SECURITY ISSUES IN PIM-SM LINK-LOCAL MESSAGES

Salekul Islam

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Computer Science

Concordia University

Montréal, Québec, Canada

December 2003

Your file Votre référence
ISBN: 0-612-91049-0
Our file Notre référence
ISBN: 0-612-91049-0

Abstract

Security Issues in PIM-SM Link-local Messages

Salekul Islam

Protocol Independent Multicast-Sparse Mode (PIM-SM) routing protocol attracts most of the attention of the Internet community due to its scalability and flexibility. From the very beginning, multicast communication faced various difficulties in its security areas. PIM-SM is also not free from this problem. Security features of a routing protocol consist of two orthogonal planes: data plane and control message plane. The first one ensures distribution of data packets securely while the other deals with security of control messages.

Most of the PIM-SM control messages fall into the *link-local* category, and are sent to adjacent routers only, using TTL = 1 and ALL_PIM_ROUTERS as destination address. To protect these link-local messages, in the present Internet Draft of PIM-SM a security mechanism has been proposed that uses IPsec Authentication Header (AH) protocol. While using IPsec AH protocol, the anti-replay mechanism has been disabled. This compromise makes PIM-SM vulnerable to denial of service attack. Moreover, the Security Association lookup and number of Security Associations are also erroneous and incomplete in the document.

A new proposal has been presented in this thesis to protect PIM link-local messages while activating the anti-replay mechanism as well. Security Association lookup method has been modified also to cope with this proposal. Finally, this new proposal has been validated using a validation tool, SPIN, that uses PROMELA to design the validation model.

Acknowledgements

First and foremost, I am expressing my solemn gratitude to Almighty Allah, most gracious and most merciful. Without His help and kindness, I could not have finished my thesis.

I feel so fortunate that from the beginning of my research work I was under the supervision of Dr. J.W. Atwood. His vast experience, immense knowledge and friendly attitude directed me to achieve my research goal. In fact, I discovered the whole new world of academic research in the last few years. I am also grateful to him for the financial support he provided to me. A special thanks to my former supervisor, Dr. T.D. Bui, who funded me for a certain period of time.

I would like to thank the faculty members, staff, system analysts and my fellow graduate students of the Computer Science Department at Concordia University. Special thanks for one of my colleagues, Mr. Xingguo Song, who helped me a lot in using LaTeX.

I am forever indebted to my parents, sister and other family members. Finally, I must mention my soul mate and wife Sarah, who supported and encouraged me to complete this work.

Contents

List of Figures

List of Tables

List of Acronyms

AH .....Authentication Header

DoS .....Denial of Service

DR .....Designated Router

ESN .....Extended Sequence Number

ESP .....Encapsulating Security Payload

GCKS .....Group Controller and Key Server

GDOI .....Group Domain of Interpretation

ICV .....Integrity Check Value

ID .....Internet Draft

IETF .....Internet Engineering Task Force

IGMP .....Internet Group Management Protocol

IKE .....Internet Key Exchange

IPsec .....IP security

MRIB .....Multicast Routing Information Base

MSEC .....Multicast Security

PIM-SM .....Protocol Independent Multicast - Sparse Mode

PROMELA .....PROcess MEta LAnguage

RFC .....Request For Comment

RP .....Rendezvous Point

RPF .....Reverse Path Forwarding

RPT .....Rendezvous Point Tree

SA .....Security Association

SAD .....Security Association Database

SG .....Security Gateway

SPD .....Security Policy Database

SPI .....Security Parameter Index

SPIN .....Simple PROMELA INterpreter

SPT .....Shortest Path Tree

TTL .....Time To Live

WG .....Working Group

Chapter 1

Introduction

The Internet is the world's largest network and it has displayed remarkable flexibility as it has evolved from a research-oriented network to one with a number of commercial applications. The present Internet is facing a series of serious challenges that were non-existent at the beginning. In the Internet, the number of users as well as the number of various applications are growing exponentially. Many applications, previously available only to a limited number of power users with high-end workstations, are starting to become mainstream applications in the PC world. Videoconferencing, video broadcasting, collaborative applications, etc. are very common applications nowadays. Many of the new applications rely on one-to-many or many-to-many communications, where one or more sources are sending data to multiple receivers. It is possible to provide transmissions to multiple receivers in three different ways — unicast, broadcast, and multicast.

In unicast communication, a separate copy of data is delivered to each recipient. In such cases, the number of receivers is limited by the sender's bandwidth and if the number of receivers is large, a huge bandwidth is wasted. Transferring a file from an FTP (File Transfer Protocol) file server to a host computer is an example of unicasting.

If ten different users want to download the same file from an FTP file server, the server would have to send the file to each of the ten recipients separately, using ten times as much bandwidth as a single file transfer.

Broadcast communication forwards a data packet to all portions of the network even if only a few of the destinations are interested to receive it. The definite advantage for the sender is that the sender transmits a single copy of the packet to the appropriate broadcast address and the network devices such as routers and switches duplicate the packet as needed to cover the network. For example, in a WAN (Wide Area Network) broadcasting is often used for maintaining or diagnosing the state of the inter-network. In broadcast, the message is sent to all the workstations or to the host computers whether they are intended recipients or not. This creates an unwanted computational burden for the host computers, since they have to process at least part of the message to determine if it is something of interest.

Multicasting falls between unicasting and broadcasting. Rather than sending data to a single host (unicast) or all hosts in a network (broadcast), multicasting delivers data only to all intended recipients. A group of host computers wishing to receive multicast data, create a multicast group first. This type of group is called a *host group* and is defined by a specific multicast address. Once a host group is set up and the sender starts transmitting packets, the underlying network takes the responsibility for delivering the packets to all members who have already joined. Only one copy of a multicast packet passes over any link in the network. When the path is divided at a router, multiple copies of the packet are replicated by the router for different paths. This helps to conserve bandwidth [18, 27].

1.1 IP Multicasting

The extensions required of a host implementation of the Internet Protocol (IP) to support multicasting were first specified by Stephen Deering in RFC1112 [7]. As this is not simple peer-to-peer communication like unicast, we have to consider m-to-n communication in multicasting. To implement multicasting successfully, we can identify four major steps or processes responsible for the whole complex tasks. Figure 1 describes the schematic diagram of these four processes.

1. At first, a multicast host group that has a multicast address (Class D Address) should be created by the group owner and this group address should be announced throughout to the potential receivers.

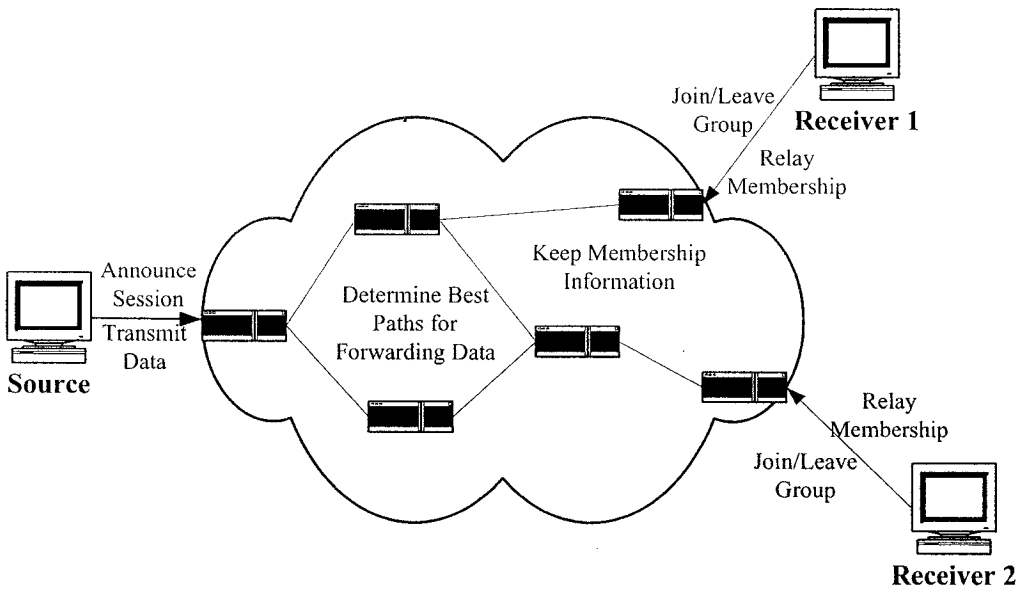


Figure 1: Basic Components of IP Multicasting

2. There should be some mechanism through which the end receivers/hosts will join/leave the multicast group. This membership information should be updated from time to time.

3. The third component is the multicast routing protocol. A series of these types of protocols have been specially formulated to support multicasting.

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

1.1.1 Multicast Address

IPv4 addresses are divided into five major classes, from A to E. The first three classes, A, B and C support unicast communication while class D supports multicast communication. The last one, class E is reserved for experimental use. Each address consists of four octets, or a set of eight binary digits. They are separated by decimals (.) in the customary notation or dotted decimal notation. Table 1 summarizes five different classes.

Table 1: Different Classes of IP Address

Class	Range	Type
A	0.0.0.0 to 127.255.255.255	Unicast
B	128.0.0.0 to 191.255.255.255	Unicast
C	192.0.0.0 to 223.255.255.255	Unicast
D	224.0.0.0 to 239.255.255.255	Multicast
E	240.0.0.0 to 247.255.255.255	Experimental

The first step in setting up a multicast session is the selection of a host group address. First, the initiator, or owner of the group of a multicast session has to select a destination address for the multicast data from the available IPv4 Class D (or equivalent IPv6) address. This destination address corresponds to an appropriate host group. The source host can then begin transmitting data packets on the internetwork using that group host address as the destination. Receivers of a multicast group must be aware of the group

address being used and joined to that group. Presently, there is no standard procedure or application for maintaining a list of multicast group addresses. Some mechanism such as offline group's TV guide, email, or maintaining a secure website can be used to inform all potential group members of the host group's address.

1.1.2 Internet Group Management Protocol

Internet Group Management Protocol (IGMP) [6] is the protocol used by IPv4 systems to report their IP multicast group memberships to neighboring multicast routers. IGMP runs between hosts and their immediate neighboring multicast routers. If there is more than one IP multicast router on the LAN, one of the routers is elected as "interrogator". It is responsible for sending queries throughout the LAN for the presence of any group member. A host wishing to join a specific multicast group can either join explicitly by sending an IGMP Join message or waiting for an IGMP Query message, and then responding to this query by sending a Join message. Based on the group membership information learned via IGMP, a router is able to determine whether any specific traffic should be forwarded or not to the router's subnet.

1.1.3 Multicast Routing Protocols

Multicast routing protocols are not as simple and straightforward as unicast routing protocols. The multicast routers have to respond to changes in network topology and in group membership. Moreover, they need to do this in a timely fashion, not wasting bandwidth by sending unwanted data to resigned members or failing to forward multicast traffic to new members. No single multicast routing protocol can satisfy all the necessary criteria. Multicast routing protocols can be divided into two categories depending on the relative distance of the receivers.

At first, it was assumed that the group members were densely distributed, and that we have plentiful bandwidth to use. In such scenarios, it is very reasonable to use flooding to broadcast data throughout the network. Routers, not wishing to receive data from a particular source, can build a source specific prune state and forward it towards an upstream source. That means all the routers will receive data first and only a few will send prune message towards the upstream source. This is quite reasonable as we have assumed at the beginning that receivers are densely distributed over the network. Examples of such protocols are Distance Vector Multicast Routing Protocol (DVMRP) [25], Multicast Extensions to Open Shortest Path First (MOSPF) [21] and Protocol Independent Multicast-Dense Mode (PIM-DM) [1].

The recent additions to the set of multicast routing protocols are called *sparse mode* protocols. These routing protocols are designed to operate efficiently over a wide area network where bandwidth is scarce and group members may be distributed quite sparsely. Sparse does not necessarily assume small group, rather it is meant to convey that the group members are widely dispersed. For this reason, these protocols are more concerned in preserving the bandwidth. The basic difference between sparse mode and dense mode protocols is that the sparse mode protocols do not flood group data across the entire network. In such a case, the receivers join explicitly by sending a join message towards the distribution tree. Sparse mode protocols include Protocol Independent Multicast-Sparse Mode (PIM-SM) [8] and Core-Based Trees (CBT) [2].

Though it is clear that a single protocol cannot fulfil all types of requirements in different situations, among all the protocols PIM-SM is the most cited and in the core of research of multicast community. It is probably the most used protocol because of its scalability features.

1.2 Motivation of My Thesis

At the early days of the Internet the number of users was very low and its use was limited. At present, the Internet carries heterogeneous traffic ranging from voice and video to various types of data. As a result the Internet is not a safe place for communication particularly for multicast communication, which is vulnerable to different types of attacks. Among the multicast protocols PIM-SM is used most widely and it attracts the attention of Internet community due to its scalability. If we expect its large deployment, we certainly have to be cautious about its security issues.

In the present Internet Draft (ID) of PIM-SM [8] issued by the Internet Engineering Task Force (IETF), the authors have added a different section to discuss security issues. Different security techniques have been proposed for different types of control messages including link-local messages. The messages that are sent to neighbor routers (with TTL=1) within the same domain are called link-local messages. These messages play an important role in building the shared tree through which multicast data is forwarded. We have found that the proposed mechanism to secure these link messages in the present PIM-SM ID is incomplete and impossible to implement. Consequently, we have proposed our own mechanism to create secure link-local messages.

1.3 Thesis Organization

This thesis is composed of six chapters. The first three chapters are meant to develop the reader's background knowledge on PIM-SM protocol and IP security. In the fourth chapter we describe our own proposal, and in the fifth chapter we present our validation and finally we have drawn conclusion of our thesis in the last chapter.

Chapter 1 starts by introducing some basic concepts of IP multicasting. Then we describe the goal or purpose of our thesis.

Chapter 2 illustrates how PIM-SM protocol works and its different phases. After that, we explain different link-local messages. We also add explanations on the effects of forged link-local messages.

Chapter 3 presents an overview of IP security. We explain IP Security (IPsec) architecture first. We cover the definition of Security Association (SA), different required databases to maintain an SA, and SA lookup mechanism. As we are going to use Authentication Header (AH) protocol later, we explain the Authentication Header packet format and anti-replay mechanism of this protocol. Finally, we explain what changes for AH protocol have been proposed to use it in multicasting.

Chapter 4 is divided into three sections. In the first section, present security features of PIM-SM link-local messages are listed. We identify the limitations of the proposal presently existing. Next, we add another proposal that uses Group Domain of Interpretation to solve the same issue. We have pointed out some limitations of this concept also. Finally, we have presented our own proposal and explained its different features in detail.

Chapter 5 describes the validation method of our own proposal. We use a formal validation language, PROMELA, to develop our validation model and a generic validation tool, SPIN, to validate this model.

Chapter 6 concludes our thesis. We list our major contributions and add some notes for future research.

Chapter 2

PIM-SM Protocol

Protocol Independent Multicast-Sparse Mode (PIM-SM) [8] efficiently routes multicast data for a group that may span a wide area and be inter-domain in the Internet. It is based on the assumption that group members are likely to be located far away from each other. The available bandwidth tends to be small, and members are available only in some of the subnetworks involved. It is considered as being protocol-independent as it has been designed to be used with any available unicast routing protocol. It can use either the underlying unicast routing information base or a separate Multicast Routing Information Base (MRIB).

PIM-SM builds unidirectional shared trees rooted at a Rendezvous Point (RP) per group and finally switches to the shortest-path trees per source. Group members have to join explicitly by sending joining report towards RP. The data are then routed through the RP to the domain of the newly joined member. For this reason, PIM-SM is considered one of the most optimized and scalable multicast routing protocols. The Following criteria are established for it [27]:

- Minimize status information in routers

- Minimize number of control packets and user data
- Minimize bandwidth consumption in the network

2.1 How it Works

For multicast protocol, it is a necessary condition that data packets should be routed from sources to receivers without either the sources or receivers knowing beforehand about the existence of the others. Another assumption is that a multicast group may be dynamic in nature. That means senders and receivers can join/leave any time to/from the group. PIM depends on the routing table that is presently stored in the MRIB. Regardless of how this routing table is created, the primary function of MRIB is to provide the next hop router along a multicast-capable path to each destination subnet. In this way, MRIB determines the path along which a Join/Prune message will be forwarded. Data are forwarded in the reverse direction of the Join/Prune message. Thus the MRIB provides reverse-path information and indicates the path that a multicast data packet would take from its origin subnet. This mechanism is called 'Reverse Path Forwarding' or RPF.

PIM-SM is done in three phases and these phases may occur simultaneously. Next, we shall explain these three phases in brief.

2.1.1 Phase One: RP Tree

In each subnet, one of the local routers is elected as Designated Router (DR) and it is responsible for forwarding/receiving any packet to/from the subnet. Within a subnet, all the host computers communicate with the DR using the Internet Group Management Protocol [6] or any other similar protocol. At first, a potential group member will send a

join message to its local DR using IGMP. The DR then sends a PIM Join message, (*,G) towards RP. This is the Join message for all sources of group G. It travels hop-by-hop towards the RP of that group and creates multicast tree state for group G in each router it passes through. Eventually, this Join message either reaches the RP or a router that already has (*,G) Join state for group G. When more than one Join messages reach an RP, they all create RP Tree (RPT) or shared tree. Join messages are re-sent periodically to refresh join state. When all receivers on a leaf-network leave the group, the DR sends a PIM (*,G) Prune message towards the RP of that group.

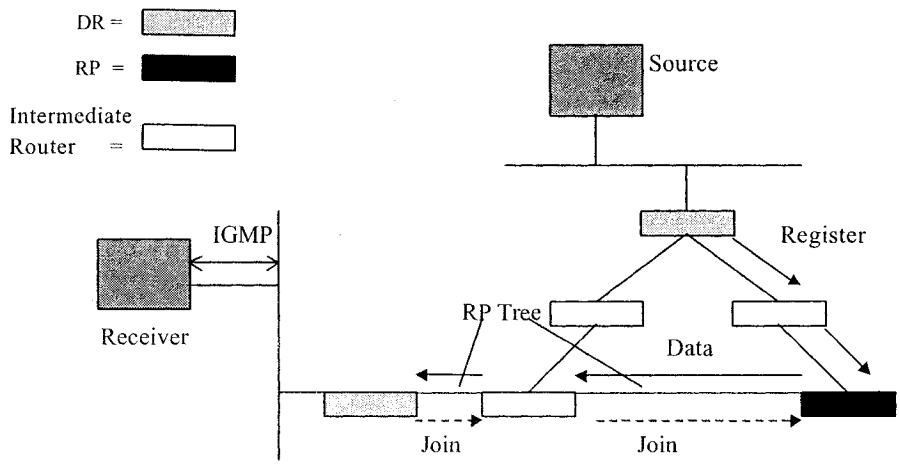


Figure 2: PIM-SM Routing RP Tress is formed

A multicast sender transmits data to its local DR first. The DR receives this packet and unicast-encapsulate it and sends it directly to the RP of that group. The RP receives this packet, decapsulates it and forwards it onto the RPT or shared tree. The packet then follows the (*,G) multicast tree state in the routers and finally reaches all the receivers who have already joined to that group. The way the sender sends data packets towards RP is known as 'registering' and these packets are called 'PIM Register Packets'.

At the end of phase one, we can conclude that multicast traffic is flowing in encapsulated form towards the RP, and the RP is forwarding the decapsulated data using the

RPT to all the receivers.

2.1.2 Phase Two: Register Stop

Register-encapsulation process in phase one is very inefficient for the following two reasons:

- Encapsulation and decapsulation are very costly, and for each packet routers have to go through these processes. Occasionally, a router may not have hardware support to make this efficient.
- A data packet is sent to the RP first, and then the RP forwards it to all the receivers. For a specific receiver, there may be a shorter path coming directly from the sender.

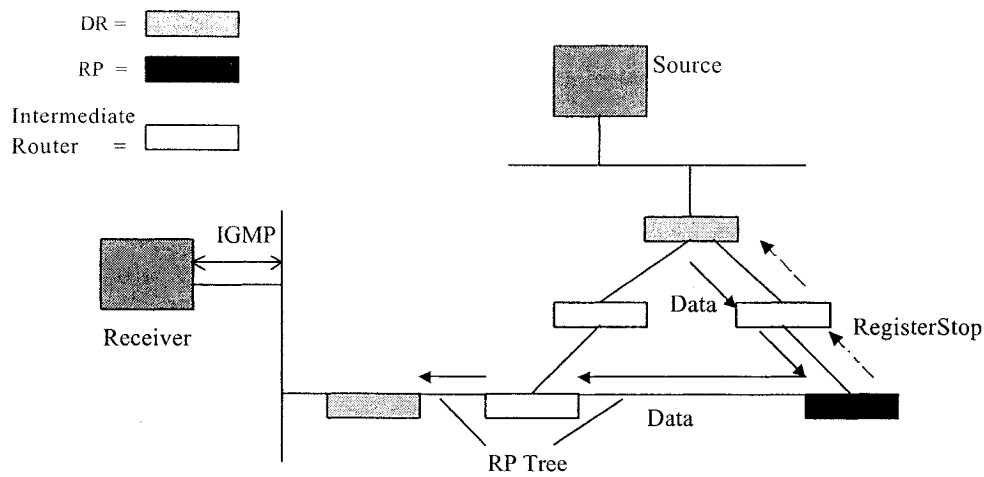


Figure 3: PIM-SM Routing RegisterStop message is sent

At this moment the RP will send a source specific Join message, (S,G) towards source, S. (S,G) stands for a source specific Join message to group G in which the joiner wants to receive data only from source S. This message is transmitted hop-by-hop, and creates

(S,G) multicast state in each router and ultimately reaches the S's subnet, a router that has already (S,G) multicast state. Now, S will send data without encapsulation through (S,G) states towards RP. These data packets may reach routers with (*,G) state on their way towards RP, and they can short-cut onto the RP tree at this point.

The RP will receive two copies of data from source S; one is in encapsulated form, and the other is through (S,G) multicast state, and this copy is not encapsulated. So, the RP will discard the encapsulated packet, and send a RegisterStop message to the DR of source S not to send the encapsulated packet any more. Upon receiving the RegisterStop message, the DR of S will stop sending unnecessary encapsulated packets.

At the end of this phase, data will flow from source to RP through source specific tree and from RP to receivers through shared tree or RPT. If these two trees interact at any point, traffic may transfer from source specific tree to RPT. Moreover, phase one and phase two may occur simultaneously, and even phase two may start before phase one. In that case sender may start sending before or after receivers join.

2.1.3 Phase Three: Switching to SPT

So far, the encapsulation overhead has been removed but still it does not completely optimize the forwarding paths. For many receivers there may be an existing shorter path directly from the source rather than via the RP. Therefore, a receiver may switch to a source specific Shortest Path Tree or SPT. To do this, the DR of a receiver will send an (S,G) Join message towards source S. This Join message will traverse hop-by-hop, create (S,G) multicast state, and finally will reach to S's subnet or to a router that has (S,G) state.

At this point, the receiver will receive two copies of data, one from SPT and the other from RPT. Just after receiving the first packet from SPT, the receiver will send

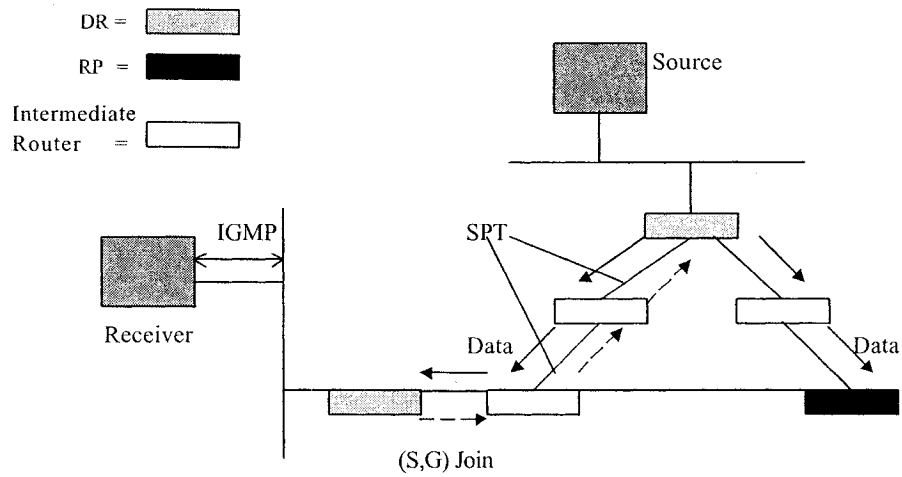


Figure 4: PIM-SM Routing switching from RP-tree to SPT

an (S,G) Prune towards RP. This message is known as (S, G, rpt) Prune. This message will traverse towards RP, instantiating state along the path indicating that traffic from S for G should not be forwarded in this direction. The Prune will propagate until it reaches the RP or other router that needs traffic from S for other receiver(s).

As far as the receivers are concerned, this is the final state as they are receiving data through SPT and not through RPT. Still RP is receiving data from S, but this traffic is not forwarded to receivers through RPT.

2.2 Link-local Messages

PIM-SM control messages are always sent in encapsulated form within IP packets. The following figure shows how we can encapsulate such a control message inside an IP packet [20].

All PIM control messages have IP protocol number 103. These messages are either unicast (e.g., Register and RegisterStop), or multicast with TTL = 1. The source

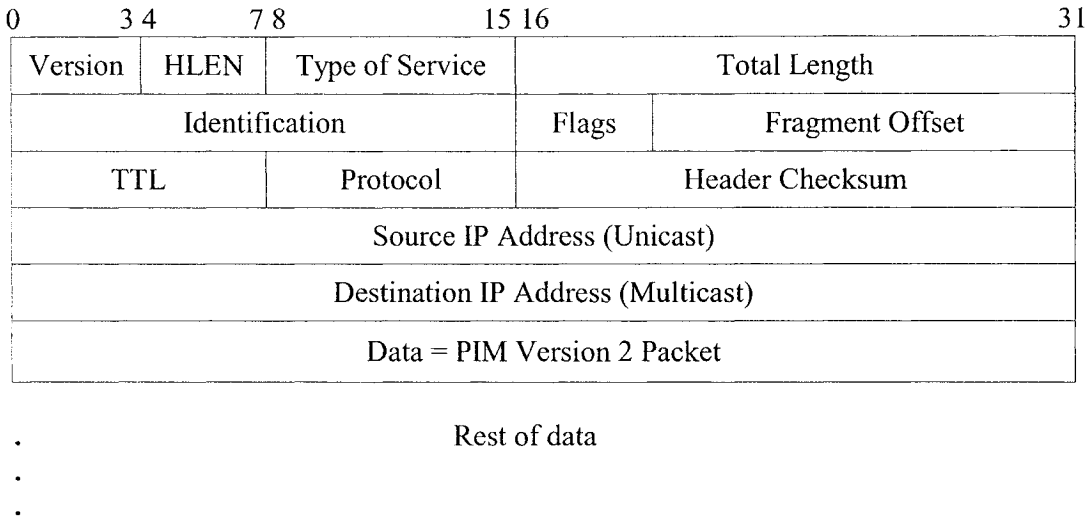


Figure 5: Encapsulated PIM Control Message

address used for unicast messages is a domain-wide reachable address. While multicast, a link-local address of the interface on which the message is being sent is used as source address and a special multicast address ALL_PIM_ROUTERS is used as destination address. ALL_PIM_ROUTERS is a fixed multicast address and specified as 224.0.0.13 in IPv4 and ff02::d in IPv6. From the above figure it is clear that, a PIM-SM control message is appended in the data part of an IP packet. All the PIM-SM control messages have two section. The first section is a general header format and the rest is the actual body of the control message. PIM-SM general header format is shown in Figure 6.

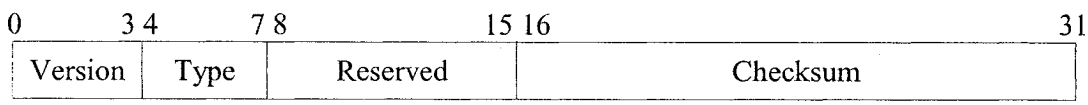


Figure 6: PIM-SM version 2 Packet Header

The fields in the header have the following meanings:

- *Version* is the PIM version number. For version 2, the value is 2

- *Type* is the value associated with the particular control message (see Table 2 below).
- *Reserved* is transmitted as 0. It is ignored upon receipt.
- *Checksum* is the 16-bit one's complement of the one's complement sum of the entire PIM message (excluding the data portion in the Register message).

Table 2: Different types of PIM-SM version 2 Messages

Type	Description	Destination
0	Hello	Multicast to ALL_PIM_ROUTERS
1	Register	Unicast to RP
2	RegisterStop	Unicast to source of Register packet
3	Join/Prune	Multicast to ALL_PIM_ROUTERS
4	Bootstrap	Multicast to ALL_PIM_ROUTERS
5	Assert	Multicast to ALL_PIM_ROUTERS
8	Candidate-RP-Advertisement	Unicast to Domain's BSR

If a PIM-SM control message is sent to the ALL_PIM_ROUTERS destination address, the message is called as *"link-local"* message. This is called link-local as the address of the interface on which the message is sent is used as source address. These messages are sent with TTL=1 and thus are not forwarded by a compliant router. *Hello*, *Join/Prune* and *Assert* are included in this category. Next, we shall discuss each of these link-local messages in brief.

2.2.1 Hello Message

PIM Hello messages are sent periodically on each PIM-enabled interface. They allow a router to learn about the neighboring PIM routers on each interface. Hello messages

are also the mechanism used to elect a Designated Router (DR), and to negotiate additional capabilities. A router must record the Hello information received from each PIM neighbor. Hello messages must be sent on all active interfaces, including physical point-to-point links, and are multicast to address 224.0.0.13 (the ALL_PIM_ROUTERS group). A shared-media LAN such as an Ethernet may have multiple PIM-SM routers connected to it. If the LAN has directly connected hosts, then a single one of these routers, the DR, will act on behalf of those hosts with respect to the PIM-SM protocol. DR election is performed using Hello messages.

2.2.2 Join/Prune Message

A PIM Join/Prune message consists of a list of groups and a list of Joined and Pruned sources for each group. PIM-SM routers do not send Join/Prune messages on a per group basis. All the outstanding Join/Prune messages are accumulated and sent by one message to the upstream router. In general, a PIM Join/Prune message should only be accepted for processing if it comes from a known PIM neighbor. A PIM router hears about PIM neighbors through PIM Hello messages. If a router receives a Join/Prune message from a particular IP source address and it has not seen a PIM Hello message from that source address, then the Join/Prune message should be discarded without further processing. In addition, if the Hello message from a neighbor was authenticated using IPsec AH then all Join/Prune messages from that neighbor must also be authenticated using IPsec AH.

2.2.3 Assert Message

Where multiple PIM routers peer over a shared LAN it is possible for more than one upstream router to have valid forwarding state for a packet, which can lead to packet duplication. PIM does not attempt to prevent this from occurring. Instead it detects

when this has happened and elects a single forwarder amongst the upstream routers to prevent further duplication. This election is performed using PIM Assert messages. Assert messages are also received by downstream routers on the LAN, and these cause subsequent Join/Prune messages to be sent to the upstream router that won the Assert. In general, a PIM Assert message should only be accepted for processing if it comes from a known PIM neighbor. If the Hello message from a neighbor was authenticated using IPsec AH then all Assert messages from that neighbor must also be authenticated using IPsec AH.

2.3 Effects of Forged Link-local Messages

A forged link-local message is sent to the ALL_PIM_ROUTERS multicast address by an attacker. A forged message can reach a LAN, if it were sent by a local host, or allowed onto the LAN by a compromised router. This type of message affects the construction of the distribution tree. These effects vary for different types of forged messages. Some of the effects are very severe whereas some are minor. Following are the effects of different forged link-local messages:

1. A forged Join message allows a non-member host to receive group data. In case there is no other legitimate group member on that LAN, potential bandwidth will be wasted. Certainly, this type of forged message creates a threat for a closed multicast group where only the legalized members are permitted to join the group.

2. Normally, a forged Prune message has no significant effect on a multi-access LAN because any legitimately joined router on the LAN would overwrite the Prune message with a Join message before the upstream router stops forwarding data to the LAN.

3. Hello messages are used to elect the Designated Router (DR). For this reason,

by forging a Hello message, an unauthorized router can cause itself to be elected as the DR on a LAN. The DR on a LAN plays an important role by forwarding traffic to that LAN on behalf of any local members. The DR is also responsible for sending register packet towards RP while the sender is on that LAN. Thus, by forging Hello message successfully, an attacker can prevent local hosts from sending and receiving multicast traffic.

4. By forging an Assert message on a multi-access LAN, an attacker could cause the legitimate designated forwarder to stop forwarding traffic to the LAN. Such a forgery would prevent any hosts downstream of that LAN from receiving traffic.

Chapter 3

IP Security (IPsec)

The goal of IP Security (IPsec) architecture is to provide various security services for traffic at the IP layer in both the IPv4 and IPv6 environments. The basic component of IPsec architecture, the goals of such systems and how they fit together with each other and into the IP environment, are described in RFC2401 [17]. This document also describes the security services offered by the IPsec protocols, and how these services can be employed in the IP environment. IPsec can be used to protect one or more paths between a pair of hosts, between a pair of security gateways (SG), or between a host and a security gateway. By security gateway we refer to an intermediate system that implements IPsec protocols; for example, a router or a firewall. By host, we refer to an end system such as a personal computer connected to the Internet that implements IPsec protocols. IPsec is always deployed in the IP layer and does not really effect application layer at all. When these mechanisms are correctly in place, they ought not to adversely affect users, hosts, and other Internet components that do not employ these security mechanisms for protection of their traffic. The whole thing is designed so flexibly that different users may select different cryptographic authentication algorithms if required.

IPsec provides a number of essential services and an implementer is free to implement

either all of them or a subset of them according to his requirements. Such services are listed in the following:

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets, a form of partial sequence integrity
- Confidentiality or encryption
- Limited traffic flow confidentiality

To provide these traffic security IPsec uses two different protocols. Both of these protocols offer a unique security service as well as basic security services. As a result, these protocols may be implemented alone or in combination with each other to meet a desired set of security requirements.

1. The IP **Authentication Header (AH)** [15] protocol provides data origin authentication and an optional anti-replay service along with other security services. Consequently, if we want to activate the anti-replay mechanism and to make sure that the received packet was originated from its sender we should implement the AH protocol.
2. The **Encapsulating Security Payload (ESP)** [16] protocol provides encryption or data confidentiality along with other security services. So, if we want to send data in encrypted form, we must choose ESP protocol.

As we have described it so far, IPsec provides security between a sender and a receiver (either host or security gateway). To ensure these security services, we must have some

mechanisms to supply encryption keys to both sender and receiver, and certainly we may have to update these keys from time to time if required. We have two options here. First, we can use manual key configuration, where key distribution and update are accomplished manually in person by the system administrator. Second, we have some automated key management mechanisms such as Internet Key Exchange (IKE) [10] and Internet Security Association and Key Management Protocol (ISAKMP) [19]. At this point, it is clear that IPsec is a very complex mechanism which consists of different co-related pieces. In the following sections of this chapter, we shall focus in IPsec architecture first, and then our main interest will be to explore how Authentication Header (AH) protocol works, and how it can be used in multicasting communication.

3.1 IP Security (IPsec) Architecture

The concept of "Security Association" (SA) is fundamental to IPsec. Both AH and ESP make use of SAs, and a major function of IKE is the establishment and maintenance of Security Associations. A Security Association is a simplex connection that affords security services to the traffic carried by this connection. To offer security services, we have to implement one of the protocols AH or ESP. To implement both AH and ESP together we need two different SAs. Again, as we have mentioned, SA is a simplex connection; to establish a bi-directional connection between sender and receiver, we need two different SAs, one for each direction.

An SA is uniquely identified by the following three parameters:

1. The Security Parameter Index (SPI) is an arbitrary 32-bit value that is used by a receiver to identify the SA to which an incoming packaet is bound. In unicast communication the SPI is generated by the receiver. The SPI value of zero (0) is reserved for local, implementation-specific use.

2. In conjunction with SPI, the destination address is used to distinguish a specific SA. The destination address may be a unicast address, an IP broadcast address, or a multicast group address. Although IPsec SA management mechanisms are currently defined only for point-to-point (unicast) communication, the concept is also applicable in the point-to-multipoint communication.

3. The third parameter is a security protocol (AH or ESP) identifier that discerns which protocol (AH or ESP) is being used.

There are two types of SAs: transport mode and tunnel mode. A transport mode SA is always established between two hosts whereas a tunnel mode SA is applied to an IP tunnel. Whenever either end of a Security Association is a security gateway, the SA must be in tunnel mode. Thus, an SA between two security gateways or between a host and a security gateway is always a tunnel mode SA.

3.1.1 Security Association Databases

Many of the details associated with processing IP traffic in an IPsec implementation are largely a local matter. However, some external aspects of the processing must be standardized to ensure inter-operability. There are two nominal databases in the IPsec model: the Security Policy Database (SPD) and the Security Association Database (SAD). The SPD specifies the policies that determine the disposition of all IP traffic inbound or outbound from a host or a security gateway. The SAD contains parameters that are associated with each active SA. Another vital concept is that Selectors such as a set of IP and upper layer protocol field values are used by the SPD to map traffic to a policy, i.e., an SA.

3.1.1.1 Security Policy Database (SPD)

In Security Association management, the underlying Security Policy Database plays a vital role by specifying what services are to be offered to IP datagrams and in what fashion. The form of the database and its interface are local issues. However, the SPD must be consulted during the processing of all traffic (inbound and outbound), including non-IPsec traffic. The SPD contains an ordered list of policy entries which are also known as SPD rules [9]. Each rule consists of one or more Selectors, which distinguish among the packets, and an action to be applied. Three possible actions can result from the application of SPD rules:

- **Discard the packet**. All the unsecured packets are prohibited from being sent or received.
- **Bypass the packet without IPsec processing**. A host or SG may bypass some types of packets to be sent or received without IPsec processing.
- **Apply IPsec processing to the packet**. If IPsec protection is required for a packet, the SPD specifies the details of processing such as the IPsec header(s) to be applied, the cryptographic algorithms to be used, the encapsulation mode, and so on. Each outbound SPD rule points to all SAs in the SAD that have been negotiated to satisfy the rule. More than one SPD rule may have to be applied to a single inbound packet.

3.1.1.2 Selectors

An SA may be fine-grained or coarse-grained, depending on the Selectors used to define the set of traffic for the SA. For example, all the traffic between two hosts may be carried through a single SA or may be spread over multiple SAs, depending on the applications being used. Some Selector parameters that must be used are destination and source

IP addresses, data sensitivity level, transport layer protocol and source and destination ports.

3.1.1.3 Security Association Database (SAD)

For each active SA, there should be an entry in the SAD that defines the parameters associated with that SA. During outbound processing, entries are pointed to by the rules (entries) in the SPD. If an SPD rule does not currently point to an SA, the implementation creates an appropriate SA and links the SPD rule to the SAD entry. For inbound processing, each entry in the SAD is indexed by three parameters, IP destination address, SPI and IPsec protocol identifier.

3.1.2 Example of SA Databases

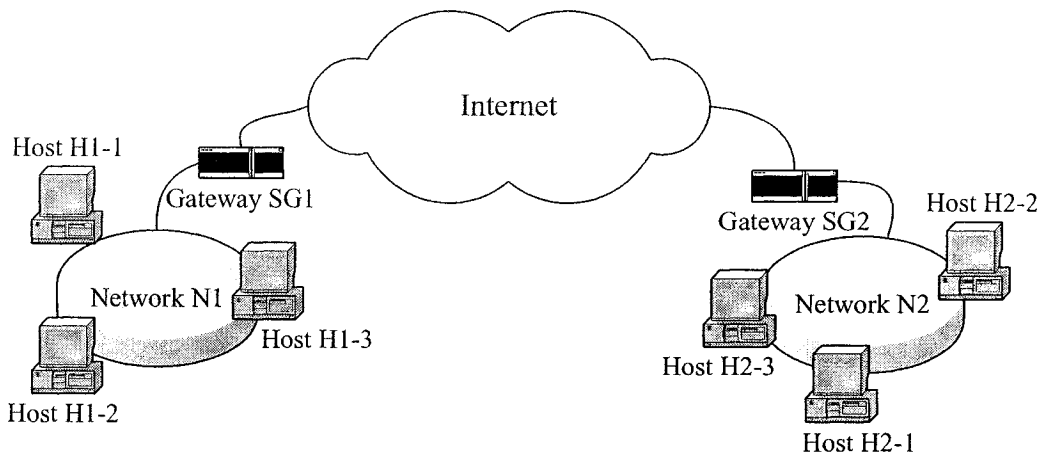


Figure 7: Communication Scenario

To explain how different databases of an SA are maintained, we shall consider a simple example of small-scale Virtual Private Networks (VPN) [9]. According to Figure 7, we have two separate networks, each protected by a security gateway that screens all communications to and from its associated network. This topology can represent a

single business with several branch locations or with separate departmental networks in the same location.

The following table demonstrates the SPD rules that might govern communications between the hosts on Networks N1 and N2 and between the security gateways (SG1 and SG2) themselves. This sample SPD could be either SG1's outbound SPD or SG2's inbound SPD. The selectors used are the source and destination addresses, the source and destination ports, and the protocols. If IPsec protection is to be applied, each rule specifies the IPsec header, encryption and authentication algorithms, and transport mode. For example, rule 1 allows IKE packets, which customarily are sent on port 500, to be sent or received without any IPsec protection. For supersecure host H1-1, rule 3 ensures that all its communication must be encrypted with AES and authenticated with MD5.

Table 3: Sample SPD Rules for a Security Gateway

Rule No	Src Addr	Dest Addr	Src Port	Dest Port	Prot	Action	IPsec Hdr	Enc Alg	Auth Alg	Mode
1	SG1	SG2	500	500	Any	Accept	-	-	-	-
2	SG1	SG2	Any	Any	Any	IPsec	AH	-	MD5	Tunnel
3	H1-1	N2	Any	Any	Any	IPsec	ESP	AES	MD5	Tunnel
4	N1	N2	Any	Any	Any	IPsec	ESP	3DES	MD5	Tunnel

The relationship between SPD rules and SAs is not necessarily a one-to-one relationship. A single SPD rule can spawn multiple SAs. If each of the rule's selectors has a single value, then only one SA is negotiated for that rule. However, if any of the rule's selectors is a wild card or a range then multiple SAs result from that single rule. For example, in our scenario, security gateways SG1 and SG2 each negotiate SAs on behalf of multiple machines. In Table 3, rule 3 covers all communications between host H1-1 on network N1 and any host on network N2. The gateways can satisfy that rule by negotiating a single SA to protect all traffic between H1-1 and network N2. Alternatively,

they can negotiate one SA for each pair of protected hosts. The later approach will result in three different SAs attached to a single SPD rule. The following table shows the SAs resulting from the one-SA-per-host-pair approach.

Table 4: SAs Generated from an SPD Rule: one SA per Host Pair

SA No	Src Addr	Dest Addr	Src Port	Dest Port	Prot	IPsec Hdr	Enc Alg	Auth Alg	Mode
1	H1-1	H2-1	Any	Any	Any	ESP	AES	MD5	Tunnel
2	H1-1	H2-2	Any	Any	Any	ESP	AES	MD5	Tunnel
3	H1-1	H2-3	Any	Any	Any	ESP	AES	MD5	Tunnel

3.2 Authentication Header (AH) Protocol

IPsec implementation is dependent on two protocols: Authentication Header (AH) protocol and Encapsulating Security Payload (ESP) protocol. AH provides several security services such as connectionless integrity, data origin authentication and optional anti-replay protection. The basic difference between AH and ESP is that ESP provides confidentiality or encryption of data and AH provides data origin authentication. For this reason, while using AH, all the fields inside the header(s) are sent in normal or decrypted form. For data origin authentication and connectionless integrity, all the immutable fields of the header(s) are used to calculate a message digest, which is called Integrity Check Value (ICV) here, and this ICV is added at the end of the header by the sender. After receiving this packet, a receiver can calculate the ICV again and compare it with the received one. For anti-replay protection, sliding window protocol is used [15].

3.2.1 Authentication Header Format

The protocol header (IPv4, IPv6 or IPv6 Extension) that precedes the AH header will contain the value 51 in its Next Header field. The Authentication Header comprises six mandatory fields. They are always present in AH format and used in the *Integrity Check Value (ICV)* calculation. The first five fields have fixed length of three 32-bit words, and the last field has variable length.

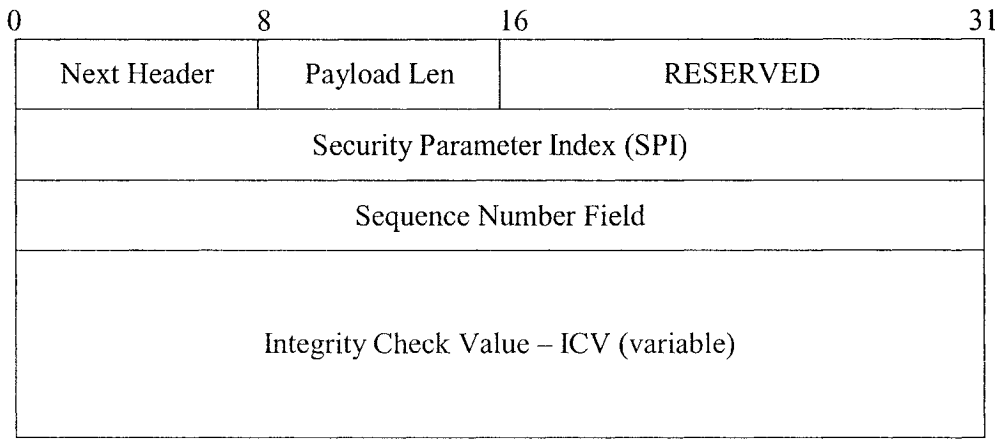


Figure 8: Authentication Header Format

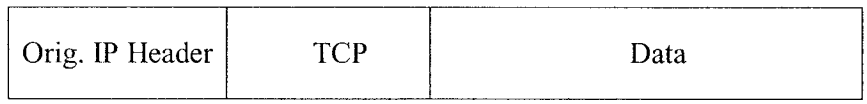
A brief description of the individual fields is as follows:

- *Next Header* is an 8-bit field that specifies the type of the next payload just after the AH header.
- *Payload Length* is another 8-bit field that identifies the length of the total AH in words (32-bit) minus 2.
- *RESERVED* is a 16-bit field that is not being used at present and is set to 0. May be it will be used in future.
- *Security Parameter Index (SPI)*, a 32-bit arbitrary value is used by the receiver to point to its SA database.

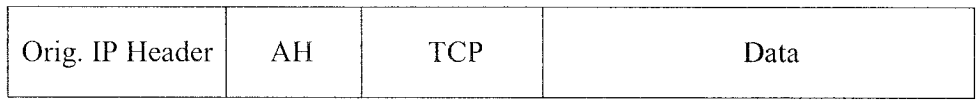
- *Sequence Number Field* is an unsigned 32-bit field, and it contains a counter. This counter is initialized to zero at the beginning. The sender increases it by one each time a new packet is being sent. This is used for anti-replay protection.

- *Integrity Check Value (ICV)* is the last field that fulfills the main purpose of Authentication Header. This field is of variable size and padded if necessary so that the total length of AH remains an exact multiple of 32-bit words.

3.2.2 Authentication Header Location



(a) Before Applying AH

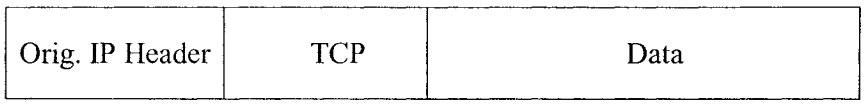


(b) After Applying AH

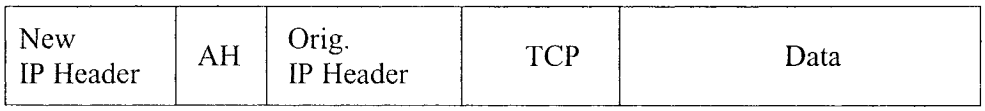
Figure 9: AH Placement in Transport Mode

AH supports two different modes: transport mode and tunnel mode. Transport mode is used primarily for end-to-end authentication between two hosts. In this mode AH is inserted just after the IP header and before a next layer protocol. This next layer may be TCP, UDP, ICMP, etc. or any other IPsec header(s) that have already been inserted. The above diagram illustrates AH transport mode positioning for a typical IPv4 packet, on a "before and after" basis.

In tunnel mode, the "inner" IP header carries the ultimate (IP) source and destination addresses and the "outer" IP header contains the addresses of the IPsec "peers".



(a) Before Applying AH



(b) After Applying AH

Figure 10: AH Placement in Tunnel Mode

Here, these peers refer to security gateways. In tunnel mode, AH protects the entire inner IP packet, including the entire inner IP header. The above diagram illustrates AH tunnel mode positioning for a typical IPv4 packet, on a "before and after" basis. It is clear that the position of AH in tunnel mode, relative to the outer IP header, is the same as for AH in transport mode.

3.2.3 Anti-replay mechanism

By using anti-replay mechanism a receiver can detect a duplicate or replayed packet. The anti-replay mechanism works by keeping track of the sequence numbers in packets as they arrive. The receiver will implement the sliding window protocol and check the sequence number of the received packet against the active sliding window. The packet will be discarded if it was received before.

To accomplish anti-replay mechanism both the sender's and the receiver's counters are initialized to zero when an SA is established. The sender always assumes that anti-replay is enabled, unless otherwise notified by the receiver. Thus the first packet sent using a given SA will contain a Sequence Number 1. If anti-replay is enabled, the sender

checks to ensure that the counter has not cycled before inserting the new value in the Sequence Number field. In other words, the sender must not send a packet on an SA if doing so would cause the Sequence Number to cycle. If anti-replay is disabled, the sender does not need to monitor or reset the counter. However, the sender still increments the counter, and when it reaches the maximum value, the counter rolls over back to zero.

If the receiver does not enable anti-replay for an SA, no inbound checks are performed on the Sequence Number. To avoid having the sender do unnecessary sequence number monitoring and SA setup, if an SA establishment protocol such as IKE is employed, the receiver should notify the sender, during SA establishment, that the receiver would not provide anti-replay protection.

If the receiver has enabled the anti-replay service for this SA, the receiver packet counter for the SA must be initialized to zero when the SA is established. For each received packet, the receiver must verify that the packet contains a Sequence Number that does not duplicate the Sequence Number of any other packets received during the life of this SA. This should be the first AH check applied to a packet after it has been matched to an SA, to speed rejection of duplicate packets. Duplicates are rejected through a sliding window protocol. A minimum window size of 32 must be supported and this size may be up to 64. The highest sequence number that has been received so far will be assigned as the "right edge" of the window. The "left edge" is determined by deducting the window size from the right edge. Whenever a new packet arrives, the receiver will use the following algorithm:

- If the packet is on the left side of the left edge, then the packet will be rejected.
- If the packet is inside the window and not received before, then the packet will be received (after ICV is verified properly), otherwise it will be rejected.
- If the packet is on the right side of the right edge of the window, then the packet will be received (after ICV is verified properly), and the window will be advanced

up to the newly received packet.

From the above algorithm, if the receiver decides that the packet is not a duplicate one, then the receiver proceeds to ICV verification. If the ICV verfication fails, the receiver must discard the received IP datagram as invalid.

3.2.4 Security Association Lookup for Inbound Packets

Upon receipt of a packet containing an IP Authentication Header, the receiver determines the appropriate (unidirectional) SA, based on the destination IP address, security protocol (AH), and the SPI. The SA indicates whether the Sequence Number field will be checked, specifies the algorithm(s) employed for ICV computation, and indicates the key(s) required to validate the ICV. If no valid Security Association exists for this session (e.g., the receiver has no key), the receiver must discard the packet.

3.3 AH Protocol for Multicasting

AH protocol was originally designed for unicast communication [15]. For this reason, if we want to use it in multicast communication, we need some further modification. There is a new version of IP Authentication Header as Internet Draft [14] that considers multicast communication also. Another Internet Draft [3] has been published recently to discuss different IP multicast issues with IPsec and according to [3], the following two modifications in the IPsec AH protocol are necessary:

1. Allow receivers to further refine the SA lookup. In other words, permit a party to have two different SAs, with the same destination address, the same IPsec protocol, and the same SPI but with different source addresses.

2. A wider range of replay protection should be possible.

In light of the guidance provided by the MSEC WG [3], the IPsec WG has proposed [14] that to support multicast AH protocol must modify its Security Association lookup algorithm. Upon receipt of a packet containing an IP Authentication Header, the receiver determines the appropriate (unidirectional) SA via lookup in the Security Association Database (SAD). For a unicast SA, this determination is based on the SPI or the SPI plus protocol field. If an implementation supports multicast traffic, the destination address is also employed in the lookup (in addition to the SPI), and the sender address also may be employed. That means for multicast traffic, we can expect more than one SAD entry for different source addresses with same SPI, destination address and protocol.

At present there is no recommendation in [14] that enables us to implement the anti-replay mechanism for a multi-sender multicast SA. But from the discussion of [3], it is clear that if we want to implement anti-replay mechanism in a multi-sender multicast group, we have to maintain a different sliding window for each sender at the receiver end and each sender will increment its own sequence number. In this way, we would not need further communication among the senders and they could operate freely. The main drawback of this mechanism is that the number of sliding windows to be maintained at the receiver end is directly proportional to the number of senders present in that group. However, this procedure is not feasible for all the receiver hosts, and sometimes it is impractical. For this reason, this method is not recommended in [14]. Later, we shall establish that this mechanism is feasible for our problem and its complexity is bounded to some upper value of n in every situation.

Chapter 4

Authentication of PIM-SM Link-local Messages

While we consider different security issues concerning any routing protocol, we have to deal with two orthogonal issues: the data plane and the control message plane. To provide data integrity and data origin authentication in multicast communication, an IETF Working Group (WG) named Multicast Security (MSEC) [22] has developed the Group Security Association Key Management Protocol [11] and the Group Domain of Interpretation (GDOI) [4]. They are trying to develop some generic solutions for data security that can be deployed with any routing protocol. Different routing protocols use different types of control messages. Thus, security consideration of control messages of a routing protocol is a local issue, and we cannot expect any general solution for all routing protocols. As we are considering only the link-local messages of the PIM-SM protocol, we will concentrate on these messages. More specifically, we will look at when and to whom they are sent.

This is the key chapter of our thesis. First, we shall summarize the proposed authentication mechanism for PIM link-local messages in the present Internet Draft [8]. This

ID includes the IPsec Authentication Header Protocol without anti-replay mechanism as proposed solution. There are some limitations and shortcomings in this proposal which are also mentioned.

A separate Internet Draft [24] was also published to address different security issues in the PIM-SM protocol. Authors of this draft were interested to introduce GDOI [4] to solve the security problem of all PIM-SM control messages including link-local messages. Due to its drawbacks, their proposal was not accepted finally by the PIM community. We shall present their proposal and discuss its limitations in brief.

Finally, we have developed our own solution to solve the security issues of link-local messages. We have proposed a solution that is capable of dealing with anti-replay attacks.

4.1 Authentication according to Present Internet Draft

In the present issue of the PIM-SM Internet Draft, a separate section is dedicated to discuss different security considerations. They have mainly discussed various effects of forged control messages, and finally proposed some methods to prevent all these forged messages. We are mainly concentrating on link-local messages here.

4.1.1 Authentication using IPsec

In this draft, IPsec [17] transport mode using Authentication Header [15] is recommended to prevent attacks generated by forged control messages. It is assumed that one Security Association will be established among all the PIM routers to protect all the link-local messages. The specific AH authentication algorithm and parameters, including the choice of authentication algorithm and choice of key, are configured by the

network administrator. When IPsec authentication is used, all the control messages should go through the IPsec authentication process, and a PIM router should reject any unauthorized PIM protocol messages.

According to [17, 15], the IPsec anti-replay option does not support the case of a Security Association identified by a multicast destination address. For this reason, it is recommended that the anti-replay option be disabled for these Security Associations. It is suggested that the anti-replay option should be enabled only on a Security Association having a unicast destination address. All the link-local messages of the PIM protocol are sent to the destination address ALL_PIM_ROUTERS (IP address 224.0.0.13), which is a multicast address. As a result, the anti-replay option is disabled in the present ID of PIM-SM while using the IPsec Authentication Header protocol.

4.1.2 Establishing and Maintaining Security Associations

There are two ways to establish and maintain an SA: manual technique and automated key management. The IPsec protocols, AH and ESP, are largely independent of the associated SA management techniques. The simplest form of management is manual management, in which a person manually configures each system with keying material and security association management data relevant to secure communication with other systems. Manual techniques are practical in small, static environments but they do not scale well. For widespread deployment and use of IPsec, an automated key management protocol, such as the Internet Key Exchange (IKE) [10] is employed.

The present ID of PIM-SM assumes that manual configuration of Security Associations will be performed, although it does not preclude the use of a negotiation protocol (IKE) to establish Security Associations. The administrator of a PIM network configures each PIM router with one or more Security Associations and associated SPI(s) used by senders to sign PIM protocol messages and by receivers to authenticate received PIM

protocol messages.

To protect link-local multicast messages, the network administrator should assign a Security Association and Security Parameter Index (SPI) on each link in a PIM domain, and this SA should be used to authenticate all link-local PIM protocol messages. The assigned SPI value should be 0 here. We have already discussed in the earlier chapter that to deploy Security Association mechanism successfully we have to maintain two different databases. At first the Security Policy Database (SPD) at a PIM router should be configured to ensure that all incoming and outgoing Join/Prune, Hello and Assert packets use the SA associated with the interface to which the packet is sent. If a router wants to use different authentication methods for each link, it should activate different SAs for each link. At that time, though the destination address is the same for all link-local PIM packets (ALL_PIM_ROUTERS), the selected Security Association for an inbound PIM packet can vary depending on the interface on which the packet has arrived. The network administrator has to assign a different Security Association Database (SAD) for each router interface to activate all these SAs.

4.1.3 Limitations

We have studied very carefully various aspects of the proposed authentication mechanism of PIM link-local messages in the present Internet Draft and have found some limitations and contradictions.

First of all, the anti-replay option of Authentication Header has been disabled during IPsec authentication. Though anti-replay mechanism is optional in AH protocol, it has an important role to counter Denial of Service (DoS) attack. If anti-replay is disabled, a receiver cannot differentiate between a fresh new packet and a duplicate one. A packet can be replayed for various reasons like congestion within Internet or network delay. Moreover, a man-in-the-middle attacker can listen to any packet and replay it after

some time. All these packets should be received and processed by the receiver. After receiving such a duplicate packet, its Integrity Check Value (ICV) will be calculated. The receiver will find the ICV correct because not a single bit of these packets has been changed on its way to the receiver. So, the receiver will take necessary steps according to the received packet. Thus, it will be affected in two ways. Firstly, the receiver will waste its resource by calculating the ICV of some already received packets, and it may cause Denial of Service attack. Secondly, a replayed packet may change any Join, Prune, Assert or Hello state within the receiver router.

According to [8], the network administrator will define a Security Association (SA) first, and this SA will be used to authenticate all link-local PIM protocol messages on each link in a PIM domain. The SPI value of this SA is 0. The destination addresses used for link-local messages are fixed that is ALL_PIM_ROUTERS (224.0.0.13) and AH security protocol is used all the time. These three parameters (SPI, destination address, security protocol) distinguish an SA and are used in security association lookup for inbound packet processing. Again, in [8], it is assumed that there should be a different Security Association Database (SAD) for each router interface. Thus, different authentication methods for each link may be used, and the selected Security Association for an inbound PIM packet can vary depending on the interface on which it has arrived. However, the problem will arise during the lookup stage when an inbound packet will be mapped with the appropriate SA. As mentioned earlier, an SA is distinguished by the three parameters. For PIM link-local messages, these three parameters are fixed and remain unchanged for any link-local message. Finally, we can conclude that it is not possible to use different SADs for each interface while SPI, destination address, and security protocol are used to lookup SAs for inbound packet processing.

4.2 Authentication using GDOI

In addition to the documents from the PIM WG of the IETF, we have found some other efforts in the security area of PIM protocol. Two separate Internet Drafts were published. One of them was related to authentication of PIMv2 messages [26], and another one was a Simple Key Management Protocol for PIM [5]. Due to their complex and inefficient key management protocol, none of them was accepted by the PIM community. An Internet Draft that addresses different security issues in PIM-SM was published by the Group Security (GSEC) WG of the The Internet Research Task Force (IRTF) [24]. They considered security issues of all the PIM protocol messages along with Bootstrap messages. Here, we shall discuss their proposal first, and then we shall point out why their proposal was not accepted.

A problem will be encountered if there is more than one sender, and we want to activate the anti-replay mechanism provided by IPsec. The reason for this problem is that multiple senders will increment the sequence number but there will be no communication among the senders. Thus, sequence number collision will occur and legitimate packets will be dropped by the receivers. To eliminate this difficulty, the authors of [24] proposed to use one SA per group address but more than one anti-replay window per SA while protecting link-local PIM messages. The alternative requires a modification to the IPsec protocol. At that time, receivers have to keep track of a separate sequence number window, and a separate list of received packets in that window per (SA, sender) pair. This option works well with IPsec AH as the source address is protected in the AH mechanism. However, in IPsec ESP, the source address is not protected; and as a result, any spoofed attacker can send an old packet with the source IP address. Therefore, we cannot use ESP at all in this solution.

To set up these SAs, we have to exchange keys between one sender and multiple receivers. We cannot use the conventional Internet Key Exchange (IKE) protocol as it is only used for peer-to-peer entities. However, we can employ GDOI [4] for establishing

a secured multicast group. To accomplish this, we have to modify slightly the present specification of GDOI to use it properly as specified in [24].

4.2.1 Key Exchange Mechanism using GDOI

GDOI is developed as group security association management protocol for a large dynamic group. It is assumed that there should be one Security Association for a specific multicast group. In other words, SAs are established on a per group basis. To authenticate link-local PIM-SM messages by activating anti-replay mechanism of IPsec, one SA per (sender, group) pair is required. For this reason, the following modified version of GDOI is used for establishing keys to maintain the SAs in [24]. If an SA per (sender, group) pair is required, a new SA should be created whenever a sender joins the group. In case receivers join, it is not necessary to create a new SA.

GDOI assumes the existence of a central control entity, the Group Controller and Key Server (GCKS). At first, every PIM router should authenticate GCKS via standard IKE phase 1 peer-to-peer authentication. A PIM router that is within the same administrative domain, shares a predefined secret with the Group Controller. This secret is only shared between the PIM router and the Group Controller as a peer-to-peer relation, and this is different for all the PIM entities. For establishing an SA per (sender, group) pair, the next step of the present GDOI should be modified.

When an entity initiates a phase 2 negotiation with GCKS, it will mention its status of join (as a sender or a receiver) along with the target group it wants to join. Then, the GCKS will take one of the following steps on its status of willingness:

- If the entity wants to join as a receiver, no new SA will be created for that group and the GCKS pushes the existing IPsec SAs for that group including security protocol, cryptographic algorithms, and parameters.

- If the entity wants to join as a sender, the GCKS creates a new SA for the combination (sender, multicast group address). Then GCKS assigns an SPI value that does not exist yet for that group address and fixes the security protocol, cryptographic algorithms, parameters, etc., according to a pre-configured security policy. Finally, GCKS pushes this SA to the newly joined sender and all the existing registered receivers of that multicast group. Certainly GCKS maintains the list of existing members all the time.

4.2.2 Limitations

The concept of using a modified version of GDOI to protect link-local PIM protocol messages was not accepted for the following reasons:

- The GDOI manages group security associations, which are used by IPsec and potentially other data security protocols running at the IP or application layers [4]. These security associations protect one or more key-encrypting keys, traffic-encrypting keys, or data shared by group members. GDOI is actually developed for large dynamic groups where members of the group join/leave dynamically and the number of members in a group is very large. In comparison, the number of PIM routers are not as large, and in general, a PIM router will join the multicast group that is ALL_PIM_ROUTERS (224.0.0.13) only once, at its booting time. It will not leave this multicast group unless the router goes down for some unavoidable reason.

- The proposed modified GDOI will act differently upon the joining of a sender and a receiver. Whenever a new sender is joined to the multicast group, a new Security Association should be created, and GCKS will *push* this SA to all existing PIM routers. If we consider a PIM router, it should always send PIM protocol messages to join the distribution tree unless it is the root of the distribution tree

and that is very rare. Moreover, all the PIM routers should send Hello message. Therefore, it is clearly understood that, the number of senders of the multicast group (ALL_PIM_ROUTERS) will be equal to the number of PIM routers. The modified GDOI will create that number of Security Associations. Certainly, this is not very scalable as we have to *push* all these SAs to present active PIM routers.

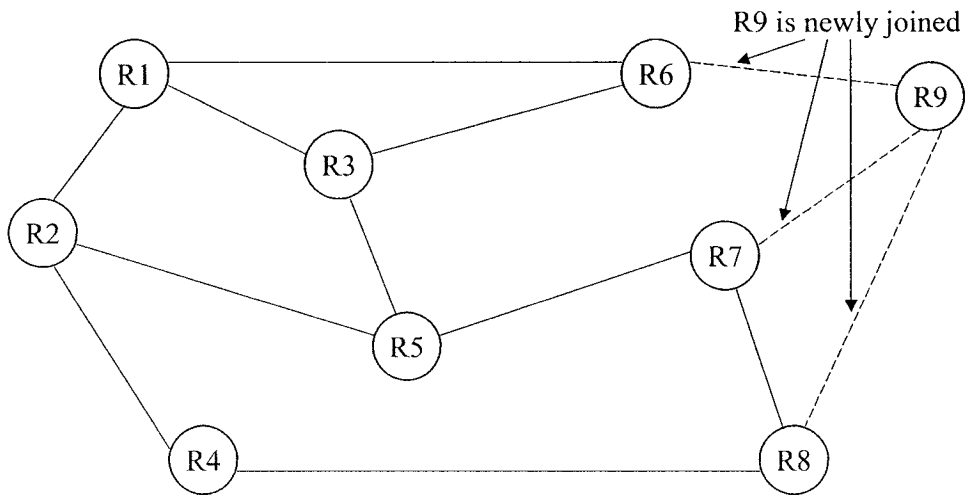


Figure 11: PIM Routers Connectivity Example

- A PIM link-local message is sent with TTL=1 and never forwarded by the receiver router to another router. That means it is only sent to the neighbour(s) directly connected to a router. If we take the scenario of Figure 11, say router R3, it is directly connected with R1, R5 and R6. When R3 will act as sender, we need an SA between R3, R1, R5 and R6 and no other router should be aware of this SA and its member. However, the modified GDOI will create a new SA whenever a new sender will join and distribute this SA to all the registered PIM routers. For instance, when R9 will join the ALL_PIM_ROUTERS multicast group as sender, a new SA will be created, and it will be *pushed* to the remaining 8 routers. This is totally inefficient because only R6, R7 and R8 should be aware of this SA, and

they should use it solely. In fact, the main characteristic of link-local messages is ignored here.

- As we have mentioned earlier, a routing protocol has two orthogonal planes: data plane and control plane. GDOI is actually developed as a general solution to secure the data plane, and for this reason, it is not suitable to secure protocol messages. Protocol messages widely vary from one protocol to another and must be considered on the basis of their own characteristics.

4.3 Proposed Authentication Techniques

So far we have discussed the mechanism that is proposed in the Internet Draft [8] and found the various limitations of this proposal. We have also demonstrated why GDOI based mechanism is not suitable to protect PIM link-local messages. We are now in the position to represent our own proposal to protect these messages. Just before doing that, we want to set two goals that we hope to achieve in this particular issue. They are as follows:

1. We must be able to activate the anti-replay mechanism while sending/receiving any PIM link-local message.
2. For more flexibility, we want to be able to deploy a different authentication method for each sender. In other words, we want to maintain at least different Security Association Database (SAD) per peer sender.

4.3.1 Activating the Anti-replay Mechanism

There is no doubt that if we want to activate the anti-replay mechanism in a multi-sender multicast group communication we have to maintain (in the receiver) one separate sliding

window or counter per sender. In this manner, all the senders will use their own sequence number, and we will not need any communication between the senders. This solution is not always feasible and in some cases impossible. The number of senders in a multicast group varies from one to thousands, and for a host computer it is not possible to handle thousands of sliding windows. This number is directly proportional to the number of senders. However, the whole scenario is different for PIM link-local messages. These messages are sent to local links with TTL = 1 and in aggregate fashion. A router will send all outstanding Join/Prune messages, and in one message it may send Join and Prune message for more than one multicast group. Here, the number of senders is proportional to the number of local links or interfaces connected to a router. Moreover, these messages never propagate from one router to another, and as they are always sent with TTL = 1, they can travel only one router. We can add here one point for more clarification: it may happen that more than one router is connected through the same interface with a particular router. In this case, we have to consider each router differently. In such a situation, we have to activate and maintain a sliding window per peer router, not per interface, though all of them are connected through the same interface.

Finally, we can conclude that we can activate the anti-replay mechanism and maintain different sliding windows on a per interface basis, and in case there is more than one router is connected through the same interface, we should maintain different sliding windows per peer router.

4.3.2 Security Association Lookup

If we want to use different authentication methods and different keys for each interface of a router, we have to maintain one Security Association Database (SAD) per interface. In other words, we are expecting that an inbound PIM packet Security Association may vary depending on the interface on which the packet arrived. The major drawback in

the present SA lookup algorithm is it uses SPI, IP destination address and the Protocol to differentiate among these SAs. This mechanism is ineffective because these three parameters (SPI = 0, destination address = ALL_PIM_ROUTERS and protocol used = AH) are the same for all the SAs. In fact, according to [14], we do not need to check the destination address. Instead, we can use the source address in conjunction with the SPI to sort out a particular SA from all the SAD entries. According to our proposal number of SAD entries will be slightly higher than the proposed method in the PIM specification. We are expecting that there should be an SAD entry in a receiver router for each source that is connected with that router, even though previously in PIM spec it was for each interface. In some cases, it may happen that more than one router is connected with a router through one interface. As we want to activate the anti-replay mechanism, this refined scenario will be more suitable.

4.3.3 Manual Key Configuration

To implement IPsec SA, we have to establish the SA first. Here, manual key configuration will be more feasible than automatic key configuration. We are assuming that the network administrator will configure a router manually during its boot up process. At that time, we have to configure so many parameters manually that it will not be so difficult to configure a router with the SA that should be used to send link-local messages. Certainly, we have to configure an authentication method and keys per sender basis for each interface. Normally, we are expecting one sender per interface. We have to also create the SAD and Security Policy Database (SPD) entries for each sender connected with this router.

Moreover, we can use a negotiation protocol such as the Internet Key Exchange to establish the SA between routers and negotiate their suitable authentication method and keys. In that case, we have to go through IKE phase 1 authentication first. Whether we use digital signature, pre-shared secret or certificate authentication, we have to do some

offline task or manual configuration to each router. For this reason, we are recommending manual configuration of SA. Another important point to consider is that we don't expect that a router will join/leave very frequently. In a dynamic group, it is natural that a host computer may leave/join from a multicast group frequently and certainly automatic key configuration is the best choice at that time. In contrast to a host computer, a router is always connected with other routers. It is not member of a particular multicast group, and does not serve a particular group only.

4.3.4 Extended Sequence Number

In [14], it is recommended that if we want to activate the anti-replay mechanism in Authentication Header (AH) protocol, we should use automatic key configuration. In general, we use a 32-bit counter to generate the sequence number while anti-replay is activated [17]. This counter starts from zero, and we can send at most $(2^{32} - 1)$ packets. Then, we have to reset the counter in both the receiver and the sender end. When this occurs, there is no automated recovery process for manual key distribution. For this reason, automatic key distribution is recommended in the AH protocol specification.

In the new version of AH protocol [14] there is a provision for a 64-bit Extended Sequence Number (ESN). Both the sender and the receiver maintain 64-bit counter for the sequence number though only the lower order, 32 bits, is sent in the transmission. In other words, it will not affect the present header format of AH. If we use ESN, we can send at most $(2^{64} - 1)$ packets. This number is a huge one, and if we consider it from a router's point of view, a router can never exceed this number in its lifetime.

From the above discussion, it is clear that we can safely use manual key configuration while using IPsec AH protocol. However, the condition is that we must use the Extended Sequence Number when activating the anti-replay mechanism.

Chapter 5

Validation using SPIN

We are now in the position to validate our proposal. In the literature, the difference between protocol *validation* and *verification* is often ambiguous. In some cases, verification means to verify general properties of a protocol, such as the absence of deadlock, unspecified reception, and livelocks. On the other hand, validation means to validate specific properties of a protocol against the specification requirements. In his book [12], Holzmann uses both validation and verification for the same meaning. We are performing validation of our model in our thesis.

In this research, we have used the formal validation language, PROMELA (PROcess MEta LAnguage) to specify the validation model, and then used a tool, SPIN (Simple PROMELA INterpreter), to validate our model. We have presented a short description of PROMELA and SPIN first. Then, we have listed a set of requirements that our model should satisfy. For a better understanding, different processes of our model have been shown in sequence diagram or flow chart. Finally, the results of our validation have been given.

5.1 PROMELA: A Protocol Validation Language

To validate a protocol, we have to develop a *validation model* of the protocol. This model is called partial description of the protocol because a validation model defines the interactions of processes in a distributed system. It says nothing about the implementation details, the format of a message, or how a message should be transmitted or encoded. The validation model only concentrates in the design of a complete and consistent set of rules to govern the interactions in a distributed system. PROMELA [12] is a specification and modelling language that can be used to develop the validation model of different protocols. In comparison to other programming languages, PROMELA has several unusual features that make it suitable for modelling distributed systems.

In PROMELA, procedure rules are used as formal programs to model distributed systems. The model should be as simple as possible yet sufficiently powerful to represent all types of coordination problems that can occur in a distributed system. A validation model is defined in terms of three specific types of objects:

- **Variables**: In PROMELA a variable may be either global or local depending on its place of declaration. There are six predefined data types, such as *bit, bool, byte, short, int* and *chan*. The first five types are the basic data types and are used to specify objects that can hold a single value. The last type specifies message channels.

- **Process**: A process is defined by *proctype* declaration, and it is followed by its name and instance or body. The following, declares a process with one local variable named *state*:

```
proctype A() {
    byte state;
    state = 3
```

```
}
```

A proctype definition declares process behavior only. However, to execute a process, we have to *run* it. Initially, just one process named *init( )* is executed. This is similar to the *main( )* function of C programming language. The following is an example of *init( )* process that executes process A:

```
proctype init() {
    run A()
}
```

- **Message Channels**: To model the transfer of data between two processes, a message channel is used. This type of channel may be either global or local and can be declared in the same way variables of the basic data types, using the keyword *chan*. The following is an example of declarations, where a and b are simple message channels and c is an array:

```
chan a, b;
chan c[3]
```

5.2 Components of SPIN

SPIN [13] is a generic validation system that supports the design and the validation of models written in PROMELA. It can simulate the execution of a validation model by interpreting PROMELA statements on the fly. SPIN validation models are focused on proving the correctness of process interactions. These process interactions can be specified in SPIN with rendezvous primitives, with asynchronous message passing through

buffered channels, through access to shared variables, or with any combination of these. It accepts correctness claims specified in the syntax of standard Linear Temporal Logic (LTL).

Following are some special features of SPIN [23]:

- SPIN is used as an efficient software verification and not as a hardware verification. It can detect any types of logical design error in distributed systems and checks the logical consistency of a specification. It reports on deadlock, livelock and improper termination.
- It works on-the-fly and avoids the need to construct a global state graph as a prerequisite for the verification of the system properties.
- The correctness of a model can be specified as system or process invariants (when *assert* statement is used), as Linear Temporal Logic requirements (LTL), as formal *Büchi Automata*, or more broadly in the syntax of *never* claim.
- SPIN supports dynamic increase or decrease in the number of processes.
- For interactions between two processes, rendezvous message passing, buffered message passing or communication through shared memory can be used. Both synchronous and asynchronous communications are supported.
- The tool supports random, interactive and guided simulation. Both exhaustive and partial proof techniques based on either depth-first or breadth-first search can be used.
- To optimize the verification runs, the tool exploits efficient partial order reduction techniques.

The basic structure and different steps of the SPIN model checker is illustrated in Figure 12 [13]. A graphical interface XSPIN is used in the starting phase to specify the

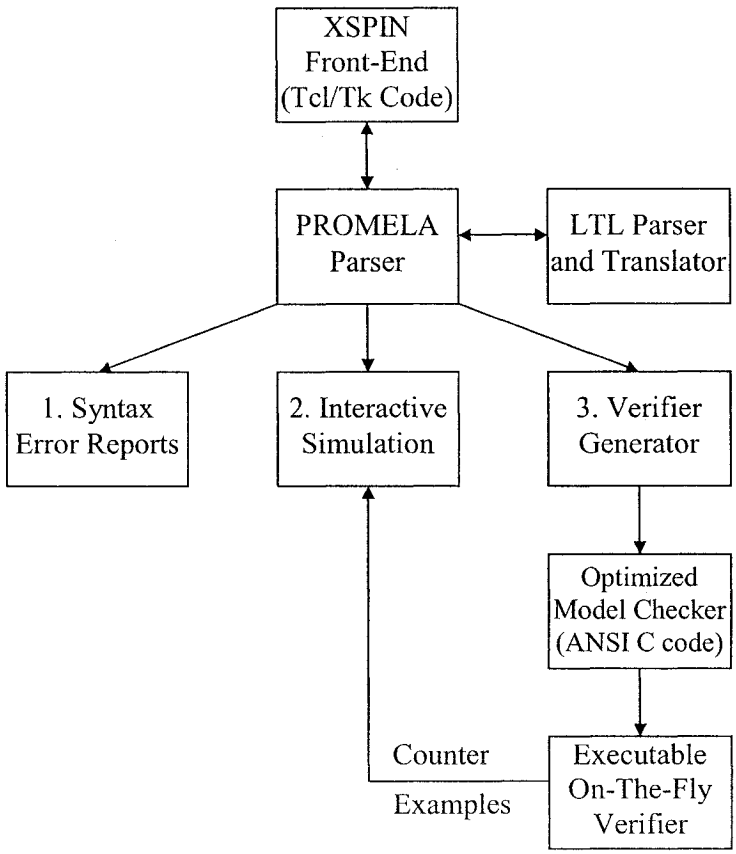


Figure 12: The Structure of SPIN Simulation and Verification

high level model of a concurrent system or distributed algorithm. A PROMELA parser is used to fix the syntax errors. In the next step, interactive simulation is performed to gain the basic confidence that the design is behaving as expected. Then, in the third step, SPIN is used to generate an optimized on-the-fly verification program from the high level specification. This verifier or model-checker (pan.c) is nothing but a program written in ANSI C code. It is possible to compile this program (pan.c) by any standard C compiler. Different compile-time options can be used to choose the appropriate reduction algorithm. Then, the compiled program can be executed with different run-time options. If any counterexamples to the correctness claims are detected, these can be fed back into the interactive simulator and be inspected in detail to establish and remove their cause.

Otherwise, the output will confirm the correctness of the verifier.

5.3 Specification of the Validation Model

The validation of a protocol does not mean the implementation of it. While we are validating it, we can define a set of requirements, and design the validation model to satisfy those requirements. For this reason, we have listed the following requirements, and we should be aware of this list during the construction of a validation model.

1. We want to activate a different Security Association (SA) for each link for a router. More specifically, if more than one source routers are connected with a receiver router through one interface the number of activated SAs will be equal to the number of connected routers. From a receiver router point of view, there should be a different entry in its own Security Association Database (SAD) for each source router connected to it. When this occurs, we have the freedom to use a different authentication and encryption algorithm for different SAs.

2. An SA will be distinguished by the source address and the SPI not by the three parameters: SPI, destination address and protocol used. When a new packet is received, we have to lookup the entries in the SAD, and for this purpose we have to use the sender address and the SPI of this packet.

3. We shall activate the anti-replay mechanism while sending/receiving a link-local message. We have to use a sliding window per connected source router, which means per SA. We have to maintain a necessary database for each SAD entry.

4. Our validation model should demonstrate that a receiver is capable enough to face various attacks such as replay attack or impersonating the sender address, etc.

5. Finally, in the new version of Authentication Header (AH) protocol [14], there is a provision of 64-bit Extended Sequence Number (ESN) for anti-replay mechanism.

In this version of AH protocol, a new algorithm for anti-replay window that uses ESN has been also presented. In our model, we also want to validate this algorithm and demonstrate that it works well while the anti-replay mechanism is activated.

5.4 Description of our Validation Model

We have constructed a validation model using PROMELA and validated this model by SPIN. We have designed the model in such a way that it should be as simple as possible but at the same time, satisfy all the specification mentioned in the previous section. Our model consists of one receiver and three senders, and among the senders two of them are true senders and sending valid messages to the receiver. To simulate different attacks, the third sender will send various invalid or false packets to the receivers. In this section, we have presented each process of our model using a sequence chart or a flow diagram to illustrate different functionalities of our model.

As we have mentioned, all the PROMELA programs should have an *init* process that runs at the beginning. In our *init* process, we have initialized SADs entries and started one *receiver* and three *sender* processes.

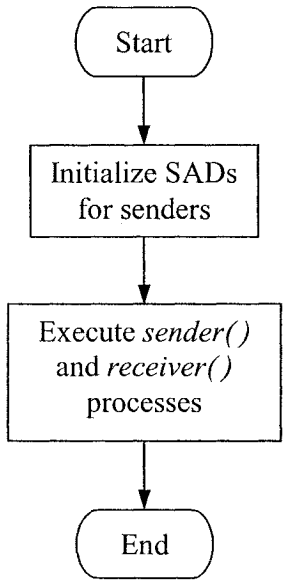


Figure 13: Sequence Diagram of *init* procedure

In the following figure the *sender* process is explained. This sender represents a true or valid one generating valid messages. In this process, we have initialized its own data first and then the value of Extended Sequence Number (ESN). In the original program, we have instantiated two processes of this type and each process sends various valid messages to the receiver one after another. The anti-replay window of our model uses ESN to maintain a sliding window protocol. For this reason, we have generated messages in such a way that we can cover all the cases of this sliding window protocol.

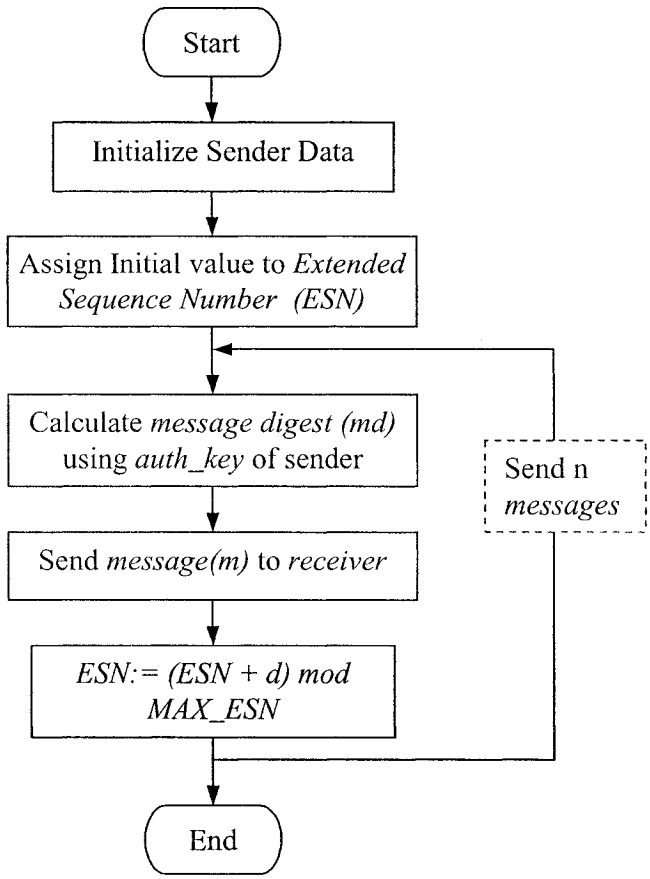


Figure 14: Sequence Diagram of *sender* procedure for a regular sender

The *receiver* process will work until a *timeout* has occurred which means no message will be received for a particular amount of time. Once a new message is received, it will create a new instance of *anti-replay* process if its SPI = 0 and destination address is ALL_PIM_ROUTERS. Otherwise, the message is simply discarded.

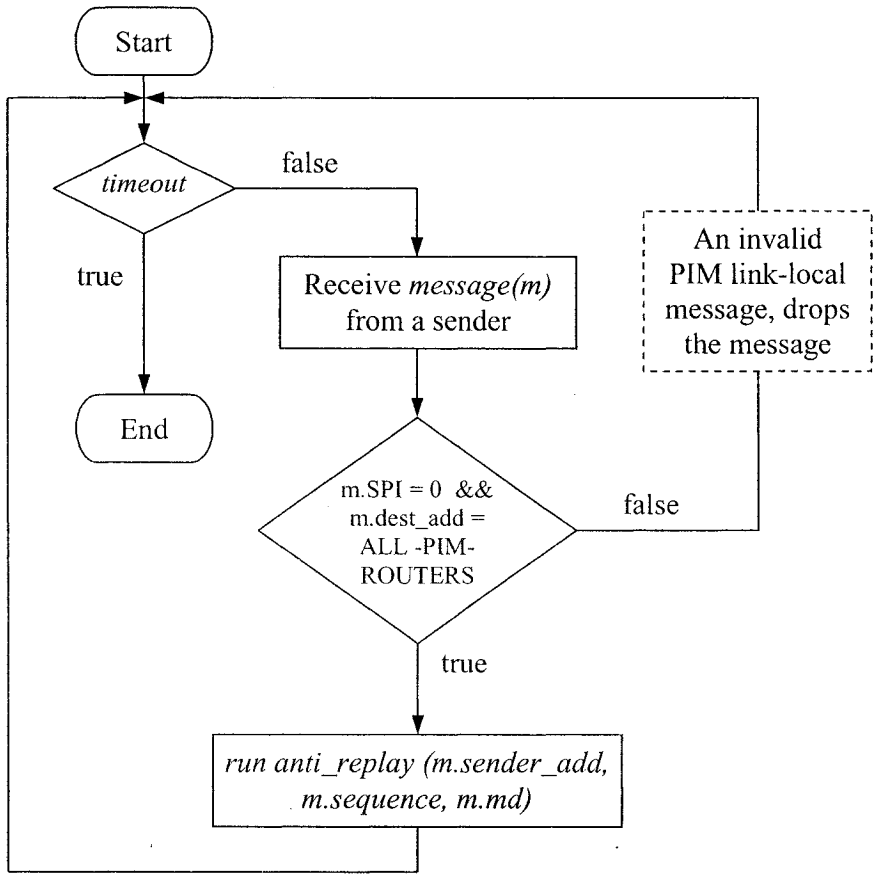


Figure 15: Sequence Diagram of *receiver* procedure

At the beginning of the *anti-replay* process, SADs entries will be looked up using the sender address. If an appropriate SAD is found, the message will be authenticated using the authentication key. Otherwise, the message will be discarded. Then, the sequence number of the message will be checked using a sliding window protocol. If the packet was received before, it will be discarded. Or else, the message will be received and the window will be advanced if necessary.

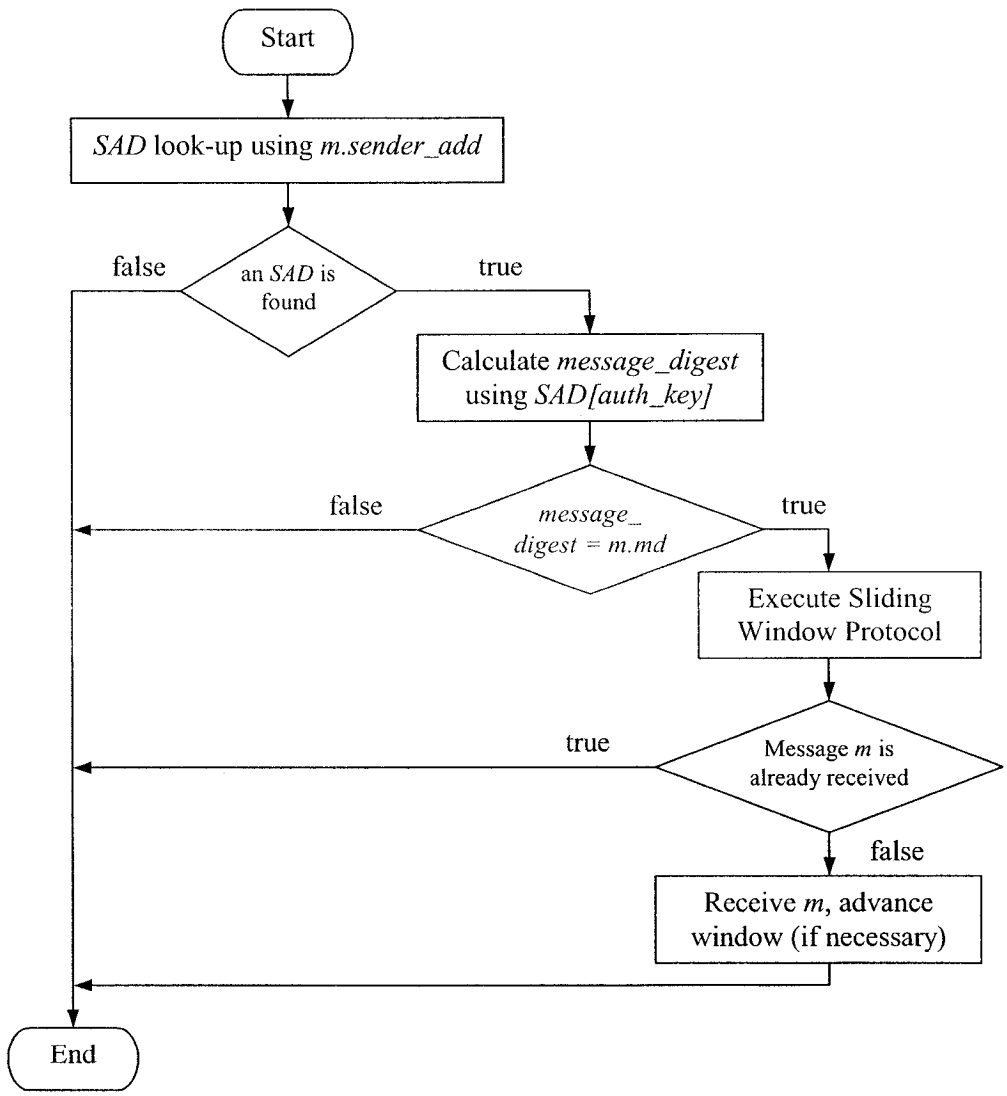


Figure 16: Sequence Diagram of *anti-replay* procedure

Our last process is to simulate a forging sender that will generate different false messages or replay any previously sent messages. In this instance, we attempted to test all possible attacks a receiver may face.

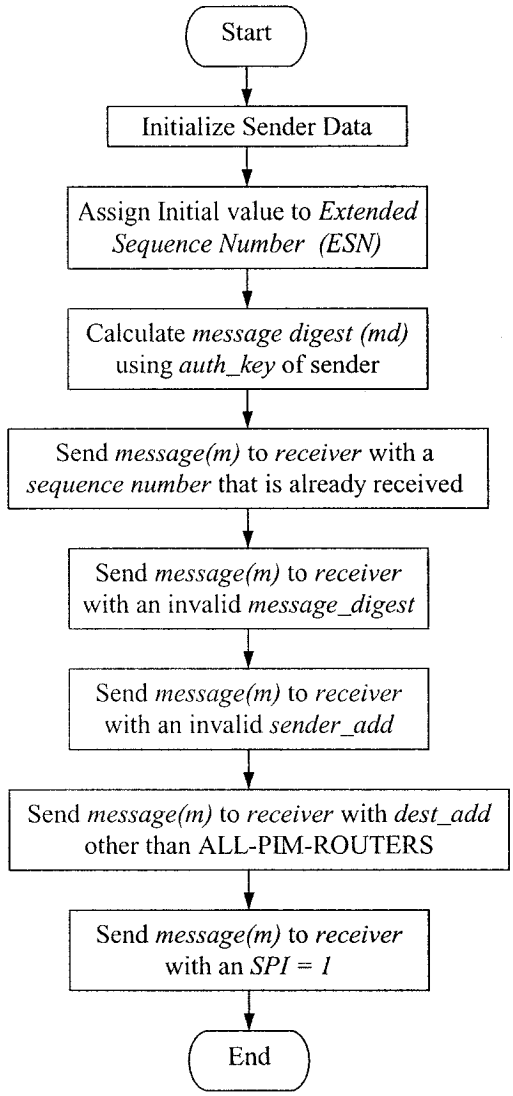


Figure 17: Sequence Diagram of *sender* procedure for an attacker sender

5.5 Validation Results

As we have presented before in Figure 12, a SPIN validation has three consecutive steps. We have used SPIN version 4.0.6 throughout the process. In first step, a graphical interface, XSPIN is used to specify the high level model that is written in PROMELA. Then, a PROMELA parser is used to fix the syntax error if present.

Once we are sure that our validation model is free from syntax error, different random simulation runs are performed using either no option or various options. Some of the options are c (for columnated output), p (to show process moves), etc. We have found no error during simulation. All these simulation runs have helped to build our confidence that our model is behaving as expected. Before advancing to our next step, we have generated the verifier using a option. This time the model-checker or verifier (pan.c) is generated.

We have compiled this verifier using different compile time options. One should chose the appropriate option(s) as the nature of reachability analysis that would be used depends on these options. There are three basic modes: random mode, full exhaustive search and partial state space search. Instead of an exhaustive search, we have used *DBITSTATE* option during compilation. When the validator runs, it uses controlled partial state space search technique or *supertrace* mode. This mode can be accomplished in much smaller amounts of memory and still retain excellent coverage of the state space. To reduce the required memory, we have used another compile time option, *DSAFETY* that optimizes for the case where no cycle detection is needed.

We are now prepared to run our verifier. Once this verifier is executed, the output confirms that our model is free from different errors such as assertion violation and invalid end state. From the output, it is also established that there is no unreachable state in our design.

Chapter 6

Conclusion

There is no doubt that multicast communication was suffering from the probability of a security breach from the very beginning. If we expect large and commercial deployment of multicasting through Internet Service Providers (ISP), we have to satisfy its security in every way. PIM-SM is going to be the dominant routing protocol for multicasting based applications, if we can provide security for data packets and for the control messages as well. The core interest of our research work was to protect PIM link-local control messages from all sorts of attacks. We have reasonably established that the existing method as proposed in the Internet Draft of PIM-SM is not sufficient. Moreover, it contradicts itself in some issues such as SA lookup and in the number of active SAs. We have proposed a very simple and complete solution. We were very much conscious so that our solution would not add much more overhead and would be compatible with the original specification of PIM-SM. Finally, we used a tool, SPIN, that is widely used for protocol validation. In our validation model, we have covered all the possible attacks a typical PIM router may face in the real world through link-local messages. Our validation model proved that our proposal is capable enough to detect all sorts of attacks and can take necessary steps. At this point, we can confidently conclude that we have achieved our goal, and it will help the PIM community to secure the link-local messages.

6.1 Contributions

We have found the following contributions from our research work:

- A complete solution has been proposed to protect the PIM link-local messages from all sorts of attacks.
- The Security Association lookup method has been modified, and instead of using the three parameters (Destination Address, SPI and Protocol used) the Sender Address and the SPI are used by the receiver router to find out the appropriate SA from Security Association Database (SAD).
- To make everything more flexible, it is now possible to assign different authentication and encryption algorithms for each active SA between two routers, and it will not produce any contradictions with the number of active SAs and their related SAD.
- To validate the proposed methods, a validation model has been designed using PROMELA, and using SPIN the model has been validated.
- While validating the proposed model, an Extended Sequence Number (64-bit sequence number) algorithm has been used to implement sliding window protocol that means this algorithm has been validated also.

6.2 Future Work

Though most of the PIM control messages fall in link-local category, there are other types of control messages. There are two unicast control messages such as Register, Register-Stop, and there are other Bootstrap messages which are not specified in the PIM Internet Draft. These messages are part of the Bootstrap mechanism but have an

important role in Rendezvous Point selection. Our future work will concentrate on these messages, and we should consider security issues for all the control messages.

Bibliography

[1] Adams, A., Nicholas, J., Siadak, W. *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised).* Internet Draft, draft-ietf-pim-dm-new-v2-04.txt, Work in Progress, IETF, September 2003.

[2] Ballardie, A. *Core Based Trees (CBT version 2) Multicast Routing.* RFC2189, IETF, September 1997.

[3] Baugher, M., Canetti, R., Hardjono, T., Weis,B. *IP Multicast issues with IPsec.* Internet Draft, draft-ietf-msec-ipsec-multicast-issues-01.txt, Work in Progress, IETF, December 2002.

[4] Baugher, M., Hardjono, T., Harney, H., Weis, B. *The Group Domain of Interpretation.* RFC3547, IETF, July 2003.

[5] Cain, B. *Simple Key Management Protocol for PIM.* Internet Draft, draft-ietf-pim-simplekmp-01.txt, Work in Progress, IETF, February 2000.

[6] Cain, B., Deering, S., Kouvelas, I., Fenner, B., Thyagarajan, A. *Internet Group Management Protocol, Version 3.* RFC3376, IETF, October 2002.

[7] Deering, S. *Host Extensions for IP Multicasting.* RFC1112, IETF, August 1989.

[8] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I. *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised).* Internet Draft, draft-ietf-pim-sm-v2-new-08.txt, Work in Progress, IETF, October 2003.

[9] Frankel, S. *Demystifying the IPsec Puzzle.* Artech House, Inc., 2001.

[10] Harkins, D., Carrel, D. *The Internet Key Exchange (IKE).* RFC2409, IETF, November 1998.

[11] Harney, H., Colegrove, A., Harder, E., Meth, U., Fleischer, R. *Group Security Association Key Management Protocol.* Internet Draft, draft-ietf-msec-gsakmp-sec-03.txt, Work in Progress, IETF, August 2003.

[12] Holzmann, G. J. *Design and Validation of Computer Protocols.* Prentice Hall, 1991.

[13] Holzmann, G. J. *The Model Checker SPIN.* IEEE Transactions on Software Engineering, Vol. 23, No. 5, May 1997.

[14] Kent, S. *IP Authentication Header.* Internet Draft, draft-ietf-ipsec-rfc2402bis-05.txt, Work in Progress, IETF, September 2003.

[15] Kent, S., Atkinson, R. *IP Authentication Header.* RFC2402, IETF, November 1998.

[16] Kent, S., Atkinson, R. *IP Encapsulating Security Payload (ESP).* RFC2406, IETF, November 1998.

[17] Kent, S., Atkinson, R. *Security Architecture for the Internet Protocol.* RFC2401, IETF, November 1998.

[18] Kosiur, D. *IP Multicasting: The Complete Guide to Interactive Corporate Networks.* Wiley Computer Publishing, 1998.

[19] Maughan, D., Shertler, M., Schneider, M., Turner, J. *Internet Security Association and Key Management Protocol.* RFC2408, IETF, November 1988.

[20] Microsoft Corporation. *PIM-SM Multicast Routing Protocol.* White Paper, December 1999.

[21] Moy, J. *Multicast Extension to OSPF.* RFC1584, IETF, March 1994.

[22] Multicast Security (msec) Working Group, IETF. http://www.ietf.org/html.charters/msec-charter.html.

[23] SPIN's homepage. http://spinroot.com/spin/whatispin.html.

[24] Van Moffaert, A., Paridaens, O. *Security issues in Protocol Independent Multicast - Sparse Mode (PIM-SM).* Internet Draft, draft-irtf-gsec-pim-sm-security-issues-02.txt, Work in Progress, IETF, June 2002.

[25] Waitzman, D., Patridge, C., Deering, S. *Distance Vector Multicast Routing Protocol.* RFC1075, IETF, November 1988.

[26] Wei, L. *Authenticating PIM Version 2 Messages.* Internet Draft, draft-ietf-pim-v2-auth-00.txt, Work in Progress, IETF, November 1998.

[27] Wittmann, R., Zitterbart, M. *Multicast Communication Protocols and Applications.* Morgan Kaufmann Publishers, 2001.