Computer Communications 34 (2011) 1708–1721

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

On performance of group key distribution techniques when applied to IPTV services

António Pinto [a,b,*], Manuel Ricardo [c]

[a] INESC Porto, Portugal
[b] Escola Superior de Tecnologia e Gestão de Felgueiras, Politécnico do Porto, Portugal
[c] INESC Porto, Faculdade de Engenharia, Universidade do Porto, Portugal

ARTICLE INFO

Article history:
Received 19 April 2010
Received in revised form 5 October 2010
Accepted 19 March 2011
Available online 3 April 2011

Keywords:
IPTV
User groups
Admission control
Multicast
Security

ABSTRACT

IPTV services consist of multiple video channels grouped in bundles, such as sports, movies or generic bundles; users typically subscribe multiple bundles, including the generic bundle. Secure IP multicast can be used to implement IPTV services, but it still has problems to be addressed. Current solutions require high computational power in video channel zapping situations, lack support for groups sourced at the users, and present a weak support for admission control in IP multicast for both sources and receivers in dynamically configured environments.

This work proposes a new, secure and efficient 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 signaling overheads; (e) does not introduce perceivable delays, particularly in video channel zapping situations. Moreover, this solution can be easily integrated in the IPTV architectures being developed by ETSI and ITU-T.

1. Introduction

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, the RTP which enables the transmission of video, voice and multimedia contents in IP packets, along with other protocols for controlling the video streaming. More recently [1,2], 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. Despite of the scalability obtained by using multicast techniques, network operators have been reluctant to use them [14] due to their lack of native control over groups, making it difficult for network operators and service providers to perform access control, traffic accounting, and network management. Thus, the use of multicast in current IPTV services exists but it is limited, namely by not allowing user generated multicast traffic and by only allowing downlink multicast traffic on separated (virtual) circuits which are specifically used by the IPTV service.

Secure IP multicast [3–5] may be used to support the secure transmission of IP packets to groups of receivers in IPTV services but neglects access control and network management. Key distribution solutions for secure group communications usually apply key refreshing techniques upon a group change (member join or leave) in order to impose both perfect forward and backward secrecy.

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 users will be compelled to generate and distribute their own real-time videos to groups of other users, directly from their premises. This scenario requires network operators to protect also user generated videos in what concerns confidentiality and access control.

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 signaling overheads; (e) does not introduce perceivable delays, particularly in video channel zapping situations.

The reference scenario adopted for this work is shown in Fig. 1. It describes an IPTV service where video channels are distributed as IP packets and transmitted to a multicast address – one multi-

* Corresponding author. Tel.: +351 255314002.
E-mail addresses: apinto@inescporto.pt, apinto@inescporto.pt (A. Pinto), mricardo@inescporto.pt (M. Ricardo).

0140-3664/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2011.03.011

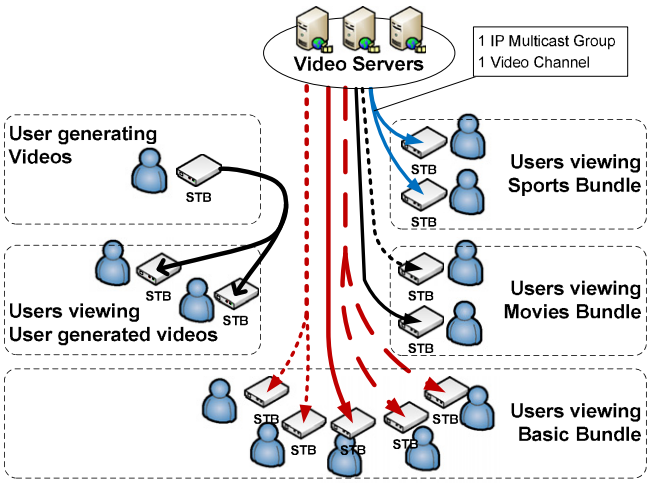


Fig. 1. Reference scenario.

cast 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 by 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 extend our previous secure multicast technique [3] by adding the support for multicast admission control and user generated groups. The video channels are generated by Video Servers (VS) to groups of Set-Top Boxes (STB). A STB may also generate video contents and uses heterogeneous access networks to access the IPTV service, including xDSL, WiMAX or UMTS.

This paper is organized as follows. Section 2 provides an overview of the different existing proposals for multicast admission control and summarizes their key differences. Section 3 provides an overview of secure multicast, classifying existing solutions according to four types of key distribution. Section 4 describes the key components and interfaces of the proposed solution and assesses the deployment of the proposed solution over heterogeneous access networks. Section 6 presents the obtained results. Section 7 draws the conclusions of this work.

2. Multicast admission control

Work related to IP multicast AAA is being carried out within the IETF MBONE Workgroup. In [22] the requirements for multicast AAA were specified, and in [23] a general multicast AAA framework is being designed to satisfy 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 in 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.

2.1. Adopted notation

Table 1 presents the adopted notation. 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 an 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 communicating node A. $X \cdot Y$ 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.

Table 1
Adopted notation.

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
$X \cdot Y$	Field X concatenated with field Y
APriK	Private key of entity A
APubK	Public key of entity A
Ts	Time stamp

2.2. Additional control layer approach

In [21], the authors propose a new communication protocol, the MCOP, used to exchange messages between the edge router and the 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 2 details the message exchange for a receiver access control operation. A host wiling 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.

In [24] 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 retrieve information regarding the edge device associated with the request.

2.3. Protocol modification approach

SMKD [15] consists in 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 GKDC that holds the group ACL and distributes cryptographic keys to authorized routers and hosts. Table 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 time stamp 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 the group related cryptographic keys. At this

Table 2
MCOP protocol for receiver access control.

Sequence	Entities	Messages
1	*Host* → *Router*	IGMP/MLD Join
2	*Router* → *MCA*	Validate:Group_Address.Host_Address
3	*MCA* → *Router*	Result:Group_Address.Host_Address

Table 3
SMKD protocol for receiver access control.

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

Table 4
Gothic protocol for receiver access control.

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

Table 5
G-CBA protocol for receiver access control.

Sequence	Entities	Messages
1	*GC* → *Host*	$\{PriKGroup\}_K \cdot \{PubK_{G\_GCBA\_Address}\}_{GCPubK}$
2	*Host* → *Router*	$MLD(\{G\_CBA\_Address \cdot GroupPubK\}_{GroupPriK})$

moment, the router will store the group ACL and assume the GKDC functionality for future downstream group join requests.

Gothic [16] proposed the use of PKI in conjunction with IGMP and MLD message modification to include X.509 certificates. It introduces a new entity called ACS that is responsible for the authorization of join requests. Table 4 details the message exchange for a receiver access control operation.

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 field 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 acknowledgement message.

G-CBA [17] 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 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 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[1] transmits the key pair to group members by sending a message that comprises the group's private key encrypted with *K*, the group's public key and the 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.

In [18] the authors propose a framework to add AAA capabilities to standard IP multicast by modifying IGMPv3 and, in [19], they introduced IGMP-AC to support multicast access control. IGMP-AC consists on using EAP combined with IPSec and in the modification of IGMPv3 messages to impose multicast group access control for both senders and receivers. The modification of IGMPv3 messages consists on adding user authentication data. In both [18,19], the modification of the IGMPv3 messages is not specified.

The MCDA2 detailed in [20] makes use of a previous IGMPv2 modification proposal [27]. These solutions follow the general AAA architecture as defined in [28] but they are applied to multicast sessions. The NAS, upon receiving an 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 the intended IP multicast group.

2.4. Summary on multicast admission control

Table 6 compares the multicast admission control techniques identified in this section. The comparison criteria are the following: demand for group management protocol modification (1st column); usage of cryptographic techniques (2nd column); PKI demand (3rd column); sender access control (4th column); receiver access control (5th column); access control to nodes of the multicast distribution tree (6th column); usage of standard protocol for AAA (7th column); IPv4 and IPv6 support (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 address only the IPv4 protocol. The MCOP solution is also the unique solution that does not require group management protocol modification nor the use of cryptographic techniques. Nevertheless, MCOP addresses only IPv4 networks and uses a non standard protocol for authorization.

3. Secure multicast

Secure multicast is a group transmission technique that enforces confidentiality. It uses cryptographic techniques to encrypt data and perform access control. A secure multicast architecture needs to consider the size of the groups, group memberships, and security contexts such as encryption keys.

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 a 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. Several cryptographic keys are used in secure group communication architectures. The common ones are Key Encryption Keys (KEK) and Data Encryption Keys (DEK), which are managed by an entity called Group Controller (GC). KEK is a key assigned to a member, known only by that member and the GC, and it is used to secure communications between the member and the GC. DEK is used to encrypt the group data and must be known by all the members of the group. In a re-key operation, for instance, the DEK may be transmitted by the GC to valid members in a message consisting of $[\{DEK\}_{KEK_1}, \{DEK\}_{KEK_2}, \ldots, \{DEK\}_{KEK_n}]$.

Group confidentiality is obtained by changing the decryption key, and by transmitting it securely to the authorized receivers. Decryption keys can be transmitted periodically, upon a group

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

Table 6
Comparison of multicast admission control techniques.

Name	IGMP/MLD Modification	Uses Crypto. Techniques	Uses PKI	Sender Control	Receiver Control	Tree node Control	Standard 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

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). A group change implies key renewal to impose forward and backward secrecy. Forward secrecy imposes that a member departing from a group should stop receiving cryptographic material. Backward secrecy 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.

We adopted a classification of current solutions for key distribution that combines the classifications presented in [25] and in [7]. Our classification comprises 4 types of key distribution: centralized, decentralized, distributed and hierarchical. The distributed type assumes that all members participate in key generation, key distribution, and perform access control; the group controller role is not usually present because group keys are generated with contributions from all members. Group Diffie–Hellman Key Exchange [29] is an example of the distributed type. The hierarchical type assumes that users are classified according to their priorities and access levels and impose cryptographic access control to classes of users; this type was firstly addressed in [30] by adopting a hierarchical key assignment. In hierarchical key assignment, an user belonging to a certain class can derive the cryptographic keys of lower class users. 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 [11] is an example of a decentralized key distribution.

When considering the applicability of these key distribution types to IPTV services, we identified [3] the centralized and decentralized as the types more adequate to IPTV services.

4. Proposed solution

The solution proposed in this paper, named Secure Multicast IPTV with efficient support for video channel *zapping* (SMIz), has the architecture shown in Fig. 2 and it consists of a Group Controller (GC), a Video Server (VS), Set-Top Boxes (STB), a Multicast Controller (MC) [13] and an Authentication, Authorization and Accounting (AAA) server [32]. The GC is in charge of key generation and distribution, STB authentication and authorization, and of updating 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 announce).

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 having a different VEK. KEKs, one per bundle (group of IPTV channels with some affinity), are used to secure the transmission of VEKs.

At bootstrap, the STB requests its 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. 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

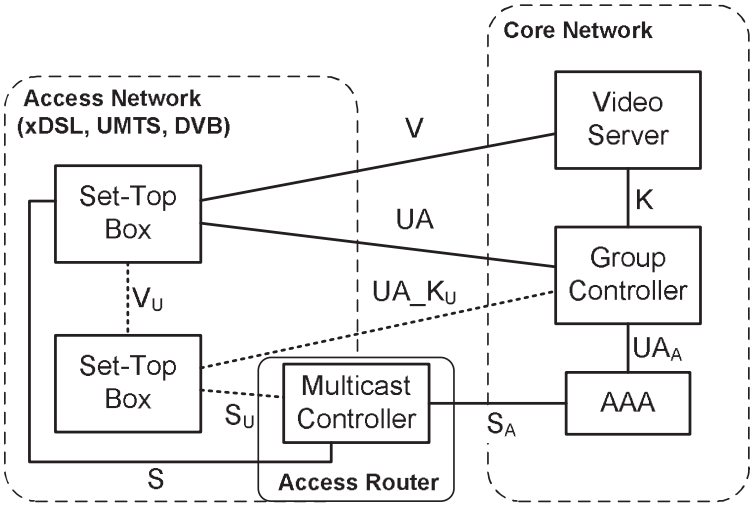


Fig. 2. Proposed solution.

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 announces (at a rate of 10 announces per second), transmitted from the VS to the same group address of the video channel. VEK announces are secured by KEK ($\{VEK\}_{KEK}$) and only one VEK announce is required for all group members, since all of them share the same KEK. This procedure leads to savings in signaling, specially because VEKs are the keys which are re-keyed more frequently.

User generated multicast streams are supported by the proposed solution by allowing source users to obtain, from the GC, VEKs for their content 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).

Complementary to confidentially assurance, multicast admission control enables an additional 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 through 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 multicast packet to be normally processed.

The AAA, based on the multicast profiles, rejects or authorizes multicast accesses. The AAA server is assumed to store the IP addresses 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.

Several interfaces are identified in Fig. 2. The *V* interface is used to transmit the video streams from the VS to the STB; these streams are transmitted as IP packets having as destination a multicast address. The V_U interface is similar to *V* interface, but used to transmit the video streams generated by the domestic users as IP multicast. 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 that STB requests for KEKs from the GC when generating a video multicast stream, 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 an 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.

4.1. UA interface

The *UA* interface is used by the GC to authenticate and authorize STB access to video channels. Messages required to bootstrap the STB, request the KEK, and refresh the KEK are exchanged through this interface. Table 7 summarizes the messages exchanged in these 3 phases.

The Bootstrap phase is executed during the STB bootstrap and enables mutual authentication of STB and GC by means of the symmetric pre-shared key (K_{ab}). The initiator is the STB and it starts by sending a message composed of the initiator's identification (A), the result of an hash function of a fresh nonce ($H(N_a)$), and a set of 3 fields ($A \cdot T_{s1} \cdot N_a$) encrypted with the 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 stamp against previous values. The nonce must not be repeated, and the time stamp must be higher than the last time stamp (typically, the 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 stamp (T_{s2}). The STB will verify the nonce and time stamp. Upon successful verification, the STB will reply with a new message composed by the identification of both, the result of an hash function of both nonces, a new time stamp (T_{s3}), and a new encrypted set of fields. This set of fields is composed by the identifications A and B, the time stamp T_{s3}, and an 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 stamp 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 stamps (T_{s1} through T_{s4}) and nonces (N_a and N_b) are used to prevent replay attacks. Using only time stamp, 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 stamps, 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

Table 7
Messages exchanged through the *UA* interface.

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

is required. The KEK refresh messages are sent by the GC to individual STB as unicast messages. These messages contain the KEKs, which are sent periodically 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.

4.2. 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 8.

4.3. V interface

The *V* interface is used to transmit the video channels and the VEK announces from the VS to the STB. Both are transmitted to IP multicast groups. It comprises 2 phases: Streaming, and VEK Refresh. Table 9 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 refresh, also referred to as VEK announce. The VEKs are sent periodically 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 announce 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. 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 as reached its maximum value (roll-over counter).

The support of fast switching between video channels is achieved by transmitting the periodic and frequent VEK announces, in multicast, to the same group address of the video channel. VEK announces are secured by $\{VEK\}_{KEK}$ and only one VEK announce per video channel is needed for all members, since all of them share the same KEK. This procedure of frequently transmitting the VEKs in multicast leads to significant savings in signaling, because a single message containing the new VEK can be received by multiple users. It also facilitates users zapping between channels of the same bundle because, in this case, STBs are already in possession of the KEK.

4.4. 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 which, in turn, is secured by KEK. KEKs are generated by the GC. It consists of 2 phases: KEK Request, and KEK Setup.

Table 10 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, it enables a VS to request a KEK Setup of a video channel bundle that, for some reason, was not successfully concluded previously. The KEK Setup phase has two messages. The first message is, in structure, similar to the VEK refresh message of Section 4.3; 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 an 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 acknowledge message shown in Table 10.

4.5. S_A Interface

The S_A interface consists of RADIUS 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 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 user. Table 11 summarizes the multicast session identifiers adopted, obtained from IP multicast packets. A member's session can be identified in one of two ways,

Table 8
Messages exchanges through the *S* and S_A interfaces.

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

Table 9
Messages exchanged through the *V* interface.

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

Table 10
Messages exchanged through the *K* interface.

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

Table 11
Multicast session IDs source.

Received packet	Multicast session IDs
IGMPv1/v2	SA, GDA
IGMPv3	SA, GDA, GSA
UDP multicast	SA, DA

Table 12
Messages exchanges through the UA_K_U interface.

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

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.

4.6. UA_K_U Interface

The UA_K_U interface is used by an STB, which is 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 12 summarizes the messages exchanged.

STB starts by sending the first message from the UGV Setup phase, shown in Table 12. 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 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.

4.7. 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 its own video channels. Table 13 summarizes the messages exchanged.

Table 13
Messages exchanges through the UA_A interface.

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

The first phase (Profile Setup) shown in Table 13 enables the initial multicast profile definition for an STB. It consists of 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 2 and it consists of the list of channels subscribed by the STB and the UGV field (line 12). The UGV field indicates if user generated video streams are allowed (value 1) or not (value 0).

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 a XML formatted data with the relevant information. An example of such information is shown in Listing 3 and comprises the source identification, the multicast address and port number to be used in the video channel streaming, and the 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.

4.8. V_U and S_U interfaces

The V_U interface, similarly to *V* interface, is used to transmit the video channels IP multicast streams 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 an user generated video channel stream. The messages exchanged through this interface, shown in Table 14, consist of the first SRTP packets of the stream that, when received by the MC, will trigger the authorization validation by the AAA. Upon a successful authorization, the MC will start to forward all the packets of the multicast stream. Upon a unsuccessful authorization, the MC will discard all the other packets of the multicast stream.

4.9. Heterogeneous access networks support

The solution proposed for IP multicast admission control operates at the network level and makes it adequate to any access network supporting IP multicast. Nevertheless, aspects such 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 then to define where and how to perform access control, for both traditional multicast groups and groups sourced at the user pre-

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

Listing 1. Example of XML formatted data of user generated videos setup.

```
<?xml version="1.0"?>
<MCASTPROFILE>
    <USER>user1@exampledomain.com</USER>
    <LIST>
        <CHANNEL> <ID>1</ID>
            <DESCRIPTION>Channel 1</DESCRIPTION>
            <ADDRESS>229.0.0.1</ADDRESS>
        </CHANNEL>
        <CHANNEL> <ID>2</ID>
            <DESCRIPTION>Channel 2</DESCRIPTION>
            <ADDRESS>229.0.0.2</ADDRESS>
        </CHANNEL>
    </LIST>
    <UGV>1</UGV>
</MCASTPROFILE>
```

Listing 2. Example of XML formatted multicast profile (XMLPROFILE).

```
<?xml version="1.0"?>
<UGV>
    <SOURCE>user1@exampledomain.com</SOURCE>
    <MCAST ADDRESS>232.1.0.1</MCAST ADDRESS>
    <MCAST PORT>1234</MCAST PORT>
    <RECEIVERS>
        <USERNAME>user2@exampledomain.com</USERNAME>
        <USERNAME>user3@exampledomain.com</USERNAME>
        <USERNAME>user4@exampledomain.com</USERNAME>
    </RECEIVERS>
</UGV>
```

Listing 3. Example of XML formatted data sent by GC to AAA (XMLUGV).

Table 14
Messages exchanges through the S_U interface.

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

mises, 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 Fig. 3.

Table 15 summarizes the support each access technology has for multicast. The elements responsible for the 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 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 multicast packets forwarding. The BNG is also the NAS where users authenticate themselves during network attachment by 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 to 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, the 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, there is no support for user generated multicast streams.

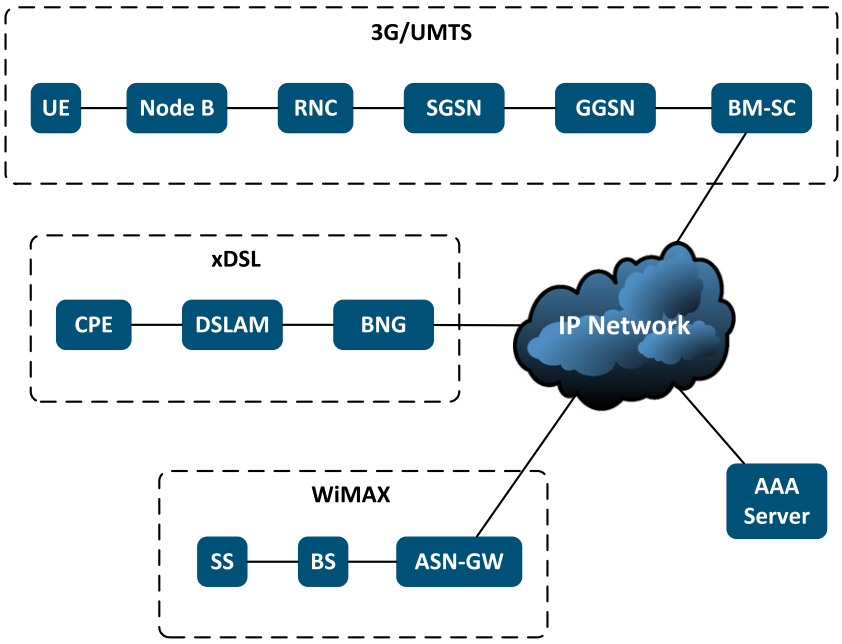


Fig. 3. Adopted network architecture.

Table 15
Multicast support comparison.

	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

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.

5. NGN deployment scenarios

Current IPTV services rely on IETF standardized protocols such as RTP for transport, IGMP or SIP for signaling, 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 lead to the creation of interest groups focused on the development of solutions which 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 [1] 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 [34] 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, SIP, and DIAMEETER, it does not specify the majority of the interfaces between functional blocks; these are still marked for further studies.

We claim that the solution proposed can be used in 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. Fig. 4 shows the architecture of our proposed solution, adopting a graphical style similar 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 comprises Content Preparation Functions, which is responsible for interacting with End-User Functions be 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, interacts 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.

5.1. Non-NGN IPTV functional architecture

The Non-NGN IPTV Functional architecture uses legacy technologies for the delivery of IPTV services. The set of functionalities common with the proposed solution comprises the SCP Functions,

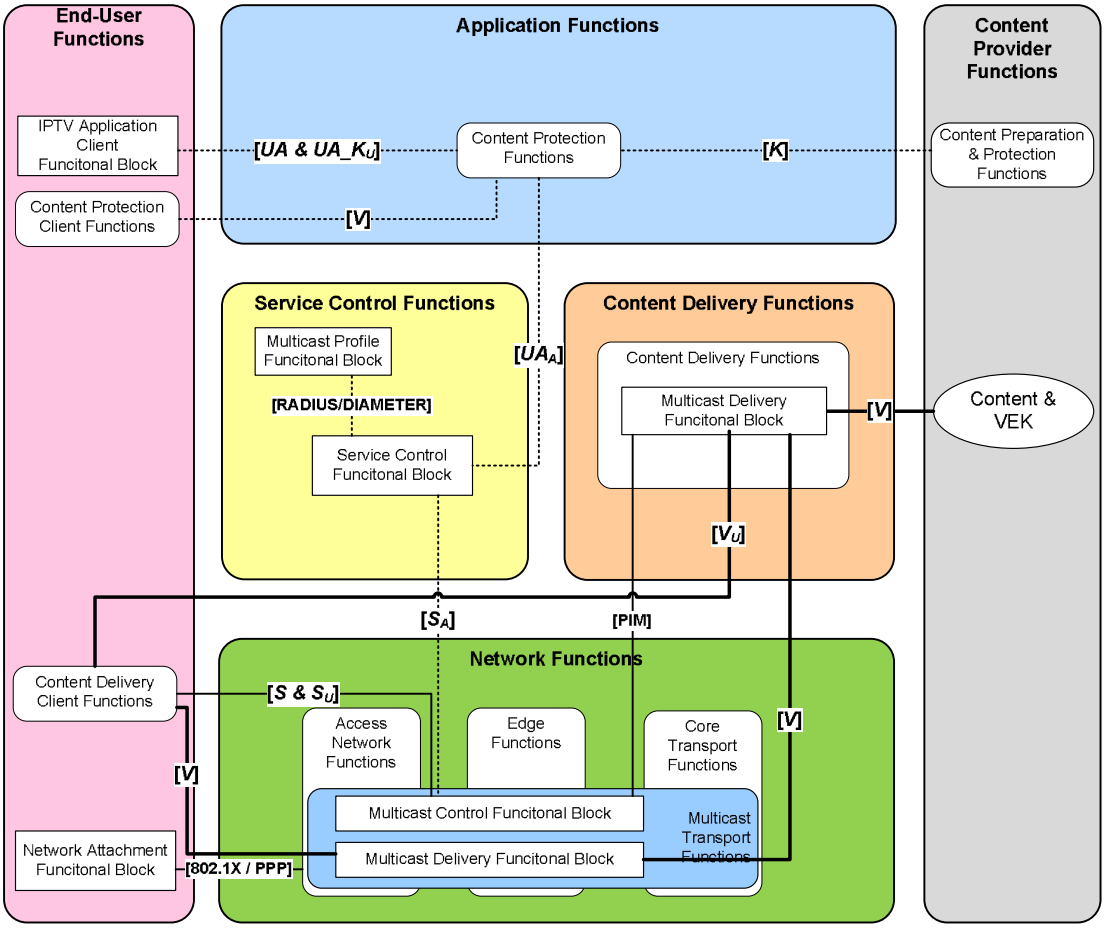


Fig. 4. Architecture of the proposed solution.

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;
- 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's 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.

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.

5.2. 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 16. 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

Table 16
Relationship between the functions of NGN-based IPTV and NGN architectures.

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

Service Stratum Functions. In particular, the NGN Service Control Functions may include functionalities other than those of the ITU-T IPTV Service Control Functions [33]. ITU-T IPTV Applications Functions correspond to both Application Support Functions and Service Support Functions, also included in NGN Service Stratum Functions. The Content Delivery Functions are not specified in NGN functional requirements and architecture [33,34]; moreover, NGN Content Delivery Functions and NGN Applications Functions may be deployed by a third party service provider.

The set of functionalities common with the proposed solution 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;

NACF and 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.

5.3. 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.The set of functionalities common with the proposed solution 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);
- 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 signaling 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 signaling would be used to trigger multicast admission control and video channel transmission.

5.4. Summary

Standard protocols such as RTP for transport, IGMP or SIP for signaling, 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 lead 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 [1] for that purpose.

Our proposed solution was compared to the three IPTV architectural approaches identified in [1]. 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, demands session signaling. The NACF and RACF are new functional blocks and they handle session signaling in order to impose network access control and resource reservation, respectively. The proposed solutions does not require the use of session related signaling besides IGMP/MLD signaling; 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 signaling protocol. The proposed solution does not adopt SIP, or any other session management protocol for session signaling 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 signaling to trigger multicast admission control and video channel transmission.

6. Results

In [3] we showed that our group key management solution requires less bandwidth than current solutions present in

Table 17
Average execution times of cryptographic functions.

	Execution cycles	Average execution time (μs)
AES 128	1000	5.9
MD5	1000	69.9
AES 128 Key Generation	1000	5.4

Table 18
Encryption cost of join/leave operations comparison.

	Single join	Single leave
LKH	$((d+2)(h-1)/2)E$	$((d+2)(h-1)/2)E$
OFT	$G+(h-1)H+(h-1)E$	$(h-1)G+(2h-3)E$
ELK	$(2n-1)H+E$	$(2d)H+(2d)E$
OFCT	–	$G+(2h-1)H$
LKH++	$(h+2)H$	$2hH$
SMIz	$H+E$	–

literature. Typical IPTV services use one DEK per video channel and refresh them frequently, leading to a high usage of network resources. Our solution, by using two types of keys (one for video channels and another for video channel bundles), is capable of efficiently enforcing security in IPTV services without requiring frequent re-key operations due to group changes.

In this paper we have extended our solution presented in [3] to support user generated video channels and to impose IP multicast admission control. A user behaving as a video source can obtain VEKs for his video channels and distribute them to restricted groups of receivers, triggering the system to automatically adapt to this new channel.

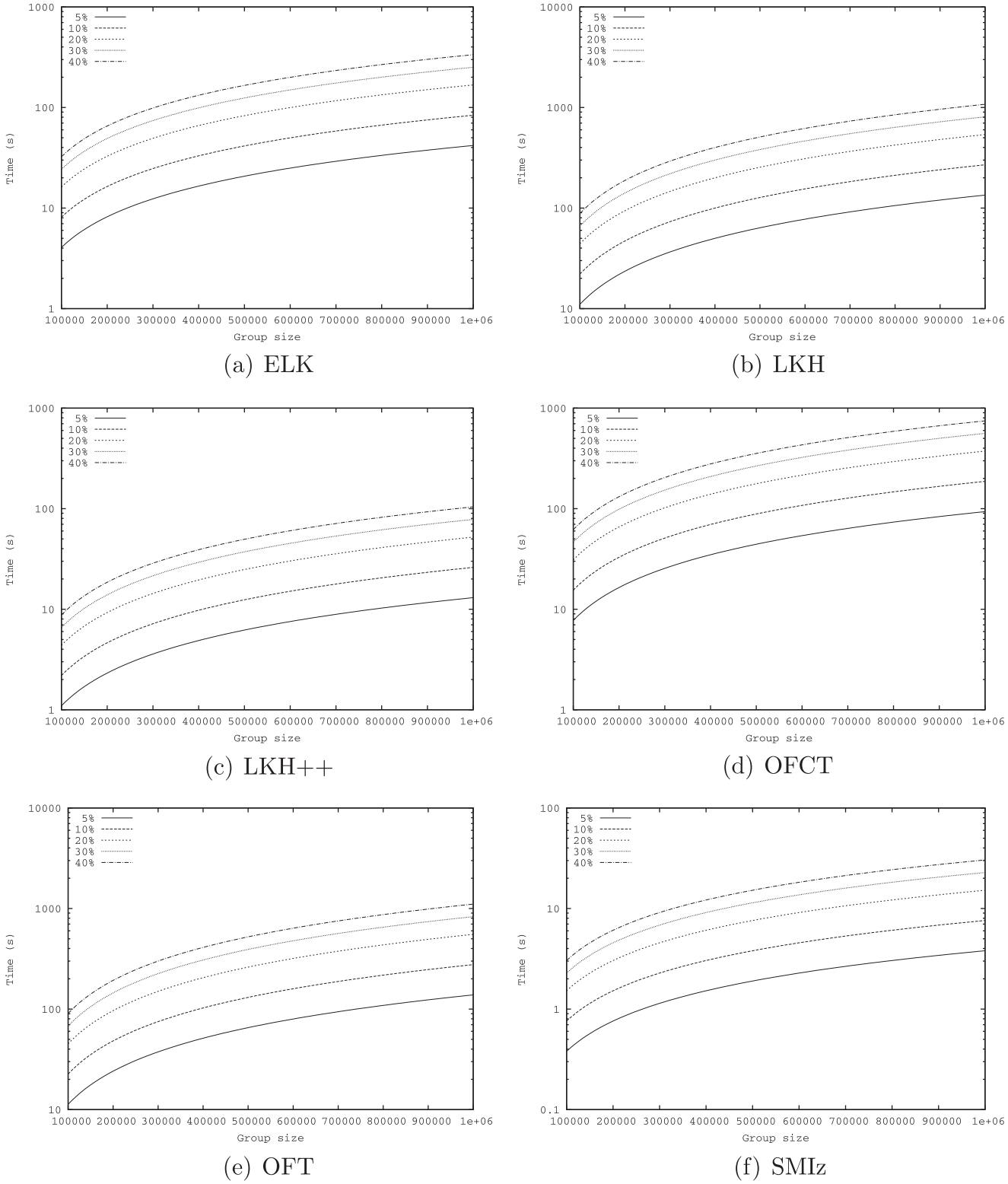
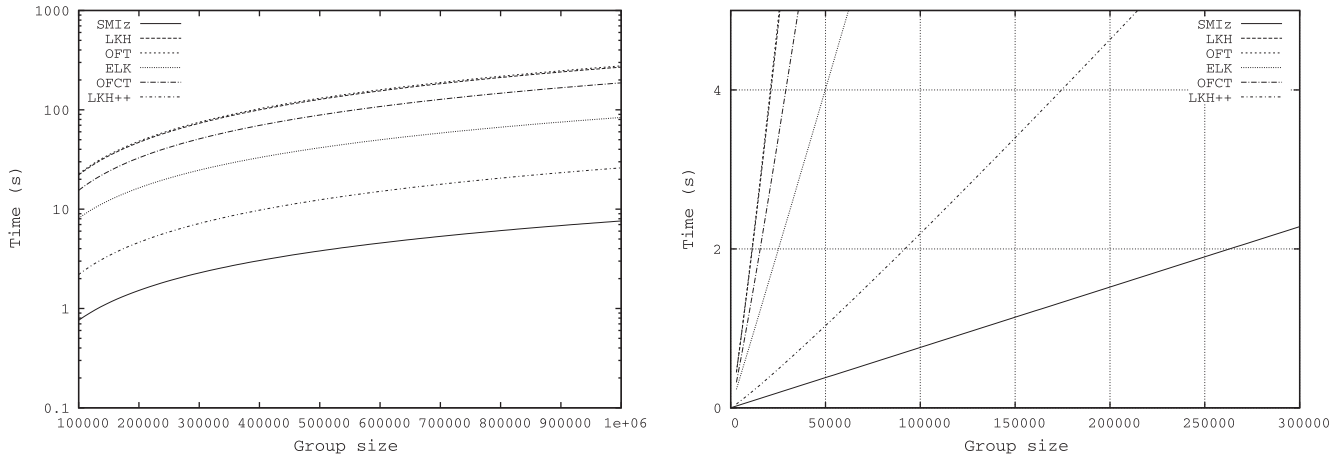


Fig. 5. Time used by GC to perform cryptographic functions by group size and percentage of *zapping* members.



(a) Logarithmic scale (b) 2 seconds period

Fig. 6. Comparison of time used by GC to perform cryptographic functions by group size.

Another important issue associated with secure group communications is the computational power required at the GC what lead us to analyse the time required by the GC to successfully execute the cryptographic functions in scenarios where users switch rapidly between the multiple available video channels.

The two cryptographic functions most frequently used in current solutions are symmetric encryption functions and one-way hash functions. In order to estimate the time required for these operations some experiments were executed. Table 17 summarizes the results obtained on a system with an Intel T5500 1.67 Ghz processor and 1.5 GB of RAM memory. An AES encryption operation for a key size of 128 bits used approximately 69.9 μs, an AES key generation operation for a key size of 128 bits used approximately 5.5 μs, and an MD5 hash function used approximately 5.9 μs of computational power, in average.

Table 18 compares the computational cost of both join and leave operations, where d and h denote the degree and height of the tree, respectively, E denotes the cost of a symmetric encryption operation, H denotes the cost of an hash operation, and G denotes the cost of a key generation operation. In the particular case of OFT [9] we have omitted the asymmetric encryption operations because none of the remaining solutions uses asymmetric encryption.

Fig. 5 graphically describes results for group sizes up to one million of members and for percentages of zapping members of 5%, 10%, 20%, 30% and 40%. For instance, for a group size of one million members of which 10% are switching video channels, our solution would require 7.6 s, LKH++ [26] would require 26.1 s, ELK [10] would require 83.8 s, OFCT [31] would require 186.9 s, LKH [8] would require 269.1 s, and OFT would require 276.7 s, approximately.

Fig. 6(a) compares the several solutions and graphically describes, for groups up to one million members, the time used by the GC on performing the cryptographic functions required when 10% of the members are switching video channels. A logarithmic scale was used. Fig. 6(b) is analogous to Fig. 6(a) but limited to a 2 s period, which the maximum period assumed as comfortable for a video channel switch. In particular and for this maximum period, our solution is capable of processing more than 250.000 video channel switch operations, while the second best solution (LKH++) processes less than 100.000 operations.

7. Conclusions

The proposed IPTV solution evolves our previous group key management solution [3]. This new solution is able to efficiently enforce individual access control to groups of real-time IPTV video channels and perform IP multicast admission control for both multicast senders and receivers, while supporting user generated videos. Moreover, this solution demands low central computational power and does not introduce perceivable delays in video channel zapping situations.

Our group key management 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 low computational power in refresh operations. While not enabling perfect forward and backward secrecy, it enables a significant reduction of central computational power in situations where users switch rapidly between channels.

Our admission control technique for both IP multicast senders and receivers enables the management of multicast sessions that span over heterogeneous access networks. It generates and distributes multicast profiles that specify if users are allowed to generate their videos and contains lists of authorized video channels for each user. These multicast profiles are stored in an AAA server that responds to queries from the MC. Upon successful verification, the MC authorizes 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 users generating content over connections with renewable IP configurations assigned by DHCP.

References

[1] IPTV functional architecture ITU-T Telecommunication Standardization Sector of ITU, Recommendation Y.1910 2008.
[2] Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); IPTV Architecture; Dedicated subsystem for IPTV functions ETSI TS 182 028, 2008.
[3] A. Pinto, M. Ricardo, Secure multicast in IPTV services, Computer Networks 54 (10) (2010) 1531–1542, doi:10.1016/j.comnet.2009.12.007.
[4] A. Pinto, M. Ricardo, Multicast deflector – A secure video distribution system, Telecommunication Systems 37 (4) (2008) 145–156.
[5] A. Pinto, M. Ricardo, SMIz - Secure Multicast IPTV with efficient support for video channel zapping, in: Proceedings of Networking and Electronic Commerce Research Conference(NAEC 2008), 2008.
[7] J. Cao, L. Liao, G. Wang, Scalable key management for secure multicast communication in the mobile environment, Pervasive and Mobile Computing 2 (2006) 187–203.
[8] C. Wong, M. Gouda, S. Lam, Secure group communications using key graphs, IEEE/ACM Transactions on Networking 8 (2000) 16–30.
[9] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, B. Plattner, The VersaKey framework: versatile group key management, IEEE Journal on Selected Areas in Communications 17 (1999) 1614–1631.
[10] A. Perrig, D. Song, D. Tygar, ELK, a new protocol for efficient large-group key distribution, in: Proceedings of IEEE Symposium on Security and Privacy, 2001.

[11] S. Mittra, Iolus: a framework for scalable secure multicast,. in: Proceedings of ACM SIGCOMM, 1997.
[13] P. Santos, A. Pinto, M. Ricardo, F. Fontes, T. Almeida, Admission control in IP Multicast over Heterogeneous access networks, in: Proceedings of International Conference and Exhibition on Next Generation Mobile Applications, Services and Technologies (NGMAST), 2008.
[14] C. Diot, B.N. Levine, B. Lyles, H. Kassem, D. Balensiefen, Deployment issues for the IP multicast service and architecture, IEEE Network 14 (2000) 78–88.
[15] A. Ballardie, Scalable multicast key distribution, RFC 199 (1996).
[16] P. Judge, M. Ammar, Gothic: a group access control architecture for secure multicast and anycast, in: Proceedings of IEEE Conference on Computer Communications, 2002.
[17] C. Castelluccia, G. Montenegro, Securing group management in IPv6 with cryptographically based addresses, in: Proceedings of IEEE International Symposium on Computer and Communications, 2003.
[18] S. Islam, J. Atwood, A framework to add AAA functionalities in IP multicast, in: Proceedings of Advanced International Conference on Telecommunications (AICT), 2006.
[19] S. Islam, J. Atwood, The Internet Group Management Protocol with Access Control (IGMP-AC), in: Proceedings of IEEE Conference on Local Computer Networks, 2006.
[20] Y. Hinard, H. Bettahar, Y. Challal, A. Bouabdallah, AAA based security architecture for multicast content distribution, in: Proceedings of International Symposium on Computer Networks, 2006.
[21] R. Lehtonen, J. Harju, Controlled multicast framework, in: Proceedings of IEEE Conference on Local Computer Networks, 2002.
[22] T. Hayashi, H. He, H. Satou, H. Ohta, S. Vaidya, Requirements for Multicast AAA coordinated between content provider(s) and network service provider(s), IETF Internet draft, draft-ietf-mboned-maccnt-req-05.txt, 2007.
[23] S. Satou, H. Ohta, C. Jacquenet, T. Hayashi, H. He, AAA and Admission Control Framework for Multicasting, IETF Internet draft, draft-ietf-mboned-multiaaa-framework-08.txt, 2009.
[24] O. Karppinen, O. Alanen, T. Hamalainen, Multicast access control concept for xDSL-customers, in: Proceedings of the 3rd IEEE Consumer Communications and Networking Conference, 2006.
[25] S. Rafaeli, D. Hutchison, A survey of key management for secure group communication, ACM Computing Surveys (CSUR) 35 (3) (2003) 309–329.
[26] R. Pietro, L. Mancini, S. Jajodia, Efficient and secure keys management for wireless mobile communications, in: Proceedings of ACM International Workshop on Principles of Mobile Computing, 2002.
[27] N. Ishikawa, N. Yamanouchi, O. Takahashi, An architecture for user authentication of IP multicast and its implementation, IEEE/APAN Internet Workshop, 1999.
[28] C. Metz, AAA protocols: authentication, authorization, and accounting for the Internet, IEEE Internet Computing 3 (1999) 75–79.
[29] M. Steiner, G. Tsudik, M. Waidner, Diffie-Hellman key distribution extended to group communication, in: Proceedings of ACM Conference on Computer and Communications Security, 1996.
[30] S. Akl, P. Taylor, Cryptographic solution to a problem of access control in a hierarchy, ACM Transactions on Computer Systems 1 (1983) 239–248.
[31] R. Canetti, T. Malkin, K. Nissim, Efficient communication-storage tradeoffs for multicast encryption, in: Proceedings of Advances in Cryptology (EUROCRYPT), 1999.
[32] C. Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, Generic AAA architecture, RFC 2903 (2000).
[33] ITU-T Telecommunication Standardization Sector of ITU, Recommendation Y.2012, IPTV Functional architecture, ITU-T Y 2012, 2006.
[34] ITU-T Telecommunication Standardization Sector of ITU, Draft Recommendation Y.NGN-FRA R2, Functional requirements and architecture of the NGN (release 2), Y.NGN-FRA R 2, 2008.