Internet Research Task Force
INTERNET-DRAFT
draft-ietf-msec-gspt-02.txt

Thomas Hardjono (VeriSign)
Hugh Harney (Sparta)
Pat McDaniel (U. Michigan)
Andrea Colgrove (Sparta)
Pete Dinsmore (NAI)

18 August 2003

Expires December 2003

The MSEC Group Security Policy Token

<draft-ietf-msec-gspt-02.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

Abstract

This document provides a definition for Group Security Policy, describes the set of elements that make-up an instance of group policy for a given group, and provides an explanation of the intended functions of each element of a group security policy as expressed in the form of the Policy Token or Policy Certificate.

1. Introduction

Group communications, also commonly called multicast, refers to communications in a group where the messages can be sent by any member and are received by all members. The applications and encryption can be implemented in a variety of ways and at numerous levels on the network stack. They range from mailing lists to conference calls to IP Multicasting. Often the need for data protection arises, which requires

the group to handle the messages in a consistently secure manner. To

accomplish this, cryptographic mechanisms and security policy must be shared and supported by the group as a whole. Because of this, special problems arise in managing the cryptographic and policy material as it changes or as the group changes.

2. Framework for Group Security Policy

Group Security Policy represents an important building block within the Secure Multicast Group Framework of [HCBD00]. The Framework of [HCBD00] is broad in the sense that it incorporates entities, functions and interfaces encompassing all aspects of secure groups.

2.1 The Secure Multicast Group Framework

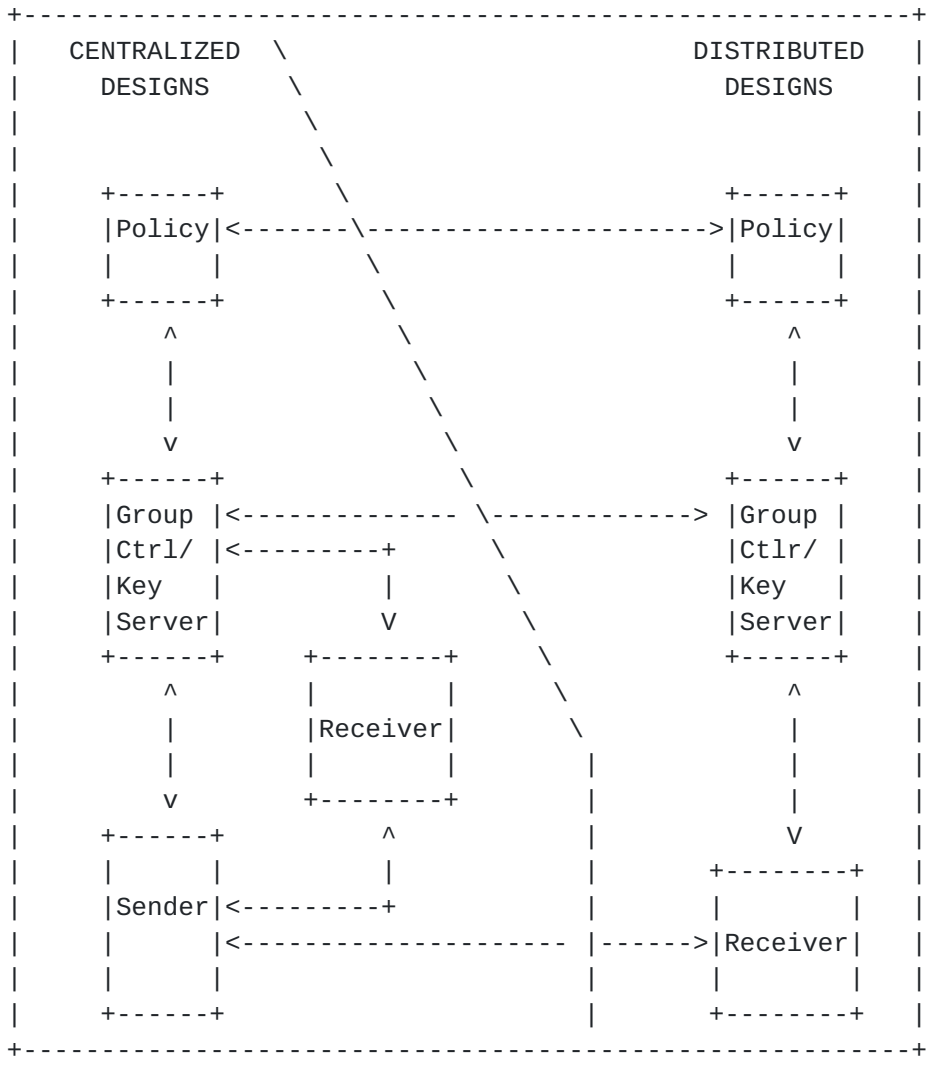

FIGURE 1: Secure Multicast Group Reference Framework

Figure 1 shows the Secure Multicast Group Reference Framework of functional entities and the interfaces between them [HCBD00]. These entities and interfaces implement a secure group, which is defined as a group of principals that share a secret. Secure groups are needed by unicast applications as well as multicast applications (single-source and multiple-source). The framework of Figure 1 identifies three group key management entities, namely the "Group Controller and Key Server" (GCKS) and two types of Members ("Receiver" and "Sender"). The GCKS entity embodies both the physical entity and functions of the group controller and the key server [RFC2093, RFC2094, RFC2627, OFT]. The Member belongs to one or more groups and may exist at different layers (e.g. user, host, process).

These entities (namely the GCKS, Sender and Receivers) represents the entities upon which Group Security Policy immediately applies. The policies exist, among others, to regulate behavior of entities that make-up a secure group.

2.2 Group Security Policy (GSP) Framework

The broad Reference Framework of [HCBD00] does not (and was not intended to) provide a policy-specific view of group security. Hence, to that extent a further refinement of the Reference Framework is provided in the following (Figure 2).

In Figure 2, the Group Owner is the entity defined to initiate the group and set the policies for the group. As such, the entity is the owner of the group and of all the information that pertains to the group.

>From the perspective of verification of Policy Tokens, the Group Owner is the root of all certificates that is used to verify Policy Tokens and certificates that delegate authority to service entities such as GCKSs or other intermediary entities. This function is tied closely to the fact of the Group Owner being the source of all authorizations for group actions carried-out by group members. The Group Owner being the source of authorization, is derived from owning or having ultimate responsibility for the data.

>From the perspective of access control to information about the group, the Group Owner determines pieces of information that are accessible service-entities (such as GCKS and Policy Repositories), group members and external entities. To this end, the Group Owner also decides form and content of group announcements made through the appropriate announcement protocols. Such information may consists of certain parts of the Policy Token or the entire Policy Token itself. Similarly, the Group Owner determines the information pertaining to a group that is

stored in the Policy Repository.

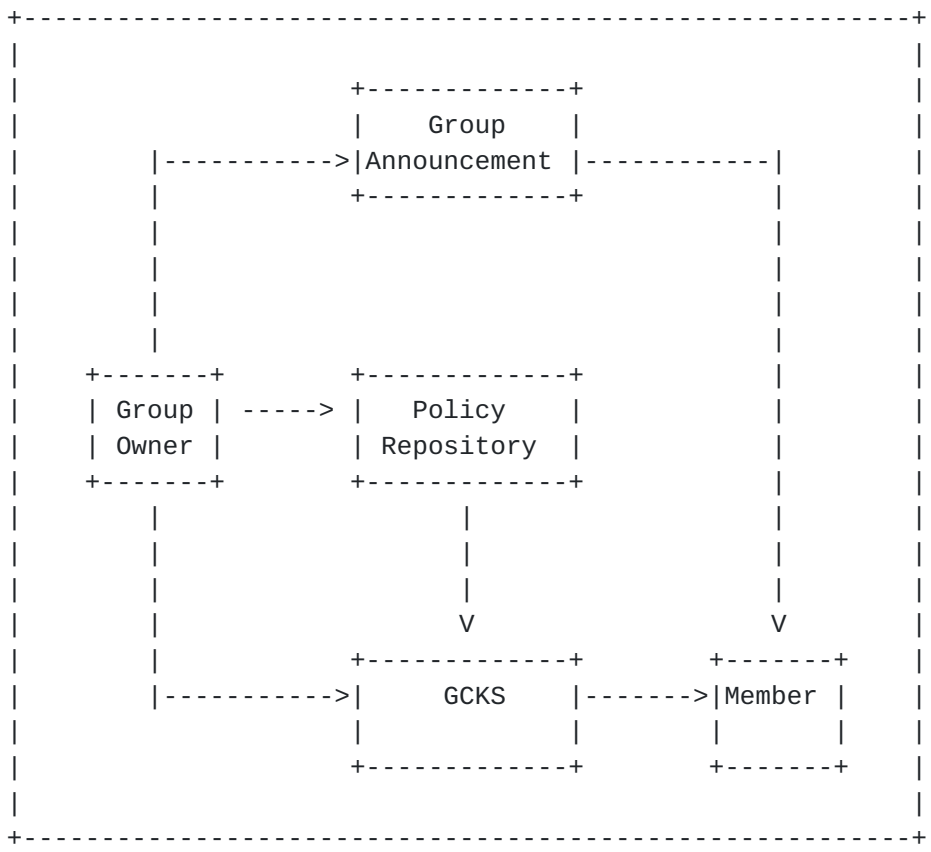

Figure 2: Group Security Policy Framework

When a (candidate) member wishes to find-out about a given group, it must learn about the existence of the group through the Group Announcement facility. Each group is associated with a unique Group Announcements and is identified in the Group Announcement through the Group IG (GID). Furthermore, announcements for different groups may carry different amounts of information regarding the groups in question.

2.3 Announcement of Group Policy Information

An important aspect of secure group communications is the availability of enough information for potential newcomers to decide as to whether they have sufficient permissions and suitable resources (e.g. crypto ability) to join the group.

The level of availability of such information is highly-dependent on the specific application employing secure groups. In some applications, all the required permissions and resources maybe pre-published in a

publicly-available location, protected using the group owner's digital

signature. In other applications, perhaps only limited information is published (e.g. required certificate-class of newcomer) to allow newcomers to decide whether or not to pursue further the act of joining the group.

In this document, we do not address the issue of which parts of the Group Policy Token are announced or be published to prospective newcomers.

3. Elements of Group Security Policy

Security-related policies for groups of users and for group communications are inherently more complex compared to policies for pair-wise (unicast) secure communications between two end-points. This complexity arises due to the fact that multi-party interaction allows for a number of situations and conditions to arise which are simply non-existent in the two-party interaction. Furthermore, in multi-party communications an error or a security breach cause by one party may have impact on the other parties in the group. In the two-party scenario, errors or security breaches may be dealt with simply by terminating the communication and restarting it.

The complex nature of group communications requires an equally complex set of elements that make-up the Group Security Policy. These elements (or categories in [polreq-draft]) specify the policies that are to be followed by members of the group, and they consist of the following:

a. Policy Identification

A group must have some means by which it can identify an instance of Group Security Policy in an unambiguous manner. Failure to correctly identify the group policies, messages, and participants can lead to incorrect and insecure operation. In the simplest form, an instance of a Policy Token must be associated with a unique Group Identifier (GID) expressly found inside the Policy Token.

b. Authorization for Group Actions

A Group Security Policy must identify the entities allowed to perform actions that affect group members. Group authorization partially determines the trust embodied by the group as a whole, by defining the parties or entities that are allowed to participate in group activities. Because of the wide range of expected environments, flexible identification of entity authorizations is highly desirable. Authorization given to an entity must be shown as being true and

authentic coming from another entity that bequeathed that authority.

c. Access Control to Group Information

Access control policy defines the entities that will have authorization to hold the key protecting the group data.

d. Mechanisms for Group Security Services

Identification of the security services used to support group communication is required. For example, policy must state the algorithms used to derive session keys and the types of data transforms to be applied to the group content. Each security service can have parameters and policies specific to its implementation. Thus, for forward compatibility with new approaches, service definitions should be extensible.

Full specification of:
- the group establishment mechanism
- the data protection mechanism
- the group management mechanism

is necessary to establish that the entire group SA is adequate to protect the data. Weakness in any one part will effect the security of the other parts.

e. Verification of Group Security Policy

Each policy must present evidence of its validity. The means by which the origin, integrity, and freshness of the policy is asserted (for example, via digital signature) must be known by each group member prior to its acquisition. In the simplest form, this consists of the Policy Token being digitally-signed by the entity authorized to issue the Group Security Policy.

4. Group Policy Token

The Group Policy Token is comprised of five fields corresponding to the identification of the group, the authorizations in it, the access control policies governing the group, the mechanisms supporting secure communications, and the authentication of the contents of the token.

Figure 3: Group Policy Token Overall Structure

Each of the fields of the GSAKMP Policy Token is further expanded in following sections. The specific data formats can be seen in the ASN.1 Policy Token Specification in Appendix A.

4.1 Token ID Field and Sub-Fields

The Token ID Field explicitly identifies the protocol family to which the Group Policy Token belongs (e.g. GSAKMP, GDOI). The Token ID Field consists of a Version number indicating Token version, Protocol ID indicating GSAKMP, Group ID consisting of IP Addresses and serial numbers, and a Timestamp.

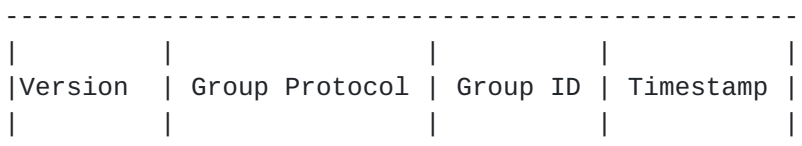

Figure 4: Token ID sub-fields

The timestamp sub field of the TokenID is of particular interest in the prevention of replay attacks. A replay attack is when an adversary inserts an authenticated message or portion of a message into a new run of a protocol. The timestamp sub-field helps to prevent such an attack. The receiver of a new token can verify that the timestamp is both appropriate for the policy token and generated at a time later than the timestamp on any previous Policy Tokens for that group. This will prevent an adversary from successfully replaying an old token and having it be accepted as a new one. Additionally, an optional expiration time field will limit the validity window of the token.

4.2 Authorization Field and Sub-Fields

The authorization field allows group members to ensure that security relevant actions are being performed by authorized group entities. This ensures that a rogue group member does not mislead other group members into an insecure action.

The Authorization Field identifies those who are authorized to act in the special roles of Group Owner, GCKS, and Re-key GCKS. This identification may be done explicitly as shown below, where the fields contain actual identifiers, or implicitly, using sets of rules to define the policy allowing one to assume the roles listed. For example, a policy rule might state that anyone in a particular domain except two people is authorized to act as a Key Server. Such a rule might be stated as "include {/o=acme/ou=responsible}, not {/o=acme/ou=responsible/s=bob, {/o=acme/ou=responsible/s= ted}."

The Re-Key GCKS is included to cater for situations in which a back-up GCKS (other than then one the member initially joined to) is specified

to handle emergency situations (e.g. Primary GCKS crashed, network partitions, etc).

Authorization Fields will fulfill various needs during the lifetime of a group. Both new and current members will need to make use of the authorization field to maintain the level of security of the communications group.

For new or potential members, this field of the group's token should be used as input into the decision for joining a particular group and for the acceptance of keying material. Specifically, the rules should be examined to determine whether they are stringent enough for the potential member's security needs, and the member trusts the entities that will be assuming the roles. In the rule-based example above, Bob might be known to have particularly shoddy security practices. A new member would be reassured that the secure distribution of the group's encryption keys will not be managed by Bob. Furthermore, upon acceptance of the invitation to join the group, the member will receive the group communication keys. At that point, the member would need to verify that the GCKS sending the keys fit the profile indicated by the Authorization Field. The integrity of keys received from someone not entitled to act as Key Server should be considered suspect.

The most common use of this field will be by current group members. Current group members use the Authorization Field upon receipt of key management or administrative messages. Before acting upon these messages, it must be determined that the sender did indeed have the necessary permissions to initiate a given action. For example, an unauthorized re-key message might have the result of placing a member on an incorrect key, thereby denying him access to the group's communications.

Figure 5: Authorization sub-fields

4.3 Access Control Field and Sub-Fields

Access to a group implies that a potential group member is given permission to receive an appropriate subset of the group's keys. This is explicitly stated in the policy token to ensure a common access decision space.

The Access Field defines who is permitted to join the group. As with authorizations, access controls can be defined by a combination of rules and explicit names. The rules portion of the Access Field is broken down into a set of permissions that determine access into a group and

the definition (or pointer to the definition) of those permissions for a

particular information domain. The Permissions are hierarchical in the traditional military sense where access at a higher level encompasses all the access of the lower levels plus new ones and dominance rules apply. The Access List can also restrict access to those in a particular knowledge group or groups.

As an example of how the Access Field might be filled, consider a hypothetical group consisting of scientists with research and development permissions working on the company's new product . Each scientist could be given a permission rating. This permission rating would be reflected in that scientist's personnel certificate. The access rule in the policy token could make it mandatory for a potential group member to have a permission rating greater than or equal to "R&D".

In addition to the permission based access decisions, it may be desired to restrict access to the group to scientists who are members of a specific work group. This work group could have a common element in their Distinguished Name fields in their common PKI. For example, the scientist may all be working in the Emerald City, in the land of Oz. The DN access rule could be {/o=acme/ou=Emerald City/ou=Oz/*}.

Figure 6: Access Control sub-fields

4.4 Mechanism Fields and Sub-Fields

The security services and related mechanisms used to protect the data must be consistent throughout the group to maintain uniform data protection throughout the data flow. For example, if the confidentiality of data is protected by the Advanced Encryption Standard (AES) at one point and by the Data Encryption Standard (DES) at another, the overall protection afforded the data is only the strength offered by the weakest mechanism. The data mechanisms used to protect the various phases of the group communications are specified in the Mechanisms Field of the Group Policy Token.

The three Categories of SAs (REF[2]) are described in this field.

The Category-1 SA defines the information for a newcomer to join the group by opening a secure channel to the GCKS. The secure channel establishment requirements and parameters are described in the Category-1 **SA sub-field.**

The Category-2 SA (or Re-Key SA) describes the Security Association that

will be used by the GCKS to perform a re-key of the group-key (TEK

and/or KEK vector) within the group. This sub-field is actually broken down into further fields: Protected Key Management and Bypass. The Protected Key Management SA identifies the security mechanisms set up for key management messages. For cases in which Protected Key Management messages cannot be correctly received and read by all members, the Bypass SA will allow particular messages through without confidentiality processing. Both of these correspond to the Category-2 SA (Re-Key SA) of the MSEC Key Management Architecture (GKM Arch [REF]).

The Category-3 SA (or Data Security SA) describes the protection afforded Multicast Group Communications. The actual format of this field is largely determined by the choice of security protocol for the protection of data. Mechanism and mode choices for confidentiality and authentication, key determination, key length, and rekey must all be considered. Furthermore, agreed upon key validity intervals (cryptoperiods) and possible data source authentication must also be specified.

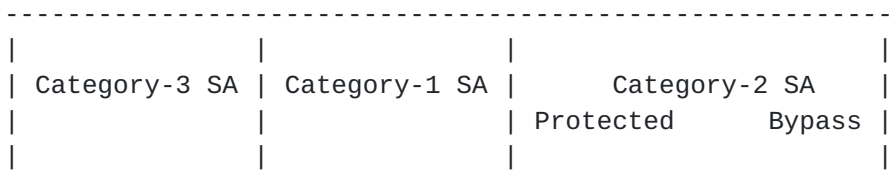

Figure 7: Mechanism Sub-Fields

4.5 Signature Field and Sub-Fields

The signature field contains the information that verifies the authenticity of the group policy token. It clearly identifies the signer of the token -- the Group Owner, the PKI information needed to establish what is the proper certificate to use for the signature verification, and the signature data. The signature covers all of the fields of the Group Policy Token except for the Signature Field itself. Because of this, any unauthorized change in the group policy token will invalidate the signature.

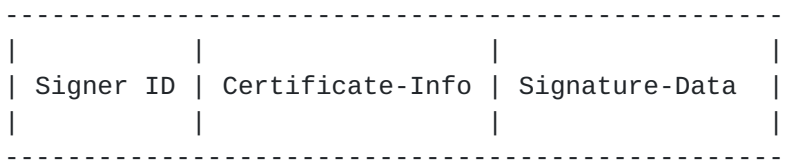

Figure 8: Signature Sub-Fields

5. Group Policy Token: Header and Payloads

5.1 Generic Payload Header

Each payload defined in the following sections begins with a generic header, shown in Figure 3, which provides a payload ``chaining`` capability and clearly defines the boundaries of a payload. The Generic Payload Header fields are defined as follows:

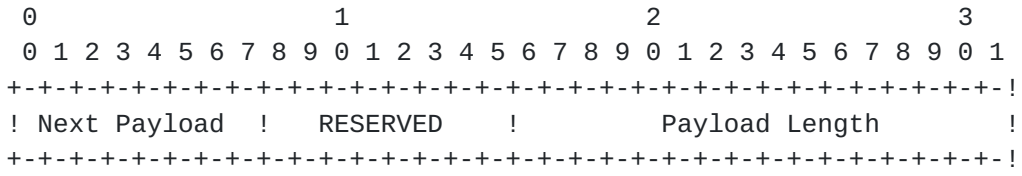

Figure 3: Generic Payload Header

Next Payload (1 octet) - Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be 0. This field provides the ``chaining`` capability.

RESERVED (1 octet) - Unused, set to 0.

Payload Length (2 octets) - Length in octets of the current payload, including the generic payload header.

5.2 Policy Token Payload

The Policy Token Payload contains group specific information that describes the group security relevant behaviors, access control parameters, and security mechanisms. This information may contain a digital signature(s) to prove authority and integrity of the information. Figure 4 shows the format of the payload.

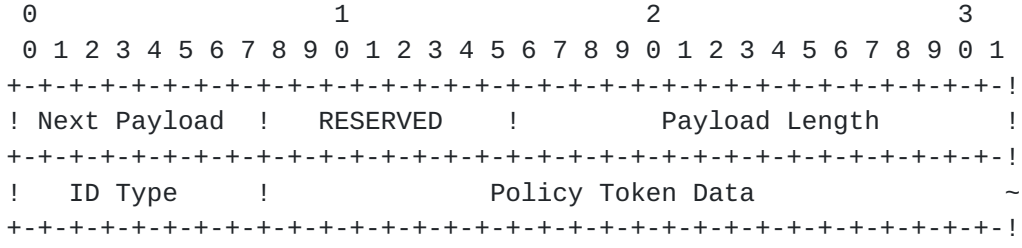

Figure 4: Policy Token Payload Format

The Policy Token Payload fields are defined as follows:

Next Payload (1 octet) - Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be 0.

RESERVED (1 octet) - Unused, set to 0.

Payload Length (2 octets) - Length in octets of the current payload, including the generic payload header.

ID Type (1 octet) - Specifies the type of Policy Token being used. Table 12 identifies the types of policy tokens.

Table 1: Policy Token Types

ID_Type	Value
Group	0
Auxiliary	1
Reserved	2-63
Unassigned	64-255

Policy Token Data (variable length) - Contains Policy Token information. The values for this field are group specific and the format is specified by the ID Type field.

The payload type for the Policy Token Payload is one (1).

6. Example of Group Policy Tokens

The following section describes an example of a Group Policy Token. The full token is provided first, though the reader is encouraged first to read the parts specific to each of the categories of SAs.

6.1 Description of Example

The this example the group security protocol (namely GSAKMP) creates an association between multiple entities connected by the Internet. The intent of the group security protocol is to use existing protocol-services that are standardized for the Internet to facilitate setting up these secure groups. IPsec is one of the standard secure services that can be used, though others (e.g. S/MIME or even SSL) can be used as a unicast security association mechanism.

Additionally, the Group Policy Token defines the policy for protecting the multicast data traffic (Category-3 SA). Once again, IPsec can be used to protect these messages, as can S/MIME.

In this particular example, IPsec is used as the underlying security association protocol for both unicast (Cat-1) and multicast messages (Cat-2 and Cat-3).

To configure IPsec, interaction will occur with the Security Policy Database (SPD) and the Security Association Database (SAD). Since the group security protocol exists above the IPsec network layer encryption engine, configuration must include that of the appropriate databases controlling IPsec. This is to ensure that IPsec processes its messages appropriately.

The current example requires a unicast SA to protect some of the exchanges. It also requires a bypass of IPsec processing for a subset of messages.

Each group can have unique IPsec processing requirements for the unicast and multicast messages pertaining to that particular group. These group unique configurations must be conveyed to the IPsec controlling databases in a way that will allow correct IPsec processing for each groups' message. Special attention must be paid to the IPsec selector fields. The standard source and destination pair selectors are not adequate for multicast groups where multiple groups share the same destination address.

The example assumes the presence of IKE [5] as a unicast SA establishment protocol for IPsec.

6.2 The IPsec Architecture

The IPsec architecture document [6] identifies two databases used by IPsec - Security Policy Database (SPD) and Secure Association Database (SAD). The former specifies the policies that determine the disposition of all IP traffic (inbound or outbound) from a host, security gateway, or BITS or BITW IPsec implementation. The latter database contains parameters that are associated with each (active) security association. The information in these databases allows the IPsec protocol to process incoming and outgoing packets.

6.2.1 SPD Inputs

Purpose of the SPD (copied from RFC 2401):

"A security association is a management construct used to enforce a security policy in the IPsec environment. Thus an essential

element of SA processing is an underlying Security Policy Database

(SPD) that specifies what services are to be offered to IP datagrams and in what fashion. The form of the database and its interface are outside the scope of this specification. However, this section does specify certain minimum management functionality that must be provided, to allow a user or system administrator to control how IPsec is applied to traffic transmitted or received by a host or transiting a security gateway.

The SPD must be consulted during the processing of all traffic (INBOUND and OUTBOUND), including non-IPsec traffic. In order to support this, the SPD requires distinct entries for inbound and outbound traffic. One can think of this as separate SPDs (inbound vs. outbound). In addition, a nominally separate SPD must be provided for each IPsec-enabled interface.

An SPD must discriminate among traffic that is afforded IPsec protection and traffic that is allowed to bypass IPsec. This applies to the IPsec protection to be applied by a sender and to the IPsec protection that must be present at the receiver. For any outbound or inbound datagram, three processing choices are possible: discard, bypass IPsec, or apply IPsec. The first choice refers to traffic that is not allowed to exit the host, traverse the security gateway, or be delivered to an application at all. The second choice refers to traffic that is allowed to pass without additional IPsec protection. The third choice refers to traffic that is afforded IPsec protection, and for such traffic the SPD must specify the security services to be provided, protocols to be employed, algorithms to be used, etc."

The critical elements of the SPD are

- Destination IP Address
- Source IP Address
- Name
- Data sensitivity level
- Transport Layer Protocol
- Source and Destination (e.g., TCP/UDP) Ports
- Specific IPsec processing
- Specific IPsec Algorithms (spi, service, algorithm, mode)

6.2.2 SAD Inputs

Purpose of the SAD (copied from RFC 2401):

"In each IPsec implementation there is a nominal Security Association Database, in which each entry defines the parameters associated with one SA. Each SA has an entry in the SAD. For outbound processing, entries are pointed to by entries in the SPD. Note that if an SPD entry does not currently point to an SA that

is appropriate for the packet, the implementation creates an

appropriate SA (or SA Bundle) and links the SPD entry to the SAD entry (see Section 5.1.1). For inbound processing, each entry in the SAD is indexed by a destination IP address, IPsec protocol type, and SPI. The following parameters are associated with each entry in the SAD. This description does not purport to be a MIB, but only a specification of the minimal data items required to support an SA in an IPsec implementation."

The critical elements of the SAD are

- SAD index
- Outer Header's Destination IP address
- IPsec Protocol
- SPI
- Sequence Number Counter
- Sequence Counter Overflow [only outbound]
- Anti-Replay Window: [only for inbound.]
- AH Authentication algorithm, keys, etc. [REQUIRED for AH implementations]
- ESP Encryption algorithm, keys, IV mode, IV, etc. [REQUIRED for ESP implementations]
- ESP authentication algorithm, keys, etc [REQUIRED for ESP implementations]
- Lifetime of this Security Association: Required
- IPsec protocol mode: tunnel, transport or wildcard
- Path MTU: [REQUIRED for all implementations but used only for outbound traffic]

6.3 Mapping from Policy Token for Ipsec

One of the major components of the MSEC Key Management Architecture is the definition of three distinct categories of Secure Association bundles. All three bundles together define a GSA.

The GKM BB addresses three distinct interactions that make up a GSA:

1. Cat-1: Unicast secure channel between key server and potential group member
2. Cat-2: Key management messages over multicast communications (Key server to members)
3. Cat-3: traffic data SA s (Sender to Receivers)

There are several messages that need to be configured into IPsec SPDs. There is a need for some key management messages to bypass IPsec processing. These messages are self-protecting or do not require protection. During the process of establishing the group there are unicast messages between the key servers and the members -- these messages may be sensitive in nature and require IPsec processing. Once a group has been established, there are messages that manage the group

as a single entity, some these messages (LKH rekey [10]) are self

protecting and do not need IPsec, but other management messages like policy updating or group deletion may be sensitive and need IPsec protection. Finally, the group key that is protecting the application data, separate from GSAKMP data will need IPsec processing. In each of these instances, the group policy token will carry the configuration information needed to configure their SPD and SAD.

There is an enumerated example policy token that follows this section. We have included the field numbers from that example into the following SDP and SAD table entries to allow a mapping from the policy token to the databases.

6.4 Bypass

At least two types of messages need to bypass IPsec processing - request to join and rekey. The request to join message may be sent from the member to the key server before any GSA or SA is created, so no association may exist to encrypt this message. Another message that must bypass IPsec is the group rekey message -- this message is self-protecting and is sent to the entire group. The group rekey message may be used to restore cryptographic synchronization in a group. Processing this message through IPsec would negate its ability to restore synchronization.

Bypass SPDs	Incoming	Outgoing
Destination IP Address	From (102)	From (109)
Source IP Address	From (101)	From (108)
Name	n/a	n/a
Data sensitivity level	n/a	n/a
Transport Layer Protocol	From (103)	From (110)
Source and Destination (e.g., TCP/UDP) Ports:	Source: */Dest from (104)	Source: */Dest from(111)
Specific IPsec processing:	From (105)	From (112)
Specific IPsec Algo (spi, service, algo, mode):	From (99)	From (106)

6.5 Category-1 SA

During the initial Regsitration phase between a newcomer/group member and the GCKS, sensitive information will be exchanged.

The group policy token example will specify the exact IPsec services that are required between key server and group member. It will also specify the key creation parameters that satisfy the group data security requirements.

In this example, we assumed the use of IKE as the key creation and negotiation protocol. The group policy token will specify the correct IKE parameters for the group. IKE will create the pairwise key and assign an appropriate SPI.

Two SPDs are required to completely specify this interaction -- one for inbound messages in one for outbound messages.

Of special note: the standard selectors for IPsec are inadequate to differentiate between multiple groups on a single multicast address. One technique that can mitigate this problem is the assignment of a 4-byte random number to a unique group on a multicast address.

Cat-1 SPDs	Outgoing	Incoming
Destination IP Address	From (79)	From (58)
Source IP Address	From (78)	From (57)
Name	From (82)	From (61)
Data sensitivity level	From (83)	From (62)
Transport Layer Protocol	From (80)	From (59)
Source & Destination Ports	Source: */ (81)	Source: */Dest: (60)
Specific IPsec processing	IPsec process	IPsec process
Specific IPsec Algo (spi, service, algo, mode):	Spi: IKE	Spi: from IKE
IPsec service:	From (71,73-77)	From: (50,52-56)
IKE attributes:	From (84-91)	From: (63-70)

Cat-1 SADs	Outgoing	Incoming
SAD Index:		
Outer Dest. IP addr	* (w/ multicast addr masked out)	Local
Ipsec Protocol	From (71)	From (50)
SPI	From IKE	From IKE
Sequence Number Counter	From IKE	From IKE
Sequence Counter Overflow [only outbound]	From IKE	From IKE
Anti-Replay Window [only for inbound]	From IKE	From IKE
AH Authentication algo., keys, etc. [REQUIRED for AH implementations]	n/a	n/a
ESP Encryption algorithm, keys, IV mode, IV, etc. [REQUIRED for ESP		

implementations] From (73), IKE keys From (52), IKE keys

ESP authentication alg., keys, etc [REQUIRED for ESP implementations]	From (74), IKE keys	From (53), IKE keys
Lifetime of this Security Association: Required	From (76)	From (55)
IPsec protocol mode: tunnel, transport or wildcard	From (75)	From (54)

6.6 Category-2 SA

In the management of groups, the GCKS uses the Category-2 SA to send out control messages, which may contain keying material (re-key message) or simply consists of commands (group maintenance).

The rekey messages will change the group traffic encryption keys and associated rekey arrays in response to some problem with the group security. In the case of rekey messages, one cannot assume that a single group traffic encryption key is available. Therefore, the rekey messages are self-protecting and bypass IPsec processing.

Normal group maintenance messages perform actions that apply to the entire group -- policy updates, group synchronization, and group deletion. These messages may contain sensitive information and usually are sent during times where the group is stable. Therefore, IPsec processes these messages.

IPsec will bypass the rekey messages as defined in the bypass SPD above. The group security protocol (ie. GSAKMP, GDOI) must maintain internal configurations for processing the rekey messages.

For rekey messages the Selectors are taken from the Policy Token (3), while the Services are taken from the Policy Token (92-98).

6.7 Category-3 SA

Category 3 is the final set of IPsec configurations (SPD and SAD entries) used to protect the data traffic (nb. Hence also called "Data Security SA"). Here, it is assumed that Ipsec will be used to protect the data.

The group policy token will specify the IPsec parameters that are needed to protect the data. A group Security Parameter Index (SPI) will be created and distributed across the entire group.

Cat-3 SPDs	Outgoing	Incoming
Destination IP Addr.	From (32)	From (45)
Source IP Addr.	From (31)	From (44)
Name	From (34)	From (47)
Data sensitivity level	From (35)	From (48)
Transport Layer Protocol	From (33)	From (46)
Src & Dest Ports	Src: */Dest:*	Src: */Dest:* mask protocol bypass
Specific IPsec processing	IPsec process	IPsec process
Specific IPsec Algo (spi, service, algo, mode)	From (24a) - IPsec service from (26,27,28) - Key attributes: from protocol	From (24a) - IPsec service from (39,40,41) - Key attributes: from protocol

Cat-3 SADs	Outgoing	Incoming
Outer Dest. IP Addr.	Multicast from (32)	Multicast from (45)
Ipsec Protocol	From (24)	From (24)
SPI		
Sequence Number Counter	(only for 1-to-Many)	(only for 1-to-many)
Sequence Counter Overflow [only outbound]	(only for 1-to-Many)	(only for 1-to-many)
Anti-Replay Window [only for inbound]	(only for 1-to-Many)	(only for 1-to-many)
AH Auth. Algo., keys, etc	n/a	n/a
ESP Encryption algorithm, keys, IV mode, IV, etc. [REQUIRED for ESP implementations]	From (26) Keys from Cat-1	From (39) Keys from Cat-1
ESP authentication alg., keys, etc [REQUIRED for ESP implementations]	From (27) Keys from Cat-1	From (40) Keys from Cat-1
Lifetime of this Security Association: Required	From (29, 30)	From (42, 43)
IPsec protocol mode: tunnel, transport or wildcard	From (28)	From (41)

Path MTU: [REQUIRED for all implementations but used only for outbound traffic]	?	?

6.8 A Complete Group Policy Token

In the following, the complete group policy token is presented, parts from which have been presented in then precious three subsections above.

TOKEN-ID FIELD		
(1)	Version	v1.0
(2)	Protocol	p1.0
(3)	GroupID	IPv4, 224.0.1, 12345678
(4)	Timestamp	20000316182632Z
(5)	Expiration Date	20000616182632Z
AUTHORIZATION FIELD		
(6) Group Owner	Distinguished Name	/C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./ CN=Jane Owner
(7)	Serial Number	87654321
(8)	Certificate Type	X.509v3-DSS-SHA1
(9)	Key Length	1024
(10)	Root CA	C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./ CN =John Root
(11) GCKS	Distinguished Name	C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./ CN = Sally Member
(12)	Certificate Type	X.509v3-DSS-SHA1
(13)	Key Length	1024
(14)	Root CA	C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./ CN =John Root
(15) Rekey GCKS	Distinguished Name	C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./ CN = Johnny Member
(16)	Certificate Type	X.509v3-DSS-SHA1
(17)	Key Length	1024
(18)	Root CA	C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./ CN =John Root

ACCESS CONTROL FIELD			
(19)	Permissions	employee, projectA	
(20)	Access List	Distinguished Name	C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./CN = *
(21)		Certificate Type	X.509v3-DSS-SHA1
(22)		Key Length	1024
(23)		Root CA	C=US/ST=MD/L=Columbia/ O=SPARTA,Inc./ CN =John Root
MECHANISMS FIELD			
(24a)	Cat-3 SA	SPI	4847474747474747 ...
(24)		Security Protocol	ESP
(25)		Processing Direction	Outgoing
(26)		ESP Algorithm	ESP-DES
(27)		ESP Authentication	HMAC-SHA
(28)		Encapsulation Mode	Tunnel
(29)		SA Lifetype	kilobytes
(30)		SA Lifetime	1000
(31)		Source Addr	*
(32)		Destination Addr	224.0.1 (Multicast)
(33)		Transport Protocol	UDP
(34)		GroupID	224.0.1, 12345678
(35)		Security Label	Reference [7]
(36)		Key Creation	preplaced (via GSAKMP)
(24a)		SPI	4847474747474747 ...
(37)		Security Protocol	ESP
(38)		Processing Direction	Incoming
(39)		ESP Algorithm	ESP-DES
(40)		ESP Authentication	HMAC-SHA
(41)		Encapsulation Mode	Tunnel
(42)		SA Lifetype	kilobytes
(43)		SA Lifetime	1000
(44)		Source Addr	*
(45)		Destination Addr	224.0.1, 12345678
(46)		Transport Protocol	UDP
(47)		GroupID	224.0.1, 12345678
(48)		Security Label	Reference [7]
(49)		Key Creation	preplaced (via GSAKMP)
(50)	Cat-1 SA	Security Protocol	ESP
(51)		Processing Direction	Incoming
(52)		ESP Algorithm	ESP-DES
(53)		ESP Authentication	HMAC-SHA
(54)		Encapsulation Mode	Tunnel
(55)		SA Lifetype	seconds
(56)		SA Lifelength	1000
(57)		Source Addr	* (except multicast)
(58)		Destination Addr	self

(59) Transport Protocol TCP

(60)	Destination Port	555 (GSAKMP)
(61)	GroupID	224.0.1, 12345678
(62)	Security Label	Reference [7]
(63)	Key Creation IKE	
(64)	IKE Encr. Algorithm	DES-CBC
(65)	IKE Hash Algorithm	SHA
(66)	IKE Auth. Method	DSS-Signature
(67)	IKE Group Desc.	MODP-1024
(68)	IKE Group Type	MODP
(69)	IKE Group Prime	7
(70)	IKE Group Gen.	2
(71)	Security Protocol	ESP
(72)	Processing Direction	Outgoing
(73)	ESP Algorithm	ESP-DES
(74)	ESP Authentication	HMAC-SHA
(75)	Encapsulation Mode	Tunnel
(76)	SA Lifetype	seconds
(77)	SA Lifelength	1000
(78)	Source Addr	self
(79)	Destination Addr	* (except Multicast)
(80)	Transport Protocol	TCP
(81)	Destination Port	555 (GSAKMP)
(82)	GroupID	224.0.1, 12345678
(83)	Security Label	Reference [7]
(84)	Key Creation	IKE
(85)	IKE Encr. Algorithm	DES-CBC
(86)	IKE Hash Algorithm	SHA
(87)	IKE Auth. Method	DSS-Signature
(88)	IKE Group Desc.	MODP-1024
(89)	IKE Group Type	MODP
(90)	IKE Group Prime	7
(91)	IKE Group Gen.	2
(92) Cat-2 SA	Key Creation Method	Diffie-Hellman
(93)	D-H n	779
(94)	D-H q	2
(95)	Key Encr. Algorithm	Triple-DES-ECB
(96)	Rekey Method	LKH
(97)	Signature Algorithm	DSS
(98)	Hash	SHA
(99) Cat-2 Bypass	Security Protocol	IPsec None
(100)	Processing Direction	Incoming
(101)	Source Addr	*
(102)	Destination Addr	*
(103)	Transport Protocol	*
(104)	Destination Port	777

(105)	Processing	BYPASS

(106)	Security Protocol	IPsec None
(107)	Processing Direction	Outgoing
(108)	Source Addr	self
(109)	Destination Addr	*
(110)	Transport Protocol	*
(111)	Destination Port	777
(112)	Processing	BYPASS

SIGNATURE FIELD

(113)	Signature Algorithm	DSS
(114)	Hash	SHA
(115)	Signature Data	948456945040...

REFERENCES

[BF99] B. Briscoe, I. Fairman, Nark: Receiver-based Multicast, Non-repudiation and Key Management, Proceedings of ACM E-Commerce'99, rbriscoe@bt.co.uk.

[BMS99] D. Balenson, D. McGrew, A. Sherman, Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization, http://www.ietf.org/internet-drafts/draft-balenson-groupkeymgmt-oft-00.txt, February 1999, Work in Progress.

[BR93] M. Bellare, P. Rogaway, Entity Authentication and Key Distribution, Advances in Cryptology - Crypto '93 Proceedings, Springer-Verlag, 1993.

[Bris99] B. Briscoe, MARKS: Zero Side Effect Multicast Key Management using Arbitrarily Revealed Key Sequences, Proceedings of NGC'99, rbriscoe@bt.co.uk.

[CP99] R. Canetti and B. Pinkas, A taxonomy of multicast security issues, http://www.ietf.org/internet-drafts/draft-irtf-smug-taxonomy-01.txt, April 1999, Work in Progress.

[DVW92] Diffie, P. van Oorschot, M. J. Wiener, Authentication and Authenticated Key Exchanges, Designs, Codes and Cryptography, 2, 107-125 (1992), Kluwer Academic Publishers.

[FS00] N. Ferguson and B. Schneier, A Cryptographic Evaluation of IPsec, CounterPane, http://www.counterpane.com/ipsec.html.

[HCBD99] T. Hardjono, R. Canetti, M. Baugher, P. Disnmore, Secure IP Multicast: Problem areas, Framework, and Building Blocks,

http://www.ietf.org/internet-drafts/draft-irtf-smug-framework-00.txt,
Work in Progress 1999.

[HCD99] T. Hardjono, B. Cain, N. Doraswamy, A framework for group key management for multicast security, http://www.ietf.org/internet-drafts/draft-ietf-ipsec-gkmframework-01.txt, July 1999, Work in Progress.

[HH99a] H. Harney, E. Harder, Multicast Security Management Protocol (MSMP) Requirements and Policy, draft-harney-msmp-sec-00.txt, March 1999, Work in Progress.

[HH99b] H. Harney, E. Harder, Group Secure Association Key Management Protocol, http://search.ietf.org/internet-drafts/draft-harney-sparta-gsakmp-sec-00.txt, April 1999, Work in Progress.

[Kraw96] H. Krawczyk, SKEME: A Versatile Secure Key Exchange Mechanism for Internet, ISOC Secure Networks and Distributed Systems Symposium, San Diego, 1996.

[RFC1889] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, January 1996.

[RFC2093] Harney, H., and Muckenhirn, C., "Group Key Management Protocol (GKMP) Specification," RFC 2093, July 1997.

[RFC2094] Harney, H., and Muckenhirn, C., "Group Key Management Protocol (GKMP) Architecture," RFC 2094, July 1997.

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

[RFC2407] D. Piper, The Internet IP Domain of Interpretation for ISAKMP, November 1998.

[RFC2408] D. Maughan, M. Shertler, M. Schneider, J. Turner, Internet Security Association and Key Management Protocol, November 1998.

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

[RFC2412] H. Orman, The OAKLEY Key Determination Protocol, November 1998.

[RFC2522] P. Karn, W. Simpson, Photuris: Session-Key Management Protocol, March 1999.

[RFC2627] D. M. Wallner, E. Harder, R. C. Agee, Key Management for Multicast: Issues and Architectures, September 1998.

[SDNS88] H. L. Rogers, An Overview of the CANEWARE Program, 10th National Security Conference, National Security Agency, 1988.

[WL98] C.K.Wong, S.S. Lam, Digital Signatures for Flows and Multicasts, Proceedings of IEEE ICNP'98, October 14-16, 1998.

Authors Address:

Thomas Hardjono
VeriSign
401 **Edgewater Pl, Suite 280**
Wakefield, MA 01880, USA
thardjono@verisign.com

Hugh Harney
SPARTA, Inc.
Secure Systems Engineering Division
9861 **Broken Land Parkway, Suite 300**
Columbia, MD 21046-1170, USA
+1 410 381 9400 (ext. 203)
hh@columbia.sparta.com

Patrick McDaniel
Department of Electrical Engineering and Computer Science
University of Michigan
3115 **EECS Building**
1301 **Beal Avenue**
Ann Arbor, MI 48109
pdmcdan@eecs.umich.edu

Andrea Colgrove
SPARTA, Inc.
Secure Systems Engineering Division
9861 **Broken Land Parkway, Suite 300**
Columbia, MD 21046-1170, USA
+1 410 381 9400 (ext. 203)
ac@columbia.sparta.com

Peter Dinsmore
NAI Labs
3060 **Washington Road,**
Glenwood, MD 21738
(443) 259-2346
Pete_Dinsmore@NAI.com