Multicast receiver access control by IGMP-AC

Salekul Islam, J. William Atwood ∗

Department of Computer Science and Software Engineering, Concordia University, 1455 De Maisonneuve Blvd. West, Montréal, Québec, Canada H3G 1M8

Article info

Article history:
Received 26 May 2008
Received in revised form 27 November 2008
Accepted 3 December 2008
Available online 24 December 2008

Responsible Editor: L. Salgerelli

Keywords:
Secure multicasting
Internet Group Management Protocol (IGMP)
Receiver access control
Authentication Authorization and Accounting (AAA)
Extensible Authentication Protocol (EAP)

ABSTRACT

IP multicast is best-known for its bandwidth conservation and lower resource utilization. The present service model of multicast makes it difficult to restrict access to authorized End Users (EUs) or paying customers. Without an effective receiver access control, an adversary may exploit the existing IP multicast model, where a host or EU can join any multicast group by sending an Internet Group Management Protocol (IGMP) join message without prior authentication and authorization. We have developed a novel, scalable and secured access control architecture for IP multicast that deploys Authentication Authorization and Accounting (AAA) protocols to control group membership.

The principal feature of the access control architecture, receiver access control, is addressed in this paper. The EU or host informs the multicast Access Router (AR) of its interest in receiving multicast traffic using the IGMP protocol. We propose the necessary extensions of IGMPv3 to carry AAA information, called IGMP with Access Control (IGMP-AC). For EU authentication, IGMP-AC encapsulates Extensible Authentication Protocol (EAP) packets. EAP is an authentication framework to provide some common functions and a negotiation of the desired authentication mechanism. Thus, IGMP-AC can support a variety of authentications by encapsulating different EAP methods. Furthermore, we have modeled the IGMP-AC protocol in PROMELA, and also verified the model using SPIN. We have illustrated the EAP encapsulation method with an example EAP method, EAP Internet Key Exchange (EAP-IKEv2). We have used AVISPA to validate the security properties of the EAP-IKEv2 method in pass-through mode, which fits within the IGMP-AC architecture. Finally, we have extended our previously developed access control architecture to accomplish inter-domain receiver access control and demonstrated the applicability of IGMP-AC in a multi-domain environment.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

In the last few years, the Internet speed has increased tremendously. Even a regular PC user is enjoying an Internet speed of several megabits per seconds, which consequently shifted the End Users’ (EU) interest. They are no longer satisfied by browsing different sites, exchanging emails and communicating through text chat. Bandwidth-intensive applications including audio and video streaming, Voice over IP (VoIP), online games with high-resolution graphics and video-conferencing are becoming popular and mainstream applications. Many of the new applications (e.g., Internet TV, multi-player games,
distance learning, video-conferencing) rely on one-to-many or many-to-many communications [1], where one or more sources are sending data to multiple receivers. The Network Service Providers (NSP) could save a significant amount of bandwidth by using IP multicast for these types of applications. It is an efficient technology to deliver data to many recipients who may be geographically scattered in a wide area. For each recipient, multicast replicates data at a point as close to that recipient as possible. Thus, it is considered as a bandwidth conservation technique with respect to unicast, specially, when there are many recipients and a large amount of data (e.g., streaming video) is being sent. However, multicast, with all its advantages, is not widely deployed yet, although the Internet Engineering Task Force (IETF) commenced its standardization many years ago [2].

We have seen significant progress in the last decade in the area of multicast security, which is composed of two orthogonal planes: the data plane that protects multicast data and the control plane that secures multicast routing protocol messages. For secure data transmission, multicast data should be sent with authentication information and in encrypted format. For this purpose, the MSEC Working Group (WG) at the IETF has developed the Group Security Architecture [3] to maintain, update and distribute necessary keys only to the authenticated group members. Moreover, some group key management protocols (e.g., SIM-KM [4]) have been designed to protect multicast data. For control plane security, each multicast routing protocol is handled individually. For example, security of Protocol Independent Multicast–Sparse Mode (PIM–SM) [5] link-local messages is addressed in [6]. A secured routing protocol will resist an adversary that sends forged routing protocol message(s) to attack the Data Distribution Tree (DDT).

However, it is evident that in spite of the availability of the secured multicast model, large scale deployment of multicast has not materialized yet. This fact suggests the existence of a hole in the security of multicasting, and that hole is lack of access control over the multicast groups. Access control is used in this paper to mean Authentication Authorization and Accounting (AAA) functionalities for both sender(s) and receivers of a multicast group. The classical “anyone can send, anyone can receive” multicast model has created a circular cause and consequences phenomenon: the NSPs are not supporting multicast as there are no applications, and multicast applications are not being developed due to lack of support. The only way to come out of this situation is to make it feasible to generate revenue from the EUs. However, with no identification of participants, the classical model is making it impossible for the NSPs and the Content Providers (CP) to generate revenue when deploying multicast based applications.

AAA protocols [7] are being used very successfully to ensure revenue generation in unicast communication, by controlling access to network resources. In recent years, the specific aspects of (End User) authentication and authorization have been delegated to the Extensible Authentication Protocol (EAP) [8], which is a versatile framework that facilitates the use of multiple authentication methods, such as pre-shared secret, one time pass-word, public key authentication, etc. EAP is also useful for managing access at the application layer, for cases where authorization to access the network does not automatically include access to a particular application or service. In particular, EAP procedures can be adapted for use in multicast-based applications, to authenticate the users, to authorize them, and to account for their group-level activity. This will imply that there are two separate AAA/EAP uses, one to control access to the network as a whole, and the other to control access to specific services (e.g., multicast groups) provided by the network. We have already designed an overall access control architecture for secure and accountable multicasting [9,10] to incorporate the AAA framework into the existing multicast model. Furthermore, we have recognized e-commerce communication and policy enforcement as an integral part of the access control architecture.

To permit coupling network-level forwarding control with group-level access control, we previously proposed an extension to the (network-level) Internet Group Management Protocol version 3 [11] to permit it to carry group level authorization data, and validated its operation using a simple password-based scheme [12]. We called this extended protocol IGMP with Access Control (IGMP-AC).

This paper addresses the key component of the access control architecture [10], receiver access control, by further developing the functionality of IGMP-AC and the environment in which it will operate. There are four contributions of the paper. Firstly, the scope of IGMP-AC has been broadened in terms of authentication methods it provides, by encapsulating EAP [8] packets. Secondly, the operation of IGMP-AC in the more general multiple-round-trips scenario has been validated using SPIN [13]. Thirdly, the security properties of an EAP method, EAP Internet Key Exchange (EAP-IKEv2) [14], have been verified using AVISPA [15]. The AVISPA model has reported a Man-in-the-Middle attack in the EAP-IKEv2 method, which has been fixed by adding an optional message. Finally, an inter-domain receiver access control architecture is proposed. This demonstrates how IGMP-AC would be deployed in the case of inter-domain multicast groups.

The rest of the paper is organized as follows: Section 2 discusses the necessary background knowledge, such as IGMPv3, AAA protocols and the multicast access control architecture. Section 3 explains the security vulnerabilities and possible attacks a multicast group might face due to the present IGMP protocol. Section 4 briefly presents the existing methods and their limitations. Section 5 describes the IGMP-AC protocol in detail. A model of the operation of IGMP-AC in PROMELA [16] and the verification of this model using SPIN are presented in Section 6. The next two sections present how EAP is used for authentication, and the validation of the security properties of EAP-IKEv2 method using AVISPA. Section 9 presents the inter-domain receiver access control architecture and defines the behavior of IGMP-AC in this architecture. Section 10 discusses different issues related to IGMP-AC, such as scalability, packet delivery time, message complexity and usability for mobile receivers. Finally, Section 11 concludes the paper and summarizes the work that we have planned for the future.
2. Background work

Deployment of present multicast-based applications consists of four steps or processes. Fig. 1 shows the schematic diagram of these four processes.

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

2. An EU or host will communicate with its local Access Router (AR) to join/leave a multicast group, using Internet Group Management Protocol (IGMPv3) [11] in IPv4 or Multicast Listener Discovery (MLDv2) protocol [17] in IPv6. This membership information will be updated periodically.

3. The third component is the multicast routing protocol (e.g., PIM–SM [5]). Several such protocols have been specially formulated to support multicasting. The Core Routers (CR) will use one of the routing protocols to build the DDT.

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

2.1. Internet Group Management Protocol (IGMP)

Internet Group Management Protocol version 3 (IGMPv3) [11] has been standardized by the IETF for IPv4 systems (host or router) to inform the neighboring router(s) about the multicast group memberships of these systems. IGMPv3 supports "source filtering" through which a system can request to receive multicast packets only from a specific list of sources, or from all but specific sources, sent to a particular multicast group. IGMP performs three main operations:

- A system sends a **join message** when it wants to join a multicast group or some specific sources of a group.
- A system sends a **leave message** when it wants to unsubscribe from a multicast group.
- A router periodically checks which multicast groups are of interest to the hosts that are directly connected to that router.

A **Membership Query Message** is sent by the IP multicast routers to update the multicast reception state of neighboring interfaces. There are three types of queries. A General Query will give the complete multicast reception state to the multicast router. A Group-Specific Query will inform the router of the reception state with respect to a single multicast address. If the router wants to learn the reception state with respect to a list of sources of a specific multicast group, it will send a Group-and-Source-Specific Query.

An end system will send a **Membership Report Message** to the neighboring routers to inform them of the present reception state or any change in the reception state of that system.

Note that MLDv2 [17] provides similar functionalities for IPv6 systems that IGMPv3 does for IPv4. This paper is presented in terms of IPv4 operation, and the IGMP-AC protocol is designed to provide receiver access control for IPv4 systems. However, with little effort, a similar extension to MLD could be done to provide receiver access control for IPv6 systems.

2.2. AAA protocols

The AAA framework [7] has two components: AAA Server (AAAS) and Network Access Server (NAS). The AAA framework operates in a client-server manner, where a NAS acts as a client and communicates with the AAAS using one of the AAA protocols. The AAAS – multiple servers can be used for resiliency – is attached to the network and it serves as a central repository for storing and distributing AAA information. The device acting as the point of entry into the network is typically a NAS (although it could also be a router, a terminal server, or perhaps another host) that contains an AAA client function. The AAAS maintains the database of user profiles and configuration data, and provides distributed services to NASs.
The most widely deployed AAA protocol is Remote Authentication Dial In User Service (RADIUS) [18]. RADIUS was initially developed for dial-up PPP (Point to Point Protocol) and terminal server access. Over the years, the Internet has moved forward and many advanced technologies, including wireless, DSL, Mobile IP and Ethernet have evolved. As a result, the task of the NAS has become very complex and dense, and RADIUS is not capable enough to deal with all these issues. The next generation AAA protocol is called Diameter [19]. It has a mandatory-to-implement base protocol, and all other extensions and applications (i.e., Cryptographic Message Syntax (CMS), NASREQ and Mobile IP) will be implemented on top of it. It is a light-weight, simple to implement and easily extensible protocol with a large Attribute-Value Pair (AVP) address space. This ensures the future extensibility of Diameter.

2.3. Access control architecture with e-commerce communication

We have developed an architecture for securing the entire multicast distribution tree, accommodating the necessary e-commerce components, and sharing the revenues generated. The architecture shown in Fig. 2 was first proposed in [9], and further illustrated and extended in [10]. A number of parties that participate in a multicast session, either before the session or during it, have been identified. Before explaining the architecture, it is necessary to understand the classification we have introduced for two types of multicast groups. First, an open group – to/from which data can be sent/received any time, and second, a secured group – to/from which only authenticated and authorized member(s) can send/receive data. On the other hand, a multicast group can be classified according to the distribution of its group members (both sender(s) and receivers). In an intra-domain group, all the members reside inside a single domain or Autonomous System (AS), whereas an inter-domain group consists of members distributed among more than one domain or AS. This distinction is important for routing protocols (such as PIM–SM [5]) that are inherently single-domain for ASM groups and multi-domain for SSM groups.

The determination that a group is open or secured is (relatively) easy within a single domain, since group properties are all overseen by a single administration. Discussion of this issue for inter-domain groups is deferred to Section 9.

In Fig. 2, the CP offers the product to be delivered, and makes use of a Content Server (CS) to send to the multicast group. The EU receives the content. The NSP delivers the data, making use of ARs, Core Routers (CR), one or more instances of an AAAS and a NAS associated with each AR. To give the EU a single point of contact for a variety of services, we will assume the existence of a Merchant (MR). Finally, we will assume that the ability of the End User to pay for services will be certified by a Financial Institution (FI). The GO is responsible for the creation and overall activities of the group. We have assumed that GO and MR are co-located and reside in the same entity. However, this constraint can be removed given appropriate communications between them. The architecture we have developed is composed of three functional areas, which are briefly explained in the following.

2.3.1. Participant access control

Participant access control can be further divided into receiver and sender access controls. Receiver access control will be implemented at the EUs’ end. In Fig. 2, the directly connected router (i.e., AR2) to the subnet of the EUs will perform two tasks: it will receive and process the IGMP messages carrying AAA information of the EUs and will also act as a NAS by communicating with the AAAS. Hence, each EU would be authenticated and authorized by the one-hop AR before allowing him/her to join a secured group. However, the present standard of the IGMP protocol [11] does not make any provision of carrying AAA information. Therefore, there is an immediate need for extending the IGMPv3
protocol. Sultana and William Atwood [20] proposed combining AAA protocols, a public-key-based host identity, and a minor extension to the IGMP. However, their proposal lacks the support of EAP, which is the present standard of the IETF for flexible authentication. Hence, we proposed a different extension of IGMPv3 (i.e., IGMP-AC), which was partially presented in [12]. In this paper, we are presenting the complete version of IGMP-AC.

Sender access control will be deployed at the sender’s end, where AR1 will authenticate and authorize the CP with the proper interaction of the AAAS. On successful authentication and authorization, an IPSec SA will be established between the CP and AR1 to cryptographically authenticate each packet before forwarding it to the DDT. Hence, a multicast packet that fails to be authenticated will not be forwarded to the DDT. We have presented our sender access control method in [21].

2.3.2. e-Commerce communication

An access control architecture designed to be deployed for revenue generation would be indeed incomplete without secure e-commerce communications. MR, FI, EUs and AAAS are involved parties in e-commerce communications in Fig. 2. In the multicast case, there is one supplier, but many purchasers. In addition, the set of purchasers is likely to be unknown (at least in detail) to the supplier. This implies that considerable adaptation of the unicast e-commerce protocols will be necessary to meet the accounting and revenue-generation goals. Venkataiahgari et al. have specified a set of protocols among the e-commerce participants and validated their proposal [22].

2.3.3. Policy enforcement

A policy is a set of rules that dictates the behavior and implementation of a system or service. Security policies specify the conditions to be fulfilled before allowing access to some restricted information or resources. There is a need for a flexible, scalable and easily adaptable policy framework for multicast access control for the successful deployment of the architecture shown in Fig. 2. For simplicity, the policy server and other entities related to the policy framework are not shown in Fig. 2. The GO/MR could specify rules for accepting clients, specific tokens (identities) to be matched, keys to distribute once a client is accepted, or accounting data to be collected, etc. The GO/MR can then provide these policies to the AAAS associated with potential clients, either by pushing them to likely servers, or by having the AAAS pull the policy when triggered by user requests to join a group. In Ref. [23], a policy framework for multicast group access control has been proposed. It uses the eXtensible Access Control Markup Language (XACML) [24] for policy specification, and the Security Assertion Markup Language (SAML) [25] for carrying policy information.

2.3.4. Limitation of the architecture

The architecture we have developed is the first (to our knowledge) that addresses participant access control with necessary e-commerce communication and policy enforcement. However, the architecture shown in Fig. 2 assumes that all the participants reside in a single domain or network. This is a restrictive assumption. In reality, different participants may be distributed over various domains and inter-domain communication will be required. It is worthwhile to note that, specially, for participant (i.e., sender and receiver) access control and policy enforcement, the access control architecture needs to be extended to support inter-domain groups. We deal this issue in Section 9, where we have proposed an inter-domain receiver access control architecture in which the CP, EUs and GO are distributed over different domains.

3. Problem definition

We have already stated that receiver access control of our access control architecture (shown in Fig. 2) will be addressed in this paper. In this section, we will explain how vulnerable the IGMP reception states are in the absence of receiver access control. We will also discuss why a group key management protocol fails to provide receiver access control, and how receiver access control could be achieved by coupling EU authentication and authorization with IGMP operations. On the other hand, an effective receiver access control must protect the IGMP reception states.

3.1. Effects of forged IGMP report messages

A multicast routing protocol builds the DDT among the CRs, while the ARs extend that tree up to the edges of the network. The extent of possible damage due to a forged report message depends on the type of the message accepted by the AR. There are three types of IGMP report messages.

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

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

3. Current-state message: A report message that a host sends periodically to keep alive the existing reception states maintained by the ARs. A forged current-state message may cause the ARs to think there are group members of a group on a network when there are not.
A forged report might be sent targeting an intra-domain or inter-domain multicast group. In an intra-domain attack, receivers and senders are in the same domain, and it is easy to prevent the attack by enforcing access control. In an inter-domain attack, receivers and senders are not in the same domains and the multicast DTD may cross the receivers' and/or the senders' domains. Therefore, access control is not easy and may require inter-domain communication.

In the IGMPv3 specification [11], a list of attacks has been summarized that might be generated due to forged IGMP messages. The consequences of these attacks are wastage of the local subnet's bandwidth and of the resources of the hosts and the routers. IGMPv3 recommends the use of IPsec [26] with Authentication Header [27] to authenticate IGMP messages. IGMPv3 has been developed to be deployed for open groups only. Hence, the suggested security features of IGMPv3 are unable to enforce the access control for the IGMP report messages. It should be noted that the attacks mentioned earlier could be prevented by implementing access control but not by implementing message authentication only [28].

3.2. Goals of receiver access control

In this paper, “access control” is used to address all the AAA functionalities for multicast receivers or EUs: to authenticate an EU successfully, to verify authorization to receive group data for which an IGMP-AC Report message has been sent, and also to keep accounting information for each EU. However, joining/leaving a multicast group is performed by the IGMP protocol. If we take into consideration the vulnerabilities due to the forged IGMP report messages, access control will be reduced to two goals:

1. Only an authenticated and authorized EU will be allowed to modify the reception states of a secured group.
2. Accounting will be accomplished for every EU, who participates in the activity of a secured group.

It is worthwhile to mention that access control will be enforced only for the members of the secured groups. Therefore, the effects of the forged IGMP report messages will be still present for the open groups. However, we are not addressing the security of open groups in this paper. Moreover, we expect that activity of an open group will be restricted to a closed network, which is not accessible from outside that closed network.

3.3. Group key management vs. receiver access control

In the literature, there exist many group key management (GKM) protocols (e.g., Gothic [29]) that provide receiver access control for multicast groups. It should be noted that their definition of “access control” is a subset of the access control (i.e., AAA functionalities) we have addressed in this paper. Most of the GKM protocols only provide indirect EU authentication and a crude authorization.

In a GKM system, an EU has to register with the Group Controller/Key Server (GCKS) to receive the Data Encryption Key (DEK), which is used to encrypt/decrypt multicast data. The EU authentication is performed during registration. Hence, the possession of the right DEK implies an authenticated member. Next, an authenticated member joins the secured group by sending an ordinary IGMP message. Therefore, there is no direct authentication while joining a multicast group, and an adversary can always join a secured group. It should be noted that group key distribution is happening in the application layer whereas IGMP runs between the network layer and the transport layer. Therefore, the two authentications - authentication through IGMP to join a secured group and authentication to get the group keys – should not be coupled and performed in the same layer. By deploying a GKM protocol no flexible or sophisticated authorization could be achieved. The only way to authorize a member is to play with the DEK. The third issue, accounting, could never be achieved by a GKM protocol. However, accounting is an essential element for a revenue generating application. Therefore, it could be concluded that a GKM could never be a solution for multicast receiver access control. We are clearly distinguishing the functionalities of a GKM protocol and the receiver access control mechanism that we have developed.

3.4. Relationship of receiver access control to key management and accounting

As noted in Section 3.3, manipulation of group keys is not a good way to achieve our access control goals. However, the functions of group key management are essential for achieving the integrity and confidentiality of multicast data. Therefore, our proposals must be used in conjunction with a GKM system. In simple situations (for example, if forward and backward secrecy are not required, and simple key hygiene is sufficient), it is possible that the group keys will be managed at the application layer (i.e., entirely within a GKM system), with the initial keys distributed to the End Users prior to the initiation of the network-level join. An example application of this type is scheduled video streams. The group could be entirely re-keyed every 10 min or every 30 min, with accounting policies specified in 10-min or 30-min “chunks”.

In situations where keys must be dynamically managed, and some form of Logical Key Hierarchy or other key management scheme [4] is required to ensure protection from the “one affects n” problem, the necessary keying operations (in the AR and in the interior routers in the network) will be specified as policies for the group, and communicated to the appropriate routers using mechanisms specific to this key management. As such, key management can be considered to be a separate problem, with its own solution space. It is to be noted, however, that this key management will probably make use of the positive identification provided by our proposals.

To achieve the accounting goals, the policies specified in the AAAS for a particular group will be communicated to the AR at the time that the acceptance decision is rendered, using standard AAA procedures. This approach separates
accounting from access control, allowing arbitrary policies (pay-per-view, flat rate, subscription, free access, etc.) to be specified.

3.5. Receiver access control through extended IGMP

We have reasonably established that one of the major weaknesses of the existing multicast service model is the IGMP protocol, which fails to provide any sort of receiver access control. However, IGMP is the only means to join/leave (not considering the silent leave) a IP multicast group. Therefore, receiver access control must be enforced by the ARs during the join/leave operation (which is performed by the IGMP protocol) of a secured group. Therefore, the NSPs should deploy an access control mechanism at the ARs, which will enforce a mandatory authentication and authorization of EUs before joining or leaving a secured group through IGMP. Therefore, each time an EU sends a state-change join/leave message, it will trigger an authentication session between the EU and the AR. If a current-state message (targeting a secured group) comes from an EU who had not been authenticated and authorized previously, the message will be dropped by the AR.

3.5.1. Coupling access control with IGMP

The problem we are addressing in this paper is composed of two closely related sub-problems: deploying receiver access control and protecting a multicast secured group from the vulnerabilities raised due to the unprotected IGMP reception states. The problem could be solved by performing access control (i.e., authenticating and authorizing an EU) each time an EU wants to join/leave a secured group. Before going further detail one important issue should be solved, and that is, coupling access control with IGMP operations. It should be noted that a solution that does not couple EU authentication and authorization with the joining/leaving from a secured group, will be vulnerable to the Man-in-the-Middle (MitM) attack without proper channel or identity binding. Therefore, the secured way of solving the problem would be transporting EU authentication information (e.g., password or shared secret) encapsulated over the IGMP protocol.

3.5.2. Extending the IGMPv3 protocol

Although researchers have been devoting much effort to the development of secured group communication and secured routing protocols, their efforts fail to provide necessary functionalities for receiver access control. Therefore, it could be concluded that our design goal has narrowed down to transporting EU authentication information inside IGMP messages. The current IGMP specification (i.e., IGMPv3 [11]) does not provide any such facility. Therefore, the IGMPv3 protocol must be extended properly. We will present our extended version of IGMP in Section 5.

4. Related work

AAA issues are not addressed well in IP multicast by the researchers. We have found only a few attempts that meet our requirements. We have mentioned already that a number of researchers have proposed EU authentication in conjunction with group key management. Some of these methods have been designed for “group communication” instead of multicast communication, where the underlying communication technique is not necessarily IP multicast. Moreover, a number of group communication protocols do not use IGMP at all for joining/leaving a group. Hence, in the following, we have excluded group key management mechanisms in our list of related work.

The End User Identification and Accounting (EUIA) [20] system deploys AAA protocols and a Group Policy Server for EU authentication, authorization, accounting, and host identification. In this model, an AR acting as a NAS, authenticates an EU by communicating with the AAAS, and the Group Policy Server will confirm the authentication status of the EU. EUIA extends IGMPv3 by adding two new messages. Although EUIA supports different types of authentications, it fails to provide a flexible authentication framework (e.g., EAP [8]).

The Internet Group Membership Authentication Protocol (IGAP) [30] provides all the functionalities of IGMPv2 [31] with the addition of EU authentication and accounting by deploying RADIUS [18] protocols. It supports simple password authentication and challenge-response or CHAP authentication. It suffers from a serious threat due to its use of plain-text passwords for authentication. Moreover, it does not provide the “source filtering” feature.

A multicast architecture has been presented in [32], where a centralized Multicast Manager (MM) for each domain has been introduced. When a Leaf Network Router (LNR) or AR receives an IGMPv3 report message, the LNR relays the message to the MM, which authenticates a host or receiver using its IP address only. The MM uses the “source filtering” option of IGMPv3 to authorize or control the access of the receivers. This model clearly fails to provide a secured access control mechanism. The relayed messages are not even integrity protected and IP address based authentication is always vulnerable to the address spoofing attack.

Ishikawa et al. [33] deploys AAA architecture to authenticate each user (sender and receiver). They proposed a challenge-response protocol with shared secret for EU authentication. In this model, a host willing to receive data should communicate with the nearest egress router or AR using an extended IGMPv2 protocol, and a challenge-response session between the egress router and the host will be initiated. The egress router collects authentication information and forwards it to a back-end AAAS to perform the actual authentication. This approach and all other subsequent ones that we are going to discuss do not provide any authorization and accounting features.

Multicast receiver access control based on a one-time token is presented in [34]. This token is digitally signed, and distributed by the Key Server (KS) to the users. This token contains a symmetric key called the IGMP-key that is used by the receiver to authenticate the IGMP Report message. The KS also distributes the same token to the multicast routers.

A proposal to modify IGMPv3 to authenticate a Report message when needed [35] was presented in a (now
Internet Draft (I-D). The goal of this I-D is to protect the IGMP reception states from any malicious host or EU by authenticating the IGMP messages. It added two messages to the IGMPv3 protocol: Authentication Query and Authentication Report. A host sends authentication information inside an Authentication Report message in the format of 32-bits words only once. This method is not applicable for an authentication (e.g., IKEv2 [36]) that requires multiple round-trips.

A summary of the different approaches regarding AAA issues is presented in Table 1. We have listed only the methods that have goals that are similar to our research goals. Comprehensive lists of receiver and sender access control mechanisms including group key management protocols can be found in [37,38]. Analyzing the existing methods we have identified the following essential findings:

1. Only a few models deal with authorization and accounting issues in IP multicast.
2. Sender authentication is overlooked most of the time.
3. Some of the methods are based on IGMPv2, which has already been outdated by IGMPv3.
4. Most of the architectures are confined to a specific authentication mechanism, and they do not support a wide range of authentication schemes.
5. The unique advantage of IGMPv3 over IGMPv2 is “source filtering”, which is absent in most cases.
6. Only one method [33] has successfully used an AAA protocol (RADIUS). Two others, EUIA and IGAP, intend to use AAA protocols, but have not presented any detail at the architectural level.

Finally, the MBONE Deployment (MBONED) Working Group at the IETF is developing the functional requirements for accounting and access control for multicasting. A multicast network will be identified as “fully AAA enabled” when it fulfills the requirements defined in [39]. They have classified different business models with respect to Content Delivery Services. Moreover, a framework to satisfy these requirements has been presented in [40]. However, they have designed a hypothetical framework, and they have not presented any detail or identified different modules of the framework yet. While the intent of this framework is to provide a mechanism for user tracking and billing, and the associated access control, it does not address the interaction among the EU, his/her Financial Institutions (FI), and the collection of Merchants (MR) who might offer the content for purchase. This is in keeping with the areas of interest of the IETF. However, it is our belief that any proposed solution must mandate a simple control interface between the NSP and the e-commerce world. This will allow the EU to purchase individual “items” from a variety of MRs, for delivery via the facilities of the NSP.

5. IGMP with Access Control (IGMP-AC)

The Internet Group Management Protocol with Access Control (IGMP-AC) performs access control of the EUs (only if required for a specific application) in addition to the IGMPv3 functionalities.

5.1. Requirements

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

1. The IGMP-AC will provide a generic client-server authentication protocol, where the EU will act as a client, the AAAS will act as a server and the AR will perform the forwarding task. Thus, any suitable authentication protocol (e.g., EAP [8]) having client–server entities can be incorporated with the IGMP-AC architecture.

Table 1
Summary of different approaches regarding AAA issues.

<table>
<thead>
<tr>
<th>Approach</th>
<th>IGMP version</th>
<th>User authentication</th>
<th>Authorization</th>
<th>Accounting</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUIA [20]</td>
<td>IGMPv3</td>
<td>Flexible authentication</td>
<td>Yes</td>
<td>Yes</td>
<td>Does not provide flexible authentication framework. Does not specify protocol for communication between NAS and Group Policy Server</td>
</tr>
<tr>
<td>IGAP [30]</td>
<td>IGMPv2</td>
<td>Password (mandatory) or CHAP</td>
<td>Yes</td>
<td>Yes</td>
<td>Sends plain-text password. Based on IGMPv2</td>
</tr>
<tr>
<td>IGMPv3 to multicast access [32]</td>
<td>IGMPv3</td>
<td>By using host IP address only</td>
<td>Uses source filtering of IGMPv3</td>
<td>No explicit mechanism</td>
<td>Does not support any advanced authentication scheme. Single point of failure</td>
</tr>
<tr>
<td>Authentication using RADIUS [33]</td>
<td>IGMPv2</td>
<td>CHAP</td>
<td>No</td>
<td>No</td>
<td>Provides sender authentication also. Suffers from replay attack. Does not provide authorization and accounting mechanism</td>
</tr>
<tr>
<td>IGMP key establishment [34]</td>
<td>IGMPv2</td>
<td>Access token signed by digital signature</td>
<td>No</td>
<td>No</td>
<td>A Group Key Management protocol must be in place beforehand. Adds overhead to end host and multicast routers</td>
</tr>
<tr>
<td>Upload authentication info by IGMPv3 [35]</td>
<td>IGMPv3</td>
<td>No specific scheme</td>
<td>No</td>
<td>No</td>
<td>Supports source filtering feature. Not applicable for every authentication mechanism</td>
</tr>
</tbody>
</table>
2. The EU authentication process should support all sorts of authentications – from simple password based authentication to certificate based authentication. Hence, depending on the required level of security, the NSP and the GO will have freedom to pick a specific authentication mechanism.

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

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

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

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

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

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

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

5.2. Protocol descriptions

It is important to clarify when the IGMP-AC protocol will perform access control. A host or end system could join/leave any time to/from an Open group by sending regular IGMPv3 messages. The access control mechanism of the IGMP-AC will take place to join/leave to/from a secured group. While the IGMP-AC supports "source filtering", a secured group will be either Group-Specific, (\(G\)) or Group-Source-Specific, (\(S\), \(G\)). Here, "\(G\)" means absence of any specific multicast source address, and "\(S\)" means one or more multicast source address(es).

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

In the state diagrams, \(query\) is the standard IGMPv3 Membership Query Message and \(report\) is an extended IGMPv3 Membership Report Message that carries the user identity (\(id\)) and the filter mode (\(rstate\)). Using the Reserved fields in the Membership Report Message, this extension could be attained with little effort. \(auquery\) (Authentication Unicast Query), \(areport\) (Authentication Report) and \(aresult\) (Authentication Result) have been created for the IGMP-AC protocol. IGMP-AC has been designed to encapsulate EAP\[8\] packets: \(eap_req\) (Request), \(eap\_resp\) (Response) and \(eap\_result\) (Success or Failure). In Section 7, EAP encapsulation will be explained in detail.

5.2.1. Host behavior

The state diagram for a host that communicates with an AR using the IGMP-AC is shown in Fig. 3. Here, \(API\) means Application Program Interface or any upper layer protocol.

---

Fig. 3. State diagram for host.
of the three values: add, del or current. When a host receives a join message, it sends a report message to the AR that carries rstate equal to add, the identity or id of the EU and g_or_gs. Then, it creates a new reception state and assigns the filter mode with the value add. When it receives a leave message it sends a report message to the AR with rstate equal to del, the id of the EU and g_or_gs. The filter mode is also changed from current to del. The filter mode is assigned with the value current for the period of time the host maintains membership for a group. It is to be noted that the report message will not carry any authentication information.

In response to a query(g_or_gs), the host will send a report message with current status if it has an entry for g_or_gs.

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

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

5.2.2. Role of AAA Server (AAAS)

The state diagram for the AAAS that communicates with the AR using the AAA protocol Diameter [19] is presented in Fig. 4. We have assumed that grequest, ganswer, request, answer and account are messages of the Diameter protocol. The construction of these messages is out of the scope of this paper and is included in our future study. We mention that it is possible to extend the Diameter protocol by defining new AVPs while all data are delivered by the protocol in the form of AVPs.

By sending a grequest(g_or_gs), the AR wants to know if the type of the group for g_or_gs is Secured or Open. In response to this, the AAAS will send a ganswer with secured or open, depending on the type of the group.

If the AR receives an eap result packet, it will send the packet inside a request message to the AAAS. The purpose of sending a request is to verify the identity and the authentication information of the EU, and also if he/she is authorized to join the group, g_or_gs. If the EU is successfully authenticated and also authorized for the group g_or_gs, the AAAS will send an answer that encapsulates the eap_result packet. If authentication or authorization has failed, the eap result packet will reflect it. Authorization information will be sent only for successful authentication. If it requires more information for successful (or failed) authentication, the AAAS will send an eap_reqs packet inside an answer message. This is the case when more than one round trip is required (e.g., CHAP). The sequence number of eap_reqs has been incremented from n to n+1 to indicate that this is the next EAP Request message.

![Fig. 4. State diagram for AAA Server.](image-url)
The AAAS will update the user database indexed by the identity of EU, id, on receiving an account message. This database can be accessed later by the CP.

5.2.3. Role of Access Router (AR)

The state diagram for the AR is presented in Fig. 5. We have already explained all the messages of this diagram. We have summarized the functionalities of the AR in the following:

- The AR will maintain a list of existing groups, gtype. If a report with rstate equal to add/del and a new g_or_gs is received, it will send a grequest to the AAAS to inform itself about it. The AR will update the list gtype after receiving the ganswer message from the AAAS.
- If the AR receives a report with rstate as current, it will reset the timer, querytimer.
- If the AR receives a report with rstate as add/del for an Open group, the IGMPv3 protocol will be followed.
- If the AR receives a report with rstate as add/del for a secured group, it will create a new reception state with add filter mode if case of rstate equal to add or it will change the filter mode of the reception state (with g_or_gs) from current to del. The AR will also send an auquery to the host. Thus, authentication is only necessary during joining/leaving to/from a secured group. The auquery will carry the first EAP Request packet (eap_reqs) to trigger an EAP authentication method. It is worthwhile to mention that there is a potential vulnerability for DoS attack. An AR will create a new reception state with add filter mode if it receives a report with rstate as add. An adversary may create numerous forged report messages with rstate as add, and the AR will create thousands of new reception states. To protect against such a DoS attack, an AR

![State diagram for Access Router](image-url)
implementation could restrict the number of reception states an AR would be allowed to create with add filter mode. A similar mechanism has been suggested in the IGMPv3 specification [11], to protect against the DoS attack a host may face through forged Group-and-Source-Specific Queries.

- The AR will act as a pass-through device. When it receives an areport from the host, it will extract the eap_reqs packet and forward it to the AAAS inside a request message. If the AR receives an answer from the AAAS, it will extract the eap_result packet and forward it to the host inside an auquery message.

- If the authentication is successful and the rstate was equal to add, the AR will add a new reception state for (g_or_gs, id) by changing the reception state from add to current.

- If the AR is not already receiving multicast data for the secured multicast group G, for which the reception state (g_or_gs, id) has been changed from add to current, the AR will send a multicast routing protocol join to the upstream router.

- For successful authentication, with the rstate value as del, the AR will delete the reception state only for the EU. Next, an account message will be sent to the AAAS and an IGMPv3 query will be sent to all the hosts inside the subnet to query if any other host is interested to receive data from that group. Finally, the querytimer for the group g_or_gs will be set.

- In the case where the querytimer for the group g_or_gs has expired, all the entries with g_or_gs in reception states will be deleted. Moreover, the AR will send a multicast routing protocol prune message for group g_or_gs to the upstream router.

5.3. Additional messages

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

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

2. Authentication Report: In the above state diagrams, Authentication Report is presented by areport. It carries EU authentication information through eap_result packet. This report is sent by a host in response to an auquery only.

3. Authentication Result: The AR forwards authentication information of the EU to the AAAS and gets back the eap_result packet from the AAAS. This eap_result is relayed to the host by an Authentication Result or areport message.

In addition, the IGMPv3 Membership Report Message is extended to carry the user identity and the filter mode. These extra fields will be ignored for Open groups.

5.4. Required reception states

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

5.4.1. Reception states maintained by the host

A host will be informed through the Application Program Interface (API) or upper-layer protocol interface when it should perform a join operation to a multicast group. The host will be able to differentiate between an Open group and a secured group by checking the information received from the API or upper-layer protocol interface. If it is an Open group the host will follow the IGMPv3 standard [11]. In case of a secured group, the host will have to maintain a list of:

- **Group address:** The secured group address to which an EU has joined or sent an IGMP-AC report message to join from the host.
- **Source address:** For the Group-Specific membership the source address field will be empty and for the Group-and-Source-Specific membership this field will contain one or more source address(es).
- **Identity of EU:** An EU identity will be in the format of Network Access Identifier (NAI) (e.g., bob@example.com) [41]. A NAI consists of a user name (e.g., bob) and a domain name (e.g., example.com). This will be further discussed in Section 9 when we will present our inter-domain access control architecture.
- **Authentication information:** The content of authentication information will depend on the EAP method used for EU authentication. It might be a simple password, a shared secret key or a digital certificate.
- **Filter mode:** It will be one of the values add, del or current.

5.4.2. Reception states maintained by the AR

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

- **Group address:** The secured group address to which an EU has joined or sent an IGMP-AC report message to join.
6. Verification of IGMP-AC using SPIN

We have used the formal verification language PROcess MEta LAnguage (PROMELA) [16] to specify the verification model, and used the tool Simple PROMELA INterpreter (SPIN) [13] to verify our model.

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

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

6.1. Model description

We have developed the PROMELA model from the state diagrams of the IGMP-AC protocol presented in Section 5. We have designed the model in such a way that it should be as simple as possible, but at the same time, satisfy all the states and transitions of the diagrams. Our model consists of four processes: host for a host, ar for the AR, aaas for the AAAS and init to start the first three processes. The processes communicate with each other according to the architecture of Fig. 2. Thus, the hosts send messages to the AR using a common channel and the AR sends messages to the three hosts using three unique channels (to simulate sending of the unicast message, auquery). The AR and the AAAS communicate using separate channels. To model the communication that a host may receive messages from any upper layer protocol interface (see Fig. 3), another channel has been created.

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

6.2. Verification results

SPIN can be used for either simulation or verification. Once we were sure that our model was free from syntax error, different random simulation runs were performed with various options, and no errors were reported by the simulator. All these simulation runs have helped to build our confidence that our model is behaving as expected.

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

**Identity of EU:** An EU identity will be in the format of Network Access Identifier (NAI) (e.g., bob@example.com) [41]. A NAI consists of a user name (e.g., bob) and a domain name (e.g., example.com). This will be further discussed in Section 9 when we will present our inter-domain access control architecture.

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

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

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

In addition to the reception states, the AR maintains a list of multicast groups (i.e., gtype in the diagrams) of g_or_ga format. Whenever a report with an unknown group address (which is not present in the existing group list) arrives, the AR will communicate with the AAAS to update the list. This list is very simple and consists of

(group address, source address, status).

Here, the group address and the source address are the same as for the reception states, and the status is either Open or Secured.

5.5. Securing IGMP-AC messages

In RFC 3376 [11], IPsec [26] with Authentication Header (AH) protocol [27] has been suggested to use for the IGMPv3 messages to provide connectionless integrity, data origin authentication and replay protection. Thus, an AR will be certain that a received IGMPv3 message was originated from a system (or, more specifically, a system with the proper key) on the LAN to which the AR is directly connected. Two types of authentication mechanisms are possible, a symmetric signature algorithm with a single key for the LAN or an asymmetric signature algorithm, where a sender can be authenticated individually. We are adopting the same concept of using IPsec with AH for the messages of the IGMP-AC protocol. However, the AH protocol provides limited security services as the IGMP messages are not encrypted. Moreover, in case of manual key configuration, anti-replay is not supported either. After explaining the EAP authentication method, EAP-IKEv2 [14], we will present an enhanced security scheme to authenticate and encrypt the IGMP-AC messages in Section 7.3.

In RFC 3376 [11], IPsec [26] with Authentication Header (AH) protocol [27] has been suggested to use for the IGMP messages to provide connectionless integrity, data origin authentication and replay protection. Thus, an AR will be certain that a received IGMP message was originated from a system (or, more specifically, a system with the proper key) on the LAN to which the AR is directly connected. Two types of authentication mechanisms are possible, a symmetric signature algorithm with a single key for the LAN or an asymmetric signature algorithm, where a sender can be authenticated individually. We are adopting the same concept of using IPsec with AH for the messages of the IGMP-AC protocol. However, the AH protocol provides limited security services as the IGMP messages are not encrypted. Moreover, in case of manual key configuration, anti-replay is not supported either. After explaining the EAP authentication method, EAP-IKEv2 [14], we will present an enhanced security scheme to authenticate and encrypt the IGMP-AC messages in Section 7.3.
The simulator also produces a Message Sequence Chart (MSC) to demonstrate the interactions of different processes. A partial MSC of our model generated by the SPIN simulator is shown in Fig. 6.

Next, we have generated the verifier (a program written in C) from the PROMELA model using SPIN. This C program is compiled several times to produce the executable verifiers with different search techniques: exhaustive or full state exploration, depth-first, breadth-first, bitstate storage algorithm, hash compact, etc. The default search technique it uses is the depth-first search. However, we can change it to the breadth-first search using the compile time option, DBFS. When the verifier is executed, it looks for different errors such as assertion violation, invalid end state, non-progress cycle, never claim and unreachable code. In Table 2, the results of different verifications using different search techniques are listed. Here, “No” means absence of a specific error and “–” means that specific error was not looked for. Various depth levels reached by the verifier in search of errors are also presented.

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

7. Authentication using EAP

The Extensible Authentication Protocol (EAP) supports multiple authentication mechanisms. It has been implemented between a host and a router connected via switched circuit or dial-up line that uses PPP [42]. It has also been implemented between a switch and an access point that uses IEEE 802.1X [43]. The EAP runs between an authenticator and a peer, where the authenticator can act as a pass-through, and a back-end authentication server (or EAP server) may be connected with the authenticator. The actual authentication will be performed by the back-end server. The pass-through authenticator is nothing but a NAS and the back-end server is a AAAS. In such an environment, the EAP will be used by the EU (or host) and the NAS, and one of the AAA protocols will be used by the NAS and the AAAS. The EAP packets that arrive at the NAS are sent to the AAAS by encapsulating them inside the AAA packets, and the NAS will decapsulate the AAA packets received from the AAAS and forward the EAP packets to the EU. The Diameter EAP application that carries the EAP packets inside the Diameter packets between a NAS (EAP authenticator) and an AAAS (back-end authentication server) is already standardized by the IETF in RFC 4072 [44].

There are four types of EAP messages: Request, Response, Success and Failure. The Request is always sent by the authenticator to the peer, and the peer replies to it by sending a Response to the authenticator. The sequence of different messages between the NAS and the EU, and between the NAS and the AAAS, is shown in Fig. 7. The EAP Request and Response messages are sent inside the Diameter messages. Depending on the authentication method used, more than one round-trip may be required. In this scenario, after N round-trips, the AAAS authenticates the EU and sends an EAP Success message in-
side an EAP-Payload. If authorization is requested appropriate authorization AVPs are sent also.

EAP has not been developed for a specific authentication mechanism. It is an authentication framework to provide some common functions and a negotiation of the desired authentication mechanism. Such a mechanism is called an EAP method, and there are currently about 40 different EAP methods. Although EAP is standardized in RFC 3748 [8], the specification only supports the MD5-Challenge, One-Time Password (OTP) and Generic Token Card (GTC) based authentications. All these methods are commonly known as “legacy methods”. These methods have some short-comings: support one way authentication only, no key derivation is possible, vulnerable to some known attacks, etc. To satisfy present needs, a number of EAP methods have been developed, which have better security properties and provide more flexibility. The IETF has already standardized some of the methods, such as EAP-POTP [45], EAP-PSK [46], EAP-AKA [47], EAP-SIM [48], etc. The last two methods are specially developed for use in wireless networks.

Table 2
Verification results of IGMP-AC model using SPIN.

<table>
<thead>
<tr>
<th>Search technique</th>
<th>Compile option</th>
<th>Never claim</th>
<th>Assertion violation</th>
<th>Acceptance cycle</th>
<th>Invalid end states</th>
<th>Unreached code</th>
<th>Depth reached</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitstate storage algorithm</td>
<td>DBITSTATE</td>
<td>–</td>
<td>No</td>
<td>–</td>
<td>No</td>
<td>No</td>
<td>438</td>
</tr>
<tr>
<td>Breadth first search</td>
<td>DBFS DHC</td>
<td>–</td>
<td>No</td>
<td>–</td>
<td>No</td>
<td>No</td>
<td>437</td>
</tr>
<tr>
<td>Exhaustive &amp; Collapse compression</td>
<td>DCOLLAPSE</td>
<td>–</td>
<td>No</td>
<td>–</td>
<td>No</td>
<td>No</td>
<td>438</td>
</tr>
<tr>
<td>Non-progress cycle check</td>
<td>DNP DCOLLAPSE</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>–</td>
<td>No</td>
<td>873</td>
</tr>
<tr>
<td>Hash compact</td>
<td>DHC</td>
<td>–</td>
<td>No</td>
<td>–</td>
<td>No</td>
<td>No</td>
<td>438</td>
</tr>
</tbody>
</table>

7.1. EAP encapsulation over IGMP-AC

IGMP-AC has been designed to carry EAP packets, and thus, it will support any EAP method. It should be implemented in pass-through mode, although it can be used in peer-to-peer mode also. The protocol stacks implementing an EAP method on top of IGMP-AC are shown in Fig. 8. Thus, it will add an extra layer on the peer side between the EAP layer and the lower layer. IGMP-AC will act as the immediate lower layer of the EAP layer. A similar implementation can be found in [43], where the EAPOL (EAP encapsulation over LAN) messages encapsulate the EAP packets.

In RFC 3748 [8], six requirements have been identified for a layer to serve as the lower layer for EAP. We have extended IGMP-AC from IGMPv3 [11] by defining new messages and reception states only, and we are not going to modify the packet format or other functionalities of IGMPv3. Thus, in the following we have justified how IGMPv3 satisfies the requirements on an EAP lower layer protocol:

![Fig. 7. EAP and Diameter messages.](image-url)
EAP runs over an unreliable transport and thus lower layer reliability is not required. IGMPv3 does not offer any reliability. There is retransmission of messages in IGMPv3, but that is performed due to a specific state change (e.g., reception state or filter state).

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

- EAP does not depend on the lower layer security services such as confidentiality, integrity, replay protection, etc. However, the lower layer may have its own security services. IGMPv3 provides optional limited security services by implementing IPsec [26] with AH [27] protocol.

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

- If the lower layer is reliable, it may provide non-duplication stream to EAP. IGMPv3 is not able to remove duplicate packets. However, EAP with the help of its Identifier field, can detect duplicate packets. Thus, this is not a mandatory requirement for successful deployment of EAP.

- The lower layer of EAP should guarantee the ordered delivery of the EAP packets. Although nothing is given in the IGMP specification regarding ordering of packets, the IGMP packets are sent with Time-to-Live (TTL) = 1 and are never relayed beyond the subnet. An IGMP packet is always going to be picked up by the Designated Router or AR on a single hop. Thus, it is not likely that IGMP packets will go through reordering in a simple LAN structure. However, if the architecture of a LAN is very complex, IGMP packets may arrive out of order. In that case, EAP authentication failure may occur, which will cause EAP authentication to be re-run.

We have explored two other alternatives of using EAP by deploying Protocol for Carrying Authentication for Network Access (PANA) [49]. PANA is a link-layer agnostic network access authentication protocol that carries EAP authentication methods encapsulated inside EAP. PANA runs between a PANA Client (PaC) and a PANA Authentication Agent (PAA). The PAA is implemented inside a NAS, and PANA uses an AAAS for authentication. The architecture is similar to the EAP pass-through mode.

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

The second option we explored and finally discarded is encapsulating the PANA packets inside IGMP-AC the same way IGMP-AC is carrying the EAP packets in our present model. It will create an extra PANA layer over IGMP and thus, increase overall packet size, traffic, and per packet processing time in the sender and the receiver ends. Moreover, from the above discussion we have reasonably established that the IGMP-AC protocol is fully capable of carrying EAP packets without any modification.

### 7.2. EAP-IKEv2 protocol

In this paper, we are considering EAP-IKEv2 [14], an EAP method that is based on the Internet Key Exchange Protocol version 2 (IKEv2) [36]. It provides mutual authentication and session key establishment between an EAP peer and an EAP server. We have selected the EAP-IKEv2 method to demonstrate the encapsulation over IGMP-AC, due to its support of a variety of authentication techniques: asymmetric key pairs, passwords and symmetric keys. Most of the EAP methods are confined to a specific authentication mechanism. Moreover, EAP-IKEv2 follows the header format of IKEv2, which is a widely used authentication protocol. Thus, it will ease the deployment of EAP-IKEv2 method for the network administrators and security personnel.

The peer-to-peer version of the EAP-IKEv2 protocol is shown in Fig. 9, where P is the peer (EU in IGMP-AC) and S is the server (AAAS in IGMP-AC). The authenticator (AR or NAS) and some optional exchanges are not shown here. The first two messages are defined in [8], and are standard EAP Identity Request and Response messages. HDR is the EAP-IKEv2 header, which contains SPI. Two separate SPls are chosen by S and P for each direction on a per protocol run basis. In message 3, S informs P about the set of cryptographic algorithms it prefers (SAs), Diffie–Hellman payload (KEs) and its Nonce (Ns). Similar information is sent by P in message 4. The identity of P, IDp is sent in encrypted format in message 4 in the case of symmetric key authentication only. This payload is optional to send in the EAP-IKEv2 specification.

At this moment, a set of keys (e.g., SK_e, SK_a, etc.) are computed by both S and P according to [36]. These keys are used to generate the AUTH payload and the key, SK is used in the 5th and the 6th messages. Here, SK{x} means...
x is both encrypted (using SK_e) and integrity protected (using SK_s). Two sets of SKs are calculated and are to be used in the opposite directions. IDs and IDp are the identities of the server and the peer respectively. AUTH is the Authentication payload and is defined in [36].

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

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

In Fig. 10, we have shown the complete message sequence of a join scenario, where an EU joins a secured group using IGMP-AC. We can draw a similar message sequence when an EU is authenticated using EAP-IKEv2 method in pass-through mode. At that time, the first EAP-IKEv2 message, EAP-Request/Identity should be sent by the AR, instead of the AAAS. In Fig. 10, the eap_req1 message (inside the first aquery), which is sent from the AR to the host, should be replaced by the EAP-Request message. Similarly, the second EAP-IKEv2 message, EAP-Response/Identity(Id) will replace the eap_respl inside the first aresp message (from the host to the AR) and also the eap_respl inside the first areq message (from the AR to the AAAS). In this way, three pairs of EAP Request and Response messages will be required for a successful authentication. Finally, after three complete round trips, if the EU is authenticated and authorized, an EAP-Success, which replaces the eap_result in the last answer message of Fig. 10, will be sent.

7.3. Enhanced security for IGMP-AC messages

A standard EAP method provides mutual authentication and secret key derivation between the peer (host) and the server (AAAS) after a successful conversation. In pass-through mode, the authenticator (AR) sits between these two parties and relays messages from one to the other. Thus, the peer and the authenticator do not authenticate or exchange their identities. However, after a successful EAP communication, if further information is exchanged between the authenticator and the peer, that information can be protected using the keys established between the peer and the server. In case of IGMP-AC, a host will establish keys with the AAAS, while a host and the AR will communicate (send IGMP-AC messages) after a successful EAP run. We can protect any subsequent IGMP-AC messages, which are sent after the initial authentication and key establishment of the host and the AAAS. It should be noted that the initial IGMP-AC messages that are exchanged to complete an EAP authentication could not be protected using the keys established at the end of EAP session. To provide enhanced security for the IGMP-AC messages, the appropriate keying materials must be transported from the AAAS to the AR using an AAA protocol. In this way, we can provide confidentiality and other security services that are absent in the present IGMPv3 specification [11] as explained previously in Section 5.5.

The details are outside the scope of this paper, and the IETF is still working on the vulnerabilities of it [50,51]. The major weakness of this key transportation lies in the inappropriate binding of keys or channel bindings. The AR is not directly authenticated by a host, rather a host determines that the authenticator has been authorized by the AAAS by confirming that the AR has the same AAAS provided keying materials. Thus, this mechanism suffers from possible MitM attack as reported in [52]. However, following the guidelines of [50,51,8], this attack and other potential attacks can be solved.

8. Validation of EAP-IKEv2 method using AVISPMA

We have modeled the EAP-IKEv2 method using Automated Validation of Internet Security Protocols and Applications (AVISPA) [15] to validate different security properties that EAP-IKEv2 is designed for. Previously, we have used SPIN to verify the logical consistency of the IGMP-AC protocol. However, SPIN is not designed for verification of security protocols, and hence, it is not applicable for validating the security properties of the EAP-IKEv2 method. On the contrary, AVISPA is a push-button tool with industrial-strength technology for the analysis of different Internet security protocols and applications. It is being used by the developers of different security protocols and by academic researchers also. The security protocols standardized by the IETF have been analyzed by the AVISPA community, and some of the protocols have even been found to be flawed.

An AVISPA model of the EAP-IKEv2 method is available in [53], which is a peer-to-peer model (without the pass-through authenticator). We have started our validation with this model. We have found an attack on the method and extended it to remove the flaw by adding some messages. Finally, we modified the peer-to-peer model to a pass-through model by introducing the authenticator. The validations of the peer-to-peer model and the pass-through model are presented in Sections 8.2 and 8.3, respectively.

8.1. Security properties of the EAP-IKEv2 method

A number of security properties that are met by the EAP-IKEv2 method have been listed in [14]. We are not
using all the functionalities (e.g., fast reconnect) of the EAP-IKEv2 method, and thus, not all those properties are relevant to our model. Here, we are mentioning some of the important properties:

**Mutual authentication:** EAP-IKEv2 provides mutual authentication by the AUTH payload, which is either signed (for asymmetric key authentication) or hashed (for symmetric key or password authentication) by $P$ and $S$. The construction of the AUTH payload is described in [36].

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

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

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

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

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

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

### 8.2. The peer-to-peer model

The EAP-IKEv2 protocol shown in Fig. 9 is presented in Alice and Bob notation in Fig. 11 to validate its security properties using AVISPA. The name of the messages (i.e., EAP-Req and EAP-Resp) and the entire header payload (i.e., HDR) have been removed in this version as these portions of a message do not carry any security information. However, it is still possible to capture all the important security properties using the simplified version and validate these properties with AVISPA. In Fig. 11, “.” means concatenation, $\{x\}_K$ means $x$ is signed by the key, $K$, and $inv[K]$ stands for the private key corresponding to $K$.

#### 8.2.1. Limitations of the peer-to-peer model

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

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

2. EAP-IKEv2 provides different use cases in terms of authentication mechanism (asymmetric key, symmetric key, password based). In this model, only asymmetric key or digital signature based authentication has been validated.
1. \( P \leftarrow S: \text{request_id} \)
2. \( P \rightarrow S: \text{respond_id}.P \)
3. \( P \leftarrow S: S.A.KEs.Na \)
4. \( P \rightarrow S: S.A.KEp.Np.\{[IDp].SKp\} \)
5. \( P \leftarrow S: \{S.\{\text{AUTHs}\}_\text{inv}(Ks)\}_SKs \)
6. \( P \rightarrow S: \{P.\{\text{AUTHp}\}_\text{inv}(Kp)\}_SKp \)
7. \( P \leftarrow S: \text{success} \)

**IDp**: Identity of \( P \)

**SA**: Cryptographic algorithms from \( S \)

**SA**: \( P \)'s selected cryptographic algorithms

**KEs**: \( \exp(G, DHs) \), \( S \)'s Diffie-Hellman value

**DHs**: \( S \)'s Diffie-Hellman exponent (nonce of \( S \))

**KEp**: \( \exp(G, DHp) \), \( P \)'s Diffie-Hellman value

**DHp**: \( P \)'s Diffie-Hellman exponent (nonce of \( P \))

**Na**: nonce of \( S \)

**Np**: nonce of \( P \)

**SKp**: \( \text{hash}(Ns.Np.\exp(exp(G, DHs), DHp)) \)

**SKs**: \( \text{hash}(Ns.Nr.\exp(exp(G, DHp), DHs)) \), here \( SKp = SKs \)

**AUTHs**: \( S.A.\text{KEs}.Na.Np \)

**AUTHp**: \( S.A.\text{KEp}.Na.Na \)

**Fig. 11**: Alice and Bob notation of EAP-IKEv2 protocol.

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

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

5. In Ref. [36], a key seed, named **SKEYSEED** has been calculated first using the following equation:

\[
\text{SKEYSEED} = \text{prf}(Ns.Np.\exp(exp(G, DHs), DHp))
\]

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

### 8.2.2. Security goals

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

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

The High-Level Protocol Specification Language (HLPSL) specification of the peer-to-peer model is constructed by following the simplified protocol of **Fig. 11**. This specification has two roles, one for the server and the other for the peer. The built in function, \( \text{hash_func} \) of AVISPA is used as the function, \( \text{prf} \). An active intruder, who can play both roles of the server and the peer simultaneously has been introduced. Two parallel sessions of the protocol have been executed to find the replay attack. For validation, the HLPSL code is transformed to **IP** format using the translator **HLPSL2IP**. Next, we have used all four back-ends of AVISPA. The first two, **OFMC** and **CL-AtSe** have produced similar MitM attacks. The other two, **SATMC** and **TA4SP** have reported **NOT_SUPPORTED** and produced **INCONCLUSIVE** results.

#### 8.2.3. Finding the attack

The attack reported by the **OFMC** back-end is shown in **Fig. 12**. For space consideration and similarity of the attacks, the one produced by the **CL-AtSe** is not shown here. In this attack, the intruder is able to generate a MitM attack by opening two parallel sessions with \( S \) and \( P \). The intruder convinced \( P \) that he was talking with \( S \), when in fact \( S \) did not participate in the same session. Rather, the intruder relayed messages to \( P \) from the session he opened with \( S \). The intruder is not able to gain anything directly by performing this attack, as she has not accessed the secret key that \( P \) believes himself to have established with \( S \). Thus, she cannot exploit the flaw by intercepting any messages that \( P \) sends to \( S \). However, still this is a flaw of the protocol and it must be fixed with care. Sometimes, a flaw appears harmless and remains so for a while. However, it may shift to an unsafe one due to further extension or modification of the flawed protocol. The attack we have found is analogous to the one reported in [54] and originates from the use of IKEv2 for authentication.

#### 8.2.4. Securing the peer-to-peer model

The first modification we have made is addition of **Message ID** (**MID**) in the fifth and the sixth messages. As per [14], **MID** is sent as a field of the **HDR** payload in the third to the sixth messages. However, in the third and the fourth messages, **HDR** is not cryptographically protected. Thus, we have not added **MID** in these messages. In the fifth and the sixth messages, the integrity of **MID** is protected.
SKs for a pair of Request message. Thus, a specific value is assigned to response with the same MID. Here, $P$ using the keys derived from these two messages accordingly:

In spite of this modification, AVISPA reported the same MID is equal to one. If we look carefully at the ATTACK TRACE of Fig. 12, we will get the answer. The intruder is communicating with $S$ using its own identity, $i$. She has not produced any encrypted or authenticated message. She receives a message from $P$, and just relays it to $S$. To convince $S$ of her identity, she must send $\{i\}_SK$: in message 4. She does not have access to any secret key nor she can generate it. As a result, the intruder is not able to supply the appropriate fourth message to $S$. Thus, it is not possible for the intruder to mount any MitM attack.

8.3. The pass-through model

We are now prepared to develop our final model by extending the peer-to-peer one to a pass-through model. This will be accomplished by replacing the $eap\_reqs$ and the $eap\_resp$ messages with the simplified EAP-IKEv2 messages of Fig. 11 (with appropriate modifications explained in the previous section to preclude MitM attack). The replacement procedure is explained in Section 7.2 with the example of joining an EU to a secured group (see Fig. 10). We have made the appropriate changes in our HLPSL code by adding a new role of the NAS (or authenticator), which will do nothing but relay messages from $P$ to $P$ and vice versa. The intruder ability is extended also to perform the role of the NAS. Thus, an active intruder can play any role of server, peer or NAS, and even play multiple roles simultaneously. Two parallel sessions of the protocol have been executed to discover any replay attack.

First, we have verified our HLPSL specification using Security Protocol ANimator (SPAN) [55], a tool that interactively builds Message Sequence Charts (MSC) from HLPSL code. SPAN also provides an intruder mode that interactively builds attack traces from the HLPSL specification. Thus, by executing the Protocol Simulation option of SPAN we are able to generate a full trace or MSC of our specification. It confirms that the entities are communicating properly and the model is working as expected. Next, we have used all four back-ends of AVISPA to find an attack if any. The replacement procedure is explained in Section 7.2 with the example of joining an EU to a secured group (see Fig. 10). We have made the appropriate changes in our HLPSL code by adding a new role of the NAS (or authenticator), which will do nothing but relay messages from $P$ to $P$ and vice versa. The intruder ability is extended also to perform the role of the NAS. Thus, an active intruder can play any role of server, peer or NAS, and even play multiple roles simultaneously. Two parallel sessions of the protocol have been executed to discover any replay attack.

First, we have verified our HLPSL specification using Security Protocol ANimator (SPAN) [55], a tool that interactively builds Message Sequence Charts (MSC) from HLPSL code. SPAN also provides an intruder mode that interactively builds attack traces from the HLPSL specification. Thus, by executing the Protocol Simulation option of SPAN we are able to generate a full trace or MSC of our specification. It confirms that the entities are communicating properly and the model is working as expected. Next, we have used all four back-ends of AVISPA to find an attack if any. The first two, OPMC and CL-AtSe have reported SAFE for BOUNDED_NUMBER_OF_SESSIONS. The other two, SATMC and TA4SP have reported NOT_SUPPORTED and produced INCONCLUSIVE results. Due to the complexity of the model, we have to run OPMC with bounded depth. Finally, we can conclude that the pass-through model that we have developed is free from the attacks that AVISPA is able to find. Hence, the security goals we have mentioned in Section 8.2.2 have been validated.

9. Inter-domain receiver access control

An inter-domain multicast group is distributed over more than one domain, which makes it quite difficult to design a scalable access control solution. The AAA protocol, Diameter [19], is being used extensively in our architecture. Specially, the functionalities provided by the Diameter agents will facilitate the communication of different AAA entities in the inter-domain access control. Therefore, before presenting the inter-domain access control archi-
tecture, we are explaining the communications between the Diameter agents that we have deployed.

9.1. Diameter agents

Diameter is a peer-to-peer protocol (any node can initiate a request) with three types of nodes: NAS, agents and AAAS. Diameter agents – Proxy, Relay and Redirect – are Diameter nodes that do not authenticate and/or authorize messages by themselves. Diameter messages (i.e., Request and Answer) are always originated by the NAS or the AAAS. An agent sits between the NAS and the AAAS and forwards a Diameter message to the appropriate node. In the Diameter specification [19], the term “realm” is used to address administrative domain, and hence, the words realm and domain will be used interchangeably in this paper. Given that our proposed inter-domain access control architecture deploys Redirect and Relay agents, the roles of these two agents have been explained briefly in the following.

A Relay agent maintains a Realm Routing Table and on receiving a Request message routes the message to another Diameter node based on the information found in the Request message and the Realm Routing Table. A Redirect agent does not forward any message. However, it notifies the requesting node which route to be used next. Fig. 13 illustrates how a NAS communicates with a Home Diameter Server (HMS) (a remote AAAS, which is deployed outside the local realm) with the aid of the Redirect and the Relay agents. In this example, the NAS is an access control device for the user with identity, bob@example.com. The user identity is presented in the format of the Network Access Identifier (NAI) [41]. As we have mentioned before, bob is the user name and example.com is the domain name of this user identity. It is worthwhile to note that, a user identity must be globally unique and could be verified only by the HMS that maintains the user’s database.

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

9.2. Proposed inter-domain architecture

In Fig. 14, we have shown the inter-domain receiver access control architecture spanning three domains (i.e., NW1, NW2 and NW3), which has been designed by extending the access control architecture shown in Fig. 2. Given that this paper is devoted in receiver access control only, the other features of the architecture, such as sender access control and e-commerce communications are not shown here. Furthermore, to avoid complexity the Diameter nodes (i.e., AAAS, Relay agent and Redirect agent) are not shown for each domain. However, each domain has all three Diameter nodes, and the links between two Diameter nodes are secured following the Diameter specification. The sender, S, that sends data to the group, G, is connected with NW1, while the EUs are connected with NW2. These two domains are linked by two Border Routers (BR): BR1 and BR2. The Multiprotocol Extensions for BGP-4 (M-BGP) [56] will be used by the BRs to build the forwarding table. The DDT, which is being built by the CRs and the BRs, will be distributed over the two domains. The shape of the tree will depend on the underlying routing protocol. For example, in an SSM group [57], the DDT will be a shortest-path tree rooted at the source and the BRs will be the internal nodes of the DDT. We are also assuming that the GO is connected through NW3. The participants maintain

Fig. 13. Diameter agents: Redirect and Relay.

Fig. 14. Inter-domain receiver access control.
an account relationship with the AAAS of NW3, which is referred as Home AAAS (HAAAS). Hence, HAAAS has direct access to the participants' database and is able to verify the participants' identities.

9.2.1. Enforcing secured-group status for inter-domain groups

If the local NAS incorrectly determines that a group is open, then the multicast DDT could be pulled into the local domain. In addition, as noted above, the open/secured status of a group is determined by reference to the HMS. If the local domain does not, in fact, implement our procedures, it could attempt to pull the multicast DDT into the local domain. This problem of pulling the DDT must be addressed by protecting the DDT at the boundary points between adjacent domains. Procedures for doing this are specified in [58].

For successful implementation and co-existence of the Open and Secured groups, a simple mechanism is required to distinguish between two types of groups for inter-domain receiver access control. The easiest and most efficient way would be to delegate a range of class D multicast group addresses (which should be assigned by the Internet Assigned Numbers Authority (IANA) [59]) for secured group operations only. Hence, the routers, the AAA nodes and the end hosts will easily distinguish a secured group from an Open group. This procedure will reduce the communications between the AR and the AAAS, and therefore, will simplify the state diagrams of AR and AAAS, which are shown in Figs. 5 and 4, respectively. However, in case an IANA assigned block from class D addresses is not available, a global directory of the secured groups will have to be maintained and a AAAS will be allowed to access the directory securely to get the status of a multicast group.

9.2.2. IGMP-AC behavior

At first, an EU will send AR2 an IGMP-AC report message, which will carry the EU’s identity (in the format of bob@example.com) and the secured group address to which the EU is willing to join. On receiving the IGMP-AC report, AR2 will compose a Diameter Request message. Next, AR2 will have to decide to which Diameter node the Request would be sent. Given that the EUs and the AAAS are located in different domains in the example scenario shown in Fig. 14, Diameter agents of NW2 would have to be involved in inter-domain communication. By extracting the domain name from the EU’s identity (e.g., example.com) AR2 will understand that it would not be possible for the local AAAS (i.e., AAAS of NW2) to answer the Request. Hence, AR2 will forward the Request to the DRL of NW2. The DRL will follow the procedure explained in the previous section, and will communicate with the DRD (if required) to get the route of the HAAAS. Finally, the DRL will relay the Request to the HAAAS.

9.2.3. Distributed vs. centralized database

There are two ways of maintaining the database of participants for inter-domain access control. In centralized database method, the participants’ database would be maintained only by the HAAAS. A local Diameter node will act as a pass-through device and will never maintain any AAA information. Therefore, in our receiver access control architecture, if an EAP method requires n round-trips to be completed, n pairs of Diameter messages (Request and Answer) will be exchanged between AR2 and HAAAS in the distributed database method, the participants' database might be distributed over a number of local AAASes. When the HAAAS will receive the first Diameter Request originated by AR2, the HAAAS will search the participants' database using the EU’s username (e.g., using bob in case the identity is bob@example.com). If the AAA information of the EU is found, it will be forwarded to the local AAAS (i.e., AAAS of NW2). From then on, AR2 will communicate with the local AAAS directly to authenticate and authorize the EU.

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

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

10. Discussion

In this section, the scalability of IGMP-AC has been discussed. Moreover, the effects of IGMP-AC on packet delivery time and message complexity have been summarized. Finally, we have discussed extending the scope of IGMP-AC to wireless networks.

10.1. Scalability

In the case of an intra-domain multicast group, IGMP-AC will be a scalable solution. The number of EUs will not affect the scalability of the solution. However, in the case of an inter-domain group, if the HAAAS is far away from the LAAAS, the distributed database method (i.e., forwarding the AAA information of the EU to the LAAAS) will significantly reduce the communication overhead. Moreover, the number of EU will not largely affect the solution.

10.2. Delay in packet delivery

IGMP-AC will be implemented between the end host and the AR, and the rest of the DDT will be unaffected. Therefore, there will be very little impact on packet delivery time. IGMP-AC will slightly increase the join and leave latencies from the standard IGMPv3 due to EU authentication and authorization. The latencies will depend on the EAP method that is deployed for EU authentication and authorization. The latencies will be completed, n pairs of Diameter messages (Request and Answer) will be exchanged between AR2 and HAAAS. These messages will always be relayed through the DRL.

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

IGMP-AC has added three new messages (Authentication Unicast Query, Authentication Report and Authentication Result) to the IGMPv3 protocol. Moreover, the existing Membership Report message has been extended to carry EU identity. All these messages will be exchanged during the joining/leaving of an EU to/from a secured group.

10.4. Mobility of End Users

The scope of IGMP-AC could be broadened by deploying it in wireless networks for mobile receiver (or EU) access control. In addition, using the EAP Re-authentication Protocol (ERP), a secured and fast handoff of mobile EUs in wireless networks could be designed. The procedure has been presented in detail in [60].

11. Conclusion and future work

This paper presents the IGMP-AC protocol that facilitates the present multicast model with a light-weight and practicable receiver access control mechanism. In the way of developing IGMP-AC, we have incorporated different protocols and frameworks, which have been standardized by the IETF and are being used broadly by different communities. Therefore, our solution should easily fit within the existing Internet architecture. The major contributions of this paper are listed in the following:

- A set of requirements has been defined that an extended version of IGMP should provide for effective receiver access control. This set has been defined to overcome all the limitations of the previous methods.
- The complete version of the IGMP-AC that meets the listed requirements has been presented with detailed state diagrams and textual description. The list of additional messages and reception states has also been documented.
- The verification model of IGMP-AC has been developed with the formal language, PROMELA. This model has been verified using the tool, SPIN, which has reported the model free from any error. SPIN is a powerful tool, used frequently in industry and academia for formal verification of network protocols and distributed programs.
- The EAP encapsulation procedure over IGMP-AC has been explained elaborately. Exact locations of different protocols (i.e., EAP method, EAP peer, IGMP-AC, AAA, etc.) in the protocol stack have been identified. Moreover, the serviceability of the IGMP-AC as an EAP lower-layer has been justified by comparing the IGMPv3 properties with the requirements given for a layer to serve as the lower layer for EAP in [8].
- EAP encapsulation over IGMP-AC has been further demonstrated with an example EAP method, EAP-IKEv2. Any other EAP methods can be deployed following this example.
- The security properties of EAP-IKEv2 in peer-to-peer mode have been validated using an AVISPA model. A MitM attack has been found, which has been fixed by adding an optional message.
- The peer-to-peer model (that we have fixed from the MitM attack) has been extended to the pass-through model by introducing an authenticator (NAS) between the peer (host) and the server (AAAS). AVISPA reported the model free from any attack, which reinforces the security claims of the EAP-IKEv2 method in pass-through mode.
- An inter-domain receiver access control architecture that deploys IGMP-AC and Diameter agents has been presented. This is the first (to our knowledge) inter-domain receiver access control architecture that uses an extended version of IGMP.

It is our strong belief that the future impact of this paper on large scale deployment of IP multicast would be significant. A key component of the multicast access control architecture [10], receiver access control, has been presented in this paper. Specially, the inter-domain receiver access control architecture would motivate the service providers to consider IGMP-AC and our access control architecture as an implementable and realistic solution.

Security is not free from cost. By adding AAA functionalities, we have assigned extra tasks to the ARs and the hosts. A small amount of bandwidth will also be consumed by the messages sent for EU authentication. In future, we want to accomplish a performance study for our architecture by using a proper simulation tool. We plan to analyze the security properties of other EAP methods in pass-through mode, such as EAP-POTP [45] and EAP-PSK [46], which have already been standardized by the IETF. Finally, we have to define the packet formats of the messages and the values of different timers that are newly created for the IGMP-AC protocol.

Acknowledgements

J.W. Atwood acknowledges the support of the Natural Sciences and Engineering Research Council (NSERC) of Canada, through its Discovery Grants program. S. Islam acknowledges the support of Québec Government, through its Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) scholarship program and of Concordia University.

References

Salekul Islam is a Postdoctoral Fellow at the Energy, Materials, and Telecommunications Center of the National Institute of Scientific Research, Canada. He graduated from Bangladesh University of Engineering and Technology (BUET) in 2000 with a Bachelor of Science in Computer Science and Computer Engineering, and from Concordia University in 2003 with a Master of Computer Science and in 2008 with a Doctor of Philosophy in Computer Science. He was a Junior Lecturer in the School of Communication, Independent University Bangladesh (IUB), Dhaka, from September 2000 to April 2001. His research interests are in the design, analysis and validation of network protocols for secure multicast.

J. William Atwood graduated from McGill University in 1963 with a Bachelor of Engineering, from the University of Toronto in 1965 with a Master of Applied Science and from the University of Illinois at Urbana-Champaign in 1970 with a Doctor of Philosophy, all in Electrical Engineering. He is a Distinguished Professor Emeritus in the Department of Computer Science and Software Engineering at Concordia University, Montreal. His research interests include the design, analysis, validation and deployment of protocols for secure multicast, MPLS, and QoS signaling.