Multicast Receiver Access Control in the Automatic Multicast Tunneling (AMT) Environment

Veera Nagasiva Tejeswi Malla
J. William Atwood

Department of Computer Science and Software Engineering
Concordia University, Montreal, Quebec, Canada
Email: tejam.vns@gmail.com, william.atwood@concordia.ca

***Abstract*—The Automatic Multicast Tunneling protocol extends the range of multicast data distribution from a multicast-enabled network region to a network region that supports only unicast routing. Previous work has shown how to achieve access control in network regions that fully support multicast routing. In this paper, we show how to achieve the access control in the extended (unicast-only) network region, without modifying the original interactions of the access control. We also formally validate the security of our solution using the Automated Validation of Internet Security protocols and Applications (AVISPA) tools.**

***Index Terms*—Automatic Multicast Tunneling; Access Control; Unicast Network; Multicast Network.**

I. Introduction

Some applications require data to be delivered from a sender to multiple receivers. Examples of such applications include audio and video broadcasts, real-time delivery of stock quotes, and teleconferencing. A service where data are delivered from a sender to multiple receivers is called multipoint communication or Multicast. It provides an efficient way to support high bandwidth, one-to-many applications on a network. One major problem in IP multicast is that even hosts without any permissions are able to join multicast groups, i.e., there is no mechanism to prevent unauthorized users from accessing a multicast network. Consequently it became impossible for service providers to justify billing for multicast data usage.

To overcome the problem of revenue generation, *Participant Access Control* (PAC) was introduced in [1]. PAC includes *Sender Access Control* (SAC) [2] and *Receiver Access Control* (RAC) [3], [4]. RAC is a scalable, distributed and secure architecture, where authorized end users can be authenticated before delivering any data. Although PAC provides access control for IP multicast, it is limited to native multicast environments.

To overcome the requirement to support native multicast routing, a solution was proposed by the Internet Engineering Task Force (IETF) called Automatic Multicast Tunneling (AMT). Without requiring any manual configuration, AMT allows a device in a network region supporting only unicast routing to receive multicast traffic from the native multicast infrastructure. The goal of AMT is to provide a migration path from no multicast support to full multicast support, and thus foster the deployment of native IP multicast. An Internet Service Provider can offer AMT-based service until such time as the number of multicast-capable customers justifies the expenditure for multicast-capable routers. Although AMT provides a simple-to-implement way to improve multicast availability, it provides no RAC for multicast groups.

In this paper, we have proposed a design architecture that provides RAC in AMT. We have also formally validated the security features of our model using the Automated Validation of Internet Security protocols and Applications (AVISPA) tool [5].

The rest of the paper is organized as follows: Section II gives background information on the PAC architecture, the Internet Group Management Protocol (IGMP), the Protocol Independent Multicast - Sparse Mode (PIM-SM) routing protocol, the Extensible Authentication Protocol (EAP), the Protocol for Carrying Authentication for Network Access (PANA), the Secure IGMP (SIGMP) protocol, the Group Security Association Management (GSAM) protocol, and AMT. Section III provides the problem definition. Section IV defines our proposed solution. Section V discusses some alternate approaches. Section VI shows how we have modeled our solution using the AVISPA formal modeling tool. Section VII concludes our paper.

II. Background

In this section, we first present the PAC architecture for native IP multicast that was developed within our group. This is followed by a brief description of the related protocols.

A. PAC Architecture

The architecture shown in Figure 1 was proposed in [6]. A number of parties that participate in a multicast session, either before the session or during it, have been identified. The *Content Provider* offers the product to be delivered to the multicast group. The *End User* (EU) receives the content. The *Network Service Provider* (NSP) delivers the data, making use of *Access Routers* (ARs), *Core Routers* (CRs), one or more instances of an *Authentication, Authorization and Accounting*

Server (AAAS), and a *Network Access Server* (NAS) associated with each AR. We will assume that the ability of the EU to pay for services will be certified by a *Financial Institution* (FI). The *Group Owner* (GO) is responsible for the creation and overall activities of the group. PAC can be further divided into SAC and RAC.

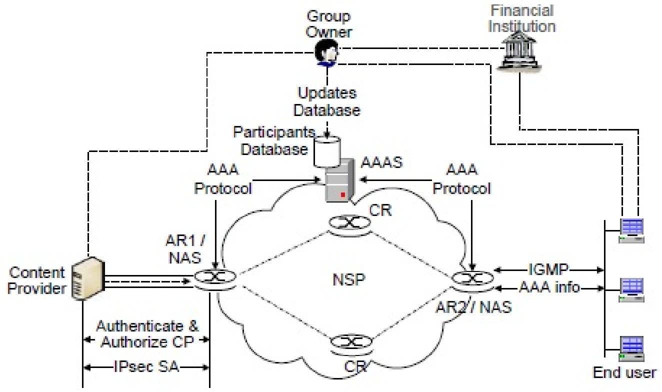

Fig. 1. Access Control Architecture for Multicast Participants.

SAC will be deployed at the interface between the CP and the network, where AR1 will authenticate and authorize the CP after an interaction with the AAAS. On successful authentication and authorization, an Internet Protocol Security (IPsec) *Security Association* (SA) [7] will be established between the CP and AR1 to cryptographically authenticate each data packet before forwarding it to the multicast distribution tree [2]. As AMT is targeted for extending the options for receivers, we assume that the Content Provider is in a multicast-capable region, and will not discuss SAC any further.

RAC will be implemented at the interface between the network and the EU's device. AR2 will receive and process the network level join (IGMP) messages (see Section II-B) and the messages carrying *Authentication, Authorization and Accounting* (AAA) information (see Section II-C and Section II-D). It will also act as a NAS by communicating with the AAAS. It is assumed that the Group Owner has supplied the user authentication information or AAA information to the AAAS when the EU purchased the service. Hence, each EU will be authenticated and authorized by the one-hop AR before allowing him/her to join a secured group [3], [4]. Several IPsec SAs will be established to cryptographically authenticate the Secure IGMP messages (see Section II-E) exchanged between the EU device and the network [8].

B. IGMP and PIM-SM

IGMP [9] 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. IGMP performs three main operations:

- A host sends a *join* message (through a Membership Report Message) when it wants to join a multicast group or some specific sources of a group.
- A host sends a *leave* message (through a Membership Report Message) when it wants to unsubscribe from a multicast group
- A router periodically checks (through a Membership Query Message) which multicast groups are of interest to the hosts that are directly connected to that router.

While IGMP is the protocol used between an EU host and its AR, a multicast routing protocol (typically PIM-SM [10]) is used to build the data distribution tree among the CRs and the ARs. An IGMP join message will cause the grafting of one or more new edges (if there are no existing clients on the same AR for that group) and an IGMP leave message will cause the data distribution tree to be pruned if this is the last client on the AR.

C. EAP

To achieve AAA functions, a AAA protocol (e.g., Diameter [11]) is used between a NAS and its associated AAAS. The specific aspects of (EU) authentication and authorization are typically delegated to EAP [12], which is a versatile framework that facilitates the use of multiple authentication methods, such as pre-shared secret, one time password, public key authentication, etc. Although EAP was originally intended to be used to control access to a network as a whole, it is also useful for managing access at the application layer. 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 [6]. EAP does not run directly over the IP layer. The mechanism for carrying the EAP packets will be discussed in Section II-D.

The EAP framework supports multiple authentication mechanisms called *methods*. EAP runs between an *Authenticator* (on the AR) and a *Peer* (on the EU host). The Authenticator normally acts as a pass-through to a back-end *Authentication Server* (AS), which will be co-located with the AAAS. The EAP packets that arrive at the NAS are sent to the AAAS by encapsulating them inside AAA packets, and the NAS will decapsulate the AAA packets received from the AAAS and forward the EAP packets to the Peer on the EU host.

A justification for using the method EAP-FAST in our application may be found in [13].

D. PANA

The IETF has standardized Protocol for carrying Authentication for Network Access (PANA) [14], a protocol that carries EAP authentication methods (encapsulated inside EAP packets) between a client node (EU host) and a server in the access network.

The PANA framework [15], comprised of four functional entities, is shown in Figure 2. The *PANA Client* (PaC) residing on a requesting node (e.g., an EU host) interacts with the *PANA Authentication Agent* (PAA) in the authentication process using the PANA protocol [14]. The server implementation of PANA is the PAA, which consults an *Authentication Server*

(AS) for authentication and authorization of a PaC. If the PAA is separate from the AS, a AAA protocol (e.g., Diameter) will be used for their communication. The PAA resides on a node that is typically a NAS in the access network. The AS is a conventional back-end AAAS that terminates the EAP and the EAP methods. The *PANA Enforcement Point* (EP) allows (blocks) data traffic of authorized (unauthorized) clients. When the PAA and EP reside on the same node, they use an API for communication, otherwise, a protocol (e.g., SNMP) is required. A secure association protocol (e.g., IKEv2 [16]) is required to run between the PaC and the EP to establish an IPsec [7] Security Association (SA) [17], which can provide integrity protection, data origin authentication, replay protection and optional confidentiality protection.

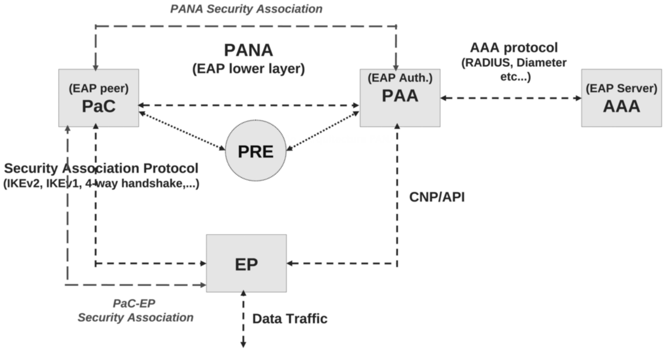

Fig. 2. PANA Framework.

When EAP-FAST or an equivalently-capable EAP method is used, a shared key becomes available to the PAA and the PaC. To establish an indirect coupling between the PANA/EAP-based authentication and IGMP join/leave operations, the shared key (or a key derived from that shared key) established during the PANA session can be used to protect IGMP messages (following the security guidelines of the IGMPv3 [9] specification).

E. SIGMP

The Secure Internet Group Management Protocol (SIGMP) [8] is an extension of IGMP, which runs among the EU hosts and the ARs. It distinguishes two types of multicast groups: *open* groups, which are identical to mulitcast groups with IGMP, and *secure* groups, for which the exchanges are protected. As for IGMP, in SIGMP the EU host implements the host portion of SIGMP while the AR implements the router portion of SIGMP. SIGMP queries and reports are each divided into two categories, Open Group Query (OGQ), Secure Group Query (SGQ), Open Group Report (OGR), Secure Group Report (SGR). OGQ and OGR are for open groups and SGQ and SGR are for secure groups. In SIGMP, queries and reports for open groups are delivered without any protection (i.e., exactly as they would be for IGMP), but for secure groups they are protected by IPsec Group Security Associations (GSAs). Two different GSA instances are used: GSA_q and GSA_r. GSA_q is used to protect the SGQ messages and GSA_r is used to protect the SGR messages.

F. GSAM

The Group Security Association Management (GSAM) protocol is used to manage the GSAs used in SIGMP (similar to IKEv2 in unicast). The network entities in GSAM are the same as those in SIGMP, including ARs and EU hosts. In IGMP (and SIGMP), if there are multiple routers on a network segment, one of them will be elected as the *Querier* (Q), and the remaining routers are called *Non-Queriers* (NQ). In GSAM, an AR (specifically, the Querier) plays the role of *Group Controller / Key Server* (GCKS). It accepts registrations from NQs and EUs that have been authorized at the application level and grants them group membership in the secure multicast groups that the EUs are authorized to join. The members of this set of EUs are called *Group Members* (GMs). The AR/Q creates and updates GSA_r and GSA_q for a secure group and distributes them to GMs in the secure group using secure tunnels. The Q, the NQs (if any), and the GMs will update their local IPsec databases (Security Association Database (SAD) and Group Security Policy Database (GSPD)) [7] according to the parameters of GSA_q and GSA_r to protect the SIGMP packets (for more details about SIGMP, GSAM and their interactions, see [8]).

G. RAC System Operation

The operation of RAC can be viewed at two levels: the application level and the network level.

At the application level, an EU will negotiate with the GO to obtain permission to access a particular product (e.g., a video stream). After consulting with the FI to determine the ability of the EU to pay, the GO will issue a “ticket” to the EU, which describes how and when to connect to the stream representing the product (i.e., it gives the network-level group address), and certifies the EU’s right to access the group. The *form* of this ticket is simultaneously (or has been previously) provided to the NSP, to permit rapid validation.

The ticket is presented to the NSP by the EU, using EAP [12]. PANA [15] is used to carry the EAP exchanges between the EU and the AR. In PANA, the PANA client (PaC) will be on the EU host. On the NSP side of the network segment, there are two PANA-related functions: the PAA and one or more EPs. The EPs are ultimately responsible for enforcing the restrictions in a network-level join. If an EU is not authorized, then a network-level join request from that EU’s host will be blocked, i.e., it will not result in a join operation in the multicast routing protocol. In the simple case (only one AR for the network segment), the PAA and the EP will be co-located with the AR. In more complex cases (more than one AR for a specific network segment), one AR will have both PAA and EP functions, and the rest will have only the EP function. An appropriate secure protocol is used to carry information from the PAA to the EPs. A AAA protocol (e.g., Diameter [11]) is used to carry the EAP exchanges between the AR and the

AAAS, where the ticket is validated. From the perspective of Diameter, the AR acts as a NAS, i.e., as a Diameter client.

Note that the election of Q for a network segment is independent of the designation of the PAA for that segment, so there is no pre-defined relationship among the PAA, the Q, the NQ (if any), and the NAS, although we do assume that the PAA resides on an AR that can act as a NAS.

As a result of the EAP exchanges, a PANA Master Session Key (MSK) becomes known to the PAA and the PaC. As defined in [18], the PAA must combine the MSK with EP-specific information to produce the PaC-EP Master Key (PEMK), which is then forwarded (securely) to the EP. As shown in [4], the EP must, in turn, combine this PEMK with group-specific information to produce the Multicast Session-Specific Key (MSSK), which will be used to protect the PaC-EP communications, and the (group-specific) network-level exchanges between an EU's host and its EP [8].

Note that since the MSK is specific to the multicast session, presentation of a ticket for a different session will result in the establishment of a new PANA session, a new MSK, and derivation of a new PEMK and a new MSSK.

The network-level join operation is requested through our secure extension to IGMP (see Section II-E). SIGMP is compatible with all existing versions of IGMP, and utilizes exactly the same packet formats.

The necessary security features are achieved using IPsec [7] and the Multicast Extensions to IPsec [19]. As noted above, the security parameters are derived from the MSK. The key management and coordination functions needed by SIGMP are provided by GSAM (see Section II-F).

H. Automatic Multicast Tunneling

Automatic Multicast Tunneling (AMT) [20] allows multicast communication to take place from one or more sources that have native multicast connectivity to hosts, sites or applications that do not have native multicast access, i.e., to request and receive Source Specific Multicast (SSM) or Any Source Multicast (ASM) traffic from within a network that does provide multicast connectivity. Without requiring any manual configuration, AMT allows the hosts to receive multicast traffic from the native multicast infrastructure. AMT operates with a pseudo interface, where UDP-based encapsulation is done to overcome problems of multicast connectivity.

We assume that the multicast-enabled ISP provides the *AMT Relay* service. As shown in Figure 3, the hosts connected to the unicast-only network are acting as *AMT Gateways*.

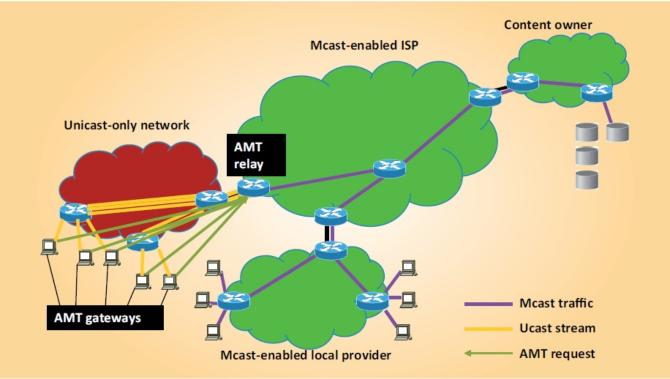

Fig. 3. AMT Architecture.

1) When host wants to join a multicast group, it sends a membership report to the Gateway thinking that it is an IGMP router (Querier).
2) Before forwarding the received report, the Gateway will send a Request message to the Relay to solicit a General Query response. The Relay responds by sending a Membership Query message back to the gateway. The Membership Query message carries an encapsulated general query that is processed by the IGMP or MLD protocol implementation on the Gateway to produce a membership/listener report. Each time the Gateway receives a Membership Query message, it starts a timer whose expiration will trigger the start of a new Request. This timer-driven sequence is used to mimic the transmission of a periodic General Query by an IGMP/MLD router. This query cycle may continue indefinitely, once started by sending the initial Request message.
3) After receiving the general query from the Relay, the Gateway will send the membership report encapsulated to the Relay. Each report is encapsulated and sent to the Relay after the Gateway has successfully established communication with the Relay via a Request and Membership Query message exchange.
4) The AMT Relay will decapsulate the IGMP messages and trigger an upstream PIM join towards the source.
5) Finally the requested multicast data are transferred from the multicast source to host through the Relay and the Gateway.

AS noted in Section I, AMT is intended as an interim measure [20]. Its purpose is to provide a (relatively) low-cost mechanism that will allow the set of multicast subscribers to grow gracefully, until the point is reached where full multicast routing support can be justified. As such, considerations of efficiency and scalability are not key issues in the design of AMT. (Any tunneling-based solution will always be less efficient than a solution that does not involve tunnels.) Similarly, scalability is not a key issue, because once the subscriber base becomes large enough for scalability to be an issue, the justification for full multicast routing support will be there.

III. Problem Definition

As previously noted, native IP multicast offers scalable point-to-multipoint delivery, but no access control. AMT extends IP multicast service to a unicast-only region, but offers no access control. The PAC environment offers access control, but is limited to the native IP multicast environment. So, our goal is to combine both, i.e., in addition to the current features of AMT, we must add RAC features. This must be done without changing the interactions that are expected by the EU or the network components that reside in the native IP multicast region.

IV. Proposed Solution

As noted in Section II-H, the AMT Relay and the AMT Gateway implement the host and router portions of the IGMP interaction, respectively. Our design is based on extending the functionality of the AMT Relay and the AMT Gateway so that the EAP, PANA, SIGMP, and GSAM interactions in the AMT environment are identical to what they would be in the native IP multicast environment. Here, in this section, we explain how the RAC framework is accommodated into the AMT environment to achieve Receiver Access Control. Figure 4 shows the AMT architecture with “Receiver Access Control”. The whole design is based on the fact that all messages and data between the EU host and the AR must pass through the AMT Tunnel, i.e., between the Gateway and the Relay.

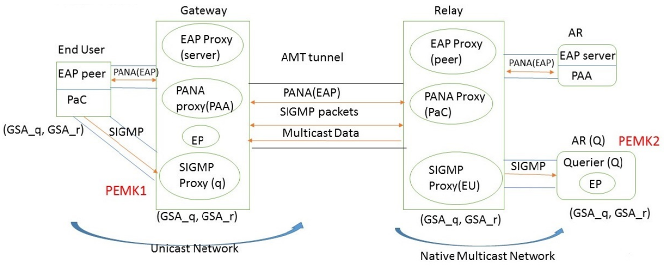

Fig. 4. Receiver Access Control in AMT.

A. EAP, PANA, SIGMP, GSAM Proxies

The need for all exchanges between the EU host (in the unicast-only region) and the AR (in the multicast-enabled region) to flow through the AMT tunnel implies that proxies must be established for all messages associated with the RAC functionality. We introduce a proxy in the Gateway for each message type; each proxy responds as if it were the AR. We introduce a corresponding proxy in the Relay, each acting as if it were on an EU host. The necessary interactions among the EAP proxy, the PANA proxy, the SIGMP proxy, and the GSAM proxy are simplified because they are all co-located in the Gateway.

Triggered by the need to send the first EAP message, the (real) PaC discovers its proxy PAA in the Gateway using the normal mechanism for PAA discovery as defined in [21], and establishes a secure relationship with it. The proxy PaC in the Relay discovers the real PAA and establishes a secure relationship with it. In effect, the Gateway and the Relay act as a “friendly” Man-in-the-Middle.

After authentication, the EAP method exports a Master Session Key (MSK) to the PaC and the PAA. As a result the EAP proxy parts in the Gateway and the Relay will know (or be able to construct) the MSK for protecting SIGMP messages.

SIGMP on the EU host interacts with the proxy SIGMP on the Gateway. It will use a GSA derived from the MSK in the same way as it would if it were in a native IP multicast environment. Similarly, the proxy SIGMP on the Relay interacts with the Querier in the native IP multicast region.

GSAM on the EU host uses the keys derived from the MSK and the proxy GSAM identity to form the necessary GSAs to protect the SIGMP exchanges between the EU host and the Gateway, and proxy GSAM on the Relay uses the keys derived from the MSK and the Querier identity to form the necessary GSAs to protect the SIGMP exchanges between the Relay and the Querier. Although the MSK has the same value, the EP identity is different in the two cases, so the derived keys will differ.

As a result, communication between the EU host and the multicast-network-based components will take place on three segments: EU host to Gateway, Gateway to Relay, and Relay to AR.

B. RAC in AMT System Operation

From the perspective of the EU, the operations proceed exactly as they would in a native IP multicast environment. However, our desired proof of security must take into consideration the existence of additional participants in these exchanges. When a SIGMP message is to be sent by the EU host for the first time, GSAM is invoked to negotiate the cryptographic parameters. This negotiation will be between the EU host and the Gateway. It will in turn trigger (through the AMT tunnel) another negotiation between the Relay and the AR in the multicast-enabled region. Further details may be seen in [22].

The RAC can be viewed at two levels: the application level and the network level.

1) Access Control at the Application Level: A PANA session consists of five phases [14]. We explain below how the PANA messages are exchanged during these phases in AMT using the PANA proxy and the EAP proxy.

1) Handshake Phase: The PaC, on receiving a request from the upper layer to join a multicast group, initiates a PANA session by sending a PANA Client Initiation (PCI) message to the Gateway thinking it is the PAA. The Gateway finds it as a PANA packet and forwards it to the Relay. The Relay, having a PANA proxy acting as a PaC, forwards the packet to the actual PAA. The response goes back from actual PAA to PaC through the Relay and the Gateway.
2) Authentication and authorization phase: After the handshake phase, EAP packets carried by PANA will be exchanged between the PaC and the PAA. For better understanding, we took an example of EAP-FAST method [23], an efficient EAP method. This method has two phases, in which phase 1 is responsible for TLS handshake resulting in a secure tunnel between peer

and server. As explained, the EAP proxy acting as an EAP server is in the Gateway and the EAP peer is in the EU. The secure tunnel is formed between the EU and the Gateway (say STunnel1), resulting in a fresh secret key between them. The same secure tunnel with another key is formed between the Relay and the PAA (say STunnel2) during phase 1. In phase 2, EAP method payloads carrying user credentials in PANA packets are transferred to the Gateway through STunnel1 and the Gateway, who shares the secret key with the EU during phase1, will decrypt and forward them to the Relay through the AMT Tunnel (assuming AMT tunnel is secured). Finally the Relay protects the payloads with keys obtained during formation of STunnel2 and forwards the EAP message to the PAA. The PAA verifies those credentials and authenticates EU and sends the results back.

After a successful authentication, the PaC and PAA derive a Master Session Key (MSK). As the Gateway and the Relay are part of PANA exchanges and acting as a friendly Man-in-the-Middle, they can compute the MSK as well. On receiving the MSK the PAA transfers MSK to EPQ (Enforcement point in Querier) using IPsec, with a key calculated in the normal way for two IPsec peers [24].

3) Access Phase: PaC and EPG (Enforcement point in Gateway), Relay and EPQ with acquired pre-shared key (MSK) during authentication phase calculate the secret key called PEMK, respectively. As the EPs are on different devices they end up calculating different PEMKs, i.e., PEMK1 between the PaC and the Gateway, PEMK2 between the Relay and the actual Querier. One way of calculating this key [18] is:

```
PEMK = prf+(MSK, "IETF PEMK"|
            SID|KID|EPID)
```

Here, prf+ is a pseudo-random function defined in [16]. “IETF PEMK” is the ASCII code representation, SID is a four-octet Session Identifier, KID is associated with the MSK and EPID is the identifier of the EP. This key is specific to the multicast group that the EU has joined at the application level, and will be used for authorization at the network layer.

With those PEMKs, they establish a two different IPsec GSAs between them for cryptographic protection of IGMP messages. Each IPsec GSA contains one GSA_r and one GSA_q (for details see Section IV-B2). This phase is also used to test liveness of the PANA session.

4) Re-authentication and Termination phases are similar to that described in [14], except the fact that these PANA messages are exchanged through the AMT Tunnel.

2) Access Control at Network Level: In SIGMP [8], some messages are protected by IPsec GSAs. In this protocol all the operations for OGQ (Open Group Query) and OGR (Open Group Report) are retained from IGMPv3. However, for the access control of secure groups, a few operations are added in it. The material below describes how SIGMP is fitted into AMT.

- EU Operations: Once the Authentication is done at the application level, the EU will make his/her request for the network-level join and will send an SIGMP report message, believing it is being sent to the real Q. (In fact, it will be received by the SIGMP proxy in the Gateway). If this is the first time, when the report is sent to the IPsec (GSA) module, GSAM will be invoked to negotiate the cryptographic parameters (keys and SPIs) (see bullet 3, below). The IPsec module will then be able to send the report protected by those secure parameters to the Gateway where the SIGMP proxy (q) is implemented. The q in the Gateway will forward the message to the Relay through a secured tunnel (assumed) and finally the Relay will forward it to the actual Q that accepts the request.
- Q Operations: On receiving a secured report, Q will invoke IPsec module to decrypt it.
- GSAM: Group Security Association Management Protocol (GSAM) manages IPsec GSAs in two phases. In phase1, mutual authentication of EU and Q is done to achieve the registration of an EU. In phase 2, Q creates and distributes a GSA pair (GSA_q, GSA_r), named GSAM-TEK-SA to protect SIGMP messages (for details see [8]). Usually, in an IP multicast environment, GSAM negotiations are done between the EU and the real Q, but in AMT we must not let the EU communicate directly with the actual Q. As explained earlier, we implement an SIGMP proxy, which acts as querier functionality (q) in the Gateway, so that EU starts mutual authentication with the Gateway (q) using the derived PANA secret key, i.e., PEMK1. After authentication is done the Gateway (q) creates and distributes GSAM-TEK-SA (SA pair) to EU. On the other side of AMT tunnel SIGMP proxy acting as EU in the Relay performs mutual authentication with the actual Querier (Q) using PEMK2 and receives a GSA pair from Q.

V. Alternate Solutions

To our knowledge, the only other solution to providing access control for multicast services is based on having access control lists, either in the Set Top Box (STB) adjacent to the customer equipment, or in the access router. These solutions assume that the ISP has strong control over the STB or the access router. Our solution makes it possible for control to be exercised within the software of the Gateway. The existence of the “ticket”, and the keys derived from the information in the ticket, ensure that the GO retains control of the session, in spite of the fact that the Gateway software would be freely available for downloading by the subscribers.

VI. AVISPA

The Automated Validation of Internet Security Protocols and Applications (AVISPA) project [5] has built a suite of tools that provides a modular and expressive formal language (High Level Protocol Specification Language, HLPSL) for specifying protocols and their security properties, and integrates different back-ends that implement a variety of automatic protocol analysis techniques. Experimental results, carried out on a large library of Internet security protocols, indicate that the AVISPA Tool is a state-of-the-art tool for Internet security protocol analysis as, to our knowledge, no other tool exhibits the same level of scope and robustness while enjoying the same performance and scalability [5]. In this section, we describe how we have transformed our model into HLPSL code, and how we have formulated the security goals to achieve the desired validation of the protocol.

- Our model in HLPSL code has four basic roles. They are client, server, gateway, relay. (Roles in AVISPA begin with a lower-case letter.) The roles client and server serve as PaC and PAA, respectively. As per our model, gateway and relay are acting as a friendly Man-in-the-Middle; they form SAs with client and server, respectively, and forward the EAP/PANA messages accordingly. The roles of the gateway and the relay are important because attacks are possible on both the gateway and the relay. So, we consider all four roles as main actors in HLPSL.
- In the real world, there is a large number of clients asking for a specific multicast application and they may request different multicast data streams as well. So, there is a need to distinguish all these clients and their requests. For that reason, we have added constants such as request-id and response-id, which assign a random unique number for each request made by clients. We transferred these constants along with nonces of client and server in initial request messages.
- After a few initial messages, PANA starts carrying EAP method (EAP-FAST) for authentication. EAP-FAST is already validated between two nodes in [13]. Now we implement it among four nodes in our HLPSL code. As phase 1 in EAP-FAST results in a shared key (SA) between two nodes, to make it simpler we introduced a shared key K1 between client, gateway and K2 between relay and server. Client and gateway protect further data with K1 and relay and server with K2.
- After authentication all the four roles are able to calculate a secret key (MSK). Using MSK and PANA nonces, they calculate MAC (Message Authentication Code) value as well. Our goal is to maintain the secrecy of secret keys MSK, K1, K2. Derivation of secret key (MSK) and MAC is shown in Figure 5 below.
- After calculation of above mentioned keys, the results are passed to client from server.
- The session role defines executing of several basic roles in parallel. In our HLPSL code, the session role is composed of client, gateway, relay and server roles. Every role has two channels, send and receive, on which they send and receive messages. We should run these four roles in parallel for messages to pass through the AMT tunnel (see Figure 6).

```
% Calculation of Master Session Key.
Msk’ := H(Nec’.Nes.Psk’)

% Calculation of Message Authentication Code
Mac’ := INT(PRF(H(Nec’.Nes.Psk’).Nps.Npc.
Kid).Pmsg)
```

Fig. 5. Secret Key and Message Authentication Code.

```
role session(
C,G,R,S :agent
K1,K2 :symmetric
H,PRF,INT :hash_func)
def=
local SC,RC,SG,RG,SR,RR,SS,RS :channel (dy)
composition
client (C,G,R,S,K1,H,PRF,INT,SC,RC)
/\ gateway(C,G,R,S,K1,H,PRF,INT,SG,RG)
/\ relay (C,G,R,S,K2,H,PRF,INT,SR,RR)
/\ server (C,G,R,S,K2,H,PRF,SS,RS)
end role
```

Fig. 6. Session Role.

- In the environment role, we can modify the number of parallel sessions and the knowledge of intruder. In our code, the intruder has been given the knowledge of all the hash functions, agents and his own private key. First, we executed a session without any intruder. In the next step, we executed session with client as intruder and then gateway, relay, server as intruders (see Figure 7).

```
role environment()
def=
const C,G,R,S :agent,
KK1,KK2 :protocol_id,
K1,K2,Ki :symmetric_key, H,PRF,INT
:hash_func
intruder_knowledge = {c,g,r,s,h,prf,int,ki}

composition
session(c,g,r,s,k1,k2,h,prf,int)
/\ session(i,g,r,s,ki,k2,h,prf,int)
/\ session(c,i,r,s,ki,k2,h,prf,int)
/\ session(c,g,i,s,k1,ki,h,prf,int)
/\ session(c,g,r,i,k1,ki,h,prf,int)
end role
```

Fig. 7. Environment Role.

- In the goal section of our HLPSL code, we explicitly ask the AVISPA model checker to validate the secrecy of both the shared secret keys (K1, K2) and MSK, which ensures the intended security of further communications. Security goals are shown in Figure 8.
- Considering the security goals mentioned in the goal section of our HLPSL code, no attack has been found.

```
goal
%Secrecy of Shared Key between Client
% and Gateway
secrecy_of kk1

%Secrecy of Shared Key between Relay
% and Server
secrecy_of kk2

%Secrecy of Master Session Key)
secrecy_of s_msk
end goal
```

Fig. 8. Goals.

Summary results of three AVISPA back-ends OFMC, CL-AtSe and SATMC appeared to be safe. This shows our model (Receiver Access Control in AMT) in reality is immune to all those potential attacks and threats.

VII. Conclusion

In this paper, we have proposed a solution that provides receiver access control for multicast groups in the AMT environment. This solution allows only legitimate End Users in a unicast-only-network to access networks and receive multicast data from multicast enabled sources. We have used AVISPA to formally demonstrate the security of these extensions to AMT.

Acknowledgment

J. W. Atwood acknowledges the support of the Natural Sciences and Engineering Research Council of Canada, through its Discovery Grants program.

References

[1] J. W. Atwood, "An architecture for secure and accountable multicasting," in *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, Oct. 2007, pp. 73–78.

[2] S. Islam and J. W. Atwood, "Sender access and data distribution control for inter-domain multicast groups," *Computer Networks*, vol. 54, no. 10, pp. 1646–1671, 2010.

[3] ——, "Multicast receiver access control by igmp-ac," *Computer Networks*, vol. 53, no. 7, pp. 989–1013, 2009.

[4] ——, "Multicast receiver access control using pana," in *Proceedings of the 1st Taibah University International Conference on Computing and Information Technology (ICCTT 2012)*, Mar. 2012.

[5] L. Viganò, "Automated security protocol analysis with the AVISPA tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, no. 10, pp. 61–86, 2006.

[6] S. Islam, "Participant access control in ip multicasting," Ph.D. dissertation, Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada, Mar. 2012.

[7] S. Kent and K. Seo, "Security architecture for the internet protocol," Internet Engineering Task Force, Request for Comments 4301, Dec. 2005, URL: http://www.rfc-editor.org/rfc/rfc4301.txt [accessed: 2015-02-15].

[8] B. Li and J. W. Atwood, "Receiver access control for IP multicast at the network level," *Submitted to Computer Networks*, Sep. 2014.

[9] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet group management protocol, version 3," Internet Engineering Task Force (IETF), Request for Comments 3376, Oct. 2002, URL: http://www.rfc-editor.org/rfc/rfc3376.txt [accessed: 2015-02-15].

[10] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol independent multicast - sparse mode (pim-sm): Protocol specification (revised)," Internet Engineering Task Force (IETF), Request for Comments 4601, Aug. 2006, URL: http://www.rfc-editor.org/rfc/rfc4601.txt [accessed: 2015-02-15].

[11] V. Fajardo, J. Arkko, J. Loughney, and G. Zorn, "Diameter base protocol," Internet Engineering Task Force, Request for Comments 6733, Oct. 2012, URL: http://www.rfc-editor.org/rfc/rfc6733.txt [accessed: 2015-02-15].

[12] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible authentication protocol (eap)," Internet Engineering Task Force, Request for Comments 3748, Jun. 2004, URL: http://www.rfc-editor.org/rfc/rfc3748.txt [accessed: 2015-02-15].

[13] M. Parham, "Validation of the security of participant control exchanges in secure multicast content delivery," Master's thesis, Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada, Sep. 2011.

[14] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin, "Protocol for carrying authentication for network access (pana)," Internet Engineering Task Force, Request for Comments 5191, May 2008, URL: http://www.rfc-editor.org/rfc/rfc5191.txt [accessed: 2015-02-15].

[15] P. Jayaraman, R. Lopez, Y. Ohba, M. Parthasarathy, and A. Yegin, "Protocol for carrying authentication for network access (pana) framework," Internet Engineering Task Force, Request for Comments 5193, May 2008, URL: http://www.rfc-editor.org/rfc/rfc5193.txt [accessed: 2015-02-15].

[16] Y. Nir, C. Kaufman, P. Hoffman, and P. Eronen, "Internet key exchange (ikev2) protocol," Internet Engineering Task Force, Request for Comments 5996, Sep. 2010, URL: http://www.rfc-editor.org/rfc/rfc5996.txt [accessed: 2015-02-15].

[17] M. Parthasarathy, "Pana enabling ipsec based access control," Internet Engineering Task Force, Internet Draft, Work in progress, Dec. 2005, URL: http://tools.ietf.org/id/draft-ietf-pana-ipsec-07.txt [accessed: 2015-02-15].

[18] Y. Ohba and A. Yegin, "Definition of master key between pana client and enforcement point," Internet Engineering Task Force, Request for Comments 5807, Mar. 2010, URL: http://www.rfc-editor.org/rfc/rfc5807.txt [accessed: 2015-02-15].

[19] B. Weis, G. Gross, and D. Ignjatic, "Multicast extensions to the security architecture for the internet protocol," Internet Engineering Task Force, Request for Comments 5374, Nov. 2008, URL: http://www.rfc-editor.org/rfc/rfc5374.txt [accessed: 2015-02-15].

[20] G. Bumgardner, "Automatic multicast tunneling," Internet Engineering Task Force, Request for Comments 7450, Feb. 2015, URL: http://www.rfc-editor.org/rfc/rfc7450.txt [accessed: 2015-03-02].

[21] L. Morand, A. Yegin, S. Kumar, and S. Madanapalli, "Dhcp options for protocol for carrying authentication for network access (pana) authentication agents," Internet Engineering Task Force (IETF), Request for Comments 5192, May 2008, URL: http://www.rfc-editor.org/rfc/rfc5192.txt [accessed: 2015-02-15].

[22] V. N. T. Malla, "Design and validation of receiver access control in the automatic multicast tunneling environment," Master's thesis, Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada, Aug. 2014.

[23] N. Cam-Winget, D. McGrew, J. Salowey, and H. Zhou, "The flexible authentication via secure tunneling extensible authentication protocol method (eap-fast)," Internet Engineering Task Force (IETF), Request for Comments 4851, May 2007, URL: http://www.rfc-editor.org/rfc/rfc4851.txt [accessed: 2015-02-15].

[24] A. Yegin, Y. Ohba, R. Penno, and C. Wang, "Protocol for carrying authentication for network access (pana) requirements," Internet Engineering Task Force, Request for Comments 4058, May 2005, URL: http://www.rfc-editor.org/rfc/rfc4058.txt [accessed: 2015-02-15].