Universidade do Porto

Faculdade de Engenharia

FEUP

António Alberto dos Santos Pinto

Key Distribution Technique for IPTV Services with Support for Admission Control and User Defined Groups

Dissertação apresentada à Faculdade de Engenharia da Universidade do Porto para obtenção do grau de Doutor em Engenharia Electrotécnica e de Computadores, realizada sob a orientação do Prof. Doutor Manuel Alberto Pereira Ricardo, Professor Associado da Faculdade de Engenharia da Universidade do Porto

Departamento de Engenharia Electrotécnica e de Computadores
Faculdade de Engenharia da Universidade do Porto
2010

To Pedro and Maria.

Abstract

Technological evolution is leading telecommunications to all-IP networks where the traffic generated by services is transported as IP packets. Among these are the group communications services with confidentiality requirements. IPTV services consist of multiple video channels grouped in bundles, such as the sports, movies or generic bundle; an user typically subscribes multiple bundles, including the generic bundle.

Secure IP multicast can be used to implement IPTV services, but it still has problems to be addressed when used in scenarios (a) consisting of large number of receivers having differentiated access rights, (b) where bundles need to be supported, and (c) users need to switch rapidly between channels (channel surfing or zapping).

Moreover, IP multicast is not optimized for admission control. Some solutions support receiver admission control, other solutions modify IGMP/MLD protocols, while others use approaches not interoperable with standard Authentication, Authorization and Accounting (AAA) protocols such as RADIUS. On the other hand, the layer 2 access network technologies are starting to provide support for multicast, but they are not yet fully integrated with the admission control mechanisms of the IP converged networks. Admission control for group communications, in particular those associated to IPTV services, still has open issues.

Enabling users to be sources of content and provide these users with the capability of identifying a list of receivers constitutes another research topic of IPTV services. This scenario requires that the user source of contents generates cryptographic contexts and a list of authorized receiver users. The formation of multicast trees involving the sender and possibly small groups of receivers, combined with the admission control techniques, is also an open issue.

This thesis addresses these issues and proposes a new and secure cryptographic key distribution technique that enables receiver access control, uses low resources for signalling, and does not produce a perceivable impact on video quality during channel switching. The proposed solution explores the concept that, besides the bundle key (KEK), each channel will also have one data key (VEK) that, by being shared by all group members, generates a constant signalling bitrate in refresh operations.

A new admission control technique for both multicast senders and receivers is also proposed. This technique enables the management of a multicast session that spawns over heterogeneous access networks. It adopts multicast profiles that specify if an user is allowed to generate videos and contains a list of video channels the user is authorized to access. These multicast profiles are stored in an AAA server. Upon authorization, the multicast tree is formed and extended to every new member. This solution seems to be easily integrated in the IPTV architectures being developed by ETSI and ITU-T. The proposed cryptographic key distribution technique was also modified in order to work in scenarios of video channels sourced at domestic users.

Resumo

A evolução tecnológica tem conduzido as telecomunicações para cenários de redes completamente baseadas em IP, em que os serviços são todos transportados sob a forma de pacotes IP. Entre estes encontram-se os serviços de comunicação para grupos com requisitos de confidencialidade. Os serviços IPTV são constituídos por múltiplos canais de vídeo agrupados em pacotes temáticos como, por exemplo pacotes de desporto, filmes ou o pacote genérico. Um utilizador contrata normalmente um ou mais pacotes, incluindo o pacote genérico.

O IP multicast seguro pode ser uma solução para implementar serviços IPTV. Contudo, quando utilizado em cenários em que exista um grande número de utilizadores com acessos diferenciados ao serviço, em que seja necessária agrupar os múltiplos canais em pacotes e em que se pretenda que os utilizadores mudem rapidamente de canais (*zapping*), ainda existem problemas que necessitam de ser estudados.

Além disso, o multicast IP não tem suporte para controlo de admissão. Algumas soluções existentes permitem apenas o controlo de admissão para receptores, outras soluções modificam os protocolos IGMP/MLD, enquanto outras não são compatíveis com protocolos AAA standard como o RADIUS. Por outro lado, as tecnologias de acesso à rede de nível 2 começam a suportar comunicações multicast, sem contudo integrarem mecanismos de controlo de admissão. O controlo de admissão em serviços de IPTV ainda tem questões em aberto.

Permitir que os utilizadores possam gerar e transmitir os seus próprios conteúdos e que possam identificar uma lista de receptores constitui outro tópico de investigação. Este cenário requer que o utilizador gerador de conteúdos seja capaz de gerar contextos criptográficos e uma lista de receptores autorizados. A formação de árvores multicast com origem no emissor,

destinadas a pequenos grupos de receptores e com suporte para o controlo de admissão é também uma questão em aberto.

Esta tese propõe uma nova técnica de distribuição segura de chaves criptográficas que impõe o controlo de acesso aos receptores, usa poucos recursos de rede para sinalização e não introduz atrasos perceptíveis durante as mudanças de canal. A solução proposta explora o conceito que, para além da chave de pacote de canais (KEK), cada canal terá também uma chave de canal (VEK). Como a chave de canal é partilhada por todos os membros do grupo há uma redução significativa dos recursos de rede utilizados no transporte da sinalização.

Adicionalmente, também é proposta uma técnica de controlo de admissão para emissores e receptores multicast. Esta técnica permite a gestão de sessões multicast sobre redes de acesso heterogéneas. Recorre a perfis multicast, armazenados no servidor AAA, para especificar se um utilizador tem permissão para gerar vídeos e para armazenar a lista de canais de vídeo a que tem acesso. As árvores multicast só são formadas após a verificação da sua autorização. A solução proposta pode ser integrada nas arquitecturas IPTV em desenvolvimento pela ETSI e pelo ITU-T. A técnica proposta para a distribuição de chaves criptográficas foi ainda modificada para permitir a sua utilização em situações em que os utilizadores domésticos geram os seus próprios canais de vídeo.

Résumé

L'évolution technologique majeure actuelle des télécoms tend vers les réseaux "tout IP", au sein desquels le trafic généré par les services est transporté dans des paquets IP, dont les communications de groupe incluant des pré-requis de confidentialité. Les services IPTV consistent en de multiples chaines vidéo regroupées en bouquets thématiques (sports, films ou bouquet de base). Typiquement, un utilisateur souscrit à plusieurs bouquets, dont le bouquet de base.

Les services d'IPTV peuvent être implantés grâce aux protocoles IP multicast sécurisés. Cependant, des problèmes importants restent à résoudre quant à leur utilisation dans des scenarii impliquant (a) utilisation d'une offre de type bouquet; (b) grand nombre de clients IPTV (récepteurs) ayant des droits d'accès différent; (c) les utilisateurs demandant pouvoir rapidement changer de chaînes (zapping).

De plus, l'IP multicast n'est pas optimisé pour le contrôle de l'admission. D'une part, Il existe des solutions supportant le contrôle d'admission du récepteur, d'autres modifiant les protocoles IGMP/MLD, tandis que d'autres utilisent des approches non interopérables avec les protocoles AAA standards tels que RADIUS. D'autre part, même si les technologies d'accès de la couche 2 commencent à fournir le support pour le multicast, elles ne sont pas encore entièrement intégres aux mécanismes de contrôle d'admission des réseaux convergents IP. Le contrôle d'admission pour les communications de groupe, particulièrement celui associé aux services IPTV, restent des sujets inexplorés.

Permettre aux utilisateurs d'être sources de contenu et les rendre capable d'identifier une liste de récepteurs constitue un autre sujet de recherche dédié aux services IPTV. Ce scénario exige que l'utilisateur source de contenus génère des contextes cryptographiques ainsi qu'une liste d'utilisateurs

récepteurs autorisés. La formation d'arbres multicast, impliquant l'émetteur et potentiellement de petits groupes de récepteurs, combinée à des techniques de contrôle d'admission, représente également un sujet encore inexploré.

Cette thèse propose une nouvelle technique sécurisé de distribution de clé cryptographique permettant le contrôle d'accès du récepteur, tout en utilisant de faibles ressources en terme de signalisation, et sans impact perceptible sur la qualité vidéo au cours du changement de chaîne. La solution proposée explore l'idée d'un concept au sein duquel, en complément de la cle "bundle" (KEK), chaque chaîne disposera également d'une clé de donnée (VEK) qui, du fait d'être partagée par tous les membres d'un même groupe, nécessite une signalisation constante pour le rafraîchissement d'opérations et permet une réduction significative de la signalisation.

Au sein de cette thèse il est également proposée une nouvelle technique de contrôle d'admission pour les émetteurs comme les récepteurs multicast. Cette technique permettra le management d'une session multicast qui se retrouve dans tous les réseaux d'accès hétérogènes. Elle adopte des profils multicast qui spécifieront si un utilisateur est autorisé à générer des vidéos et inclura une liste de chaines vidéo auxquelles l'utilisateur aura un accès autorisé. Ces profils multicast seront stockés dans un serveur AAA. Sur autorisation, l'arbre multicast se formera et s'étendra à chaque nouveau membre. Cette solution pourra être facilement intégrée au sein d'architectures IPTV développées par ETSI et ITU-T. La technique de distribution de clé cryptographique proposée a également été modifiée de sorte de pouvoir fonctionner dans des scénarii de canaux vidéo générés par des utilisateurs domestiques.

Acknowledgments

I would like to start by thanking my supervisor, Prof. Manuel Ricardo, for his help, guidance, availability and patience, especially when considering the work spent in reviewing this thesis and the short time window to do so.

I would like to thank my family for their support and understanding. To my wife, Carla, for all her efforts in creating conditions that allowed me to work.

Additionally, I would like to thank Olimpiu Negru for his help in translating the abstract to French.

Contents

List of Figures

List of Tables

Acronyms

3GPP 3rd Generation Partnership Project

AAA Authentication, Authorization and Accounting

ACL Access Control List

ACS Access Control Server

ASM Any-Source Multicast

ASN-GW Access Service Network - Gateway

BM-SC Broadcast/Multicast - Service Center

BNG Broadband Network Gateway

BS Base Station

CBA Cryptographically Based Address

CBT Core Based Trees

CID Connection Identifier

CPE Customer Premises Equipment

CR Core Router

DHCP Dynamic Host Configuration Protocol

DoS Denial of Service

DSLAM Digital Subscriber Line Access Multiplexer

DVMRP Distance Vector Multicast Routing Protocol

EAP Extensible Authentication Protocol

EPG Electronic Program Guide

ETSI European Telecommunications Standards Institute

FEC Forward Error Correction

G-CBA Group Cryptographically Based Address

GC Group Controller

GGSN Gateway GPRS Support Node

GKDC Group Key Distribution Center

GPRS General Packet Radio Service

HTTP Hypertext Transfer Protocol

IANA Internet Assigned Numbers Authority

ICMPv6 Internet Control Message Protocol for IPv6

IETF Internet Engineering Task Force

IGMP-AC Internet Group Management Protocol with Access Control

IGMP Internet Group Management Protocol

IIL Incoming Interface List

IMS IP Multimedia Subsystem

IGKMP Intra-domain Group Key Management Protocol

IPoE Internet Protocol over Ethernet

IPSec IP Security

IPTV Internet Protocol Television

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

IV Initialization Vector

MAC Medium Access Control

MBMS Multimedia Broadcast/Multicast Service

MCA Multicast Controlling Agent

MCDA2 Multicast Content Distribution Architecture with Accounting support

mCID Multicast Connection Identifier

MCOP Multicast Control Protocol

MLD Multicast Listener Discovery Protocol

MOSPF Multicast Open Shortest Path First

MSDP Multicast Source Discovery Protocol

NACF Network Attachment Control Functions

NAS Network Access Server

NGN Next Generation Networks

OIL Outgoing Interface List

OSPF Open Shortest Path First Routing Protocol

PDN Packet Data Networks

PIM-DM Protocol Independent Multicast - Dense Mode

PIM-SM Protocol Independent Multicast - Sparse Mode

PIM Protocol Independent Multicast

PKI Public Key Infrastructure

PPPoE Point-to-Point Protocol over Ethernet

QoE Quality of Experience

QoS Quality of Service

RACF Resource and Admission Control Functions

RIP Routing Information Protocol

RP *Rendevouz* Point

RPF Reverse Path Forwarding

RTCP RTP Control Protocol

RTP Real-time Transfer Protocol

RTSP Real Time Streaming Protocol

TISPAN Telecommunications and Internet converged Services and Protocols for Advanced Networking

SCP Service and Content Protection

SIP Session Initiation Protocol

SMKD Scalable Multicast Key Distribution

SPT Shortest Path Trees

SRTP Secure Real-time Transport Protocol

SS Subscriber Station

SSM Source-Specific Multicast

STB Set-Top Box

TCP Transmission Control Protocol

UDP User Datagram Protocol

UE User Equipment

UGV User Generated Videos

UMTS Universal Mobile Telecommunication System

VCR Video Cassette Recorder

VoD Video on Demand

VoIP Voice over IP

VS Video Server

WiMAX Worldwide Interoperability for Microwave Access

xDSL Digital Subscriber Line

Chapter 1

Introduction

Technological evolution is leading telecommunications to all-IP networks where the traffic generated by services is transported as IP packets. Among these are the group communications services with confidentiality requirements. IPTV services consist of multiple video channels grouped in bundles, such as sports, movies or generic bundle; an user typically subscribes multiple bundles, including the generic bundle.

Secure IP multicast [68, 67] may be used to support IPTV services, since this technology enables the secure transmission of IP packets to groups of receivers. Access network technologies also offer mechanisms that can be used to optimize multicast communications. xDSL networks may optimize multicast communications in both directions (uplink and downlink), while WiMAX and UMTS can support optimized link-layer multicast communications in the downlink. Despite the scalability of multicast techniques, the network operators have been reluctant to use them [34] due to the lack of native control they offer over groups, making it difficult for network operators and service providers to perform access control, traffic accounting, and network management.

On the other hand, the increasing bandwidth being offered to residential users, combined with the proliferation of techniques to produce rich user generated content, suggests that a user will also be compelled to generate and distribute his real-time videos to groups of other users, directly from his premises. This trend requires network operators to protect also these user generated videos to what concerns confidentiality and access control.

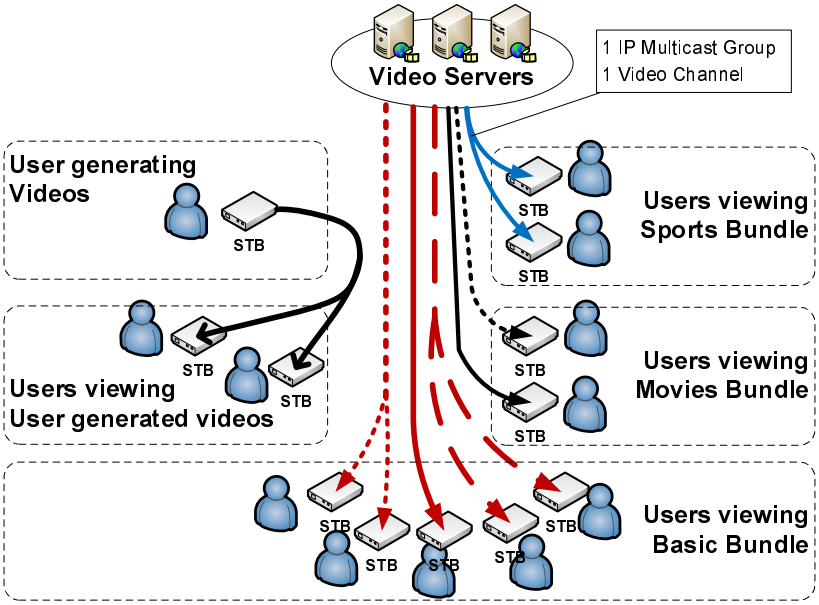

Figure 1.1: Reference scenario

1.1 Problem characterization

The reference scenario adopted for this work is shown in Figure 1.1. It describes an IPTV service where video channels are distributed as IP packets and transmitted to a multicast address - one multicast group per video channel. Multiple video channels are grouped together, in bundles, and may be distributed to a group of receivers with equal access to the video channels of the bundle. A bundle is thus composed of several video channels, each video channel transmitted to a different multicast address. In what concerns security, common IPTV services use one key for each video channel. In this work we test the hypothesis that, besides the video channel key, each bundle can also have one bundle key. The video channels are generated by Video Servers (VS) to groups of Set-Top Box (STB). A STB may also generate video contents and uses heterogeneous access networks to access the IPTV service, including xDSL, WiMAX or UMTS.

The analysis of the state of the art and work related to the reference scenario, led to the identification of three main problems:

1. **generation of large amounts of signalling in video channel zapping situations**. Secure IP multicast still has problems to be addressed when used in IPTV scenarios (a) consisting of large num-

ber of receivers having differentiated access rights, (b) where bundles need to be supported, and (c) users need to switch rapidly between channels (channel surfing or zapping). Several proposals exist in the literature for providing scalable secure group communications using secure IP multicast [26, 24, 85, 82, 65, 60, 36, 21, 75]. These solutions aim at securing the data sent to a group of users with equal access rights but do not address bundles of video channels. For instance, user A may subscribe the generic and sports bundles, while user B may only subscribe the generic bundle; in this case encryption keys are required both for individual channels and bundles, and no existing solution seems to address this problem. Moreover, existing solutions do not optimize the signalling generated by the IPTV system when users switch between groups, which happens in channel zapping situations, where the zapping user needs to retrieve new cryptographic material;

2. **lack of support for groups sourced at the users**. The problem here consists in enabling users to be sources of content and to provide these users with the capability of identifying a list of receivers. The current solutions [26, 24, 85, 82, 65, 60, 36, 21, 75] do not address the scenario where users can assume the role of content providers. The key issue to address in this case is related to group key management. Control may be imposed by delivering the group decryption keys to the allowed users, which are a subset of the remaining users. This approach requires that the source of contents also generates cryptographic contexts and provides the system with a list of authorized users. Moreover, the transmission of multicast streams directly from the user premises is not addressed by current network access technologies, making it difficult the optimized transmission for groups directly from user's homes;

3. **weak support for admission control in IP multicast for both sources and receivers**. IP multicast does not provide admission control. Some solutions [16, 52, 25] support receiver admission control, other solutions [50, 45] require modifications to IGMP/MLD protocols, while others do not use standard AAA protocols [56] such as RADIUS. The network access technologies are starting to provide support for IP

Multicast through the introduction of optimized link layer multicast, but they still neglect multicast admission control, in particular in what concerns interactions with AAA.

The main objectives of this work are then to define a secure IPTV solution that, cumulatively: a) enforces individual access control to groups of real-time IPTV video channels; b) enforces IP multicast admission control for both multicast senders and receivers; c) supports user generated videos; d) generates low signalling overheads; e) does not introduce perceivable delays, particularly in video channel zapping situations.

1.2 Contributions

The work carried out in this thesis led to the production of three original contributions:

1. a key distribution technique that enables receiver access control, uses small amount of resources for signalling, and does not produce a perceivable impact on video quality during channel switching;

2. an admission control technique adequate for both multicast senders and receivers;

3. a key distribution technique used to encrypt video content sourced at domestic users.

To the best of our knowledge these contributions are new and original.

1.3 Organization

This thesis is structured in seven chapters. The next chapter describes the state-of-the-art addressing video transmission over IP, IP multicast, and multicast transmission in access networks. Chapter 3 introduces related work, namely in multicast admission control and secure multicast. The proposed solution is presented in Chapter 4, starting with the presentation of its architecture, followed by the description of the system elements and their interfaces, and then the description of the implemented solution. Chapter

5 describes the validation of the proposed solution, starting with the security analysis of the proposed protocol, followed by the both experimental and simulation results, and then the comparison of the proposed solution against concurrent solutions. Chapter 6 aims at demonstrating that the proposed solution can be integrated into recent IPTV architectures, namely the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) IPTV Architecture that also address deployment scenarios for Next Generation Networks (NGN), with or without IP Multimedia Subsystem (IMS) functionalities. Concluding this thesis, Chapter 7 makes global considerations, characterizes the results obtained and discusses future work.

Chapter 2

IPTV and Multicast

Technological evolution is leading telecommunications to all-IP networks where traffic generated by multiple services is transported as IP packets. Among these are the video streaming services such as Internet Protocol Television (IPTV). IPTV services demand large amounts of network resources since they consist usually of multiple real-time video channels that are always available to users. IP multicast may be used to reduce bandwidth usage. This technique enables a single transmitted packet to be delivered to multiple receivers enabling some network nodes to create multiple instances of a received packet. However the optimization obtained by IP multicast becomes compromised by the behavior of some access network technologies, in particular those not supporting link layer multicast transmission.

This chapter is composed of three sections. Section 2.1 addresses the efforts currently made by the Internet Engineering Task Force (IETF) and ITU-T to support the deployment of the IPTV service; these are based on multicast communications. Section 2.2 describes the IP multicast techniques. Section 2.3 addresses the multicast support provided by the access network technologies that will be part of NGN.

2.1 Video over IP

Video transmission over IP, or video streaming as defined in [86], can be characterized by a video being played out while other parts of the same video are being received and decoded, thus avoiding full video download before decoding and visualization. Video streaming quality can be affected

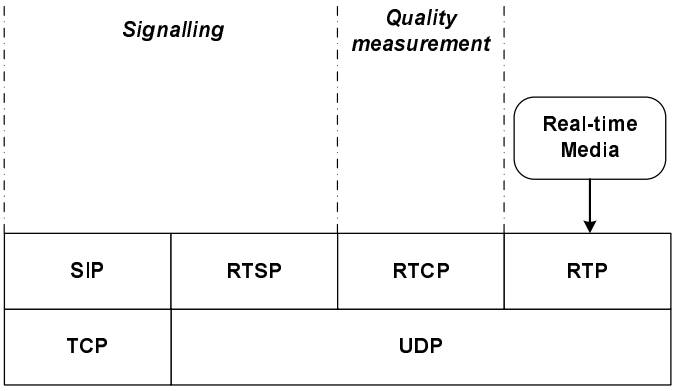

Figure 2.1: IETF Multimedia architecture

by the link quality (bit rate and bit error ratio) and load.

Efforts have been made to standardize video streaming over IP, including the functionalities required at network, transport, and session layers. The IETF multimedia architecture has defined Real-time Transfer Protocol (RTP) [73] that enables the transmission of video, voice and multimedia contents in IP packets, along with other protocols for controlling the video streaming. More recently [7, 8], these protocols have been re-used by organizations such as the ITU-T and European Telecommunications Standards Institute (ETSI) to integrate IPTV services in the NGN architecture, defined by Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN). Key issues of these ETSI and ITU-T activities are the mobile-fixed service convergence and the optimized transmission of video streams over heterogeneous access networks, namely xDSL, WiMAX and UMTS.

2.1.1 IETF Multimedia Architecture

The IETF multimedia architecture is shown in Figure 2.1 and it includes protocols such as Session Initiation Protocol (SIP) [70], Real Time Streaming Protocol (RTSP) [74], RTP and RTP Control Protocol (RTCP) [73]. RTP is used to transport real-time data such as a video stream. The other protocols are used to control the session and the flow of data transmission.

Real-time Transport Protocol

RTP is an end-to-end transport protocol suited for real-time applications. RTP provides real-time functionalities not present in UDP, including sequencing, intra-media synchronization, inter-media synchronization, payload identification, and source identification. Sequencing enables real-time packet re-ordering, should they arrive out of order to the receiver. Intra-media synchronization enables the timely playing of successive packets thus preventing time relationships at the source; for instance, a silence period does not require packet transmission, but the duration of the silence period must be conveyed. Inter-media synchronization enables, for instance, time synchronization between a video and an audio stream that composes a video channel. The payload type identification eases the decoding process. The source identification enables the receivers to rapidly distinguish among different sources.

RTP is mainly used over UDP, in order to take advantage of data multiplexing and multicast transmission. The multi-user conference was the main scenario addressed during the development of RTP protocol, but RTP can have other applications benefiting from its features, such as continuous data storage for network backup systems, interactive or distributed simulations, and continuous control or measurement applications. The RTP main actors are the receivers and senders, also known as end systems. Middle elements were also defined, in order to support new functionalities, the mixers and the translators; a translator is aimed at changing the stream coding, while the mixer aims at combining a set of streams into a single stream.

RTP Control Protocol

RTP is usually accompanied by RTCP; the latter monitors the delivery of data packets and enables operations such as participant identification, Quality of Service (QoS) feedback, control packet scaling, inter-media synchronization, and minimal session control information. For instance RTCP enables session participants to provide feedback to senders on the session reception quality by sending messages periodically.

The messages sent by receivers are named receiver reports, while the messages sent by sources are named sender reports. These reports contain information such as RTP packets lost since the last report, total number of

packets lost, jitter in packet arrival, and delay since last senders report. This information enables the estimation of transmission rates, localization of congestion points, and the evaluation of network performance [86]. RTCP can also provide a more human friendly mechanism for source identification by providing textual descriptions. In order to prevent bottlenecks and to scale, the number of packets sent on the control plane is based on a percentage of the total session bandwidth (5%), 25% of which for sender reports and 75% for receiver reports. Synchronization between associated media streams is accomplished based on information of both time and relative timestamps.

Real Time Streaming Protocol

The main objective of the RTSP protocol is to provide Video Cassette Recorder (VCR)-like control of video streaming sessions, live or recorded, but it also enables the selection of the stream transport mode (UDP, TCP, unicast, multicast). RSTP main functions include those provided by a TV remote control: stop, pause, play, fast forward and fast backward. Functionalities such as session information retrieval, and the notification of clients and servers about the availability of new media contents may also be supported when aggregated to established sessions.

Session Initiation Protocol

SIP is used to initiate and terminate sessions between one or more participants. A difference between SIP and RTSP is that SIP supports user mobility through user request proxying and redirection to the user's current position. SIP is a text client-server protocol where the clients issue requests and servers respond; the message use Hypertext Transfer Protocol (HTTP) [41] like syntax. A SIP request represents a method invocation on the server; there are six methods defined in [70], being the most relevant the INVITE, used by clients to initiate a session with a server.

The SIP architecture consists of two main component types: user agents and network servers. User agents can act either as clients or servers of the protocol; the user agent client is used to start a session, and the user agent server is used to receive a session. Both user agents support peer-to-peer operation. The network servers can be of three types: proxy, redirect or registrar. A SIP proxy is similar to an HTTP proxy server, i.e. it forwards

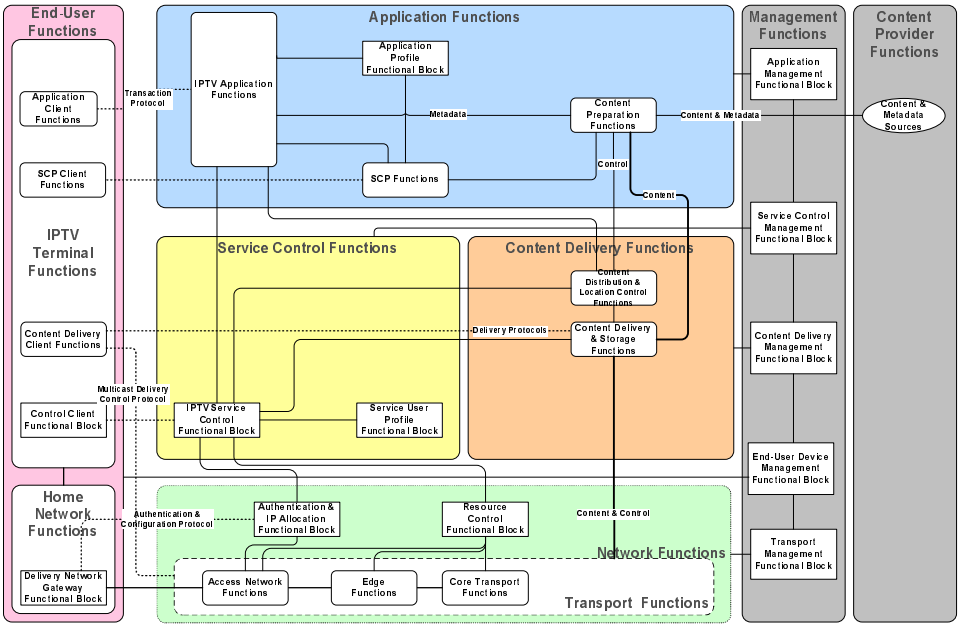

Figure 2.2: ITU-T IPTV functional architecture framework

the user agent requests to the next server. A SIP redirect server, after receiving the user agent request finds the next server and instructs the user agent to redirect the request to the next server. A SIP registrar enables user agent location registration and processes user agents REGISTER requests.

2.1.2 ITU-T IPTV Architecture

The ITU-T IPTV functional architecture, defined in [7], is characterized by the functional blocks shown in Figure 2.2. End-User Functions enable home users to access the IPTV services. Application Functions enable home users to discover, select and purchase IPTV content. Service Control Functions provide mechanisms to request and release the required network resources. Content Delivery Functions store, process and deliver content, received from the Application Functions, to the End-User Functions. Network Functions provide IP connectivity between IPTV service components. Management Functions perform the system management, enabling network operation, administration and maintenance. Content Provider Functions include the functionalities used by the content generator, namely content description and usage rights.

End-User Functions

The End-User Functions comprise both IPTV Terminal Functions and Home Network Functions, namely Application Client Functions, Service and Content Protection (SCP) Client Functions, Content Delivery Client Functions, and Control Client Functional Block. The Application Client Functions interact with the Application Functions in order to exchange IPTV service information such as Electronic Program Guide (EPG) and to perform content discovery and content selection. SCP Client Functions are responsible for content decryption and verification of usage rights. The Content Delivery Client Functions receive the content and may, optionally, support playback control. The Control Client Functional Block is responsible for the connection to the Content Delivery Functions.

Application Functions

The Application Functions enable IPTV service selection; they receive requests, perform authorization and trigger IPTV service delivery based on user profiles, content and other information such as EPG. In order to trigger content delivery, the Application Functions interact with the Content Delivery Functions.

The Application Functions comprise the Application Profile Functional Block, the SCP Functions, and the Content Preparation Function. The Application Profile Functional Block stores information such as the capabilities of the end-user's IPTV terminal device, language settings, and a list of subscribed video channels. The SCP Functions include the access control to contents, content protection using methods such as encryption, and service protection by means of authentication and authorization. The Content Preparation Functions perform operations such as *transcoding* or watermarking.

Service Control Functions

Service Control Functions includes the IPTV Service Control Functional Block and the Service User Profile Functional Block. The IPTV Service Control Functional Block handles requests for service initiation, modification and termination; it also establishes and maintains the required network resources. The Service User Profile Functional Block stores the end-user

service profile that includes the list of subscribed IPTV services (TV, Video on Demand (VoD)), location information, presence status (online/offline), and charging information; it is also responsible for replying to queries from other blocks related to the information stored.

Content Delivery Functions

The Content Delivery Functions are responsible for storage and content caching, and for delivering such content upon end-user's request; it comprises Content Distribution & Location Control Functions and Content Delivery & Storage Functions.

The Content Distribution & Location Control Functions interact with the IPTV Service Control Functional Block, control the distribution of content from the Content Preparation Functions to the Content Delivery & Storage Functions, collect information related resource usage, content distribution and load status. This information enables the selection of the appropriate instance of Content Delivery & Storage Functions to deliver the content to the end-user.

The Content Delivery & Storage Functions are responsible for the delivery of content to the end-users, content adaptation (watermarking, *transcoding* and encryption), local storage of content, status reporting to the Content Distribution & Location Control Functions, and generation of charging information.

Network Functions

The Network Functions provide IP connectivity to all components and devices; it comprises the Authentication & IP Allocation Functional Block, the Resource Control Functional Block, and the Transport Functions. The Authentication & IP Allocation Functional Block is responsible for end-user's device authentication during network attachment, and its IP configuration. The Resource Control Functional Block enables the management of the resources allocated for the delivery across access, edge and core networks. The Transport Functions are responsible for forwarding IPTV traffic throughout access, edge and core networks.

Management Functions

The Management Functions are responsible for global system management, namely status monitoring and configuration; it comprises the Application Management Functional Block, the Content Delivery Management Functional Block, the Service Control Management Functional Block, the End User Device Management Functional Block, and the Transport Management Functional Block. Each functional block is responsible for status monitoring and configuration of the respective group of functions.

Content Provider Functions

The Content Provider Functions provide the content and metadata to the Content Preparation Functions. The metadata comprises information such as content protection rights, EPG, and content source addresses.

2.2 IP Multicast

IP Multicast [32] data distribution is done to avoid unnecessary packet replication in each network segment, ensuring that only one flow of data exists in the path from the sender to the receivers. A network node - the multicast router, replicates packets only to links where there exist receivers. IPTV services rely on IP multicast to transmit the video channels only to parts of the network where there are interested users.

Key concepts associated to IP multicast are group, source, and distribution tree. A multicast group is a set of systems interested in receiving the same stream, generated by one or more sources. A group is identified by an IP multicast address. The source is the element which generates the stream; it knows only the group address, and does not have information regarding the identity of the receivers. A distribution tree, as shown in Figure 2.3, represents the path used by the stream to reach the group members, from the source of the group.

IP Multicast uses two types of protocols: group management protocols and multicasts routing protocols. Group management protocols are used by members to signal their interest in joining or leaving a multicast group; these protocols allow routers to define the links through which the data is to be forwarded. Multicast routing protocols are then used to create

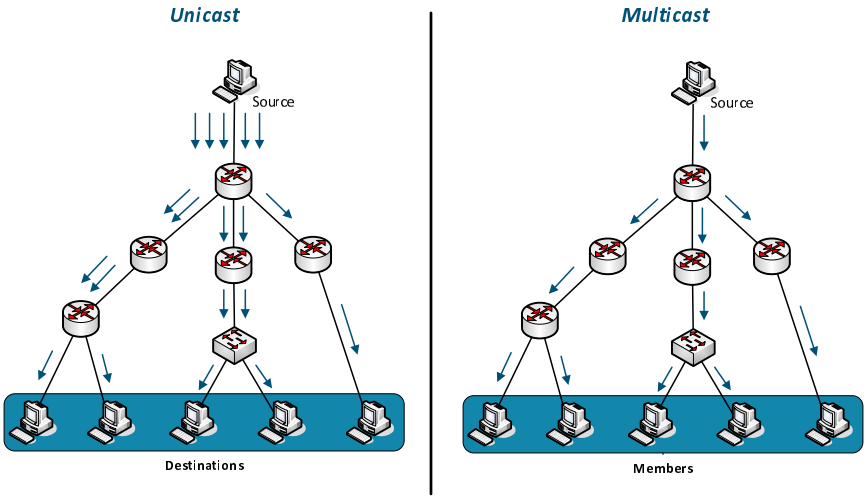

Figure 2.3: Unicast vs. Multicast transmission

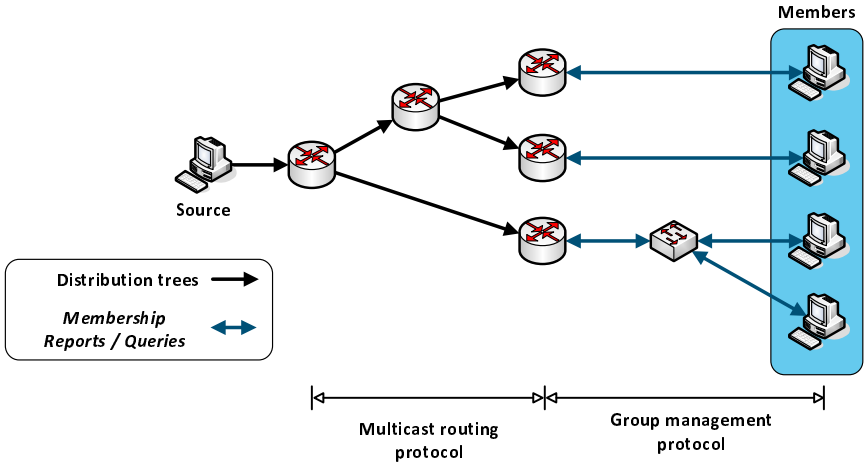

Figure 2.4: IP multicast required elements

and maintain the distribution tree associated to each group that has active members. Figure 2.4 shows the required multicast elements.

2.2.1 Group management

Group communication require group creation and group management. Two main protocols address these issues: the Internet Group Management Protocol (IGMP) [40, 22], and the Multicast Listener Discovery Protocol (MLD) [33, 80]. IGMP is the multicast group management protocol for IP version 4 (IPv4) networks, and MLD is used in IP version 6 (IPv6) networks.

IGMP

In its version 1, IGMP defines that a host wanting to belong to a multicast group must send a message to the address 224.0.0.1 (all systems on this subnet), informing the local multicast router about its interest. Besides handling this request, the router periodically polls the local network to check whether a host maintains its interest; if not, the router stops transmitting the multicast group data and informs neighbor routers about it. Two types of IGMP messages are used for these purposes:

- Membership Report, used by hosts to indicate their interest in joining a group;
- Membership Query, sent periodically by routers in order to verify whether there is at least one host interested in receiving the multicast data flow.

In IGMP version 1 the departures from the multicast group are inferred by the routers based on timeout and induce some latency; in order to reduce it, IGMP version 2 introduces a group leave message, used by the receiver to inform the local multicast router about its intention to leave the group. Besides the new leave message, IGMP version 2 works as version 1 and, in order to be retro-compatible, a new version of the Membership report was also introduced. Therefore, two new types of messages were defined in version 2:

- Membership Report, used by hosts to indicate their interest in joining a group;

- Leave Group, used by hosts to indicate their intention to leave the group.

The latest version of the IGMP protocol, version 3, introduces the possibility of a host indicating the list of sources from which it wants to receive multicast traffic. A new type of message was then introduced in version 3:

- Membership Report, used by a host to indicate its interest in joining a group and specify the source(s) from which it wants to receive data.

MLD

Similarly to IGMP on IPv4 networks, MLD enables IPv6 routers to discover the presence of multicast listeners on directly attached links, and to discover the multicast addresses of interest to those listeners. MLDv2 is a translation of IGMPv3 to IPv6 semantics. Multicast traffic filtering by source address is enabled by MLDv2; a multicast listener can describe both the sources it wants to receive multicast traffic from, as well as the sources that it wants to exclude. MLD, version 1, supports 3 message types:

- Multicast Listener Query, sent periodically by routers in order to verify whether there is at least one host interested in receiving the traffic of a multicast group;
- Multicast Listener Report, used by hosts to indicate their interest in receiving the traffic of a multicast group;
- Multicast Listener Done, used by hosts to indicate the termination of their interest in receiving traffic from a multicast group.

MLD version 2 became part of the Internet Control Message Protocol for IPv6 (ICMPv6) [30]. MLDv2 messages have as source address a link-local IPv6 address with a hop limit of 1 and a Hop-by-Hop Options header with the Router Alert option set. MLDv2 introduces a new message type:

- Version 2 Multicast Listener Report, used by hosts to indicate their interest in the traffic of a multicast group and to specify traffic filtering options.

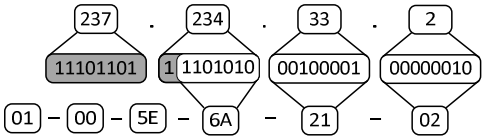

Figure 2.5: Layer 2 multicast and IPv4 address mapping

2.2.2 Multicast addressing

Unlike unicast addresses, which identify a unique network interface, multicast addresses identify groups of equipments interested in receiving the same information. Multicast addresses must be differentiated from unicast addresses and, for that purpose, Internet Assigned Numbers Authority (IANA) reserved the former IPv4 class D address range to be used in multicast communications [47]. IPv4 Class D ranges from 224.0.0.0 to 239.255.255.255. IANA also reserved a set of link layer addresses to allow Layer 2 equipment, such as Ethernet switches, to recognize multicast traffic. In this case, the 01-00-5E prefix was adopted, meaning that the link layer addresses ranging from 01-00-5E-00-00-00 to 01-00-5E-7F-FF-FF are reseved for Layer 2 multicast communications.

The Layer 2 multicast address is calculated by appending the 01-00-5E prefix to the 23 least significant bits of the IP multicast address. Figure 2.5 shows an example of such address mapping. As a result, the relation between IPv4 and Layer 2 multicast address is of 32:1, meaning that for each Layer 2 address there are 32 IPv4 address possibilities. Figure 2.6 shows the IPv4 addresses that map to the 01-00-5E-01-01-01 address. As a result, the Layer 2 equipments view of multicast flows is necessarily different from Layer 3 equipments. For instance, assuming that an user A joins the group 224.1.1.1 and an user B joins the group 238.1.1.1, if both users A and B are connected to the same switch, the switch will forward the multicast traffic of both groups to both users.

2.2.3 Reverse path forwarding

Reverse Path Forwarding (RPF) is an essential mechanism in multicast packet forwarding. Network equipment decide if they should forward multicast packets depending on the result of the RPF verification. RPF verification consists in checking whether the network interface used to receive

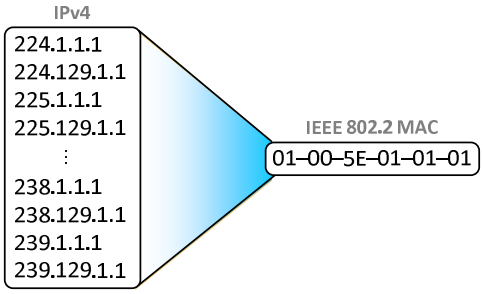

Figure 2.6: IPv4 addresses mapping to the 01-00-5E-01-01-01 address

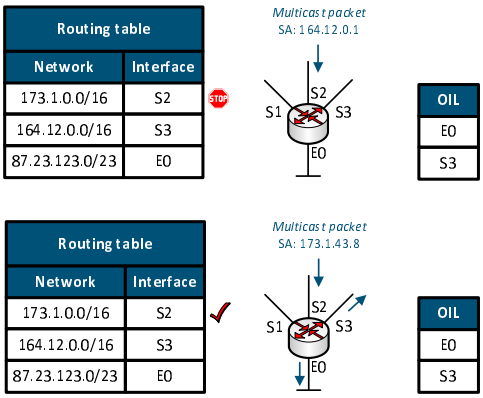

Figure 2.7: Example of successful and unsuccessful RPF verifications

the multicast packet is in the shortest path to the source of the packet; if not, the packet is discarded. In unicast, the decision of the interface to use to forward a packet, is based on the destination address of the packet; in multicast, this forwarding decision is based on the source address of the packet.

A router maintains two lists for each multicast group: the Incoming Interface List (IIL) and the Outgoing Interface List (OIL). The IIL is used to store the network interfaces which receive multicast packets. The OIL is used to store the network interfaces for which the multicast packets have to be forwarded. A router may use a multicast routing table or the unicast routing table.

Figure 2.7 shows an example of successful and unsuccessful RPF verifications. In the unsuccessful case, the multicast packet with the source address 164.12.0.1 was received from the S2 interface, which is not the interface in the shortest path to the 164.12.0.0/16 network (it should be the

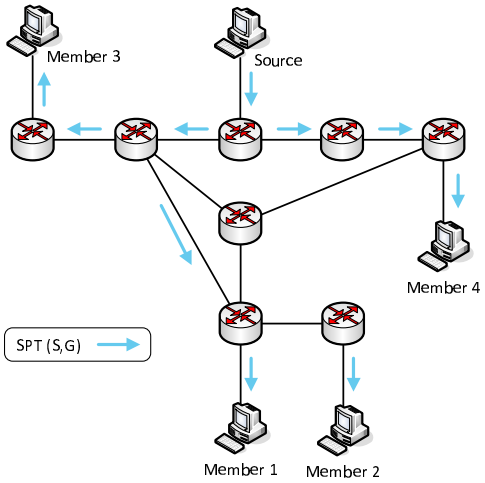

Figure 2.8: Example of a shortest path tree

S3 interface); in this case the packet is discarded. In the successful case, the multicast packet with the source network address 173.1.43.8 was received from the S2 interface, which is the interface in the shortest path to the 173.1.0.0/16 network; in this case the packet is forwarded to all interfaces associated to interested receivers, that is, the S3 and E0 interfaces.

2.2.4 Distribution trees

A multicast distribution tree represents the path used by the stream to reach all the members, starting at the source of the group. These distribution trees, built by routers using routing protocols and based on IGMP/MLD messages, are of two types: source trees, and shared trees. Source trees have as root the source of the multicast stream. Shared trees have the root of the distribution tree in a central point of the network, typically a router.

Shortest Path Trees (SPT) are an example of source trees, as shown in Figure 2.8, where the tree branches are the shortest paths to the members. A SPT is built by using the RPF verification in the intermediate routers and is represented by the tuple (S,G), where S is the unicast address of the source of the group, and G is the multicast address of the group. For each tuple (S,G) there can be only one SPT.

Core Based Trees (CBT) are an example of shared trees where the root of the tree is a specified router, named Core Router (CR) or *Rendevouz* Point (RP) depending on the routing protocol. Because a CBT can have

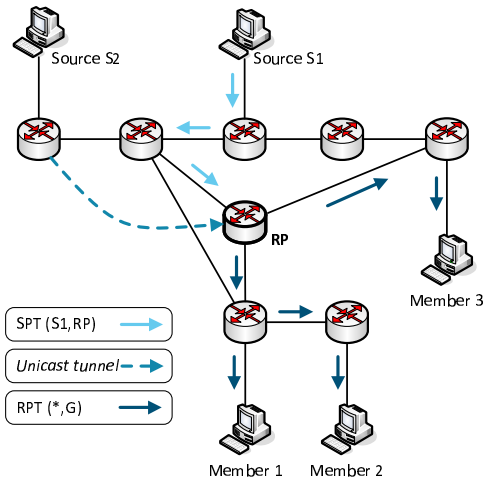

Figure 2.9: Example of a core based tree

multiple sources, it is represented by the tuple (∗,G), where ∗ represents all possible sources for the multicast group with the multicats address G. The sources send the multicast data to the RP of the tree that, in turn, sends the data towards the group members. Because the root is a central point in the network, known by the edge routers, CBT supports both unidirectional and bidirectional distribution trees.

A unidirectional CBT, as shown in Figure 2.9, is shared by the RP and the members of a group, meaning that the multicast data is transmitted only towards the members. The source of the group can transmit to the RP in one of two ways: by building an SPT (S,RP), or by using a unicast tunnel (IP-over-IP, for instance) between the source's edge router and the RP.

A bidirectional CBT, as shown in Figure 2.10, is shared between all sources and members of a group, meaning that the edge routers of the sources must also process IGMP/MLD messages, as do the edge routers of the members. This behaviour enables multicast traffic to flow in both directions of the distribution tree.

The use of CBT results in a lower mean network usage but presents higher latency than the use of SPT. CBT also presents a higher degree of traffic concentration near the source of the tree and is more subject to the creation of bottlenecks [83, 64].

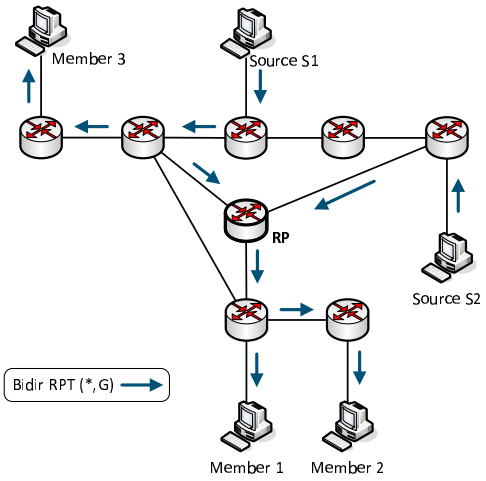

Figure 2.10: Example of a bidirectional core based tree

	ASM	SSM
Group identifier	(*,G)	(S,G)
Address designation	Group	Channel
Member action	Join/Leave	Subscribe/Unsubscribe
Address space	224.0.0.0/4 (except 232.0.0.0/8)	224.0.0.0/4 (reserved 232.0.0.0/8)

Table 2.1: ASM and SSM naming conventions

2.2.5 Multicast service models

The original IP multicast service model [32], the Any-Source Multicast (ASM) model, supports both many-to-many and one-to-many communications. In many-to-many communications, each group can have multiple sources sending data to multiple members. In one-to-many communications, each group has a single source that sends data to multiple members. The interested members need only to know the group multicast address.

The Source-Specific Multicast (SSM) model [46] supports only one-to-many communications, where each group, referred in [46] as channel, is identified both by the multicast group address and the unicast address of the source of the group. In SSM there can be only one source per multicast group or channel; the distribution trees adopted are the source trees. In [46] a new naming convention, summarized on Table 2.1, was also proposed.

ASM uses shared trees, which requires additional protocols to discover

sources of a multicast group, as well as RPs. The protocol used for that purpose is Multicast Source Discovery Protocol (MSDP) [39]. The ASM multicast optimization is compromised if multicast addresses collide, in particular when other sources use the same multicast group address [34].

SSM, by using the source unicast address as part of the group identifier, presents several advantages when compared to ASM, namely:

- **Simpler multicast addressing**: Multicast group addresses can be shared by multiple groups without the possibility of collisions; it also provides a higher number of possibilities for the multicast group identifier.
- **Simpler multicast routing**: The group members explicitly specify the source address in their join requests, which simplifies the join process and no longer requires group source discovery protocols.
- **Simpler network structures**: An RP is no longer required thus simplifying multicast deployment and network management.
- **Better security**: Denial of Service (DoS) attacks to group sources are not feasible, since it requires the spoofing of the unicast source address in the group communication packets, which will be discarded by routers running RPF verifications.

When compared to ASM, SSM presents also some disadvantages. SSM requires more state information per group; routers in ASM maintain state information on a mulitcast group address basis, while SSM requires such state information per each pair (S,G). Moreover, SSM is only supported in the 3^{rd} version of IGMP [22], which is not as widely available as previous versions, namely IGMPv2 [40].

2.2.6 Multicast routing

Multicast groups may be global or localized, and their sizes are a priori unknown. In order to enable this type of communications, routing mechanisms are required. Multicast routing protocols rely on protocols such as the multicast group management protocols or IP routing protocols such as Routing Information Protocol (RIP) [57] or Open Shortest Path First Routing Protocol (OSPF) [63]. Some well known routing protocols are: the

Distance Vector Multicast Routing Protocol (DVMRP) [81], the Multicast Open Shortest Path First (MOSPF) [62, 61], and the Protocol Independent Multicast (PIM). The latter can be deployed in two modes, the dense mode (Protocol Independent Multicast - Dense Mode (PIM-DM)) [12], and the sparse mode (Protocol Independent Multicast - Sparse Mode (PIM-SM)) [37].

The protocol independency claimed by PIM refers to IP routing protocols; any unicast routing protocol capable of building unicast routing tables can be used by PIM. Unlike other multicast routing protocols, PIM does not build a routing table for multicast traffic, but it relies on unicast routing tables, and uses the RPF verification to determine whether the multicast traffic should be forwarded or not. Unlike other multicast routing protocols such as DVMRP, PIM does not send or receive multicast routing updates.

PIM-DM

The dense operational mode of the PIM protocol consists in a router forwarding multicast traffic through its connected links, so that the traffic reaches all the network elements. The downstream routers having no interested receivers must prune unwanted traffic. If a prune message is received from a neighbor downstream router, the multicast traffic is stopped in the link through which the prune message arrived. Multicast routing state information is thus based on the prune messages received by routers.

PIM-SM

The PIM-SM works in opposition to the PIM-DM, by not assuming receivers in every network element. It requires that receivers explicitly request multicast data in order to start sending it. PIM-SM uses the RP concept. All senders and receivers must register at the RP, in order to send or receive multicast traffic. RP reside on routers. PIM-SM is particularly adequate to scenarios where receivers are scattered over geographically distant networks, since PIM-SM sends multicast data only to the network regions requesting the multicast traffic.

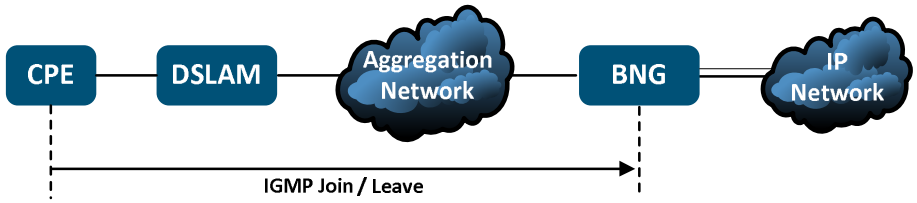

Figure 2.11: xDSL network architecture

2.3 Multicast in Access Networks

Multicast support plays an important role in NGN, where applications for video broadcast may benefit from using IP multicast. The optimization achieved by the IP multicast techniques may be compromised by access network technologies, in particular when access networks do not support link layer multicast transmissions.

2.3.1 xDSL

The Digital Subscriber Line (xDSL) architecture [29] and respective network elements are shown in Figure 2.11. The Broadband Network Gateway (BNG) is the IGMP router; its roles are to receive and process IGMP messages and to forward multicast packets. The BNG is also the Network Access Server (NAS) where users authenticate themselves during network attachment.

In xDSL networks, connections use the Point-to-Point Protocol over Ethernet (PPPoE) [58], being the connection endpoints the Customer Premises Equipment (CPE) and the BNG. These connections are required for user authentication and authorization during network attachment. The point-to-point nature of these connections implies that the multicast packet replication is performed at the BNG. Although this provides a central point for multicast management, it leads to data redundancy in the aggregation network. Multicast packets are replicated on a per PPPoE connection basis, even if the path through the aggregation network is shared by two or more group members. This effectively nullifies the bandwidth savings offered by multicast.

Optimized multicast in xDSL, where all network elements would perform multicast packet replication, requires a migration to Ethernet aggre-

gation and the use of Internet Protocol over Ethernet (IPoE) for multicast data. Additionally, layer 2 equipment such as Digital Subscriber Line Access Multiplexer (DSLAM) and switches, should either perform IGMP snooping [28] or function as IGMP proxies [38]. Considering that PPPoE still continues to be used for authentication purposes, but optimized xDSL multicast requires the establishment of two network connections: one for typical Internet access (PPPoE), and another for multicast services (IPoE). IGMP messages can be sent through both connections or just through the IPoE connection. When sent through both, the BNG can monitor individual members by correlating IGMP messages with the PPPoE connection from which they were received. When IGMP messages are only sent through the IPoE connection, the BNG may be able to track individual members depending on whether the DSLAM performs IGMP snooping or behaves as an IGMP proxy. IGMP snooping at the DSLAM enables the BNG to identify individual members based on the IP or MAC addresses. IGMP proxy, however, prevents BNG from identifying individual group members since the DSLAM, which in this case generates the IGMP messages, would act as an IGMP router for users and as an IGMP client for the BNG.

2.3.2 WiMAX

Worldwide Interoperability for Microwave Access (WiMAX) network architecture [84] is depicted in Figure 2.12. Before packets can be transmitted, an IEEE 802.16 transport connection must be created between a Base Station (BS) and a Subscriber Station (SS). These connections are identified by a 16-bit Connection Identifier (CID) number, and by a layer 2 tunnel between the BS and Access Service Network - Gateway (ASN-GW). In WiMAX the role of IGMP router falls upon the ASN-GW network element, which is also responsible for client AAA [31].

Upstream connections (SS to ASN-GW) are exclusively point-to-point. Downstream connections can be used to transmit data to a group of SSs (under the same BS), using multicast CIDs (mCIDs). Multicast CIDs are therefore suited for IP Multicast data transmission. This solution requires the establishment and management of mCIDs and their associations with IP multicast-based services. These management mechanisms and related protocols are still under development by the WiMAX Forums Networking Group. There are some unresolved issues with the use of mCID [76, 51],

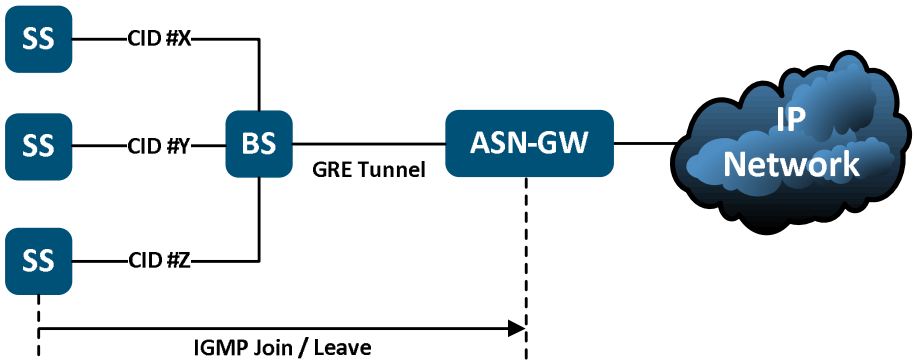

Figure 2.12: WiMAX network architecture

namely the reduced transmission efficiency of mCIDs for small multicast groups, the missing support for unidirectional broadcast channels, and the existence of additional security threats associated with broadcast channels in a power-conservative wireless system.

2.3.3 UMTS

Universal Mobile Telecommunication System (UMTS) networks, since Release 99, have support for IP multicast. The IGMP router role is performed by the Gateway GPRS Support Node (GGSN) [9]. IP multicast packet transmission inside the UMTS network is performed over point-to-point tunnels (from the GGSN to the User Equipment (UE)), thus no sharing gains are achieved. In Release 6, the Multimedia Broadcast/Multicast Service (MBMS) [10] was introduced with the purpose of supporting native multicast transport connections within the UMTS network (see Figure 2.13).

MBMS adds a new network element to the UMTS network, the Broadcast/Multicast - Service Center (BM-SC), which is the central point for MBMS management decisions. Its functions include MBMS multicast session announcements, user authentication and authorization, and signaling. In order to support MBMS services all UMTS network elements require additional functionality.

MBMS multicast data distribution is designed only for downstream connections (from the BM-SC to the UE); any upstream multicast traffic must go to the GGSN and then be forwarded to the intended recipients. Multicast

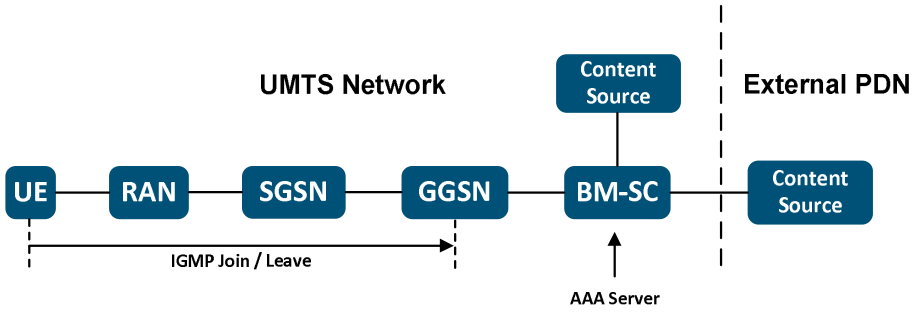

Figure 2.13: UMTS/MBMS network architecture

group joining and leaving is carried out through IGMP messages and multicast groups are represented by IPv4 multicast addresses. MBMS is designed for IP multicast interoperability; however, the interface that connects the BM-SC to external Packet Data Networks (PDN) is not yet specified in the latest 3rd Generation Partnership Project (3GPP) release [10]. Therefore, MBMS services are limited to a single UMTS network.

2.4 Summary

Video transmission over IP consists of a video being played out while other video parts are being received and decoded, thus avoiding a full video download before visualization. Current efforts to standardize video streaming over IP include functionalities required at network, transport, and session layers. The IETF multimedia architecture has defined, in particular, RTP [73] which enables the transmission of video, voice and multimedia contents in IP packets, along with other protocols for controlling the video streaming. More recently [7, 8], these protocols have been re-used by organizations such as the ITU-T and ETSI to integrate IPTV services in the NGN architecture, defined by TISPAN. Key issues of these ETSI and ITU-T activities are the mobile-fixed service convergence and the optimized transmission of video streams over heterogeneous access networks, namely xDSL, WiMAX and UMTS.

IP multicast is of particular appeal for IPTV services, since it enables significant savings in terms of network resources by only transmitting once for all active receivers. A host wanting to belong to a multicast group must send a message to inform the local multicast router about its interest. A host

wanting to create a multicast group starts transmitting the data packets to the network, using an IP multicast address as the destination address of these data packets. The multicast router is the network element responsible for both multicast group management, based on IGMP/MLD messages, and multicast packet forwarding, based on multicast routing protocols, such as PIM.

Since IP multicast is lower layer agnostic, the optimization it achieves may be compromised by access network technologies when such access networks do not support link layer multicast transmissions. This problem led us to devote our attention to multicast support in the access networks considered by the NGN architecture, namely xDSL, WiMAX and UMTS. In particular, the components of each network architecture that are responsible for IGMP/MLD processing were identified. Optimized multicast in xDSL requires a migration to Ethernet aggregation and the use of IPoE for multicast data. In WiMAX, there are some unresolved issues with the use of mCID [76, 51] that restrict optimized multicast transmissions. MBMS added the BM-SC to UMTS networks in order to support multicast transmissions; the BM-SC is designed only for downstream connections and the interface that connects the BM-SC to external networks is not yet specified.

While IP multicast supports sources placed anywhere on the network, enabling users to be sources to their own content, the lower layer support for multicast sourced at the user is not common. xDSL access networks with Ethernet aggregation enable upstream multicast traffic. In WiMAX, upstream multicast traffic is not possible because upstream connections (SS to ASN-GW) are exclusively point-to-point, requiring that such traffic goes to the ASN-GW and then forwarded to the intended receivers. The same happens in UMTS, where the upstream multicast traffic must go to the GGSN in order to be forwarded.

Chapter 3

Admission Control and Security in Multicast

The deployment of IP multicast-based services has not been as widely exploited as expected. Some operators already offer multicast-based IPTV services to their customers, but customers are still unable to be the source of multicast trees, and more dynamic multicast services are not offered. One of the reasons behind this limited adoption is the lack of control that network operators have over multicast groups [34]. IP multicast has an open group architecture, where any user is free to receive or transmit data from/to a multicast group. Although this grants high scalability to multicast based services, this openness raises problems for network operators such as access control and traffic accounting. From the operator's point of view, AAA multicast capabilities are essential and they must be associated to IP multicast-based services, so that functions such as accounting, billing or regular network management may take place.

Some degree of control over IP multicast groups can be achieved with end-to-end encryption of IP multicast data or IP multicast session access control [53]. While the first solution protects multicast data from unauthorized access and potential eavesdroppers, it does not prevent a user from joining a group to which he does not have access to, thus causing the unnecessary extension of the multicast distribution tree. This multicast tree extension results in wasted bandwidth and wasted computer power. IP multicast session access control, on the other hand, enables network operators to manage IP multicast at network level by performing access control at the

A, B, C, D	Communicating nodes
K_{ab}	Symmetric pre-shared key between communicating nodes
N_a	Nonce generated by A
$H(M)$	Hash function of M
$\{M\}_K$	M encrypted with key K
SEK_i	Current session encryption key of entity A
$X.Y$	Field X concatenated with field Y
$APriK$	Private key of entity A
$APubK$	Public key of entity A
Ts	Time stamp

Table 3.1: Adopted notation

edge-router. The edge-router is the node where IGMP messages are processed, so operators are able to identify group access requests and multicast transmissions; this functionality also enables network operators to specify the multicast streams each user can receive data from or send data to.

This chapter is composed of three sections. Section 3.1 describes the adopted notation. Section 3.2 addresses multicast admission control. Section 3.3 describes existing work related to secure multicast.

3.1 Adopted notation

Table 3.1 presents the notation adopted throughout this work. Capital letters such as A, B, C and D represent communication nodes. K_{ab} represents a symmetric key previously shared between the nodes A and B. N_a represents a nonce generated by node A. $H(M)$ represents the output of a hash function of input data M. $\{M\}_K$ represents an M message encrypted with the key K. SEK_i represents the current Session Encryption Key (SEK) of a communicating node. $X.Y$ represents field X concatenated with field Y. $APriK$ represents the private key of entity A. $APubK$ represents the public key of entity A. Ts represents a time stamp.

3.2 Multicast admission control

Work related to IP multicast AAA is being carried out within the IETF MBONE Workgroup. In [44] the requirements for multicast AAA were specified, and in [72] a general multicast AAA framework is being designed to

Sequence	Entities	Messages
1	$Host \rightarrow Router$	IGMP/MLD Join
2	$Router \rightarrow MCA$	Validate:Group_Address.Host_Address
3	$MCA \rightarrow Router$	Result:Group_Address.Host_Address

Table 3.2: MCOP protocol for receiver access control

meet these requirements.

Research proposals regarding AAA in IP multicast typically follow one of two approaches: the introduction of an additional control layer, or the modification of IGMP/MLD signaling. The first approach consists on introducing an intermediate control layer between IP and IGMP/MLD processing. The second approach requires the modification of the group management protocols (IGMP/MLD) in order to carry user authentication information.

3.2.1 Additional control layer approach

MCOP

In [56], the authors propose a new communication protocol, the Multicast Control Protocol (MCOP), used to exchange messages between the edge router and the Multicast Controlling Agent (MCA). The MCA is responsible for multicast session access validation and it uses IP addresses contained in the IP/IGMP packets. No protocol modifications, such as IGMP modifications, are required.

Table 3.2 details the message exchange for a receiver access control operation. A host willing to participate in a group, sends an IGMP join message to access the requested group. The designated router, triggered by the join request, sends an authorization request to the MCA. Upon a successful validation by the MCA, the router will process the join request and extend the distribution tree.

NetWrapper

In [54] the authors suggest a portal-based system where a user, in order to receive a multicast stream, would authenticate himself on a web portal and then, after a successful authentication, an entity called NetWrapper would configure the edge device to enable multicast distribution. No mention is made on how IGMP messages fit in their scheme or how would the portal

Sequence	Entities	Messages
1	$Host \rightarrow Router$	IGMP.$\{ID.Ts.N_{Host}\}_{HostPriK}$
2	$Router \rightarrow GKDC$	$\{\{ID.Ts.N_{Host}\}_{HostPriK}.\{ID.Ts.N_{Router}\}_{RouterPriK}\}_{RouterPriK}$
3	$GKDC \rightarrow Router$	$\{\{ID.Ts.N_{Host}\}_{HostPriK}.\{ACL\}_{GKDCPriK}.\{Keys\}_{RouterPubK}\}_{GKDCPriK}$
4	$Router \rightarrow Host$	$\{ID.Ts.N_{Host}\}_{HostPriK}.\{ACL\}_{HostPubK}$

Table 3.3: SMKD Protocol for receiver access control

retrieve information regarding the edge device associated with the request.

3.2.2 Protocol modification approach

SMKD

Scalable Multicast Key Distribution (SMKD) [16] consists of a secure version of CBT that uses cryptographic techniques to protect the addition of routers to the distribution path, in order to impose receiver access control, and to perform group key distribution. In SMKD, each group has a Group Key Distribution Center (GKDC) that holds the group Access Control List (ACL) and distributes cryptographic keys to authorized routers and hosts. Table 3.3 details the message exchange for a receiver access control operation.

A host willing to participate in a group sends an IGMP join message, modified to include a digital signed token, to its designated router. The token contains the host identification, a timestamp and a nonce. In turn, the router verifies the token and initiates the group distribution tree extension by forwarding the hosts token to the GKDC. Upon successful verification, the GKDC sends back a signed ACL and group related cryptographic keys. At this moment, the router will store the group ACL and assume the GKDC functionality for future downstream group join requests.

Gothic

Gothic [52] proposed the use of Public Key Infrastructure (PKI) [1] in conjunction with IGMP and MLD message modification to include X.509 certificates. It introduces a new entity called Access Control Server (ACS) that is responsible for the authorization of host join requests. Table 3.4 details the message exchange for a receiver access control operation.

[1] A PKI provides certification functions, namely key generation, certificate generation, key distribution, certificate renewal, certificate distribution to end users and certificate revocation

Sequence	Entities	Messages
1	$Host \rightarrow ACS$	$\{Host\_X509.Group\_ID\}_{HostPriK}$
2	$ACS \rightarrow Host$	$\{Host\_IP.Group\_ID.Expire\_Time.ACS\_X509\}_{ACSPriK}$
3	$Host \rightarrow Router$	$\{Host\_IP.Group\_ID.Expire\_Time.ACS\_X509\}_{ACSPriK}$
4	$Router \rightarrow Host$	JoinACK

Table 3.4: Gothic protocol for receiver access control

Sequence	Entities	Messages
1	$GC \rightarrow Host$	$\{PriKGroup\}_K.\{PubK_G\_GCBA\_Address\}_{GCPubK}$
2	$Host \rightarrow Router$	$MLD\,(\{G\_CBA\_Address.GroupPubK\}_{GroupPriK})$

Table 3.5: G-CBA protocol for receiver access control

A host willing to participate in a group, firstly sends an access request to the ACS that comprises the host certificate and the IP address of the group, both signed with the host's private key. The ACS, upon successful validation of the host's request, replies with a message that contains a set of fields, signed with private key of the ACS that will be used as access credentials. The set of fields comprises the host IP address, the group IP address, an expiration time for the credentials, and the ACS certificate. The host, when in possession of the access credentials, will send an IGMP/MLD join message, including the access credentials. The router, upon successful verification of the host's access credentials, must reply with a join acknowledgment message.

G-CBA

Group Cryptographically Based Address (G-CBA) [25] proposed a receiver access control mechanism for IPv6 multicast groups where a public-private key pair is associated to the IPv6 address of each equipment. A Cryptographically Based Address (CBA) is then derived by the Group Controller (GC) for each group. Such CBA derivation is based on applying a one-way hash function over the public-key of the group, resulting in a 64 bit suffix. The 64 bit suffix of the group is then concatenated with the 64 bit network prefix to obtain the IPv6 address of the group. Table 3.5 details the message exchange for a receiver access control operation.

The GC initially generates a private-public key pair for the group and

derives the corresponding G-CBA; then securely [2] transmits the key pair to group members by sending a message that comprises the group's private key encrypted with K, and the group's public key and G-CBA encrypted with GC's public key. A host willing to participate in a group, sends a modified MLD join message to its designated router. The modified MLD message comprises the G-CBA concatenated with the group's public key, digitally signed with the group's private key. The designated router verifies if the G-CBA was generated from the group public key and if the signature is valid. Upon successful verification, the router accepts the MLD message.

IGMP-AC

In [49] the authors propose a framework to add AAA capabilities to standard IP multicast by modifying IGMPv3 and, in [50], they introduced Internet Group Management Protocol with Access Control (IGMP-AC) to support multicast access control. IGMP-AC consists in using Extensible Authentication Protocol (EAP) [11] combined with IP Security (IPSec) [55] and in the modification of IGMPv3 messages to impose multicast group access control for both senders and receivers. The modification of IGMPv3 messages consists in adding user authentication data. In both [49] and [50], the modification of the IGMPv3 messages is not specified.

MCDA2

The Multicast Content Distribution Architecture with Accounting support (MCDA2) detailed in [45] makes use of a previous IGMPv2 modification proposal [48]. These solutions follow the general AAA architecture as defined in [59] but they are applied to multicast sessions. The NAS, upon receiving a IGMP join request, uses the authentication information contained in the IGMP packet to send an authorization request to an AAA server in order to verify if the user has access to the intended IP multicast group.

3.2.3 Summary on multicast admission control

Table 3.6 compares the multicast admission control techniques identified in this section. The comparison criteria are the following: demand for

[2] The G-CBA protocol does not specify how the key K is securely exchanged

Name	IGMP/MLD Modification	Uses Crypto. Tecniques	Uses PKI	Sender Control	Receiver Control	Tree node Control	AAA Protocol	IP Version
SMKD	Y	Y	N	N	Y	Y	N (SMKD)	IPv4
Gothic	Y	Y	Y	N	Y	N	N (Gothic)	IPv4/IPv6
G-CBA	Y	Y	N	N	Y	N	N (G-CBA)	IPv6
IGMP-AC	Y	Y	N	Y	Y	N	Y (DIAMEETER)	IPv4/IPv6
MCDA2	Y	N	N	Y	N	N	N (MCDA2)	IPv4/IPv6
MCOP	N	N	N	Y	Y	N	N (MCOP)	IPv4

Table 3.6: Comparison of Multicast admission control techniques

group management protocol modification (1^{st} column); usage of cryptographic techniques (2^{nd} column); PKI requirement (3^{rd} column); sender access control (4^{th} column); receiver access control (5^{th} column); access control to nodes of the multicast distribution tree (6^{th} column); usage of standard protocol for AAA (7^{th} column); support for both IPv4 and IPv6 (last column).

For instance, the MCOP solution does not require modifications to the IGMP messages, does not use cryptographic techniques nor PKI, imposes both sender and receiver access control, does not impose access control to routers that take part in the multicast distribution trees, proposes the MCOP protocol for AAA, and addresses only the IPv4 protocol. The MCOP solution is also the unique solution that does not require group management protocol modification nor the use of encryption techniques. Nevertheless MCOP addresses only IPv4 networks and uses a non standard protocol for authorization.

3.3 Secure Multicast

Secure multicast is a group transmission technique that enforces confidentiality. It uses cryptographic techniques to encrypt data and, by doing that, this type of techniques also enforces access control. A secure multicast architecture needs to consider the size of the groups, group memberships, and security contexts such as encryption keys. Architectures such as those defined in [78] are efficient for small groups, while the architecture defined by the Multicast Security (MSEC) group [3] is being developed for large groups [43, 18].

In its simplest scheme, the source of the group sends data to an IP multicast address; a receiver interested in the data signals its interest to its local multicast router using an IGMP or an MLD join message. Access

control is imposed by encrypting the data prior to its transmission, and by sending the decryption key to the authorized receivers. Group confidentiality is achieved by changing the decryption key, and by transmitting it securely to the authorized receivers. Decryption keys can be transmitted regularly, upon a group change, or using a combination of both methods. Group changes occur either by the departure of a member (group leave) or by the arrival of a new member (group join). The operations of key renewal are referred to in the literature as re-key operations, and are managed by an entity called Group Controller (GC).

Several types of cryptographic keys are used in secure group communication architectures. The common types are Key Encryption Keys (KEK), and Data Encryption Keys (DEK). KEK is a key assigned to a member and it is known only by that member and the GC; KEK is used to secure communications between each member and its GC. DEK is a key used to encrypt the group communications data and must be known by all the members of the group. In a re-key operation, for instance, DEK may be securely transmitted by the GC to valid members in a message consisting of $[\{DEK\}_{KEK_1}, \{DEK\}_{KEK_2}, ..., \{DEK\}_{KEK_n}]$. In this example, the notation $\{DEK\}_{KEK_1}$ means that DEK is encrypted with KEK of the first member, and n represents the number of receivers in the group.

Confidentiality requirements can be classified in four classes [26]: 1) non-group confidentiality; 2) forward secrecy; 3) backward secrecy; 4) collusion resistance. The first class imposes that users that had never participated in the group should not access any cryptographic material. The second class imposes that a member departing from a group should stop receiving cryptographic material, therefore ensuring that this member is unable to decrypt group communications after leaving the group. The third class imposes that a receiver arriving to the group should not access previous cryptographic material, ensuring that this member is unable to decrypt past group communications. The last class imposes that current cryptographic material should not be inferable by non-members.

In [69] three approaches for key distribution were identified: centralized, distributed, and decentralized. More recently, Cao et al. [24] extended this classification and identified four schemes: simplest scheme, centralized scheme, decentralized scheme, and hierarchical scheme. The first scheme is a subset of the centralized approach; the centralized and the decentralized

schemes are the centralized and the decentralized approaches respectively.

In this work we adopt a classification that combines both classifications and comprises 4 types of key distribution: centralized, decentralized, distributed and hierarchical. The distributed type assumes that every member can participate in the key distribution, perform access control, and contribute to the generation of the group key. The group controller role is not usually present because the group keys are generated with contributions from all members. Group Diffie-Hellman Key Exchange [77] is an example of the distributed type.

The hierarchical type assumes that users have not the same priorities and impose cryptographic access control to classes of users with different access levels. This type was firstly addressed in [13] where a hierarchical key assignment to users was adopted. A user belonging to a certain class can derive the cryptographic keys of lower class users. The hierarchical type presents the drawback of requiring extra computational power from the members, similarly to the distributed type.

In the decentralized type the group is split into subgroups, each having its manager. The subgroup manager generates the local encryption key and processes the local membership changes (subgroup member join/leave operations). Iolus [60], DEP [36], MARKS [21], IGKMP [42], Kronos [75] and Multicast Deflector [67] are examples of decentralized key distributions.

The centralized type is characterized by the existence of a unique entity that manages the entire group. The Group Controller (GC) encrypts the DEK using each member's key (KEK_i) and then it transmits the n keys to the group members. Despite its simplicity, this scheme suffers from the single point of failure problem; in case of failure of the GC, the cryptographic material is not renewed and the new members become unable to receive the cryptographic material required to decrypt the data. Logical Key Hierarchy (LKH) [85], One-way Function Tree (OFT) [82], One-way Function Chain Tree (OFCT) [23], Efficient Large-group Key (ELK) [65], LKH++ [66], and SMIz [68] are examples of centralized key distributions.

3.3.1 Decentralized type

In the decentralized type, the group is split into subgroups, each having its manager. The subgroup manager generates the local encryption key and processes the local membership changes (subgroup member join/leave

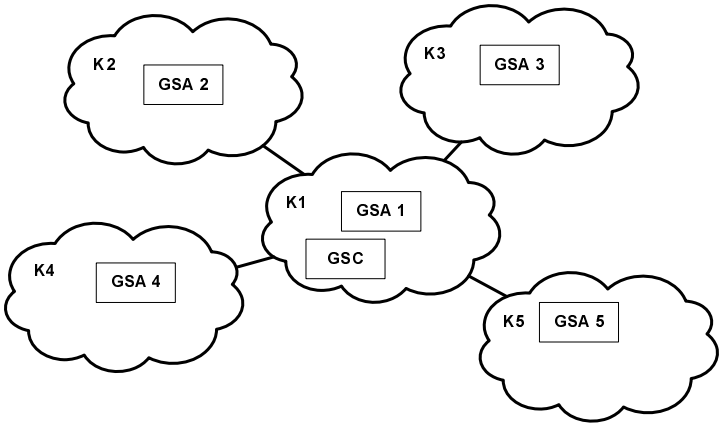

Figure 3.1: Iolus framework

operations).

IOLUS

Iolus [60] is a decentralized scheme where each subgroup is managed by a Group Security Agent (GSA). GSAs are hierarchically organized and form a top-level group managed by the Group Security Controller (GSC), as shown in Figure 3.1. Iolus requires that the GSC trusts the GSAs. In Iolus, the keys used by each subgroup are independent, which enables membership changes to be treated locally. Key independence implies that the GSA must be in the data path so that it can decrypt the original data and re-encrypt it using its local subgroup key. In case of failure of a GSA, only the subgroup it manages is affected. Despite the scalability advantage when compared to the centralized type, the Iolus' GSAs is itself a bottleneck because it decrypts and encrypts all the data for the group it manages.

DEP

Dondeti et al. proposed DEP [36]. Here, the element responsible for subgroup management is called Subgroup Manager (SGM) and is similar to the Iolus' GSA. The scenario adopted considers the deployment of SGMs in third party equipment such as Service Provider (SP) routers, which requires data and key distribution schemes that avoid group data disclosure to these SGMs. DEP classifies SGMs as member and participant. The

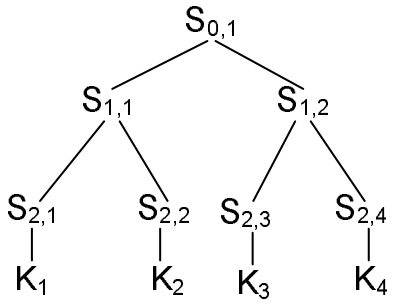

Figure 3.2: MARKS Binary hash tree

member SGMs are entitled to access both DEP and group data, where the participant SGMs are not. DEP uses one DEK and two types of KEK: 1) KEKs known by the GC, senders and members; 2) Local Subgroup keys (LSs), known by members and participant SGMs. The DEK, required by all members in order to decrypt the group communications, is sent by the GC to the SGMs in the form $\{DEK\}_{KEK_i}$. In turn, the participant SGM encrypts the encrypted DEK with its LS_i, and transmits $\{\{DEK\}_{KEK_i}\}_{LS_i}$ to its subgroup members. Every member is now able to decrypt the double encrypted DEK. Despite involving trusted third parties, this solution does not enable participant SGMs to access DEK but it still enables them to process subgroup membership changes by refreshing their LS_i. This solution has two drawbacks: it requires extra encryption/decryption operations, and the leaving members, in possession of the current DEK, will be able to access the group data until the DEK is refreshed.

MARKS

In contrast with the session oriented group keys, MARKS [21] associates seeds to time slices of group communications, each time slice corresponding to a key. The seeds form a binary hash tree, where leaves correspond to time slices; the seeds in the path to the root are required in order to generate the time slice cryptographic key. Figure 3.2 shows a scenario with four time slices and their cryptographic keys (K_1, K_2, K_3, K_4). If a user wants to access the first time slice, he will need seed $S_{2,1}$; if he also wants to access time slices 3 and 4, he will need to obtain seed $S_{1,2}$. In the possession of seed $S_{1,2}$, every member is able to generate seeds $S_{2,3}$ and $S_{2,4}$.

IGKMP

The Intra-domain Group Key Management Protocol (IGKMP) [42] assumes 2 main entities: the Area Key Distributor (AKD), and the Domain Key Distributor (DKD). Each AKD manages an area subgroup. The DKD is responsible for the group key generation and their transmission to AKDs that, in turn, send the group key to their area members. All the key distributors form a multicast group that is used in key refresh operations. The group key is the same in all areas, therefore no decryption or re-encryption are required in group communications from one area to the other.

KRONOS

Kronos [75] was proposed by Setia et al. and it focuses in periodic batch key refreshing, neglecting membership changes during each period. A new group key is generated and distributed after a period of time without considering inner period member join and leave operations. There are subgroup managers (AKDs), which can independently generate the same group key and do not require its transmission from the DKD. In order to generate the same group key, the AKDs need a synchronized clock and have to agree in two secret factors, which are used to derive the first key. The following keys are derived by encrypting the current key with the master key, which is one of the two secret factors. Because all the keys are derived from the current key, security may be compromised if one key is disclosed. This decentralized approach requires mechanisms such as clock synchronization and conflict resolution.

Summary of Decentralized type

The evaluation of the efficiency of solutions of this type can be made based on factors such as key independence, local re-key, and re-key per membership. Additional attributes that can be used include the type of communication and the use of decentralized group controllers. Key independence is related to the past distributed keys not being compromised upon a key disclosure and thus protecting previous group communications from access by a later joined element that has saved past group communications. Local re-key is related to the number of elements affected by a group size change; that is, if there is a departure from the group, it should be a local departure and

	Key independence	Local rekey	Rekey per membership	Communication type	Decentralized Management
IOLUS	Y	Y	Y	1-to-n	Y
DEP	Y	N	N	m-to-n, 1-to-n	N
MARKS	N	N	N	m-to-n, 1-to-n	Y
IGKMP	Y	N	Y	m-to-n, 1-to-n	Y
KRONOS	N	N	N	m-to-n, 1-to-n	Y

Table 3.7: Comparison of decentralized approaches

affect only the elements within the same local subgroup, avoiding scalability problems such as the "1-affects-n" problem. Re-key per membership is related to backward and forward secrecy; in other words, if when a member joins a group it gets access to a key that will allow it to decrypt messages sent previously, then there is no backward secrecy; if after a member group leave operation it still can decrypt group messages, then there is no forward secrecy. The group communications are mainly of two types: groups having only one data source, and groups having multiple data sources. The use of decentralized controllers improves the availability of the service; the failure of a central group controller affects all group communications, and induces failures similar to those caused by the group key management protocols described earlier.

Table 3.7 compares the decentralized architectures described above. Kronos and MARKS do not have key independence, and future cryptographic keys can be derived from current key; in Kronos the new key is generated from old keys, and in MARKS new keys are generated from seeds that, if compromised, imply the failure of future secrecy. Iolus is the only one supporting local rekey procedures. Its rekey operation involves only the members of a subgroup, and does not affect the other subgroups. Local rekey enables a certain degree of scalability and bypasses the "1-affects-n" problem, but it has visible implications in the data path. Translations are required in communications from one subgroup to another. Past and future secrecy can be achieved by doing a rekey when a group membership changes. DEP, for instance, adopted a timed rekey that rekeys the group independently of membership changes but periodically, it enables periods of time during which the leaving members still can access the group data, that is, there is no future secrecy in that time period. In DEP there exists a central group controller entity that must be contacted on every group join

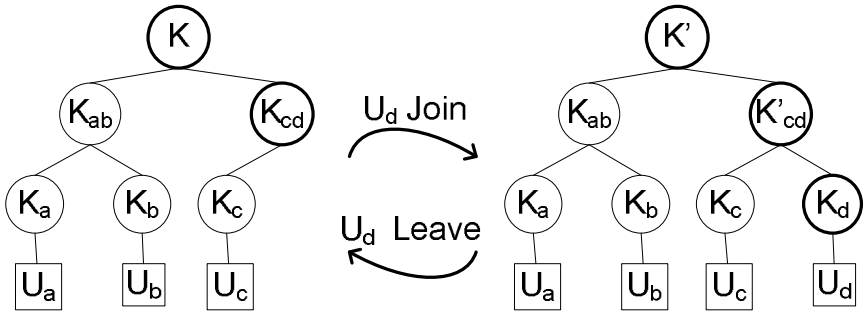

Figure 3.3: KEKs affected by member U_d join/leave.

authentication, despite the existence of subgroups and subgroup managers, which creates a single point of failure.

3.3.2 Centralized type

The centralized type assumes that a member has no knowledge of other group members and that the access control is performed by a unique entity.

Logical Key Hierarchy (LKH)

In order to address problems such as key storage space and the support of highly dynamic groups, Wong et al. [85] proposed the use of a Logical Key Hierarchy (LKH). In LKH, the GC stores the keys in the form of a balanced tree of keys whose leafs are the individual member KEKs and the intermediate nodes represent other KEKs required by the members. An example of such a tree is shown in Figure 3.3. The root of the tree holds the group DEK (K). When a new member joins the tree, it is added as a leaf to the tree and all the keys in the path from its parent node to the root are changed. These keys will then be used by the new member to obtain the group key, i.e. the root of the tree. The groups with high rates of member departure and arrival can be supported by using these trees, since only the affected keys are refreshed.

Upon a group change, the DEK K must be refreshed in order to maintain forward and backward secrecy. For instance, the join operation of the member U_d, shown in the Figure 3.3, requires several encryptions. The key K' becomes the new root DEK and it must be sent to all members. For that

purpose, two messages are generated and sent: the $\{K'\}_{K_{ab}}$ is sent to users U_a and U_b, and the message $\{K'\}_{K'_{cd}}$ is sent to users U_c and U_d. The key K'_{cd} must also be sent by the GC to the members U_c and U_d, by sending the messages $\{K'_{cd}\}_{K_c}$ and $\{K'_{cd}\}_{K_d}$, respectively.

This method generates a large number of re-keying messages. For a group of n users with a tree of height h, the total number of keys that need to be maintained by all elements is $2n - 1$; the number of keys stored by each user is equal to its distance to the root of the tree ($h + 1$ keys). Upon a join operation $(2h - 1) + (h + 1)$ keys must be refreshed; upon a leave operation $2h$ keys must be refreshed.

One-way Function Tree (OFT)

The solution proposed by Waldvogel et al [82] is similar to LKH, differing only in the join operations. Instead of generating and sending new keys, the solution makes use of one-way functions over the keys that must be changed. If a receiver knows of the current keys, it will be able to generate the new keys. This algorithm is also referred to in the literature as LKH+.

Upon a group change, and in order to maintain forward and backward secrecy, each member must calculate the new key for each node in the path from its parent's node to the root. This strategy reduces the number of re-keying messages to half, but it substitutes the message cost by a computational cost.

For a group of n users with a tree of height h, the total number of keys that need to be maintained by all elements is $2n - 1$; the number of keys stored by each user is $h + 1$. Upon a join operation $2(h + 1)$ keys must be refreshed; upon a leave operation $h + 1$ keys must be refreshed.

One-way Function Chain Tree (OFCT)

Canneti et al. [23] proposed another variation of OFT that consists in using pseudo-random number generators instead of one-way functions; these generators are used to derive new KEKs from the current ones, and they are used only in group leave operations. Let us assume two functions, $H(x)$ and $L(x)$, which are related. $H(x)$ generates a random number that is then used by L() to generate a new KEK, that is $L(H(x))$. For instance, when considering the leave operation of member U_d, shown in Figure 3.3, the

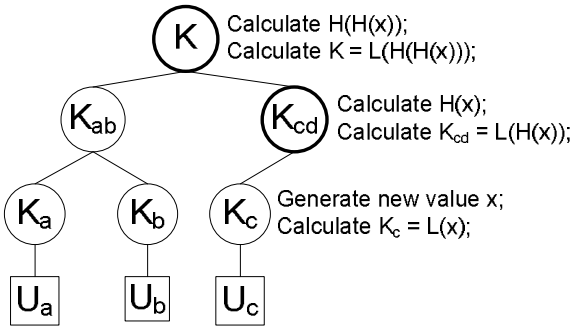

Figure 3.4: Computation required by OFCT upon member U_d leave.

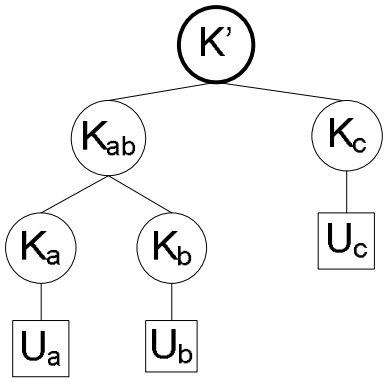

Figure 3.5: ELK key tree, rearranged upon member U_d leave.

GC sends a new value x to the member U_c; U_c then calculates $K_c = L(x)$. Moreover U_c will also derive the other keys up to the root by calculating $K_{cd} = L(H(x))$ and $K = L(H(H(x)))$. These computations are shown in Figure 3.4. OFCT requires less network resources at the expense of a higher computational cost.

For a group of n users with a tree of height h, the total number of keys that need to be maintained by all elements is $2n - 1$; the number of keys stored by each user is $h + 1$. Upon a join operation $h + 1$ keys must be refreshed; upon a leave operation $h + 1$ keys must be refreshed.

Efficient Large-group Key (ELK)

ELK [65] proposed another variant of OFT that uses Pseudo Random Functions. ELK addresses large groups and it enables the group members to

update all the keys either upon group membership changes or periodically.

Each group member generates the key of each tree node based on contributions from the left and right child keys. Upon a member leave operation, the tree requires rearranging. Assuming the leave operation shown in Figure 3.3, the resulting tree would be similar to the tree shown in Figure 3.5, where K_{cd} is eliminated and a new key K' is generated from K_{ab} and K_c; in order to do it, the GC calculates the left and right child node contributions of K' and sends the left contribution to U_c, and the right contribution to users U_a and U_b. A second property of ELK consists in allowing members to generate new keys using hints that are appended to data packets. For a group of n users with a tree of height h, the total number of keys that need to be maintained by all elements is $2n - 1$; the number of keys stored by each user is $h+1$. Upon a join operation $h+1$ keys must be refreshed; upon a leave operation h keys must be refreshed.

LKH++

LKH++ was proposed in [66] and it exploits one-way hash functions in combination with information already shared by the users, namely the keys belonging to the common tree nodes in the path from the users to the root. Considering the scenario of Figure 3.3, U_a and U_b share the keys K_{ab} and K, for instance. These shared keys are passed through one-way hash functions in order to generate the new keys. In particular, upon a user leave operation, the users that share some part of the tree with the leaving user may autonomously generate the new keys in the path toward the root, thus reducing the number of re-keying messages generated by the GC.

For a group of n users with a tree of height h, the total number of keys that need to be maintained by all elements is $2n - 1$; the number of keys stored by each user is $h + 1$. Upon a join operation $h + 1$ keys must be refreshed; upon a leave operation $h + 1$ keys must be refreshed.

Summary of Centralized type

Table 3.8 compares the centralized types identified in this thesis. In this table, n represents the number of members in the group and h represents the height of the tree used to maintain the keys in the GC. The 1^{st} column (GC) shows the number of keys maintained by the Group Controller. The 2^{nd}

	Number of Keys		Re-key message size	
	GC	Member	Join	Leave
Simplest	n	1	n	n
LKH	$2n-1$	$h+1$	$(2h-1)+(h+1)$	$2h$
LKH++	$2n-1$	$h+1$	$h+1$	$h+1$
ELK	$2n-1$	$h+1$	$h+1$	h
OFT	$2n-1$	$h+1$	$2(h+1)$	$h+1$
OFCT	$2n-1$	$h+1$	$h+1$	$h+1$

Table 3.8: Centralized approaches comparison

column (Member) shows the number of keys required by each member. The 3^{rd} and 4^{th} (Join and Leave) columns show the message size, in numbers of keys, that must be transmitted upon group memberships changes in order to preserve secrecy. For the *simplest* approach, the group controller maintains one key per each member in the group; each member requires only one key, and both the group join and leave operations require the transmission of a re-key message with size n times the key length (one key per member). The remaining solutions in the table, when compared with the simplest approach, show reductions in the bandwidth required for re-key operations upon membership changes.

3.3.3 Distributed and Hierarchical types

The distributed type assumes that every member can participate in the key distribution, perform access control, and contribute to the generation of the group key. The group controller role is not usually present because the group keys are generated with contributions from all the members. Group Diffie-Hellman Key Exchange [77] is an example of the distributed scheme. It consists in the extension of the Diffie-Hellman key agreement protocol to groups of users. A group of n members firstly agrees on a pair of prime numbers (q and α). Each member generates also its secret number (s_i). The first member must calculate α^{s_1}, as its intermediate value, and sends it to the next member. Each subsequent member i will, in turn, generate a set comprising i intermediate values with $i-1$ exponents, plus a cardinal value containing all exponents. The fourth member, for instance, receives the set: $[\alpha^{s_2 s_3}, \alpha^{s_1 s_3}, \alpha^{s_1 s_2}, \alpha^{s_1 s_2 s_3}]$, and transmits to the fifth member: $[\alpha^{s_2 s_3 s_4}, \alpha^{s_1 s_3 s_4}, \alpha^{s_1 s_2 s_4}, \alpha^{s_1 s_2 s_3}, \alpha^{s_1 s_2 s_3 s_4}]$ The cardinal value of a member is the last value of the set generated; in this example $\alpha^{s_1 s_2 s_3 s_4}$ is the

fourth member's cardinal value. The n^{th} member can obtain the group key k by calculating: $(k = \alpha^{s_1 \cdots s_n} \, mod \, q)$. Since all members contribute to the group key, the processing time and network resources usage of this scheme increases linearly with the group size.

The hierarchical type assumes that users have not the same priorities and impose cryptographic access control for classes of users with different access levels. This scheme was firstly addressed in [13] where a hierarchical key assignment to users was adopted. A user belonging to a certain class can derive the cryptographic keys of lower classes.

Summary of Distributed and Hierarchical types

The distributed type presents the drawback of requiring extra computational power at the members, because all members participate in the generation of the group keys. The hierarchical type requires that users placed in high levels of the hierarchy must manage and store a large number of keys.

The distributed and the hierarchical types seem not to be applicable to IPTV services. The first type because the content is owned by a content or service provider and not by the users; that is, the content is not generated by the group and to the group but, instead, it is generated by a third party that needs to maintain control over the members that access the contents. The second type assumes that there are users hierarchically above the others and that those users are able to access content subscribed by users hierarchically below them. In a typical IPTV service, while there are users who access more video channels than others, no user can access content for which he has not a valid subscription.

3.4 Summary

In this chapter the current solutions related to multicast admission control and secure multicast were addressed. When considering multicast admission control two approaches were identified: the modification of IGMP/MLD signaling, and the introduction of an intermediate control layer between IP and IGMP processing.

With respect to secure multicast, the section starts by presenting the simplest form of secure multicast, followed by a categorization of current solutions. This categorization identifies 4 types of secure multicast, but

only the decentralized and the centralized types were detailed, since the distributed and the hierarchical types seem not to be applicable to IPTV services. The decentralized type focuses on scalability by splitting the group into sub-groups, each group having a sub-group controller. The centralized type addresses the reduction of signalling upon group changes, which is also the key issue of our thesis.

A comparison of multicast admission control techniques and centralized secure multicast solutions is also made. Secure multicast enables the enforcement of confidentiality requirements by not distributing cryptographic keys to non-members; non-members are not able to decrypt the content, thus preventing eavesdropping. With secure multicast no control is enforced in terms of data transmission, i.e a non-member can issue a join request for an unauthorized video channel and will trigger the extension of the multicast distribution tree and receive the encrypted video channel. On the other hand, multicast admission control prevents the extension of the multicast distribution tree to unauthorized users.

A combination of both secure multicast and multicast admission control prevent eavesdropping and unauthorized multicast distribution tree extensions. In particular, when using wireless access networks confidentiality is required. The multicast admission control also allows the network operators to effectively manage multicast traffic, including the user generated multicast sessions.

Chapter 4

Proposed solution - the SMIz Architecture

In IPTV services, the video is distributed as IP packets destined to multicast addresses. Each video channel is usually associated to an IP multicast group. Video channels may be grouped in bundles. A user subscribes usually one or more bundles, such as the "generic", the "movies" or the "sports" bundles. By subscribing a bundle, the user is enabled to receive and decrypt all the channels belonging to the bundle. Although a bundle is composed of multiple video channels, each video channel is transmitted to a unique multicast address.

The channel switch time is the time interval since the user signals his interest in viewing a new channel until the time instant where all the information required to view the channel is available to the user. In current solutions [85, 82, 23, 65, 66], channel switching requires signalling related to the user's departure from the current group and the user's join to the new group. Besides IGMP/MLD signalling, two group key refreshes are usually required in order to maintain group confidentiality. In turn, group key refreshes requires signalling to be sent to all group members. The amount of signalling exchanged increases with the group size.

Current solutions do not address scenarios where users can play the role of content provider. This scenario requires a new approach to group key management. In this situation, access control may be imposed by delivering the group decryption keys only to the allowed users, which are a subset of the remaining users. The content source must generate cryptographic

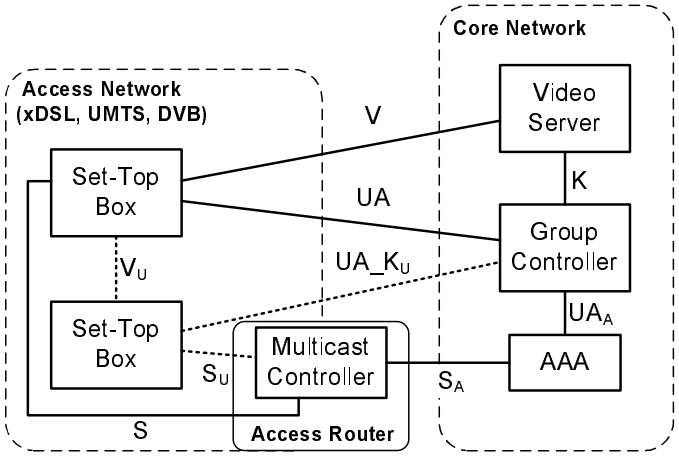

Figure 4.1: Proposed solution

contexts for the contents it generates, and be able to provide the system with a list of authorized users. Moreover, the transmission of multicast streams directly from the user premises is not addressed by current network access technologies, making difficult optimized group transmission directly from user's homes.

IP multicast does not provide admission control. Some solutions [16, 52, 25] support receiver admission control, while other solutions [50, 45] require modifications to IGMP/MLD protocols.

The network access technologies are starting to provide support for IP Multicast through the introduction of optimized link layer multicast. Nevertheless, such technologies still neglect multicast admission control, in particular in what concerns the interactions with AAA in order to benefit from both authentication and authorization.

4.1 Overview of the solution

The solution proposed in this thesis, named Secure Multicast IPTV with efficient support for video channel *zapping* (SMIz), has the architecture shown in Figure 4.1 and it consists of a Group Controller (GC), a Video Server (VS), Set-Top Boxes (STB), a Multicast Controller (MC) [71] and an Authentication, Authorization and Accounting (AAA) server [31]. The GC is responsible for key generation and distribution, STB authentication and authorization, and the updating of the STB multicast profiles stored in the

AAA server. The VS and the STB are responsible for the stream transformation, i.e. for the encryption or decryption of the video channel streams. The VS transforms the audio and video content into an encrypted stream of IP multicast packets; it also generates the Video Encryption Keys (VEKs) and distributes them to valid members (VEK announcement).

Prior to video channel request and visualization, the STB must obtain its cryptographic context from the GC. Three types of cryptographic keys are used in this solution: 1) Session Encryption Keys (SEKs); 2) Key Encryption Keys (KEKs); 3) Video Encryption Keys (VEKs). SEKs are used for securing unicast communications between STBs and the GC. VEKs are used to (de)encrypt video channels; each channel has a different VEK. KEKs, one per bundle (group of IPTV channels with some affinity), are used for securing the transmission of the VEKs.

At bootstrap, the STB requests the Session Encryption Key (SEK). This key (SEK) will be used to secure the video channel requests sent by the STB to the GC, enabling the STB to securely obtain the current KEK for the bundle to which the requested video channel belongs to. In order to receive the multicast transmission of the requested video channel, the STB must also send an IGMP join message to its designated multicast router. The destination address of this join request is the group address assigned to the video channel the user wants to receive. Each video channel is transmitted to its multicast group address in the form of Secure Real-time Transport Protocol (SRTP) packets, encrypted with a VEK. The VEKs are sent periodically with each video channel stream, to the same IP multicast group address, but to a different UDP port number; they are encrypted with the KEK. The STB decrypts the VEK with the KEK, and then decrypts the video channel stream with the VEK. To ensure a high level of security, all cryptographic keys must be refreshed periodically. These key refresh operations (re-keys) must not interfere with the video channel visualization of current receivers. Thus, each VEK is associated to a maximum SRTP packet sequence number, after which the VEK is no longer valid. The SEK is renewed upon each STB bootstrap and the KEKs are sent periodically by the GC to all STB, prior to their expiration. As a fall back procedure, the STB is also able to request the current KEK.

Fast switching between multiple video channels is enabled by the adoption of frequent and periodical VEK announcements (at a rate of 10 an-

nouncements per second), transmitted from the VS to the same group address of the video channel. VEK announcements are secured by $\{VEK\}_{KEK}$ and only one VEK announcement is needed for all members, since all of them share the same KEK. This procedure leads to savings in signalling, especially because VEKs are the keys that are re-keyed more frequently.

User generated multicast streams are supported by the proposed solution by allowing source users to obtain VEKs for their content, from the GC, and to distribute them to restricted groups of other users, which are a subset of the remaining users. In turn, the GC will trigger the update of the user's multicast profile stored at the AAA. These multicast profiles contain the users access rights in terms of multicast access (receivers case) and if they are authorized to generated their own multicast streams (senders case).

In addition to confidentially assurance, musticast admission control enables an extra degree of security by not allowing unnecessary extensions of the IP multicast distribution trees. Multicast admission control also promotes effective multicast session management by network operators. The MC is a new functional block added to existing access nodes that is responsible for the detection of new multicast sessions, be it by means of the detection of an IGMP packet (receiver control) or the detection of a new multicast data stream (sender control), and the subsequent authorization request to the AAA server. Upon a negative reply from the AAA, the MC will immediately discard the related multicast packet. On the other hand, upon a positive reply, the MC will allow the related multicat packet to be normally processed.

The AAA, based on the multicast profiles, rejects or authorizes multicast accesses. The AAA server can be placed anywhere on the network and we assume it stores the IP address assigned to each STB, as well as the multicast profile of each STB. The information stored in the AAA must enable STB identification by the MC. MC can only access information exchanged at the STB network attachment moment (802.1X or PPP packets) and information contained in IP packets.

4.2 Interfaces

Several interfaces are identified in Figure 4.1. The V interface is used to transmit the video streams from the VS to the STB; these streams are

Phases		Messages
1	Bootstrap	$STB \rightarrow GC : A.H(N_a).\{A.T_{s1}.N_a\}_{Kab}$ $GC \rightarrow STB : A.B.H(N_b).\{A.B.T_{s2}.N_b\}_{Kab}$ $STB \rightarrow GC : A.B.H(T_{s3}.N_a.N_b).\{A.B.T_{s3}.H(T_{s3}.N_a.N_b)\}_{Kab}$ $GC \rightarrow STB : A.B.H(T_{s3}.N_a.N_b).\{A.B.T_{s4}.SEK_i\}_{Kab}$
2	KEK Request	$STB \rightarrow GC : A.H(N'_a).\{A.N'_a.ChID\}_{SEK_i}$ $GC \rightarrow STB : A.B.H(N'_a.ChID).\{TTL.KEK\}_{SEK_i}$
3	KEK Refresh	$GC \rightarrow STB : A.B.H(N'_a.ChID).\{N'_a.TTL.KEK\}_{SEK_i}$

Table 4.1: Messages exchanged through the UA interface

transmitted as IP packets having a multicast address as destination. The V_U interface is similar to V interface, but used to transmit the video streams generated by the domestic users as IP multicast to other users. Through the UA interface the GC authenticates and authorizes STBs for video channel access. The K interface is used by VS to inform the GC about the VEK associated with each video channel. The UA_A interface is used by the GC to update the STB multicast profiles stored at the AAA server. The UA_K_U interface enables the STB generating a video multicast stream to request a KEK from the GC, and to inform the GC about the users which may access the user generated video channel. The S interface is used by a STB to signal its interest in receiving a video channel IP multicast stream. The S_U interface is used by a STB to signal its interest in transmitting a user generated video channel stream. The S_A interface is used by the MC in order to obtain from the AAA the authorization to access or retransmit an IP multicast stream.

4.2.1 UA Interface

The UA interface is used by the GC to authenticate and authorize an STB to access a video channel. Through this interface the messages required to bootstrap the STB, request the KEK, and refresh the KEK are exchanged. Table 4.1 summarizes the messages exchanged in these 3 phases.

The Bootstrap phase is executed during the STB bootstrap and it enables mutual authentication of STB and GC by means of the symmetric pre-shared key (K_{ab}). In this phase, the initiator is the STB and it starts by sending a message composed of the initiator's identification (A), the result of a hash function of a fresh nonce ($H(N_a)$), and a set of 3 fields ($A.T_{s1}.N_a$) encrypted with a pre-shared symmetric key (K_{ab}). The GC decrypts this

set, using the initiator's identification to select the correct pre-shared key, and tests both the nonce and the time seed against previous values. The nonce must not be repeated, and the time seed must be higher than the last time seed (typically, T_{s3} from the previous bootstrap). The GC will reply with a similar message that, besides the identification of the GC (B), contains a fresh nonce (N_b) generated by GC and its time seed (T_{s2}). The STB will verify the nonce and time seed. Upon successful verification, the STB will reply with a new message composed of the identification of both, the result of a hash function of both nonces, a new time seed (T_{s3}), and a new encrypted set of fields. This set of fields is composed of the identifications A and B, the time seed T_{s3}, and a hash result. In turn, and upon successful verification, the GC will reply with a message that differs only in the encrypted set of fields, which contains a new time seed generated at the GC and a new SEK for that specific STB (SEK_i). At the end of the bootstrap phase, the STB will be in possession of its new SEK_i and no other entity, besides GC, knows SEK_i. The time seeds (T_{s1} through T_{s4}) and nonces (N_a and N_b) are used to prevent replay attacks. Using only time seeds, we would be able to perform mutual authentication, but a secure time synchronization mechanism would be required. On the other hand, using only nonces would imply possible men-in-the-middle attacks. Combining both techniques, nonces and time seeds, mutual authentication with replay attack prevention is obtained.

The second phase is the channel request (KEK Request). The channel request is sent by the STB to the GC and it is secured by the SEK_i obtained during the first phase; this request aims at obtaining the KEK currently associated to the bundle to which the requested video channel, identified by the channel identifier $ChID$, belongs to. The GC answers with the KEK and its associated time-to-live (TTL).

The third phase (KEK refresh) is analogous to the second phase, with the difference that it is initiated by the GC when a KEK refresh is required. The KEK refresh messages are sent by the GC to individual STB as unicast messages. These messages contain the KEKs that are periodically sent to the STB prior to their expiration, and are used to encrypt a bundle; as a fall back procedure, the STB can also request the current KEK.

Sequence	Messages
1	$STB \rightarrow MC$: IGMP Join
2	$MC \rightarrow AAA$: Auth. Request (Mcast_Session_ID.User_ID)
3	$AAA \rightarrow MC$: Auth. Response (Accept)
4	$MC \rightarrow STB$: Multicast Stream

Table 4.2: Messages exchanges through the S and S_A interfaces

Phases		Messages
1	Streaming	$VS \rightarrow STB : \{SRTP_i\}_{VEK}$
2	VEK Refresh	$VS \rightarrow STB : C.A.MsgID.H(N'_b).\{N'_b.ChID.ChCTX.VEK\}_{KEK}$

Table 4.3: Messages exchanged through the V interface

4.2.2 S Interface

The S interface is used by an STB to signal its interest in receiving a video channel distributed as an IP multicast stream. Before receiving the multicast stream of the requested video channel, the STB must send an IGMP/MLD join message to its designated multicast router. The destination address of this join request is the group address (IP multicast address) assigned to the video channel the user wants to receive. Each video channel is transmitted to its multicast group address. Upon receiving the join request, the MC will send an authorization request to the AAA, through the S_A interface. The request contains both the identification of the multicast session and the STB identification. If authorized, the MC will process the join request, allowing the extension of the distribution tree; if not, the MC will discard the join request. The messages exchanged through the S interface and their relationship with messages exchanged through the S_A interface are shown in Table 4.2.

4.2.3 V Interface

The V interface is used to transmit the video channels and the VEK announcements from the VS to the STB. Both are transmitted to IP multicast groups. It comprises 2 phases: Streaming, and VEK Refresh. Table 4.3 summarizes the messages exchanged in these phases.

The first phase consists of the video channel transmission to its multicast group address in the form of Secure Real-time Transport Protocol (SRTP) packets, encrypted with a VEK. The second phase represents the VEK re-

Phases		Messages
1	KEK Request	$VS \rightarrow GC : C.B.MsgID.H(N'_c).\{N'_c.ChID\}_{K_{cb}}$
2	KEK Setup	$GC \rightarrow VS : B.C.MsgID.H(N'_b).\{N'_b.ChID.KEK\}_{K_{cb}}$ $VS \rightarrow GC : C.B.MsgID.H(N'_c).\{N'_c.ChID.KeyACK\}_{K_{cb}}$

Table 4.4: Messages exchanged through the K interface

fresh, also referred to as VEK announcement. The VEKs are periodically sent by the VS, in multicast, to the same IP multicast group address of the video stream, but to a different UDP port. The VEK is encrypted with the KEK of the bundle. We recall that there is one KEK for each bundle and one VEK for each channel.

The STB decrypts the VEK announcement with the KEK, and then decrypts the video channel stream with the VEK. To ensure a high level of security, all cryptographic keys must be periodically refreshed. These key refresh operations (re-keys) must not interfere with the video channel visualization of current receivers. For that purpose, each VEK is associated to a channel context ($ChCTX$) that contains a maximum SRTP packet sequence number, after which the VEK is no longer valid. The $ChCTX$ also contains the video channel SSRC identifier, a 64 bitmap used by SRTP to prevent replay attacks, and the number of times this bitmap has reached its maximum value (roll-over counter).

The support of fast switching between video channels is achieved by transmitting the periodic and frequent VEK announcements, in multicast, to the same group address of the video channel. VEK announcements are secured by $\{VEK\}_{KEK}$ and only one VEK announcement per video channel is needed for all members, since all of them share the same KEK. This procedure of transmitting the frequently refreshed VEKs in multicast leads to significant savings in signalling, because a single message containing the new VEK can be received by multiple users. Users zapping between channels of the same bundle do not need to obtain a new KEK.

4.2.4 K Interface

The K interface is used to synchronize the cryptographic context of a video channel between VS and the GC. The VS is responsible for the VEK refresh message generation that, in turn, is secured by KEK, which is generated by the GC. It consists of 2 phases: KEK Request, and KEK Setup.

Received Packet	Multicast Session IDs
IGMPv1/v2	SA, GDA
IGMPv3	SA, GDA, GSA
UDP multicast	SA, DA

Table 4.5: Multicast session IDs source

Table 4.4 summarizes the messages exchanged in these phases. The messages exchanged through this interface are protected by means of a symmetric key (K_{ab}) previously shared between the VS and the GC. The KEK Request phase exists as a fall back procedure enabling the VS to request a KEK Setup of a video channel bundle when, for some reason, the VS was unable to successfully conclude previous KEK Setup phases. The KEK Setup phase has two messages. The first message is, in structure, similar to the VEK refresh message of Section 4.2.3; it comprises the identifications of the involved entities (B and C), the $MsgID$ field that identifies the message as a KEK Setup message, the result of a hash function over a fresh nonce ($H(N'_c)$), and a set of fields encrypted with the pre-shared key. The set of encrypted fields comprises the video channel identifier ($ChID$) and the respective bundle KEK. Upon reception of a KEK Setup message, the VS is expected to confirm its reception by sending the acknowledgement message shown in Table 4.4.

4.2.5 S_A Interface

The S_A interface consists of RADIUS or DIAMETER messages. This interface is used to send authorization requests to the AAA server when multicast sessions are detected, be it an IGMP packet (receiver control) or a multicast data stream (sender control). The authorization requests sent by MC (RADIUS or DIAMETER messages) contain the user and multicast session identifiers, obtained from the available network information at the access node (e.g. IP address). Upon a successful authorization, the IGMP packet (receiver case) or the multicast stream (sender case) are processed by the access node. In case of an unauthorized access, the packets are discarded before they reach the IP layer at the access node.

In order to enable sender and receiver multicast IP control at the access node, the MC must be able to uniquely identify the multicast session and authenticate its users. Table 4.5 summarizes the multicast session identifiers

Phases		Messages
1	UGV Setup	$STB \rightarrow GC : A.B.MsgID.H(N'_a).\{N'_a.XMLDATA\}_{SEK_i}$ $GC \rightarrow STB : B.A.MsgID.H(N'_b).\{N'_b.M\_Add.M\_Port.KEK\}_{SEK_i}$

Table 4.6: Messages exchanges through the UA_K_U interface

adopted, obtained from IP multicast packets. A member's session can be identified in one of two ways, depending on whether IGMPv1/v2 or IGMPv3 is used. In the case of IGMPv1/v2, the member's session is identified by the user's IP Source Address (SA) and the Group Destination Address (GDA). In case of IGMPv3, along with the SA and GDA, a third identifier is also used, the Group Source Address (GSA). The SA is the user's IP address; the GDA is the group's IP address the user wants to join or is currently a member of; the GSA is the IP address of the multicast group's source.

4.2.6 UA_K_U Interface

The UA_K_U interface is used by an STB transmitting an user generated video multicast stream, in order to request a KEK from the GC and to inform the GC about the list of users that may access the user generated video channel. Table 4.6 summarizes the messages exchanged.

STB starts by sending the first message from the UGV Setup phase, shown in Table 4.6. It comprises the identification of the involved entities (A and B), a message identifier, the hash of a fresh nonce ($H(N'_b)$), and an encrypted set of fields. The encrypted set is protected with the STB's SEK and it comprises: the fresh nonce and XML formatted data that lists the allowed users. An example of such XML data is shown in Listing 4.1. If the STB is authorized to generate video channels, the GC replies with a message that contains an encrypted set of fields composed by the multicast address (M_Add), port number (M_Port), and KEK to be used in the video channel streaming. The KEK will be used to protect VEK refresh messages.

```
<?xml version="1.0"?>                                                    1
<UGV>                                                                    2
  <SOURCE>user1@exampledomain.com</SOURCE>                               3
  <RECEIVERS>                                                            4
    <USERNAME>user2@exampledomain.com</USERNAME>                         5
    <USERNAME>user3@exampledomain.com</USERNAME>                         6
    <USERNAME>user4@exampledomain.com</USERNAME>                         7
```

Phases		Messages
1	Profile Setup	$GC \rightarrow AAA : B.D.MsgID.H(N'_b).\{N'_b.XMLPROFILE\}_{K_{db}}$
2	UGV Setup	$GC \rightarrow AAA : B.D.MsgID.H(N'_b).\{N'_b.XMLUGV\}_{K_{db}}$

Table 4.7: Messages exchanges through the UA_A interface

```
  </RECEIVERS>                                                    8
</UGV>                                                            9
```

Listing 4.1: Example of XML formated data of a User Generated Video Setup

4.2.7 UA_A Interface

The GC uses the UA_A interface to send STB multicast profiles to the AAA server and to update these profiles in scenarios where a user generates his own video channels. Table 4.7 summarizes the messages exchanged.

The first phase (Profile Setup) shown in Table 4.7 enables the initial multicast profile definition for an STB. Is consists on the GC sending to the AAA a message protected with a previously shared key (K_{db}) that comprises a XML formatted multicast profile of the STB. An example of such profile is shown in Listing 4.2 and it consists of the list of channels subscribed by the STB and the UGV field (line 12) that indicates whether user generated video streams are allowed (value 1) or not (value 0).

```
<?xml version="1.0"?>                                             1
<MCASTPROFILE>                                                    2
  <USER>user1@exampledomain.com</USER>                            3
  <LIST>                                                          4
    <CHANNEL> <ID>1</ID>                                          5
      <DESCRIPTION>Channel 1</DESCRIPTION>                        6
      <ADDRESS>229.0.0.1</ADDRESS>                                7
    </CHANNEL>                                                    8
    <CHANNEL> <ID>2</ID>                                          9
      <DESCRIPTION>Channel 2</DESCRIPTION>                        10
      <ADDRESS>229.0.0.2</ADDRESS>                                11
    </CHANNEL>                                                    12
  </LIST>                                                         13
  <UGV>1</UGV>                                                    14
</MCASTPROFILE>                                                   15
```

Listing 4.2: Example of XML formated multicast profile (XMLPROFILE)

Sequence	Messages
1	$STB \rightarrow MC$: Multicast Stream (S,G)
2	$MC \rightarrow AAA$: Auth. Request (Mcast_Session_ID.User_ID)
3	$AAA \rightarrow MC$: Auth. Response (Accept)
4	$STB \rightarrow MC$: Multicast Stream (S,G)

Table 4.8: Messages exchanges through the S_U interface

The second phase (UGV Setup) is used by the GC to inform the AAA server about new user generated videos. For that purpose, the GC sends a message to AAA containing an XML formated data with the relevant information. An example of such information is shown in Listing 4.3 and comprises the source identification, the multicast address and port number to be used in the video channel streaming, and a list of users allowed to access the content. This information is required by the AAA server in order to reply to MC queries with respect to multicast admission control.

```
<?xml version="1.0"?>                                                    1
<UGV>                                                                    2
  <SOURCE>user1@exampledomain.com</SOURCE>                               3
  <MCAST_ADDRESS>232.1.0.1</MCAST_ADDRESS>                               4
  <MCAST_PORT>1234</MCAST_PORT>                                          5
  <RECEIVERS>                                                            6
    <USERNAME>user2@exampledomain.com</USERNAME>                         7
    <USERNAME>user3@exampledomain.com</USERNAME>                         8
    <USERNAME>user4@exampledomain.com</USERNAME>                         9
  </RECEIVERS>                                                           10
</UGV>                                                                   11
```

Listing 4.3: Example of XML formated data sent by GC to AAA (XMLUGV)

4.2.8 V_U and S_U Interfaces

The V_U interface, similarly to V interface, is used to transmit the IP multicast streams of the video channels generated by domestic users. It consists of the stream's SRTP packets, which are protected with VEK.

The S_U interface is used by an STB to signal its interest in transmitting a user generated video channel stream. The messages exchanged through this interface, shown in Table 4.8, are the first SRTP packets of the stream that, when received by the MC, will trigger the authorization validation by

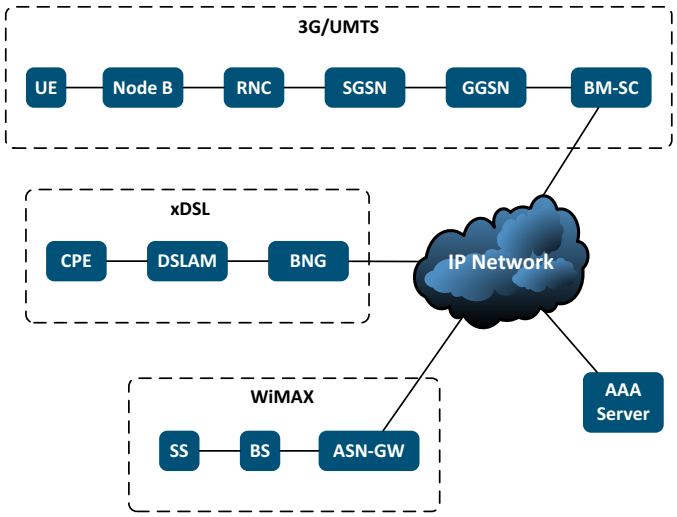

Figure 4.2: Adopted network architecture

the AAA. Upon a successful authorization, the MC will start to forward all the packets of the multicast stream. Upon an unsuccessful authorization, the MC will discard all the other packets of the multicast stream.

4.3 Heterogeneous access networks support

The solution proposed for IP multicast admission control operates at the network level, which makes it adequate to any access network supporting IP multicast. Nevertheless, such aspects as the adequacy of the access technology to IP multicast, as well as the functionalities available in the functional elements of each access technology, influence the global multicast solution. The challenge is to define where and how to perform access control, for both traditional multicast groups and groups sourced at the user premises, in the context of heterogeneous access networks (UMTS, xDSL and WiMAX). The network architecture considering relevant network elements of these access network technologies is shown in Figure 4.2.

Table 4.9 summarizes the support each access technology has for multicast. The elements responsible for IGMP message processing are the BNG (xDSL), the ASN-GW (WiMAX) and the GGSN (UMTS); these elements enable the IP multicast support. On the other hand, when considering optimized link-layer multicast communications, some differences arise. For instance, while xDSL networks may optimize multicast communications in

	IP Multicast	L2 Multicast (DL)	L2 Multicast (UL)	IMGP Processor
xDSL	Yes	Yes	Yes	BNG
WiMAX	Yes	Yes	No	ASN-GW
UMTS	Yes	Yes	No	GGSN

Table 4.9: Multicast support comparison

both directions (uplink and downlink), WiMAX and UMTS networks only support optimized link-layer multicast communications in the downlink. In xDSL networks and in order to support uplink optimized link-layer multicast communications, DSLAMs must assume the role of multicast packet replication by either being IP-aware or by implementing IGMP snooping functionality. In WiMAX, there are only multicast CID for the downlink, meaning that multicast groups sourced at the user premises must be transmitted at least until the ASN-GW. In UMTS, the MBMS services are also designed to operate only for the downlink (multicast groups destined to the users) and for multicast sources known to the BM-SC.

The BNG is the access router in xDSL; its roles include the processing of IGMP messages and forwarding of multicast packets. The BNG is also the NAS where users authenticate themselves during network attachment and using PPP. The proposed solution can be supported by introducing the MC functionality into the BNG. If L2 multicast replication exists, then the control over the last multicast replication point should also be extended to those L2 multicast replication network elements, namely DSLAMs.

In WiMAX the roles of IGMP processing and multicast packet forwarding fall upon the ASN-GW network element, which is also responsible for client AAA using, in this case, 802.1X. The proposed solution can be supported by introducing the MC functionality into the ASN-GW.

UMTS networks, since Release 99, have support for IP multicast. The IGMP processing and multicast packet forwarding roles are performed by the GGSN. IP multicast packet transmission inside the UMTS network is performed over point-to-point tunnels (from the GGSN to the UE). In Release 6, MBMS was introduced to support native multicast and was designed for IP multicast interoperability. There are then two possible deployment scenarios for multicast, one with MBMS and another with typical IP Multicast. With the latter no sharing gains are obtained but the proposed solution can be fully supported. With MBMS, although multicast control is achieved,

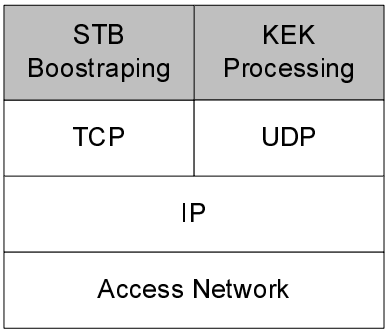

Figure 4.3: GC prototype architecture

there is no support for user generated multicast streams.

The solution proposed does not require any changes to user equipment or multicast protocols but will benefit from all the support these technologies provide to multicast.

4.4 Design

This section describes the implementation process, the prototypes developed and the main data structures used in GC, STB, VS and MC. Two prototypes were developed to validate both the performance and functionality of the proposed solution. Two groups of experiments were made: signalling and admission control, what lead us to the development of two prototypes, one for each group of experiments.

4.4.1 GC

The architecture of the GC prototype is shown in Figure 4.3. The GC has two threads, one responsible for STB bootstrap processing and the other responsible for KEK processing. The first thread implements the GC functionalities related to phase 1 of the *AU* interface shown in Table 4.1. The STB bootstrap phase is used to authenticate and exchange the SEK of the STB, which requires reliability and thus is implemented over TCP connections. The KEK Processing thread implements the GC functionalities related to phases 2 and 3 of the *AU* interface shown in Table 4.1, and phases 1 and 2 of the *K* interface shown in Table 4.4. Two open source libraries were used, namely libSRTP [2] and Libgcrypt [1]. LibSRTP is as open source implementation of the Secure Real-time Transport Protocol (SRTP) [19],

which enables the encryption and decryption of the RTP streams used to transport the video channels. Libcrypt is a general purpose cryptographic library that provides functions for both symmetric and asymmetric ciphers.

```
subscriber_t* subscriber_hashtable[HASH_SIZE];

typedef struct{
  u_int32_t settopbox_id;
  datum_t sek;
  datum_t iv;
  u_int32_t channels_bitmap[32];
  xtd_replay_db_t replay_db;
  audience_t *last_channel_seen;
  time_t last_sek_request_tstamp;
  subscriber_t *next;
  }subscriber_t
```

Listing 4.4: STB State information

Listing 4.4 shows the data structures used by the GC to store STB state information. This data structure contains the STB identifier, the STB SEK, the Initialization Vector (IV) required by the cryptographic algorithms, the list of channels the STB is allowed to access, the anti-replay database used by libSRTP, a reference to the list of STBs that are viewing the same channel, and a reference to the next STB list element. In order to store this information and reduce memory seek times a hash table was used.

```
struct channel_t channelList[1024];

typedef struct{
  int available;
  u_int64_t kek_ttl;
  u_int32_t kek_cipher;
  rekey_crypto_policy_t kek_policy_buffer[3];
  audience_t *listeningSubscribers;
  video_server_t *delegated_vs;
  }channel_t

typedef struct{
  datum_t kek;
  datum_t iv;
  }rekey_crypto_policy_t

```

```
typedef struct{                                    17
  subscriber_t* current;                           18
  audience_t* next;                                19
  audience_t* prev;                                20
  }audience_t                                      21
```

Listing 4.5: Video channel state information

Listing 4.5 shows the data structures used by the GC to store video channel state information. In this case, the state information is stored in three structures: 1) the channel data structure (channel_t), 2) the channel cryptographic context data structure (rekey_crypto_policy_t), and 3) the channel audience data structures (audience_t). The channel_t structure includes the other two structures, information regarding the video channel availability, the KEK refresh period, and the cipher algorithm. rekey_crypto_policy_t structure maintains the cryptographic contexts of each video channel and contains information such as the KEK and the IV. The audience_t structure is used to maintain the list of STB that are currently viewing each video channel. An array of 1024 positions is used to maintain the IPTV system's video channel list; while limiting the number of video channels available in the IPTV service, it limits the memory seek times when searching for the state information of a video channel.

```
typedef struct{                                    1
  u_int32_t videoServerID;                         2
  u_int32_t ip_address;                            3
  int psk_algo                                     4
  datum_t cipher_key                               5
  datum_t iv                                       6
  struct video_server* next;                       7
  } video_server                                   8
```

Listing 4.6: VS State information

Listing 4.6 shows the data structure used by the GC to store the VS state information. This data structure (video_server) consists of the VS identification, the VS IP address, the cryptographic algorithm, the pre-shared key, the IV, and a reference to the next VS in the list of VS.

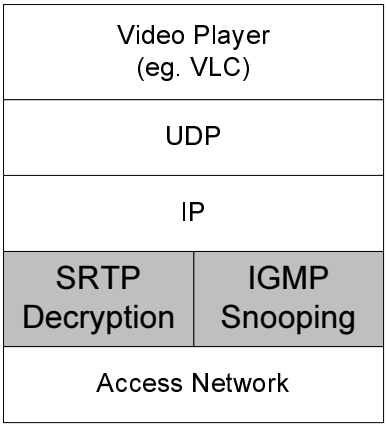

Figure 4.4: STB prototype

4.4.2 STB

The STB prototype has the architecture shown in Figure 4.4. It consists of two threads shown in gray color: the IGMP snooping, and the SRTP decryption threads. The IGMP snooping thread in responsible for the detection of IGMP join requests sent by the STB and for sending video channel requests to the GC. The SRTP packet decryption thread is responsible for the decryption of VEK announce messages and for the requested video channel decryption. Both the IGMP snooping functionality and the SRTP packet capture were implemented using the *iptables/netfilter* [4] framework[1]. The libSRTP [2] was used to decrypt the video channel.

```
typedef struct{                                          1
  int channel_id;                                        2
  u_int32_t mapped_ssrc;                                 3
  u_int32_t mcast_address;                               4
  int srtp_ctx_inited;                                   5
  srtp_t srtp_session;                                   6
  srtp_policy_t srtp_policy;                             7
  int xtd_seq_inited;                                    8
  u_int64_t current_xtd_seq;                             9
  u_int16_t rtp_sequence_reference;                      10
  kek_policy_t kek_policy_buffer;                        11
  vek_policy_t current_vek_policy;                       12
  announce_ctx_t announce_ctx;                           13
```

[1]Netfilter framework enables packet capturing, filtering and queuing on the Linux operating system

```
}channel_context_t                                                    14
```

Listing 4.7: Video channel state information

Listing 4.7 shows the data structure (channel_context_t) used by the STB to store active video channel information. This data structure consists of: the video channel identifier; the SRTP stream identifier; the video channel multicast address; a flag to indicate if the SRTP context was correctly initialized; the SRTP session description; the SRTP policy description; a flag indicating whether the libSRTP anti-replay functionality was activated; the libSRTP anti-replay extension sequence number[2]; the SRTP sequence number; the KEK; the VEK; and the VEK announce context information.

```
typedef struct {                                                      1
  u_int64_t srtp_rekey_seq;                                           2
  datum_t vek;                                                        3
  }vek_policy_t                                                       4
```

Listing 4.8: STB VEK state information

The vek_policy_t data structure shown in Listing 4.8 contains the VEK and the SRTP stream VEK expiration sequence number, meaning that the SRTP packets containing a sequence number higher than that stored in the data structure are encrypted with a different VEK.

```
typedef struct {                                                      1
  u_int8_t announce_id;                                               2
  int process_current_vek;                                            3
  }announce_ctx_t                                                     4
```

Listing 4.9: VEK announce state information

The announce_ctx_t data structure shown in Listing 4.9 consists of the announcement identification and the flag process_current_vek. The announcement identification stores the identification of the last successful announce. The flag value indicates whether the current announcement was already processed.

[2]An SRTP session includes a sequence number to avoid the processing of the same packet more than once. Because the SRTP packet sequence number is a small value, it often reaches its maximum value in packet intensive streams such as a video stream. To avoid attacks that resend previously captured packets and to augment the packet sequence number, libSRTP can use an extra value that is concatenated with the sequence number. This extra value is named extension sequence number

```
typedef struct {
  int id;
  u_int32_t multicast_address;
  channel_info_t *next;
}channel_info_t
```

Listing 4.10: STB Video channel list

Listing 4.10 shows the data structure (channel_info_t) used by the STB to store the video channel list and contains the video channel multicast address and the respective video channel identification. This information is used by the IGMP snooping process to obtain the identification of the channel associated to the multicast addresses present in the IGMP join messages. The channel identification is part of the video channel request messages sent by the STB to the GC, represented in Table 4.4.

4.4.3 VS

The VS prototype consists of a process responsible for the video channel stream encryption, using the VEK, and respective VEK announcements. The video channel state information is maintained using a data structure similar to the STB's data structure shown in Listing 4.7, and differing only in the nested data structure named vek_policy_t.

```
typedef struct {
  rekey_clock_t clock;
  datum_t current_vek;
  datum_t next_vek;
}vek_policy_t

typedef struct {
  u_int64_t rekey_interval;
  u_int64_t rekey_seq;
}rekey_clock_t
```

Listing 4.11: VEK state information used by the VS

The structure of the vek_policy_t used by the VS is shown in Listing 4.11 and it comprises the current and the next VEK of each video channel stream, and a nested data structure related to the VEK re-key interval. The latter data structure comprises the re-key intervals in number of packets and the re-key message (VEK announcement) sequence number.

4.4.4 MC

The MC operates at the IP level. It captures multicast packets, being them a join request to an existing multicast stream or a data packet generated by a user to a multicast group. After capturing a packet, the MC issues an authorization request to the AAA server. Upon a positive reply, the MC allows the respective multicast packets to be normally processed; upon a negative reply, the multicast packets are discarded. The MC identifies the requesting users using the information available in the multicast packets, according to Table 4.5.

The MC functionalities are then fourfold: 1) source detection, 2) member detection, 3) authorized session management, and 4) authorization requests. Sources are detected by analysing the multicast packet header. When an UDP header is detected then the MC may detect a multicast stream generated by the user in the access network; in this case the relevant identification information is extracted from the IP header of the packet and an authorization request is sent to the AAA server. Members are detected similarly; if the captured multicast packet is an IGMP packet, then we are in the presence of a member also located at the access network and, after the extraction of the relevant identification information, the authorization request is sent. In order to improve the performance of the MC, after the authorization of a multicast session, the session is added to a white list, preventing per packet authorization requests to the AAA server. The authorization requests sent to the AAA server contain the identification information extracted from the multicast packets, using the IP address of the user as its identification.

Prototype

The prototype developed to evaluate the multicast admission control in heterogeneous access networks is shown in Figure 4.5. This prototype addresses the main aspects of admission control, namely STB authentication during network attachment, multicast source and receiver detection at the access router, and the use of an AAA server for authorization requests. The AAA server was setup in the same machine of the Access Router, for the sake of simplicity. Additionally, two machines were setup as clients (User 1 and User 2), each of them emulating an STB using a different authentication protocol at the moment of network attachment. User 1 used IPoE plus

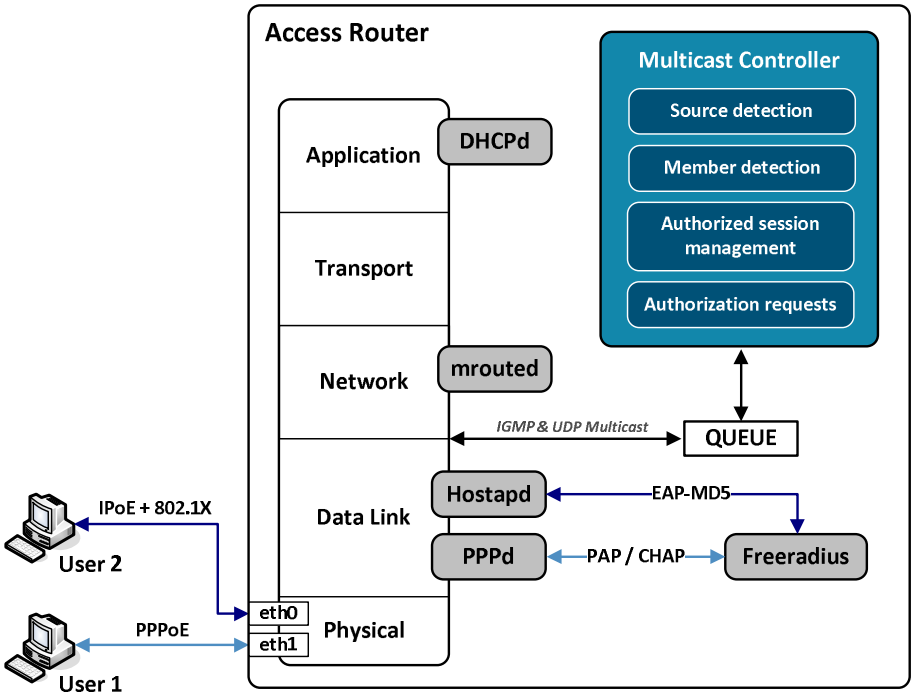

Figure 4.5: MC functional architecture

801.1X, typical of wireless access networks such as WiMAX; User 2 used PPPoE, normally used in xDSL access networks.

The software used to build the prototype was based on open source software. The AAA server was implemented with *FreeRADIUS*. The network attachment authentication protocols were implemented with *pppd* (for the PPPoE conection of User 1) and *wpa_supplicant* (for the 802.1X connection of User 2). The authenticator process of 802.1X was implemented with the *hostapd* software. A *DHCPd* was also used to provide IP configuration to both User 1 and User 2. For the specific case of IGMP packet processing, the software used was *mrouted* in combination with *netfilter* framework. In particular, the *libnetfilter_queue* and *libnetfilter_conntrack* libraries were used to capture multicast packets, including IGMP messages. Figure 4.5 shows the functional architecture of the MC prototype. The open source software packages used are identified in gray color. The MC main functionalities are identified in blue color.

Initially, users perform the network attachment authenticating themselves by means of a user name and password. After that, all the multi-

```
[root@localhost McastControl]# iptables -t raw -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
QUEUE      udp  --  anywhere             anywhere            ADDRTYPE match dst-type MULTICAST
QUEUE      igmp --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@localhost McastControl]#
```

Figure 4.6: Multicast traffic queuing configuration

cast traffic, including the IGMP messages, is redirected to a packet queue (QUEUE) using the *netfilter* libraries. The multicast packets are captured prior to their processing by the IP stack. Once in the QUEUE, the MC will process all packets, one at a time. Figure 4.6 shows the configuration used to capture and queue the multicast traffic.

Source detection

The packet queue QUEUE contains multicast packets, including IGMP packets. The MC must firstly identify the type of multicast packet being processed. If not an IGMP join message, then the MC must extract the multicast session information from the packet header (see Table 4.5); this information enables the authorization request to the AAA server. Listing 4.12 shows the data structure used to maintain multicast source related data, namely the IP addresses of the source and of the multicast group.

```
typedef struct{
  ipaddr ip;
  ipaddr group;
} source_data
```

Listing 4.12: Multicast source data structure

Member detection

If the captured multicast packet is an IGMP Join message sent from the access network, the MC must also process the IGMP packet header. For that purpose, the MC firstly identifies the type of IGMP message (join, leave or membership report) and the IGMP version (1,2 or 3); the MC extracts the required information from the packet and fills the data structure (member_data) shown in Listing 4.13.

```
[root@localhost mrouted-3.9b3]# iptables -t raw -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  192.168.2.10         224.0.1.20
QUEUE      udp  --  anywhere             anywhere            ADDRTYPE match dst-type MULTICAST
QUEUE      igmp --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@localhost mrouted-3.9b3]#
```

Figure 4.7: White list traffic queuing configuration

```
typedef struct{
  ipaddr ip;
  ipaddr group;
  ipaddr source;
} member_data
```

Listing 4.13: Multicast member data structure

Authorized sessions management

In order to improve the system performance and minimize authorization requests, the MC keeps a temporary white list of previously authorized multicast sessions, and translates this list into access control rules. In this way, the packets belonging to previously authorized sessions are immediately accepted, eliminating the delay associated with the inclusion of the MC's functionality at the access node. This is particularly useful for multicast sources, where the multicast stream can have a high packet rate and delays can influence the video stream quality. The same strategy is used for maintaining a blacklist of recently unauthorized multicast sessions.

Besides performance issues, MC is also required to detect the new IGMPv1 and IGMPv2 join requests sent by users. The IGMPv1 and IGMPv2 do not have an explicit join message, meaning that the first membership report sent by a user will serve as a join request. Thus, MC is required to maintain state information to distinguish both situations (membership report, as a join request or as a reply to a membership query). This lists are periodically refreshed, and for each entry in the list a new authorization request is sent to the AAA. Upon a negative authorization reply, the entry is removed from the list.

For the particular case of user generated multicast streams with high packet rates, where the processing time introduced by the MC could be seen

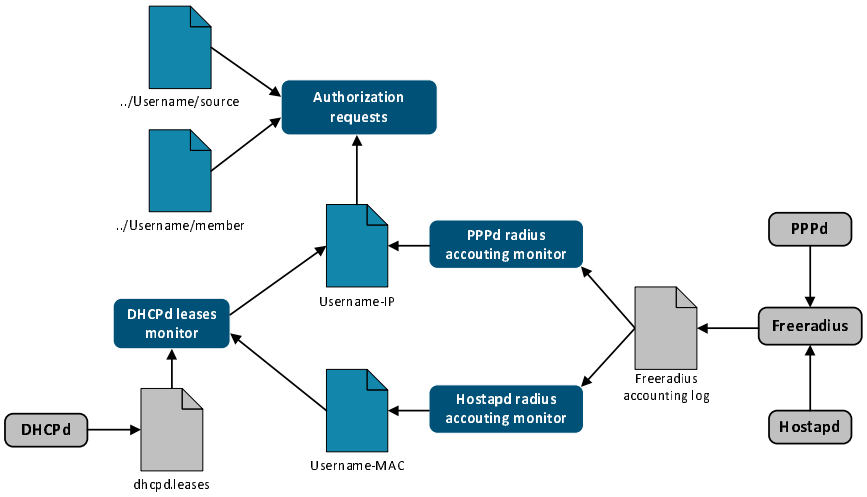

Figure 4.8: MC log monitoring process

as a bottleneck, upon authorization by the AAA and after being included in the white list, a new rule is added to the system's firewall to avoid packet reprocessing. Figure 4.7 shows the configuration of the system's firewall, after the insertion of a source by the MC. After this configuration, the multicast packets of the stream sourced at the user will no longer be captured nor processed by the MC. Periodically the system's firewall configuration is cleared of all authorizations to force stream reauthorization.

Authorization requests

Besides the multicast session identifiers present in Table 4.5, the MC also uses the IP address as the user identifier. This identification requires an entity to store the relationship User/IP address, along with the user's multicast profile. Mainly due to the fact of having used open source software, this identification presented itself as a major problem. If PPPoE is used during network attachment, the RADIUS server will store the IP address assigned to the user's equipment; if 802.1X is used, the authentication and IP configuration processes are two different processes, requiring *hostapd* to authenticate and then *DHCPd* to deliver IP configuration. In 802.1X connections, the RADIUS server stores the MAC address.

The solution adopted to solve this problem was based on an additional log monitoring process, developed to create and maintain the required User/IP address associations. Figure 4.8 shows, in blue, the threads used by the

monitoring process. FreeRADIUS log file contains references that map user names and IP address for the PPPoE connections, and maps user names and MAC addresses for the IPoE plus 802.1X connections. In order to obtain the required User name/IP address for the 802.1X authenticated connection, it is required to also monitor the DHCPd leases log, where a mapping of MAC and IP addresses can be found.

The MC is now able to verify two additional text files, named source and member. These text files compose the multicast profile, and are stored inside a directory named after the user's address. The source text file contains the list of IP multicast addresses for which the user is authorized to generate and transmit multicast streams. The member text file contains the list of IP multicast addresses the user is authorized to join.

4.5 Summary

This chapter describes the proposed solution, named Secure Multicast IPTV with efficient support for video channel *zapping* (SMIz). It comprises a GC, a VS, STBs, a MC and an AAA server. The GC generates and distributes cryptographic keys, authenticates STBs, and updates the STB multicast profiles stored in the AAA server. The VS and the STB are responsible respectively for the encryption and decryption of the video channel streams. The VS also generates and distributes VEKs to valid members. The MC is responsible for the detection of multicast sessions for both sources and receivers, and the subsequent authorization request to an AAA server.

The proposed solution uses three types of cryptographic keys: SEKs, KEKs, and VEKs. SEKs are used for securing unicast communications between STBs and the GC. VEKs are used to (de)encrypt video channels, each channel having a different VEK. KEKs, one per bundle, are used for securing the transmission of the VEKs. VEK announcements consist in transmitting the VEK encrypted with the respective bundle KEK. The combined usage of KEKs and VEKs allows the periodic and frequent transmission of all VEKs of all video channels (VEK announcements), without requiring significant network resources.

The multicast admission control technique of the proposed solution consists in introducing the MC functionality into relevant network elements of the heterogeneous access networks considered (xDSL, WiMAX, and UMTS).

The MC functionality consists in capturing multicast packets originated at the access networks and then, based on the identification information extracted from the multicast packets, sending authorization requests to the AAA server. AAA stores multicast profiles that contain the users access rigths in terms of multicast access (receivers case) and if they are authorized to generated multicast streams (senders case). Upon a successful authorization the packets are allowed to be normally processed by the IP stack; if not, the packets are discarded.

User generated multicast streams are supported by the proposed solution by allowing source users to obtain VEKs for their content, from the GC, and to distribute them to restricted groups of other users, which are a subset of the remaining users. In turn, the GC will trigger the update of the user's multicast profile stored at the AAA.

This chapter also details the implementation process, the prototypes developed and the main data structures used in GC, STB, VS and MC. Prototypes were developed to validate both performance and functionality. These experiments were grouped in two: signalling and admission control, which led to the development of two prototypes, one for each group of experiments.

Chapter 5

Validation

The solution proposed in this thesis was validated with respect to performance and functional aspects. When considering the expected duration of the three different types of keys (VEK, KEK and SEK), SEK are the keys less often refreshed and are issued per STB. We consider SEK the most critical type of key in terms of confidentiality assurance, which led us to the security analysis of the STB Bootstrap phase. Two groups of experiments were planned: signalling and admission control. A prototype was developed for each group of experiments and Figure 5.1 shows the functional blocks used in each prototype. The signalling related experiments are fourfold: stressing the GC regarding the STB Bootstrap phase; stressing the GC regarding the KEK Request phase; analysis of the time required by an STB to obtain both KEK and VEK; and evaluation of the overhead introduced by the proposed solution to the VS. The admission control related experiments focused on the verification of both the behavior and the performance of the MC element.

The scalability of the proposed solution was evaluated by means of simulation. STB Bootstrap and KEK Request phases use unicast connections, meaning that the signalling requirements of these phases will grow linearly with the group size. The STB Bootstrap phase is assumed not to occur frequently; on the other hand, the KEK Request will occur whenever an STB switches to a video channel that belongs to a different bundle (out-of-bundle zapping). *Zapping* situations that require the STB to obtain both VEK and KEK are expected to be significantly less than zapping situations where the STB already possess the KEK. Note that, while each video chan-

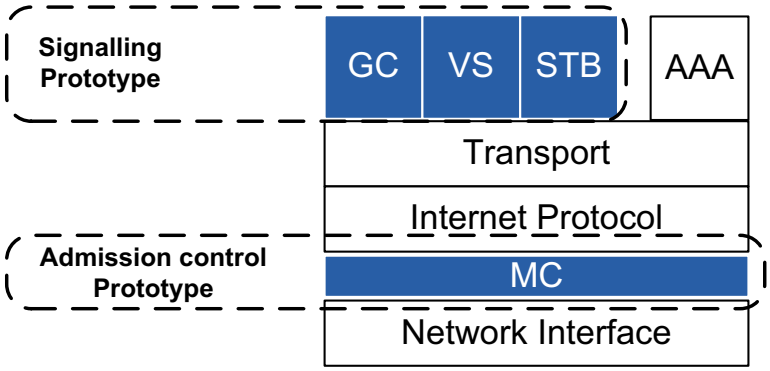

Figure 5.1: Elements used in experimentation

nel is encrypted with a different VEK, all VEK announcements for all video channels that constitute a bundle will share the same KEK. Moreover, STBs are allowed to cache KEKs until they expire. The signalling requirements of the proposed solution were accounted for all these scenarios.

A comparison with current solutions is also presented with respect to signalling requirements; in particular, the signalling requirements of scenarios where multiple STB switch rapidly between multiple video channels.

This chapter consists of five sections. Section 5.1 addresses the security analysis of the STB bootstrap phase. Section 5.2 describes the experimental results obtained using prototypes. Section 5.3 presents and discusses the results related to the scalability and signalling of the proposed solution, obtained by simulation. Section 5.4 compares the proposed solution with current solutions. Section 5.5 discusses confidentiality requirements and the impact of packet loss on the proposed solution.

5.1 Security analysis

The Automated Validation of Internet Security Protocols (AVISPA) [14] tool was used to perform the security validation of the STB Bootstrap phase of the proposed solution.

5.1.1 AVISPA

The AVISPA tool enables the automated validation of security protocols described in High Level Protocol Specification Language (HLPSL) [27].

AVISPA converts the HLPSL specification language to an intermediate format, usable by multiple verification tools embedded in AVISPA. HLPSL, in turn, has the expressiveness required to describe both the protocol behavior and the security properties it must satisfy, such as secrecy and authentication.

The attacker model adopted by AVISPA is the Dolev-Yao intruder model [35]. This model is characterized by the intruder being in complete control of the network, meaning that the intruder is capable of intercepting all the messages in the network, replaying previous messages, and generating his own messages based on any part of the intruder knowledge.

The verification techniques supported by AVISPA are fourfold: 1) On-the-Fly Model Checker (OFMC); 2) Constraint-Logic based ATtack SEarcher (CL-AtSe); 3) SAT based Model-Checker (SATMC); 4) Tree Automata based Protocol Analyser (TA4SP) [20].

The OFMC [17] technique models the protocol as a transition system, where the states are represented by the states of honest participants plus the intruder knowledge, and state transitions are triggered by actions of honest participants and of the intruder. The security properties, formalized as predicates characterizing unsafe states, are evaluated after each transition. The CL-AtSe [79] technique verifies, after each protocol transition, that the security properties are not compromised by imposing constraints over the intruder knowledge. The SATMC [15] technique creates a propositional formula encoding possible attacks on the protocol and validates it using a SAT solver. The TA4SP [20] technique validates security protocols by over-estimating or under-estimating the intruder knowledge through the use of regular tree description languages, and then by checking on the reachability of such states.

5.1.2 HLPSL specification

The automated security verification with AVISPA requires the specification of the environment and the specification of security goals. The environment in HLPSL consists of a set of protocol sessions, each session is described by the involved participants and their shared knowledge, if any. The security goals supported by HLPSL are secrecy and authentication.

```
role environment()                                                    1
```

```
  def=  const sec_sek, auth_sek : protocol_id,                    2
        h    : hash_func,                                         3
        a, b     : agent,                                         4
        kab, kib  : symmetric_key                                 5
  intruder_knowledge = { a, b, kib }                              6
  composition                                                     7
        session(a,b,kab,h)                                        8
    /\  session(a,i,kib,h)                                        9
    /\  session(i,b,kib,h)                                        10
end role                                                          11
```

Listing 5.1: Environment specification

In our specification (Appendix A), the environment describes three sessions, as shown in the lines 8, 9, and 10 of Listing 5.1. One session is the legitimate session (line 8), involving only honest participants (a and b), while in the other two sessions the intruder impersonates either of the honest participants (lines 9 and 10). As initial knowledge we assume the intruder knows the honest participants, and that has a cryptographic key that was pre-shared with the GC. In this way, it is possible to verify the protocol security even when the intruder is a legitimate user, but tries to impersonate other users.

```
role stb(                                                         1
...                                                               2
     /\ request(A,B,auth_sek,Sek')                                3
end role                                                          4
role group(                                                       5
...                                                               6
     /\ witness(B,A,auth_sek,Sek')                                7
     /\ secret(Sek',sec_sek,{A,B})                                8
end role                                                          9
...                                                               10
goal                                                              11
  secrecy_of sec_sek                                              12
  authentication_on auth_sek                                      13
end goal                                                          14
```

Listing 5.2: Security goals specification

Listing 5.2 is an excerpt of our protocol specification that shows our definition of security goals. These goals are the secrecy of SEK (lines 8 and 12), and the ability of SEK to serve as an authentication token (lines

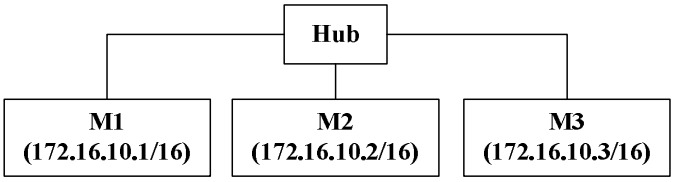

Figure 5.2: Test bed used in the experimental results

	Average	Std. Dev.
AES-128	20814	345
AES-192	20121	356
AES-256	19308	754

Table 5.1: KEK Requests processed per second

3, 7, and 13) between the participating entities. The secrecy (line 12) says that any time the intruder obtains the SEK, and it is not an explicit secret between the intruder and the GC, then we are in presence of an attack. The authentication goal (line 13) is used to verify that an STB is right in believing that its GC has reached a specific state, associated with the current session, and that GC agrees on that specific SEK.

We performed the security verification with all four techniques available in AVISPA. None of them was able to find an attack to our protocol. For the specific case of the TA4SP technique, the result was considered inconclusive: when executed by under approximation, the protocol is reported as unsafe because the intruder may know some critical information but it is unable to identify which critical information; when executed by over approximation, the TA4SP reports a safe protocol.

5.2 Experimental results

5.2.1 Signalling

The testbed implemented and used to obtain experimental results is represented in Figure 5.2. It consists of 3 computers interconnected through an Ethernet hub. All the computers have the same hardware and software characteristics, namely the Fedora Core Linux operating system, an 3000+

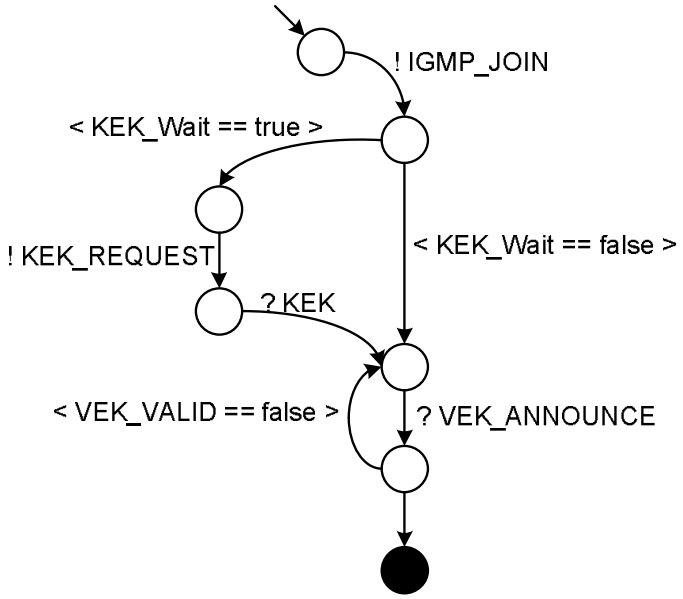

Figure 5.3: Third experiment state machine

VEK Re-key interval (ms)	VEK Processing time		SRTP Packets transmitted (average)
	Average	Std. Dev.	
500	972.79 ms	3.61 ms	55 packets
100	131.66 ms	3.70 ms	11 packets

Table 5.2: VEK Processing time (ms)

AMD Athlon 64 processor and 1GB of RAM memory.

The first experimental test addresses the STB bootstrap phase. The GC was setup in M1 and the STB in M2. The test consists in stressing the GC regarding the bootstrap phase, that is, in stressing the process of mutual authentication and SEK exchange between GC and STB. As a result, we observed that the GC was able to correctly process on average 839 requests per second.

The second experimental test had as objective to stress the GC in the KEK Request phase. This procedure consists in the reception of the KEK request message by the GC, extraction of STB identification, request validation and authentication and, upon successful authentication, construction and transmission of the KEK reply message. The STB was setup in M2 and sends periodic KEK requests. The GC was setup in M1. Each test was

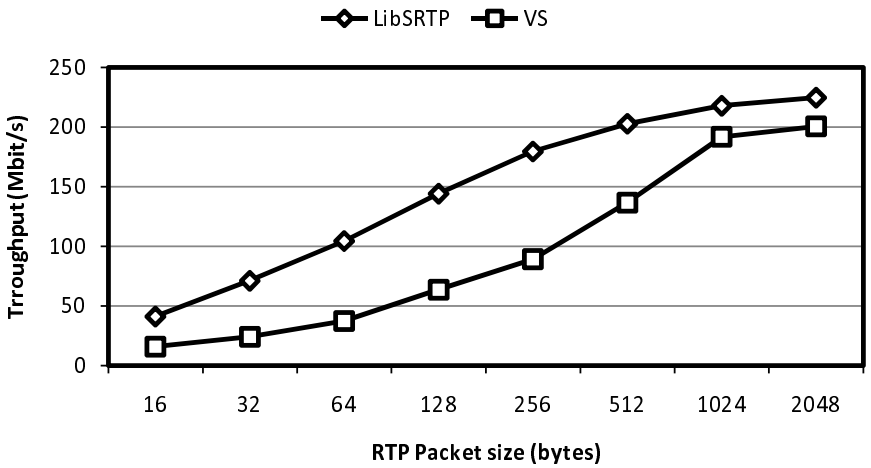

Figure 5.4: Video Server throughput

repeated three times using different key sizes of the Advanced Encryption Standard (AES) algorithm. Table 5.1 shows the results obtained. The GC was able to process on average 20,814 KEK requests per second when using 128 bit AES encryption, 20,121 KEK requests per second when using 192 bit AES encryption, and 19,308 KEK requests per second when using 256 bit AES encryption. The average time for an STB to obtain a KEK reply was 7 ms.

The third experiment focused on the time required by an STB to obtain both KEK and VEK, which are both required to decrypt a video channel. VEK announcement intervals of 500 and 100 ms were assumed. A simplified state machine of this procedure in STB_i is shown in Figure 5.3, and it assumes that the STB is already in possession of its SEK. Here, ! and ? represent respectively the transmission and the reception of messages. The GC was setup in M2, the STB was setup in M3, and the VS in M1. Each test was executed 8 times for each VEK re-key interval. The results obtained are presented in Table 5.2, where the values represent time, in ms, since the STB switches for a new channel (join operation) until it receives the VEK and it can start decrypting the new TV channel. From Table 5.2 we can also observe that for re-key intervals of 500 and 100 ms, a mean number of 55 and 11 SRTP packets are transmitted, respectively.

The last experiment aimed at evaluating the VS overhead introduced by the proposed solution. For that purpose, the throughput of the native

libSRTB was measured and compared with the throughput obtained using our solution. The results obtained are shown in Figure 5.4 and they show that, for typical packets of 1500 bytes, the differences of throughput are of about 10%, in favour of our solution. Our solution uses smaller keys that, in turn, are renewed frequently (every 100 or 500 ms).

5.2.2 Admission control

In order to verify the basic functionalities of the proposed solution, an MC daemon was developed and implemented in a test bed consisting of 3 computers. One computer acted both as Access Node/Router (AR) and AAA server. The other two acted as users' machines, each having a different connection type, PPPoE and 802.1X. The main functionalities implemented in this prototype were the following: user authentication; detection of join/leave messages; detection of multicast source transmission; multicast authorization permission checks; and unauthorized multicast traffic filtering.

Tests were executed to verify both the behaviour and the performance of the MC. The first type of tests focused on the functional validation of the proposed solution, which included the system basic connectivity and the MC's behaviour in the following use cases: authorized/unauthorized group join request; multicast transmission to an authorized/unauthorized multicast group; and unauthorize a previously authorized source/member.

In these tests one user acted as a source and the other as the receiver (group member). All the tests were successfully concluded, meaning that users were correctly authenticated on network attachment, and only when their respective access permissions allowed them they could access or transmit multicast content.

The second type of tests aimed at evaluating the performance of the proposed solution. In this case, two users repeatedly sent IGMP join requests, at the highest possible rate, towards the AR. Small modifications were made to the MC so it would always process the join request, namely by not creating both the white and black lists. The MC's maximum processing rate was 1250 IGMP requests per second.

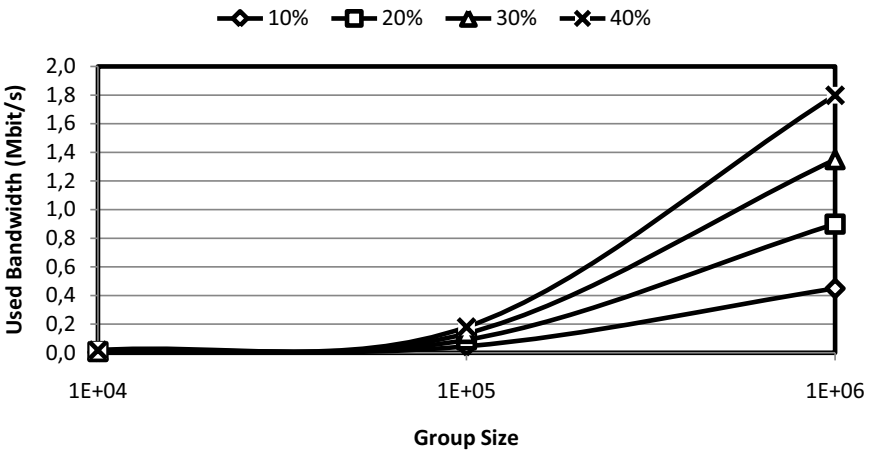

Figure 5.5: Signalling represented as a function of group size and percentage of bootstrapping members

5.3 Simulation results

Figure 5.5 shows the bandwidth used in signalling for different percentages of bootstrapping members. Results are shown for a group size of up to one million members and for percentages of bootstrapping member of 10%, 20%, 30%,and 40%. A logarithmic scale is used. For instance, for a group of one million members, of which 40% are bootstrapping, our solution will consume approximately 1.8 Mbit/s in signalling. The STB bootstrap is the procedure that consumes most of the signalling, due to the need of a secure protocol for authentication and SEK exchange. The SEK are the keys that are refreshed less frequently. We assumed a SEK refresh period of at least one week, analogous to typical subscription periods of IPTV services.

KEK refresh is the operation that may affect the performance of our solution, since it is carried by a unicast UDP packet per subscriber and per KEK re-key interval; the bandwidth required for KEK re-key grows linearly with the group size. Figure 5.6 presents the cost, in terms of network resources, of the KEK refresh operation, normalized to a video channel bandwidth (4 Mbit/s). Results are shown for group sizes up to one million members and for KEK re-key intervals of 5, 10, 15, 30, 60 and 1440 minutes. A logarithmic scale is used. For a group size of one million members, and for a KEK re-key interval of one day (1440 minutes), the bandwidth usage will be 0.1% of one video channel.

A KEK request may also be required in zapping situations, meaning that

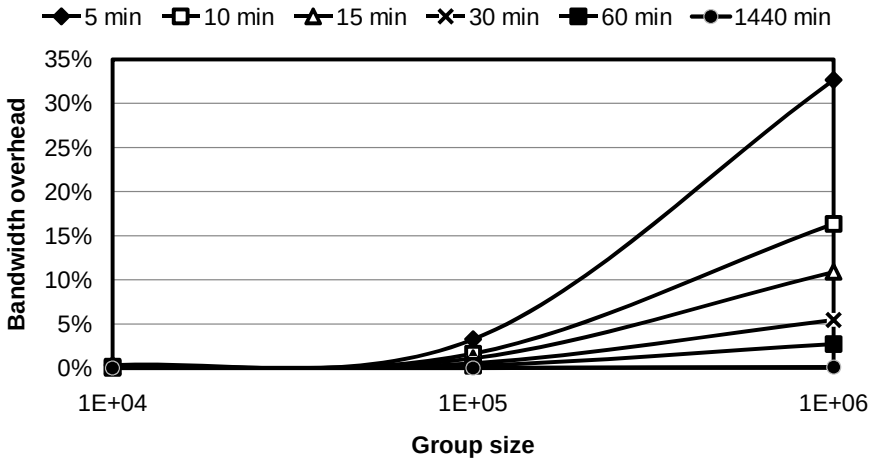

Figure 5.6: Bandwidth overhead, normalized to one video channel bandwidth, assuming different KEK refresh intervals

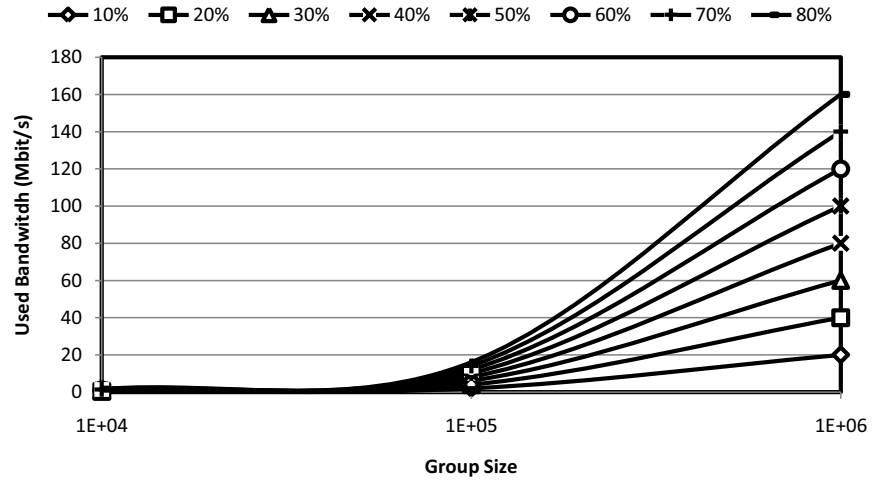

Figure 5.7: KEK+VEK signalling represented as a function of group size and percentage of *zapping* members. In this case a *zapping* STB needs to obtain both KEK and VEK

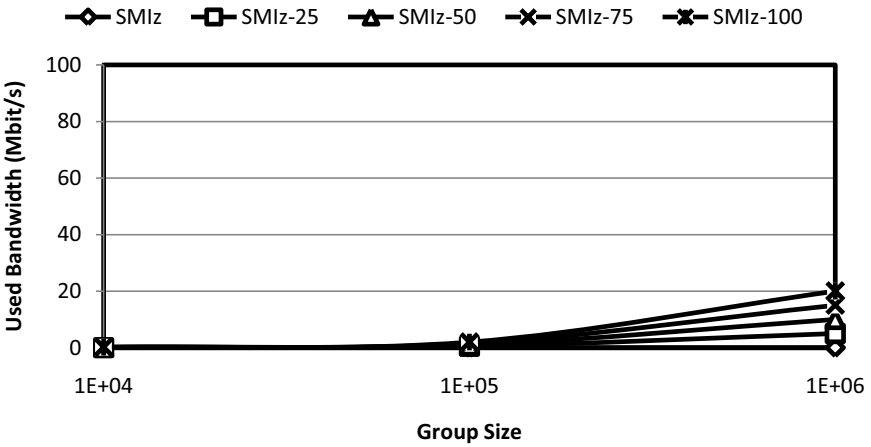

Figure 5.8: KEK signalling represented as a function of group size and percentage for 10% of *zapping* members

if an STB switches to a video channel that belongs to a different bundle, it must obtain both VEK and KEK in order to be able to correctly decrypt the video channel. Figure 5.7 presents the cost of such KEK requests represented as a function of group size and percentage of members zapping through the multiple video channels. Results are shown for a group size up to one million members and for percentages of members zapping of 10%, 20%, 30%, 40%, 50%, 60%, 70%, and 80%. A logarithmic scale is used. For instance, for a group of one million members, of which 10% are switching channels, our solution will consume approximately 20 Mbit/s in signalling. For a percentage of 80%, and a group of one million users, the signalling required amounts to 160 Mbit/s.

Zapping situations that require the STB to obtain both VEK and KEK are expected to be significantly less than zapping situations where the STB already possess the KEK. While each video channel is encrypted with a different VEK, all VEK announcements for all video channels that constitute a bundle will share the same KEK. Assuming the existence of a "classic" bundle that holds the higher number of video channels, it is expected that most video channel switches will only require the reception of the VEK announcement of the new video channel.

Figures 5.8 to 5.11 show the bandwidth used in signalling generated by zapping channels. It is represented as a function of group size and percentage of members zapping through the multiple video channels. The SMIz curve

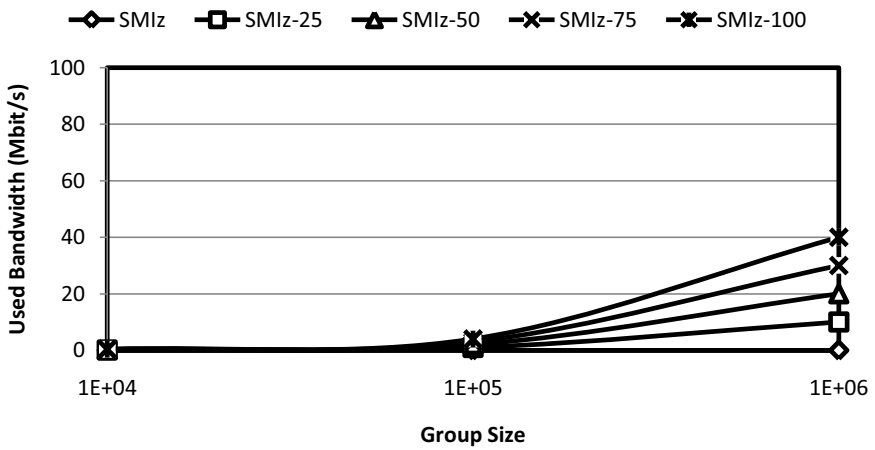

Figure 5.9: KEK signalling represented as a function of group size and percentage for 20% of *zapping* members

represents the signalling used in *zapping* when only the reception of the VEK announcement is required for 10% (Figure 5.8), 20% (Figure 5.9), 30% (Figure 5.10), and 40% (Figure 5.11) of the group size. The SMIz-25 curve represents the signalling used in *zapping* when both VEK and KEK are required for 25% of the *zapping* members in each case. The SMIz-50 curve represents the signalling used in *zapping* when both VEK and KEK are required for 50% of the *zapping* members. The SMIz-75 curve represents the signalling used in *zapping* when both VEK and KEK are required for 75% of the *zapping* members. The SMIz-100 curve represents the signalling used in *zapping* when both VEK and KEK are required for all the *zapping* members.

For instance, for a group of one million members, if 10% of the users are switching channels (Figure 5.8) and 25% of the zapping users require both VEK and KEK (SMIz-25 curve), this scenario will consume approximately 5 Mbit/s in signalling. For a percentage of 75% of members requiring both VEK and KEK (SMIz-75 curve), and a group of one million users, the signalling required amounts to 15 Mbit/s.

For a group of one million members, if 20% of the users are switching channels (Figure 5.9) and 25% of the zapping users require both VEK and KEK (SMIz-25 curve), such will consume approximately 10 Mbit/s in signalling. For a percentage of 75% of members requiring both VEK and KEK (SMIz-75 curve), and a group of one million users, the signalling required

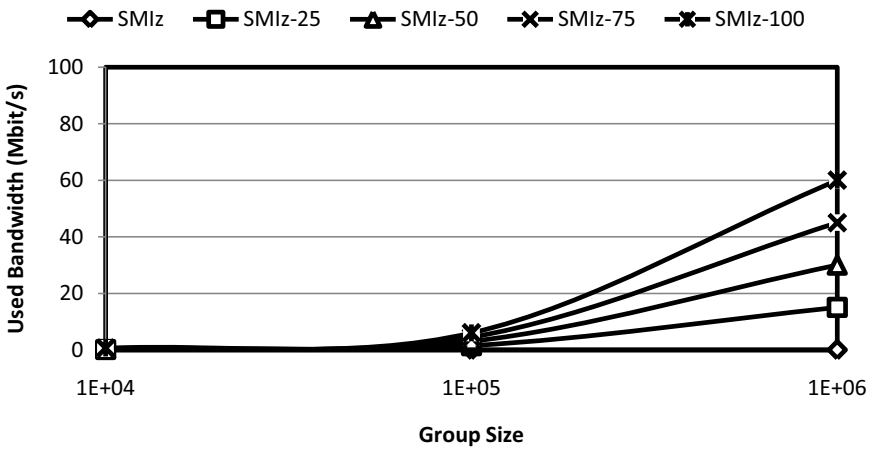

Figure 5.10: KEK signalling represented as a function of group size and percentage for 30% of *zapping* members

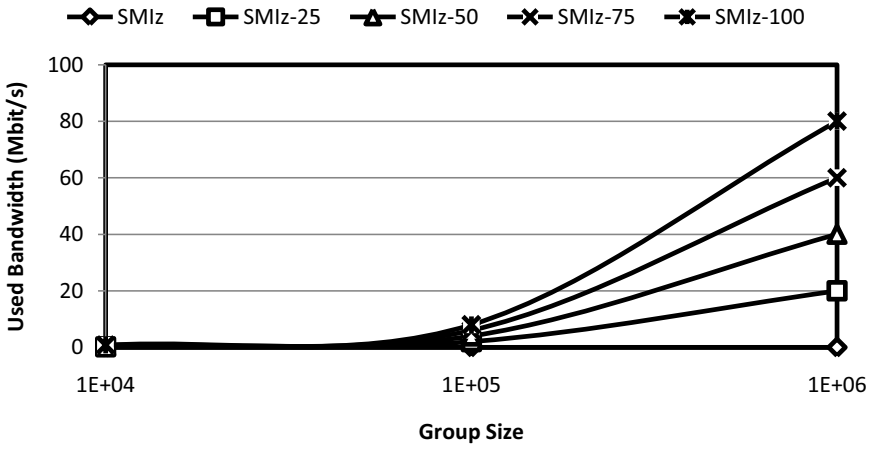

Figure 5.11: KEK signalling represented as a function of group size and percentage for 40% of *zapping* members

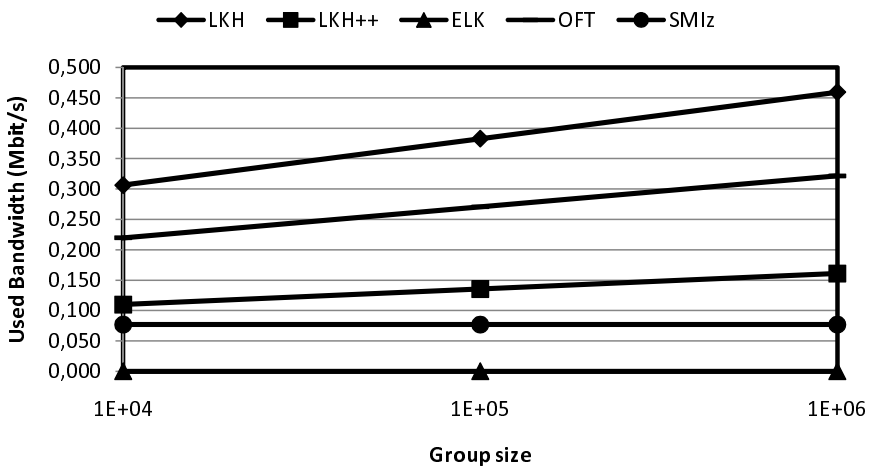

Figure 5.12: Bandwidth used in VEK re-key operations

amounts to 30 Mbit/s.

For a group of one million members, if 30% of the users are switching channels (Figure 5.10) and 25% of the zapping users require both VEK and KEK (SMIz-25 curve), such will consume approximately 15 Mbit/s in signalling. For a percentage of 75% of members requiring both VEK and KEK (SMIz-75 curve), and a group of one million users, the signalling required amounts to 45 Mbit/s.

For a group of one million members, if 40% of the users are switching channels (Figure 5.11) and 25% of the zapping users require both VEK and KEK (SMIz-25 curve), such will consume approximately 20 Mbit/s in signalling. For a percentage of 75% of members requiring both VEK and KEK (SMIz-75 curve), and a group of one million users, the signalling required amounts to 60 Mbit/s.

5.4 Comparison

Figure 5.12 compares our solution with the centralized solutions identified in literature that require less bandwidth in signalling, the main goal of our solution. This comparison is made in terms of the bandwidth used in VEK re-key operations during video channel visualization. The bandwidth used in the member's bootstrap and the first group join operations are not considered. The SMIz and ELK solutions lead to a constant bandwidth usage or, in other words, the required bandwidth does not grow with group size.

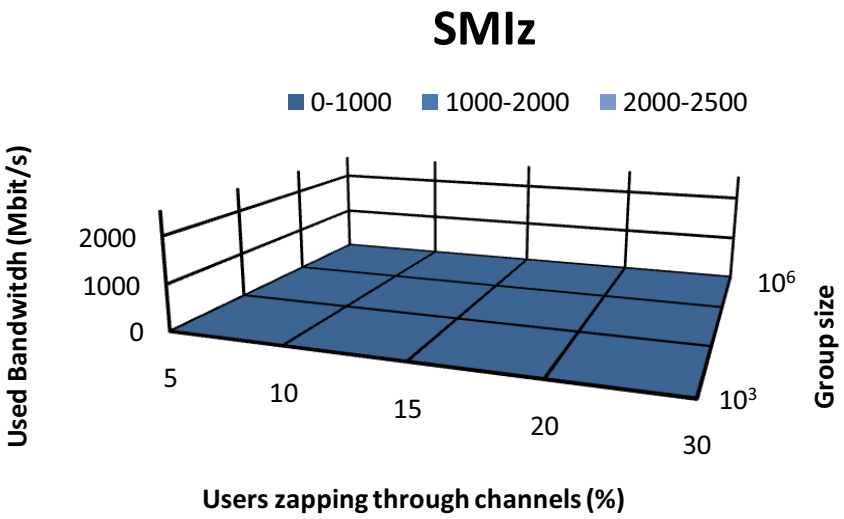

Figure 5.13: SMIz signalling represented as a function of group size and percentage of *zapping* members

ELK does not demand traffic in this situation because the new keys are obtained by each member by computing Pseudo-Random Functions (PRF) over the tree information in their possession. SMIz does not require multiple PRF computation at each member per refreshed key but it enables key independence, since future keys do not depend on previous keys.

Figures 5.13 to 5.16 show the bandwidth used in signalling generated by zapping channels. It is represented as a function of group size and percentage of members zapping through the multiple video channels. In Figure 5.14, for instance, a group of one million members, of which 30% are switching channels, consume approximately 803 Mbit/s in this signalling. The bandwidth required for signalling in all solutions, except ours, grows with group size. It was assumed a tree height equal to $log_2(n)$, n being the group size.

Figure 5.17 shows the bandwidth spent in signalling when 10% of the existing STBs are zapping through channels. Our solution (the SMIz curve) is characterized by constant and low signalling in scenarios where the users zapp through channels in the same bundle. In Figure 5.17 we have also considered two scenarios where both VEK and KEK re-keys were required. These scenarios appear when users switch between channels belonging to different bundles. The results obtained when 20% of the zapping users switch between different bundles is shown in curve SMIz-20; the curve SIMz-80

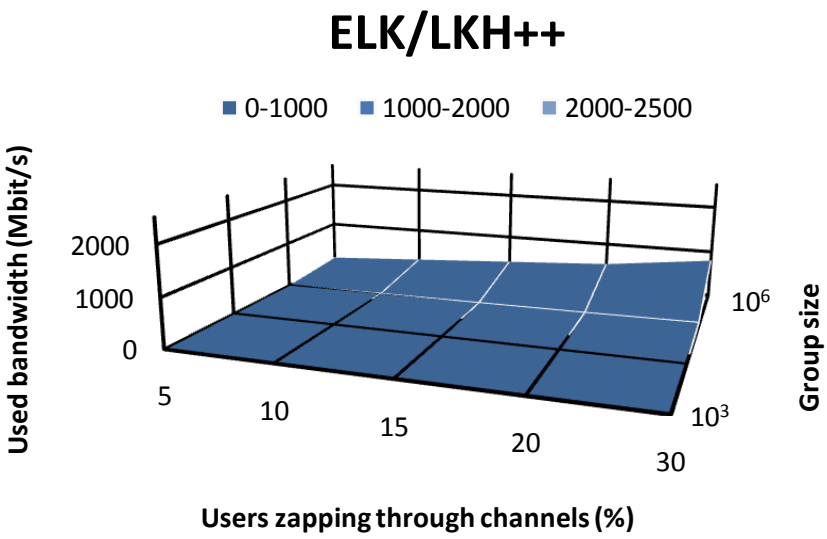

Figure 5.14: ELK/LKH++ signalling represented as a function of group size and percentage of *zapping* members

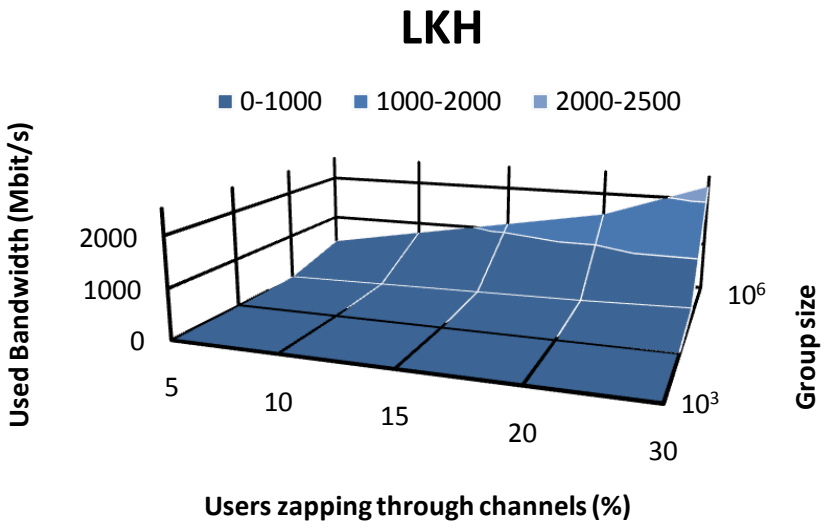

Figure 5.15: LKH signalling represented as a function of group size and percentage of *zapping* members

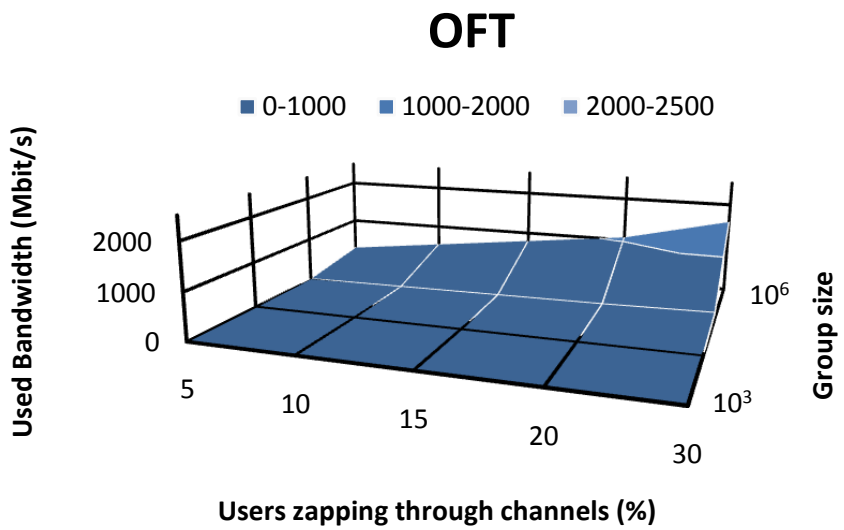

Figure 5.16: OFT signalling represented as a function of group size and percentage of *zapping* members

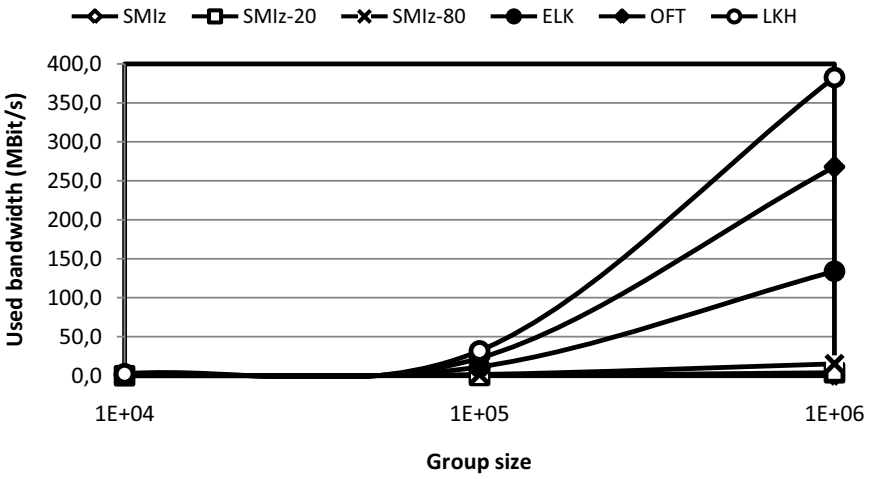

Figure 5.17: Signalling represented as a function of group size

shows equivalent results for 80% of zapping users switching between different bundles. The bandwidth required is represented as a function of group size for an average channel viewing time of 2 seconds. For instance, for a group of one million members, of which 10% are switching channels, our solution will consume approximately 39.3 kbit/s in signalling for the SMIz scenario (VEK re-key only), 3.8 Mbit/s for the SMIz-20 scenario, and 15.3 Mbit/s for the SMIz-80 scenario. The bandwidth required for signalling in all the other solutions grows with group size, except ours when only VEK re-key is required.

Our solution requires significantly less bandwidth than all the others but ELK solution. ELK does not require signalling in VEK re-key operations and, for this reason, has not key independence. ELK derives new VEK keys from past ones, meaning that upon a key disclosure, all past and forward group communications may be compromised. In our solution, each video channel is transmitted to a different multicast group address and it is secured by a different VEK. Switching from one channel to another implies a group leave operation (from the current channel), a group join operation (to the new channel), and re-key operations. All the other solutions require KEK related signalling in group join and leave operations, except ours for the majority of video channel switches.

5.5 Discussion

5.5.1 Confidentiality requirements

The secure multicast confidentiality requirements, identified in Section 3.3, are non-group confidentiality, forward secrecy, backward secrecy, and collusion resistance. The non-group confidentiality is achieved by not distributing cryptographic material to non-members. Moreover, the proposed solution addresses also the case when an ill-intentioned but valid group member distributes valid cryptographic material to non-members. In this case, the non-group confidentiality is achieved by enforcing multicast admission control and thus not allowing unauthorized extensions of the multicast distribution trees.

Forward and backward secrecy are achieved by periodically refreshing cryptographic material. It does not enable perfect forward secrecy since a group member, after leaving a group, will still be able to decrypt the

content until the end of the key refresh period. The same applies to perfect backward secrecy; a group member, after joining a group, will be able to decrypt previous communications up to the previous key refresh instant. Nevertheless, IPTV services are usually based on monthly subscription fees and, as long as an user maintains an active subscription, leaving a video channel group does not mean that the user is not entitled to access the content. Perfect forward or backward secrecy are not, from our point of view, applicable to IPTV services.

Collusion resistance is also a feature of the proposed solution. Not being collusion resistant means that non members are able to derive cryptographic material. Our solution exchanges securely SEKs that are used to secure the following cryptographic material exchange. Moreover, the cryptographic material used is not derived from previous cryptographic material nor from any other previously exchanged information.

5.5.2 Packet loss

Packet loss may impact on the user's QoE. In particular, when a significant number of data packets are lost, the STB may not receive complete video frames in time. In this case, the STB will drop the video frames and the video playback quality will suffer significantly. Packet loss is normally attributed to the network, due to traffic congestion at one or more network equipments or to networks being under provisioned in terms of bandwidth.

In this work we assume that networks are provisioned with the resources required to cope with a multi-channel real-time IPTV service. Permanent packet loss problems related to under provisioned networks are not expected. If these occasional problems are not short-timed, then the network infrastructure should be improved. On the other hand, occasional and short-timed packet loss problems may occur.

Short-timed packet losses, for a period of one second, would imply the loss of about 50 data packets and 10 VEK announcements. If this period overlaps with a VEK re-key period, the STB would correctly receive the data packets after the short loss period, but would be unable to decrypt these packets until it receives the next VEK announcement. Thus, the frequent transmission of the VEK announcements (10 per second) enables STBs not only to rapidly switch video channels, but it can also be considered as a technique to redundantly provide VEKs to the STB.

5.6 Summary

This chapter describes the validation of the proposed solution. In particular, when considering the expected duration of the three different types of keys (VEK, KEK and SEK), the SEK are the keys less often refreshed and are issued per STB. We consider the SEK as the most critical type of key in terms of confidentiality assurance, which lead us to the security analysis of the STB Bootstrap phase. We used AVISPA, an automated security validation tool, and performed the security verification with all four techniques available. None of them was able to find an attack to our protocol. For the specific case of the TA4SP technique, the result was considered inconclusive: when executed by under approximation, the protocol is reported as unsafe because the intruder may know some critical information but AVISPA is unable to identify which; when executed by over approximation, the TA4SP reports a safe protocol.

The elements of the proposed solution were also validated with respect to both performance and functionality by means of experiments carried out over prototypes. These experiments were grouped in two: signalling and admission control. Two different prototypes were developed, one for each group of experiments. The signalling related experiments are fourfold: stressing the GC regarding the STB Bootstrap phase; stressing the GC regarding the KEK Request phase; analysis of the time required by a STB to obtain both KEK and VEK; evaluation of the overhead introduced by the proposed solution to the VS. The GC was able to correctly process an average 839 STB bootstrap requests per second. The GC was able to process an average of 20,814 KEK requests per second, resulting in an average delay of 7 ms for a STB to obtain a KEK reply. The time required by a STB to obtain both KEK and VEK was 131.66 ms for a re-key interval of 100 ms. The overhead introduced to the VS by the proposed solution, for typical packets of 1500 bytes, is about 10%. The admission control related experiments focused on the verification of both the behaviour and the performance of the MC element. The MCs maximum processing rate was 1,250 IGMP requests per second.

The scalability of the proposed solution was evaluated by means of simulation. STB Bootstrap and KEK Request phases use unicast connections, meaning that the signalling requirements of these phases will grow linearly

with the group size. For instance, for a group size of one million members (10^6), and for a KEK re-key interval of one day (1440 minutes), the bandwidth usage will be of 0.1% of a video channel. While the STB Bootstrap phase is assumed not to occur frequently, the KEK Request will occur whenever a STB switches to a video channel that belongs to a different bundle (out-of-bundle zapping). A group size of one million members, of which 10% are switching channels, will consume approximately 20 Mbit/s in signalling.

Zapping situations that require the STB to obtain both VEK and KEK are expected to be significantly less than zapping situations where the STB already possess the KEK. Note that, while each video channel is encrypted with a different VEK, all VEK announces for all video channels that constitute a bundle will share the same KEK. For instance, for a group of one million members, if 10% of the users are switching channels and 25% of the zapping users require both VEK and KEK, approximately 5 Mbit/s in signalling will be required. For a percentage of 75% of members requiring both VEK and KEK, the signalling required for the same group size amounts to 15 Mbit/s.

A comparison with current solutions is also presented with respect to signalling aspects; in particular, the signalling requirements of scenarios where multiple STB switch rapidly between multiple video channels. The bandwidth required for signalling in all the other solutions grows with group size, except ours when only VEK re-key is required. Our solution demands significantly less bandwidth than all the others but ELK solution. ELK does not require signalling in VEK re-key operations and, for this reason, has no key independence what implies that, upon a key disclosure, all future keys are compromised.

Chapter 6

Deployment scenarios

Current IPTV services rely on IETF standardized protocols such as RTP for transport, IGMP or SIP for signalling, RADIUS for AAA, and 802.1X or PPP for network attachment. On the other hand, these protocols do not completely satisfy the needs of IPTV service operators, in particular when considering functionalities such as content confidentiality, content access control and multicast session management. The lack of such functionalities has led to the creation of interest groups focused on the development of solutions that complement the IETF standardized protocols and integrate the IETF multimedia architecture in the telecom operator networks. ITU-T is an organization addressing this problem.

The architectural approaches recommended by ITU-T [7] for IPTV service deployment are threefold: 1) Non-NGN IPTV Functional architecture; 2) NGN-based non-IMS IPTV Functional architecture; 3) NGN IMS-based IPTV Functional architecture. The first architecture is based on existing network components and protocols, where these adopted network components, protocols and interfaces are already in use, hence considering existing IPTV services. The second architecture adopts components from the NGN architecture [6] in order to enable the deployment of IPTV services in NGN. The third architecture adopts the components of NGN and includes the IMS components in order to support the deployment of IPTV services in conjunction with other IMS services. The ITU-T Y.1910 Recommendation identifies the functions, functional blocks and interfaces required for an IPTV service. Besides the IETF standard protocols, namely RTP, IGMP, and SIP, it does not specify the majority of the interfaces between functional blocks; these

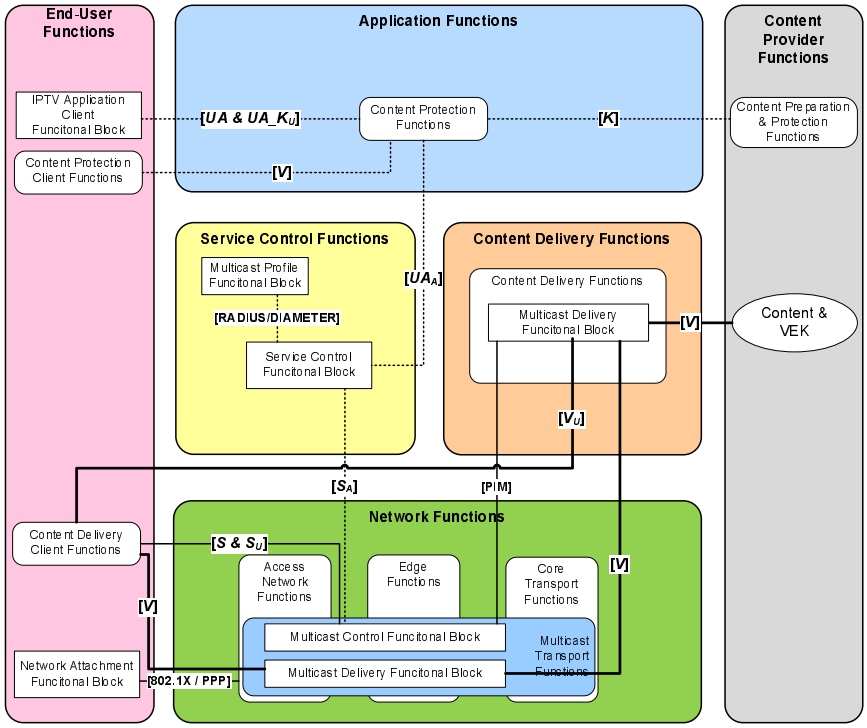

Figure 6.1: Architecture of the proposed solution

are still marked for further studies.

We claim that the solution proposed in this thesis can be used as a basis for the ITU-T IPTV service. This is particularly relevant for some of the communications interfaces. Moreover, some of the functionalities of our proposed solution are either considered out of scope or not addressed at all by the ITU-T Recommendation, namely content confidentiality or the support for user generated video channels. Out of scope is here interpreted as good news since these functionalities are not forbidden by the current ITU-T recommendation. Figure 6.1 shows the architecture of our proposed solution, adopting a graphical style similar to those used in ITU-T IPTV Recommendations, in order to ease the comparison of the proposed solution and the three IPTV architectural approaches of ITU-T. It consists of Application Functions, Content Provider Functions, End-User Functions, Service Control Functions, Content Delivery Functions and Network Functions; all the components are analogous to those found in ITU-T IPTV Recommendation. Application Functions comprise Content Preparation Functions, which are responsible for interacting with End-User Functions by means of the UA,

UA_K_U and V interfaces. Application Functions also interact with the Service Control Functional Block of the Service Control Functions be means of the UA_A interface, and with the Content Preparation & Protection Functions be means of the K interface. The Content Delivery Client Functions, of the End-User Functions, interact with the Multicast Delivery Functional Block through the V interface, and with the Multicast Control Functional Block through both S and S_U interfaces. The Multicast Control Functional Block also interacts with the Service Control Functional Block, using the S_A interface.

This chapter aims at comparing the proposed solution with the ITU-T IPTV functional architectures. It consists of four sections in which common functions and functional blocks are identified and discussed. Section 6.1 compares the proposed solution with the Non-NGN IPTV Functional architecture; Section 6.2 compares the proposed solution with the NGN-based non-IMS IPTV Functional architecture; Section 6.3 compares the proposed solution with the NGN IMS-based IPTV Functional; Section 6.4 presents the conclusions of these comparisons.

6.1 Non-NGN IPTV Functional architecture

The Non-NGN IPTV Functional architecture uses legacy technologies for the delivery of IPTV services. The support of legacy solutions is also one of the requirements of the proposed solution (see Section 1.1). Figure 6.2 shows the proposed solution architecture, highlighting the functional blocks common to both ITU-T Non-NGN and the proposed solution. The set of common functionalities comprises the SCP Functions, the SCP Client Functions, Content Preparation Functions, and the following functional blocks:

- IPTV Application and IPTV Client Application;
- Multicast Delivery and Multicast Content Delivery Client;
- Delivery Network Gateway;
- Authentication and & IP Allocation;
- IPTV Service Control;
- Service User Profile;

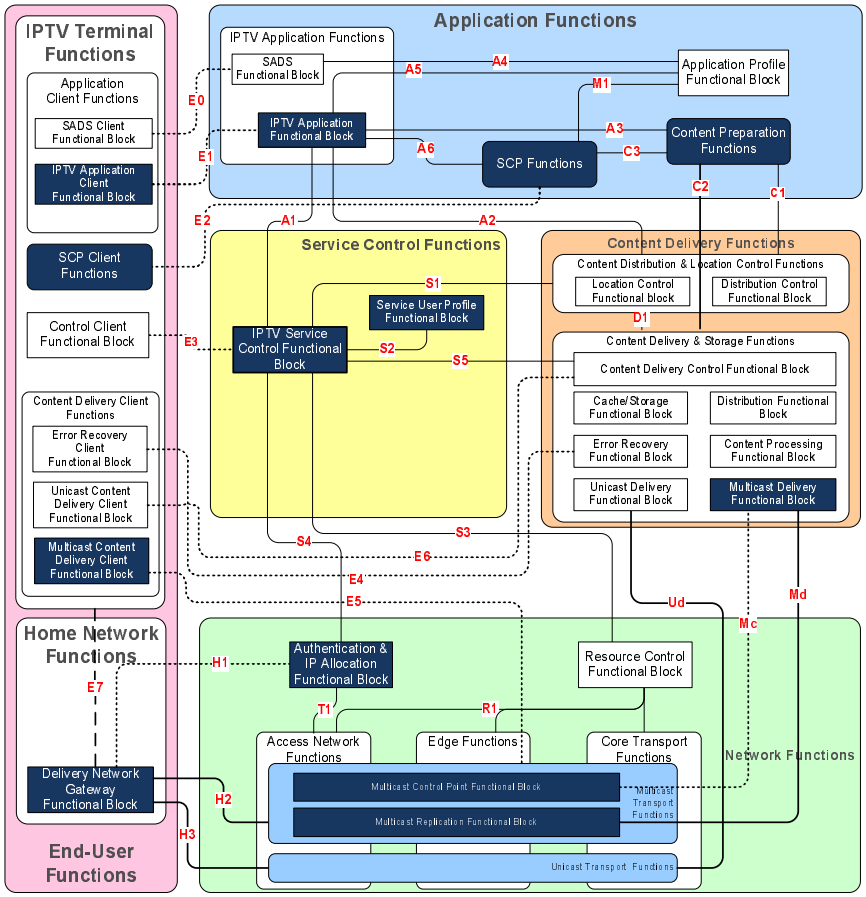

Figure 6.2: Functional blocks common to ITU-T Non-NGN and the proposed solution (in blue)

- Multicast Control Point and Multicast Replication.

The Content Preparation Functions, the SCP Functions, and the SCP Client Functions enable the distribution of VEK to authorized STBs (V interface), and the cryptographic context synchronization between the GC and the VS (K interface). The IPTV Application and the IPTV Application Client Functional Blocks are used in the proposed solution to enable STB bootstrap and video channel requests, which represent the UA interface or the UA_K_U interface, if it is the case of user generated content. The Multicast Delivery and the Multicast Content Delivery Functional Blocks, in conjunction with the Delivery Network Gateway Functional Block, enable the delivery of the video channel multicast packets from VS to STBs.

IPTV Functional Architecture	NGN Functional Architecture
Network functions	Transport stratum
End-user functions	End-user functions
Management functions	Management functions
Service control functions	Service control functions of Service stratum
Application functions	Application support functions & Service support functions of Service stratum

Table 6.1: Relationship between the functions of NGN-based IPTV and NGN architectures

The information stored at Authentication & IP Allocation Functional Block in conjunction with the IPTV Service Control Functional Block, using the Multicast Profiles stored at the Service User Profile Functional Block, enable multicast admission control for both senders and receivers. The Multicast Control Point Functional Block is analogous to the MC of the proposed solution. In the proposed solution, the information required to enforce multicast admission control is exerted from the messages exchanged between the STB and the Network Functions at the moment of network attachment (801.1X or PPP). In the ITU-T Non-NGN IPTV Functional architecture this information will be maintained by the Authentication & IP Allocation Functional Block.

6.2 NGN-based non-IMS IPTV Functional architecture

The relationship between the functions of the ITU-T IPTV Functional architecture and the NGN architecture is summarized in Table 6.1. ITU-T IPTV Network Functions correspond to the NGN Transport Stratum Functions. End-user Functions and Management Functions are analogous in name and functionality in both architectures. ITU-T IPTV Service Control Functions correspond to the Service Control Functions, included in the NGN Service Stratum Functions. In particular, the NGN Service Control Functions may include functionalities other than those of the ITU-T IPTV Service Control Functions [5]. ITU-T IPTV Applications Functions correspond to both Application Support Functions and Service Support Functions, also included in

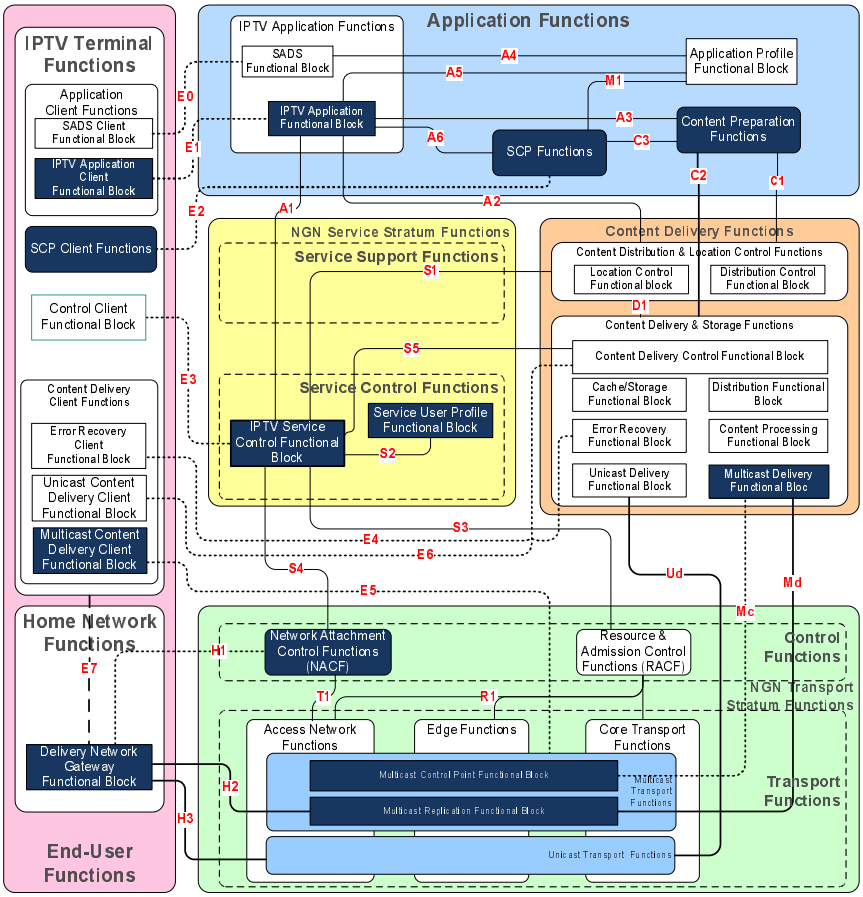

Figure 6.3: Functional blocks common to ITU-T NGN Non-IMS and the proposed solution (in blue)

NGN Service Stratum Functions. The Content Delivery Functions are not specified in NGN functional requirements and architecture [5, 6]; moreover, NGN Content Delivery Functions and NGN Applications Functions may be deployed by a third party service provider.

Figure 6.3 shows the architecture of our proposed solution, highlighting the functional blocks which are common to both the NGN-based non-IMS IPTV and the proposed solution. The set of common functionalities comprises the SCP Functions, the SCP Client Functions, Content Preparation Functions, and the following functional blocks:

- IPTV Application and IPTV Client Application;
- Multicast Delivery and Multicast Content Delivery Client;

- Delivery Network Gateway;
- Network Attachment Control Functions (NACF);
- IPTV Service Control;
- Service User Profile;
- Multicast Control Point and Multicast Replication;

Network Attachment Control Functions (NACF) and Resource and Admission Control Functions (RACF) are the functional blocks that differ from the previous architecture (Non-NGN IPTV Functional architecture). The NACF is a functional block common also to the proposed solution. In particular, it comprises the functions of the Authentication and & IP Allocation Functional Block of the Non-NGN IPTV Functional architecture, which include the STB identification based on information exerted from network attachment protocols (i.e. 802.1X or PPP). Such STB identification is required by the MC in order to impose multicast admission control for both senders and receivers.

6.3 NGN IMS-based IPTV Functional architecture

The NGN IMS-based IPTV Functional architecture uses Core IMS functions to provide service control functions. IMS services are session oriented services and use SIP to impose service deployment and control. Figure 6.4 shows the architecture of the proposed solution, highlighting the functional blocks common to both ITU-T NGN IMS-based IPTV and the proposed solution. The set of common functionalities comprises the Core IMS Functions, SCP Functions, the SCP Client Functions, Content Preparation Functions, and the following functional blocks:

- IPTV Application and IPTV Client Application;
- Multicast Delivery and Multicast Content Delivery Client;
- Delivery Network Gateway;
- Network Attachment Control Functions (NACF);

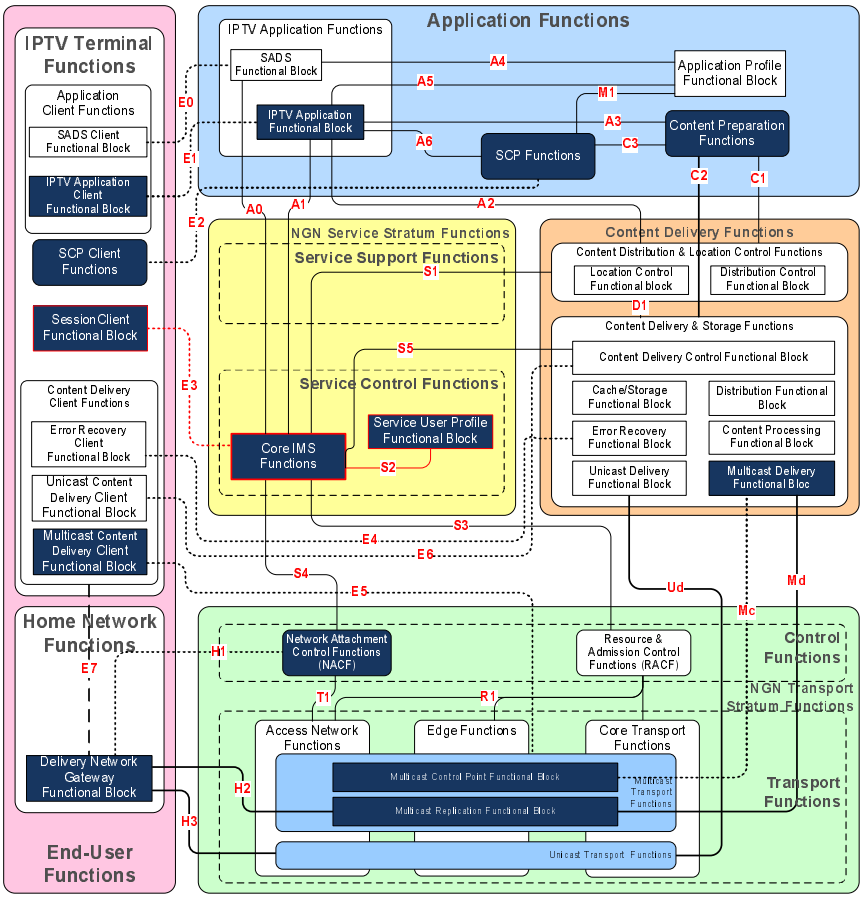

Figure 6.4: Functional blocks common to ITU-T NGN IMS-based and the proposed solution (in blue)

- Session Client;
- Service User Profile;
- Multicast Control Point and Multicast Replication.

Core IMS is the new element and it interacts with RACF to ensure the reservation of resources upon user service request. The user service request is triggered by the Session Client Functional Block, using SIP. In both Non-NGN and NGN non-IMS IPTV architectures, the Control Client Functional Block was the responsible for session establishment, modification, and termination. The proposed solution does not adopt the use of session related signalling besides IGMP/MLD.

Nevertheless, with the addition of the Control Client Functional Block, the STB of the proposed solution would be able to be integrated in a NGN-based IMS IPTV service deployment. In this case, SIP would be used to reserve resources and for accounting; IGMP/MLD signalling would be used to trigger multicast admission control and video channel transmission.

6.4 Summary

Standard protocols such as RTP for transport, IGMP or SIP for signalling, RADIUS for AAA, and 802.1X or PPP for network attachment, are widely used in current IPTV architectures. One of the requirements of the proposed solution is the support for current networks and protocols, so these solutions are part of the proposed solution. Nevertheless, some of the functionalities available in the proposed solution are not currently available or standardized, namely the support for user generated videos, multicast session management, and content confidentiality with efficient video channel zapping. The lack of these and other functionalities has led to the creation of standardization groups focused on the development of solutions that complement the IETF standardized protocols. ITU-T is a key example of such interest groups and it has issued the Y.1910 Recommendation [7] for that purpose.

Our proposed solution was compared to the three IPTV architectural approaches identified in [7]. When compared to the Non-NGN IPTV Functional architecture, we observed that the additional functions proposed in

this thesis were indeed identified in the Recommendation but not yet specified. Thus, the proposed solution can be seen also as an architectural proposal for these functions.

The NGN-based non-IMS IPTV Functional architecture, which is a service oriented architecture, requires session signalling. The NACF and RACF are new functional blocks and they handle session signalling in order to impose network access control and resource reservation, respectively. The proposed solution does not require the use of session related signalling besides IGMP/MLD signalling; for that reason, the deployment of the proposed solution in an NGN-based non-IMS IPTV scenario is feasible, since the first solution is a subset of the second.

The NGN IMS-based IPTV Functional architecture additionally includes the Core IMS in order to provide service control functions to session oriented services, and it uses SIP as the signalling protocol. The proposed solution does not adopt SIP, or any other session management protocol for session signalling but IGMP/MLD. In order to integrate the proposed solution in the NGN IMS-based IPTV Functional architecture, a Control Client Functional Block must be added to the STB. In this case, the STB would use SIP to ensure resource reservation and accounting, and IGMP/MLD signalling to trigger multicast admission control and video channel transmission.

Chapter 7

Conclusions

The main goal of this thesis was to define a cryptographic key management solution capable of enforcing individual access control to groups of real-time IPTV video channels. This cryptographic key management solution should optimize the use of signalling when users switch between video channels without compromising rapid video channel *zapping* and to optimize the video channel transmission to groups of users that may spawn over heterogeneous access networks (xDSL, WiMAX, and UMTS). Additionally, the solution should support user generated videos and it should be capable of efficiently enforcing IP multicast admission control for both multicast senders and receivers.

The selected reference scenario consists in an IPTV service, where multiple video channels are distributed as IP packets in multicast (one multicast group per video channel) and users subscribe to one or more bundles, such as the "classical bundle", the "movies bundle", or the "sports bundle". By subscribing a bundle, the user is enabled to receive and decrypt all the channels that compose the bundle. Although a bundle is composed of several video channels, each video channel is transmitted to a unique multicast address. Common IPTV services use one key for each bundle; in our solution, we explore a concept such that, besides the bundle key (KEK), each channel will also have one data key (VEK).

7.1 Review of the work

Video transmission over IP consists of a video being played out while other video parts are being received and decoded, thus avoiding a full video download before decoding and visualization. Chapter 2 presents efforts to standardize video streaming over IP, including the functionalities required at network, transport, and session layers. The IETF multimedia architecture has defined, in particular, RTP [73] that enables the transmission of video, voice and multimedia contents in IP packets, along with other protocols for controlling the video streaming. More recently [7, 8], these protocols have been re-used by organizations such as the ITU-T and ETSI to integrate IPTV services in the NGN architecture, defined by TISPAN. Key issues of these ETSI and ITU-T activities are the mobile-fixed service convergence and the optimized transmission of video streams over heterogeneous access networks, namely xDSL, WiMAX and UMTS. Chapter 2 also describes IP multicast and its support in access networks. IP multicast is of particular appeal to IPTV services, since it enables significant savings in terms of network resources by transmitting once for all active receivers. However the optimization obtained by IP multicast becomes compromised by the behaviour of some access network technologies, in particular those not supporting link layer multicast transmissions. This fact led our attention to multicast support in the access networks considered by the NGN architecture, namely xDSL, WiMAX and UMTS. In particular, the multicast related components of each network architecture were characterized.

Chapter 3 addresses the current solutions related to both multicast admission control and to secure multicast. The deployment of IP multicast-based services has not been so widely exploited as expected. One of the reasons behind this limited adoption is the lack of control network operators have over multicast groups [34]. IP multicast has an open group architecture, where any user is free to receive or transmit data from/to a multicast group. Some degree of control over IP multicast groups can be obtained with end-to-end encryption of IP multicast data or IP multicast session access control [53]. While the first solution protects multicast data from unauthorized access and potential eavesdroppers, it does not prevent a user from joining a group to which he does not have access to, thus causing the unnecessary extension of the multicast distribution tree. Alternatively,

and as proposed in this thesis, a combination of both techniques may be adopted. This solution enables multicast data confidentiality and prevents unnecessary extensions to the multicast distribution trees.

Chapter 4 describes the proposed solution, which comprises a GC, a VS, STBs, a MC and an AAA server. The GC generates and distributes cryptographic keys, authenticates STBs, and updates the STB multicast profiles stored in the AAA server. The VS and the STB are responsible for the encryption or decryption of the video channel streams, respectively. The VS also generates and distributes VEKs to valid members. The MC is responsible for the detection of multicast sessions for both sources and receivers, and the subsequent authorization request to an AAA server. The proposed solution uses three types of cryptographic keys: 1) SEKs; 2) KEKs; 3) VEKs. SEKs are used for securing unicast communications between STBs and the GC. VEKs are used to (de)encrypt video channels, each channel having a different VEK. KEKs, one per bundle, are used for securing the transmission of the VEKs. VEK announcements consist in transmitting the VEK encrypted with the respective bundle KEK. The combined usage of KEKs and VEKs allows the periodic and frequent transmission of all VEKs of all video channels (VEK announcements), without requiring significant network resources. Chapter 4 also describes the interfaces between all components of the proposed solution, shows how the proposed solution supports heterogeneous access networks, and describes the implementation of the developed prototypes and the main data structures adopted in the key elements of this work, namely the GC, STB, VS and MC.

Chapter 5 validates the proposed solution. In particular, when considering the expected duration of the three different types of keys (VEK, KEK and SEK), the SEK are the keys less often refreshed and are issued per STB. We consider the SEK as the most critical type of key in terms of confidentiality assurance, which led us to the security analysis of the STB Bootstrap phase. The elements of the proposed solution were also validated with respect to both performance and functionality by means of experiments carried out over prototypes. These experiments were grouped in two: signalling and admission control. Two different prototypes were developed, one for each group of experiments. The signalling related experiments are fourfold: stressing the GC regarding the STB Bootstrap phase; stressing the GC regarding the KEK Request phase; analysis of the time required by an

STB to obtain both KEK and VEK; evaluation of the overhead introduced by the proposed solution to the VS. The GC was able to correctly process an average 839 STB bootstrap requests per second. The GC was able to process an average of 20,814 KEK requests per second, resulting in an average delay of 7 ms for an STB to obtain a KEK reply. The time required by an STB to obtain both KEK and VEK was 131.66 ms for a re-key interval of 100 ms. The overhead introduced to the VS by the proposed solution, for typical packets of 1500 bytes, is of about 10%. The admission control related experiments focused on the verification of both the behaviour and the performance of the MC element. The MCs maximum processing rate was 1,250 IGMP requests per second.

The scalability of the proposed solution was evaluated by means of simulation. STB Bootstrap and KEK Request phases use unicast connections, meaning that the signalling requirements of these phases will grow linearly with the group size. For instance, for a group size of one million members (10^6), and for a KEK re-key interval of one day (1440 minutes), the bandwidth usage will be 0.1% of a video channel. While the STB Bootstrap phase is assumed not to occur frequently, the KEK Request will occur whenever an STB switches to a video channel that belongs to a different bundle (out-of-bundle zapping). For a group size of one million members, of which 10% are switching channels, will consume approximately 20 Mbit/s in signalling.

Zapping situations that require the STB to obtain both VEK and KEK are expected to be significantly less than zapping situations where the STB already possess the KEK. Note that, while each video channel is encrypted with a different VEK, all VEK announcements for all video channels that constitute a bundle will share the same KEK. For instance, for a group of one million members, if 10% of the users are switching channels and 25% of the zapping users require both VEK and KEK, approximately 5 Mbit/s in signalling will be required. For a percentage of 75% of members requiring both VEK and KEK, the signalling required for the same group size amounts to 15 Mbit/s.

A comparison with current solutions is also presented with respect to signalling aspects; in particular, the signalling requirements of scenarios where multiple STB switch rapidly between multiple video channels. The bandwidth required for signalling in all the other solutions grows with group size, except ours when only VEK re-key is required. Our solution requires

significantly less bandwidth than all the others but ELK. ELK does not require signalling in VEK re-key operations and, for this reason, has no key independence and implies that, upon a key disclosure, all future keys will be compromised.

Current IPTV services rely on IETF standardized protocols such as RTP for transport, IGMP or SIP for signalling, RADIUS for AAA, and 802.1X or PPP for network attachment. On the other hand, these protocols do not completely satisfy the needs of IPTV service operators, in particular when considering functionalities such as content confidentiality, content access control and multicast session management. The lack of such functionalities led to the creation of study groups focused on the development of standard solutions that complement the IETF protocols. ITU-T is an example of these groups and recommends three architectural approaches for IPTV service deployment[7]: 1) Non-NGN IPTV Functional architecture; 2) NGN-based non-IMS IPTV Functional architecture; 3) NGN IMS-based IPTV Functional architecture. Chapter 6 shows that the proposed solution may be seen as a partial implementation of these approaches, without modification, except for the third architecture that requires a SIP functional block to be added to the STB of the proposed solution.

7.2 Main Contributions

This thesis provides three original contributions:

1. **A key distribution technique.** A new and secure cryptographic key distribution technique is proposed, which enables receiver access control and rapid video channel switching while using a small amount of resources for signalling. Current key distribution solutions for secure group communications usually apply key refreshing techniques upon a group change (member join or departure) in order to impose both perfect forward and backward secrecy. Frequent key refreshes lead to high usage of network resources. In IPTV, a member's group departure does not mean that the user is not entitled to the video channel; instead, it means that the user has probably switched to a different video channel. The proposed solution explores the concept that, besides the bundle key (KEK), each channel will also have one data key (VEK) that, by being shared by all group members, requires constant

signalling in refresh operation. While not enabling perfect forward and backward secrecy, it enables a significant reduction in signalling in situations where users switch rapidly between channels.

2. **A multicast admission control technique.** A new admission control technique for both multicast senders and receivers is proposed, which enables the management of multicast sessions that spawn over heterogeneous access networks. This technique adopts multicast profiles that specify whether a user is allowed to generate his videos and contains a list of video channels the user is authorized to access. These multicast profiles are stored in an AAA server that responds to queries from the MC. Upon successful verification, the MC will authorize the extension of the multicast tree to the new user. Moreover, the MC can be integrated in the access networks considered by NGN and supports the dynamic configuration required by user generating content over connections with renewable IP configurations assigned by Dynamic Host Configuration Protocol (DHCP).

3. **A key distribution technique for user generated video.** The adaptation of the first contribution to support key distribution for secure video channels sourced at domestic users led to a new and secure key distribution technique. This technique enables a user behaving as a video source to obtain VEKs for his video channels and to distribute these keys to restricted groups of receiver users. The restricted groups are identified by the source user.

7.3 Future work

In the course of the work presented in this thesis, the following limitations were identified and may be assumed as future research directions:

Decentralization

The group key distribution technique of the proposed solution is centralized. Nevertheless, this solution was shown to scale up to one million members without having impact on users rapidly switching between channels. In order to increase the scalability of the proposed solution, the re-key intervals for

both SEK and KEK would have to be extended. Moreover, the service availability of the proposed solution depends on one single entity, the GC.

Scalability and service availability can be improved by decentralizing the GC functionality. Assuming the existence of two GC, then the proposed solution would support two million users, while requiring the same level of network resources and the same re-key intervals.

STB bootstrap

The STB bootstrap uses a key pre-shared between the STB and the GC, to secure the SEK exchange. The STB bootstrap was demonstrated to be secure and the pre-shared key is never transmitted through the network. We assume that the STB is a trusted platform and, thus, the pre-shared key refresh is not required. On the other hand, it requires that such pre-shared keys must be installed into the STBs, requiring additional work prior to their commercialization.

An alternative solution that may reduce the work required in the process of STB preparation for commercialization would be to substitute the place where the pre-shared key is stored. For instance, a smart-card could be used for this purpose, but additional considerations must be made in order to maintain the security of the STB bootstrap process. In particular, the pre-shared key must not be accessible to other processes, except to the STB bootstrap process.

Bibliography

[1] Libgcrypt. http://directory.fsf.org/project/libgcrypt/.

[2] libSRTP. http://srtp.sourceforge.net/srtp.html.

[3] Multicast security (MSEC). IETF Working Group.

[4] The netfilter/iptables project. http://www.netfilter.org/.

[5] IPTV functional architecture. *ITU-T Telecommunication Standardization Sector of ITU, Recommendation Y.2012*, Sep 2006.

[6] Functional requirements and architecture of the NGN (release 2). *ITU-T Telecommunication Standardization Sector of ITU, Draft Recommendation Y.NGN-FRA R2*, Sep 2008.

[7] IPTV functional architecture. *ITU-T Telecommunication Standardization Sector of ITU, Recommendation Y.1910*, Sep 2008.

[8] Telecommunications and internet converged services and protocols for advanced networking (TISPAN); IPTV architecture; dedicated subsystem for IPTV functions. *ETSI TS 182 028*, Jan 2008.

[9] 3GPP. Interworking between the public land mobile network (PLMN) supporting packet based services and packet data networks (PDN) (Release 7). TS 29.061, 3rd Generation Partnership Project (3GPP), March 2008.

[10] 3GPP. Multimedia broadcast/multicast service (MBMS); architecture and functional description (Release 9). TS 23.246, 3rd Generation Partnership Project (3GPP), Mar 2009.

[11] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible authentication protocol (EAP). *RFC 3748*, 2004.

[12] A. Adams, J. Nicholas, and W. Siadak. Protocol independent multicast - dense mode (PIM-DM): Protocol specification (revised). *RFC 3973*, 2005.

[13] S. Akl and P. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer Systems*, 1:239–248, 1983.

[14] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heam, O. Kouchnarenko, and J. Mantovani. The avispa tool for the automated validation of internet security protocols and applications. In *Proceedings of CAV*, volume 5, pages 281–285. Springer, 2005.

[15] A. Armando and L. Compagna. Satmc: A sat-based model checker for security protocols. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 730–733, 2004.

[16] A. Ballardie. Scalable multicast key distribution. *RFC 1949*, 1996.

[17] D. Basin, S. Mödersheim, and L. Viganò. Ofmc: A symbolic model checker for security protocols. *International Journal of Information Security*, 4:181–208, 2005.

[18] M. Baugher, R. Canetti, and L. Dondeti. Multicast security (MSEC) group key management architecture. *RFC 4046*, 2005.

[19] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The secure real-time transport protocol (SRTP). *RFC 3711*, 2004.

[20] Y. Boichut, P. C. Heam, O. Kouchnarenko, and F. Oehl. Improvements on the genet and klay technique to automatically verify security protocols. In *Proceedings of AVIS*, volume 4, 2004.

[21] B. Briscoe. MARKS: Zero side effect multicast key management using arbitrarily revealed key sequences. In *Proceedings of First International Workshop on Networked Communication*, pages 301–320, 1999.

[22] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet group management protocol, version 3. *RFC 3376*, 2002.

[23] R. Canetti, T. Malkin, and K. Nissim. Efficient communication-storage tradeoffs for multicast encryption. In *Proceedings of Advances in Cryptology EUROCRYPT'99*, 1999.

[24] J. Cao, L. Liao, and G. Wang. Scalable key management for secure multicast communication in the mobile environment. *Pervasive and Mobile Computing*, 2:187–203, April 2006.

[25] C. Castelluccia and G. Montenegro. Securing group management in ipv6 with cryptographically based addresses. In *Proceedings of 8th IEEE International Symposium on Computer and Communications*, 2003.

[26] Y. Challal, H. Bettahar, and A. Bouabdallah. SAKM: A scalable and adaptive key management approach for multicast communications. *ACM SIGCOMM Computer Communication Review*, 34(2):55–70, 2004.

[27] Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, J. Mantovani, S. Modersheim, and L. Vigneron. A high level protocol specification language for industrial security-sensitive protocols. *Proc. SAPS*, 4:193–205.

[28] M. Christensen, K. Kimball, and F. Solensky. Considerations for internet group management protocol (IGMP) and multicast listener discovery (MLD) snooping switches. *RFC 4541*, 2006.

[29] A. Cohen and E. Shrum. Migration to ethernet-based dsl aggregation. *DSL Forum TR-101, May*, 2006.

[30] A. Conta and S. Deering. Internet control message protocol (ICMPv6) for the Internet Protocol version 6 (IPv6) specification. *RFC 2436*, 1998.

[31] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, and D. Spence. Generic AAA architecture. *RFC 2903*, 2000.

[32] S. Deering. Host extentions for IP multicasting. *RFC 1112*, 1989.

[33] S. Deering, W. Fenner, and B. Haberman. Multicast listener discovery (MLD) for IPv6. *RFC 2710*, 1999.

[34] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment issues for the ip multicast service and architecture. *IEEE Network*, 14:78–88, 2000.

[35] D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29:198–208, 1983.

[36] L. Dondeti, S. Mukherjee, and A. Samal. Scalable secure one-to-many group communication using dual encryption. *Journal of Computer Communications*, 23:1681–1701, 2000.

[37] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. Protocol independent multicast - sparse mode (PIM-SM): Protocol specification (revised). *RFC 4601*, 2006.

[38] B. Fenner, H. He, B. Haberman, and H. Sandick. Internet group management protocol (IGMP)/multicast listener discovery (MLD)-based multicast forwarding (IGMP/MLD Proxying). *RFC 4605*, 2006.

[39] B. Fenner and D. Meyer. Multicast source discovery protocol (MSDP). *RFC 3618*, 2003.

[40] W. Fenner. Internet group management protocol, version 2. *RFC 2236*, 1997.

[41] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1. *RFC 2616*, 1999.

[42] T. Hardjono, B. Cain, and I. Monga. Intra-domain group key management protocol (IGKMP), 2000.

[43] T. Hardjono and B. Weis. The multicast group security architecture. *RFC 3740*, 2004.

[44] T. Hayashi, H. He, H. Satou, H. Ohta, and S. Vaidya. Requirements for multicast AAA coordinated between content provider(s) and network service provider(s), 2009.

[45] Y. Hinard, H. Bettahar, Y. Challal, and A. Bouabdallah. AAA based security architecture for multicast content distribution. *International Symposium on Computer Networks*, pages 85–90, 2006.

[46] H. Holbrook and B. Cain. Source-specific multicast for ip. *RFC 4607*, 2003.

[47] IANA. Special-use ipv4 addresses. *RFC 3330*, 2002.

[48] N. Ishikawa, N. Yamanouchi, and O. Takahashi. An architecture for user authentication of IP multicast and its implementation. *Internet Workshop*, pages 81–87, 1999.

[49] S. Islam and J. Atwood. A framework to add AAA functionalities in IP multicast. In *Proceedings of Advanced International Conference on Telecommunications (AICT)*, 2006.

[50] S. Islam and J. Atwood. The internet group management protocol with access control (IGMP-AC). In *Proceedings of the 31st IEEE Conference on Local Computer*, 2006.

[51] H. Jeon. Transmission of IP over Ethernet over IEEE 802.16 networks. *draft-ietf-16ng-ip-over-ethernet-over-802.16-02 (work in progress)*, July 2007.

[52] P. Judge and M. Ammar. Gothic: A group access control architecture for secure multicast and anycast. In *Proceedings of IEEE INFOCOM)*, 2002.

[53] P. Judge and M. Ammar. Security issues and solutions in multicast content distribution: a survey. *IEEE Network*, 17:30–36, 2003.

[54] O. Karppinen, O. Alanen, and T. Hamalainen. Multicast access control concept for xDSL-customers. In *Proceedings of 3rd IEEE Consumer Communications and Networking Conference*, 2006.

[55] S. Kent and R. Atkinson. Security architecture for the internet protocol. *RFC 2401*, 1998.

[56] R. Lehtonen and J. Harju. Controlled multicast framework. In *Proceedings of 27th Annual IEEE Conference on Local Computer Networks*, pages 565–571, 2002.

[57] G. Malkin. RIP version 2. *RFC 2453*, 1998.

[58] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R. Wheeler. A method for transmitting PPP over ethernet (PPPoE). *RFC 2516*, 1999.

[59] C. Metz. AAA protocols: authentication, authorization, and accounting for the Internet. *Internet Computing, IEEE*, 3:75–79, 1999.

[60] S. Mittra. Iolus: a framework for scalable secure multicast. In *Proceedings of ACM SIGCOMM'97*, pages 277–288, Cannes, France, 1997.

[61] J. Moy. Mospf: Analysis and experience. *RFC 1585*, 1994.

[62] J. Moy. Multicast extensions to ospf. *RFC 1584*, 1994.

[63] J. Moy. OSPF version 2. *RFC 2328*, 1998.

[64] P. Paul and S. Raghavan. Survey of multicast routing algorithms and protocols. In *Proceedings of 15th International Conference on Computer Communication (ICCC)*, 2002.

[65] A. Perrig, D. Song, and D. Tygar. ELK, a new protocol for efficient large-group key distribution. In *Proceedings of IEEE Symposium on Security and Privacy, S&P2001*, pages 247–262, 2001.

[66] R. Pietro, L. Mancini, and S. Jajodia. Efficient and secure keys management for wireless mobile communications. In *Proceedings of the second ACM International Workshop on Principles of Mobile Computing*, pages 66–73, 2002.

[67] A. Pinto and M. Ricardo. Multicast deflector. *Telecommunication Systems*, 37:145–156, April 2008.

[68] A. Pinto and M. Ricardo. SMIz - secure multicast IPTV with efficient support for video channel zapping. In *Proceedings of Networking and Electronic Commerce Research Conference 2008 (NAEC 2008)*, 2008.

[69] S. Rafaeli and D. Hutchison. A survey of key management for secure group communication. *ACM Computing Surveys (CSUR)*, 35(3):309–329, 2003.

[70] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session initiation protocol. *RFC 3261*, 2002.

[71] P. Santos, A. Pinto, M. Ricardo, F. Fontes, and T. Almeida. Admission control in IP multicast over heterogeneous access networks. In *2nd International Conference and Exhibition on NEXT GENERATION MOBILE APPLICATIONS, SERVICES and TECHNOLOGIES (NGMAST'08)*, Sep 2008.

[72] H. Satou, H. Ohta, C. Jacquenet, T. Hayashi, and H. He. AAA and admission control framework for multicasting, 2009.

[73] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. *RFC 3550*, 2003.

[74] H. Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (RTSP). *RFC 2326*, 1998.

[75] S. Setia, S. Koussih, S. Jajodia, and E. Harder. Kronos: a scalable group rekeying approach for secure multicast. In *Proceedings of 2000 IEEE Symposium on Security and Privacy*, pages 215–228, 2000.

[76] I. Std 802.16-2004. IEEE standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems, 2004.

[77] M. Steiner, G. Tsudik, and M. Waidner. Diffie-hellman key distribution extended to group communication. In *Proceedings of the 3rd ACM conference on Computer and communications security*, pages 31–37, New Delhi, India, 1996.

[78] M. Steiner, G. Tsudik, and M. Waidner. CLIQUES: A new approach to group key agreement. In *Proceedings of IEEE ICDCS'98*, 1998.

[79] M. Turuani. *The CL-Atse Protocol Analyser*, pages 277–286. 2006.

[80] R. Vida and L. Costa. Multicast listener discovery version 2 (MLDv2) for IPv6. *RFC 3810*, 2004.

[81] D. Waitzman, C. Partridge, and S. Deering. Distance vector multicast routing protocol. *RFC 1075*, 1988.

[82] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner. The versakey framework: versatile group key management. *IEEE Journal on Selected Areas in Communications*, 17:1614–1631, 1999.

[83] L. Wei and D. Estrin. A comparison of multicast trees and algorithms. In *Proceedings of IEEE Inforcomm*, 1994.

[84] WiMAX Forum NWG. WiMAX forum network architecture stage 2-3. release 1, version 1.2, 2008.

[85] C. Wong, M. Gouda, and S. Lam. Secure group communications using key graphs. *IEEE/ACM Transactions on Networking*, 8:16–30, 2000.

[86] D. Wu, Y. Hou, W. Zhu, Y. Zhang, and J. Peha. Streaming video over the Internet: Approaches and directions. *IEEE Transactions On Circuits and Systems for Video Technology*, 11, 2001.

Appendices

Appendix A

HLSPL specification of STB bootstrap

```
role stb(A, B : agent,
           H : hash_func,
           Kab : symmetric_key,
           SND, RCV : channel (dy))
played_by A
def=
   local Na,Nb,CHid : text,
         State : nat,
         Sek : symmetric_key,
     Ts1,Ts2,Ts3,Ts4 : text,
 Finished : text
   init  State := 0
   transition

   0.  State = 0
       /\ RCV(start)
       =|>
       State' := 2
       /\ Ts1' := new()
       /\ Na' := new()
       /\ SND(A.H(Na').{A.Ts1.Na'}_Kab)
```

```
   2.  State = 2
       /\ RCV(A.B.H(Nb').{A.B.Ts2'.Nb'}_Kab)
       =|>
       State' := 4
       /\ Ts3' := new()
       /\ SND(A.B.H(Ts3'.Na.Nb').{A.B.Ts3'.H(Ts3'.Na.Nb')}_Kab)

   4.  State = 4
       /\ RCV(A.B.H(Ts3.Na.Nb).{A.B.Ts4'.Sek'}_Kab)
       =|>
       State' := 6
       /\ request(A,B,auth_sek,Sek')
end role

role group(A, B : agent,
         H : hash_func,
         Kab : symmetric_key,
         SND, RCV : channel (dy))
played_by B
def=
   local Na,Nb : text,
         State : nat,
         Sek : symmetric_key,
     Ts1,Ts2,Ts3,Ts4 : text
   init  State := 1
   transition

   1.  State = 1
       /\ RCV(A.H(Na').{A.Ts1'.Na'}_Kab)
       =|>
       State' := 3
       /\ Nb' := new()
       /\ Ts2' := new()
       /\ SND(A.B.H(Nb').{A.B.Ts2'.Nb'}_Kab)

   3.  State = 3
```

```
       /\ RCV(A.B.H(Ts3'.Na.Nb).{A.B.Ts3'.H(Ts3'.Na.Nb)}_Kab)
       =|>
       State' := 5
       /\ Ts4' := new()
       /\ Sek' := new()
       /\ SND(A.B.H(Ts3'.Na.Nb).{A.B.Ts4'.Sek'}_Kab)
       /\ witness(B,A,auth_sek,Sek')
       /\ secret(Sek',sec_sek,{A,B})
end role

role session(A,B: agent,
             Kab  : symmetric_key,
             H : hash_func)
def=
   local  SA, SB, RA, RB: channel (dy)
   composition
               stb(A,B,H,Kab,SA,RA)
          /\   group(A,B,H,Kab,SB,RB)
end role

role environment()
def=
   const sec_sek, auth_sek : protocol_id,
         h : hash_func,
         a, b    : agent,
         kab, kib : symmetric_key

   intruder_knowledge = { a, b, kib }
   composition
        session(a,b,kab,h)
     /\  session(a,i,kib,h)
     /\  session(i,b,kib,h)
end role

goal
  secrecy_of sec_sek
```

```
  authentication_on auth_sek
end goal

environment()
```