DESIGN AND VALIDATION OF A SECURED TUNNEL IN THE AUTOMATIC MULTICAST TUNNELING (AMT) ENVIRONMENT

Abonti Ferdous

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

November 2015

© Abonti Ferdous, 2015

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Abonti Ferdous**

Entitled: **DESIGN AND VALIDATION OF A SECURED TUNNEL IN THE AUTOMATIC MULTICAST TUNNELING (AMT) ENVIRONMENT**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

_____ Chair

Dr. Nikolaos Tsantalis

_____ Examiner

Dr. Hovhannes A. Harutyunyan

_____ Examiner

Dr. Brigitte Jaumard

_____ Supervisor

Dr. J.W. Atwood

Approved by _____

Dr. V. Haarslev

Graduate Program Director

_____ 2015. _____

Dr. Amir Asif, Dean

Faculty of Engineering and Computer Science

Abstract

DESIGN AND VALIDATION OF A SECURED TUNNEL IN THE AUTOMATIC MULTICAST TUNNELING (AMT) ENVIRONMENT

Abonti Ferdous

IP multicasting is a communication mechanism in which data are communicated from a server to a set of clients who are interested in receiving those data. Any client can dynamically enter or leave the communication. The main problem of this system is that every client that is interested in receiving the multicast data has to be in a multicast enabled network. The Network Working Group at the Internet Engineering Task Force (IETF) has come up with a solution to this problem. They have developed a protocol named Automatic Multicast Tunneling (AMT). This protocol offers a mechanism to enable the unicast-only clients to join and receive multicast data from a multicast enabled region through an AMT tunnel, which is formed between the two intermediate participants named Gateway and Relay. However, AMT does not provide any Participant Access Control (PAC).

Malla has designed an architecture for adding PAC at the receiver's end in the AMT environment. His work is based on the assumption that the AMT tunnel is secure and the tunnel can recognize and pass the additional message types that his design requires. We have designed the solution to secure the AMT tunnel. We also defined the additional message types. Lastly, we validated our work using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to ensure that our design is secure.

Acknowledgments

I would like to thank my supervisor **Dr. J. William Atwood** from the bottom of my heart for his encouragement, supervision and support from the very beginning of this research work. His guidance helped me to develop a good understanding of the subject.

I would also like to thank my parents — **Mr. Kazi Abul Manjur** and **Mrs. Ferdous Ara Shanta** and my family, without whom, I would not be here where I am today. Finally, I would like to thank my husband **Md Tariqul Amin** whose constant support made this master's degree possible.

Contents

List of Figures

List of Tables

List of Acronyms

AAA	Authentication, Authorization and Accounting
AAAS	Authentication, Authorization and Accounting Server
AMT	Automatic Multicast Tunneling
AR	Access router
AVISPA	Automated Validation of Internet Security Protocols and Applications
CA	Certificate Authority
CP	Content Provider
CR	Core Router
CS	Content Server
DDoS	Distributed Denial of Service
DoS	Denial of Service
EAP	Extensible Authentication Protocol
EAP-FAST	Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling
EP	Enforcement Point
EU	End User
EUD	End User Device
FI	Financial Institution
GSA	Group Security Association
GSAM	Group Security Association Management
GSPD	Group Security Policy Database
HLPSL	High Level Protocol Specification Language

IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGMPv2	Internet Group Management Protocol version 2
IGMPv3	Internet Group Management Protocol version 3
IKE	Internet Key Exchange protocol
IKEv2	Internet Key Exchange protocol version 2
IPsec	Internet Protocol Security
MAC	Message Authentication Code
MLD	Multicast Listener Discovery
MLDv1	Multicast Listener Discovery version 1
MLDv2	Multicast Listener Discovery version 2
MSK	Master Session Key
MR	Merchant
NAS	Network Access Server
NQ	Non-Querier
NSP	Network Service Provider
PAA	PANA Authentication Agent
PAC	Participant Access Control
PaC	PANA Client
PAD	Peer Authentication Database
PANA	Protocol for Carrying Authentication for Network Access
Q	Querier

QQIC	Querier's Query Interval Code
RAC	Receiver Access Control
SA	Security Association
SAC	Sender Access Control
SAD	Security Association Database
SPD	Security Policy Database
SPI	Security Parameter Index
SIGMP	Secure Internet Group Management Protocol
UDP	User Datagram Protocol

Chapter 1

Introduction

Today, with the rise of the Internet, efficient and intelligent use of bandwidth is becoming more and more important. Applications that need multiple participants are becoming more popular day by day. Video-conferencing, on-line games, Internet Protocol TV (IPTV) etc., are very common example of such applications. These applications depend on one-to-many or many-to-many communications. One or multiple sources communicate to multiple destinations [30]. The technology through which this situation can conserve a large amount of bandwidth is IP multicast. It eliminates unnecessary packet replications in the network. IP multicast saves bandwidth by forcing the network to do packet replication only when necessary. A source sends a packet only once for a group of destinations, while intermediate routers forward the packets to only those networks and hosts that need to receive them.

However, the main problem of IP multicast is every link on the network, every router and firewall between source and receiver, requires multicast protocols to be enabled. The users who are in a unicast only network cannot receive multicast data. Due to some unfortunate reasons, IP multicast is not deployed largely. So the applications that could benefit from IP multicast were unable to do that. The Internet Engineering Task Force (IETF) proposed a new protocol named Automatic Multicast Tunneling (AMT) [12] to overcome this problem. This protocol enables the devices that are in a unicast only network to participate in receiving multicast traffic even in the absence of end-to-end multicast connectivity. Without the need for any explicit tunnel, AMT provides a migration path between the source and the destination. AMT requires

two intermediate components, one on the unicast only network called gateway and the other on the multicast only network named relay. In AMT, multicast queries and reports are encapsulated in User Datagram Protocol (UDP) packets by the gateway and sent to a relay, which then transmits them natively toward the source. Gateways and relays exchange multicast control messages by creating a dynamic tunnel. The main objective of AMT is to substitute the deployment of native IP multicast. AMT could make multicast available on an Internet-wide basis. If vendors implement AMT in their devices then many hosts will be able to reach a much wider variety of multicast content in parallel to the existing unicast-only content. However, AMT has some disadvantages too. It does not provide any Participant Access Control (PAC), i.e., control over which participants can send multicast data, and which participants can receive it.

Malla [38] proposed an architecture to provide Receiver Access Control (RAC) in AMT using Extensible Authentication Protocol (EAP), the Protocol for Carrying Authentication for Network Access (PANA) and the Secure Internet Group Management Protocol (SIGMP). Details about these protocols can be found in chapter 4. In this design, the author adapted the RAC architecture of IP multicast [31] and incorporated that architecture into the AMT environment. This design assumes that the AMT tunnel between the gateway and the relay is secured. Our work is to secure the AMT tunnel. In our solution we have incorporated Internet Protocol Security (IPsec) [35] between gateway and relay to secure the tunnel. We also validated our work using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool.

The organization of the thesis is as follows:

- Chapter 2 gives a brief description of IP multicast and its benefits and challenges. It also talks about the access control architecture in IP multicast.
- Chapter 3 describes AMT, its components, 3-way handshake, advantages and security considerations.
- Chapter 4 discusses the architecture for RAC in AMT and underlying protocols of this architecture. In this chapter, we also introduce the additional message

types in AMT.

- Chapter 5 states the problem classification.
- Chapter 6 depicts the proposed solution.
- Chapter 7 gives information about the AVISPA tools and how we validate our solution.
- Chapter 8 summarizes and concludes the thesis with some possible future work.

Chapter 2

Background

2.1 IP Multicast

2.1.1 Definition

IP multicast is a bandwidth-conserving technology that reduces traffic by simultaneously delivering a single stream of information to potentially thousands of corporate recipients and homes. IP multicast delivers application source traffic to multiple receivers without burdening the source or the receivers while using a minimum of network bandwidth. Multicast packets are replicated in the network at the point where paths diverge by routers enabled with multicast protocols, resulting in the most efficient delivery of data to multiple receivers [28].

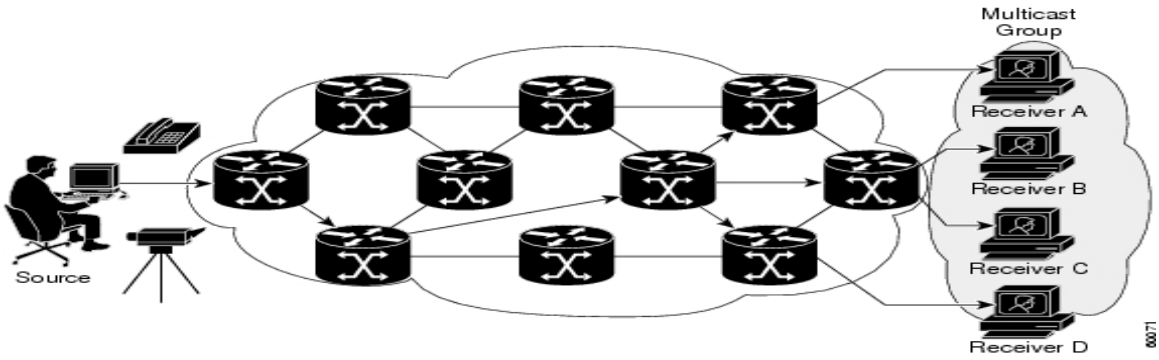

Figure 1: Multicast Transmission to Many Receivers [28].

Figure 1 shows how IP multicast is used to deliver data from one source to many interested recipients.

2.1.2 Benefits of IP Multicast

Any form of network communication involving the transmission of information to multiple recipients can benefit from the bandwidth efficiency of multicast technology. Examples of applications involving one-to-many or many-to-many communications include: video and audio broadcasts, video conferencing/collaboration, database replication, software downloads, and website caching [27].

To understand the efficiency of multicasting, consider a video server offering a single channel of content. For full-motion, full-screen viewing, assume that a sender, S, wants to send a message to receivers R1 and R2, as shown in Figure 2 . In case of unicast transmission, S should transmit the same data twice and the bandwidth usage between the sender and the intermediate node is doubled. In broadcasting, other receivers such as R3 will get the packets although they are not relevant to R3, causing unnecessary bandwidth consumption. But in multicasting, only a single copy of the message is transmitted from the sender and it is copied at the intermediate node to be sent to the multicast group. A multicast group can range in size from a few nodes to several thousands [43].

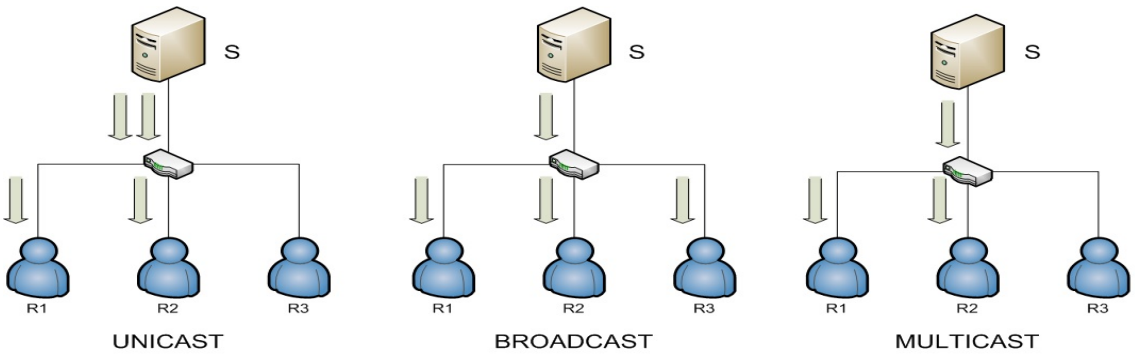

Figure 2: Benefits of IP Multicast [43].

Obviously, where there is large number of recipients of a replicated transmission, multicast technology makes a tremendous difference in both server load and network load, even in a simple network with a small number of router and switch hops [27].

2.1.3 Challenges with IP Multicast

1. End users who do not have native multicast connectivity cannot get multicast data. End users who belong to a multicast enabled network can receive multicast data. Figure 3 demonstrate the situation. Only hosts with multicast connectivity can join the multicast group and receive multicast data. However, hosts with unicast only network connectivity can neither join nor receive multicast data [38].

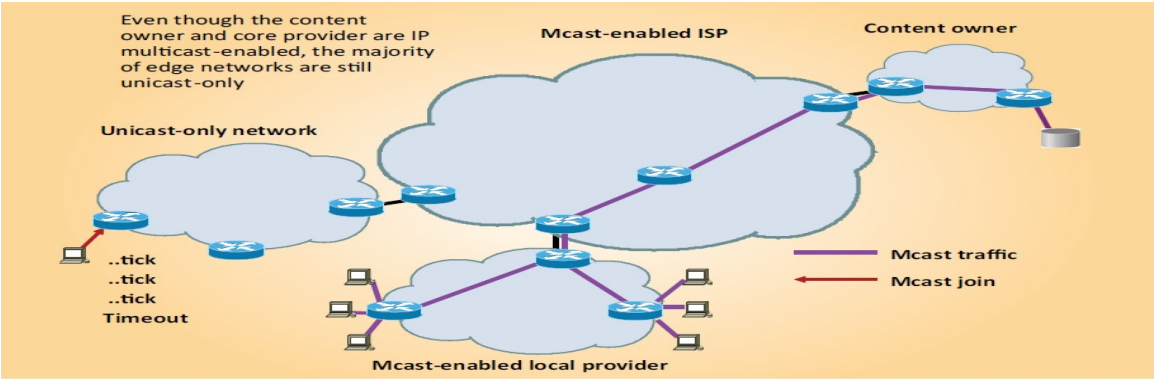

Figure 3: Challenges with IP Multicast [36].

2. Network Service Provider (NSP)s want a secure way to monitor the accountability of each end user. There is no easy way to make multicast into a commercial service [36].

2.2 Access Control in IP Multicast

This section illustrates a reference multicast architecture that has both sender and receiver access control [9]. Figure 4 depicts the participating components of this architecture.

2.2.1 Components

A brief description of the components is as follows:

- **Content Provider (CP):** An organization that offers the multicast data to be delivered.

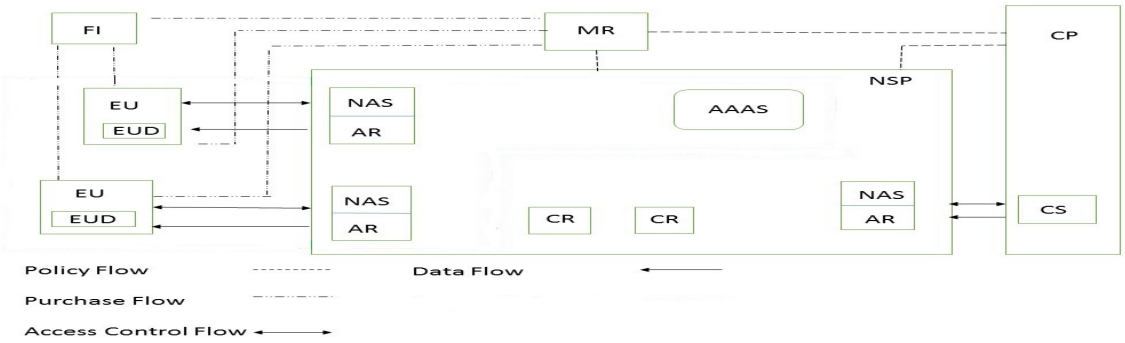

Figure 4: Reference Multicast Architecture [9].

- **Content Server (CS)**: A device that is used by the CP to distribute the content.
- **End User (EU):** An entity who wishes to receive multicast data.
- **End User Device (EUD):** A device that is operated by an EU to receive multicast data.
- **Network Service Provider (NSP):** An organization that delivers content from a CS to an EUD.
- **Merchant (MR):** Merchant is the single point of contact for the EU(s) to ask for one or more services. It advertises contents from multiple CP(s) and acts as a middleman between CP(s) and EU(s).
- **Financial Institution (FI):** Financial Institution is responsible for certifying an EUŠs capability to pay for the service it has ordered. It has secure and trusted connections with MR through which FI conveys the certificate to MR.
- **Authentication, Authorization and Accounting Server (AAAS):** A server for managing authentication, authorization and accounting within the NSP.
- **Access router (AR):**A routing device within the NSP, close to the EUD, which is responsible for deciding access rights to the network.

- **Network Access Server (NAS):**The enforcement function for managing authentication, authorization and accounting within the NSP. Normally co-located with the AR.
- **Core Router (CR):**A routing device within the NSP that does not have any EUD connected to it directly.

2.2.2 Information Flow

As shown in Figure 4, information flow between the components can be categorized into four category.

- **Policy Flow:** Exchange of policy information.
- **Purchase Flow:** The transactions related to subscribing to and paying for a group session.
- **Access Control Flow:**The presentation of authentication and authorization information. There are two categories: RAC flow, which is between the NSP and receivers (e.g., EUs) and Sender Access Control (SAC) flow, which is between the NSP and senders (e.g., CP).
- **Data Flow:** The delivery of the subscribed data stream, from the CS through the NSP to the EUD.

Chapter 3

Automatic Multicast Tunnelling (AMT)

3.1 Introduction

AMT provides a way for multicast data to travel from a multicast enabled network to a unicast only network without configuring any explicit tunnel between the source and the destination. AMT uses UDP based encapsulation to perform dynamic tunneling. This method allows the consumers to request and receive multicast traffic even when there is no end to end multicast connectivity [36].

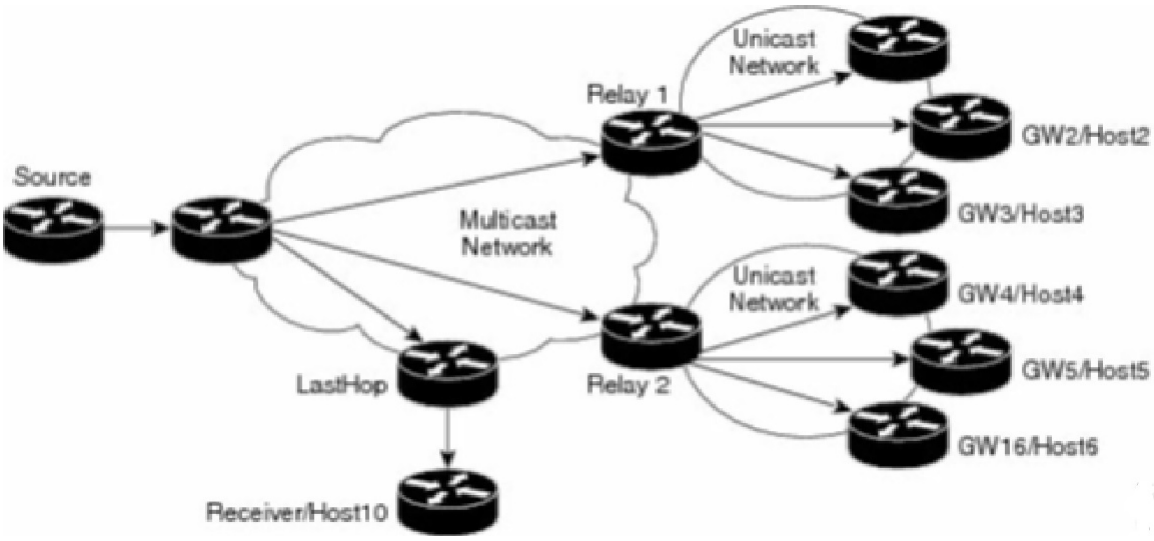

Figure 5: AMT Communication [29].

3.2 Components

Figure 5 portrays the overall scenario of an AMT communication. Relay 1 and Relay 2 are two relays that encapsulate multicast traffic to the tunnels and send a copy to each of the AMT gateways. The role of each component is described below.

- **AMT Relay:** AMT relay serves as a multicast router that is configured to receive requests from AMT gateways. AMT relay is one end of an AMT tunnel and encapsulates the requested multicast traffic into those tunnels. At one side, it has one or more interfaces connected to the multicast enabled network and on the other side, it has zero or more interfaces connected to the unicast only network and an AMT pseudo-interface.
- **AMT Gateway:** AMT gateway is a host or a router that is connected to the non-multicast capable network. It is the other end of an AMT multicast tunnel and supports the AMT pseudo-interface. The AMT gateway de-encapsulates multicast traffic from those tunnels.
- **AMT Pseudo-Interface:** AMT pseudo-interface is the point where AMT encapsulation and de-encapsulation takes place. It is basically a network interface on the gateway and the relay. Within implementations the AMT pseudo-interface can be treated the same as any other interface or the same as a tunnel end-point.

3.3 AMT Relay Discovery

The technique by which a gateway discovers a relay on the network is described here. There should be an address that is recognized throughout the Internet. In an IP network, one way of providing this function is via an anycast address. Each NSP with an AMT relay needs to advertise this address as reachable throughout the Internet or at least throughout the part of the Internet that a particular relay is intended to serve.

To start the relay discovery process the AMT gateway sends a special message called AMT Relay Discovery Message along with a special code (a Nonce) to the AMT

anycast address. This message is sent to a reserved UDP port 2268. Messages to that address are only responded to by AMT relays. Figure 6 depicts this step.

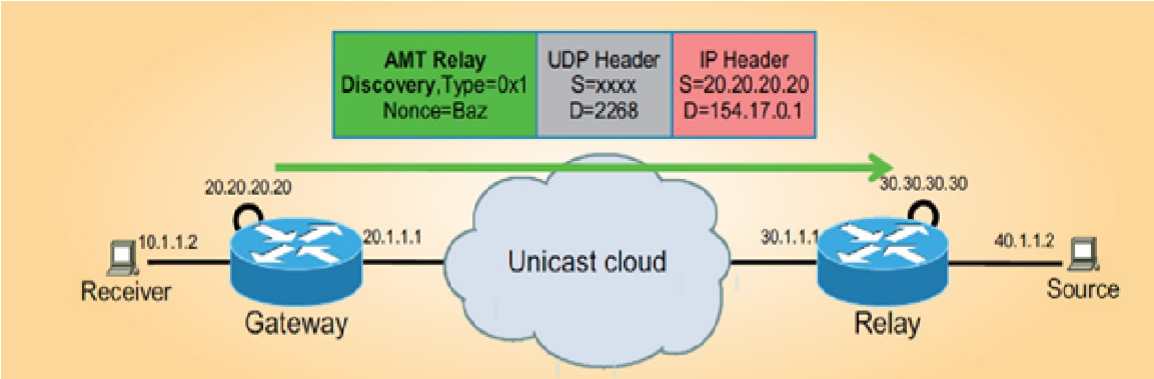

Figure 6: AMT Relay Discovery Message [36].

Upon receiving the AMT Relay Discovery Message the relay replies with an AMT Relay Advertisement message to the gateway. This reply contains the unique IP address of the relay. Thus the gateway understands that the following conversation will be targeted to this address. Also the reply contains the nonce originated by the gateway for ensuring secure communication. Figure 7 describes this part.

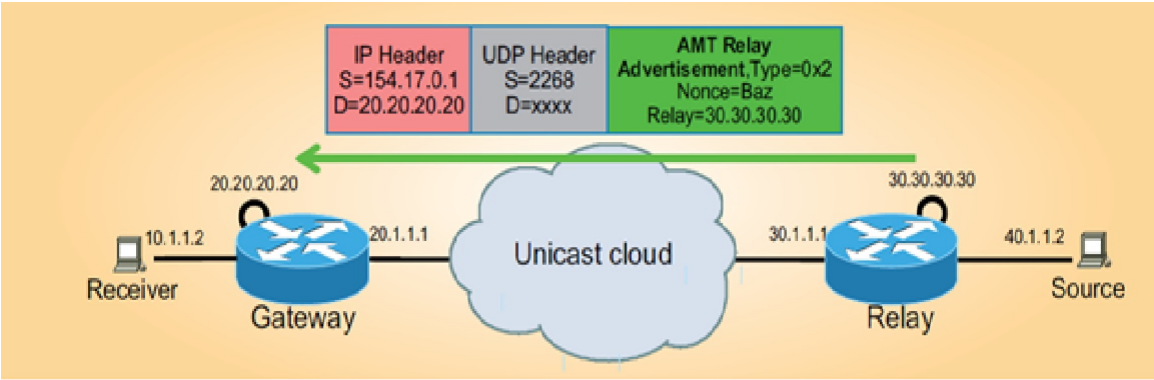

Figure 7: AMT Relay Advertisement Message [36].

3.4 AMT 3-way Handshake

After receiving the AMT Relay Advertisement message, AMT starts the 3-way handshake. At first, it sends an AMT Request Message to the relay. This time it sends

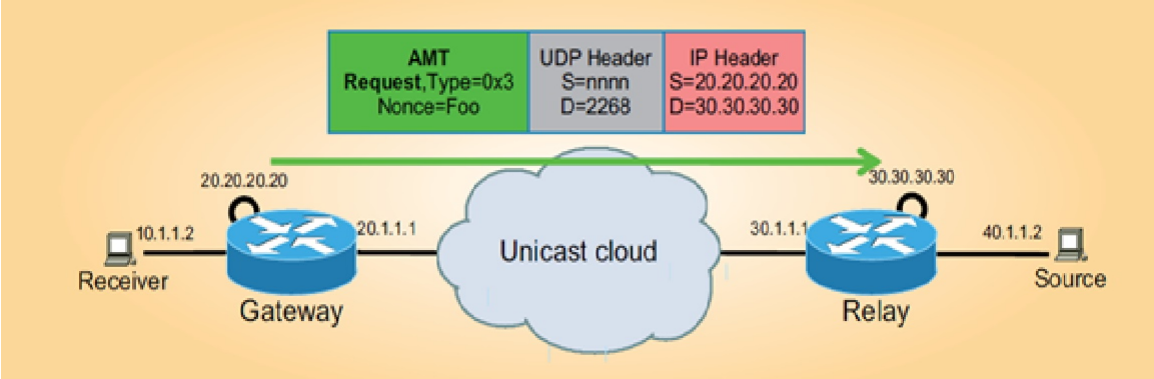

Figure 8: AMT Request Message [36].

this message to the relay's unique IP address. In addition, for security it sends a new nonce along with the message. Figure 8 illustrates this technique.

After that, the relay responds with an AMT Query, which includes the new Nonce from the AMT Request, as well as an opaque security code Message Authentication Code (MAC) that it will expect in any future messages from the gateway. Figure 9 describes this.

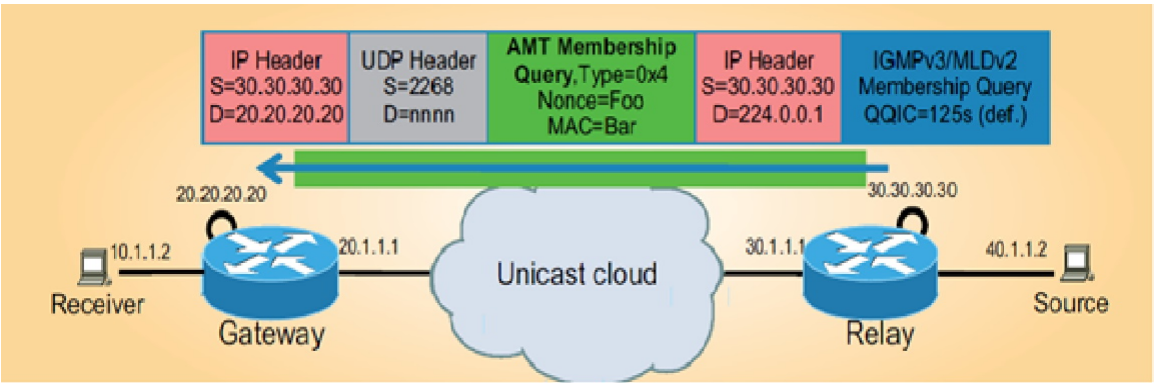

Figure 9: AMT Query Message [36].

The AMT query in fact encapsulates the underlying IGMP membership query and includes the Querier's Query Interval Code (QQIC), which specifies the Query Interval used by the querier.

Now, to join any upstream sources, the gateway responds with an AMT Membership Update, which includes the opaque security code, the original nonce from the AMT Request, and an encapsulated Internet Group Management Protocol version 3 (IGMPv3) packet. Figure 10 depicts this scenario.

The relay will now examines the code and establish the tunnel for further multicast

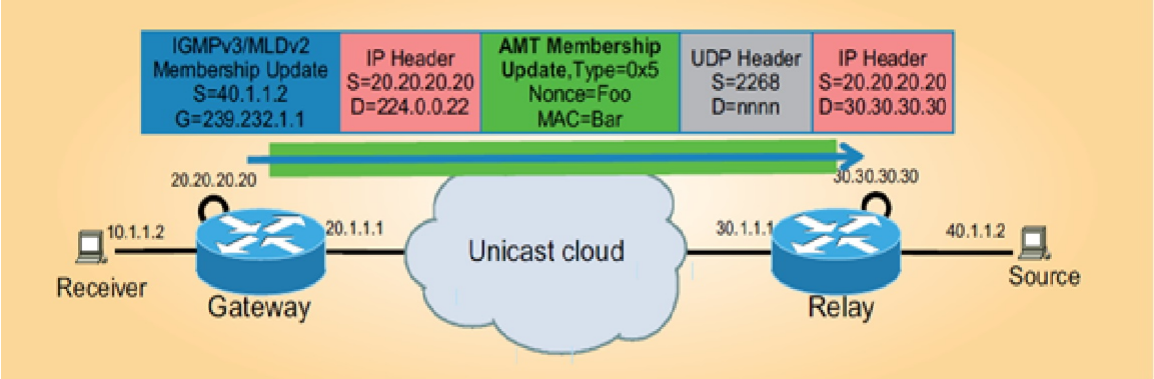

Figure 10: AMT Membership Update Message [36].

traffic. The relay adds the appropriate pseudo/tunnel interface to the multicast route for that particular stream and begins replicating and encapsulating packets to the gateways. Further streams will do the same 3-way handshake process.
However, the discovery message is not required to be resent as the tunnel has already been established. If any relay does not respond with AMT Query then the gateway will start the discovery process all over again. The gateway sends periodic AMT Membership Updates to refresh the state on the relay, sending the appropriate update to leave the group when the traffic is no longer desired. Once the tunnel is no longer required by any more receivers, it is maintained by the gateway / relay for a further time-out period. In that way a new receiver does not need to build a new tunnel if that receiver becomes active again shortly afterwards.

3.5 Advantages of AMT

The advantages of AMT are:

- **Simplicity:** Without experiencing the overhead of any manual configuration and maintenance, the receiving part of AMT simply sends AMT advertisements to a well-known any-cast prefix. The rest of the tunnel establishes automatically without the need for additional supervision.

- **Resiliency:** Because the relay discovery uses an any-cast address, gateways automatically find the closest relay. If that relay becomes unreachable or inaccessible, the routing table will update itself to find the next closest available

relay.

- **Efficiency:** For more efficient link utilization AMT permits transit routers to perform flow-based load balancing.

3.6 Security Considerations for AMT

The following terminology is largely adapted from RFC 7450 — Automatic Multicast Tunneling [12].

AMT is not intended to be a strongly secured protocol. In general, the protocol provides the same level of security and robustness as is provided by the UDP, IGMP and MLD protocols on which it relies. The lack of strong security features can largely be attributed to the desire to make the protocol light-weight by minimizing the state and computation required to service a single gateway, thereby allowing a relay to service a larger number of gateways.

Many threats and vectors may be employed against the protocol to launch various types of denial-of-service attacks that can affect the functioning of gateways or their ability to locate and communicate with a relay.

As is the case for UDP, IGMP and MLD, the AMT protocol provides no mechanisms for ensuring message delivery or integrity. The protocol does not provide confidentiality — multicast groups, sources and streams requested by a gateway are sent in the clear.

The protocol does use a three-way handshake to provide trivial source authentication for state allocation and updates. The protocol also requires gateways and relays to ignore malformed messages and those messages that do not carry expected address values or protocol payload types or content.

- **Relays:** The three-way handshake provided by the membership update message sequence provides a defence against source spoofing-based resource-exhaustion attacks on a relay by requiring source authentication before state allocation. However, attackers may still attempt to flood a relay with Request and Membership Update messages to force the relay to make the hash computations in an effort to consume computational resources. Implementations may choose to limit the frequency with which a relay responds to Request messages sent from

a single IP address or IP address and UDP port pair, but support for this functionality is not required. The three-way handshake provides no defense against an eavesdropping or man-in-the-middle attacker. Attackers that execute the gateway protocol may consume relay resources by instantiating a large number of tunnels or joining a large number of multicast streams. A relay implementation should provide a mechanism for limiting the number of tunnels (Multicast Data message destinations) that can be created for a single gateway source address. Relays should also provide a means for limiting the number of joins per tunnel instance as a defense against these attacks. Relays may withdraw their AMT anycast prefix advertisement when they reach configured maximum capacity or exhaust required resources. This behavior allows gateways to use the relay discovery process to find the next topologically-nearest relay that has advertised the prefix. This behavior also allows a successful resource exhaustion attack to propagate from one relay to the next until all relays reachable using the anycast address have effectively been taken offline. This behavior may also be used to acquire the unicast addresses for individual relays, which can then be used to launch a Distributed Denial of Service (DDoS) attack on all of the relays without using the relay discovery process. To prevent wider disruption of AMT-based distribution network, relay anycast address advertisements can be limited to specific administrative routing domains. This will isolate such attacks to a single domain.

- **Gateways:** A passive eavesdropper may launch a Denial of Service (DoS) attack on a gateway by capturing a Membership Query or Membership Update message and using the request nonce and message authentication code carried by the captured message to send a spoofed Membership Update or Teardown message to the relay. The spoofed messages may be used to modify or destroy group membership state associated with the gateway, thereby changing or interrupting the multicast traffic flows. A passive eavesdropper may also spoof Multicast Data messages in an attempt to overload the gateway or disrupt or supplant existing traffic flows. A properly implemented gateway will filter Multicast Data messages that do not originate from the expected relay address and should filter non-multicast packets and multicast IP packets whose group or source addresses are not included in the current reception state for the gateway

pseudo-interface. An active eavesdropper may launch a man-in-the-middle attack in which messages normally exchanged between a gateway and relay are intercepted, modified, spoofed or discarded by the attacker. The attacker may deny access to, modify or replace requested multicast traffic. The AMT protocol provides no means for detecting or defending against a man-in-the-middle attack — any such functionality must be provided by multicast receiver applications through independent detection and validation of incoming multicast datagrams. The anycast discovery technique for finding relays introduces a risk that a rogue router could introduce a bogus route to a specific Relay Discovery Address prefix, and thus divert or absorb Relay Discovery messages sent by gateways. Network managers must guarantee the integrity of their routing to a particular Relay Discovery Address prefix in much the same way that they guarantee the integrity of all other routes.

- **Encapsulated IP Packets:** An attacker forging or modifying a Membership Query or Membership Update message may attempt to embed something other than an IGMP or MLD message within the encapsulated IP packet carried by these messages in an effort to introduce these into the recipient's IP stack. A properly implemented gateway or relay will ignore any such messages — and may further choose to ignore Membership Query messages that do not contain an IGMP/MLD General Query or a Membership Update message that does not contain IGMP/MLD membership reports. Properly implemented gateways and relays will also filter encapsulated IP packets that appear corrupted or truncated by verifying packet length and checksums.

Chapter 4

Receiver Access Control in AMT

4.1 Introduction

The AMT architecture defined in chapter 3 has limitations. It cannot distinguish between authenticated and un-authenticated users. A solution has been proposed by Malla [38] to achieve RAC in AMT. In order to add RAC in AMT, Malla [38] incorporated different protocols, which are explained briefly in subsections below.

4.1.1 Extensible Authentication Protocols (EAP)

EAP [4] provides for support of multiple authentication methods.

EAP components:
Architecturally, an EAP infrastructure consists of the following. Figure 11 portrays the components.

- **EAP Peer:** Device that is attempting to access a network.
- **EAP Authenticator:** An access point or NAS that is requiring EAP authentication prior to granting access to a network.
- **Authentication Server:** A back end server that negotiates the use of a specific EAP method with an EAP peer, validates the EAP peer's credentials, and authorizes access to the network [3].

Exchanges for EAP authentication:

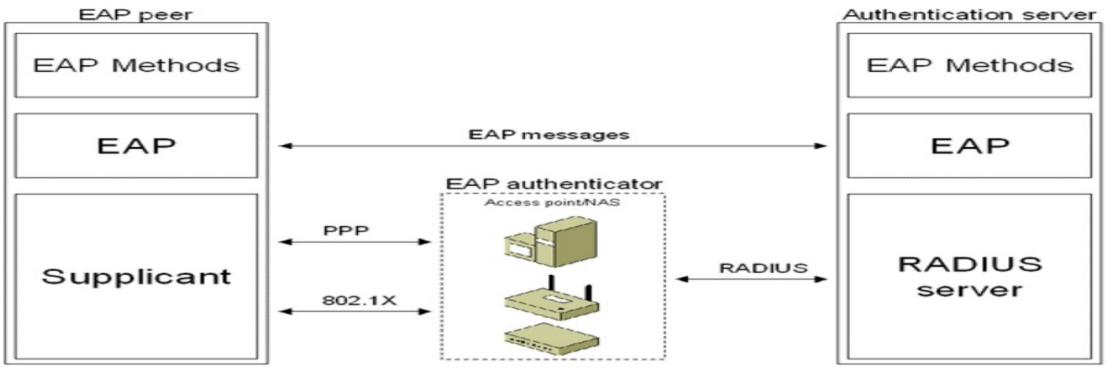

Figure 11: EAP Components [3].

- The authenticator sends a Request to authenticate the peer. The Request has a Type field to indicate what is being requested. Examples of Request Types include Identity, MD5-challenge, etc. [4].
- The EAP peer sends a Response packet in reply to a valid Request. As with the Request packet, the Response packet contains a Type field, which corresponds to the Type field of the Request [4].
- The authenticator sends an additional Request packet, and the peer replies with a Response. The sequence of Requests and Responses continues as long as needed. EAP is a 'lock step' protocol, so that other than the initial Request, a new Request cannot be sent prior to receiving a valid Response [4].
- The conversation continues until the authenticator cannot authenticate the peer, in that case the authenticator must transmit an EAP Failure message. On the other hand, the authentication conversation can continue until the authenticator determines that successful authentication has occurred, in which case the authenticator must transmit an EAP Success message [4].

4.1.2 Protocol for Carrying Authentication for Network Access (PANA)

PANA is an IP-based protocol that allows a host and a network to authenticate each other for network access. An end user can get access to a network's backend

Authentication, Authorization and Accounting (AAA) infrastructure without knowing details about the used protocols. Figure 12 shows the general PANA framework.

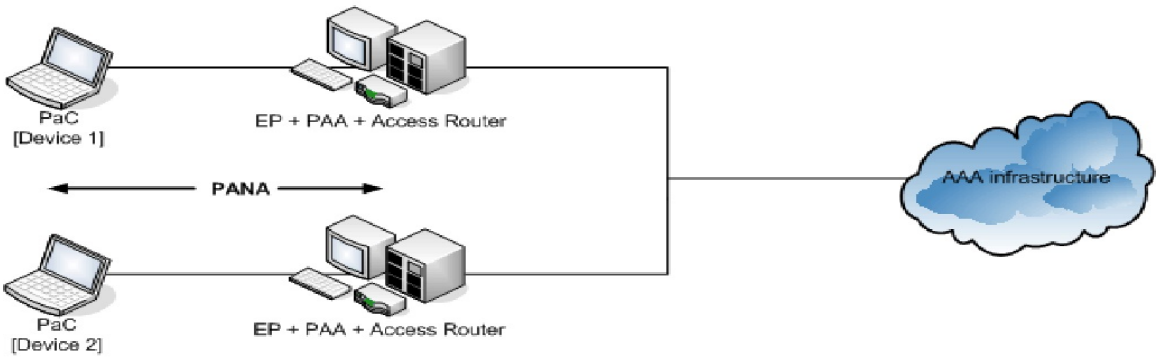

Figure 12: PANA Framework [45].

Components

- **PANA Client(PaC):** Provides the credentials to prove its identity for network access authorization. The PANA Client (PaC) resides in a device that wants to identify itself in order to get access to a network.
- **PANA Authentication Agent (PAA):** The PANA Authentication Agent (PAA) verifies the credentials provided by a PaC and grants or denies access to the device. This is the counterpart to the PaC on the access network.
- **Enforcement Point (EP):** A node on the access network where decisions on per-packet filtering rules for network traffic are implemented depending on the information presented by the PaC.

Protocol Overview

A PANA session consists of four distinct phases:

1. **Authentication and Authorization phase:** This is the phase that initiates a new PANA session and executes EAP between the PaC and PAA. The PAA conveys the result of authentication and authorization to the PaC at the end of this phase.
2. **Access phase:** After successful authentication and authorization, the End User device gains access to the network. Now it can send and receive IP traffic through the Enforcement Point (EP).

3. **Re-authentication phase:** During the access phase, PAA may and PaC should initiate re-authentication if they want to update the PANA session lifetime before it expires. EAP is carried by PANA for re-authentication.

4. **Termination phase:** During this phase an explicit disconnect message is sent by either PaC or PAA to discontinue the access service at any time.

4.1.3 IPsec

IPsec is a set of protocols developed by the IETF to support secure exchange of packets at the IP layer. It can be used to protect any traffic across an IP network. The protocol suite is composed of four main components.

- Security protocols provide traffic security services, such as authentication and encryption. There are two variants: Authentication header (AH) [33] and Encapsulating Security Payload (ESP) [34].

- The security architecture [35] is based on the concept of a Security Association (SA). An SA is a simplex logical connection that affords security services. In IPsec, an SA is a network-level abstraction implemented through the use of AH or ESP. An arbitrary 32-bit value, called a Security Parameter Index (SPI), is used by the receiving end of the connection to identify the SA to which the incoming traffic should be bound. Access to IPsec SAs is managed using three conceptual databases, they are Security Association Database (SAD), Security Policy Database (SPD), Peer Authentication Database (PAD) [35].This architecture is extended by [49], to support the multicast environment.

- The keys that are used in IPsec are managed automatically by some key management Protocols, which will be introduced in the following section.

- For authentication and encryption, some mandatory algorithms are specified in [39] and [44]. These protocols are used in security protocols and key management protocols.

4.1.4 Internet Key Exchange

IKE defined in [24] is the protocol used to set up an SA in the IPsec protocol suite. It was introduced in 1998 and was superseded by IKEv2 [32] later. Figure 13 shows a comparison of the two exchanges.

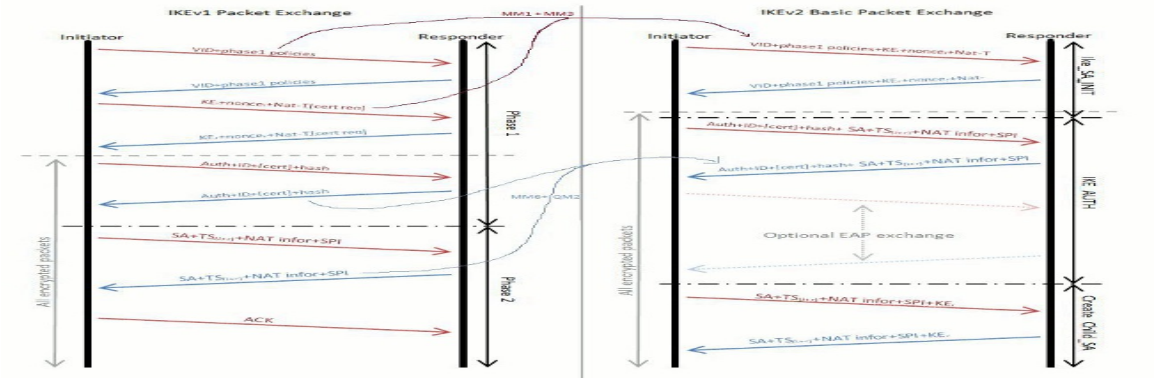

Figure 13: Comparison of IKE and IKEv2 exchanges [10].

IKEv2 is not compatible with IKE and has several advantages, such as simplifying the message exchanges, decreasing the latency (e.g., fewer round trips for the initial exchange), increasing robustness, and so on. (See [32] for the details.) Since an SA is a simplex connection, IKE and IKEv2 establish two SAs between the two devices, one in each direction. The SPI for each direction is chosen by the receiving device, thus ensuring that there is no conflict between the chosen SPI and any other SPI in use on that device.

4.1.5 IGMP and MLD

IGMP is an internet protocol that provides a way for a computer to report its multicast group membership to adjacent routers [42]. There are three versions of IGMP, specified in [18], [22] and [14] respectively. The detailed mechanisms and message formats vary from version to version. There are two kinds of messages in IGMP: Membership Query and Membership Report (including leave group message). Membership Report messages are sent by hosts to report to their adjacent routers about their current multicast reception state, or changes in the multicast reception state of their interfaces. Membership Query messages are sent by a specific router, called the Querier (Q), to query the multicast reception state of adjacent interfaces. In a

network segment, if there are multiple routers, a Q election is required. The router that wins the Q election is the Q while the other routers are Non-Querier (NQ)s [37]. Membership query and membership report messages are sent in the clear, with no attempt at security enforcement.

MLD is the protocol corresponding to IGMP for IPv6 networks. MLD has two versions: Multicast Listener Discovery version 1 (MLDv1) [19], similar to Internet Group Management Protocol version 2 (IGMPv2), and Multicast Listener Discovery version 2 (MLDv2) [47], similar to IGMPv3 [37].

4.1.6 Secure IGMP (SIGMP)

SIGMP, as an extension of IGMP [22], [14] takes over the role of IGMP to show the users' interest in receiving the data from a multicast group. The EU implements the host portion of SIGMP while the AR implements the router portion of SIGMP. SIGMP has two working modes — mode compatible with IGMPv2 and mode compatible with IGMPv3. There are two kinds of queries and two kinds of reports in SIGMP: open group query (OGQ), secure group query (SGQ), open group report (OGR) and secure group report (SGR). In SIGMP, queries and reports for open groups are delivered without any protection, but for secure groups they are protected by IPsec Group Security Association (GSA)s. GSAs are of two kinds: GSA_q and GSA_r. GSA_q is used to protect SGQ messages and GSA_r is used to protect SGR messages [37].

4.1.7 GSAM

The 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 EUs. In GSAM, an AR (specifically, the Q) 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 SPI, 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

SADs and Group Security Policy Database (GSPD)s according to the parameters of GSA_q and GSA_r to protect the SIGMP packets [37] .

4.2 Receiver Access Control in AMT

RAC in AMT [38] is achieved by incorporating EAP, PANA, SIGMP, and GSAM interactions in the AMT environment by extending the functionality of AMT gateway and AMT relay. Figure 14 shows the architecture of this design.

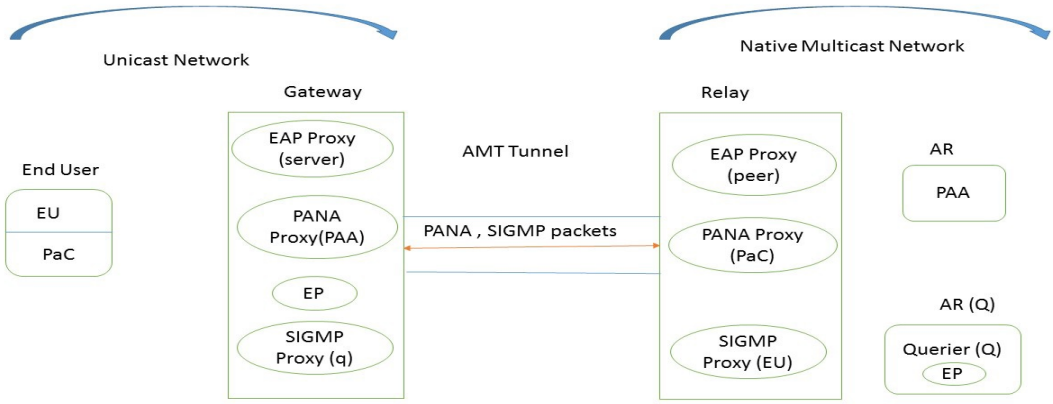

Figure 14: RAC in AMT Communication [38].

4.2.1 Components in AMT with Receiver Access Control

The design introduced in [38] needs EAP, PANA, SIGMP, and GSAM proxies so that all exchanges between the EU host (in the unicast-only region) and the AR (in the multicast-enabled region) flow through the AMT tunnel. The author introduced a proxy in the gateway for each message type; each proxy responds as if it were the AR. Malla [38] also introduced 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, IGMP proxy and the GSAM proxy are simplified because they are all co-located in the gateway.

PANA Proxy

When the (real) PaC (which is the EU) needs to send the first EAP message, it discovers its proxy PAA in the gateway using the normal mechanism for PAA discovery as defined in [41]. The PaC creates a secure connection with the proxy PAA. In the same way, the proxy PaC in the relay creates a secure connection with the real PAA. The gateway and the relay here act as a "friendly" Man-in-the-Middle.

EAP Proxy

The EAP method exports a Master Session Key (MSK) to the PaC and the PAA after the authentication part is done. In turn, the EAP proxies in both the gateway and the relay will know the MSK for protecting SIGMP messages.

SIGMP Proxy

SIGMP on the EU host interacts with the proxy SIGMP on the gateway. In the same way, the proxy SIGMP on the relay interacts with the Q in the native IP multicast region.

GSAM Proxy

To protect the SIGMP exchanges between the EU host and the gateway, GSAM on the EU host uses the keys derived from the MSK and the proxy GSAM identity to form the necessary GSAs. Similarly, to protect the SIGMP exchanges between the relay and the Q, GSAM on the relay uses the keys derived from the MSK and the Q identity to form the necessary GSAs. Although the MSK has the same value, because of the different identities of the EP, they will end up deriving different keys.

4.2.2 Receiver Access Control Interactions in AMT

The receiver access control in AMT can be visualized at two layers.

RAC at the Application Layer

The following sections explain how [38] adapted a regular PANA session, which consists of five phases [23] in AMT using the PANA proxy and the EAP proxy.

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 the actual PAA to the PaC through the relay and the gateway.

Authentication and Authorization Phase: After the handshake phase, EAP packets carried by PANA will be exchanged between the PaC and the PAA. Malla [38] gave an example of Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST) method [15], 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, which shares the secret key with the EU during phase1, will decrypt and forward them to the relay through the AMT Tunnel. Malla [38] assumed that the 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 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 the Enforcement Point in Querier (EPQ) using IPsec, with a key calculated in the normal way for two IPsec peers [51].

Access Phase: PaC and Enforcement Point in Gateway (EPG), 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 Q. With those PEMKs, they establish two different IPsec GSAs between them for cryptographic protection of IGMP messages. Each IPsec

GSA contains one GSA_r and one GSA_q. This phase is also used to test liveness of the PANA session.

Re-authentication and Termination Phases: These phases are similar to the original description [23]. The only difference is that these messages have to go through the AMT tunnel.

RAC at Network Layer

In [38], all the operations for OGQ (Open Group Query) and OGR (Open Group Report) are retained from IGMPv3. In order to provide access control for secure groups a few operations are added in it. The following parts depicts how [38] adapted SIGMP into AMT.

EU Operations: After being authenticated at the application level, the EU will make a request for the network-level join and will send an SIGMP report message. The EU will think that it is sending the request to the real Q. However, 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). 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 the AMT tunnel 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 the IPsec module to decrypt it.

GSAM: 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 [37]. Usually, in an IP multicast environment, GSAM negotiations are done between the EU and the real Q, but in AMT the communication between the EU and Q must go through the AMT tunnel. For this reason [38] implemented 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 the AMT tunnel the SIGMP

proxy acting as EU in the relay performs mutual authentication with the actual Q using PEMK2 and receives a GSA pair from Q.

4.3 Additional Message Types in AMT

The AMT protocol [12] defines seven message types for control and encapsulation. Table 1 shows the existing message types in AMT.

Message Type	Message Name
1	Relay Discovery
2	Relay Advertisement
3	Request
4	Membership Query
5	Membership Update
6	Multicast Data
7	Teardown

Table 1: AMT Message Types (Existing).

However, these message types are not enough to provide RAC in AMT. As discussed in the previous section, in order to include RAC in AMT we have to pass a few additional messages through the AMT tunnel. We have added message type 8, message type 14, message type 15 and message type 17 to the existing message types. Message type 8 is assigned the name "PANA Exchange", message type 14 is assigned the name "Membership Query (Secure)", message type 15 is assigned the name "Membership Update (Secure)" and message type 17 is assigned the name "Teardown (Secure)". Table 2 shows the additional message types in AMT.

Message Type	Message Name
8	PANA Exchange
14	Membership Query (Secure)
15	Membership Update (Secure)
17	Teardown (Secure)

Table 2: Additional AMT Message Types.

These messages are exchanged as IPv4 or IPv6 UDP datagrams. Further details about the additional message types can be found in Appendix A.

Chapter 5

Problem Classification

5.1 Deficiencies

AMT [12] provides multicast service to those areas where IP multicast technology is not supported. Users who are in a unicast-only network can access multicast content with the help of AMT. However, AMT has no security features. It fails to authenticate which users are legitimate and which are not. In other words, there is no RAC in AMT. As discussed earlier, [38] has proposed a solution architecture to provide RAC in AMT. In his work he assumed that the tunnel between the AMT gateway and the AMT relay is secured. However, the tunnel between the gateway and the relay is not secured. Everything that passes through the tunnel is open to the outside world. Any attacker can grab a packet and can see what is in it.
Salem [43] showed that the tunnel formation process in AMT [12], named three-way-handshake between gateway and relay is not safe. The author validated the AMT protocol with a formal validation tool AVISPA and proved the following:

- The relay produced MAC is used only for routability purposes by the respondent, and the originator does not need to know anything about it in terms of content or hashing algorithm. As a result, if any party impersonates itself as an authentic relay, the gateway has no way to figure out otherwise.

- Any intruder can learn the value of the MAC. This gives rise to a scenario where an intruder may be able to make use of the MAC by sending a Membership

update Leave/Done message to the relay while spoofing the source IP of the gateway. This can result in the relay disconnecting the unicast stream to the gateway.

After analyzing the validation results of [43], it is clear that the tunnel between gateway and relay is not secure. The main problem is the gateway and the relay do not have any verifiable identity so that they can validate each other in order to have a secure tunnel. Any intruder can learn the secrets and can play the role of a gateway or a relay.

5.2 Overall System Operation and Goals

In order to achieve security in the AMT tunnel environment, the gateway and the relay should have something to prove their authenticity so that, they can validate each other. We have to introduce a verifiable identity to prove the identity of the gateway and the relay. We will use certificates as verifiable identities. We also have to analyze how the whole idea of distributing the certificates fits into the AMT tunnel environment. After that, with the proper credentials in place, we will incorporate IKEv2 and IPsec to actually secure the tunnel, so that no one can see what is passing through the tunnel and also no one can impersonate the gateway and the relay. Section 3.6 stated the security considerations of AMT protocol [12]. We summarize our security goals in terms of the security considerations of the protocol.

- Entity authentication, which means only a valid gateway or a valid relay can communicate through the tunnel by presenting valid credentials. This will stop a rogue router from introducing a bogus route to a specific Relay Discovery Address prefix, and thus divert or absorb Relay Discovery messages sent by gateways. Also, this will protect against attackers that execute the gateway protocol to consume relay resources by instantiating a large number of tunnels or joining a large number of multicast streams.

- Message authentication, which will ensure source authentication and integrity authentication. This goal will stop an attacker from forging or modifying a Membership Query or Membership Update message may attempt to embed something other than an IGMP or MLD message within the encapsulated IP

packet carried by these messages in an effort to introduce these into the recipient's IP stack.

- Confidentiality of management messages. Management messages are used to distribute and update SAs and their keys. If the keys are revealed, then the whole system will be worthless.

- Resistance to man-in-the-middle attacks. This goal will eliminate the chance of passive and active eavesdropping, which includes spoofing Request and Membership Update messages and forcing the relay to make the hash computations in an effort to consume computational resources or attacking the gateway by spoofing messages on the go and change the course of action.

- Protection against replay attacks. If any attacker wants to impersonate a gateway or a relay by replaying an old message, the system must be able to ignore it. This will also provide protection against passive eavesdropping mentioned in the previous point.

- Resistance to DoS and DDoS attacks. Entity authentication, resistance to man-in-the-middle attacks and protection against replay attacks will ensure there will be no DoS or DDoS attack on the gateway or on the relay.

Considering all these issues, we propose a design solution that provides a secured AMT tunnel.

Chapter 6

Securing an AMT Tunnel

In this chapter, we are going to explain the overall architecture for distributing the certificates to the gateway and the relay. We also going to describe how IKEv2 and IPsec are introduced between the gateway and the relay to achieve a secured AMT tunnel. In order to have a better understanding, we divided the two parts into two separate sections. Section 6.1 will explain the architecture for distributing the certificates and section 6.2 will describe the accommodation of IKEv2 and IPsec in the AMT architecture.

6.1 Proposed Architecture

In order to provide the proper identities to the gateway and the relay, we propose a new architecture for AMT. Figure 15 shows the proposed architecture.

In this architecture we introduced some entities to the existing AMT architecture to help achieving the goal. The whole design is based on the fact that we trust the Certificate Authority (CA).

An EU will negotiate with the NSP to obtain authorization to install and run the gateway in order to access a particular product. The EU will present its verifiable identity or property to the service provider. The NSP will verify the identity and if it thinks that the EU is legitimate then it will provide the EU with a certificate along with the software.

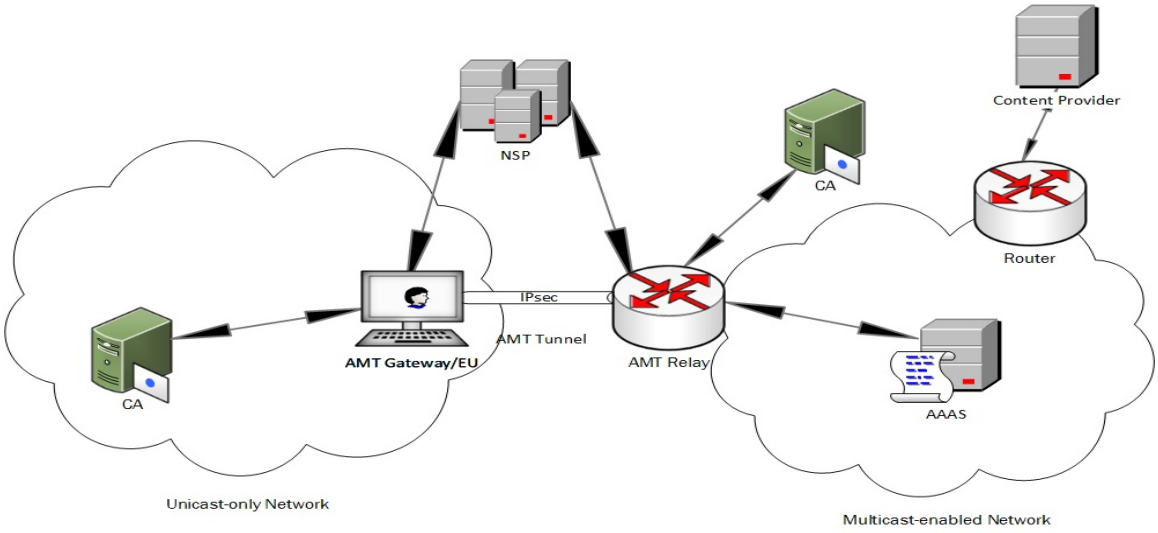

Figure 15: Proposed Architecture with Secured AMT Tunnel.

For example, an EU who wants to install gateway will log in to its personal account on the local NSP website and download the gateway software along with the certificate. The local NSP will verify whether the EU is legitimate or not by verifying its username and password. If it finds the EU legitimate it will provide the EU with a certificate. In real life, the EU will download the gateway software and run the software. The configuration file will handle the job of placing the certificate in place. This certificate can be verified by the relay. The relay is in the multicast network. It can check the validity and authenticity of the certificate by consulting with the AAAS. It can also reach a CA to validate the certificate.

However, for relays, the NSPs have to configure the relays with the necessary certificates. The relay will provide its certificate to the gateway, so that the gateway can verify the relay during authentication. To check the authenticity of the certificate provided by a relay, there should be some rules in the gateway, as it will not be in the multicast network. Gateways can only accept a particular kind of certificate from relays. The gateway can easily reach the CA to verify whether the certificate is valid or not.

Subsection 6.1.1 and subsection 6.1.2 will briefly talk about the standard classification

and verification technique of certificates along with their contents.
Section 6.2 will describe how these certificates will be used to establish IPsec SA between the gateway and the relay.

6.1.1 Classification and Verification of the Distributed Certificates

Commercial CAs use the concept of classes for different types of digital certificates. For example, VeriSign [50] has the following classification

- Class 1 for individuals, intended for email.
- Class 2 for organizations, for which proof of identity is required.
- Class 3 for servers and software signing, for which independent verification and checking of identity and authority is done by the issuing certificate authority.
- Class 4 for online business transactions between companies.
- Class 5 for private organizations or governmental security.

Other vendors may choose to use different classes or no classes at all, though, most do use classes in some form [1].

We can assume that a **Class 1** certificate will be the certificate that will be provided with the gateway software after verification of the EU. This certificate can be verified by the relay. The relay is in the multicast network. It can check the validity and authenticity of the certificate by consulting with the AAAS or a CA.
On the other hand, we can assume that gateway can only accept a **Class 3** certificate from relays. As the validation of **Class 3** certificates is done by the issuing CA, the gateway can easily reach the CA to verify whether the certificate is valid or not.

6.1.2 Contents of the Distributed Certificates

We assume that the certificates will have the standard format. X.509 Version 3 certificates [26] support the following fields . Figure 16 depicts the contents of an X.509 version 3 certificate.

Figure 16: Contents of X.509 Version 3 Certificates [1].

- **Subject:** Provides the name of the computer, user, network device, or service that the CA issues the certificate to. The subject name is commonly represented by using an X.500 or Lightweight Directory Access Protocol (LDAP) format.
- **Serial Number:** Provides a unique identifier for each certificate that a CA issues.
- **Issuer:** Provides a distinguished name for the CA that issued the certificate.
- **Valid From:** Provides the date and time when the certificate becomes valid.
- **Valid To:** Provides the date and time when the certificate is no longer considered valid. The date when an application or service evaluates the certificate must fall between the Valid From and Valid To fields of the certificate for the certificate to be considered valid.
- **Public Key:** Contains the public key of the key pair that is associated with the certificate.
- **Signature Algorithm:** The algorithm used to sign the certificate.
- **Signature Value:** Bit string containing the digital signature.

There are some optional extensions to the fields described above and are not necessarily included in each certificate that the CA issues.

6.2 Accommodation of IKEv2 and IPsec in the AMT Tunnel Environment

This section will describe with the proper credentials in place, how IKEv2 and IPsec are incorporated in the AMT environment to secure the tunnel.

6.2.1 IKEv2 in the AMT Tunnel Environment

The provided certificates will be used by IKEv2 for the authentication of an IPsec SA. The certificate that is provided with the gateway and relay will contain the necessary key and algorithm to check their authenticity. The certificates will also provide evidence that the key used to compute a digital signature belongs to the ID of the sender.

In IKEv2 there are a few other ways for peer authentication. For further details please see [32].

6.2.2 IPsec Security Associations in the AMT Tunnel Environment

Once the authentication credentials are in place, any packet that comes to the AMT tunnel will be secured. For every incoming and outgoing packet IPsec will be invoked. Whether the packet is intended for a secure group or the packet is intended for a open group, any packets that passes through the AMT tunnel will be secured by IPsec. The communication between the gateway and the relay will be unicast.

Gateway Operations for Outgoing Packets

After authentication and authorization at the application level and also at the network level, when a packet comes to the gateway in order to pass it to the relay through the AMT tunnel, the IPsec module will be invoked, no matter which multicast group

(open or secure) is it destined for. For an outgoing packet, the IPsec subsystem of the gateway matches certain fields of the IP header of the packet (primarily the source address, the destination address and the "next protocol"), against the selector of the SPD (as the destination address is a unicast address) to determine the action. If the matching entry specifies "DISCARD", the packet is discarded. If the matching entry specifies "BYPASS", no IPsec operation is needed, the packet is transmitted without any modification. Otherwise, the action specified must be "PROTECT". In that case, the IPsec subsystem determines whether the matching entry in the SPD contains a link to a specific SAD entry (specifying the SPI, the key(s) and the cryptographic algorithms to be used for the SA). If the link to the SAD exists, the outgoing packet will be protected according to the relevant entry in the SAD. However, if it is for the first time for the outgoing packet, the PAD is consulted to determine which key management protocol is to be used to establish the SAs between participants. The key management protocol is responsible for negotiating the parameters to be placed in the SPD and the SAD, corresponding to the newly-established SA.
In our case, the key management protocol will be IKEv2. IKEv2 will negotiate the keys and cryptographic algorithms to create SAs and store the details into the SAD and the SPD.

Gateway Operations for Incoming Packets

For an incoming protected packet, the SPI contained in its IPsec header is used as an index into the SAD, to determine the parameters for decoding the packet contents.

Relay Operations for Outgoing Packets

When a relay is ready to send a reply to any request that a gateway made, the IPsec module of the relay will be invoked. For an outgoing packet, the IPsec subsystem of the gateway matches certain fields of the IP header of the packet (primarily the source address, the destination address and the "next protocol"), against the selector of the SPD (as the destination address is a unicast address) to determine the action. If the matching entry specifies "DISCARD", the packet is discarded. If the matching entry specifies "BYPASS", no IPsec operation is needed, the packet is transmitted without any modification. Otherwise, the action specified must be "PROTECT". In that case, the IPsec subsystem determines whether the matching entry in the SPD contains a

link to a specific SAD entry (specifying the SPI, the key(s) and the cryptographic algorithms to be used for the SA). If the link to the SAD exists, the outgoing packet will be protected according to the relevant entry in the SAD.

Relay Operations for Incoming Packets

For a unicast incoming protected packet, the SPI contained in its IPsec header is used as an index into the SAD, to determine the parameters for decoding the packet contents.

Chapter 7

AVISPA

7.1 Introduction

To quicken the development of the protocols and enhance their security, it is important to have appropriate tools that support the analysis of the protocols and help to find the vulnerabilities in the early stages of development [48]. Favorably, these tools should be entirely automated, robust, expressive, and easily usable, so that they can be integrated into the protocol development and standardization processes to improve the speed and quality of these processes [6]. A verification modeling language named PROMELA (Process or Protocol Meta Language) [2] was introduced to model communicating Finite State Automata but it has no concept of security. Also a number of (semi-)automated protocol analysis tools have been proposed, [7], [11], [20] can analyze small and medium-scale protocols. However, scaling up to large scale Internet security protocols is a considerable challenge, both scientific and technological. To meet the expectations, several European universities and research organizations came up with a push-button tool for the Automated Validation of Internet Security-sensitive Protocols and Applications [6]. The tool is called AVISPA. It has been used by the AVISPA Design Team to validate the security properties of a significant number of IETF networking protocols [6]. It is a "push-button" tool, in the sense that once the protocol under study has been modelled and the security goals stated, the rest of the process of validating the security properties is automatic.

Figure 17 shows the architecture of AVISPA. The first step in using the tool is to

present the analyzed protocol in a special language called HLPSL [5]. We discuss the HLPSL language in more detail in the coming section.

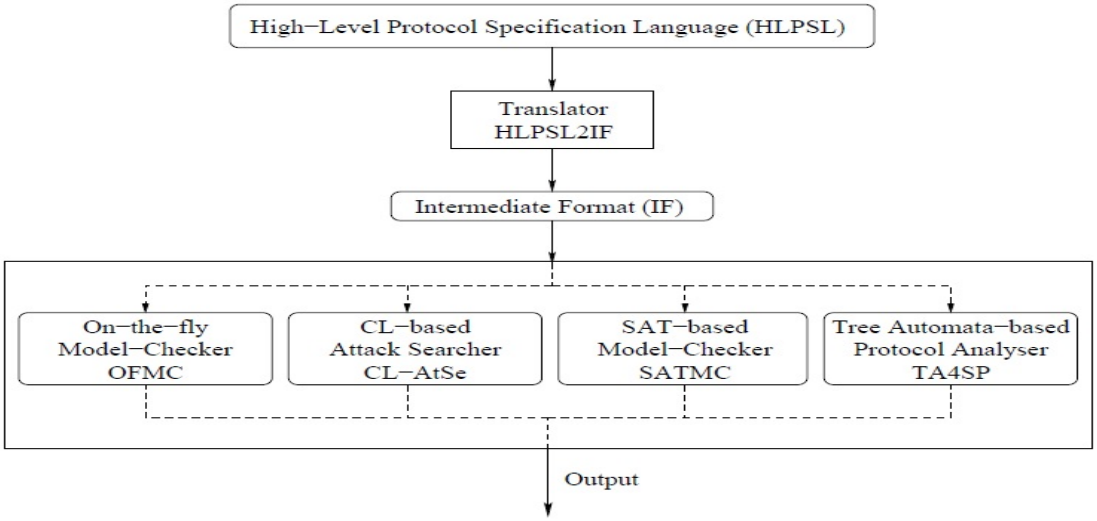

Figure 17: The Architecture of AVISPA [48].

The HLPSL presentation of the protocol is translated into a lower level language called Intermediate Format (IF). This translation is performed by the translator called HLPSL2IF. This step is totally transparent to the user. IF presentation of the protocol is used as an input to the four different back-ends: On-the- fly Model-Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC) and Tree-Automata-based Protocol Analyzer (TA4SP). These back-ends perform the analysis and output the results in a precisely defined output format stating whether there are problems in the protocol or not. Further explanation of the four back-ends is provided in section 7.1.2.

7.1.1 High Level Protocol Specification Language

AVISPA uses HLPSL to present the analyzed protocols. In this section we take a closer look into the structure of HLPSL language according to the AVISPA tutorial [46]. In order to express the protocols in HLPSL language, it is easiest to translate the protocols first into Alice-Bob format, for instance:

```
A -> S: {Kab}_Kas
S -> B: {Kab}_Kbs
```

The notation above illustrates Wide Mouth Frog (WMF) protocol [13], where endpoints A and B attempt to set up a secure session. First A generates a new session key Kab and encrypts it by using a key Kas and sends the encrypted key to the trusted server S. Kas is a key that is shared between A and S. S decrypts the message, reencrypts it by using a shared key Kbs and transmits the encrypted message to B. B can decrypt the message by using the shared secret Kbs and obtains the session key Kab.

HLPSL language is a role-based language, which means that actions of each participant are defined in a separate module, called a basic role. In the case of the WMF example above, the basic roles are: Alice (A), Bob (B) and server (S). Basic roles describe what information the corresponding participant has initially (parameters), its initial state and how the state can change (transitions). To continue the WMF example, the role of Alice would be expressed in following way:

```
role alice(A,B,S : agent,
Kas : symmetric_key,
SND, RCV : channel (dy))
played_by A def=
local
State : nat,
Kab : symmetric_key
init State := 0


transitions
...


end role
```

The role indicates that agents A, B and S are participating in the protocol suite, A has a shared key Kas with the agent S and A uses channels SND (send) and RCV (receive) for communication. Currently, the only supported channel model for communication in AVISPA is Dolev-Yao (dy) [21]. AVISPA's selection of this model is supported

by the fact that this model can emulate the actions of an arbitrary adversary, and it is also very challenging because it gives advantage to the intruder as opposed to other models [16]. Dolev-Yao is a very strong model because it assumes that the intruder can intercept every message in the channel and can build any message from the intercepted messages using for that infinite memory and processing capabilities. It is also based on perfect cryptography, which means that the intruder cannot decrypt a message M ciphered with a key K with another key K' different from K. The section called local defines the local variables of Alice, which are State (natural number (nat)) and symmetric key Kab. The initial state of Alice is 0. The transition section describes received and sent messages and how they affect the state of the role. For instance the role server has following transition called step1:

```
step1. State = 0 /\ RCV({Kab'}_Kas) =|>
State':= 2 /\ SND({Kab'}_Kbs)
```

The transition means that if the server's state is 0 and it receives a message from its RCV channel containing a key Kab' that is encrypted with a key Kas, the server changes its state to 2, encrypts the key Kab' with the Kbs and sends the encrypted key to the channel SND.
In addition to basic roles the HLPSL language defines also so called composition roles that are used to combine several basic roles. Combining the basic roles means that the roles can execute in parallel. The composition roles define the actual protocol sessions. For instance, in the case of the WMF protocol there are three basic roles Alice, Bob and Server. The composition role, called session, initiates one instance of each role and thus defines one protocol run. The composition role does not define transitions the way basic roles do, instead it initiates basic roles and defines channels used by the basic roles. The composition role is defined for instance in the following way:

```
role session(A,B,S :agent,
Kas,Kbs :symmetric_key) def=



local SA, RA, SB, RB SS, RS: channel (dy)
composition
```

```
alice (A, B, S, Kas, SA, RA)
/\ bob (B, A, S, Kbs, SB, RB)
/\ server(S, A, B, Kas, Kbs, SS, RS)


end role
```

Finally the HLPSL defines a top level role, called here as environment, that contains global variables and combines several sessions. This top level role can be used to define what information an intruder has and where the intruder can access the protocol. For example, the intruder may play a role of a legitimate user in a protocol run. The following role definition shows how a top level environment can be defined. The letter i in the definition indicates the intruder.

```
role environment()
def=


const a, b, s : agent,
kas, kbs, kis : symmetric_key


intruder_knowledge = {a, b, s, kis}


composition
session(a,b,s,kas,kbs)
/\ session(a,i,s,kas,kis)
/\ session(i,b,s,kis,kbs)


end role
```

Every security protocol has some goals that it is supposed to meet. In order to write the protocol in HLPSL format, we must know these goals. The analysis is done

against the defined security goals and the results indicate whether the protocol meets the goals or not.

The security goals of the protocol are presented in an HLPSL language section called goals. Security goals are actually defined in transition sections of basic roles. The definitions of security goals in the transition section are called goal facts. The goals section simply describes which combinations of these goal facts indicate an attack [48].

Below there is an example of a goal fact. The notation means that Bob allows that the key K1 can be shared with Alice, but it must remain secret between the two. The second argument of the secret fact is called protocol id and it simply names the secret fact and distinguishes the different security goals from each other.

```
role bob {

...
local
State : nat,
Nb,Na : text,
K1 : message
init
State := 1

transition
1. State = 1 /\ RCV({Na'}_K) =|>
State':= 3 /\ Nb' := new()
/\ SND({Nb'}_K)
/\ K1':= Hash(Na'.Nb')
/\ secret(K1',k1,{A,B})
...
end role
```

7.1.2 The Back-Ends of AVISPA

An entity that inputs a sequence of IF language statements, does analysis and produces the analysis output is known as back-end. Figure 17 shows four different back-ends of AVISPA named OFMC, CL-AtSe, SATMC and TA4SP. They are complementary rather than equivalent. Thus, the output of the back-ends may differ. All back-ends assume perfect cryptography, which means that an attacker cannot solve encryption without the knowledge of the whole key. Also, the transmission channel is assumed to be controlled by a Dolev-Yao attacker. This means that the attacker has basically full control over the channel [48].

- **The On-the-fly Model-Checker (OFMC):**
  OFMC [8] performs protocol falsification and bounded validation by exploring the transition system described by an IF specification in a demand-driven way. OFMC implements a number of correct and complete symbolic techniques. It supports the specification of algebraic properties of cryptographic operators, and typed and untyped protocol models.

- **The Constraint-Logic-based Attack Searcher (CL-AtSe):**
  CL-AtSe [8] applies constraint solving as in [17], with some powerful simplification heuristics and redundancy elimination techniques. CL-AtSe is built in a modular way and is open to extensions for handling algebraic properties of cryptographic operators. It supports type-flaw detection and handles associativity of message concatenation.

- **The SAT-based Model-Checker (SATMC):**
  SATMC [8] builds a propositional formula encoding a bounded unrolling of the transition relation specified by the IF, the initial state and the set of states representing a violation of the security properties. The propositional formula is then fed to a state-of-the-art SAT solver and any model found is translated back into an attack.

- **The Tree Automata based on Automatic Approximations for the Analysis of Security Protocols(TA4SP):**
  TA4SP [8] approximates the intruder knowledge by using regular tree languages

and rewriting. For secrecy properties, TA4SP can show whether a protocol is flawed (by under-approximation) or whether it is safe for any number of sessions (by over-approximation).

7.2 HLPSL Model

This section describes the HLPSL model that is used to validate our design.
In the previous chapter we specified that IKEv2 [32] has different variants for authenticating peers. Our design requires certificates for authentication purpose. The certificates will carry the necessary keys along with the algorithm to compute the MAC of the key to be authenticated. Ultimately, the authentication procedure will be executed by exchanging the MAC of a pre-shared secret that both nodes possess. As IKEv2 (authentication based on MAC) is already validated in [40], we adapt this model to validate our design.

7.2.1 HLPSL of Role Gateway

The role gateway defines its parameters and transitions. In the transition part, the gateway starts the IKE_SA_INIT exchange by sending the cryptographic algorithms the initiator supports for the IKE SA, initiator's Diffie-Hellman value and its nonce. After checking that relay sent the same nonce, the gateway sends the first authentication message of the IKE_AUTH exchange. As authentication Data, the gateway signs its first message and relay's nonce along with its identity.

```
role gateway(G,R: agent,
X: text,
F: hash_func,
PSK: symmetric_key,
SND_R, RCV_R: channel (dy))



played_by G
def=
```

```
local Ni, SA1, SA2, DHX: text,
Nr: text,
KEr: message, %% more spefic: exp(text,text)
SK: hash(text.text.text.message),
State: nat,
AUTH_R: message
const sec_g_SK : protocol_id

init State := 0
transition


1. State = 0 /\ RCV_R(start) =|>
   State':= 2 /\ SA1' := new()
  /\ DHX' := new()
  /\ Ni' := new()
      /\ SND_R( SA1'.exp(X,DHX').Ni' )



2. State = 2 /\ RCV_R(SA1.KEr'.Nr') =|>
   State':= 4 /\ SA2' := new()
  /\ SK' := F(Ni.Nr'.SA1.exp(KEr',DHX))
  /\ SND_R( {G.F(PSK.SA1.exp(X,DHX).Ni.Nr').SA2'}_SK' )
  /\ witness(G,R,sk2,F(Ni.Nr'.SA1.exp(KEr',DHX)))
3. State = 4 /\ RCV_R({R.F(PSK.SA1.KEr.Ni.Nr).SA2}_SK) =|>
   State':= 6 /\ AUTH_R' := F(PSK.SA1.KEr.Ni.Nr)
  /\ secret(SK,sec_g_SK,{G,R})
  /\ request(G,R,sk1,SK)

end role
```

7.2.2 HLPSL of Role Relay

The role relay defines its parameters and transitions. In the transition part, after receiving the initial message of IKE_SA_INIT exchange, the relay chooses a cryptographic suite from the gateway's offered choices and sends it to the gateway along with the completed Diffie-Hellman exchange and its nonce. After that, when the relay receives the first message of the IKE_AUTH exchange, it responds with an authentication message. as authentication data, the relay signs its first message and gateway's nonce along with its identity.

```
role relay(R,G:agent,
X: text,
F: hash_func,
PSK: symmetric_key,
SND_G, RCV_G: channel (dy))


played_by R
def=


local Ni, SA1, SA2: text,
Nr, DHY: text,
SK: hash(text.text.text.message),
KEi: message,
State: nat,
AUTH_G: message
const sec_r_SK : protocol_id
init State := 1


transition


1. State = 1 /\ RCV_G( SA1'.KEi'.Ni' ) =|>
```

```
   State':=3 /\ DHY' := new()
 /\ Nr' := new()
 /\ SND_G(SA1'.exp(X,DHY').Nr')
 /\ SK' := F(Ni'.Nr'.SA1'.exp(KEi',DHY'))


2. State = 3 /\ RCV_G( {G.F(PSK.SA1.KEi.Ni.Nr).SA2'}_SK ) =|>
   State':=5 /\ SND_G( {R.F(PSK.SA1.exp(X,DHY).Ni.Nr).SA2'}_SK )
 /\ AUTH_G' := F(PSK.SA1.KEi.Ni.Nr)
 /\ witness(R,G,sk1,SK)
 /\ secret(SK,sec_r_SK,{G,R})
 /\ request(R,G,sk2,SK)

end role
```

7.2.3 HLPSL of Role Session

This section defines the role Session. This session role describes sessions of the protocol by instantiating basic roles gateway and relay, so they execute together in parallel.

```
role session(G, R: agent,
PSK: symmetric_key,
X: text,
F: hash_func)
def=


local SG, RG, SR, RR: channel (dy)


composition
   gateway(G,R,X,F,PSK,SG,RG)
/\ relay(R,G,X,F,PSK,SR,RR)
```

```
end role
```

7.2.4 HLPSL of Role Environment

Finally the role environment contains the number of parallel sessions and also the intruder's knowledge.

```
role environment()

def=
const sk1, sk2 : protocol_id,
g, r : agent,
kgr, kgi, kri : symmetric_key,
x : text,
f : hash_func



intruder_knowledge = {x,f,g,r,i,kgi,kri
 }
composition
   session(g,r,kgr,x,f)
/\ session(g,i,kgi,x,f)
/\ session(i,r,kri,x,f)

end role
```

7.2.5 Security Goals

Security goals are specified in HLPSL by augmenting the transitions of basic roles with so-called goal facts and by then assigning them a meaning by describing, in the HLPSL goal section, what conditions — that is, what combination of such facts —

indicate an attack. The two most frequently used security goals are authentication and secrecy [46].
Secret events are goal facts where the goal facts assert which values should be kept as secret and between whom. The goal declaration in the goal section describes that anytime the intruder learns the secret value, then it should be considered an attack [46].
The witness and request events are goal facts related to authentication of an agent. They are used to check that a participant is right in believing that its intended peer is present in the current session, has reached a certain state, and agrees on a certain value, which typically is fresh [46].
The HLPSL model of our goal section is given below. The goal sections describes that key SK should be secret between the gateway and the relay. It also describes that gateway should authenticate relay on sk1 and relay should authenticate gateway on sk2.

```
goal

%secrecy_of SK
secrecy_of sec_g_SK, sec_r_SK

authentication_on sk1

authentication_on sk2

end goal
```

7.3 Results

No attack has been found, considering the security goals mentioned in goal section. AVISPA back-ends OFMC and CL-AtSe have produced the result as safe. Based on the validation result and the security goals in 5.2, we summarize the safety of our proposed design in the following part.

- Entity authentication, message authentication, protection against replay attacks, resistance to man-in-the-middle attacks, resistance to DoS and DDoS

attacks are achieved by the fact that the gateway authenticates relay on sk1 and relay authenticates gateway on sk2.

- The secrecy of SK states that the session key which is used for SA is not compromised.

This shows our design for achieving a secured AMT tunnel is safe to all those possible attacks.

Chapter 8

Conclusions and Future Work

This thesis proposes a solution for securing an AMT tunnel. This solution implies that, any packet that comes to the AMT tunnel will be secured. We explained how the negotiation of proper credentials for IKEv2 is done so that the security goals can be met. Later we also explain, with the proper credentials in place, how we can accommodate IPsec in the tunnel environment — that is between the gateway and the relay. We have also added four new types of messages to the existing AMT message types in order to achieve RAC in AMT. Finally we validated our solution through AVISPA, which showed that the design is safe.

Our proposed solution is the solution for the last part of the problem for achieving RAC in AMT. Future work can be the implementation of these ideas. Implementation of the whole solution of RAC in AMT can help us realizing how it fits into the real life scenarios.

Appendix A

Additional AMT Message Types

A.1 PANA Exchanges

The UDP/IP datagram containing this message MUST carry a valid, nonzero UDP checksum and carry the following IP address and UDP port values:

- **Source IP Address**- The destination IP address carried by the Relay Discovery message (i.e., the Relay Discovery Address advertised by the relay).

- **Source UDP Port** - The destination UDP port carried by the Relay Discovery message (i.e., the IANA-assigned AMT port number).

- **Destination IP Address** - The source IP address carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.

- **Destination UDP Port** - The source UDP port carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway. PANA Exchanges message format is shown in Figure 18.

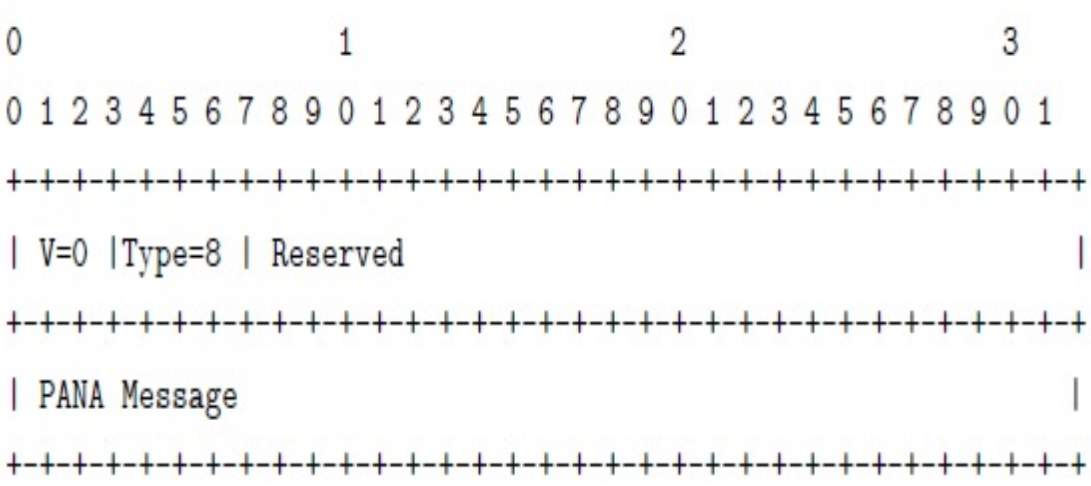

Figure 18: PANA Exchanges Message Format.

A.1.1 Version (V)

The protocol version number for this message is 0.

A.1.2 Type

The type number for this message is 8.

A.1.3 Reserved

Bits that MUST be set to zero by the relay and ignored by the gateway.

A.1.4 PANA Message

Any PANA message generated by the PAA - PANA Proxy(gateway) and the PaC - PANA Proxy(relay).

A.2 Membership Query (Secure)

A relay sends a Membership Query (Secure) message to a gateway to solicit a Membership Update response, but only after receiving a Request message from the gateway. If the relay wants to join an open group then, for Membership Query message, message

type 4 will be used. Message type 4 (Membership Query) uses the normal IGMPv3 messages for IPv4 or MLDv2 messages for IPv6. However, if the relay wants to join a secure group then, for Membership Query message, type 14 will be used. Message type 14 (Membership Query (Secure)) uses Secure IGMP (SIGMP) messages. As the packet formats are the same for both IGMP and SIGMP messages, the only change in AMT packet format will be in the message type.
The successful delivery of this message to a gateway marks the start of the second stage in the three-way handshake used to create or update tunnel state within a relay. The UDP/IP datagram containing this message MUST carry a valid, nonzero UDP checksum and carry the following IP address and UDP port values:

- **Source IP Address**- The destination IP address carried by the Relay Discovery message (i.e., the Relay Discovery Address advertised by the relay).

- **Source UDP Port** - The destination UDP port carried by the Relay Discovery message (i.e., the IANA-assigned AMT port number).

- **Destination IP Address** - The source IP address carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.

- **Destination UDP Port** - The source UDP port carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway. Membership Query (Secure) message format is shown in Figure 19.

A.2.1 Version (V)

The protocol version number for this message is 0.

A.2.2 Type

The type number for this message is 14.

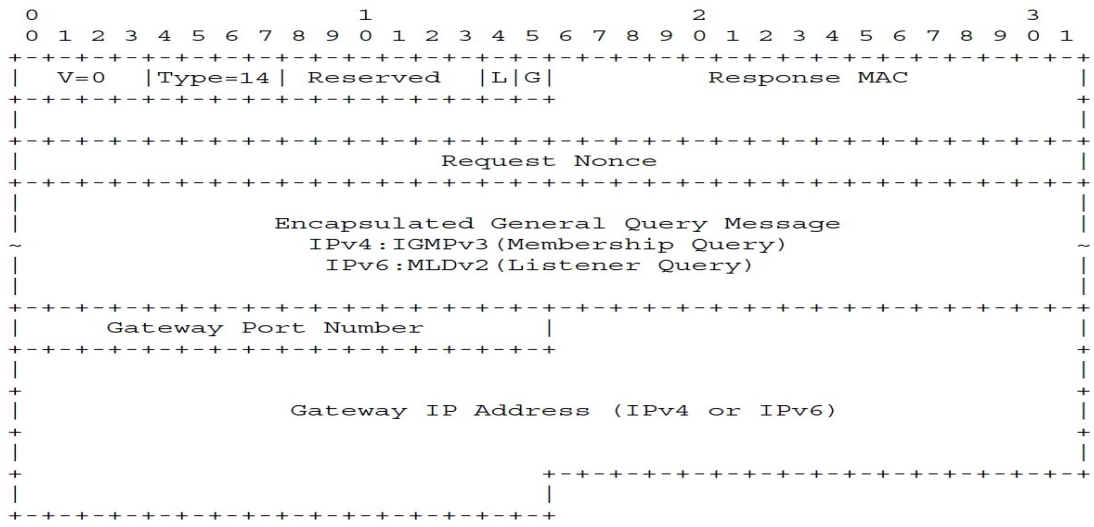

Figure 19: Membership Query (Secure) Message Format.

A.2.3 Reserved

Bits that MUST be set to zero by the relay and ignored by the gateway.

A.2.4 Limit (L) Flag

A 1-bit flag set to 1 to indicate that the relay is NOT accepting Membership Update messages from new gateway tunnel endpoints and that it will ignore any that are. A value of 0 has no special significance - the relay may or may not be accepting Membership Update messages from new gateway tunnel endpoints. A gateway checks this flag before attempting to create new group subscription state on the relay to determine whether it should restart relay discovery. A gateway that has already created group subscriptions on the relay may ignore this flag. Support for this flag is RECOMMENDED.

A.2.5 Gateway Address (G) Flag

A 1-bit flag set to 0 to indicate that the message does NOT carry the Gateway Port and Gateway IP Address fields, and 1 to indicate that it does. A relay implementation that supports the optional teardown procedure SHOULD set this flag and the Gateway Address field values. If a relay sets this flag, it MUST also include the Gateway Address fields in the message. A gateway implementation that does not support the

optional teardown procedure MAY ignore this flag and the Gateway Address fields if they are present.

A.2.6 Response MAC

A 48-bit source authentication value generated by the relay. The gateway echoes this value in subsequent Membership Update messages to allow the relay to verify that the sender of a Membership Update message was the intended receiver of a Membership Query sent by the relay.

A.2.7 Request Nonce

A 32-bit value copied from the Request Nonce field carried by a Request message. The relay will have included this value in the Response MAC computation. The gateway echoes this value in subsequent Membership Update messages. The gateway also uses this value to match a Membership Query to a Request message.

A.2.8 Encapsulated General Query Message

An IP-encapsulated IGMP or MLD message generated by the relay. This field will contain one of the following IP datagrams:

IPv4: IGMPv3 Membership Query
IPv6: MLDv2 Listener Query

The source address carried by the query message should be set as described in Section 5.3.3.3 in [12]. The Querier's Query Interval Code (QQIC) field in the general query is used by a relay to specify the time offset a gateway should use to schedule a new three-way handshake to refresh the group membership state within the relay (current time + Query Interval). The QQIC field is defined in Section 4.1.7 in [14] and Section 5.1.9 in [47].
The Querier's Robustness Variable (QRV) field in the general query is used by a relay to specify the number of times a gateway should retransmit unsolicited membership reports, encapsulated within Membership Update messages, and optionally, the number of times to send a Teardown message. The QRV field is defined in Section 4.1.6

in [14] and Section 5.1.8 in [47].

A.2.9 Gateway Address Fields

The Gateway Port Number and Gateway Address fields are present in the Membership Query message if, and only if, the "G" flag is set. A gateway need not parse the encapsulated IP datagram to determine the position of these fields within the UDP datagram containing the Membership Query message — if the G-flag is set, the gateway may simply subtract the total length of the fields (18 bytes) from the total length of the UDP datagram to obtain the offset.

- **Gateway Port Number** A 16-bit UDP port containing a UDP port value. The Relay sets this field to the value of the UDP source port of the Request message that triggered the Query message.
- **Gateway IP Address** A 16-byte IP address that, when combined with the value contained in the Gateway Port Number field, forms the gateway endpoint address that the relay will use to identify the tunnel instance, if any, created by a subsequent Membership Update message. This field may contain an IPv6 address or an IPv4 address stored as an IPv4-compatible IPv6 address, where the IPv4 address is prefixed with 96 bits set to zero (See [25]). This address must match that used by the relay to compute the value stored in the Response MAC field.

A.3 Membership Update (Secure)

A gateway sends a Membership Update (Secure) message to a relay to report a change in group membership state, or to report the current group membership state in response to receiving a Membership Query (Secure) message. The gateway encapsulates the IGMP or MLD message as an IP datagram within a Membership Update message and sends it to the relay, where it will be de-encapsulated and processed by the relay to update group membership and forwarding state. If the relay wants to join an open group, then for Membership Update message, message type 5 will be used. Message type 5 (Membership Query) will use the normal IGMPv3 messages for IPv4 or MLDv2 messages for IPv6. However, if the relay wants to join a secure group, then

for Membership Update message, type 15 will be used. Message type 15 (Membership Query (Secure)) will use SIGMP messages. As the packet formats are same for both IGMP and SIGMP messages, the only change in AMT packet format will be in the message type.

A gateway cannot send a Membership Update (Secure) message until it receives a Membership Query (Secure) from a relay because the gateway must copy the Request Nonce and Response MAC values carried by a Membership Query (Secure) into any subsequent Membership Update messages it sends back to that relay. These values are used by the relay to verify that the sender of the Membership Update (Secure) message was the recipient of the Membership Query (Secure) message from which these values were copied. The successful delivery of this message to the relay marks the start of the final stage in the three-way handshake. This stage concludes when the relay successfully verifies that sender of the Membership Update (Secure) message was the recipient of a Membership Query (Secure) message sent earlier. At this point, the relay may proceed to process the encapsulated IGMP or MLD message to create or update group membership and forwarding state on behalf of the gateway.

The UDP/IP datagram containing this message MUST carry a valid, non-zero UDP checksum and carry the following IP address and UDP port values:

- **Source IP Address**- The destination IP address carried by the Relay Discovery message (i.e., the Relay Discovery Address advertised by the relay).

- **Source UDP Port** - The destination UDP port carried by the Relay Discovery message (i.e., the IANA-assigned AMT port number).

- **Destination IP Address** - The source IP address carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway.

- **Destination UDP Port** - The source UDP port carried by the Relay Discovery message. Note: The value of this field may be changed as a result of network address translation before arriving at the gateway. Membership Update (Secure) message format is shown in Figure 20.

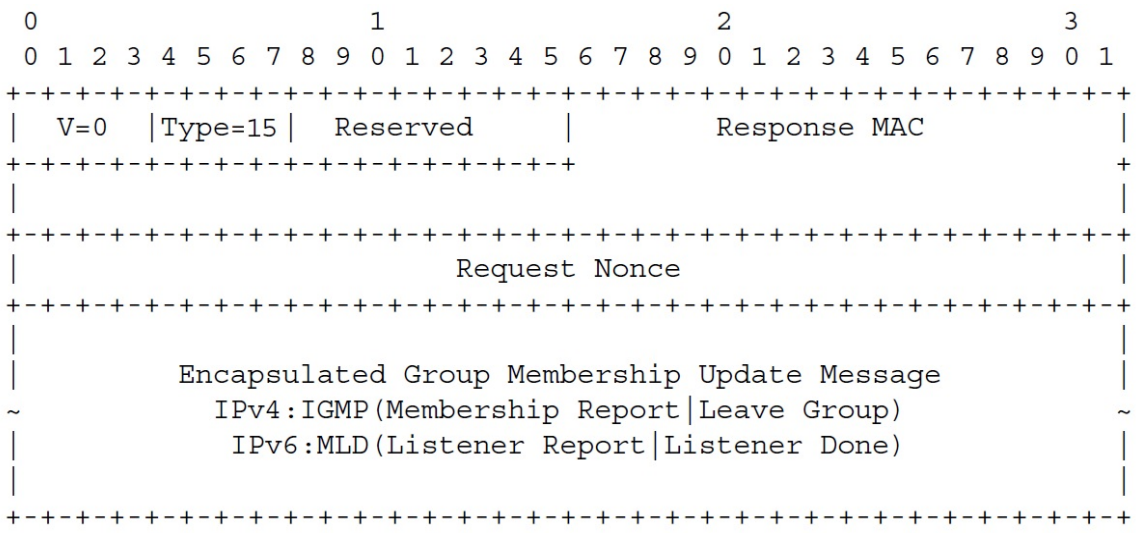

Figure 20: Membership Update (Secure) Message Format.

A.3.1 Version (V)

The protocol version number for this message is 0.

A.3.2 Type

The type number for this message is 15.

A.3.3 Reserved

Bits that MUST be set to zero by the relay and ignored by the gateway.

A.3.4 Response MAC

A 48-bit value copied from the Response MAC field (Section 5.1.4.6 in [12]) in a Membership Query (Secure) message. Used by the relay to perform source authentication.

A.3.5 Request Nonce

A 32-bit value copied from the Request Nonce field in a Request or Membership Query (Secure) message. Used by the relay to perform source authentication.

A.3.6 Encapsulated Group Membership Update Message

An IP-encapsulated SIGMP message produced by the host-mode SIGMP protocol running on a gateway pseudo-interface. This field will contain one of the following IP datagrams:
IPv4:IGMPv2 Membership Report
IPv4:IGMPv2 Leave Group
IPv4:IGMPv3 Membership Report
IPv6:MLDv1 Multicast Listener Report
IPv6:MLDv1 Multicast Listener Done
IPv6:MLDv2 Multicast Listener Report
The source address carried by the message should be set as described in Section 5.2.1 in [12].

A.4 Teardown (Secure)

A gateway sends a Teardown message to a relay to request that it stop sending Multicast Data messages to a tunnel endpoint created by an earlier Membership Update message. A gateway sends this message when it detects that a Request message sent to the relay carries an address that differs from that carried by a previous Request message. The gateway uses the Gateway IP Address and Gateway Port Number Fields in the Membership Query message to detect these address changes.
To provide backwards compatibility with early implementations of the AMT protocol, support for this message and associated procedures is considered OPTIONAL — gateways are not required to send this message and relays are not required to act upon it.
The UDP/IP datagram containing this message MUST carry a valid, nonzero UDP checksum and carry the following IP address and UDP port values:

- **Source IP Address**- The IP address of the gateway interface used to send the message. This address may differ from that used to send earlier messages. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

- **Source UDP Port** - The UDP port number. This port number may differ from that used to send earlier messages. Note: The value of this field may be changed as a result of network address translation before arriving at the relay.

- **Destination IP Address** - The unicast IP address of the relay.

- **Destination UDP Port** - The IANA-assigned AMT port number. Teardown (Secure) message format is shown in Figure 21.

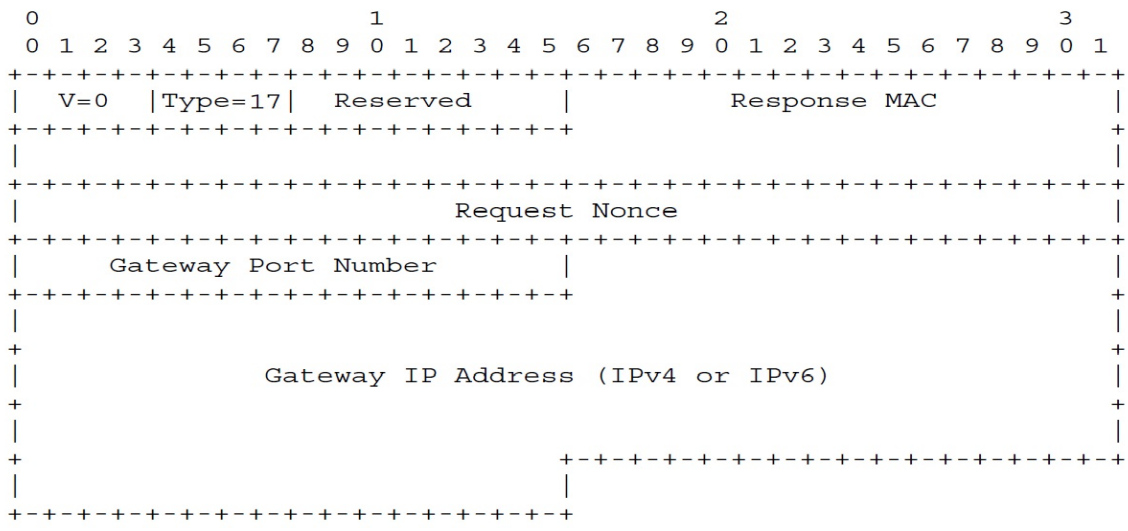

Figure 21: Teardown (Secure) Message Format.

A.4.1 Version (V)

The protocol version number for this message is 0.

A.4.2 Type

The type number for this message is 17.

A.4.3 Reserved

Bits that MUST be set to zero by the relay and ignored by the gateway.

A.4.4 Response MAC

A 48-bit value copied from the Response MAC field (Section 5.1.4.6 in [12]) in the last Membership Query (Secure) message the relay sent to the gateway endpoint address of the tunnel to be torn down. The gateway endpoint address is provided by the Gateway IP Address and Gateway Port Number fields carried by the Membership Query message. The relay validates the Teardown message by comparing this value with one computed from the Gateway IP Address, Gateway Port Number, Request Nonce fields and a private secret.

A.4.5 Request Nonce

A 32-bit value copied from the Request Nonce field (Section 5.1.4.7 [12]) in the last Membership Query (Secure) message the relay sent to the gateway endpoint address of the tunnel to be torn down. The gateway endpoint address is provided by the Gateway IP Address and Gateway Port Number fields carried by the Membership Query (Secure) message. This value must match that used by the relay to compute the value stored in the Response MAC field.

A.4.6 Gateway Port Number

A 16-bit UDP port number that, when combined with the value contained in the Gateway IP Address field, forms the tunnel endpoint address that the relay will use to identify the tunnel instance to tear down. The relay provides this value to the gateway using the Gateway Port Number field (Section 5.1.4.9.1 in [12]) in a Membership Query message. This port number must match that used by the relay to compute the value stored in the Response MAC field.

A.4.7 Gateway IP Address

A 16-byte IP address that, when combined with the value contained in the Gateway Port Number field, forms the tunnel endpoint address that the relay will used to identify the tunnel instance to tear down. The relay provides this value to the gateway using the Gateway IP Address field (Section 5.1.4.9.2 in [12]) in a Membership Query (Secure) message. This field may contain an IPv6 address or an IPv4 address stored

as an IPv4-compatible IPv6 address, where the IPv4 address is prefixed with 96 bits set to zero (See [25]). This address must match that used by the relay to compute the value stored in the Response MAC field.

Bibliography

[1] Web service security tutorial. https://sites.google.com/site/ddmwsst/digital-certificates. Accessed: 2015-08-26.

[2] *Concise Promela Reference.* *http://spinroot.com/spin/Man/Quick.html*, June 1997. Accessed: 2015-08-11.

[3] *Extensible Authentication Protocol Overview.* *https://technet.microsoft.com/en-us/library/bb457039.aspx*, 2015. Accessed: 2015-09-11.

[4] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible authentication protocol (EAP). https://www.rfc-editor.org/rfc/rfc3748.txt. Request for Comments 3748, Internet Engineering Task Force, June 2004.

[5] A. Alessandro. *The High Level Protocol Specification Language.* *http://www.avispa-project.org/delivs/2.1/d2-1.pdf*, 2003.

[6] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P. H. Drielsma, O. Héam, P. C.and Kouchnarenko, J. Mantovani, et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *Computer Aided Verification*, pages 281–285. Springer, 2005.

[7] A. Armando, D. Basin, M. Bouallagui, Y. Chevalier, L. Compagna, S. Möder-sheim, M. Rusinowitch, M. Turuani, L. Viganò, and L. Vigneron. The aviss security protocol analysis tool. In *Computer Aided Verification*, pages 349–354. Springer, 2002.

[8] A. Armando and L. Compagna. Sat-based model-checking for security protocols analysis. *International Journal of Information Security*, 7(1):3–32, 2008.

[9] J. W. Atwood. An architecture for secure and accountable multicasting. In *32nd IEEE Conference on Local Computer Networks, 2007. LCN 2007.*, pages 73–78. IEEE, 2007.

[10] A. Basu and J. Young. *IKEv2 Packet Exchange and Protocol Level Debugging.* *http://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/115936-understanding-ikev2-packet-exch-debug.html*, March 2013. Accessed: 2015-09-15.

[11] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Computer Security Foundations Workshop*, page 0082. IEEE, 2001.

[12] G. Bumgardner. Automatic multicast tunneling. `https://www.rfc-editor.org/rfc/rfc7450.txt`. Request for Comments 7450, Internet Engineering Task Force, February 2015.

[13] M. Burrows. *Wide Mouthed Frog.* *http://www.lsv.ens-cachan.fr/Software/spore/wideMouthedFrog.html*, 1989.

[14] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet group management protocol, version 3. `https://www.rfc-editor.org/rfc/rfc3376.txt`. Request for Comments 3376, Internet Engineering Task Force, October 2002.

[15] N. Cam-Winget, D. McGrew, H. Zhou, and J. Salowey. The flexible authentication via secure tunneling extensible authentication protocol method (EAP-FAST). `https://www.rfc-editor.org/rfc/rfc4851.txt`. Request for Comments 4851, Internet Engineering Task Force, May 2007.

[16] I. Cervesato. The dolev-yao intruder is the most powerful attacker. In *Proceedings of the Sixteenth Annual Symposium on Logic in Computer Science | LICS'01*, pages 16–19. IEEE Computer Society Press, 2001.

[17] Y. Chevalier and L. Vigneron. Automated unbounded verification of security protocols. In *Computer Aided Verification*, pages 324–337. Springer, 2002.

[18] S. Deering. Host extensions for ip multicasting. https://www.rfc-editor.org/rfc/rfc1112.txt. Request for Comments 1112, Internet Engineering Task Force, August 1989.

[19] S. Deering, W. Fenner, and B. Haberman. Multicast listener discovery (MLD) for IPv6. https://www.rfc-editor.org/rfc/rfc2710.txt. Request for Comments 2710, Internet Engineering Task Force, October 1999.

[20] G. Denker and J. Millen. Capsl integrated protocol environment. In *DARPA Information Survivability Conference and Exposition*, page 0207. IEEE, 2000.

[21] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[22] W. Fenner. Internet group management protocol, version 2. https://www.rfc-editor.org/rfc/rfc2236.txt. Request for Comments 2236, Internet Engineering Task Force, November 1997.

[23] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin. Protocol for carrying authentication for network access (PANA). https://www.rfc-editor.org/rfc/rfc5191.txt. Request for Comments 5191, Internet Engineering Task Force, May 2008.

[24] D. Harkins and D. Carrel. The internet key exchange (IKE). https://www.rfc-editor.org/rfc/rfc2409.txt. Request for Comments 2409, Internet Engineering Task Force, November 1998.

[25] R. Hinden and S. Deering. IP version 6 addressing architecture. https://www.rfc-editor.org/rfc/rfc4291.txt. Request for Comments 4291, Internet Engineering Task Force, February 2006.

[26] R. Housley, W. Polk, D. Cooper, S. Santesson, S. Farrell, and S. Boeyen. Internet X. 509 public key infrastructure certificate and CRL profile. https://www.rfc-editor.org/rfc/rfc5280.txt. Request for Comments 5280, Internet Engineering Task Force, May 2008.

[27] Cisco Systems Inc. *Overview of IP Multicast.* *http://www.cisco.com/en/US/tech/tk828/technologies_white_paper09186a0080092942.shtml*. Accessed: 2015-03-13.

[28] Cisco Systems Inc. *IP Multicast Technology Overview.* *http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.pdf*, 2002. Accessed: 2015-03-13.

[29] Cisco Systems Inc., editor. *Cisco IOS XR Multicast Configuration Guide for the Cisco CRS Router, Release 4.3.x*, chapter 3. May 2013. Accessed: 2015-03-13.

[30] S. Islam and J. W. Atwood. A framework to add AAA functionalities in IP multicast. In *Telecommunications, 2006. AICT-ICIW'06. International Conference on Internet and Web Applications and Services*, pages 58–58. IEEE, 2006.

[31] S. Islam and J.W. Atwood. Multicast receiver access control using PANA. `http://www.taibahu.edu.sa/iccit/alliccitpapers/pdf/p816-islam.pdf`. In *1st Taibah University International Conference on Computing and Information Technology (ICCIT)*, pages 816–821, 2012.

[32] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen. Internet key exchange protocol version 2 (IKEv2). `https://www.rfc-editor.org/rfc/rfc5996.txt`. Request for Comments 5996, Internet Engineering Task Force, September 2010.

[33] S. Kent. IP authentication header. `https://www.rfc-editor.org/rfc/rfc4302.txt`. Request for Comments 4302, Internet Engineering Task Force, December 2005.

[34] S. Kent. IP encapsulating security payload (ESP). `https://www.rfc-editor.org/rfc/rfc4303.txt`. Request for Comments 4303, Internet Engineering Task Force, December 2005.

[35] S. Kent and K. Seo. Security architecture for the internet protocol. `https://www.rfc-editor.org/rfc/rfc4301.txt`. Request for Comments 4301, Internet Engineering Task Force, December 2005.

[36] T. Kernen and S. Simlo. AMT-automatic ip multicast without explicit tunnels. `http://www.octoshape.com/wp-content/uploads/2012/05/trev_2010-q4_amt_kernen_simlo-1.pdf`, 2010.

[37] B. Li and J. W. Atwood. Receiver access control for ip multicast at the network level. *Submitted to Computer Networks*, November 2015.

[38] 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, 2014.

[39] V. Manral. Cryptographic algorithm implementation requirements for encapsulating security payload (ESP) and authentication header (AH). https://www.rfc-editor.org/rfc/rfc4835.txt. Request for Comments 4835, Internet Engineering Task Force, April 2007.

[40] S. Modersheim and P. H. Drielsma. Automated validation of internet security protocols and applications. http://www.avispa-project.org/, 2003. Accessed: 2015-09-13.

[41] L. Morand, A. Yegin, S. Kumar, and S. Madanapalli. Dhcp options for protocol for carrying authentication for network access (PANA) authentication agents. https://www.rfc-editor.org/rfc/rfc5192.txt. Request for Comments 5192, Internet Engineering Task Force, May 2008.

[42] M. Rouse. *Internet Group Management Protocol (IGMP) definition. http://searchnetworking.techtarget.com/definition/Internet-Group-Management-Protocol*, May 2010. Accessed: 2015-09-15.

[43] A. Salem. Formal validation of security properties of AMT's three-way handshake. Master's thesis, Department of Computer Science and Software Engineering, Concordia University, 2011.

[44] J. Schiller. Cryptographic algorithms for use in the internet key exchange version 2 (IKEv2). https://www.rfc-editor.org/rfc/rfc4307.txt. Request for Comments 4307, Internet Engineering Task Force, December 2005.

[45] M. Schneider. Protocol for carrying authentication for network access (PANA) requirements. http://www.informatik.fh-nuernberg.de/professors/trommler/internet_security/pana_requirements_research_paper.pdf, 2004. Accessed: 2015-02-10.

[46] AVISPA Team et al. *HLPSL Tutorial: A Beginner's Guide to Modelling and Analysing Internet Security Protocols.* *http://www.avispa-project.org/package/tutorial.pdf*, 2006.

[47] R. Vida, L. Costa, S. Fdida, S. Deering, B. Fenner, I. Kouvelas, and B. Haberman. Multicast listener discovery version 2 (MLDv2) for IPv6. `https://www.rfc-editor.org/rfc/rfc3810.txt`. Request for Comments 3810, Internet Engineering Task Force, June 2004.

[48] L. Viganò. Automated security protocol analysis with the AVISPA tool. *Electronic Notes in Theoretical Computer Science*, 155:61–86, 2006.

[49] B. Weis, G. Gross, and D. Ignjatic. Multicast extensions to the security architecture for the internet protocol. `https://www.rfc-editor.org/rfc/rfc5374.txt`. Request for Comments 5374, Internet Engineering Task Force, November 2008.

[50] Wikipedia. *Verisign.* *https://en.wikipedia.org/wiki/Verisign*. Accessed: 2015-08-26.

[51] A. Yegin, Y. Ohba, R. Penno, G. Tsirtsis, and C. Wang. Protocol for carrying authentication for network access (PANA) requirements. `https://www.rfc-editor.org/rfc/rfc4058.txt`. Request for Comments 4058, Internet Engineering Task Force, May 2005.