Network Working Group
Request for Comments: 5374
Category: Standards Track

B. Weis
Cisco Systems
G. Gross
Secure Multicast Networks LLC
D. Ignjatic
Polycom

November 2008

Multicast Extensions to the Security Architecture for the Internet Protocol

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The Security Architecture for the Internet Protocol describes security services for traffic at the IP layer. That architecture primarily defines services for Internet Protocol (IP) unicast packets. This document describes how the IPsec security services are applied to IP multicast packets. These extensions are relevant only for an IPsec implementation that supports multicast.

Table of Contents

1. Introduction

The Security Architecture for the Internet Protocol [RFC4301] provides security services for traffic at the IP layer. It describes an architecture for IPsec-compliant systems and a set of security services for the IP layer. These security services primarily describe services and semantics for IPsec Security Associations (SAs) shared between two IPsec devices. Typically, this includes SAs with traffic selectors that include a unicast address in the IP destination field, and results in an IPsec packet with a unicast address in the IP destination field. The security services defined in RFC 4301 can also be used to tunnel IP multicast packets, where the tunnel is a pairwise association between two IPsec devices. RFC 4301 defined manually keyed transport mode IPsec SA support for IP packets with a multicast address in the IP destination address field. However, RFC 4301 did not define the interaction of an IPsec subsystem with a Group Key Management protocol or the semantics of a tunnel mode IPsec SA with an IP multicast address in the outer IP header.

This document describes OPTIONAL extensions to RFC 4301 that further define the IPsec security architecture in order for groups of IPsec devices to share SAs. In particular, it supports SAs with traffic selectors that include a multicast address in the IP destination field and that result in an IPsec packet with an IP multicast address in the IP destination field. It also describes additional semantics for IPsec Group Key Management (GKM) subsystems. Note that this document uses the term "GKM protocol" generically and therefore does not assume a particular GKM protocol.

An IPsec implementation that does not support multicast is not required to support these extensions.

Throughout this document, RFC 4301 semantics remain unchanged by the presence of these multicast extensions unless specifically noted to the contrary.

1.1. Scope

The IPsec extensions described in this document support IPsec Security Associations that result in IPsec packets with IPv4 or IPv6 multicast group addresses as the destination address. Both Any-Source Multicast (ASM) and Source-Specific Multicast (SSM) [RFC3569] group addresses are supported. These extensions are used when management policy requires that IP multicast packets protected by IPsec remain IP multicast packets. When management policy

requires that the IP multicast packets be encapsulated as IP unicast packets (e.g., because the network connected to the unprotected interface does not support IP multicast), the extensions in this document are not used.

These extensions also support Security Associations with IPv4 Broadcast addresses that result in an IPv4 link-level Broadcast packet, and IPv6 Anycast addresses [RFC2526] that result in an IPv6 Anycast packet. These destination address types share many of the same characteristics of multicast addresses because there may be multiple candidate receivers of a packet protected by IPsec.

The IPsec architecture does not make requirements upon entities not participating in IPsec (e.g., network devices between IPsec endpoints). As such, these multicast extensions do not require intermediate systems in a multicast-enabled network to participate in IPsec. In particular, no requirements are placed on the use of multicast routing protocols (e.g., Protocol Independent Multicast - Sparse Mode (PIM-SM) [RFC4601]) or multicast admission protocols (e.g., Internet Group Management Protocol (IGMP) [RFC3376]).

All implementation models of IPsec (e.g., "bump-in-the-stack", "bump-in-the-wire") are supported.

This version of the multicast IPsec extension specification requires that all IPsec devices participating in a Security Association be homogeneous. They MUST share a common set of cryptographic transform and protocol-handling capabilities. The semantics of an "IPsec composite group" [COMPGRP], a heterogeneous IPsec cryptographic group formed from the union of two or more sub-groups, is an area for future standardization.

1.2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

The following key terms are used throughout this document.

Any-Source Multicast (ASM)
: The Internet Protocol (IP) multicast service model as defined in RFC 1112 [RFC1112]. In this model, one or more senders source packets to a single IP multicast address. When receivers join the group, they receive all packets sent to that IP multicast address. This is known as a (*,G) group.

Group
A set of devices that work together to protect group communications.

Group Controller Key Server (GCKS)
A Group Key Management (GKM) protocol server that manages IPsec state for a group. A GCKS authenticates and provides the IPsec SA policy and keying material to GKM Group Members.

Group Key Management (GKM) Protocol
A key management protocol used by a GCKS to distribute IPsec Security Association policy and keying material. A GKM protocol is used when a group of IPsec devices require the same SAs. For example, when an IPsec SA describes an IP multicast destination, the sender and all receivers need to have the group SA.

Group Key Management Subsystem
A subsystem in an IPsec device implementing a Group Key Management protocol. The GKM subsystem provides IPsec SAs to the IPsec subsystem on the IPsec device. Refer to RFC 3547 [RFC3547] and RFC 4535 [RFC4535] for additional information.

Group Member
An IPsec device that belongs to a group. A Group Member is authorized to be a Group Sender and/or a Group Receiver.

Group Owner
An administrative entity that chooses the policy for a group.

Group Security Association (GSA)
A collection of IPsec Security Associations (SAs) and GKM subsystem SAs necessary for a Group Member to receive key updates. A GSA describes the working policy for a group. Refer to RFC 4046 [RFC4046] for additional information.

Group Security Policy Database (GSPD)
The GSPD is a multicast-capable security policy database, as mentioned in RFC 3740 and Section 4.4.1.1. of RFC 4301. Its semantics are a superset of the unicast Security Policy Database (SPD) defined by Section 4.4.1 of RFC 4301. Unlike a unicast SPD-S, in which point-to-point traffic selectors are inherently bi-directional, multicast security traffic selectors in the GSPD-S include a "sender only", "receiver only", or "symmetric" directional attribute. Refer to Section 4.1.1 for more details.

GSPD-S, GSPD-I, GSPD-O
Group Security Policy Database (secure traffic), (inbound), and (outbound), respectively. See Section 4.4.1 of RFC 4301.

Group Receiver
A Group Member that is authorized to receive packets sent to a group by a Group Sender.

Group Sender
A Group Member that is authorized to send packets to a group.

Source-Specific Multicast (SSM)
The Internet Protocol (IP) multicast service model as defined in RFC 3569 [RFC3569]. In this model, each combination of a sender and an IP multicast address is considered a group. This is known as an (S,G) group.

Tunnel Mode with Address Preservation
A type of IPsec tunnel mode used by security gateway implementations when encapsulating IP multicast packets such that they remain IP multicast packets. This mode is necessary for IP multicast routing to correctly route IP multicast packets protected by IPsec.

2. Overview of IP Multicast Operation

IP multicasting is a means of sending a single packet to a "host group", a set of zero or more hosts identified by a single IP destination address. IP multicast packets are delivered to all members of the group either with "best-efforts" reliability [RFC1112] or as part of a reliable stream (e.g., NACK-Oriented Reliable Multicast (NORM) [RFC3940]).

A sender to an IP multicast group sets the destination of the packet to an IP address that has been allocated for IP multicast. Allocated IP multicast addresses are defined in [RFC3171], [RFC3306], and [RFC3307]. Potential receivers of the packet "join" the IP multicast group by registering with a network routing device ([RFC3376], [RFC3810]), signaling its intent to receive packets sent to a particular IP multicast group.

Network routing devices configured to pass IP multicast packets participate in multicast routing protocols (e.g., PIM-SM) [RFC4601]. Multicast routing protocols maintain state regarding which devices have registered to receive packets for a particular IP multicast group. When a router receives an IP multicast packet, it forwards a copy of the packet out of each interface for which there are known receivers.

3. Security Association Modes

IPsec supports two modes of use: transport mode and tunnel mode. In transport mode, IP Authentication Header (AH) [RFC4302] and IP Encapsulating Security Payload (ESP) [RFC4303] provide protection primarily for next layer protocols; in tunnel mode, AH and ESP are applied to tunneled IP packets.

A host implementation of IPsec using the multicast extensions MAY use either transport mode or tunnel mode to encapsulate an IP multicast packet. These processing rules are identical to the rules described in Section 4.1 of [RFC4301]. However, the destination address for the IPsec packet is an IP multicast address, rather than a unicast host address.

A security gateway implementation of IPsec MUST use a tunnel mode SA, for the reasons described in Section 4.1 of [RFC4301]. In particular, the security gateway needs to use tunnel mode to encapsulate incoming fragments, since IPsec cannot directly operate on fragments.

3.1. Tunnel Mode with Address Preservation

New (tunnel) header construction semantics are required when tunnel mode is used to encapsulate IP multicast packets that are to remain IP multicast packets. These semantics are due to the following unique requirements of IP multicast routing protocols (e.g., PIM-SM [RFC4601]). This document describes these new header construction semantics as "tunnel mode with address preservation", which is described as follows.

- When an IP multicast packet is received by a host or router, the destination address of the packet is compared to the local IP multicast state. If the (outer) destination IP address of an IP multicast packet is set to another IP address, the host or router receiving the IP multicast packet will not process it properly. Therefore, an IPsec security gateway needs to populate the multicast IP destination address in the outer header using the destination address from the inner header after IPsec tunnel encapsulation.

- IP multicast routing protocols typically create multicast distribution trees based on the source address as well as the group address. If an IPsec security gateway populates the (outer) source address of an IP multicast packet (with its own IP address, as called for in RFC 4301), the resulting IPsec-protected packet may fail Reverse Path Forwarding (RPF) checks performed by other routers. A failed RPF check may result in the packet being

dropped. To accommodate routing protocol RPF checks, the security gateway implementing the IPsec multicast extensions SHOULD populate the outer IP address from the original packet IP source address. However, it should be noted that a security gateway performing source address preservation will not receive ICMP Path MTU (PMTU) or other messages intended for the security gateway (triggered by packets that have had the outer IP source address set to that of the inner header). Security gateway applications not requiring source address preservation will be able to receive ICMP PMTU messages and process them as described in Section 6.1 of RFC 4301.

Because some applications of address preservation may require that only the destination address be preserved, specification of destination address preservation and source address preservation are separated in the above description. Destination address preservation and source address preservation attributes are described in the Group Security Policy Database (GSPD) (defined later in this document), and are copied into corresponding Security Association Database (SAD) entries.

Address preservation is applicable only for tunnel mode IPsec SAs that specify the IP version of the encapsulating header to be the same version as that of the inner header. When the IP versions are different, IP multicast packets can be encapsulated using a tunnel interface, for example as described in [RFC4891], where the tunnel is also treated as an interface by IP multicast routing protocols.

In summary, propagating both the IP source and destination addresses of the inner IP header into the outer (tunnel) header allows IP multicast routing protocols to route a packet properly when the packet is protected by IPsec. This result is necessary in order for the multicast extensions to allow a host or security gateway to provide IPsec services for IP multicast packets. This method of RFC 4301 tunnel mode is known as "tunnel mode with address preservation".

4. Security Association

4.1. Major IPsec Databases

The following sections describe the GKM subsystem and IPsec extension interactions with the IPsec databases. The major IPsec databases need expanded semantics to fully support multicast.

4.1.1. Group Security Policy Database (GSPD)

The Group Security Policy Database is a security policy database capable of supporting both unicast Security Associations as defined by RFC 4301 and the multicast extensions defined by this

specification. The GSPD is considered to be the SPD, with the addition of the semantics relating to the multicast extensions described in this section. Appendix B provides an example of an ASN.1 definition of a GSPD entry.

This document describes a new "address preservation" (AP) flag indicating that tunnel mode with address preservation is to be applied to a GSPD entry. The AP flag has two attributes: AP-L, used in the processing of the local tunnel address, and AP-R, used in the processing of the remote tunnel process. This flag is added to the GSPD "Processing info" field of the GSPD. The following text reproduced from Section 4.4.1.2 of RFC 4301 is amended to include this additional processing. (Note: for brevity, only the "Processing info" text related to tunnel processing has been reproduced.)

- o Processing info -- which action is required -- PROTECT, BYPASS, or DISCARD. There is just one action that goes with all the selector sets, not a separate action for each set. If the required processing is PROTECT, the entry contains the following information.
  - IPsec mode -- tunnel or transport
  - (if tunnel mode) local tunnel address -- For a non-mobile host, if there is just one interface, this is straightforward; if there are multiple interfaces, this must be statically configured. For a mobile host, the specification of the local address is handled externally to IPsec. If tunnel mode with address preservation is specified for the local tunnel address, the AP-L attribute is set to TRUE for the local tunnel address and the local tunnel address is unspecified. The presence of the AP-L attribute indicates that the inner IP header source address will be copied to the outer IP header source address during IP header construction for tunnel mode.
  - (if tunnel mode) remote tunnel address -- There is no standard way to determine this. See Section 4.5.3 of RFC 4301, "Locating a Security Gateway". If tunnel mode with address preservation is specified for the remote tunnel address, the AP-R attribute is set to TRUE for the remote tunnel address and the remote tunnel address is unspecified. The presence of the AP-R attribute indicates that the inner IP header destination address will be copied to the outer IP header destination address during IP header construction for tunnel mode.

This document describes unique directionality processing for GSPD entries with a remote IP multicast address. Since an IP multicast address must not be sent as the source address of an IP packet

[RFC1112], directionality of Local and Remote addresses and ports is maintained during incoming SPD-S and SPD-I checks rather than being swapped. Section 4.4.1 of RFC 4301 is amended as follows:

> Representing Directionality in an SPD Entry
>
> For traffic protected by IPsec, the Local and Remote address and ports in an SPD entry are swapped to represent directionality, consistent with IKE conventions. In general, the protocols that IPsec deals with have the property of requiring symmetric SAs with flipped Local/Remote IP addresses. However, SPD entries with a remote IP multicast address do not have their Local and Remote addresses and ports in an SPD entry swapped during incoming SPD-S and SPD-I checks.

A new Group Security Policy Database (GSPD) attribute is introduced: GSPD entry directionality. The following text is added to the bullet list of SPD fields described in Section 4.4.1.2 of RFC 4301.

> o Directionality -- can be one of three types: "symmetric", "sender only", or "receiver only". "Symmetric" indicates that a pair of SAs are to be created (one in each direction, as specified by RFC 4301). GSPD entries marked as "sender only" indicate that one SA is to be created in the outbound direction. GSPD entries marked as "receiver only" indicate that one SA is to be created in the inbound direction. GSPD entries marked as "sender only" or "receiver only" SHOULD support multicast IP addresses in their destination address selectors. If the processing requested is BYPASS or DISCARD and a "sender only" type is configured, the entry MUST be put in GSPD-O only. Reciprocally, if the type is "receiver only", the entry MUST go to GSPD-I only.

GSPD entries created by a GCKS may be assigned identical Security Parameter Indexes (SPIs) to SAD entries created by IKEv2 [RFC4306]. This is not a problem for the inbound traffic as the appropriate SAs can be matched using the algorithm described in Section 4.1 of RFC 4301. However, the outbound traffic needs to be matched against the GSPD selectors so that the appropriate SA can be created.

To facilitate dynamic group keying, the outbound GSPD MUST implement a policy action capability that triggers a GKM protocol registration exchange (as per Section 5.1 of [RFC4301]). For example, the Group Sender GSPD policy might trigger on a match with a specified multicast application packet that is entering the implementation via the protected interface or that is emitted by the implementation on the protected side of the boundary and directed toward the

unprotected interface. The ensuing Group Sender registration exchange would set up the Group Sender's outbound SAD entry that encrypts the multicast application's data stream. In the inverse direction, group policy may also set up an inbound IPsec SA.

At the Group Receiver endpoint(s), the IPsec subsystem MAY use GSPD policy mechanisms that initiate a GKM protocol registration exchange. One such policy mechanism might be on the detection of a device in the protected network joining a multicast group matching GSPD policy (e.g., by receiving a IGMP/MLD (Multicast Listener Discovery) join group message on a protected interface). The ensuing Group Receiver registration exchange would set up the Group Receiver's inbound SAD entry that decrypts the multicast application's data stream. In the inverse direction, the group policy may also set up an outbound IPsec SA (e.g., when supporting an ASM service model).

Note: A security gateway triggering on the receipt of unauthenticated messages arriving on a protected interface may result in early Group Receiver registration if the message is not the result of a device on the protected network actually wishing to join a multicast group. The unauthenticated messages will only cause the Group Receiver to register once; subsequent messages will have no effect on the Group Receiver.

The IPsec subsystem MAY provide GSPD policy mechanisms that automatically initiate a GKM protocol de-registration exchange. De-registration allows a GCKS to minimize exposure of the group's secret key by re-keying a group on a group membership change event. It also minimizes cost on a GCKS for those groups that maintain member state. One such policy mechanism could be the detection of IGMP/MLD leave group exchange. However, a security gateway Group Member would not initiate a GKM protocol de-registration exchange until it detects that there are no more receivers behind a protected interface.

Additionally, the GKM subsystem MAY set up the GSPD/SAD state information independent of the multicast application's state. In this scenario, the Group Owner issues management directives that tell the GKM subsystem when it should start GKM registration and de-registration protocol exchanges. Typically, the registration policy strives to make sure that the group's IPsec subsystem state is "always ready" in anticipation of the multicast application starting its execution.

4.1.2. Security Association Database (SAD)

The SAD contains an item describing whether tunnel or transport mode is applied to traffic on this SA. The text in RFC 4301 Section 4.4.2.1 is amended to describe address preservation.

- o IPsec protocol mode: tunnel or transport. Indicates which mode of AH or ESP is applied to traffic on this SA. When tunnel mode is specified, the data item also indicates whether or not address preservation is applied to the outer IP header. Address preservation MUST NOT be specified when the IP version of the encapsulating header and IP version of the inner header do not match. The local address, remote address, or both addresses MAY be marked as being preserved during tunnel encapsulation.

4.1.3. Group Peer Authorization Database (GPAD)

The multicast IPsec extensions introduce a new data structure called the Group Peer Authorization Database (GPAD). The GPAD is analogous to the PAD defined in RFC 4301. It provides a link between the GSPD and a Group Key Management (GKM) Subsystem. The GPAD embodies the following critical functions:

- o identifies a GCKS (or group of GCKS devices) that is authorized to communicate with this IPsec entity
- o specifies the protocol and method used to authenticate each GCKS
- o provides the authentication data for each GKCS
- o constrains the traffic selectors that can be asserted by a GCKS with regard to SA creation
- o constrains the types and values of Group Identifiers for which a GCKS is authorized to provide group policy

The GPAD provides these functions for a Group Key Management subsystem. The GPAD is not consulted by IKE or other authentication protocols that do not act as GKM protocols.

To provide these functions, the GPAD contains an entry for each GCKS that the IPsec entity is configured to contact. An entry contains one or more GCKS Identifiers, the authentication protocol (e.g., Group Domain of Interpretation (GDOI) or Group Secure Association Key Management Protocol (GSAKMP)), the authentication method used (e.g., certificates or pre-shared secrets), and the authentication data

(e.g., the pre-shared secret or trust anchor relative to which the peer's certificate will be validated). For certificate-based authentication, the entry also may provide information to assist in verifying the revocation status of the peer, e.g., a pointer to a Certificate Revocation List (CRL) repository or the name of an Online Certificate Status Protocol (OCSP) server associated with either the peer or the trust anchor associated with the peer. The entry also contains constraints a Group Member applies to the policy received from the GKCS.

4.1.3.1. GCKS Identifiers

GCKS Identifiers are used to identify one or more devices that are authorized to act as a GCKS for this group. GCKS Identifiers are specified as PAD entry IDs in Section 4.4.3.1 of RFC 4301 and follow the matching rules described therein.

4.1.3.2. GCKS Peer Authentication Data

Once a GPAD entry is located, it is necessary to verify the asserted identity, i.e., to authenticate the asserted GCKS Identifier. PAD authentication data types and semantics specified in Section 4.4.3.2 of RFC 4301 are used to authenticate a GCKS.

See GDOI [RFC3547] and GSAKMP [RFC4535] for details of how a GKM protocol performs peer authentication using certificates and pre-shared secrets.

4.1.3.3. Group Identifier Authorization Data

A Group Identifier is used by a GKM protocol to identify a particular group to a GCKS. A GPAD entry includes a Group Identifier to indicate that the GKCS Identifiers in the GPAD entry are authorized to act as a GCKS for the group.

The Group Identifier is an opaque byte string of IKE ID type Key ID that identifies a secure multicast group. The Group Identifier byte string MUST be at least four bytes long and less than 256 bytes long.

IKE ID types other than Key ID MAY be supported.

4.1.3.4. IPsec SA Traffic Selector Authorization Data

Once a GCKS is authenticated, the GCKS delivers IPsec SA policy to the Group Member. Before the Group Member accepts the IPsec SA Policy, the source and destination traffic selectors of the SA are compared to a set of authorized data flows. Each data flow includes a set of authorized source traffic selectors and a set of authorized

destination traffic selectors. Traffic selectors are represented as a set of IPv4 and/or IPv6 address ranges. (A peer may be authorized for both address types, so there MUST be provision for both v4 and v6 address ranges.)

4.1.3.5. How the GPAD Is Used

When a GKM protocol registration exchange is triggered, the Group Member and GCKS each assert their identity as a part of the exchange. Each GKM protocol registration exchange MUST use the asserted ID to locate an identity in the GPAD. The GPAD entry specifies the authentication method to be employed for the identified GCKS. The entry also specifies the authentication data that will be used to verify the asserted identity. This data is employed in conjunction with the specified method to authenticate the GCKS before accepting any group policy from the GCKS.

During the GKM protocol registration, a Group Member includes a Group Identifier. Before presenting that Group Identifier to the GCKS, a Group Member verifies that the GPAD entry for authenticated GCKS GPAD entry includes the Group Identifier. This ensures that the GCKS is authorized to provide policy for the Group.

When IPsec SA policy is received, each data flow is compared to the data flows in the GPAD entry. The Group Member accepts policy matching a data flow. Policy not matching a data flow is discarded, and the reason SHOULD be recorded in the audit log.

A GKM protocol may distribute IPsec SA policy to IPsec devices that have previously registered with it. The method of distribution is part of the GKM protocol and is outside the scope of this memo. When the IPsec device receives this new policy, it compares the policy to the data flows in the GPAD entry as described above.

4.2. Group Security Association (GSA)

An IPsec implementation supporting these extensions will support a number of Security Associations: one or more IPsec SAs plus one or more GKM SAs used to download the parameters that are used to create IPsec SAs. These SAs are collectively referred to as a Group Security Association (GSA) [RFC3740].

4.2.1. Concurrent IPsec SA Life Spans and Re-key Rollover

During a secure multicast group’s lifetime, multiple IPsec Group Security Associations can exist concurrently. This occurs principally due to two reasons:

- There are multiple Group Senders authorized in the group, each with its own IPsec SA, which maintains anti-replay state. A group that does not rely on IP security anti-replay services can share one IPsec SA for all of its Group Senders.

- The life spans of a Group Sender's two (or more) IPsec SAs are allowed to overlap in time so that there is continuity in the multicast data stream across group re-key events. This capability is referred to as "re-key rollover continuity".

The re-key continuity rollover algorithm depends on an IPsec SA management interface between the GKM subsystem and the IPsec subsystem. The IPsec subsystem MUST provide management interface mechanisms for the GKM subsystem to add IPsec SAs and to delete IPsec SAs. For illustrative purposes, this text defines the re-key rollover continuity algorithm in terms of two timer parameters that govern IPsec SA life spans relative to the start of a group re-key event. However, it should be emphasized that the GKM subsystem interprets the group's security policy to direct the correct timing of IPsec SA activation and deactivation. A given group policy may choose timer values that differ from those recommended by this text. The two re-key rollover continuity timer parameters are:

1. Activation Time Delay (ATD) - The ATD defines how long after the start of a re-key event to activate new IPsec SAs. The ATD parameter is expressed in units of seconds. Typically, the ATD parameter is set to the maximum time it takes to deliver a multicast message from the GCKS to all of the group's members. For a GCKS that relies on a Reliable Multicast Transport Protocol (RMTP), the ATD parameter could be set equal to the RTMP's maximum error recovery time. When an RMTP is not present, the ATD parameter might be set equal to the network's maximum multicast message delivery latency across all of the group's endpoints. The ATD is a GKM group policy parameter. This value SHOULD be configurable at the Group Owner management interface on a per group basis.

2. Deactivation Time Delay (DTD) - The DTD defines how long after the start of a re-key event to deactivate those IPsec SAs that are destroyed by the re-key event. The purpose of the DTD parameter is to minimize the residual exposure of a group's keying material after a re-key event has retired that keying material. The DTD is independent of, and should not to be confused with, the IPsec SA soft lifetime attribute. The DTD parameter is expressed in units of seconds. Typically, the DTD parameter would be set to the ADT plus the maximum time it takes to deliver a multicast message from the Group Sender to all of the group's members. For a Group Sender that relies on an RMTP, the DTD parameter could be set

equal to ADT plus the RMTP's maximum error recovery time. When an RMTP is not present, the DTD parameter might be set equal to ADT plus the network's maximum multicast message delivery latency across all of the group's endpoints. A GKM subsystem MAY implement the DTD as a group security policy parameter. If a GKM subsystem does not implement the DTD parameter, then other group security policy mechanisms MUST determine when to deactivate an IPsec SA.

Each group re-key multicast message sent by a GCKS signals the start of a new Group Sender IPsec SA time epoch, with each such epoch having an associated set of two IPsec SAs. Note that this document refers to re-key mechanisms as being multicast because of the inherent scalability of IP multicast distribution. However, there is no particular reason that re-keying mechanisms must be multicast. For example, [ZLLY03] describes a method of re-key employing both unicast and multicast messages.

The group membership interacts with these IPsec SAs as follows:

- As a precursor to the Group Sender beginning its re-key rollover continuity processing, the GCKS periodically multicasts a Re-Key Event (RKE) message to the group. The RKE multicast MAY contain group policy directives, new IPsec SA policy, and group keying material. In the absence of an RMTP, the GCKS may re-transmit the RKE a policy-defined number of times to improve the availability of re-key information. The GKM subsystem starts the ATD and DTD timers after it receives the last RKE re-transmission.

- The GKM subsystem interprets the RKE multicast to configure the group's GSPD/SAD with the new IPsec SAs. Each IPsec SA that replaces an existing SA is called a "leading edge" IPsec SA. The leading edge IPsec SA has a new Security Parameter Index (SPI) and its associated keying material, which keys it. For a time period of ATD seconds after the GCKS multicasts the RKE, a Group Sender does not yet transmit data using the leading edge IPsec SA. Meanwhile, other Group Members prepare to use this IPsec SA by installing the leading edge IPsec SAs to their respective GSPD/SAD.

- After waiting for the ATD period, such that all of the Group Members have received and processed the RKE message, the GKM subsystem directs the Group Sender to begin to transmit using the leading edge IPsec SA with its data encrypted by the new keying material. Only authorized Group Members can decrypt these IPsec SA multicast transmissions.

- The Group Sender's "trailing edge" SA is the oldest Security Association in use by the group for that sender. All authorized Group Members can receive and decrypt data for this SA, but the Group Sender does not transmit new data using the trailing edge IPsec SA after it has transitioned to the leading edge IPsec SA. The trailing edge IPsec SA is deleted by the group's GKM subsystems after the DTD time period has elapsed since the RKE transmission.

This re-key rollover strategy allows the group to drain its in-transit datagrams from the network while transitioning to the leading edge IPsec SA. Staggering the roles of each respective IPsec SA as described above improves the group's synchronization even when there are high network propagation delays. Note that due to group membership joins and leaves, each Group Sender IPsec SA time epoch may have a different group membership set.

It is a group policy decision whether the re-key event transition between epochs provides forward and backward secrecy. The group's re-key protocol keying material and algorithm (e.g., Logical Key Hierarchy; refer to [RFC2627] and Appendix A of [RFC4535]) enforces this policy. Implementations MAY offer a Group Owner management interface option to enable/disable re-key rollover continuity for a particular group. This specification requires that a GKM/IPsec implementation MUST support at least two concurrent IPsec SAs per Group Sender as well as this re-key rollover continuity algorithm.

4.3. Data Origin Authentication

As defined in [RFC4301], data origin authentication is a security service that verifies the identity of the claimed source of data. A Message Authentication Code (MAC) is often used to achieve data origin authentication for connections shared between two parties. However, typical MAC authentication methods using a single shared secret are not sufficient to provide data origin authentication for groups with more than two parties. With a MAC algorithm, every Group Member can use the MAC key to create a valid MAC tag, whether or not they are the authentic originator of the group application's data.

When the property of data origin authentication is required for an IPsec SA shared by more than two parties, an authentication transform where the receiver is assured that the sender generated that message should be used. Two possible algorithms are Timed Efficient Stream Loss-Tolerant Authentication (TESLA) [RFC4082] or RSA digital signature [RFC4359].

In some cases (e.g., digital signature authentication transforms), the processing cost of the algorithm is significantly greater than a Hashed Message Authentication Code (HMAC) authentication method. To

protect against denial-of-service attacks from a device that is not authorized to join the group, the IPsec SA using this algorithm may be encapsulated with an IPsec SA using a MAC authentication algorithm. However, doing so requires the packet to be sent across the IPsec boundary a second time for additional outbound processing on the Group Sender (see Section 5.1 of [RFC4301]) and a second time for inbound processing on Group Receivers (see Section 5.2 of [RFC4301]). This use of AH or ESP encapsulated within AH or ESP accommodates the constraint that AH and ESP define an Integrity Check Value (ICV) for only a single authenticator transform.

4.4. Group SA and Key Management

4.4.1. Co-Existence of Multiple Key Management Protocols

Often, the GKM subsystem will be introduced to an existent IPsec subsystem as a companion key management protocol to IKEv2 [RFC4306]. A fundamental GKM protocol IP security subsystem requirement is that both the GKM protocol and IKEv2 can simultaneously share access to a common Group Security Policy Database and Security Association Database. The mechanisms that provide mutually exclusive access to the common GSPD/SAD data structures are a local matter. This includes the GSPD-O cache and the GSPD-I cache. However, implementers should note that IKEv2 SPI allocation is entirely independent from GKM SPI allocation because Group Security Associations are qualified by a destination multicast IP address and may optionally have a source IP address qualifier. See Section 2.1 of [RFC4303] for further explanation.

The Peer Authorization Database does require explicit coordination between the GKM protocol and IKEv2. Section 4.1.3 describes these interactions.

5. IP Traffic Processing

Processing of traffic follows Section 5 of [RFC4301], with the additions described below when these IP multicast extensions are supported.

5.1. Outbound IP Traffic Processing

If an IPsec SA is marked as supporting tunnel mode with address preservation (as described in Section 3.1), either or both of the outer header source or destination addresses are marked as being preserved.

Header construction for tunnel mode is described in Section 5.1.2 of RFC 4301. The first bullet of that section is amended as follows:

- o If address preservation is not marked in the SAD entry for either the outer IP header Source Address or Destination Address, the outer IP header Source Address and Destination Address identify the "endpoints" of the tunnel (the encapsulator and decapsulator). If address preservation is marked for the IP header Source Address, it is copied from the inner IP header Source Address. If address preservation is marked for the IP header Destination Address, it is copied from the inner IP header Destination Address. The inner IP header Source Address and Destination Addresses identify the original sender and recipient of the datagram (from the perspective of this tunnel), respectively. Address preservation MUST NOT be marked when the IP version of the encapsulating header and IP version of the inner header do not match.

Note (3), regarding construction of tunnel addresses in Section 5.1.2.1 of RFC 4301, is amended as follows. (Note: for brevity, Note (3) of RFC 4301 is not reproduced in its entirety.)

(3) Unless marked for address preservation, Local and Remote addresses depend on the SA, which is used to determine the Remote address, which in turn determines which Local address (net interface) is used to forward the packet. If address preservation is marked for the Local address, it is copied from the inner IP header. If address preservation is marked for the Remote address, that address is copied from the inner IP header.

5.2. Inbound IP Traffic Processing

IPsec-protected packets generated by an IPsec device supporting these multicast extensions may (depending on its GSPD policy) populate an outer tunnel header with a destination address such that it is not addressed to an IPsec device. This requires an IPsec device supporting these multicast extensions to accept and process IP traffic that is not addressed to the IPsec device itself. The following additions to IPsec inbound IP traffic processing are necessary.

For compatibility with RFC 4301, the phrase "addressed to this device" is taken to mean packets with a unicast destination address belonging to the system itself, and also multicast packets that are received by the system itself. However, multicast packets not received by the IPsec device are not considered addressed to this device.

The discussion of processing inbound IP Traffic described in Section 5.2 of RFC 4301 is amended as follows.

The first dash in item 2 is amended as follows:

- If the packet appears to be IPsec protected and it is addressed to this device, or appears to be IPsec protected and is addressed to a multicast group, an attempt is made to map it to an active SA via the SAD. Note that the device may have multiple IP addresses that may be used in the SAD lookup, e.g., in the case of protocols such as SCTP.

A new item is added to the list between items 3a and 3b to describe processing of IPsec packets with destination address preservation applied:

3aa. If the packet is addressed to a multicast group and AH or ESP is specified as the protocol, the packet is looked up in the SAD. Use the SPI plus the destination or SPI plus destination and source addresses, as specified in Section 4.1. If there is no match, the packet is directed to SPD-I lookup. Note that if the IPsec device is a security gateway, and the SPD-I policy is to BYPASS the packet, a subsequent security gateway along the routed path of the multicast packet may decrypt the packet.

Figure 3 in RFC 4301 is updated to show the new processing path defined in item 3aa.

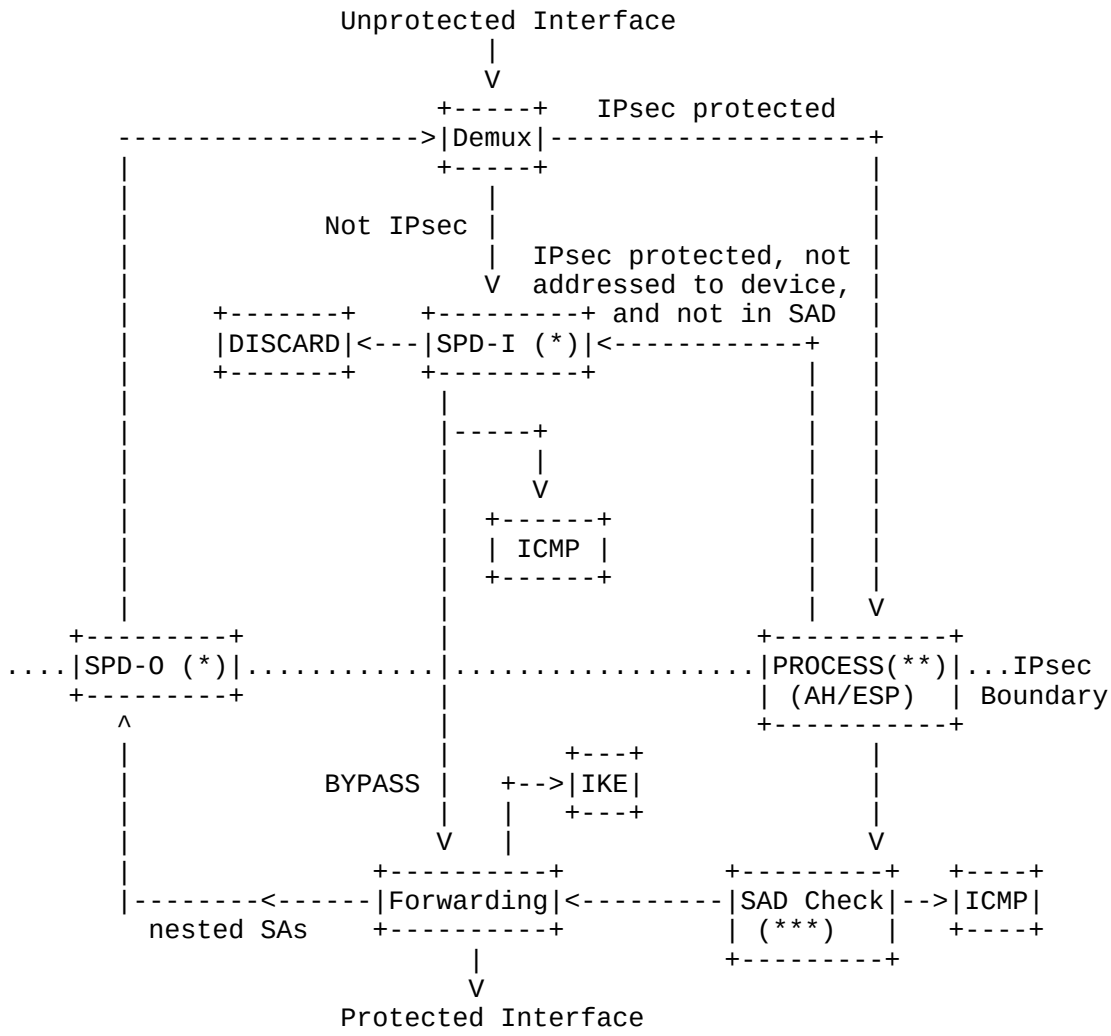

Figure 1. Processing Model for Inbound Traffic
(amending Figure 3 of RFC 4301)

The discussion of processing inbound IP traffic in Section 5.2 of RFC 4301 is amended to insert a new item 6 as follows.

6. If an IPsec SA is marked as supporting tunnel mode with address preservation (as described in Section 3.1), the marked address(es) (i.e., source and/or destination address(es)) in the outer IP header MUST be verified to be the same value(s) as in the inner IP header. If the addresses are not consistent, the IPsec system MUST discard the packet and treat the inconsistency as an auditable event.

6. Security Considerations

The IP security multicast extensions defined by this specification build on the unicast-oriented IP security architecture [RFC4301]. Consequently, this specification inherits many of RFC 4301's security considerations, and the reader is advised to review it as companion guidance.

6.1. Security Issues Solved by IPsec Multicast Extensions

The IP security multicast extension service provides the following network layer mechanisms for secure group communications:

- Confidentiality using a group shared encryption key.

- Group source authentication and integrity protection using a group shared authentication key.

- Group Sender data origin authentication using a digital signature, TESLA, or other mechanism.

- Anti-replay protection for a limited number of Group Senders using the ESP (or AH) sequence number facility.

- Filtering of multicast transmissions identified with a source address of systems that are not authorized by group policy to be Group Senders. This feature leverages the IPsec stateless firewall service (i.e., SPD-I and/or SDP-O entries with a packet disposition specified as DISCARD).

In support of the above services, this specification enhances the definition of the SPD, PAD, and SAD databases to facilitate the automated group key management of large-scale cryptographic groups.

6.2. Security Issues Not Solved by IPsec Multicast Extensions

As noted in Section 2.2. of RFC 4301, it is out of the scope of this architecture to defend the group's keys or its application data against attacks targeting vulnerabilities of the operating environment in which the IPsec implementation executes. However, it should be noted that the risk of attacks originating by an adversary in the network is magnified to the extent that the group keys are shared across a large number of systems.

The security issues that are left unsolved by the IPsec multicast extension service divide into two broad categories: outsider attacks and insider attacks.

6.2.1. Outsider Attacks

The IPsec multicast extension service does not defend against an adversary outside of the group who has:

- the capability to launch a multicast, flooding denial-of-service attack against the group, originating from a system whose IPsec subsystem does not filter the unauthorized multicast transmissions.

- compromised a multicast router, allowing the adversary to corrupt or delete all multicast packets destined for the group endpoints downstream from that router.

- captured a copy of an earlier multicast packet transmission and then replayed it to a group that does not have the anti-replay service enabled. Note that for a large-scale, any-source multicast group, it is impractical for the Group Receivers to maintain an anti-replay state for every potential Group Sender. Group policies that require anti-replay protection for a large-scale, any-source multicast group should consider an application layer multicast protocol that can detect and reject replays.

6.2.2. Insider Attacks

For large-scale groups, the IP security multicast extensions are dependent on an automated Group Key Management protocol to correctly authenticate and authorize trustworthy members in compliance to the group's policies. Inherent in the concept of a cryptographic group is a set of one or more shared secrets entrusted to all of the Group Members. Consequently, the service's security guarantees are no stronger than the weakest member admitted to the group by the GKM system. The GKM system is responsible for responding to compromised Group Member detection by executing a re-key procedure. The GKM re-keying protocol will expel the compromised Group Members and

distribute new group keying material to the trusted members. Alternatively, the group policy may require the GKM system to terminate the group.

In the event that an adversary has been admitted into the group by the GKM system, the following attacks are possible and can not be solved by the IPsec multicast extension service:

- The adversary can disclose the secret group key or group data to an unauthorized party outside of the group. After a group key or data compromise, cryptographic methods such as traitor tracing or watermarking can assist in the forensics process. However, these methods are outside the scope of this specification.

- The insider adversary can forge packet transmissions that appear to be from a peer Group Member. To defend against this attack, for those Group Sender transmissions that merit the overhead, the group policy can require the Group Sender to multicast packets using the data origin authentication service.

- If the group's data origin authentication service uses digital signatures, then the insider adversary can launch a computational resource denial-of-service attack by multicasting bogus signed packets.

6.3. Implementation or Deployment Issues that Impact Security

6.3.1. Homogeneous Group Cryptographic Algorithm Capabilities

The IP security multicast extensions service can not defend against a poorly considered group security policy that allows a weaker cryptographic algorithm simply because all of the group's endpoints are known to support it. Unfortunately, large-scale groups can be difficult to upgrade to the current best-in-class cryptographic algorithms. One possible approach to solving many of these problems is the deployment of composite groups that can straddle heterogeneous groups [COMPGRP]. A standard solution for heterogeneous groups is an activity for future standardization. In the interim, synchronization of a group's cryptographic capabilities could be achieved using a secure and scalable software distribution management tool.

6.3.2. Groups that Span Two or More Security Policy Domains

Large-scale groups may span multiple legal jurisdictions (e.g., countries) that enforce limits on cryptographic algorithms or key strengths. As currently defined, the IPsec multicast extension service requires a single group policy per group. As noted above, this problem remains an area for future standardization.

6.3.3. Source-Specific Multicast Group Sender Transient Locators

A Source Specific Multicast (SSM) Group Sender's source IP address can dynamically change during a secure multicast group's lifetime. Examples of the events that can cause the Group Sender's source address to change include but are not limited to NAT, a mobility-induced change in the care-of-address, and a multi-homed host using a new IP interface. The change in the Group Sender's source IP address will cause GSPD entries related to that multicast group to become out of date with respect to the group's multicast routing state. In the worst case, there is a risk that the Group Sender's data originating from a new source address will be BYPASS processed by a security gateway. If this scenario was not anticipated, then it could leak the group's data. Consequently, it is recommended that SSM secure multicast groups have a default DISCARD policy for all unauthorized Group Sender source IP addresses for the SSM group's destination IP address.

7. Acknowledgements

The authors wish to thank Steven Kent, Russ Housley, Pasi Eronen, and Tero Kivinen for their helpful comments.

The "Guidelines for Writing RFC Text on Security Considerations" [RFC3552] was consulted to develop the Security Considerations section of this memo.

8. References

8.1. Normative References

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

8.2. Informative References

[COMPGRP] Gross G. and H. Cruickshank, "Multicast IP Security Composite Cryptographic Groups", Work in Progress, February 2007.

[RFC2526] Johnson, D. and S. Deering, "Reserved IPv6 Subnet Anycast Addresses", RFC 2526, March 1999.

[RFC2627] Wallner, D., Harder, E., and R. Agee, "Key Management for Multicast: Issues and Architectures", RFC 2627, June 1999.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3171] Albanna, Z., Almeroth, K., Meyer, D., and M. Schipper, "IANA Guidelines for IPv4 Multicast Address Assignments", BCP 51, RFC 3171, August 2001.

[RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, August 2002.

[RFC3307] Haberman, B., "Allocation Guidelines for IPv6 Multicast Addresses", RFC 3307, August 2002.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC3569] Bhattacharyya, S., Ed., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, July 2003.

[RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security Architecture", RFC 3740, March 2004.

[RFC3810] Vida, R., Ed., and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC3940] Adamson, B., Bormann, C., Handley, M., and J. Macker, "Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol", RFC 3940, November 2004.

[RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, April 2005.

[RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, June 2005.

[RFC4306] Kaufman, C., Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4359] Weis, B., "The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4359, January 2006.

[RFC4535] Harney, H., Meth, U., Colegrove, A., and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol", RFC 4535, June 2006.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC4891] Graveman, R., Parthasarathy, M., Savola, P., and H. Tschofenig, "Using IPsec to Secure IPv6-in-IPv4 Tunnels", RFC 4891, May 2007.

[ZLLY03] Zhang, X., et al., "Protocol Design for Scalable and Reliable Group Rekeying", IEEE/ACM Transactions on Networking (TON), Volume 11, Issue 6, December 2003.

Appendix A. Multicast Application Service Models

The vast majority of secure multicast applications can be catalogued by their service model and accompanying intra-group communication patterns. Both the Group Key Management (GKM) subsystem and the IPsec subsystem MUST be able to configure the GSPD/SAD security policies to match these dominant usage scenarios. The GSPD/SAD policies MUST include the ability to configure both Any-Source Multicast groups and Source-Specific Multicast groups for each of these service models. The GKM subsystem management interface MAY include mechanisms to configure the security policies for service models not identified by this standard.

A.1. Unidirectional Multicast Applications

Multimedia content-delivery multicast applications that do not have congestion notification or re-transmission error-recovery mechanisms are inherently unidirectional. RFC 4301 only defines bi-directional unicast traffic selectors (as per RFC 4301, Sections 4.4.1 and 5.1 with respect to traffic selector directionality). The GKM subsystem requires that the IPsec subsystem MUST support unidirectional SPD entries, which cause a Group Security Association (GSA) to be installed in only one direction. Multicast applications that have only one Group Member authorized to transmit can use this type of Group Security Association to enforce that group policy. In the inverse direction, the GSA does not have an SAD entry, and the GSPD configuration is optionally set up to discard unauthorized attempts to transmit unicast or multicast packets to the group.

The GKM subsystem’s management interface MUST have the ability to set up a GKM subsystem group having a unidirectional GSA security policy.

A.2. Bi-Directional Reliable Multicast Applications

Some secure multicast applications are characterized as one Group Sender to many receivers but have inverse data flows required by a reliable multicast transport protocol (e.g., NORM). In such applications, the data flow from the sender is multicast and the inverse flow from the Group’s Receivers is unicast to the sender. Typically, the inverse data flows carry error repair requests and congestion control status.

For such applications, it is advantageous to use the same IPsec SA for protection of both unicast and multicast data flows. This does introduce one risk: the IKEv2 application may choose the same SPI for receiving unicast traffic as the GCKS chooses for a group IPsec SA covering unicast traffic. If both SAs are installed in the SAD, the SA lookup may return the wrong SPI as the result of an SA lookup. To

avoid this problem, IPsec SAs installed by the GKM SHOULD use the 2-tuple {destination IP address, SPI} to identify each IPsec SA. In addition, the GKM SHOULD use a unicast destination IP address that does not match any destination IP address in use by an IKEv2 unicast IPsec SA. For example, suppose a Group Member is using both IKEv2 and a GKM protocol, and the group security policy requires protecting the NORM inverse data flows as described above. In this case, group policy SHOULD allocate and use a unique unicast destination IP address representing the NORM Group Sender. This address would be configured in parallel to the Group Sender's existing IP addresses. The GKM subsystems at both the NORM Group Sender and Group Receiver endpoints would install the IPsec SA, protecting the NORM unicast messages such that the SA lookup uses the unicast destination address as well as the SPI.

The GSA SHOULD use IPsec anti-replay protection service for the sender's multicast data flow to the group's Receivers. Because of the scalability problem described in the next section, it is not practical to use the IPsec anti-replay service for the unicast inverse flows. Consequently, in the inverse direction, the IPsec anti-replay protection MUST be disabled. However, the unicast inverse flows can use the group's IPsec group authentication mechanism. The Group Receiver's GSPD entry for this GSA SHOULD be configured to only allow a unicast transmission to the sender node rather than a multicast transmission to the whole group.

If an ESP digital signature authentication is available (e.g., RFC 4359), source authentication MAY be used to authenticate a receiver node's transmission to the sender. The GKM protocol MUST define a key management mechanism for the Group Sender to validate the asserted signature public key of any receiver node without requiring that the sender maintain state about every Group Receiver.

This multicast application service model is RECOMMENDED because it includes congestion control feedback capabilities. Refer to [RFC2914] for additional background information.

The GKM subsystem's Group Owner management interface MUST have the ability to set up a symmetric GSPD entry and one Group Sender. The management interface SHOULD be able to configure a group to have at least 16 concurrent authorized senders, each with their own GSA anti-replay state.

A.3. Any-To-Many Multicast Applications

Another family of secure multicast applications exhibits an "any-to-many" communications pattern. A representative example of such an application is a videoconference combined with an electronic whiteboard.

For such applications, all (or a large subset) of the Group Members are authorized multicast senders. In such service models, creating a distinct IPsec SA with anti-replay state for every potential sender does not scale to large groups. The group SHOULD share one IPsec SA for all of its senders. The IPsec SA SHOULD NOT use the IPsec anti-replay protection service for the sender's multicast data flow to the Group Receivers.

The GKM subsystem's management interface MUST have the ability to set up a group having an Any-To-Many Multicast GSA security policy.

Appendix B. ASN.1 for a GSPD Entry

This appendix describes an additional way to describe GSPD entries, as defined in Section 4.1.1. It uses ASN.1 syntax that has been successfully compiled. This syntax is merely illustrative and need not be employed in an implementation to achieve compliance. The GSPD description in Section 4.1.1 is normative. As shown in Section 4.1.1, the GSPD updates the SPD and thus this appendix updates the SPD object identifier.

B.1. Fields Specific to a GSPD Entry

The following fields summarize the fields of the GSPD that are not present in the SPD.

- direction (in IPsecEntry)
- DirectionFlags
- noswap (in SelectorList)
- ap-l, ap-r (in TunnelOptions)

B.2. SPDModule

```
  SPDModule

  {iso(1) org (3) dod (6) internet (1) security (5) mechanisms (5)
   ipsec (8) asn1-modules (3) spd-module (1) }

     DEFINITIONS IMPLICIT TAGS ::=

     BEGIN

     IMPORTS
         RDNSequence FROM PKIX1Explicit88
           { iso(1) identified-organization(3)
             dod(6) internet(1) security(5) mechanisms(5) pkix(7)
             id-mod(0) id-pkix1-explicit(18) } ;

     -- An SPD is a list of policies in decreasing order of preference
     SPD ::= SEQUENCE OF SPDEntry

     SPDEntry ::= CHOICE {
         iPsecEntry       IPsecEntry,               -- PROTECT traffic
         bypassOrDiscard  [0] BypassOrDiscardEntry } -- DISCARD/BYPASS

     IPsecEntry ::= SEQUENCE {       -- Each entry consists of
         name         NameSets OPTIONAL,
         pFPs         PacketFlags,    -- Populate from packet flags
                            -- Applies to ALL of the corresponding
                            -- traffic selectors in the SelectorLists
         direction    DirectionFlags, -- SA directionality
         condition    SelectorLists,  -- Policy "condition"
         processing   Processing      -- Policy "action"
         }

     BypassOrDiscardEntry ::= SEQUENCE {
         bypass       BOOLEAN,        -- TRUE BYPASS, FALSE DISCARD
         condition    InOutBound }

     InOutBound ::= CHOICE {
         outbound     [0] SelectorLists,
         inbound      [1] SelectorLists,
         bothways     [2] BothWays }
```

```
BothWays ::= SEQUENCE {
    inbound     SelectorLists,
    outbound    SelectorLists }

NameSets ::= SEQUENCE {
    passed      SET OF Names-R,  -- Matched to IKE ID by
                                 -- responder
    local       SET OF Names-I } -- Used internally by IKE
                                 -- initiator

Names-R ::= CHOICE {                    -- IKEv2 IDs
    dName       RDNSequence,            -- ID_DER_ASN1_DN
    fqdn        FQDN,                   -- ID_FQDN
    rfc822      [0] RFC822Name,         -- ID_RFC822_ADDR
    keyID       OCTET STRING }          -- KEY_ID

Names-I ::= OCTET STRING       -- Used internally by IKE
                               -- initiator

FQDN ::= IA5String

RFC822Name ::= IA5String

PacketFlags ::= BIT STRING {
            -- if set, take selector value from packet
            -- establishing SA
            -- else use value in SPD entry
    localAddr  (0),
    remoteAddr (1),
    protocol   (2),
    localPort  (3),
    remotePort (4)  }

DirectionFlags ::= BIT STRING {
            -- if set, install SA in the specified
            -- direction. symmetric policy is
            -- represented by setting both bits
    inbound   (0),
    outbound  (1)  }

SelectorLists ::= SET OF SelectorList

SelectorList ::= SEQUENCE {
    localAddr   AddrList,
    remoteAddr  AddrList,
    protocol    ProtocolChoice,
    noswap      BOOLEAN }  -- Do not swap local and remote
                           -- addresses and ports on incoming
```

```
                              -- SPD-S and SPD-I checks

     Processing ::= SEQUENCE {
         extSeqNum   BOOLEAN, -- TRUE 64 bit counter, FALSE 32 bit
         seqOverflow BOOLEAN, -- TRUE rekey, FALSE terminate & audit
         fragCheck   BOOLEAN, -- TRUE stateful fragment checking,
                              -- FALSE no stateful fragment checking
         lifetime    SALifetime,
         spi         ManualSPI,
         algorithms  ProcessingAlgs,
         tunnel      TunnelOptions OPTIONAL } -- if absent, use
                                              -- transport mode

     SALifetime ::= SEQUENCE {
         seconds   [0] INTEGER OPTIONAL,
         bytes     [1] INTEGER OPTIONAL }

     ManualSPI ::= SEQUENCE {
         spi     INTEGER,
         keys    KeyIDs }

     KeyIDs ::= SEQUENCE OF OCTET STRING

     ProcessingAlgs ::= CHOICE {
         ah          [0] IntegrityAlgs,  -- AH
         esp         [1] ESPAlgs}        -- ESP

     ESPAlgs ::= CHOICE {
         integrity       [0] IntegrityAlgs,       -- integrity only
         confidentiality [1] ConfidentialityAlgs, -- confidentiality
                                                  -- only
         both            [2] IntegrityConfidentialityAlgs,
         combined        [3] CombinedModeAlgs }

     IntegrityConfidentialityAlgs ::= SEQUENCE {
         integrity       IntegrityAlgs,
         confidentiality ConfidentialityAlgs }

     -- Integrity Algorithms, ordered by decreasing preference
     IntegrityAlgs ::= SEQUENCE OF IntegrityAlg

     -- Confidentiality Algorithms, ordered by decreasing preference
     ConfidentialityAlgs ::= SEQUENCE OF ConfidentialityAlg
```

```
-- Integrity Algorithms
IntegrityAlg ::= SEQUENCE {
    algorithm   IntegrityAlgType,
    parameters  ANY -- DEFINED BY algorithm -- OPTIONAL }

IntegrityAlgType ::= INTEGER {
    none              (0),
    auth-HMAC-MD5-96  (1),
    auth-HMAC-SHA1-96 (2),
    auth-DES-MAC      (3),
    auth-KPDK-MD5     (4),
    auth-AES-XCBC-96  (5)
--  tbd (6..65535)
    }

-- Confidentiality Algorithms
ConfidentialityAlg ::= SEQUENCE {
    algorithm   ConfidentialityAlgType,
    parameters  ANY -- DEFINED BY algorithm -- OPTIONAL }

ConfidentialityAlgType ::= INTEGER {
    encr-DES-IV64   (1),
    encr-DES        (2),
    encr-3DES       (3),
    encr-RC5        (4),
    encr-IDEA       (5),
    encr-CAST       (6),
    encr-BLOWFISH   (7),
    encr-3IDEA      (8),
    encr-DES-IV32   (9),
    encr-RC4       (10),
    encr-NULL      (11),
    encr-AES-CBC   (12),
    encr-AES-CTR   (13)
--  tbd (14..65535)
    }

CombinedModeAlgs ::= SEQUENCE OF CombinedModeAlg

CombinedModeAlg ::= SEQUENCE {
    algorithm   CombinedModeType,
    parameters  ANY -- DEFINED BY algorithm -- }
                    -- defined outside
                    -- of this document for AES modes.
```

```
CombinedModeType ::= INTEGER {
    comb-AES-CCM    (1),
    comb-AES-GCM    (2)
--  tbd (3..65535)
    }

TunnelOptions ::= SEQUENCE {
    dscp        DSCP,
    ecn         BOOLEAN,    -- TRUE Copy CE to inner header
    ap-l        BOOLEAN,    -- TRUE Copy inner IP header
                            -- source address to outer
                            -- IP header source address
    ap-r        BOOLEAN,    -- TRUE Copy inner IP header
                            -- destination address to outer
                            -- IP header destination address
    df          DF,
    addresses   TunnelAddresses }

TunnelAddresses ::= CHOICE {
    ipv4        IPv4Pair,
    ipv6        [0] IPv6Pair }

IPv4Pair ::= SEQUENCE {
    local       OCTET STRING (SIZE(4)),
    remote      OCTET STRING (SIZE(4)) }

IPv6Pair ::= SEQUENCE {
    local       OCTET STRING (SIZE(16)),
    remote      OCTET STRING (SIZE(16)) }

DSCP ::= SEQUENCE {
    copy      BOOLEAN, -- TRUE copy from inner header
                       -- FALSE do not copy
    mapping   OCTET STRING OPTIONAL} -- points to table
                                     -- if no copy

DF ::= INTEGER {
    clear   (0),
    set     (1),
    copy    (2) }

ProtocolChoice::= CHOICE {
    anyProt  AnyProtocol,              -- for ANY protocol
    noNext   [0] NoNextLayerProtocol,  -- has no next layer
                                       -- items
    oneNext  [1] OneNextLayerProtocol, -- has one next layer
                                       -- item
```

```
        twoNext  [2] TwoNextLayerProtocol, -- has two next layer
                                           -- items
        fragment FragmentNoNext }          -- has no next layer
                                           -- info

    AnyProtocol ::= SEQUENCE {
        id          INTEGER (0),    -- ANY protocol
        nextLayer   AnyNextLayers }

    AnyNextLayers ::= SEQUENCE {      -- with either
        first       AnyNextLayer,     -- ANY next layer selector
        second      AnyNextLayer }    -- ANY next layer selector

    NoNextLayerProtocol ::= INTEGER (2..254)

    FragmentNoNext ::= INTEGER (44)   -- Fragment identifier

    OneNextLayerProtocol ::= SEQUENCE {
        id          INTEGER (1..254),   -- ICMP, MH, ICMPv6
        nextLayer   NextLayerChoice }   -- ICMP Type*256+Code
                                        -- MH   Type*256

    TwoNextLayerProtocol ::= SEQUENCE {
        id          INTEGER (2..254),   -- Protocol
        local       NextLayerChoice,    -- Local and
        remote      NextLayerChoice }   -- Remote ports

    NextLayerChoice ::= CHOICE {
        any         AnyNextLayer,
        opaque      [0] OpaqueNextLayer,
        range       [1] NextLayerRange }

    -- Representation of ANY in next layer field
    AnyNextLayer ::= SEQUENCE {
        start       INTEGER (0),
        end         INTEGER (65535) }

    -- Representation of OPAQUE in next layer field.
    -- Matches IKE convention
    OpaqueNextLayer ::= SEQUENCE {
        start       INTEGER (65535),
        end         INTEGER (0) }

    -- Range for a next layer field
    NextLayerRange ::= SEQUENCE {
        start       INTEGER (0..65535),
        end         INTEGER (0..65535) }
```

```
-- List of IP addresses
AddrList ::= SEQUENCE {
    v4List      IPv4List OPTIONAL,
    v6List      [0] IPv6List OPTIONAL }

-- IPv4 address representations
IPv4List ::= SEQUENCE OF IPv4Range

IPv4Range ::= SEQUENCE {    -- close, but not quite right ...
    ipv4Start   OCTET STRING (SIZE (4)),
    ipv4End     OCTET STRING (SIZE (4)) }

-- IPv6 address representations
IPv6List ::= SEQUENCE OF IPv6Range

IPv6Range ::= SEQUENCE {    -- close, but not quite right ...
    ipv6Start   OCTET STRING (SIZE (16)),
    ipv6End     OCTET STRING (SIZE (16)) }

END
```

Authors' Addresses

Brian Weis
Cisco Systems
170 W. Tasman Drive,
San Jose, CA 95134-1706
USA

Phone: +1-408-526-4796
EMail: bew@cisco.com

George Gross
Secure Multicast Networks LLC
977 Bates Road
Shoreham, VT 05770
USA

Phone: +1-802-897-5339
EMail: gmgross@securemulticast.net

Dragan Ignjatic
Polycom
Suite 200
3605 Gilmore Way
Burnaby, BC V5G 4X5
Canada

Phone: +1-604-453-9424
EMail: dignjatic@polycom.com