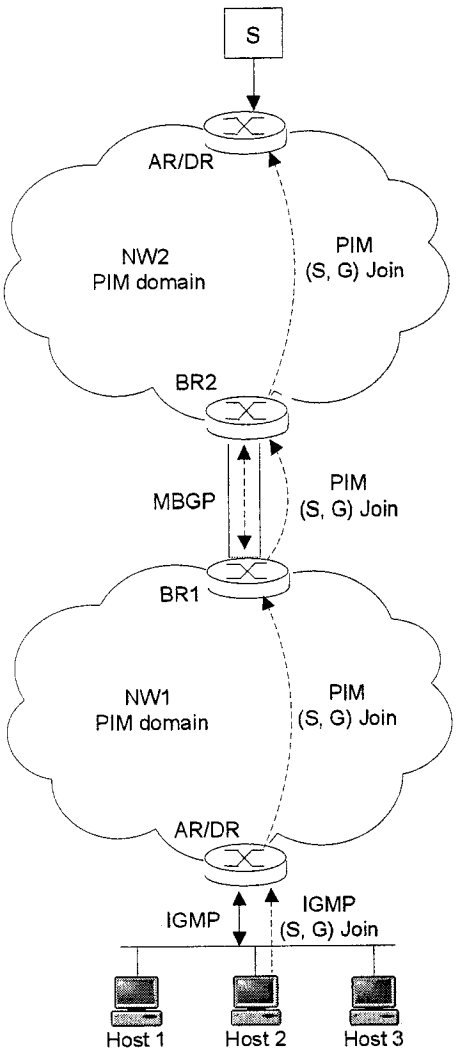

Figure 35: Source-Specific Multicast Service Model

or Designated Router (DR). The DR will originate a corresponding PIM (S, G) Join message, which will eventually reach the border router of NW1 (i.e., BR1). BR1 will forward this PIM Join message to BR2, the border router of NW2. Finally, this Join message will reach the DR or one-hop router of the sender, S. Hence, the DDT will be built for inter-domain SSM groups.

It should be noted that a PIM (S, G) Join message could be used for a group address G that is not in the range of SSM groups. However, it will be considered as an ASM group although it is source-specific. Accordingly, a PIM (*, G) Join message

for a group address, G that is in the range of SSM groups, must be ignored by the routers.

8.4.3 Extending PIM (S, G) Join message

According to the present PIM specification [32], PIM join messages do not carry the identification of the originator of the message. A link-local address of the interface to which the message is being sent is used as the source address and a special multicast address, ALL_PIM_ROUTERS (224.0.0.13 in IPv4 and ff02::d in IPv6) is used as the destination address. In our extension, a router will have the option to append its identity when it originates or forwards a PIM (S, G) Join message. It should be noted that, this message must target only an SSM and Secured group. The router identity carried in the PIM (S, G) Join message will be used by the GCKS to initiate an authentication session. Hence, the IP address of the router could be used as an identity of the router. A PIM (S, G) Join message is originated by an AR, and propagated by the CRs and the BRs towards the source as explained in Figure 35. However, not all the routers that originate/forward this join message will append their identities, and it will depend on the structure of the MSA (i.e., centralized or distributed). This issue is further explained in the following sections.

8.4.3.1 TLV's in PIM Join Messages

The PIM Working Group [93] of the IETF is on the process of specifying a generic TLV (Type-Length-Value) attribute encoding format, which could be added to the PIM join messages [16]. This new specification defines a new field in the PIM join message that allows the addition of TLVs or attribute fields. Thus, a PIM join message would be able to contain multiple attributes. The attributes are encoded as TLVs associated with a new PIM source type in the PIM message. Given that there

is no space in the default PIM source encoding to include an attribute field, a new source encoding type has been introduced and the attributes are formatted as TLV's. The new Encoded source address is shown in Figure 36.

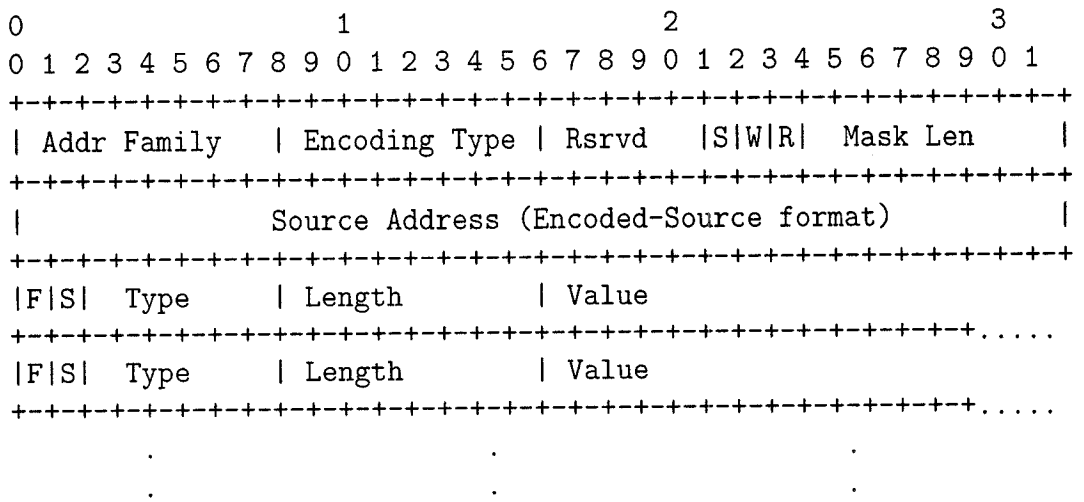

Figure 36: Format of the New Encoded Source Address of PIM Join Message

The first portion (up to the Source Address) of this Encoded Source Address is specified by the RFC 4601 [32], and the attribute part is newly added by [16]. In the following, we have explained each field:

Addr Family The PIM address family of the 'Unicast Address' field of this address. Values 0-127 are assigned by the IANA [53] for Internet Address Families in [54]. Values 128-250 are reserved to be assigned by the IANA for PIM-specific Address Families. Values 251 though 255 are designated for private use. As there is no assignment authority for this space, collisions should be expected.

Encoding Type The type of encoding used within a specific Address Family. The value '0' is reserved for this field and represents the native encoding of the Address Family.

Rsrvd This field is reserved, transmitted as zero and ignored on receipt.

S The Sparse bit is a 1-bit value, set to '1' for PIM-SM. It is used for PIM version 1 compatibility.

W The WC (or WildCard) bit is a 1-bit value for use with PIM Join/Prune messages.

R The RPT (or Rendezvous Point Tree) bit is a 1-bit value for use with PIM Join/Prune messages. If the WC bit is '1', the RPT bit must be '1'.

Mask Len The mask length field is 8 bits. The value is the number of contiguous one bits left justified used as a mask which, combined with the Source Address, describes a source subnet. The mask length must be equal to the mask length in bits for the given Address Family and Encoding Type (32 for IPv4 native and 128 for IPv6 native). A router should ignore any message received with any other mask length.

Source Address The source address.

F Forward Unknown TLV. If this bit is set the TLV is forwarded whether or not the router understands the Type.

S Bottom of Stack. If this bit is set then this is the last TLV in the stack.

Type A 6 bit field of the TLV.

Length A 1 byte field of the TLV.

Value Content of the attribute.

8.4.3.2 Use of PIM Join Attributes in Our Architecture

The Join Attributes described in the previous section would perfectly serve the extension we have recommended to the PIM (S, G) Join message. We have proposed to extend the PIM join message to carry router identification (e.g., IP address of a router). A router identification could be expressed in the TLV format and could be

sent as a Join attribute. Furthermore, we have proposed to modify the forwarding rules of the PIM (S, G) Join message in the following sections. Using the 'F' bit of Figure 36, the necessary modifications in the forwarding rules could be easily implemented. In the following we have explained the use of the 'F' bit and the TLV fields of Figure 36 in our proposed extension of the PIM (S, G) Join message:

Type An IANA assigned value should be used as attribute type.

Length Depends on Address Family of Encoded-Unicast address.

Value Encoded-Unicast Address of the router's IP address. The format of the Encoded-Unicast Address is explained in section 4.9.1. of RFC 4601 [32].

F This bit will be set if a router (who is sending a PIM (S, G) Join message with at least one IP address expressed in TLV format) wants its immediate upstream-router to forward the TLV towards the source or root of the DDT.

8.4.3.3 Securing MSA Establishment

The MSA establishment will be initiated by the PIM (S, G) Join message. Therefore, the PIM join messages must be integrity protected. Moreover, the identity of the sender (i.e., PIM routers) should be verified before forwarding a PIM join message to the upstream router. The PIM join message is a link-local message and the PIM Working Group is on the process of specifying security of PIM link-local messages [8].

Once the PIM join message will reach the GCKS, the GCKS will receive the identity of the potential member (i.e., BR or AR) of the MSA and open an authentication session with that potential member for registration. From now on, the security of the MSA operation (registration, distribution of MSA parameters, rekeying, etc.) will be achieved by the GKM deployed for establishing and maintaining the MSA.

8.4.4 Access Control by Centralized MSA

The centralized MSA mechanism is shown in Figure 37, where only one MSA will be shared by the whole SSM group. The only sender of the MSA will be the DR or the one-hop AR that is directly connected to the sender, while all other nodes will be the receivers. As we have shown in Figure 33, two sets of SAs will be established for data distribution control. The first set will be between two ingress routers that are located in two adjacent domains. The second set will be between an ingress router or BR and the AR that has directly connected receivers. For simplicity, the CRs are not shown in Figure 37. Only the ingress BRs and the ARs that have directly connected receivers are shown here. It should be noted that, in this chapter, the term "CR" is used to mean to two types of routers: the routers that reside inside a domain and the egress router. An egress router is a BR that forwards traffic outside a domain. For example, in Figure 33, BR1 is an egress router.

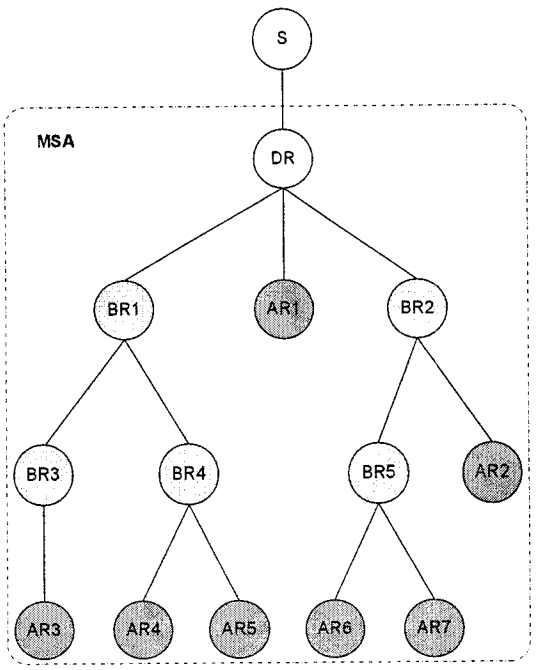

Figure 37: Access Control by Establishing a Single MSA

In the centralized method, the DR that is directly connected with the sender, will also act as the GCKS of the MSA. We are proposing to modify the present

DDT building mechanism and the PIM-SSM forwarding rules of the PIM (S, G) Join message. The proposed forwarding rules are already allowed in the Internet Draft that specifies format of TLVs in PIM messages [16]. According to the present PIM specification [32], an (S, G) Join message will be forwarded to the upstream router until it reaches the DR of the S or a sub-branch of the DDT that has been built to distribute data destined to (S, G) channel. For the successful operation of the centralized MSA method, we are proposing some modifications or extensions of the existing PIM protocol. We are recommending to use the extended PIM Join message, shown in Figure 36 to make use of the facility of adding the TLV attribute and using the 'F' bit. In the following, we have explained the functionalities of four types of routers (i.e., AR, CR, BR and DR):

1. If an AR originates a PIM (S, G) message for the first time, the AR will add its identity TLV and set the 'F' bit '1'.

2. If a CR receives an (S, G) Join with an identity TLV of a down-stream router with F = 1, the CR must forward the message to the DR even if the CR had already joined the same SSM channel and had an (S, G) forwarding state.

3. If a BR receives an (S, G) Join with an identity TLV of a down-stream router with F = 1, the BR must forward the message to the DR even if the BR had already joined the same SSM channel and had an (S, G) forwarding state. If the BR is forwarding the (S, G) message for the first time, the BR will add another TLV with its own identity and will set the 'F' bit of the newly added TLV '1'.

4. If a DR receives a PIM (S, G) Join message with one or more routers' identity TLVs, the DR will start an authentication session with each router that sent its identity to register the router.

We are explaining the construction rules of the centralized MSA with an example.

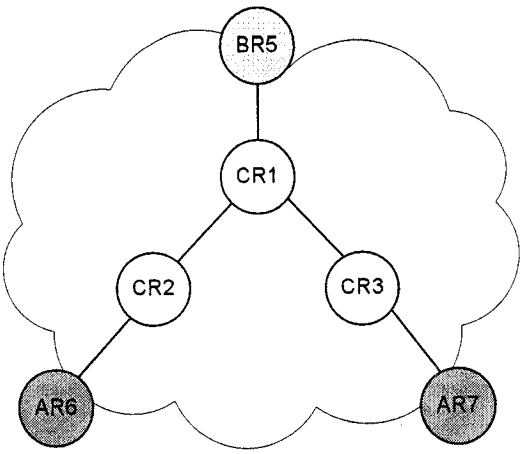

Figure 38: Detailed Architecture Inside a Domain

We are considering the routers, BR5, AR6 and AR7, which are located inside the same domain. The internal connections of these routers are shown in Figure 38, where three CRs (CR1, CR2 and CR3) are member of the DDT.

Assume that AR6 joins the distribution tree before AR7. First, AR6 will add its identity's TLV with F = 1 and send the message to CR2, which will forward the message to CR1 without any change. CR1 will forward the same message to BR5. Now, BR5 will add its own identity's TLV and forward the message towards the DR. We are skipping the detailed connections between BR5 and BR2, and assuming that they are directly connected. Hence, BR2 will receive this message, will append its identity with F = 1, and will forward it to DR. When DR will receive this message, it will find three identity TLVs of BR2, BR5 and AR6. Hence, DR will start three authentication sessions to register these routers and to add these routers to the centralized MSA. Next, AR7 will create an (S, G) message with its identity's TLV and F = 1. This message will reach CR1 through CR3 with F = 1. Although CR1 has an (S, G) forwarding state, CR1 will forward this to BR5. Given that BR5 has previously forwarded this message, BR5 will not add its own identity. It will forward this message to BR2 only. BR2 will follow the same procedure and will forward the message to DR. On receiving this message DR will open an authentication session with AR7.

8.4.5 Access Control by Distributed MSAs

The distributed MSA mechanism is shown in Figure 39. In distributed MSA, many small sized MSAs will be built, where the parent node will be the sender of an MSA, and all the children will be the receivers of the MSA. The sender router will act as the GCKS of the MSA. Therefore, each time a multicast packet enters a new domain, the packet will go through a new SA tunnel.

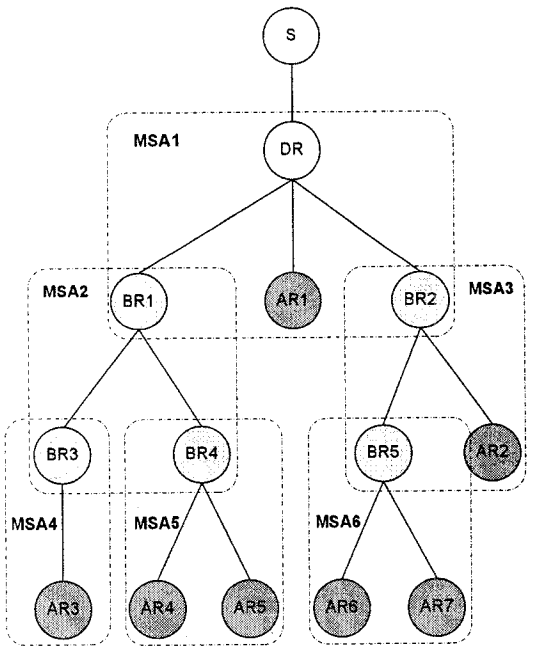

Figure 39: Access Control by Establishing Multiple MSAs

The construction method of the distributed MSA is similar to the centralized MSA, except the PIM (S, G) Join message forwarding rules are different for BRs. For ARs, CRs and DR, the forwarding rules and the building procedure of the (S, G) Join message will be same as the centralized method. The rules are different only for a BR. The distributed method also recommends the use of the newly defined PIM Join attributes [16]. Hence, the extended PIM Join message, shown in Figure 36, will be used.

When a BR receives an (S, G) Join message that carries an identity TLV, the BR

will learn that a potential receiver is willing to join the MSA for which the BR is the GCKS. The BR will remove the identity TLV and will open an authentication session with the downstream router that sent its identity. It should be noted that a BR will never receive more than one identity TLVs in one join message. If the BR is forwarding the (S, G) Join message for the first time, it will add its own identity TLV with F = 1 to the join message. For example, assume that AR6 joins the distribution tree before AR7 in Figure 39. First, AR6 will originate an (S, G) Join message with its identity TLV and F = 1. The message will reach BR5 through the CR(s). Now, BR5 will remove the AR6's identity TLV, open an authentication session with AR6, add its own identity's TLV and forward the message towards the DR. BR2 and DR will follow the procedure explained in the centralized MSA construction. Next, AR7 will create an (S, G) message with its identity's TLV and F = 1. This message will reach BR5 through the CR(s). BR5 will remove AR7's identity TLV, and open an authentication session with AR6. Given that BR5 has previously forwarded the (S, G) message towards DR and the (S, G) Join message is not carrying any identity TLV, BR5 will not forward this message to BR2 again.

8.4.6 Router Authentication

The first step of establishing an MSA will be authenticating each receiver through the GCKS. In our design, the receivers are the routers, and the tasks of the GCKS will be delegated to the sender, which is another router. Moreover, there is a need to authenticate the routers, which may be located in different networks or domains. Hence, a global identification mechanism such as Public Key Infrastructure [50] would be well-suited in our model. PKI is an arrangement that binds a public key with the respective user identity by means of a certificate authority (CA). The user identity must be unique for each CA. The binding is established through the registration and issuance process. The PKI role that assures this binding is called the Registration Authority (RA). For each user, the user identity, the public key, the binding, validity

conditions and other attributes are made unforgeable in public key certificates issued by the CA. PKI arrangements enable a set of entities, such as routers, to authenticate each other without prior contact.

PKI has been deployed in Secure Border Gateway Protocol (S-BGP) [72], which is an extended version of BGP [95]. S-BGP uses PKI based on X.509 (v3) certificates to enable a BGP speaker (i.e., BR) to validate the identities and authorization of other BGP speakers. A further extension of S-BGP is underway at the IETF [75]. The goal of this effort is to develop an architecture, based on PKI infrastructures, to support secure Internet routing.

8.4.7 MSA Operation

IPsec depends on two protocols—Authentication Header (AH) [70] and Encapsulating Security Payload (ESP) [71]—to provide traffic security services. The AH offers integrity, data origin authentication and anti-replay features. ESP provides confidentiality in addition to all these features. We have identified that the attacks the DDT of a multicast group may suffer are: replay, sender address spoofing and DoS attacks. Among these, DoS is not a direct attack rather it is a consequence of the other two attacks. It should be noted that confidentiality is not a design requirement for our sender access control architecture. If confidentiality is required it would be achieved by deploying one of the group key management protocols that provides end-to-end data encryption at the application layer. Hence, IPsec with Authentication Header (AH) protocol will be implemented for our MSAs.

IPsec SA supports two modes of use: transport mode and tunnel mode. The transport mode primarily protects next layer protocols, while the tunnel mode is applied to tunneled IP packets. According to the IPsec specification [69], it is mandatory to implement an SA in tunnel mode if either end of the SA is a security gateway.

Therefore, if one of the ends of an SA is not the final destination or the source of the IP packet the SA is protecting, the SA must be implemented in tunnel mode. Hence, in our design, the MSAs (both the centralized and the distributed) must be implemented in tunnel mode.

The MSAs we have designed depend on automated key management. Therefore, a Group Key Management (GKM) protocol should be deployed. The GCKS uses a GKM to distribute MSA policy and keying materials. A GKM protocol is used when a group of IPsec devices require the same SAs. The IETF has developed a number of GKM protocols, such as Group Domain of Interpretation (GDOI) [113] and GSAKMP [41].

IPsec SA needs an end system to maintain three major databases: Security Policy Database (SPD), Security Association Database (SAD) and Peer Authorization Database (PAD). The multicast extension of IPsec [112] has further extended the scope of the SPD and renamed it as Group Security Policy Database (GSPD). In the following we have explained the step-by-step operations of the MSA with the roles of these databases:

Receiver authentication initiation: The PIM (S, G) Join message that carries an identity of a down-stream router will initiate the receiver authentication. Receiver authentication would be performed as the first step of the GKM protocol registration exchange. This would be triggered by the policy action defined by the Group Security Policy Database.

Receiver authentication through PKI: The Peer Authorization Database (PAD) contains the group's set of one or more trusted root public key certificates. The PAD may also include the PKI configuration data to retrieve the supporting certificates, which are needed for an end entity's certificate path validation.

Downloading MSA policy and keys: On successful GKM protocol registration,

the group member will be allowed to download the MSA policy and necessary keys to establish the MSA.

Security Association Database creation: A unicast Security Association Database (SAD) is identified using a Security Parameter Index (SPI), which is created by the receiver. However, to identify an MSA in addition to SPI, the sender address and the receiver address are used. In an MSA, the SPI is always chosen by the GCKS and the source address would be the address of the sender. The destination address would be a multicast group address (definitely different from the Secured group address to which multicast data are sent), used by the GCKS to multicast MSA related information (e.g., updated keys) to the receivers.

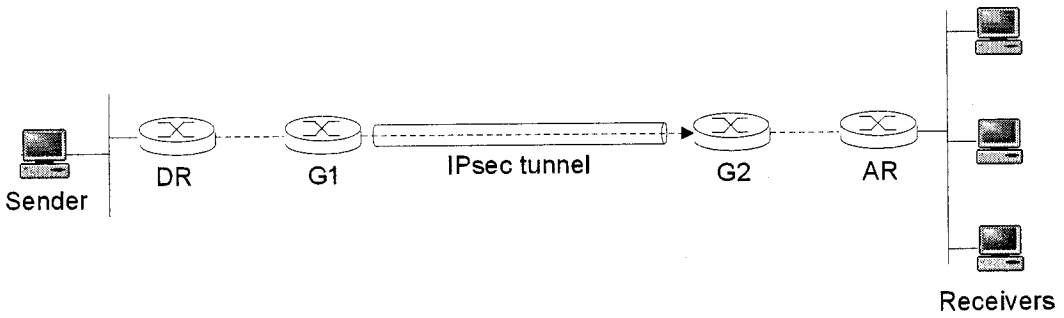

Figure 40: Address Preservation Mechanism for Multicast IPsec Tunnel

Address preservation in IPsec tunnel: Once an MSA has been established, the sender of the MSA will send multicast data using the IPsec tunnel. Although encapsulated multicast packets will be sent inside the IP tunnel, the address preservation scheme presented in [112] will be followed. This scheme is explained in Figure 40 for a multicast IPsec tunnel, where the sender, S is sending data to the multicast group, G. Two gateways—G1 and G2—are located between the sender and the receivers. An IPsec tunnel has been established between G1 and G2. According to [112], when G1 receives a multicast packet from S, it will encapsulate the packet inside the IPsec tunnel and will forward the packet to G2. However, while encapsulating the packet, G1 will use the addresses, S and

G, as the sender and the receiver addresses of the outer IP header. In case of a unicast SA, G1 had to use its own address as the sender and G2's address as the receiver addresses of the outer IP header.

Cryptographic packet authentication: By deploying one of our MSAs (either centralized or distributed), each multicast packet would be cryptographically authenticated before being forwarded to a domain. In the centralized MSA method, each member of the MSA will use the same IPsec SA in tunnel mode. Therefore, a BR, which is a receiver of the MSA, will be able to authenticate each packet on receiving it from an upstream router. If the packet is authenticated correctly, the router will forward the same packet. Given that the address preservation is followed the sender and the receiver addresses of the packet will remain unchanged. When the packet will reach an AR that has directly connected receiver, the packet will be decapsulated and would be distributed to the receivers through the shared medium.

If the distributed MSA method is deployed, a packet, each time entering to a new domain, would be cryptographically authenticated, decapsulated from the receiving IPsec tunnel and encapsulated inside the new IPsec tunnel. It should be noted that a BR may have directly connected receiver(s) also. In that case, it will function as both BR and AR. Then, the BR will perform the usual tasks of a BR (i.e., authenticating a packet, decapsulating and encapsulating if necessary, forwarding the encapsulated packet, etc.). Moreover, it will perform the tasks of an AR (i.e., authenticating a packet, decapsulating and distributing to the receivers)

Updating MSA policy and keys The GCKS may update the MSA policy and keys for different reasons, such as refreshing key materials periodically to maintain key hygiene or changing key materials due to a possible key theft. GCKS will multicast the updated policy and key materials to all the existing MSA members. We will discuss in detail about this issue again in section 8.5.3.

8.5 Comparison of Centralized and Distributed MSAs

In this section we will compare the two MSA schemes—centralized and distributed—in detail with respect to the establishment and maintenance cost, packet delivery time and security features.

8.5.1 Establishing the MSA

Establishing an MSA is composed of different sequential steps, such as forwarding the PIM (S, G) Join message to the GCKS, authenticating the router and downloading the MSA policy and keying materials. While comparing the two schemes in terms of the cost required to establish the MSA, we are not interested to know the exact cost required, rather we would like to compare in terms of some approximate values. Therefore, instead of counting the total round-trips required to establish the MSA, we are comparing the number of hops (or edges) the PIM (S, G) Join messages will travel. Given that the exact cost to establish an MSA could be found by multiplying the cost we are calculating (to forward PIM join messages) with some constant values, our comparison by calculating the cost to forward PIM join messages will not lead to fan erroneous result.

We would like to further clarify that in counting the number of hops PIM join messages are traveling, the term "hop" is used to mean two adjacent domains instead of two adjacent routers. This is clearly illustrated in Figures 37 and 39, where only the BRs and ARs are shown as nodes of the trees. According to these two figures, the distance between two nodes are counted as *one hop* in the following calculations, while the physical distance is equal to *one domain* in reality. Hence, the cost to build an MSA turns into the cost to build the DDT. In computing the costs, first,

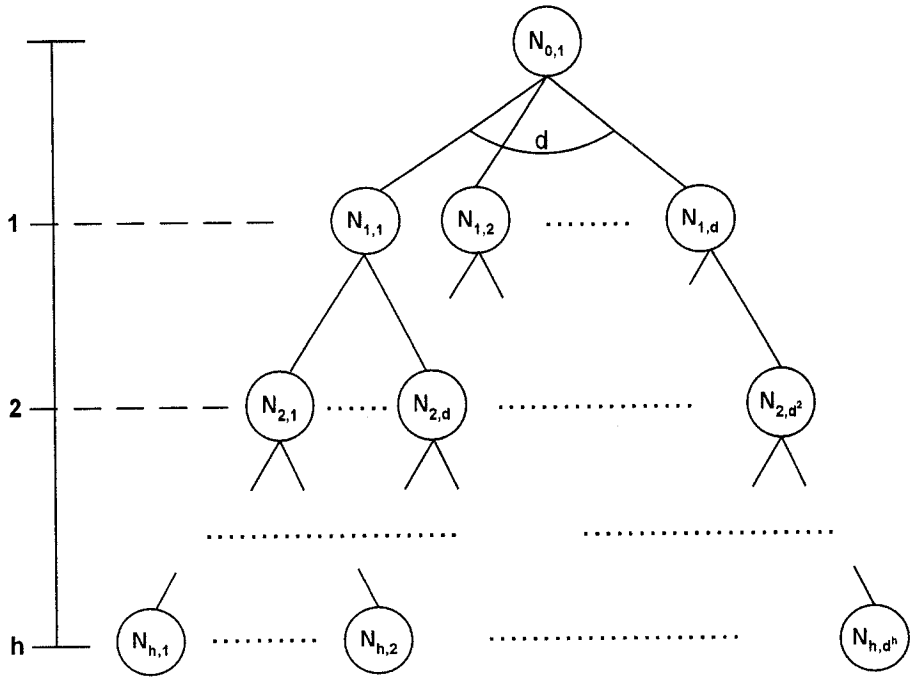

Figure 41: A *d-ary Full* Tree with Height h

we have followed the worst case analysis (with respect to the centralized method) by considering the *full* trees for both centralized and distributed methods. In a *full* tree every node (except the leaf nodes) must have maximum number of possible children. Therefore, in a *d-ary full* tree, every node (except the leaf nodes) must have d children. While computing the costs, we are considering two parameters of the tree: the height and the number of maximum children. In Figure 41, a *d-ary full* tree with height h has been shown.

8.5.1.1 Centralized MSA

First, we will calculate the required cost to build the centralized MSA. The cost will be calculated by counting the total number of hops or edges that PIM (S, G) Join messages will travel. In Figure 42, the PIM join messages for a simple binary ($d = 2$) tree with $h = 2$ is shown. In this figure, the ARs join the MSA in the order of AR_{21}, AR_{22}, AR_{23} and AR_{24}. It should be noted that, in an MSA, a BR never joins by

itself, rather it is triggered by receiving an (S, G) Join message from a down-stream BR or AR. Hence, in Figure 42, BR_{11} and BR_{12} join the MSA when they receive a PIM join message from the down-stream ARs. Moreover, in the centralized MSA, all the nodes except the root (i.e., the sender), join as receivers. Therefore, all the PIM join messages must be forwarded up to the root. This is clearly illustrated in Figure 42, where the join messages are presented by arrows. Hence, the number of arrows will be equal to the number of hops or edges that we are interested to count. We can count this number by using the height of a tree. If we observe Figure 42 carefully, we will notice that, the number of edges we are interested to count is exactly equal to the summation of the heights of the leaf nodes. By using induction, it can be proved that in a *full d-ary* tree with height h, the number of leaf nodes is equal to d^h and the summation of the heights of the leaf nodes is equal to $h.d^h$. We will further use this finding when we compare the MSA construction costs of the two methods.

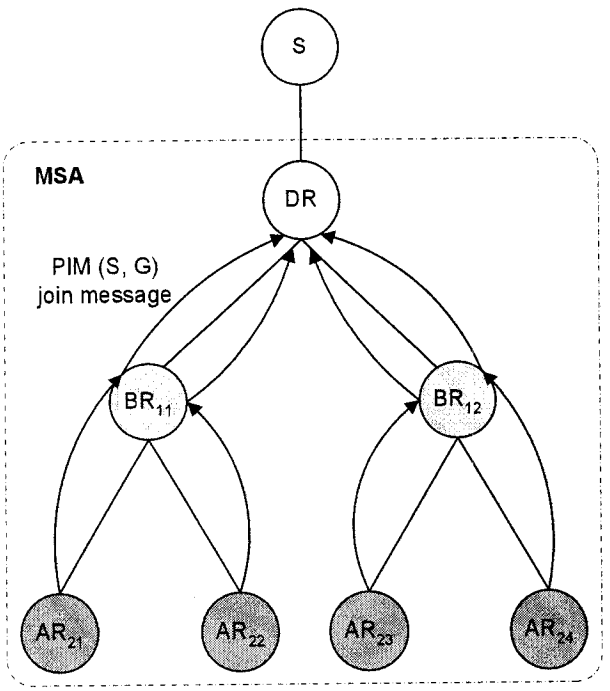

Figure 42: A Centralized MSA Tree

8.5.1.2 Distributed MSA

In Figure 43, the PIM join messages to build the distributed MSAs, for a simple binary ($d = 2$) tree with $h = 2$ are shown. In this figure, the ARs join the MSAs in the order of AR_{21}, AR_{22}, AR_{23} and AR_{24}. There are three small sized MSAs: MSA1, MSA2 and MSA3. In distributed MSAs, an (S, G) Join message is not always forwarded up to the root. If a BR has already joined to an MSA previously (by forwarding a PIM join message to its upstream router), the BR will not forward the

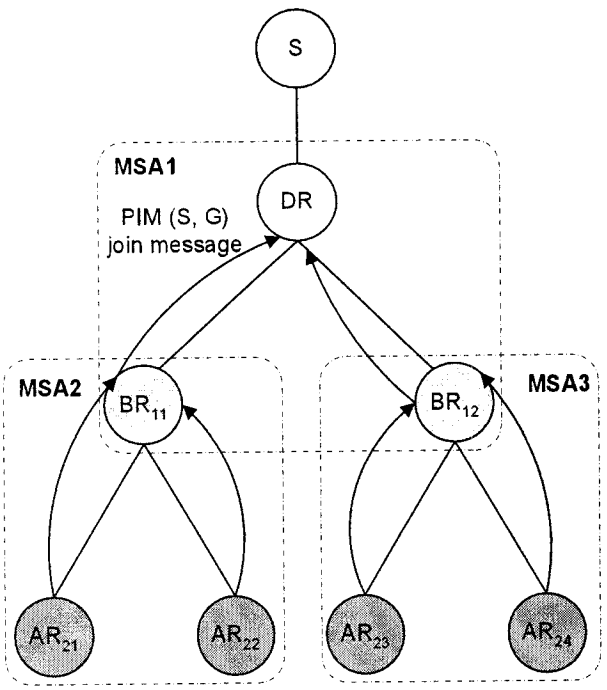

Figure 43: A Distributed MSA Tree

same PIM join message any more. For example, in Figure 43, AR_{21} will send the first (S, G) join message, which will be forwarded to the root by BR_{11}. Consequently, AR_{21} and BR_{11} will join MSA2 and MSA1, respectively. Next, AR_{22} will send a PIM join message, and will join MSA2. However, BR_{11} will not forward this join message to the root, given that BR_{11} has already joined MSA1. Hence, it can be concluded that each edge of the tree should be traversed only once. Therefore, we are interested in counting the number of edges of the tree. By using induction, it can be proved

that in a *full d-ary* tree with height h, the number of edges is equal to $\frac{d}{d-1}(d^h - 1)$. This finding will be used in the following section to compare the two methods.

8.5.1.3 Comparison of Performance

We have computed the number of edges or hops PIM (S, G) Join messages will travel for different heights (h) and fanout or number of children (d) to build MSAs using centralized and distributed methods. The results are shown in Figures 44, 45 and 46 for d equal to *2*, *3* and *4*, respectively.

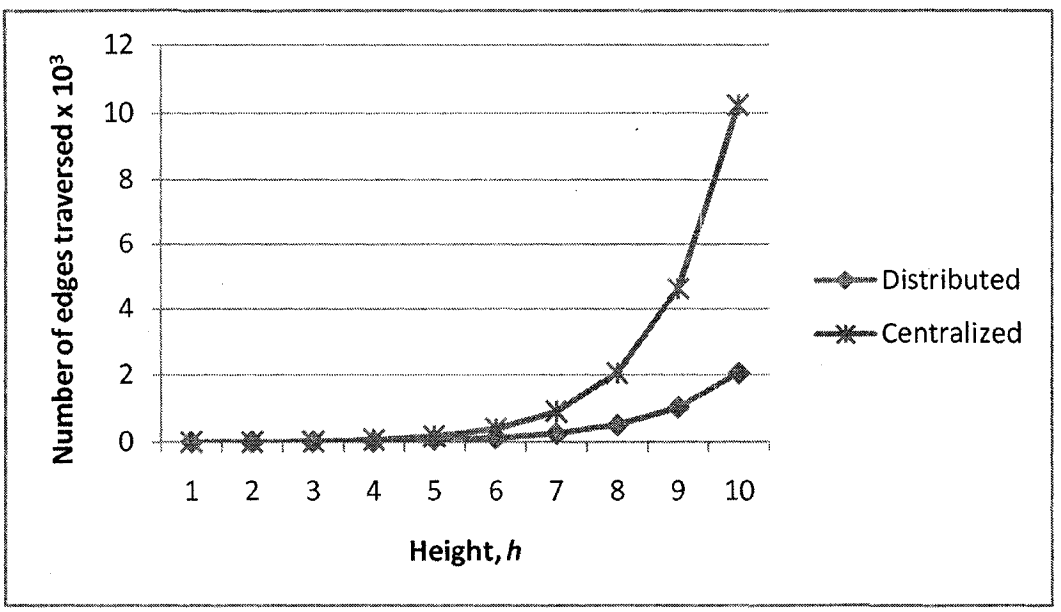

Figure 44: Number of Edges PIM Join Traversed vs. Height (h) for $d = 2$ Tree

The results we have obtained are as we expected, and clearly justifiable. For *full* trees, the centralized MSA method needs a very large number of messages compared to the distributed method. Furthermore, by studying the figures, it could be stated that the change in the number d or fanout has no significant effect in the relative or comparative results. The distributed MSA method always require fewer communications between different routers. We have studied the worst-case analysis (for the centralized MSA method) by considering the *full* trees only, which is indeed an ex-

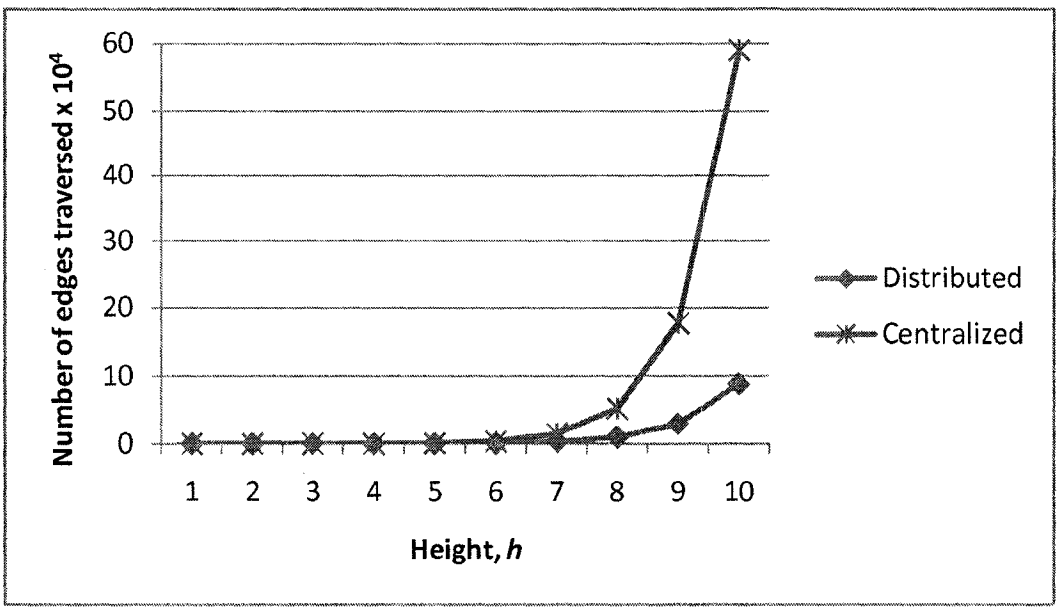

Figure 45: Number of Edges PIM Join Traversed vs. Height (h) for $d = 3$ Tree

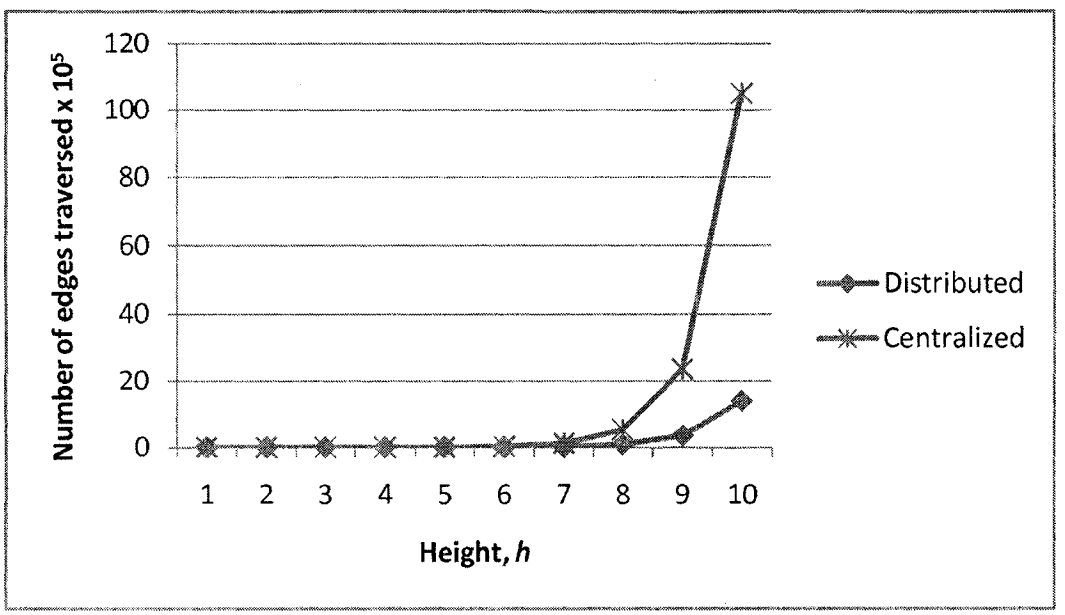

Figure 46: Number of Edges PIM Join Traversed vs. Height (h) for $d = 4$ Tree

treme scenario. Next, we will study the worst-case analysis for the distributed MSA method by considering a tree with d nodes in each level.

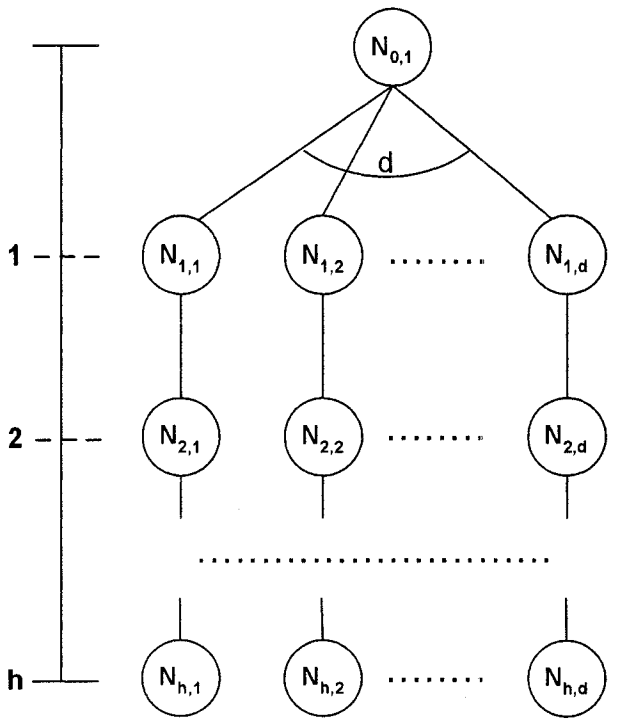

Figure 47: A Tree of Height h with d Nodes in each Level

In Figure 47, a tree of height h with d nodes in each level has been shown. Each of the internal nodes of the tree has only one child, except the root, which has d children. To calculate the DDT building costs, first, we have to replace the root with the DR or one-hop router of the sender, the internal nodes with the BRs and the leaf nodes with the ARs. If we count the number of edges or hops that PIM (S, G) Join messages will travel, we will find that for both centralized and distributed MSA methods, the results will be equal to $h.d$. Hence, both methods will perform the same to build the MSAs. It should be noted that for the distributed MSA method, this is the worst-case scenario in terms of the average number of edges needed for a new AR to join. This could be calculated by the ratio of the total number of edges PIM join messages need to travel and the number of ARs or the leaf nodes of the tree.

The tree structure shown in Figure 47 could be found only if the receivers are

distributed over a very large geographical area, and the number of receivers is very small with compared to the geographical area they are distributed over. However, if the number of receivers is very large, and they are densely distributed over a small area, the tree structure should be similar to one shown in Figure 41. Therefore, it is worth while to note that the average performance of the two methods lies somewhere between the two worst-cases.

8.5.2 Maintaining Databases and Keying Materials

For the successful deployment of an MSA, the sender and the receivers will have to maintain different databases (i.e., Group Security Policy Database, Security Association Database and Peer Authorization Database) and keying materials. In the centralized method, all the participant routers are member of a single MSA. Hence, they have to maintain databases and other information for a single MSA. However, in distributed MSAs, except the DR and the leaf nodes (i.e., the ARs), each internal node (i.e., the BRs) is a member of two MSAs. A BR is a receiver of the incoming MSA and is the sender of the outgoing MSA. Therefore, the BRs in the distributed method have to maintain databases and keying materials for two MSAs.

8.5.3 Updating Policy and Keys

There is a risk of possible theft of a secret key in case the same key is used for a prolonged period of time. This is a common phenomenon when keys are managed manually. Hence, the keying materials and policy information of an MSA need to be updated periodically by the GCKS. If a secret key is compromised, the GCKS should create and distribute a new set of keying materials to all the existing members. Moreover, to achieve perfect backward secrecy (guarantees a newly joined member will not understand previous information) and perfect forward secrecy (guarantees a

leaving member will not understand future information) GKM protocols enforce key updates in a dynamic group. In general, updated keys need to be multicast to all the group members of an MSA by the GCKS.

If we compare the two MSAs, the distributed method will provide scalability and flexibility as it is composed of many small MSAs. We have further clarified this in the following by considering different scenarios individually:

Periodic key update: If the cost to update a centralized MSA is compared with the total cost to update each small distributed MSA, then both of them will incur similar cost. However, the distributed MSAs provide flexibility by allowing the local GCKS to update each small MSA independently at a convenient time (e.g., when traffic is less) when resources are available.

Replacing keys: If the existing keys need to be replaced due to a compromised key or to achieve perfect forward/backward secrecy, the distributed MSAs will perform significantly better than the centralized MSA. The GCKS of the MSA that needs to replace the keys will distribute keying materials only to the members of the MSA. Therefore, in the distributed method the local GCKS only needs to communicate with a small number of routers. However, in the centralized method the global GCKS needs to communicate with all the routers.

8.5.4 Packet Delivery Time

For IP multicast based applications packet delivery time is a very significant feature. IP multicast based applications, more specifically SSM applications, will be mostly deployed for real-time and bandwidth intensive applications, such as Internet TV, multi-player online gaming, video-conferencing, etc. Therefore, higher packet delivery time due to delay in processing at the routers will deteriorate the quality of service and the quality of the users' experience.

The centralized method will provide a faster packet delivery when compared to the distributed method. In the centralized method, the internal routers or the BRs will follow the sequence: receive an encapsulated packet inside an IPsec tunnel, decapsulate the packet and cryptographically authenticate it, and if authenticated forward the incoming packet (i.e., the encapsulated packet) without further processing. Given that only one MSA will be deployed and the address preservation mechanism [112] will be followed, a BR will not have to create an encapsulated packet to forward through the IPsec tunnel.

However, in the distributed method, a BR will have to perform an additional task. It should create a new encapsulated packet inside the IPsec tunnel each time it forwards a packet using the outgoing MSA. This will increase the average packet delivery time. It is worth while to note that the additional packet delivery time is directly proportional to the height (h) of the tree to be constructed by the distributed MSAs (see Figures 39 and 43). This height is equal to the number of domains a packet travels before it reaches the final destination.

8.5.5 Security Control

It is reasonably easier to establish secure communication among a small number of participants, specially, if they are closely distributed. Therefore, establishing and maintaining a centralized MSA with a large number of recipients is much harder compared to many small sized distributed MSAs. The greater the number of participants, the larger the possibility of leakage of secret keys. Moreover, in distributed MSAs, the local GCKS has freedom to deploy any authentication scheme it wants depending on local needs and available hardware and software resources.

8.5.6 Trust Relation

In centralized MSA, all the participant routers are members of a single trust domain. Inter-domain trust relation is always difficult to establish and maintain. Therefore, in centralized MSA, it is quite difficult to establish a single trust domain for many routers which are distributed over multiple domains. One the other hand, for a distributed MSA, it is relatively easy to establish and maintain a number of small sized trust domains, which are administrated locally by the administrators of the networks.

8.5.7 Summary

In Table 6, a summary of comparison of the centralized and the distributed MSAs has been presented. It is shown that both methods have pros and cons depending on

Table 6: Summary of Comparison of the Centralized and the Distributed MSAs

Features	**Centralized MSA**	**Distributed MSAs**
Establishment cost	High. In worst-case, $h.d^h$. In best-case, $h.d$.	Low. In best-case, $\frac{d}{d-1}(d^h - 1)$. In worst-case, $h.d$.
Maintenance	All members maintain a single MSA.	Only the root and the leaves maintain a single MSA. Internal nodes maintain two MSAs
Updating	Less scalable and flexible. Should update all members if needed.	Scalable and flexible. A small MSA might be updated independently.
Delivery time	Fast. BRs need not create IPsec encapsulated packet.	Slow. BRs have to create IPsec encapsulated packet.
Security	Less flexible. All the routers use same authentication method and keys.	Flexible. Individual MSA may deploy different authentication method and keys.
Trust relation	Difficult to establish and maintain a large trust domain.	East to establish and maintain many small-sized trust domains.

the features. For example, for a dense multicast group, if the format of the DDT is similar to a *d-ary full* tree with a reasonably small height, h, the distributed MSA method will perform better than the centralized method. On the other hand, if we consider a sparse multicast group, the format of the DDT will be similar to the one shown in Figure 47. Moreover, if the height, h is large, the centralized method will provide the same establishment cost with fast packet delivery. Hence, the network administrators should pick one of the methods depending on the distribution of the group members and the priority of the requirements.

Chapter 9

Mobile Receiver Access Control and Secured Handoff

IP multicast will face new challenges if it has to control mobile EUs accessing valuable data in wireless networks. In absence of receiver access control the operators of the wireless networks will be intimidated to deploy IP multicast due to possible wastage of valuable bandwidth and other resources from joining of any unauthorized mobile EU and unwanted extension of the distribution tree. Moreover, multicast suffers from Denial of Service (DoS) attack more severely than any other attack because of its amplification of data packets enroute. Although, in mobile networks, both source(s) and receivers of a multicast group could be mobile, mobility of source(s) is not very common and is difficult to support. Furthermore, most of the promising multicast applications (e.g., Internet TV, online radio) with high potential to generate revenue have single source, and thus, depend on the Source-Specific Multicast (SSM) [47] service model. Given that these types of applications are hosted from well known source addresses, mobility of sources is extremely rare. Hence, in this chapter, we have considered receiver or EU mobility only.

If an authenticated EU moves to a new network, the secured relation between the EU and the Multicast Router (MR) or Access Router (AR) of the leaving network would be lost. The MR of the newly visited network should authenticate and authorize the EU again before sending a request to the upstream CRs to extend the DDT up to the visited network. However, this reauthentication or secured handoff will increase the IGMP-AC join latency, and thus, disrupt the group activity.

In this chapter, we have broadened the scope of IGMP-AC by demonstrating the usability of IGMP-AC in wireless networks for mobile receiver (or EU) access control [61]. In addition, we have presented a secured EU handoff mechanism using IGMP-AC with low join latency, which deploys the EAP Re-authentication Protocol (ERP) [90].

9.1 Related Work and its Limitations

Receiver access control and secured handoff in mobile multicast have not been addressed well by researchers. In this section, we have listed the related work that considered IGMP and/or handoff in mobile multicast. We have excluded the research work that is devoted to other issues (e.g., routing protocol [74], group key management [76]) of mobile multicast.

In [67], an extension to IGMPv2 [30] has been proposed to enhance multicast communication for mobile hosts by aggregating many multicast group addresses in a single IGMP message. The extended IGMPv2 is expected to reduce the overhead of creating the bi-directional tunnel of mobile IP.

A seamless handoff scheme for multicast receivers has been designed in [86], by joining the multicast distribution tree in the new subnet in advance while terminating the old connection more rapidly. This is indeed a hybrid method of remote

subscription and bi-directional tunneling.

To reduce handoff latency for multicast receivers, Multicast Handoff Agent (MHA) mechanism has been proposed in [73] that deploys MHA in each base station. An MHA acts as a proxy for the mobile nodes and replies to the periodic IGMP query messages sent by the multicast router. It also sends unsolicited reports when a mobile node moves to another cell without waiting for the IGMP query. In [65], a similar IGMP proxy approach has been developed for Highspeed Portable Internet (HPi) networks to reduce handoff latency, where the IGMP proxy resides in a Packet Access Router (PAR).

In [97], an adaptive solution for bi-directional tunnel method has been proposed that allows the Home Agent (HA) not to send periodic IGMP queries. Hence, a mobile host may go to sleep mode to conserve battery power when no multicast traffic is present. In addition, the HA saves significant CPU cycles by not sending periodic membership messages through bi-directional tunnel to all remote mobile receivers.

In [77], IGMP/MLD (Multicast Listener Discovery [110]) requirements for host mobility have been identified. They have suggested a number of techniques to minimize handoff latency, such as, sending unsolicited join without waiting for a query when a mobile host moves to a new network, keeping the receptions states of leaving network until the new one has been established, mobile-initiated handoff, adjusting values of different timers and counters of IGMP/MLD host and router, etc.

IGMP and MLD protocol extensions for mobile hosts and routers have been described in [6]. They have advised a number of steps, such as, keeping track of membership status by eliminating a report suppression mechanism (which was done in IGMPv3), tuning the IGMP/MLD query timers and the number of responses, and using source filtering mechanisms of Lightweight IGMPv3/MLDv2 protocol [78], etc.

From the brief discussion on the existing methods, it could be concluded that the

researchers have focused in two issues: reducing handoff latency and optimizing the communication between the mobile host and the IGMP router. Different techniques have been proposed to fulfill these two criteria, such as, IGMP proxy, changing timers of query and response intervals, mobile host initiated join, advance join to the visiting network, etc. Undoubtedly, the problems they have addressed are real problem, and thus, should be investigated. However, one of the most important requirements, which is mandatory to implement for the deployment of both stationary and mobile multicast service models, has remain overlooked. The necessities of admission control or access control for IP multicast are clearly defined by the Internet standard bodies, such as, IETF [44] and ITU-T [101]. Furthermore, in mobile multicast, secured handoff (re-authenticating a mobile EU, when he/she moves to a new network) is a pre-condition from the operator's point of view. Therefore, a receiver access control architecture with secured and fast handoff is required for mobile multicast to support the Network Service Providers (NSP) ability to authenticate, authorize and charge the mobile EUs. Additionally, there should be provision for distributing the revenue generated by the deployment of mobile multicast to two parties: the NSP for allocating network resources and the Content Provider (CP) for delivering the multicast data.

9.2 Requirements of Our Design

Our goal is to develop a receiver access control architecture with secured and fast handoff. Hence, we have categorized the requirements into two divisions:

1. Receiver Access Control

- The requirements, listed in section 3.3, for a receiver access control architecture would be applicable. Moreover, the requirements, listed in section 4.2.1, for

IGMP-AC protocol would also be satisfied.

2. Secured Handoff

- The proposed architecture should provide a secured handoff mechanism for mobile EUs. Every time an EU visits a new network, the EU's identity must be authenticated, and his/her authorization information must be checked before allowing reception of any multicast data for a secured group. However, the join latency should be minimum to reduce the handoff time.
- Given that the visited wireless network has implemented IGMP-AC, the secured handoff method should be independent of the underlying structure of the visited network.

9.3 Proposed Architecture

The proposed architecture to provide receiver access control and secured handoff for mobile multicast EUs is shown in Figure 48. This architecture has been designed by extending the IGMP-AC architecture for stationary EUs (see Figure 12). Hence, the proposed architecture has similar entities to what its predecessor has. In addition, the EU is a Mobile Node (MN), and may move from one network to another or from one NAS to another inside the same network. The NAS part of the AAA framework is assumed to be implemented in the Multicast Router (MR), which performs the task of an AR. The MR is an IGMP-AC router and communicates with the MN (EU) using the IGMP-AC protocol. Moreover, it communicates with the Local AAAS (LAAAS) using a AAA protocol. Three wireless networks are shown in Figure 48: NW1, NW2 and NW3. The participants maintain long-term credentials and an account relation with the AAAS of NW1, which is referred to as Home AAAS (HAAAS). Hence, the HAAAS has direct access to the participants' database and is able to verify the

participants' authentication and authorization information. It is also assumed that the communications between the NAS and the LAAAS, and between the LAAAS and the HAAAS are secured (i.e., integrity, mutual authentication and confidentiality are attained). Next, we have explained the receiver access control and secured handoff separately.

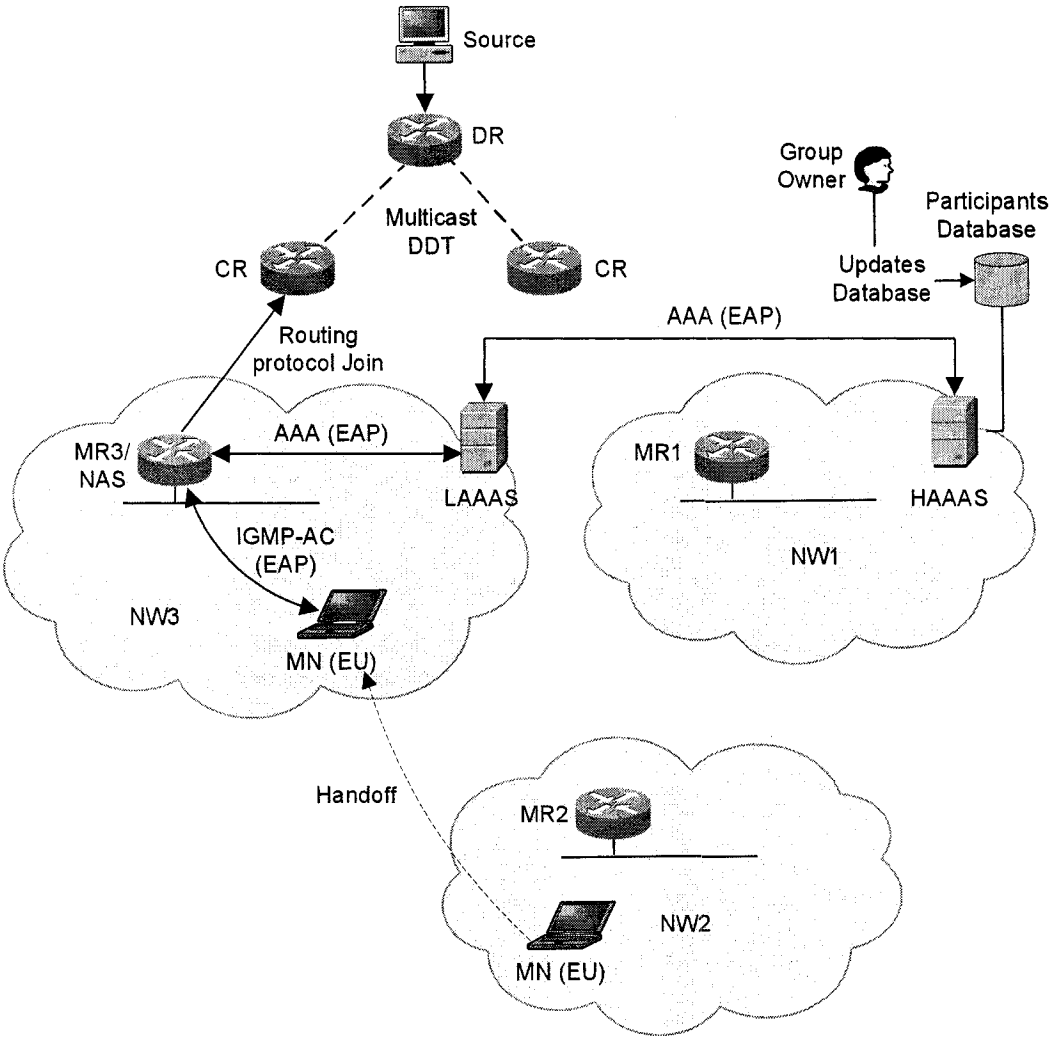

Figure 48: IGMP-AC in Mobile Multicast

9.3.1 Receiver Access Control

The receiver access control procedure is explained in Figure 49. In the proposed architecture, the MR is an IGMP router, and thus, sends a periodic IGMP-AC query (i.e., regular IGMPv3 query), which is received by the MN/EU. An incoming EU will receive this query and should send an IGMP-AC report message (i.e., regular IGMPv3 report) expressing the EU's interest in receiving multicast data destined to a secured group. However, to speed up the handoff an EU may also send an unsolicited IGMP-AC report message while moving to a network. The EU will send a Network Access Identifier (NAI) consisting of an user name and domain name inside the IGMP-AC report message.

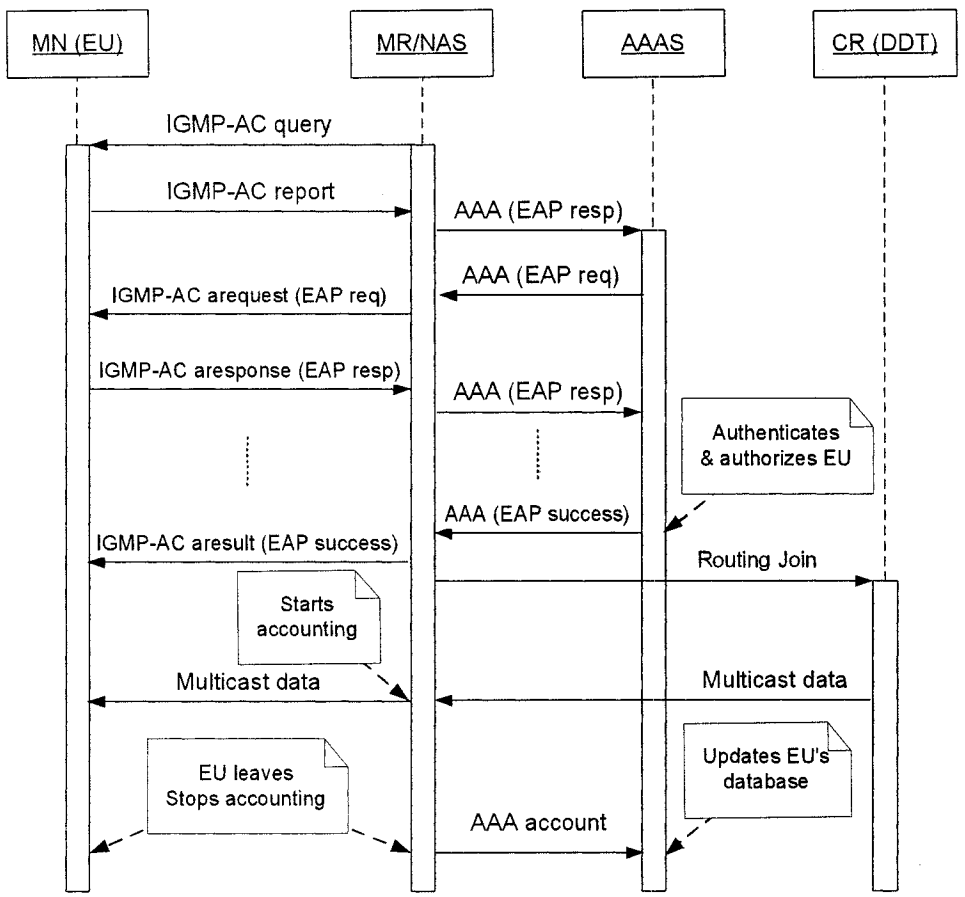

Figure 49: Receiver Access Control Sequences

Given that the EU is willing to join a secured group, the report message will trigger

an EAP authentication session. The MR will send an EAP response message to the LAAS carrying the NAI of the EU. From the domain name of the NAI, the LAAAS will get the address of the HAAAS and will forward the EAP response message to the HAAAS. The communications between the LAAAS and the HAAAS are not shown in Figure 49. The HAAAS will return an EAP request message to the LAAAS, which will forward the message to the MN. EAP request and response messages will be encapsulated over IGMP-AC `arequest` and `aresponse` messages respectively. `arequest` and `aresponse` are unicast messages, and are added in IGMP-AC [57]. Depending on the EAP method used, multiple round trips might be required to authenticate the EU between the MN and the HAAAS. However, in the next section, we have presented a scalable way of authenticating a MN that requires only one round trip between the MN and the LAAAS. If the EU is successfully authenticated and authorized, the HAAAS will send an EAP success message to the MR (through the LAAAS), which will forward the message to the EU inside an IGMP-AC `aresult` message (a unicast message, added in IGMP-AC).

On successful authentication and authorization of the MN/EU, the NAS/MR will send a multicast routing protocol join to the upstream CR to extend the multicast DDT. Hence, multicast data will be received by the MR/NAS, and will be forwarded to the EU. The MR/NAS will start accounting at this point.

Once the EU has left the secured group (either by sending an IGMP-AC leave message and going through EAP authentication, or by silently moving to another network), the MR/NAS will stop accounting. The accounting data (both usages of the network resources and multicast data) should be sent to the LAAAS. The usage of network will be used to charge (through a roaming agreement) the home network authority of the EU. The usage of the multicast data would be forwarded to the HAAAS of the EU.

The requirement that an IGMP report with a secured group address would only be

processed by the AR if the report message had been originated by an authenticated EU is achieved in IGMPv3, because the host membership report suppression was removed in this version and because of the receiver specific reception states maintained by the IGMP-AC router [57]. An EU will always send report message in response to an IGMP query, and an IGMP-AC router that maintains EU specific reception states will be able to verify if the report has been generated from a previously authenticated EU. Finally, an EAP authentication session would be triggered in case of a reception state change report (i.e., join or leave) message.

9.3.2 Secured Handoff using IGMP-AC

When an MN/EU moves to a new network, the EU must be authenticated first. Therefore, an EAP re-authentication session should be executed between the EU and the NAS of the visited network. The NAS will have to communicate with the LAAAS to re-authenticate the EU, and the LAAAS communicates with the HAAAS to complete the EAP re-authentication session. Some EAP methods reuse the initial authenticated states to reduce the computation overhead of re-authentication. However, in most EAP methods (e.g., in EAP-AKA [5]), at least two round trips between the original EAP server (i.e., HAAAS) and the local NAS are required to complete the EAP re-authentication. Furthermore, the HAAAS might be located a long way from the LAAAS. Hence, the handoff latency will be significantly increased. One way to reduce this latency is to use an improved EAP keying framework [79], which distributes the key distribution functionality to the visited domain through the LAAAS. The Handover Keying (hokey) [38] Working Group at the IETF is currently designing a more comprehensive and generic solution for the EAP re-authentication problem by developing an EAP Re-authentication Protocol (ERP) [90]. We have used this protocol in our design for secured and fast handoff of a mobile EU.

9.3.2.1 EAP Re-authentication Protocol (ERP)

The EAP Re-authentication Protocol (ERP) provides an EAP method independent re-authentication protocol for a peer that has valid and unexpired key materials with its home EAP server. These key materials had been established during an EAP authentication that took place previously between the peer and the home EAP server. The goal of ERP is to support a fast and secured re-authentication by reusing the unexpired keys. An example of an ERP protocol run has been shown in Figure 50. In the ERP protocol, three entities—ER peer, ER authenticator and ER server—are involved. The functionalities of these entities are exactly same as they perform in the EAP protocol. However, the only difference is instead of EAP, ERP is used as communication protocol. ERP is a single round trip protocol between the peer and the server. At the end of a successful ERP exchange, the peer and the server authenticate mutually, and establish necessary key(s) to form a Security Association between the peer and the authenticator.

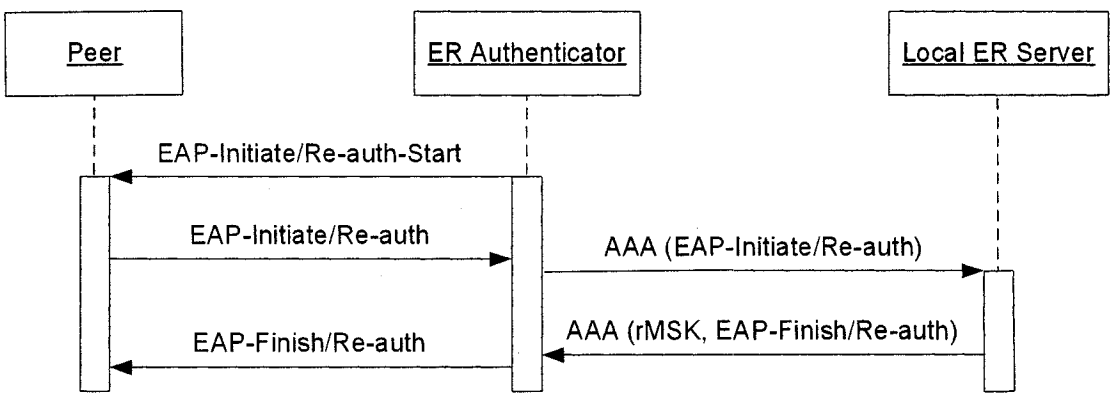

Figure 50: EAP Re-authentication Protocol (ERP)

The ERP specification [90], defines two new ERP messages, EAP-Initiate and EAP-Finish. A peer may trigger the re-authentication by sending an unsolicited EAP-Initiate/Re-auth message to the ER authenticator, or the peer may respond to the EAP-Initiate/Re-auth-Start message sent by the authenticator (see Figure 50). The authenticator forwards the EAP-Initiate/Re-auth message encapsulated inside a

AAA protocol message to the local ER server. On successful authentication of the peer, the local ER server sends EAP-Finish/Re-auth and rMSK (a secret key, generated by the local ER server for this ERP run only) to the authenticator. Finally, the authenticator sends EAP-Finish/Re-auth to the peer to confirm the authentication.

9.3.2.2 ERP Key Hierarchy

ERP supports a complex hierarchical key management for domain specific authentication of a peer and key distribution to the authenticator. In Figure 51, a partial ERP key hierarchy has been shown. On successful completion of an EAP authentication (for the first time), the peer and the EAP server (i.e., home EAP server) establish two keys: MSK and EMSK. The MSK is used for the subsequent communication between the peer and the server. However, the EMSK is used in generating hierarchical ERP keys. A mobile peer generates a set of domain specific keys when it moves to a new domain or network. First, the peer computes a Domain Specific Root Key (DSRK) using the EMSK and the domain name the peer is visiting. The local domain is responsible for announcing the domain name via the lower layer to the peer. If the peer does not know the domain name via the lower layer announcement (e.g., due to a missed announcement or lack of support for domain name announcements in a specific lower layer), the peer should initiate an ERP bootstrap exchange with the home ER server (EAP server) to obtain the domain name. The local ER server must be in the path from the peer to the home ER server (EAP server). In response to the bootstrap exchange, the home ER server sends the DSRK and the domain name. The local ER server keeps a copy of the DSRK and the domain name before forwarding to the peer. Next, using the DSRK and the domain name, the peer computes a Domain Specific re-authentication Root Key (DS-rRK). The DS-rRK is used only as a root key for re-authentication and is never used to protect any data. To prove the possession of the DS-rRK, the peer computes another key, Domain Specific root Integrity Key (DS-rIK). The DS-rIK is actually used in the mutual authentication of the peer and

the local ER server. The integrity of the EAP-Initiate/Re-auth message is protected by the DS-rIK. In addition, the peer also computes rMSK from the DS-rIK.

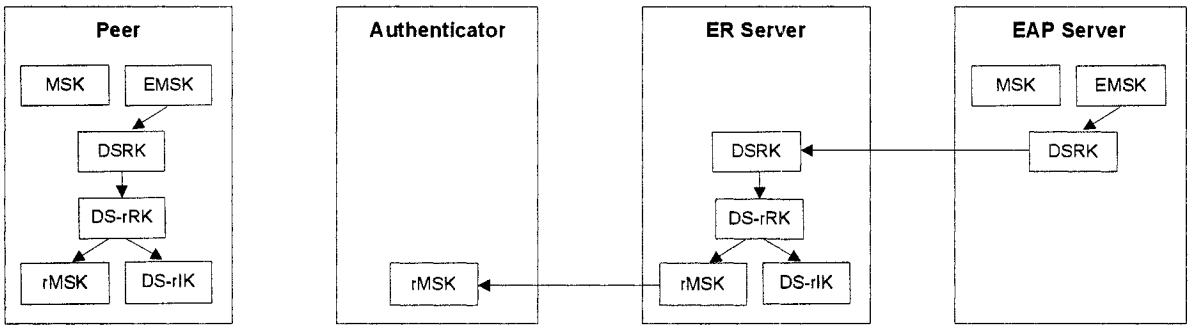

Figure 51: ERP Key Hierarchy

The EAP server computes DSRK from the EMSK and the domain name. A DSRK is transported to the ER server in two ways by the EAP server: by adding the DSRK at the end of a successful EAP authentication or in response to an ERP bootstrap request generated by the peer. On receiving the DSRK, the ER server computes DS-rRK, DS-rIK and rMSK following the same algorithm that the peer used. Now, the ER server is able to verify the integrity of the EAP-Initiate/Re-auth message. On successful authentication (i.e., integrity checking), the ER server sends EAP-Finish/Re-auth message (integrity protected by the DS-rIK), which the ER server has computed, and the rMSK to the authenticator. The authenticator will forward the EAP-Finish/Re-auth message to the peer. Finally, the peer should verify the integrity of the message. Hence, at the end of an ERP session, the peer and the ER server mutually authenticate using the possession of the right integrity key, DS-rIK. Furthermore, the peer and the authenticator establish rMSK. The algorithms to be followed in generating the keys we have mentioned are explained in [90] and [98].

The ERP supports both micro (moving from one NAS to another inside the same domain) and macro (moving from one domain to another) mobilities. These two types are explained in Figure 52, where the peer maintains long-term credentials with the Home EAP Server. The peer is attached to NAS11 of Domain 1 first, and executes a full EAP authentication with the Home EAP Server. Next, the DSRK1 is transported

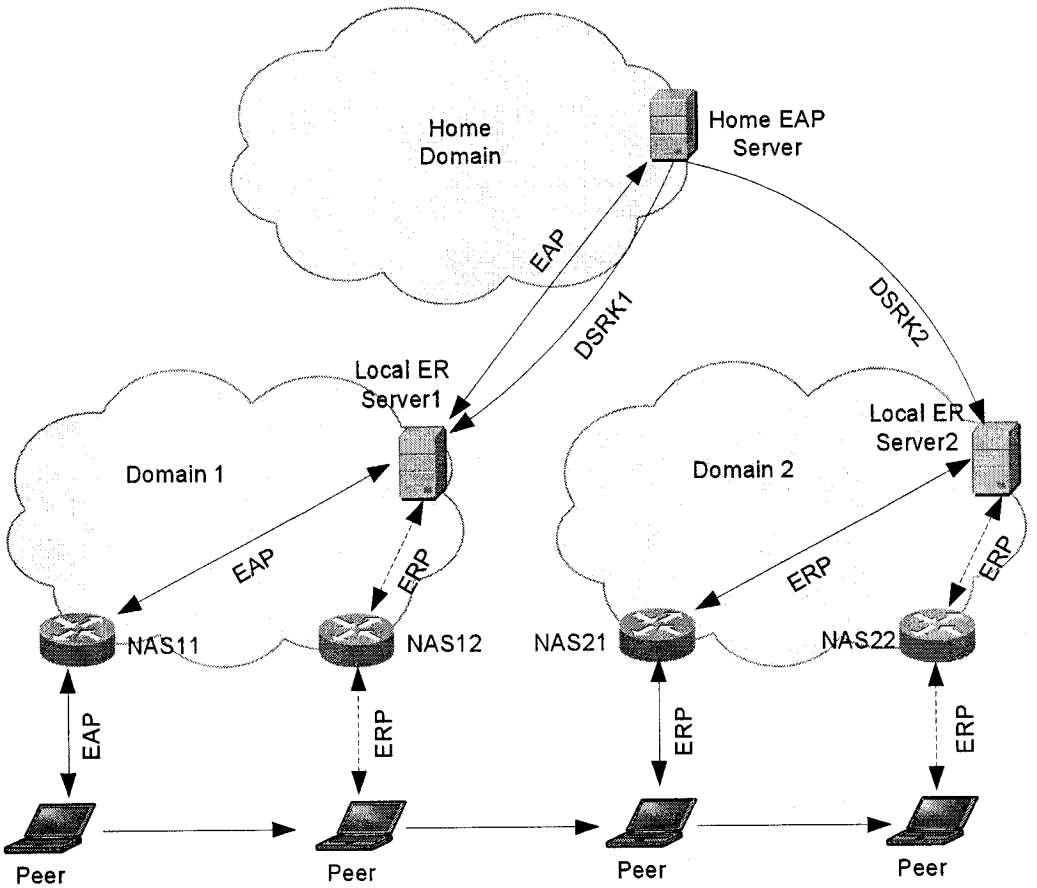

Figure 52: Different Types of Mobility using ERP

to the Local ER Server1. When the peer moves from NAS11 to NAS12 (inside Domain 1), the Local ER Server1 can authenticate the peer using DSRK1 without communicating with the Home EAP Server. When the peer moves to Domain 2, the Home EAP Sever is requested to send DSRK2 (by ERP bootstrap). DSRK2 is used to authenticate the peer in Domain 2. Similarly, when the peer moves to NAS22, the Local ER Server2 can authenticate it using DSRK2.

It should be noted that when a peer moves to a new domain, QoS and other parameters will have to be negotiated during handoff. The solution we have developed will reduce the handoff latency (authentication and authorization of a MN/EU of a multicast group) by replacing the multiple long round-trips (between the peer and the remote EAP server) with a short one round-trip (between the peer and the local

ER server). The issue of handoff latency and QoS negotiation are separate issues.

9.3.2.3 ERP Encapsulation over IGMP-AC

ERP encapsulation over IGMP-AC will minimize the number of round trips required to authenticate and authorize a mobile EU, when he/she moves to a new domain or moves to a new NAS inside a domain. Given that ERP supports different types of mobility scenarios, we will explain one example with message sequences to explain the ERP encapsulation and EU access control. The message sequences shown in Figure 53 depict the scenario of a mobile EU moving from one authenticator to another inside the same domain.

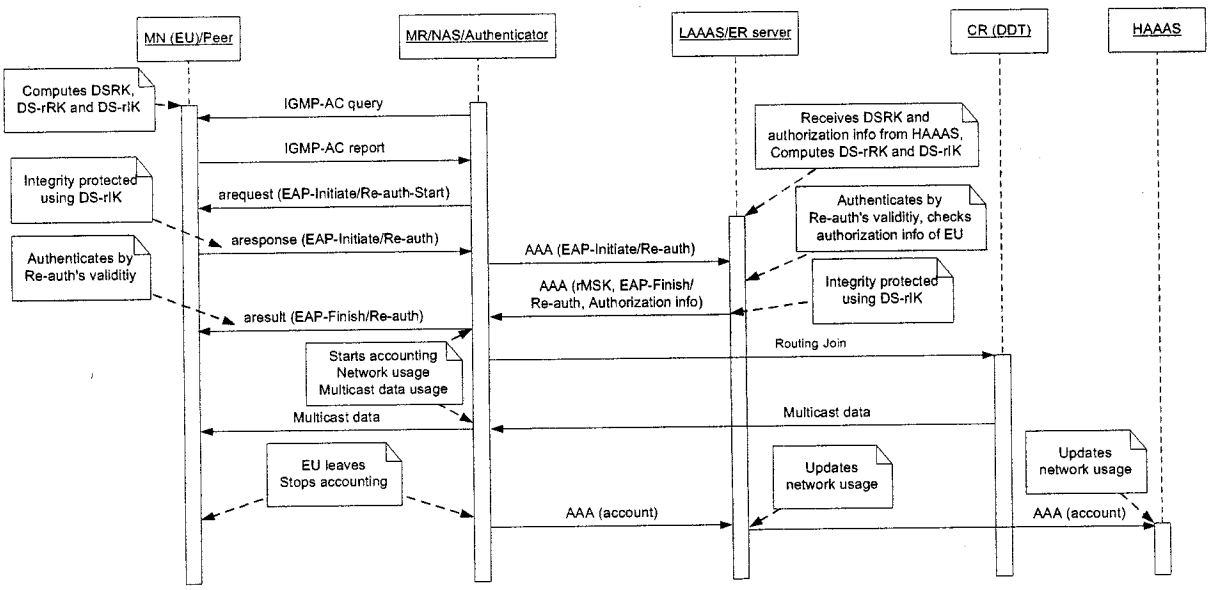

Figure 53: An EU Handoff Inside the Same Domain

The MR/NAS acts as an IGMP querier and sends periodic IGMP-AC queries that will be received by the MN (EU). The secured handoff will start with the IGMP-AC report, which the MN (EU) sends either in response to the IGMP-AC query or without receiving any query. The IGMP-AC report will carry the group address of a secured multicast group, and hence, it will trigger an ERP session between the MR (ER authenticator) and the MN (ER peer). The MR will send the first ERP message, EAP-

Initiate/Re-auth-Start, encapsulated inside an IGMP-AC `arequest` message. The MN is switching inside from one authenticator to another (inside the same domain) and should have computed `DSRK`, `DS-rRK` and `DS-rIK` before. Therefore, the response it sends, EAP-Initiate/Re-auth encapsulated inside IGMP-AC `aresponse`, must be integrity protected by the integrity key, `DS-rIK`. It should be noted that `DS-rIK` would be used to protect the integrity of the EAP-Initiate/Re-auth message only, and not the whole `aresponse` message.

On receiving the IGMP-AC `aresponse` message, the MR decapsulates the EAP-Initiate/Re-auth message that contains the keyName-NAI to identify the ER server's domain. The MR extracts the realm in the keyName-NAI [1] field to send the message (i.e., the EAP-Initiate/Re-auth message) to the appropriate ER server using one of the AAA protocols (e.g., RADIUS [34]), which is generally colocated with a Local AAAS (LAAAS) of the domain.

It is expected that the LAAAS has previously authenticated and authorized the MN/peer before. Hence, it should have received `DSRK` and authorization information of the MN from the HAAAS (i.e., Home EAP Server), and computed `DS-rRK` and `DS-rIK` before. Therefore, it is able to authenticate the MN's possession of the `DS-rRK` by checking the validity of EAP-Initiate/Re-auth using its own integrity key, `DS-rIK`. Next, the LAAAS sends `rMSK`, EAP-Finish/Re-auth and Authorization information of the MN/EU using AAA protocol to the MR. The EAP-Finish/Re-auth fields of the message are also integrity protected using `DS-rIK`.

On receiving the message, the MR will extract the `rMSK`, and keep it for future use. It should be noted that the AAA protocol used (e.g., RADIUS) is expected to provide secured communications. Otherwise, the secrecy of the `rMSK` would be compromised. The receiver access control architecture we have presented in this chapter does not use the `rMSK`. However, it could be used to provide secured communication between the MN and the MR on completion of the secured handoff. Next, the MR will extract the

EAP-Finish/Re-auth fields, and send them to the MN inside an IGMP-AC `aresult` message. Additionally, it will send a multicast routing protocol join towards the multicast DDT to extend the DDT up to the MR. From this moment, the MR also starts the accounting for the network resource usage of the MN.

The MN will extract the EAP-Finish/Re-auth message, and will check the integrity validity using its integrity key, `DS-rIK`. Hence, it authenticates the LAAAS' possession of the `DS-rRK`. Thus, both the MN and the LAAAS mutually authenticate the possession of the root Re-authentication Key, `DS-rRK`.

Once the DDT has been extended up to the MR, multicast data will be received by the MR, and will be forwarded to the MN. When the first multicast data packet is received by the MR, it will start the accounting for the multicast data usage of the MN.

The EU will continue its group activity (i.e., receiving multicast data) and will leave the group after a while. The leave operation may happen in different ways. The MN may move to another authenticator of the same domain or move to another domain without informing the MR. If the MN leaves silently, the MR will discover it eventually because it receives no IGMP-AC report (i.e., IGMPv3 report) from the MN. On the other hand, if the MN explicitly leaves the group by sending an IGMP-AC report, which contains a leave message, the MR will open an ERP session to authenticate the MN. This will protect from the attacks generated by a forged receiver, who sends a leave message impersonating the identity of a valid receiver. When the MR will confirm about the leave of the MN (EU), it will send both types of accounting information to the LAAAS. The LAAAS updates the accounting information of the network usage and forwards the multicast data usage information to the HAAAS to update the participants' database (see Figure 48).

Chapter 10

Comparisons with the State of the Art

Our research project is composed of different subproblems: receiver access control, sender access control, developing an access control policy framework, data distribution control, and secured handoff and access control for mobile receivers. In the previous chapters, we have carried out literature surveys for each subproblem (except data distribution control), and summarized different limitations of solutions representing the state of the art. However, we have not compared the access control architecture we have developed with the previous methods that exist in the literature. In this chapter, different aspects of our access control architecture are compared with the state of the art using the same comparative criteria that we have previously used. None of the previous methods targeted to solve the whole domain of the participant access control, rather they addressed only a subproblem (e.g., receiver or sender access control). Therefore, when we compare our architecture with the previous work, we should concentrate in a specific aspect of the architecture. Hence, in the following, we have compared individually receiver access control, sender access control, access control policy framework and mobile receiver access control with the previous work.

10.1 Receiver Access Control

We have designed IGMP-AC, an extended version of the IGMPv3 protocol, to provide receiver access control in multicasting. A summary of the previous work in receiver access control has been presented in Table 2. We have pointed out the limitations of the previous work, and set the requirements of IGMP-AC to overcome these limitations. IGMP-AC adds three new messages to IGMPv3 and provides a client-server communication. Hence, any authentication protocol (e.g., EAP [2]) can be encapsulated over IGMP-AC. We have also presented the operability of IGMP-AC for multi-domain distributed groups with the help of Diameter [18] agents. In Table 7, we have compared IGMP-AC with the previous receiver access control methods.

Table 7: Comparing IGMP-AC with the Previous Work

Criteria	**IGMP-AC**	**Previous Work**
IGMP Version	Based on IGMPv3 and supports "source filtering" property as well.	Not all of them are based on IGMPv3.
Authentication	Supports all sorts of authentication by encapsulating EAP packets.	Only EUIA system [105] provides different authentication methods, although it does not support EAP encapsulation.
Authorization	Provides receiver authorization by deploying AAA framework.	EUIA and IGAP [43] provide authorization.
Accounting	Provides real time accounting by deploying AAA framework.	EUIA and IGAP provide accounting.
Vulnerability to attacks	Depends on the EAP method. IGMP-AC will prevent the known attacks if the EAP method is secured.	Most of the methods (except EUIA and IGMP extension [45]) suffer from different known attacks.

10.2 Sender Access Control

In our sender access control architecture, we have deployed PANA [33], a link-layer agnostic protocol, which works as the EAP lower layer and encapsulates the EAP packets. A PANA session is initiated between the sender and the one-hop AR, when a potential sender shows her interest to send data to a secured group. At the end of a PANA session, through an EAP method, the sender and the AAAS mutually authenticate and establish a set of keys. Using a key hierarchy, a secret key (`IKE-Pre-shared Key`) is established between the sender and the AR. Finally, the sender and the AR

Table 8: Comparing Sender Access Control with the Previous Work

Criteria	**Our Method**	**Previous Work**
AAA functions	Provides all AAA functionalities by deploying AAA framework.	None of the methods provides accounting. The SACL method [111] does not provide authentication.
Authentication mechanism	Supports all sorts of authentication by deploying PANA to encapsulate EAP packets.	All of the methods are confined to a specific authentication method.
Vulnerability to attack	Prevents source address spoofing and anti-replay attack. Minimizes DoS attack.	Suffer from commonly known attacks, specially DoS attack.
Overhead	Low—if we consider sender access control only.	Low—only for challenge-response [55] method, which fails to provide secured authentication.
Routing protocol	Independent of the underlying routing protocol.	Dependent on the routing protocol, and mostly based on the CBT [13] protocol.
Intra or Inter domain	Deployable for both intra and inter domain groups.	Deployable for both intra and inter domain groups except the challenge-response method.

complete an IKEv2 session and establish an IPsec SA. From now on, the sender sends any multicast data using that SA tunnel. The SA provides anti-replay, prevents source address spoofing attack and minimizes DoS attack. We have also extended the sender access control architecture for multi-domain distributed groups. We have summarized the previous work on sender access control in Table 4. We have compared our sender access control architecture using the same criteria that we have previously used. In Table 8, the comparison between our architecture and the previous work has been presented.

10.3 Policy Framework

For successful operation of our access control architecture a scalable and distributed policy framework is required, which will facilitate the Group Owner (GO) to specify access control policy. The policy will be enforced at the ARs, therefore, the ARs will be the Policy Enforcement Point (PEP). The GCKS (an entity, explained in the MSEC framework) will perform the tasks of the Policy Decision Point (PDP). We have recommended the use of eXtensible Access Control Markup Language (XACML) [35] for policy specification, and Security Assertion Markup Language (SAML) [22] for policy transportation in our framework. However, the proposed framework is not dependent on any specific policy specification language or transportation protocol.

We have carried out a survey on multicast access control policy, and presented a summary of the previous work in Table 5. We have evaluated the proposed framework using the same features that we previously used. The comparison is shown in Table 9.

Table 9: Comparing Policy Framework with the Previous Work

Criteria	Our framework	Previous Work
Access control policy	Primarily designed for access control policy.	Only Antigone [82] supports access control policy.
Data control policy	Can be extended to support such policy enforcement.	All of the methods support such policy enforcement.
Specification language	Recommends the use of XACML [35], however, any other language can be used.	Dependent on specific language. Antigone does not specify/recommend any language.
Policy protocol	Recommends the use of SAML [22], however, any other protocol can be used.	Only DDCM [27] specifies CCNP protocol. Other methods do not specify/recommend any protocol.
Follows the IETF FW	Yes. PEP, PDP and policy repository are present.	None of the methods follows IETF FW.
Fits with the MSEC FW	Yes. GCKS works as PDP.	XML based method [87] and GSAKMP [41] fit with the MSEC FW.

10.4 Receiver Access Control and Secured Handoff for Mobile Receivers

In the absence of receiver access control, IP multicast will face new challenges if it has to control mobile EUs accessing valuable data in wireless networks. We have broadened the scope of IGMP-AC by demonstrating the usability of IGMP-AC in wireless networks for mobile receiver (or EU) access control. In addition, using EAP Re-authentication Protocol (ERP) [90], we have developed a procedure for secured and fast handoff of mobile EUs in wireless networks.

In section 9.1, we have briefly discussed the previous work in receiver access control and handoff for mobile receivers in wireless networks. We have pointed out that the

Table 10: Comparing Receiver Access Control and Secured Handoff with the Previous Work

Criteria	**Our Architecture**	**Previous Work**
Receiver access control	Supports through IGMP-AC.	None of the methods supports it.
Secured handoff	By encapsulating EAP and ERP packets inside IGMP-AC.	None of the methods addresses it.
Handoff latency	Reduced significantly and requires a single round-trip between the MN and the local ER server.	Reduced by advanced joining to the visiting domain [86] and sending unsolicited join without IGMP query [73, 6].
Optimizing IGMP communication	This problem has not been addressed, however, the existing techniques can be used.	By aggregating many IGMP messages [67], deploying proxy for MNs [73], allowing MN to go into sleep mode [97] and tuning IGMP query timer [77, 6].

researchers have concentrated on two issues: reducing handoff latency and optimizing the communication between the Mobile Node (MN) and the IGMP router. However, receiver access control and secured handoff are not properly addressed. In Table 10, we have compared our architecture with the previous work.

Chapter 11

Conclusion and Future Work

IP multicast was developed to provide an efficient and scalable one-to-many and many-to-many group communications. Two major features that made the classical "any one can send any one can receive" multicast service model scalable and simple are:

1. The receivers will remain anonymous throughout the group communication.
2. A sender is not necessarily required to join a group to send data to that group.

Due to these two constraints no security service was deployed with multicast, and without any security features the Internet Service Providers were reluctant to support multicast based applications. However, in the last few years, the revolutionary increase in the available bandwidth has made the bandwidth intensive applications, such as Internet TV, online courses, multi-player games and video-conferencing very popular. All of these applications have some common features: need a very high speed Internet connection, realtime delivery of data, have a potential to generate revenue stream from the EUs. Most importantly, all of them will benefit and be well-served by the deployment of IP multicast. As a consequence, a new wave of needs has been

generated among the Internet research community for a secured multicast service model. Hence, the IETF created the Multicast Security (MSEC) [89] Working Group to develop a secured way of transmitting data using IP multicast. MSEC has ended up with an excellent base, the Reference Framework [40], which provides a platform to develop a group key management protocol to protect multicast data. Moreover, individual Working Groups are developing solutions to protect the multicast routing protocols (e.g., PIM link-local messages' security is addressed in [8]). All of these efforts have been accomplished by minimally disrupting the scalability and simplicity of the classical multicast service model. However, in the course of time, the direction has been proven wrong. If we want to secure a communication, we have to secure every single message of the underlying protocol, and to authenticate and authorize every participant of the communication. In addition to this, a revenue generating application must relate e-commerce communication with admission control and policy enforcement.

11.1 Contributions

We have devoted our research in developing a generalized framework that will operate on top of the existing multicast service model. Furthermore, our solution should be practicable in parallel to the MSEC Reference Framework. Hence, any group key management protocol could be implemented to achieve secure data transmission. Another major aspect of our solution is the maximum reuse of the existing standard protocols and architecture. This will reduce the task of the service providers. The contributions of the thesis have been summarized in the following:

Defining the problem

The problem has been clearly defined by identifying the security vulnerabilities due to the fake IGMP report messages and the forged sender.

1. Three types of IGMP report messages and the attacks that might be generated by forging these messages have been illustrated.
2. Three severe attacks—replay, sender address spoofing and Denial of Service (DoS)—could emerge due to an attacker playing the role of a sender.

Moreover, it has been reasonably established that none of these attacks could be resisted by deploying group key management and sending data in encrypted format.

Developing a participant access control architecture

A comprehensive architecture for participant (i.e., both sender and receivers) access control has been designed. This is the first architecture (to our knowledge) that consider participant access control, e-commerce interactions and policy enforcement in a generalized framework. Different entities and their roles, which are involved in this architecture, have been identified.

Receiver access control using IGMP-AC

The Internet Group Management Protocol with Access Control (IGMP-AC), an extended version of IGMPv3, has been designed to accomplish receiver access control. IGMP-AC is unique to all existing extensions of IGMP by encapsulating EAP packets. By doing so, it supports all sorts of authentication. Moreover, it adds the minimum overhead and reuses the IETF standardized protocols. It is independent of the underlying routing protocol and deployable for inter-domain groups.

1. A survey on present solutions has been accomplished. The limitations of these solutions have been identified. A set of requirements has been defined by considering the limitations of the previous methods that an extended version of IGMP should provide.
2. The IGMP-AC protocol has been presented with detailed state diagrams and textual descriptions. The list of additional massages and reception states have also been documented.

3. The verification model of IGMP-AC has been modeled with the formal language, PROMELA. This model has been verified using the tool, SPIN, that has reported the model free from any error. SPIN is a powerful tool, used frequently in industries and academia for formal verification of network protocols and distributed programs.

4. EAP encapsulation procedure over IGMP-AC has been explained elaborately. Exact locations of different protocols (i.e., EAP method, EAP peer, IGMP-AC, AAA, etc.) in the protocol stack have been identified. Moreover, the serviceability of the IGMP-AC as an EAP lower-layer has been justified by comparing the IGMPv3 properties with the EAP lower-layer requirements identified in [2].

5. EAP encapsulation over IGMP-AC has been further demonstrated with an example EAP method, EAP-IKEv2. Any other EAP methods can be deployed following this example.

6. The security properties of EAP-IKEv2 in peer-to-peer mode have been validated using an AVISPA model. A MitM attack has been found, which has been fixed by adding an optional message.

7. The peer-to-peer model (that we have fixed from the MitM attack) has been extended to the pass-through model by introducing an authenticator (NAS) between the peer (host) and the server (AAAS). AVISPA reported the model free from any attack, which establishes the security claims of the EAP-IKEv2 method in the pass-through mode.

Sender access control

Sender access control has been achieved by authenticating and authorizing the sender, and by establishing an IPsec Security Association (SA) between the sender and the one-hop AR. This is an innovative architecture that will allow adding AAA functionalities at the sender end. Given that PANA has been developed at the IETF to function as an EAP-lower layer, PANA has been

deployed between the sender and the one-hop AR.

1. A survey on existing methods for sender access control, and a brief comparison among these methods have been presented.
2. The sender access control architecture has been illustrated with the underlying threat model and the roles of different entities.
3. Two alternate procedures to initiate the sender authentication has been explained. Moreover, how PANA framework would be deployed within the proposed architecture is explained. On successful PANA authentication, the shared-key generation and IPsec SA establishment procedures have been described.
4. The benefits of the architecture have been listed to further reinforce our research finding.

Developing a policy framework

A policy framework for specifying and enforcing access control policy for multicast group participants has been designed. In the absence of a policy framework, we could not expect flexible authorization and accounting functionalities.

1. A survey on access control policy for multicast groups has been carried out.
2. Access control policies for some of the commonly used multicast based applications have been identified.
3. A set of requirements has been listed that the proposed policy framework should satisfy. The policy framework has been presented in detail.
4. An XACML example policy for controlling access to an hypothetical online course has been developed.

Inter-domain access control

A generalized inter-domain architecture has been developed by considering the

distribution of the sender, the receivers and the GO in different domains. The receiver and the sender access controls have been addressed independently. This is the first inter-domain access control architecture for multicast group participants and will bring a breakthrough in the future research of IP multicast.

1. The inter-domain receiver access control architecture, which deploys IGMP-AC and Diameter agents, has been presented.
2. The inter-domain sender access control architecture deploys PANA and Diameter agents, and on successful authentication sets up an IPsec SA between the sender and the one-hop AR.

Protecting the inter-domain distribution tree

In the absence of the protection of the Data Distribution Tree (DDT), an adversary may tamper with multicast data enroute. This type of attack will be severe, specially, for an inter-domain DDT. A novel access control mechanism has been developed for the inter-domain DDT by deploying Multicast Security Association (MSA).

1. The PIM (S, G) Join message has been extended to carry the identity of a router. The forwarding rules of this message has been modified to trigger a router authentication.
2. Two alternate solutions of deploying MSA—centralized and distributed—have been developed. For these solutions, router authentication and MSA operations have been clearly presented.
3. These two methods have been compared with respect to a number of features: deployment and maintenance costs, packet delivery time, security, etc. Mathematical expressions have been developed to compare the establishment costs.

Mobile receiver access control and secured handoff

IP multicast will face new challenges if it has to control mobile EUs accessing valuable data in wireless networks. In the absence of receiver access control the operators of the wireless networks will be reluctant to deploy IP multicast due to possible wastage of valuable bandwidth and other resources from joining of any unauthorized mobile EU and unwanted extension of the distribution tree.

1. The scope of the IGMP-AC has been broadened by demonstrating the usability of IGMP-AC in wireless networks for mobile receiver (or EU) access control.
2. Using the EAP Re-authentication Protocol (ERP), a secured and fast handoff procedure for mobile EUs in wireless networks has been developed.
3. The novelty of the mobile receiver access control architecture lies in the coupling of secured handoff and receiver access control of mobile multicast EUs.

11.2 The Impacts of Our Research

It is our strong belief that the impacts of our research findings will be far beyond academia. We have accomplished this research having some specific goals in our mind to be achieved in the future. For example, we have intentionally concentrated in developing a generalized framework, reused the standardized protocols and followed the IETF frameworks and architectures.

The need of admission control or access control for IP multicast is clearly defined by the Internet standard bodies, such as IETF and ITU-T. After all these years, the MBONED [81] Working Group at the IETF has initiated the development of a AAA enabled multicast framework [44]. The IPTV FG at the ITU-T has developed an IPTV multicast framework [101]. It should be noted that access control is one of the

key components that needs to be solved in the IPTV multicast framework.

This thesis will be one step forward for IP multicast deployment. We have successfully projected the problems to be addressed in addition to the framework that the MBONED Working Group and IPTV FG are working with. The receiver access control and secured handoff mechanism that we have developed in wireless networks will motivate the researchers to study the applicability of multicast applications in different wireless networks. In addition, the access control architecture we have developed will facilitate the future researchers in the area of e-commerce communication with an extensible framework.

11.3 Future Work

Our future goals can be divided into short-term and long-term goals. In the near future, we would like to complete the developments of the protocols that we have deployed in our architecture by defining the packet formats, values of different timers, etc. Next, we would write Internet Drafts to bring our research findings to the appropriate Working Groups of the IETF. In this regard, we would like to add that we have already presented our intermediate research results at the MBONED Working Group during the 69th IETF Meeting [10]. We have received a couple of positive responses from persons working in the related industries. Therefore, we are expecting to write Internet Drafts to present our access control architecture at the IETF jointly with one of the industries.

Our long-term goals includes extending inter-domain DDT control for ASM groups and multicast data confidentiality, studying source mobility in wireless networks.

The DDT control mechanism we have developed is deployable for SSM groups only and dependent on the PIM-SSM routing protocol. To make it applicable for ASM

groups also, we should consider the communications among the Rendezvous Points (RP) and the switchover of the DDT from the RP-Tree (RPT) to the Shortest-Path Tree (SPT).

We have achieved multicast data integrity by deploying Multicast Security Association (MSA) for inter-domain groups. However, for secrecy or data confidentiality, our architecture is still dependent on the underlying Group Key Management (GKM) protocol. Our distributed MSA method can be used as the communication backbone in designing a hierarchical GKM protocol.

Finally, we plan to extend our wireless architecture considering source mobility as well. In our sender access control, by using PANA we have already decoupled the sender authentication and establishing access control at the AR. This will provide a privilege while addressing sender access control for a mobile source. The sender will complete a EAP session (using PANA) when it starts sending data for the first time. Next, depending on its point of attachment (which will be the Enforcement Point (EP) according to the PANA architecture), a new set of `PaC-EP-Master-Key` and `IKE Pre-shared-Key` will be calculated and transported to the new EP.

Bibliography

[1] B. Aboba and M. Beadles, "The Network Access Identifier," RFC 2486, Jan. 1999.

[2] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson and H. Levkowetz, "Extensible Authentication Protocol (EAP)," RFC 3748, Jun. 2004.

[3] B. Aboba, D. Simon, P. Eronen and H. Levkowetz. (2007) "Extensible Authentication Protocol (EAP) Key Management Framework". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-eap-keying-22.txt

[4] A. Anderson and H. Lockhart, "SAML 2.0 profile of XACML v2.0," *OASIS Standard*, Feb. 2005.

[5] J. Arkko and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," RFC 4187, Jan. 2006.

[6] H. Asaeda. (2008) "IGMP and MLD Extensions for Mobile Hosts and Routers". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-asaeda-multimob-igmp-mld-mobility-extensions-00.txt

[7] N. Asokan, V. Niemi and K. Nyberg. (2002) Man-in-the-Middle in Tunneled Authentication Protocols. IACR ePrint Archive Report. [Online]. Available: http://eprint.iacr.org/2002/163

[8] J.W. Atwood, S. Islam and M. Siami. (2008) "Authentication and Confidentiality in PIM-SM Link-local Messages". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-pim-sm-linklocal-03.txt

[9] J.W. Atwood, "An Architecture for Secure Multicast," in *Proc. of 32nd Annual Conference on Local Computer Networks*, Dublin, Ireland, Oct. 2007, pp. 73–78.

[10] J.W. Atwood and S. Islam. (2007) "End User Identification". Proceedings of the Sixty-ninth Internet Engineering Task Force. [Online]. Available: http://www.ietf.org/proceedings/07jul/slides/mboned-11.pdf

[11] Automated Validation of Internet Security Protocols and Applications (AVISPA). [Online]. Available: http://www.avispa-project.org/

[12] T. Ballardie and J. Crowcroft, "Multicast-specific Security Threats and Counter-measures," in *Proc. of IEEE Symposium on Network and Distributed System Security*, San Diego, California, USA, Feb. 1995, pp. 2–16.

[13] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture," RFC 2201, Sep. 1997.

[14] T. Bates, R. Chandra, D. Katz and Y. Rekhter, "Multiprotocol Extensions for BGP-4," RFC 4760, Jan. 2007.

[15] F. Bersani and H. Tschofenig, "The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method," RFC 4764, Jan. 2007.

[16] A. Boers, IJ. Wijnands and E. Rosen. (2008) "The PIM Join Attribute Format". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-pim-join-attributes-04.txt

[17] B. Cain, S. Deering, I. Kouvelas, B. Fenner and A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376, Oct. 2002.

[18] P. Calhoun, J. Loughney, E. Guttman, G. Zorn and J. Arkko, "Diameter Base Protocol," RFC 3588, Sep. 2003.

[19] P. Calhoun, G. Zorn, D. Spence and D. Mitton, "Diameter Network Access Server Application," RFC 4005, Aug. 2005.

[20] P. Calhoun, T. Johansson, C. Perkins, T. Hiller and P. McCann, "Diameter Mobile IPv4 Application," RFC 4004, Aug. 2005.

[21] P.R. Calhoun, S. Farrell and W. Bulley. (2002) "Diameter CMS Security Application". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-aaa-diameter-cms-sec-04.txt

[22] S. Cantor, J. Kemp, R. Philpott and E. Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," *OASIS Standard*, Mar. 2005.

[23] B. Coan, H. Nortel and B. Weis. (2002) IGMP Security Problem Statement and Requirements. Working Group Meeting of Group Security (GSEC), Co-located with IETF 53. [Online]. Available: http://www.securemulticast.org/GSEC/gsec3_ietf53_SecureIGMP1.pdf

[24] A. Colegrove and H. Harney, "Group Security Policy Token v1," RFC 4534, Jun. 2006.

[25] S. Deering, "Host Extensions for IP Multicasting," RFC 1112, Aug. 1989.

[26] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," RFC 4346, Apr. 2006.

[27] P.T. Dinsmore, D.M. Balenson, M. Heyman, P.S. Kruus, C.D. Scace and A.T. Sherman, "Policy-Based Security Management for Large Dynamic Groups:

A Overview of the DCCM Project," in *Proc. of DARPA Information Survivability Conference and Exposition*, Hilton Head, South Carolina, USA, Jan. 2000, pp. 64–73.

[28] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan and A. Sastry, "COPS (Common Open Policy Service) Protocol," RFC 2748, Jan. 2000.

[29] P. Eronen, T. Hiller and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application," RFC 4072, Aug. 2005.

[30] W. Fenner, "Internet Group Management Protocol, Version 2," RFC 2236, Nov. 1997.

[31] B. Fenner and D. Meyer, "Multicast Source Discovery Protocol (MSDP)," RFC 3618, Oct. 2003.

[32] B. Fenner, M. Handley, H. Holbrook and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)," RFC 4601, Aug. 2006.

[33] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)," RFC 5191, May. 2008.

[34] K. Gaonkar, L. Dondeti, V. Narayanan, and G. Zorn. (2008) "RADIUS Support for EAP Re-authentication Protocol". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-gaonkar-radext-erp-attrs-03.txt

[35] S. Godik and T. Moses, "OASIS eXtensible Access Control Markup Language (XACML) Version 2.0," *OASIS Standard*, Feb. 2005.

[36] A.F. Gomez-Skarmeta, A.L. Mateo Martinez and P.M. Ruiz Martinez, "IGMPv3-based Method for Avoiding DoS Attacks in Multicast-enabled Networks," in *Proc. of 25th IEEE Conference on Local Computer Networks*, Tampa, FL, USA, Nov. 2000, pp. 94–95.

[37] M. Handley, C. Perkins and E. Whelan, "Session Announcement Protocol," RFC 2974, Oct. 2000.

[38] Handover Keying (hokey) Working Group, IETF. [Online]. Available: http://www.ietf.org/html.charters/hokey-charter.html

[39] T. Hardjono and B. Cain, "Key Establishment for IGMP Authentication in IP Multicast," in *Proc. of IEEE European Conference on Universal Multiservice Networks*, CERF, Colmar, France, Sep. 2000, pp. 247–252.

[40] T. Hardjono, and B. Weis, "The Multicast Group Security Architecture," RFC 3740, Mar. 2004.

[41] H. Harney, U. Meth, A. Colegrove and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol," RFC 4535, Jun. 2006.

[42] H. Haverinen and J. Salowey, "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)," RFC 4186, Jan. 2006.

[43] T. Hayashi, D. Andou, H. He, W. Tawbi and T. Niki. (2004) "Internet Group membership Authentication Protocol (IGAP)". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-hayashi-igap-03

[44] T. Hayashi, H. He, H. Satou, H. Ohta and S. Vaidya. (2008) "Requirements for Multicast AAA coordinated between Content Provider(s) and Network Service Provider(s)". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-mboned-maccnt-req-06.txt

[45] H. He, B. Cain and T. Hardjono. (2001) "Upload Authentication Information Using IGMPv3". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-he-magma-igmpv3-auth-00.txt

[46] B. Hilt and J. Pansiot, "Using IGMPv3 to Manage Multicast Access," in *Proc. of 4th Conference on Security and Network Architectures*, Batz sur Mer, France, Jun. 2005.

[47] H. Holbrook and B. Cain, "Source-Specific Multicast for IP," RFC 4607, Aug. 2006.

[48] G.J. Holzmann, *The SPIN Model Checker.* Addison-Wesley Professional, 2004.

[49] G.J. Holzmann, "The Model Checker SPIN," *IEEE Transactions on Software Engineering*, vol. 23, no. 5, pp. 279–295, 1997.

[50] R. Housley, W. Polk, W. Ford and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 3280, Apr. 2002.

[51] R. Housley and B. Aboba, "Guidance for Authentication, Authorization, and Accounting (AAA) Key Management", RFC 4962, Jul. 2007.

[52] IEEE Standards for Local and Metropolitan Area Networks, "Port based Network Access Control," IEEE Std 802.1X-2001, Jun. 2001.

[53] Internet Assigned Numbers Authority (IANA). [Online]. Available: http://www.iana.org/

[54] IANA, "Address Family Numbers". [Online]. Available: http://www.iana.org/assignments/address-family-numbers

[55] N. Ishikawa, N. Yamanouchi and O. Takahashi, "An Architecture for User Authentication of IP Multicast and Its Implementation," in *Proc. of Internet Workshop*, Japan, Feb. 1999, pp. 81–87.

[56] S. Islam and J.W. Atwood, "A Framework to Add AAA Functionalities in IP Multicast," in *Proc. of the Advanced International Conference on Telecommunications*, Guadeloupe, French Caribbean, Feb. 2006.

[57] S. Islam and J.W. Atwood, "The Internet Group Management Protocol with Access Control (IGMP-AC)," in *Proc. of 31st IEEE Conference on Local Computer Networks*, Tampa, Florida, USA, Nov. 2006, pp. 475–482.

[58] S. Islam and J.W. Atwood, "A Policy Framework for Multicast Group Control," in *Proc. of IEEE Consumer Communications and Networking Conference—Workshop on Peer-to-Peer Multicasting*, Las Vegas, NV, USA, Jan. 2007, pp. 1103–1107.

[59] S. Islam and J.W. Atwood, "Sender Access Control in IP Multicast," in *Proc. of 32nd IEEE Conference on Local Computer Networks*, Dublin, Ireland, Oct. 2007, pp. 79–86.

[60] S. Islam and J.W. Atwood, "Multicast Receiver Access Control by IGMP-AC," *Submitted to Computer Networks.*

[61] S. Islam and J.W. Atwood, "Receiver Access Control and Secured Handoff in Mobile Multicast using IGMP-AC," *Accepted for publication in 33rd IEEE Conference on Local Computer Networks, 8 pages.*

[62] P. Jayaraman, R. Lopez, Y. Ohba, M. Parthasarathy and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) Framework," RFC 5193, May. 2008.

[63] P. Judge and M. Ammar, "Gothic: A Group Access Control Architecture for Secure Multicast and Anycast," in *Proc. of the 21st IEEE INFOCOM*, New York, NY, USA, Jun. 2002, pp. 1547–1556.

[64] P. Judge and M. Ammar, "Security Issues and Solutions in Multicast Content Distribution: A Survey," *IEEE Network*, vol. 17, no. 1, pp. 30–36, 2003.

[65] S. Jun, C. Cho and N. Park, "IGMP Proxy for Multicast Services in Wireless Mobile Networks," in *Proc. of 61st Vehicular Technology Conference*, Stockholm, Sweden, May 2005, pp. 2855–2858.

[66] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," RFC 4306, Dec. 2005.

[67] S. Kaur, B. Madan and S. Ganesan, "Multicast support for mobile IP using a modified IGMP," in *Proc. of Wireless Communications and Networking Conference*, New Orleans, LA, USA, Sep. 1999, pp. 948–952.

[68] M. Kellil, I. Romdhani, H. Lach, A. Bouabdallah and H. Betttahar, "Multicast Receiver and Sender Access Control and Its Applicability to Mobile IP Environments: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 2, pp. 46–70, 2005.

[69] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301, Dec. 2005.

[70] S. Kent and R. Atkinson, "IP Authentication Header," RFC 4302, Dec. 2005.

[71] S. Kent, "IP Encapsulating Security Payload (ESP)," RFC 4303, Dec. 2005.

[72] S. Kent, C. Lynn and K. Seo, "Secure Border Gateway Protocol (Secure-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, 2000.

[73] B. Kim and K. Han, "Multicast Handoff Agent Mechanism for All-IP Mobile Network," *Mobile Networks and Applications*, vol. 9, no. 3, pp. 185–191, 2004.

[74] J. Lai and W. Liao, "Mobile Multicast with Routing Optimization for Recipient Mobility," *IEEE Transactions on Consumer Electronics*, vol. 47, no. 1, pp. 199–206, 2001.

[75] M. Lepinski and S. Kent. (2008) "An Infrastructure to Support Secure Internet Routing". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-sidr-arch-03.txt

[76] L. Lin, X. Li and Y. Cheng, "HKM: A Hybrid Key Management Scheme for Secure Mobile Multicast," in *Proc. of Networking, Architecture, and Storage*, Guilin, China, Jul. 2007, pp. 109–114.

[77] H. Liu and H. Asaeda. (2007) "Mobile Multicast Requirements on IGMP/MLD Protocols". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-liu-multimob-igmp-mld-mobility-req-00.txt

[78] H. Liu, W. Cao and H. Asaeda. (2008) "Lightweight IGMPv3 and MLDv2 Protocols". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-mboned-lightweight-igmpv3-mldv2-03.txt

[79] R.M. Lopez, A.G. Skarmeta, J. Bournelle, M. Laurent-Maknavicus and J.M. Combes, "Improved EAP Keying Framework for a Secure Mobility Access Service," in *Proc. of International Conference On Communications And Mobile Computing*, Vancouver, British Columbia, Canada, Jul. 2006, pp. 183–188.

[80] M. Lorch, S. Proctor, R. Lepro, D. Kafura and S. Shah, "First Experiences Using XACML for Access Control in Distributed Systems," in *Proc. of ACM Workshop on XML Security*, Fairfax, VA, USA, Oct. 2003, pp. 25–37.

[81] MBONE Deployment (mboned) Working Group, IETF. [Online]. Available: http://www.ietf.org/html.charters/mboned-charter.html

[82] P. McDaniel, A. Atul and P. Honeyman, "Antigone: A Flexible Framework for Secure Group Communication," in *Proc. of the 8th USENIX Security Symposium*, Washington D.C., USA, Aug. 1999, pp. 99–114.

[83] C. Meadows, "Analysis of the Internet Key Exchange Protocol Using the NRL Protocol Analyzer," in *Proc. of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 1999, pp. 216–231.

[84] A. Meissner, S.B. Musunoori and L. Wolf, "MGMS/GML - Towards a new Policy Specification Framework for Multicast Group Integrity," in *Proc. of International Symposium on Applications and the Internet*, Tokyo, Japan, Jan. 2004, pp. 233–239.

[85] C. Metz, "AAA Protocols: Authentication, Authorization, and Accounting for the Internet," *IEEE Internet Computing*, vol. 3, no. 6, pp. 75–79, 1999.

[86] Y. Moritani and Y. Atsumi, "Seamless Hand-off Method for Multicast Receivers Based on Wireless Link Connection Intensity," in *Proc. of Wireless Communications and Networking Conference*, New Orleans, LA, USA, Mar. 2003, pp. 1236–1241.

[87] R. Mukherjee and J.W. Atwood, "XML Policy Representation for Secure Multicast," in *Proc. of IEEE SoutheastCon 2005 Conference*, Fort Lauderdale, FL, USA, Apr. 2005, pp. 580–587.

[88] R. Mukherjee and J.W. Atwood, "Scalable Solutions for Secure Group Communications," *Computer Networks*, vol. 51, no. 12, pp. 3525–3548, 2007.

[89] Multicast Security (msec) Working Group, IETF. [Online]. Available: http://www.ietf.org/html.charters/msec-charter.html

[90] V. Narayanan and L. Dondeti. (2008) "EAP Extensions for EAP Reauthentication Protocol (ERP)". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-hokey-erx-14.txt

[91] M. Nystroem, "The EAP Protected One-Time Password Protocol (EAP-POTP)," RFC 4793, Feb. 2007.

[92] M. Parthasarathy. (2005) "PANA Enabling IPsec based Access Control". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-pana-ipsec-07.txt

[93] Protocol Independent Multicast (pim) Working Group, IETF. [Online]. Available: http://www.ietf.org/html.charters/pim-charter.html

[94] B. Quinn and K. Almeroth, "IP Multicast Applications: Challenges and Solutions," RFC 3170, Sep. 2001.

[95] Y. Rekhter, T. Li and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Jan. 2006.

[96] C. Rigney, S. Willens, A. Rubens and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865, Jun. 2000.

[97] I. Romdhani, J. Munoz, H. Bettahar, and A. Bouabdallah, "Adaptive Multicast Membership Management for Mobile Multicast Receivers," in *Proc. of 2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Montreal, Canada, Jun. 2006, pp. 189–195.

[98] J. Salowey, L. Dondeti, V. Narayanan and M. Nakhjiri. (2008) "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-hokey-emsk-hierarchy-07.txt

[99] H. Satou, H. Ohta, C. Jacquenet, T. Hayashi and H. He. (2007) "AAA Framework for Multicasting". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-mboned-multiaaa-framework-06.txt

[100] Security Protocol ANimator (SPAN). [Online]. Available: http://www.irisa.fr/lande/genet/span/

[101] Y. Seo and P. Juyoung. (2007) "IPTV Multicast Frameworks". ITU-T IPTV Focus Group, FG IPTV-DOC-0190. [Online]. Available: http://www.itu.int/md/T05-FG.IPTV-DOC-0190/en

[102] C. Shields and J.J. Garcia-Luna-Aceves, "KHIP-A Scalable Protocol for Secure Multicast Routing," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 53–64, 1999.

[103] W. Simpson, "The Point-to-Point Protocol (PPP)," RFC 1661, Jul. 1994.

[104] Specification of the Problems in the High-Level Specification Language. AVISPA Project Deliverable D6.2. [Online]. Available: http://www.avispa-project.org/delivs/6.2/d6-2.pdf

[105] N. Sultana and J.W. Atwood, "Secure Multicast Communication: End User Identification and Accounting," in *Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering*, Saskatoon, SK, Canada, May 2005, pp. 1674–1677.

[106] Sun Microsystems Laboratories. Sun's XACML Implementation. [Online]. Available: http://sunxacml.sourceforge.net

[107] H. Tschofenig, D. Kroeselberg, A. Pashalidis, Y. Ohba and F. Bersani, "The Extensible Authentication Protocol-Internet Key Exchange Protocol v2 (EAP-IKEv2) Method," RFC 5106, Feb. 2008.

[108] University of Murcia XACML Editor. [Online]. Available: http://sourceforge.net/projects/umu-xacmleditor/

[109] A.K. Venkataiahgari, J.W. Atwood and M. Debbabi, "Secure E-Commerce Transactions for Multicast Services," in *Proc. of 8th IEEE Conference on E-Commerce Technology*, San Francisco, CA, USA, Jun. 2006, pp. 132–139.

[110] R. Vida and L. Costa,, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6," RFC 3810, Jun. 2004.

[111] N. Wang and G. Pavlou, "Scalable Sender Access Control for Bi-directional Multicast Routing," *Computer Networks*, vol. 43, no. 5, pp. 539–555, 2003.

[112] B. Weis, G. Gross and D. Ignjatic. (2008) "Multicast Extensions to the Security Architecture for the Internet Protocol". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-ietf-msec-ipsec-extensions-09.txt

[113] B. Weis, T. Hardjono and H. Harney, "The Group Domain of Interpretation," RFC 3547, Jul. 2003.

[114] H. Wen and C. Yao. (2007) "Improved Register Procedure in PIM-SM". Internet Draft, Work in progress. [Online]. Available: http://tools.ietf.org/id/draft-wen-pim-improved-register-00.txt

[115] R. Yavatkar, D. Pendarakis and R. Guerin, "Policy-based Admission Control," RFC 2753, Jan. 2000.