Network Working Group
Internet-Draft
Obsoletes: 4909 (if approved)
Updates: 3830, 4563, 5410, 6043
(if approved)
Intended status: Standards Track
Expires: September 8, 2011

J. Arkko
A. Keranen
J. Mattson
Ericsson

March 7, 2011

IANA Rules for MIKEY (Multimedia Internet KEYing)
draft-arkko-mikey-iana-00

Abstract

This document clarifies and relaxes the IANA rules for Multimedia Internet KEYing (MIKEY). This document updates RFCs 3830, 4563, 5410, 6043, and obsoletes RFC 4909.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 8, 2011.

Copyright Notice

described in the Simplified BSD License.

1. Introduction

This document relaxes the IANA rules for Multimedia Internet KEYing (MIKEY) [RFC3830]. The IANA rules defined in [RFC3830], [RFC4563], [RFC4909], and [RFC5410] are affected. In addition, the rules specified in [RFC6043] are re-specified here.

Most of the values in MIKEY namespaces are divided into two ranges: "IETF Review" (or "IETF Consensus" as it was previously called) and "Reserved for Private Use" [RFC5226]. This document changes, for majority of the namespaces, the requirement of "IETF Review" into "IETF Review or IESG Approval" [RFC5226]. For some namespaces, the requirement is changed to "Specification Required" [RFC5226].

The rationale for this update is that there can be situations where it makes sense to grant an allocation under special circumstances or that time has shown that the current requirement is unnecessarily strict for some of the namespaces. By changing the current IANA rules to allow also for IESG Approval [RFC5226], it becomes possible for the Internet Engineering Steering Group (IESG) to consider an allocation request, even if it does not fulfill the default rule. For instance, an experimental protocol extension could perhaps deserve a new payload type as long as a sufficient number of types still remains, and the MIKEY community is happy with such an allocation. Moreover, for some registries a stable specification would be a sufficient requirement and hence this is reflected in the updated IANA rules. For instance, for some registries an RFC via the Independent Stream at the RFC Editor is sufficient, and does not force an IETF evaluation of a particular new extension for which there is no general demand.

This document also makes some small corrections to the existing IANA registries. (RFC Editor: Please remove this paragraph upon publication as an RFC.)

The rest of this document is structured as follows. Section 2 defines the new IANA rules. Section 3 discusses the security implications of this document. Section 4, Section 5, Section 6, and Section 7, explain the changes to the RFCs 3830, 4563, 4909, 5410, and 6043.

2. IANA Considerations

IANA is requested to update the registries related to MIKEY as

specified below. All other MIKEY IANA registries are to remain unchanged.

A registry for the version field should be created, with the value 0x01 as the only currently allocated item. (RFC Editor: Please remove the preceding sentence upon publication as an RFC.) New values for the version field ([RFC3830], Section 6.1) can be allocated via IETF Review.

New values for the Data type ([RFC3830], Section 6.1) and the C envelope key cache indicator ([RFC3830], Section 6.3) fields can be allocated via IETF Review.

The requirement for adding new values into name spaces, originally defined in [RFC3830], and having requirement of "IETF Review" is to be changed into "IETF Review or IESG Approval". This change affects the following namespaces:

- Next payload ([RFC3830], Section 6.1)
- PRF func ([RFC3830], Section 6.1)
- CS ID map type ([RFC3830], Section 6.1)
- Encr alg ([RFC3830], Section 6.2)
- MAC alg ([RFC3830], Section 6.2)
- DH-Group ([RFC3830], Section 6.4)
- S type ([RFC3830], Section 6.5)
- TS type ([RFC3830], Section 6.6)
- ID type ([RFC3830], Section 6.7)
- Cert type ([RFC3830], Section 6.7)
- Hash func ([RFC3830], Section 6.8)
- SRTP Type ([RFC3830], Section 6.10)
- SRTP encr alg ([RFC3830], Section 6.10)
- SRTP auth alg ([RFC3830], Section 6.10)
- SRTP PRF ([RFC3830], Section 6.10)

- FEC order ([RFC3830], Section 6.10)
- Key Data Type ([RFC3830], Section 6.13)
- KV Type ([RFC3830], Section 6.13)

The "IETF Review" requirement for the following registries, originally defined in [RFC3830], [RFC4563], [RFC4909] and [RFC5410], is to be changed into "Specification Required".

- Prot type ([RFC3830], Section 6.10)
- Error no ([RFC3830], Section 6.12)
- General Extension Type ([RFC3830], Section 6.15)
- KEY ID Type ([RFC4563], Section 4)
- OMA BCAST Types ([RFC5410], Section 3)

The "Specification Required" requirement remains for the following namespaces:

- TS Role ([RFC6043], Section 6.4)
- ID Role ([RFC6043], Section 6.6)
- RAND Role ([RFC6043], Section 6.8)
- Ticket Type ([RFC6043], Section 6.10)

The range of valid values for certain namespaces defined in IANA considerations of [RFC3830] was not explicitly defined and is clarified here as follows:

Namespace	Valid values
C envelope key cache indicator	0 - 3
S type	0 - 15
Key Data Type	0 - 15
KV Type	0 - 15

(RFC Editor: please remove this paragraph before publication and when the IANA registry has been updated with the following changes) The current MIKEY IANA registry defines sub-registries with explicit name for certain parameters (e.g., Next Payload) whereas other parameters

(e.g., Encr alg) have no (explicit) sub-registries. IANA is requested to define explicit sub-registries for all the parameters with sub-registry names matching the names used in this document.

3. Security Considerations

This specification does not change the security properties of MIKEY. However, when new values are introduced without IETF consensus, care needs to be taken to assure that possible security concerns regarding the new values are still addressed.

4. Changes from RFC 3830

Section 2 relaxes the requirements from those defined in [RFC3830]. A number of namespaces now have the "IETF Review or IESG Approval" requirement, when they previously had the "IETF Review" requirement. In addition, some namespaces now have the "Specification Required" requirement.

5. Changes from RFC 4563

Section 2 relaxes the requirements from those defined in [RFC4563]. The KEY ID Type namespace now has the Specification Required requirement.

6. Changes from RFC 4909 and RFC 5410

Section 2 relaxes the requirements from those defined in [RFC4909]. The OMA BCAST Types namespace now has the Specification Required requirement. Note that [RFC5410] obsoleted [RFC4909] but does not actually define the IANA rules itself. As a result, from now on this RFC defines the IANA requirements for the OMA BCAST Type namespace.

7. Changes from RFC 6043

There are no changes to the rules specified in [RFC6043]. However, for sake of completeness, Section 2 re-specifies these rules in this document, and from now on this RFC defines the IANA requirements for those namespaces.

8. References

8.1. Normative References

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC4563] Carrara, E., Lehtovirta, V., and K. Norrman, "The Key ID Information Type for the General Extension Payload in Multimedia Internet KEYing (MIKEY)", RFC 4563, June 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5410] Jerichow, A. and L. Piron, "Multimedia Internet KEYing (MIKEY) General Extension Payload for Open Mobile Alliance BCAST 1.0", RFC 5410, January 2009.

[RFC6043] Mattsson, J. and T. Tian, "MIKEY-TICKET: Ticket Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY)", RFC 6043, 2011.

8.2. Informative References

[RFC4909] Dondeti, L., Castleford, D., and F. Hartung, "Multimedia Internet KEYing (MIKEY) General Extension Payload for Open Mobile Alliance BCAST LTKM/STKM Transport", RFC 4909, June 2007.

Authors' Addresses

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Ari Keranen
Ericsson
Jorvas 02420
Finland

Email: ari.keranen@ericsson.com

John Mattson
Ericsson
Stockholm SE-164 80
Sweden

Email: john.mattsson@ericsson.com

MSEC Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

B. Weis
S. Rowles
Cisco Systems
T. Hardjono
MIT
March 14, 2011

The Group Domain of Interpretation
draft-ietf-msec-gdoi-update-08

Abstract

This document describes an updated version of the Group Domain of Interpretation (GDOI) protocol specified in RFC 3547. The GDOI provides group key management to support secure group communications according to the architecture specified in RFC 4046. The GDOI manages group security associations, which are used by IPsec and potentially other data security protocols.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 15, 2011.

Table of Contents

1. Introduction

Secure group and multicast applications require a method by which each group member shares common security policy and keying material. This document describes the Group Domain of Interpretation (GDOI), which is an ISAMKP [RFC2408] Domain of Interpretation (DOI), a group key management system. The GDOI distributes security associations (SAs) for IPsec AH [RFC4302] and ESP [RFC4303] protocols and potentially other data security protocols used in group applications. The GDOI uses the group key management model defined in [RFC4046], and described more generally by the The Multicast Group Security Architecture [RFC3740].

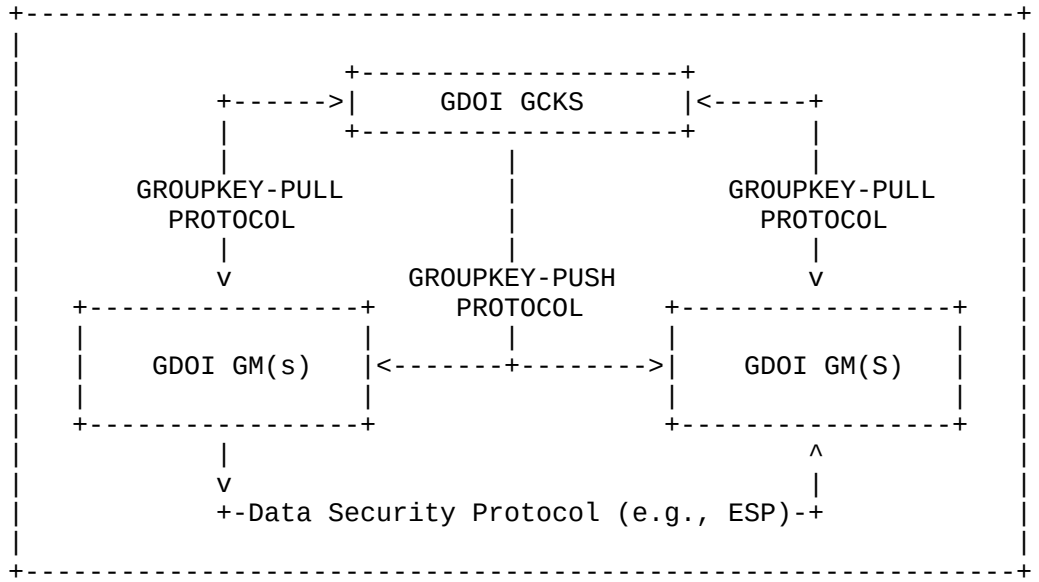
In this group key management model, the GDOI protocol participants are a "group controller/key server" (GCKS) and a group member (GM). A group member contacts ("registers with") a GCKS to join the group. During the registration mutual authentication and authorization are achieved, after which the GCKS distributes current group policy and keying material to the group member over an authenticated and encrypted session. The GCKS may also initiate contact ("rekeys") with group members to provide updates to group policy.

ISAKMP defines two "phases" of negotiation (p.16 of [RFC2408]). A Phase 1 security association provides mutual authentication and authorization, and a security association that is used by the protocol participants to execute a phase 2 exchange. This document incorporates (i.e., uses but does not re-define) the Phase 1 security association definition from the Internet DOI [RFC2407], [RFC2409]. Phase 1 security association types other than ISAKMP are possible, and are noted in Appendix A. Requirements of those phase 1 security associations are specified in Section 2. The GDOI includes two new phase 2 ISAKMP exchanges (protocols), as well as necessary new payload definitions to the ISAKMP standard (p. 14 of [RFC2408]). These two new protocols are:

1. The GROUPKEY-PULL registration protocol exchange. This exchange uses "pull" behavior since the member initiates the retrieval of these SAs from a GCKS. It is protected by an ISAKMP phase 1 protocol, as described above. At the culmination of a GROUPKEY-PULL exchange, an authorized group member has received and installed a set of SAs that represent group policy, and it is ready to participate in secure group communications.

2. The GROUPKEY-PUSH rekey protocol exchange. The rekey protocol is a datagram initiated ("pushed") by the GCKS, usually delivered to group members using a IP multicast address. The rekey protocol is an ISAKMP protocol, where cryptographic policy and keying material ("Re-key SA") is included in the group policy

distributed by the GCKS in the GROUPKEY-PULL exchange. At the culmination of a GROUPKEY-PUSH exchange the key server has sent group policy to all authorized group members, allowing receiving group members to participate in secure group communications. If a group management method is included in group policy (as described in Section 7.4), at the conclusion of the GROUPKEY-PUSH exchange some members of the group may have been de-authorized and no longer able to participate in the secure group communications.



Although the GROUPKEY-PUSH protocol specified by this document can be used to refresh the Re-key SA protecting the GROUPKEY-PUSH protocol, the most common use of GROUPKEY-PUSH is to establish keying material and policy for a data security protocol.

In summary, GDOI is a group security association management protocol: all GDOI messages are used to create, maintain, or delete security associations for a group. As described above, these security associations protect one or more data security protocol SAs, a Re-key SA, and/or other data shared by group members for multicast and groups security applications.

1.1. Requirements notation

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 [RFC2119].

1.2. Terminology

The following key terms are used throughout this document.

Data-Security SA. The security policy distributed by a GDOI GCKS describing traffic that is expected to be protected by group members. This document described the distribution of IPsec AH and ESP Data-Security SAs.

Group Controller/Key Server A device that defines group policy and distributes keys for that policy.[RFC3740]

Group Member. An authorized member of a secure group, sending and/or receiving IP packets related to the group.

GROUPKEY-PULL. A protocol used by a GDOI Group Member to request group policy and keying material.

GROUPKEY-PUSH. A protocol used by a GDOI GCKS to distribute updates of group policy and keying material to authorized group members.

Key Encrypting Key. The symmetric cipher key used to protect the GROUPKEY-PUSH message.

Logical Key Hierarchy). A group management method defined in Section 5.4 of [RFC2627].

Re-key SA. The security policy protecting a GROUPKEY-PUSH protocol.

Traffic Encryption Key. The symmetric cipher key used to protect a data security protocol (e.g., IPsec ESP).

1.3. Acronyms and Abbreviations

The following acronyms and abbreviations are used throughout this document.

AH IP Authentication Header

ATD Activation Time Delay

DOI Domain of Interpretation

DTD Deactivation Time Delay

ESP	IP Encapsulating Security Payload
GCKS	Group Controller/Key Server
GDOI	Group Domain of Interpretation
GAP	Group Associated Policy Payload
GM	Group Member
IV	Initialization Vector
KD	Key Download Payload
KEK	Key Encryption Key
LKH	Lock Key Hierarchy
SA	Security Association
SAK	SA KEK Payload
SEQ	Sequence Number Payload
SAT	SA TEK Payload
SID	Sender-ID
TEK	Traffic Encryption Key

2. GDOI Phase 1 protocol

The GDOI GROUPKEY-PULL exchange is a "phase 2" protocol which MUST be protected by a "phase 1" protocol. The "phase 1" protocol can be any protocol which provides for the following protections:

- Peer Authentication
- Confidentiality
- Message Integrity

The following sections describe one such "phase 1" protocol. Other protocols which may be potential "phase 1" protocols are described in Appendix A. However, the use of the protocols listed there are not considered part of this document.

2.1. ISAKMP Phase 1 protocol

This document defines how the ISAKMP phase 1 exchanges as defined in [RFC2409] can be used a "phase 1" protocol for GDOI. The following sections define characteristics of the ISAKMP phase 1 protocols that are unique for these exchanges when used for GDOI.

Section 7.1 describes how the ISAKMP Phase 1 protocols meet the requirements of a GDOI "phase 1" protocol.

2.1.1. DOI value

The Phase 1 SA payload has a DOI value. That value MUST be the GDOI DOI value as defined later in this document.

2.1.2. UDP port

IANA has assigned port 848 for the use of GDOI, which allows for an implementation to use separate ISAKMP implementations to service GDOI and IKEv1 [RFC2409]. A GCKS SHOULD listen on this port for GROUPKEY-PULL exchanges, and the GCKS MAY use this port to distribute GROUPKEY-PUSH messages. An ISAKMP phase 1 exchange implementation supporting NAT Traversal [RFC3947] may move to port 4500 to process the GROUPKEY-PULL exchange.

3. GROUPKEY-PULL Exchange

The goal of the GROUPKEY-PULL exchange is to establish a Re-key and/or Data-security SAs at the member for a particular group. A Phase 1 SA protects the GROUPKEY-PULL; there MAY be multiple GROUPKEY-PULL exchanges for a given Phase 1 SA. The GROUPKEY-PULL exchange downloads the data security keys (TEKs) and/or group key encrypting key (KEK) or KEK array under the protection of the Phase 1 SA.

3.1. Authorization

The Phase 1 identity SHOULD be used by a GCKS to authorize the Phase 2 (GROUPKEY-PULL) request for a group key. A group member MUST ensure that the Phase 1 identity of the GCKS is an authorized GCKS. When no authorization is performed, it is possible for a rogue GDOI participant to perpetrate a man-in-the-middle attack between a group member and a GCKS [MP04].

3.2. Messages

The GROUPKEY-PULL is a Phase 2 exchange. Phase 1 computes SKEYID_a which is the "key" in the keyed hash used in the GROUPKEY-PULL HASH payloads. When using the Phase 1 defined in this document, SKEYID_a is derived according to [RFC2409]. As with the IKEv1 HASH payload generation (Section 5.5 of [RFC2409], each GROUPKEY-PULL message hashes a uniquely defined set of values (described below). Nonces permute the HASH and provide some protection against replay attacks. Replay protection is important to protect the GCKS from attacks that a key management server will attract.

The GROUPKEY-PULL uses nonces to guarantee "liveness" as well as against replay of a recent GROUPKEY-PULL message. The replay attack is only possible in the context of the current Phase 1. If a GROUPKEY-PULL message is replayed based on a previous Phase 1, the HASH calculation will fail due to a wrong SKEYID_a. The message will fail processing before the nonce is ever evaluated.

In order for either peer to get the benefit of the replay protection, it must postpone as much processing as possible until it receives the message in the protocol that proves the peer is live. For example, the GCKS MUST NOT adjust its internal state (e.g., keeping a record of the GM) until it receives a message with Nr included properly in the HASH payload. This requirement ensures that replays of GDOI messages will not cause the GCKS to change the state of the group until it has confirmation that the initiating group member is live.

```
            Group Member                        GCKS
            ------------                        ----
            HDR*, HASH(1), Ni, ID     -->
                                      <--       HDR*, HASH(2), Nr, SA
            HDR*, HASH(3) [,GAP]      -->
                                      <--       HDR*, HASH(4), [SEQ,] KD
```

* Protected by the Phase 1 SA, encryption occurs after HDR

HDR is an ISAKMP header payload that uses the Phase 1 cookies and a message identifier (M-ID) as in IKEv1.

Hashes are computed in the manner described within RFC 2409. Each HASH computation (shown below) is a prf over the message id (M-ID) from the ISAKMP header concatenated with the entire message that follows the hash including all payload headers, but excluding any padding added for encryption. The GM expects to find its nonce, Ni, in the HASH of a returned message. And the GCKS expects to see its nonce, Nr, in the HASH of a returned message. HASH(2), HASH(3), and HASH(4) also include nonce values previously passed in the protocol (i.e., Ni or Nr minus the payload header). The nonce passed in Ni is represented as Ni_b, and the nonce passed in Nr is represented as Nr_b. The HASH payloads prove that the peer has the Phase 1 secret (SKEYID_a) and the nonce for the exchange identified by message id, M-ID.

```
HASH(1) = prf(SKEYID_a, M-ID | Ni | ID)
HASH(2) = prf(SKEYID_a, M-ID | Ni_b | Nr | SA)
HASH(3) = prf(SKEYID_a, M-ID | Ni_b | Nr_b [ | GAP ])
HASH(4) = prf(SKEYID_a, M-ID | Ni_b | Nr_b [ | SEQ ] | KD)
```

In addition to the Nonce and HASH payloads, the GM identifies the group it wishes to join through the ISAKMP ID payload.

The GCKS informs the member of the cryptographic policies of the group in the SA payload, which describes the DOI, KEK and/or TEK keying material, authentication transforms, and other group policy. The SPIs are also determined by the GCKS and downloaded in the SA payload chain (see Section 5.2). The SA KEK attribute contains the ISAKMP cookie pair for the Re-key SA, which is not negotiated but downloaded. Each SA TEK attribute contains a SPI as defined in Section 5.5 of this document.

After receiving and parsing the SA payload, the GM responds with an acknowledgement message proving its liveness. It optionally includes a GAP payload requesting resources.

The GCKS informs the GM of the value of the sequence number in the SEQ payload. This sequence number provides anti-replay state associated with a KEK, and its knowledge ensure that the GM will not accept GROUPKEY-PULL messages sent prior to the GM joining the group. The SEQ payload has no other use, and is omitted from the GROUPKEY_PULL exchange when a KEK attribute is not included in the SA payload. When a SEQ payload is included in the GROUPKEY-PULL exchange, it includes the most recently used sequence number for the group. At the conclusion of a GROUPKEY-PULL exchange, the initiating group member MUST NOT accept any rekey message with both the KEK attribute SPI value and a sequence number less than or equal to the one received during the GROUPKEY-PULL. When the first group member initiates a GROUPKEY-PULL exchange, the GCKS provides a Sequence Number of zero, since no GROUPKEY-PUSH messages have yet been sent. Note the sequence number increments only with GROUPKEY-PUSH messages. The GROUPKEY-PULL exchange distributes the current sequence number to the group member. The sequence number resets to a value of one with the usage of a new KEK attribute. Thus the first packet sent for a given Rekey SA will have a Sequence Number of 1. The sequence number increments with each successive rekey.

The GCKS always returns a KD payload containing keying material to the GM. If a Re-key SA is defined in the SA payload, then KD will contain the KEK; if one or more Data-security SAs are defined in the SA payload, KD will contain the TEKs.

3.2.1. ISAKMP Header Initialization

Cookies are used in the ISAKMP header to identify a particular GDOI session. The GDOI GROUPKEY-PULL exchange uses cookies according to ISAKMP [RFC2408].

Next Payload identifies an ISAKMP or GDOI payload (see Section 5.0).

Major Version is 1 and Minor Version is 0 according to ISAKMP (Section 3.1 of [RFC2408]).

The Exchange Type has value 32 for the GDOI GROUPKEY-PULL exchange.

Flags, Message ID, and Length are according to ISAKMP (Section 3.1 of [RFC2408]).

3.3. Group Member Operations

Before a GM contacts the GCKS, it must determine the group identifier and acceptable Phase 1 policy via an out-of-band method. Phase 1 is initiated using the GDOI DOI in the SA payload. Once Phase 1 is complete, the GM state machine moves to the GDOI protocol.

To construct the first GDOI message the GM chooses Ni and creates a nonce payload, builds an identity payload including the group identifier, and generates HASH(1).

Upon receipt of the second GDOI message, the GM validates HASH(2), extracts the nonce Nr, and interprets the SA payload. The SA payload contains policy describing the security protocol and cryptographic protocols used by the group. This policy describes the Re-key SA (if present), Data-security SAs, and other group policy. If the policy in the SA payload is acceptable to the GM, it continues the protocol. Otherwise, the GM SHOULD tear down the Phase 1 session after first notifying the GCKS that it is doing so. If a Data-security SA describes the use of a counter mode cipher, the GM determines whether it requires more than one Sender-ID (SID) (see Section 3.5). If so, it includes a GAP payload indicating how many SID values it requires.

When constructing the third GDOI message, it first reviews each Data-security SA given to it. If any include a cipher counter mode, it needs to request for one or more Sender-IDs for its exclusive use within the counter mode nonce. Do to this, the GM will include a GAP payload with its request, as described in the Section 5.4 section of this document. The GM the completes construction of the third GDOI message by creating HASH(3).

Upon receipt of the fourth GDOI message, the GM validates HASH(4).

If the SEQ payload is present, the sequence number in the SEQ payload must be checked against any previously received sequence number for this group. If it is less than the previously received number, it should be considered stale and ignored.

The GM interprets the KD key packets, where each key packet includes the keying material for SAs distributed in the SA payload. Keying material is matched by comparing the SPIs in the key packets to SPIs previously sent in the SA payloads. Once TEK keys and policy are matched, the GM provides them to the data security subsystem, and it is ready to send or receive packets matching the TEK policy. If this group has a KEK, the KEK policy and keys are marked as ready for use, and the GM knows to expect the sequence number reset to 1 with the next Rekey SA, which will be encrypted with the new KEK attribute. The GM is now ready to receive GROUPKEY-PUSH messages.

If the KD payload included an LKH array of keys, the GM takes the last key in the array as the group KEK. The array is then stored without further processing.

3.4. GCKS Operations

The GCKS passively listens for incoming requests from group members. The Phase 1 authenticates the group member and sets up the secure session with them.

Upon receipt of the first GDOI message the GCKS validates HASH(1), extracts the Ni and group identifier in the ID payload. It verifies that its database contains the group information for the group identifier, and that the GM is authorized to participate in the group.

The GCKS constructs the second GDOI message, including a nonce Nr, and the policy for the group in an SA payload, followed by SA KEK, GAP, and/or SA TEK payloads according to the GCKS policy. (See Section 5.2.1 for details on how the GCKS chooses which payloads to send.)

Upon receipt of the third GDOI message the GCKS validates HASH(3). If the message includes a GAP payload, it caches the requests included in that payload for use of constructing the fourth GDOI message.

The GCKS constructs the fourth GDOI message, including the SEQ payload (if the GCKS sends rekey messages), and the KD payload containing keys corresponding to policy previously sent in the SA TEK and SA KEK payloads. If a group management algorithm is defined as part of group policy, the GCKS will first insert the group member into the group management structure (e.g., a leaf in the LKH tree), and then create an LKH array of keys and include it in the KD payload. The first key in the array is associated with the group member leaf node, followed by each LKH node above it in the tree, culminating with the root node (which is also the KEK). If one or more Data-Security SAs distributed in the SA payload included a counter mode of operation, the GCKS includes at least one SID value in the KD payload, and possibly more depending on a request received in the third GDOI message.

3.5. Counter-modes of operation

Several new counter-based modes of operation have been specified for ESP (e.g., AES-CTR [RFC3686], AES-GCM [RFC4106], AES-CCM [RFC4309], AES-GMAC [RFC4543]) and AH (e.g., AES-GMAC [RFC4543]). These counter-based modes require that no two senders in the group ever send a packet with the same Initialization Vector (IV) using the same cipher key and mode. This requirement is met in GDOI when the following requirements are met:

- The GCKS distributes a unique key for each Data-Security SA.

- The GCKS uses the method described in [RFC6054], which assigns each sender a portion of the IV space by provisioning each sender with one or more unique SID values.

When at least one Data-Security SAs included in the group policy includes a counter-mode, the GCKS automatically allocates and distributes one SID to each group member acting in the role of sender on the Data-Security SA. The SID value is used exclusively by the group member to which it was allocated. The group member uses the same SID for each Data-Security SA specifying the use of a counter-based mode of operation. A GCKS MUST distribute unique keys for each Data-Security SA including a counter-based mode of operation in order to maintain a unique key and nonce usage.

When a group member receives a Data-Security SA in a SA TEK payload for which it is a sender, it can choose to request one or more SID values. Requesting a value of 1 is not necessary since the GCKS will automatically allocate exactly one to the sending group member. A group member MUST request as many SIDs matching the number of encryption modules in which it will be installing the TEKs in the outbound direction. Alternatively, a group member MAY request more than one SID and use them serially. This could be useful when it is anticipated that the group member will exhaust their range of Data-Security SA nonces using a single SID too quickly (e.g., before the time-based policy in the TEK expires).

When group policy includes a counter-based mode of operation, a GCKS SHOULD use the following method to allocate SID values, which ensures that each SID will be allocated to just one group member.

1. A GCKS maintains an SID-counter, which records which SIDs that have been allocated. SIDs are allocated sequentially, with the first SID allocated to be zero.

2. Each time an SID is allocated, the current value of the counter is saved and allocated to the group member. The SID-counter is then incremented in preparation for the next allocation.

3. When the GCKS distributes an Data-Security SA specifying a counter-based mode of operation, and a group member is a sender, a group member may request a count of SIDs in a GAP payload. When the GCKS receives this request, it increments the SID-counter once for each requested SID, and distributes each SID value to the group member.

4. A GCKS allocates new SID values for each GROUPKEY-PULL exchange originated by a sender, regardless of whether a group member had previously contacted the GCKS. In this way, the GCKS does not have a requirement of maintaining a record of which SID values it had previously allocated to each group member. More importantly, since the GCKS cannot reliably detect whether the group member had sent data on the current group Data-Security SAs it does not know what Data-Security counter-mode nonce values that a group member has used. By distributing new SID values, the key server ensures that each time a conforming group member installs a Data-Security SA it will use a unique set of counter-based mode nonces.

5. When the SID-counter maintained by the GCKS reaches its final SID value, no more SID values can be distributed. Before distributing any new SID values, the GCKS MUST delete the Data-Security SAs for the group, followed by creation of new Data-Security SAs, and resetting the SID-counter to its initial value.

6. The GCKS SHOULD send a GROUPKEY-PUSH message deleting all Data-Security SAs and the Rekey SA for the group. This will result in the group members initiating a new GROUPKEY-PULL exchange, in which they will receive both new SID values and new Data-Security SAs. The new SID values can safely be used because they are only used with the new Data-Security SAs. Note that deletion of the Rekey SA is necessary to ensure that group members receiving a GROUPKEY-PUSH exchange before the re-register do not inadvertently use their old SIDs with the new Data-Security SAs.

Using the method above, at no time can two group members use the same IV values with the same Data-Security SA key.

4. GROUPKEY-PUSH Message

GDOI sends control information securely using group communications. Typically this will be using IP multicast distribution of a GROUPKEY-PUSH message but it can also be "pushed" using unicast delivery if IP multicast is not possible. The GROUPKEY-PUSH message replaces a Re-key SA KEK or KEK array, and/or creates a new Data-security SA.

```
          GM                    GCKS
          --                    ----
                          <---- HDR*, SEQ, [D,] SA, KD, SIG
```

* Protected by the Re-key SA KEK; encryption occurs after HDR

HDR is defined below. The SEQ payload is defined in the Payloads section. One or more D (Delete) payloads (further described in Section 5.9) optionally specify the deletion of existing group policy. The SA defines the group policy for replacement Re-key SA and/or Data-security SAs as described in the Payloads section, with the KD providing keying material for those SAs.

The SIG payload includes a signature of a hash of the entire GROUPKEY-PUSH message (excepting the SIG payload bytes) before it has been encrypted. The HASH is taken over the string 'rekey', the GROUPKEY-PUSH HDR, followed by all payloads preceding the SIG payload. The prefixed string ensures that the signature of the Rekey datagram cannot be used for any other purpose in the GDOI protocol. The SIG payload is created using the signature of the above hash, with the receiver verifying the signature using a public key retrieved in a previous GDOI exchange. The current KEK encryption key (also previously distributed in a GROUPKEY-PULL exchange or GROUPKEY-PUSH message) encrypts all the payloads following the GROUPKEY-PUSH HDR. Note: The rationale for this order of operations is given in Section 7.3.5.

If the SA defines the use of a single KEK or an LKH KEK array, KD MUST contain a corresponding KEK or KEK array for a new Re-key SA, which has a new cookie pair. When the KD payload carries a new SA KEK attribute (section 5.3), a Re-key SA is replaced with a new SA having the same group identifier (ID specified in message 1 of section 3.2) and incrementing the same sequence counter, which is initialized in message 4 of section 3.2. Note the first packet for the given Rekey SA encrypted with the new KEK attribute will have a Sequence number of 1. If the SA defines an SA TEK payload, this informs the member that a new Data-security SA has been created, with keying material carried in KD (Section 5.6).

If the SA defines a large LKH KEK array (e.g., during group

initialization and batched rekeying), parts of the array MAY be sent in different unique GROUPKEY-PUSH datagrams. However, each of the GROUPKEY-PUSH datagrams MUST be a fully formed GROUPKEY-PUSH datagram. This results in each datagram containing a sequence number and the policy in the SA payload, which corresponds to the KEK array portion sent in the KD payload.

4.1. Use of signature keys

A signing key should not be used in more than one context (e.g., used for host authentication and also for message authentication). Thus, the GCKS SHOULD NOT use the same key to sign the SIG payload in the GROUPKEY-PUSH message as was used for authentication in the GROUPKEY-PULL exchange.

4.2. ISAKMP Header Initialization

Unlike ISAKMP or IKEv1, the cookie pair is completely determined by the GCKS. The cookie pair in the GDOI ISAKMP header identifies the Re-key SA to differentiate the secure groups managed by a GCKS. Thus, GDOI uses the cookie fields as an SPI.

Next Payload identifies an ISAKMP or GDOI payload (see Section 5.0).

Major Version is 1 and Minor Version is 0 according to ISAKMP (Section 3.1 of [RFC2408]).

The Exchange Type has value 33 for the GDOI GROUPKEY-PUSH message.

Flags MUST have the Encryption bit set according to [RFC2008, Section 3.1]. All other bits MUST be set to zero.

Message ID MUST be set to zero.

Length is according to ISAKMP (Section 3.1 of [RFC2408]).

4.3. GCKS Operations

GCKS may initiate a Rekey message for one of several reasons, e.g., the group membership has changed or keys are due to expire.

To begin the rekey datagram the GCKS builds an ISAKMP HDR with the correct cookie pair, and a SEQ payload that includes a sequence number which is one greater than the previous rekey datagram. If the message is using the new KEK attribute for the first time, the SEQ is reset to 1 in this message.

An SA payload is then added. This is identical in structure and

meaning to a SA payload sent in a GROUPKEY-PULL exchange. If there are changes to the KEK (including due to group members being excluded, in the case of LKH), an SA_KEK attribute is added to the SA. If there are one or more new TEKs then SA_TEK attributes are added to describe that policy.

A KD payload is then added. This is identical in structure and meaning to a KD payload sent in a GROUPKEY-PULL exchange. If an SA_KEK attribute was included in the SA payload then corresponding KEK keys (or a KEK update array) is included. A KEK update array is created by first determining which group members have been excluded, and then generating new keys as necessary and distribute LKH update arrays sufficient to provide the new KEK to remaining group members (see Section 5.4.1 of [RFC2627] for details). TEK keys are also sent for each SA_TEK attribute included in the SA payload.

In the penultimate step, the GCKS creates the SIG payload and adds it to the datagram.

Lastly, the payloads following the HDR are encrypted using the current KEK encryption key. The datagram can now be sent.

4.4. Group Member Operations

A group member receiving the GROUPKEY-PUSH datagram matches the cookie pair in the ISAKMP HDR to an existing SA. The message is decrypted, and the form of the datagram is validated. This weeds out obvious ill-formed messages (which may be sent as part of a Denial of Service attack on the group).

The sequence number in the SEQ payload is validated to ensure that it is greater than the previously received sequence number, and that it fits within a window of acceptable values. The SIG payload is then validated. If the signature fails, the message is discarded.

The SA and KD payloads are processed which results in a new GDOI Rekey-SA (if the SA payload included an SA_KEK attribute) and/or new Data-security SAs being added to the system. If the KD payload includes an LKH update array, the group member compares the LKH ID in each key update packet to the LKH IDs that it holds. If it finds a match, it decrypts the key using the key prior to it in the key array and stores the new key in the LKH key array that it holds. The final decryption yields the new group KEK.

If the SA payload includes Data-Security SA including a counter-modes of operation and the receiving group member is a sender for that SA, the group member uses its current SID value with the Data-Security SAs to create counter-mode nonces. If it is a sender and does not

hold a current SID value, it MUST NOT install the Data-Security SAs. It MAY initiate a GROUPKEY-PULL exchange to the GCKS in order to obtain an SID value (along with current group policy).

5. Payloads and Defined Values

This document specifies use of several ISAKMP payloads, which are defined in accordance with RFC 2408. The following payloads are extended or further specified.

Next Payload Type	Value
Security Association (SA)	1
Identification (ID)	5
Nonce (N)	10

Several payload formats specific to the group security exchanges are required.

Next Payload Type	Value
SA KEK Payload (SAK)	15
SA TEK Payload (SAT)	16
Key Download (KD)	17
Sequence Number (SEQ)	18
Group Associated Policy (GAP)	TBD-1

5.1. Identification Payload

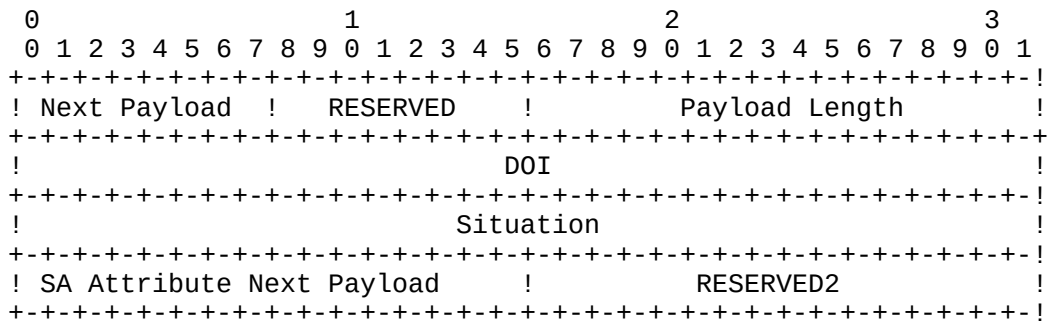
The Identification Payload is defined in RFC 2408. For the GDOI, it is used to identify a group identity that will later be associated with Security Associations for the group. A group identity may map to a specific IP multicast group, or may specify a more general identifier, such as one that represents a set of related multicast streams.

When used with the GDOI, the DOI Specific ID Data field MUST be set to 0.

When used with the GDOI, the ID_KEY_ID ID Type MUST be supported by a conforming implementation, and MUST specify a four (4)-octet group identifier as its value. Implementations MAY also support other ID Types.

5.2. Security Association Payload

The Security Association payload is defined in RFC 2408. For the GDOI, it is used by the GCKS to assert security attributes for both Re-key and Data-security SAs.



The Security Association Payload fields are defined as follows:

o Next Payload (1 octet) -- Identifies the next payload for the GROUPKEY-PULL or the GROUPKEY-PUSH message as defined above. The next payload MUST NOT be a SAK Payload or SAT Payload type, but the next non-Security Association type payload.

o RESERVED (1 octet) -- Must be zero.

o Payload Length (2 octets) -- Is the octet length of the current payload including the generic header and all TEK and KEK payloads.

o DOI (4 octets) -- Is the GDOI, which is value 2.

o Situation (4 octets) -- Must be zero.

o SA Attribute Next Payload (1 octet) -- Must be either a SAK Payload or a SAT Payload. See section 5.2.1 for a description of which circumstances are required for each payload type to be present.

o RESERVED (2 octets) -- Must be zero.

5.2.1. Payloads following the SA payload

Payloads that define specific security association attributes for the KEK and/or TEKs used by the group MUST follow the SA payload. How many of each payload is dependent upon the group policy. There may be zero or one SAK Payload, zero or one GAP Payload, and zero or more SAT Payloads, where either one SAK or SAT payload MUST be present. When present, the order of the SA Attributes payloads must be: KEK, GAP, and TEKs.

This latitude regarding SA Attributes payloads allows various group policies to be accommodated. For example if the group policy does not require the use of a Re-key SA, the GCKS would not need to send an SA KEK attribute to the group member since all SA updates would be

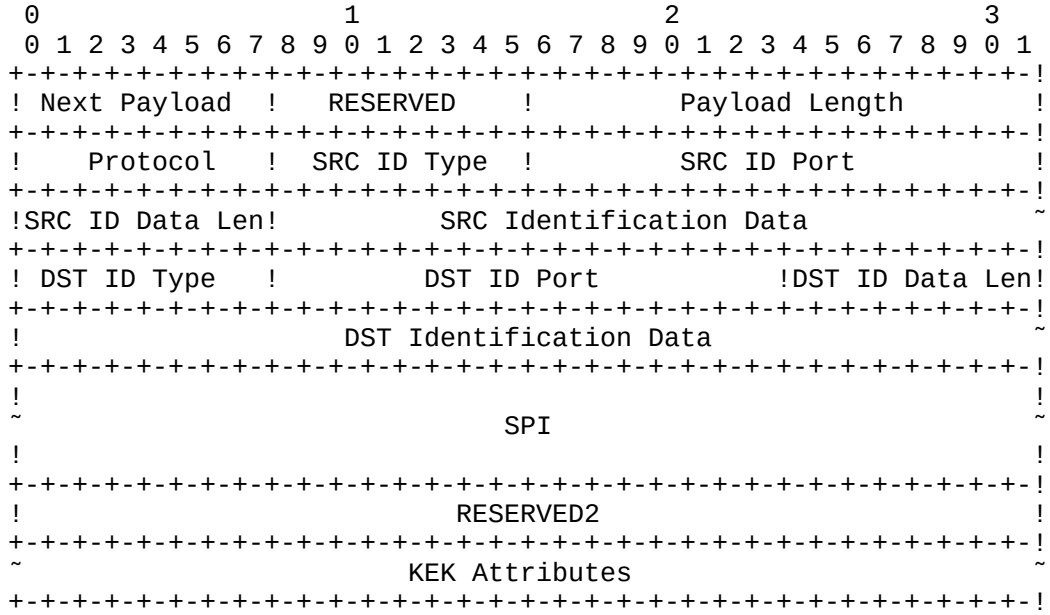
performed using the Registration SA. Alternatively, group policy might use a Re-key SA but choose to download a KEK to the group member only as part of the Registration SA. Therefore, the KEK policy (in the SA KEK attribute) would not be necessary as part of the Re-key SA message SA payload.

Specifying multiple SATs allows multiple sessions to be part of the same group and multiple streams to be associated with a session (e.g., video, audio, and text) but each with individual security association policy.

A GAP payload allows for the distribution of group-wise policy, such as instructions as to when to activate and de-activate SAs.

5.3. SA KEK payload

The SA KEK (SAK) payload contains security attributes for the KEK method for a group and parameters specific to the GROUPKEY-PULL operation. The source and destination identities describe the identities used for the GROUPKEY-PULL datagram.



The SAK Payload fields are defined as follows:

o Next Payload (1 octet) -- Identifies the next payload for the GROUPKEY-PULL or the GROUPKEY-PUSH message. The only valid next

payload types for this message are a GAP Payload, SAT Payload or zero to indicate that no SA Attribute payloads follow.

o RESERVED (1 octet) -- Must be zero.

o Payload Length (2 octets) -- Length of this payload, including the KEK attributes.

o Protocol (1 octet) -- Value describing an IP protocol ID (e.g., UDP/TCP) for the rekey datagram.

o SRC ID Type (1 octet) -- Value describing the identity information found in the SRC Identification Data field. Defined values are specified by the IPsec Identification Type section in the IANA isakmpd-registry [ISAKMP-REG].

o SRC ID Port (2 octets) -- Value specifying a port associated with the source Id. A value of zero means that the SRC ID Port field should be ignored.

o SRC ID Data Len (1 octet) -- Value specifying the length of the SRC Identification Data field.

o SRC Identification Data (variable length) -- Value, as indicated by the SRC ID Type.

o DST ID Type (1 octet) -- Value describing the identity information found in the DST Identification Data field. Defined values are specified by the IPsec Identification Type section in the IANA isakmpd-registry [ISAKMP-REG].

o DST ID Prot (1 octet) -- Value describing an IP protocol ID (e.g., UDP/TCP).

o DST ID Port (2 octets) -- Value specifying a port associated with the source Id.

o DST ID Data Len (1 octet) -- Value specifying the length of the DST Identification Data field.

o DST Identification Data (variable length) -- Value, as indicated by the DST ID Type.

o SPI (16 octets) -- Security Parameter Index for the KEK. The SPI must be the ISAKMP Header cookie pair where the first 8 octets become the "Initiator Cookie" field of the GROUPKEY-PUSH message ISAKMP HDR, and the second 8 octets become the "Responder Cookie" in the same HDR. As described above, these cookies are assigned by the GCKS.

o RESERVED2 (4 octets) -- Must be zero. This bytes represent fields previously defined but no longer used by GDOI.

o KEK Attributes -- Contains KEK policy attributes associated with the group. The following sections describe the possible attributes. Any or all attributes may be optional, depending on the group policy.

5.3.1. KEK Attributes

The following attributes may be present in a SAK Payload. The attributes must follow the format defined in ISAKMP (Section 3.3 of [RFC2408]). In the table, attributes that are defined as TV are marked as Basic (B); attributes that are defined as TLV are marked as Variable (V).

ID Class	Value	Type
RESERVED	0	
KEK_MANAGEMENT_ALGORITHM	1	B
KEK_ALGORITHM	2	B
KEK_KEY_LENGTH	3	B
KEK_KEY_LIFETIME	4	V
SIG_HASH_ALGORITHM	5	B
SIG_ALGORITHM	6	B
SIG_KEY_LENGTH	7	B
KE_OAKLEY_GROUP	8	B
Standards Action	9-127	
Private Use	128-255	
Unassigned	256-32767	

The KEK_MANAGEMENT_ALGORITHM attribute may only be included in a GROUPKEY-PULL message.

5.3.2. KEK_MANAGEMENT_ALGORITHM

The KEK_MANAGEMENT_ALGORITHM class specifies the group KEK management algorithm used to provide forward or backward access control (i.e., used to exclude group members). Defined values are specified in the following table.

KEK Management Type	Value
RESERVED	0
LKH	1
Standards Action	2-127
Private Use	128-255

5.3.2.1. LKH

This type indicates the group management method described in Section 5.4 of [RFC2627]. A general discussion of LKH operations can also be found in Section 6.3 of Multicast and Group Security [HD03]

5.3.3. KEK_ALGORITHM

The KEK_ALGORITHM class specifies the encryption algorithm in which the KEK is used to provide confidentiality for the GROUPKEY-PUSH message. Defined values are specified in the following table. A GDOI implementation MUST abort if it encounters an attribute or capability that it does not understand.

Algorithm Type	Value
RESERVED	0
KEK_ALG_DES	1
KEK_ALG_3DES	2
KEK_ALG_AES	3
Standards Action	4-127
Private Use	128-255
Unassigned	256-32767

If a KEK_MANAGEMENT_ALGORITHM is defined which defines multiple keys (e.g., LKH), and if the management algorithm does not specify the algorithm for those keys, then the algorithm defined by the KEK_ALGORITHM attribute MUST be used for all keys which are included as part of the management.

5.3.3.1. KEK_ALG_DES

This type specifies DES using the Cipher Block Chaining (CBC) mode as described in [FIPS81].

5.3.3.2. KEK_ALG_3DES

This type specifies 3DES using three independent keys as described in "Keying Option 1" in [FIPS46-3].

5.3.3.3. KEK_ALG_AES

This type specifies AES as described in [FIPS197]. The mode of operation for AES is Cipher Block Chaining (CBC) as recommended in [SP.800-38A].

5.3.4. KEK_KEY_LENGTH

The KEK_KEY_LENGTH class specifies the KEK Algorithm key length (in bits). The Group Controller/Key Server (GCKS) adds the KEK_KEY_LENGTH attribute to the SA payload when distributing KEK policy to group members. The group member verifies whether or not it has the capability of using a cipher key of that size. If the cipher definition includes a fixed key length (e.g., KEK_ALG_3DES), the group member can make its decision solely using the KEK_ALGORITHM attribute and does not need the KEK_KEY_LENGTH attribute. Sending the KEK_KEY_LENGTH attribute in the SA payload is OPTIONAL if the KEK cipher has a fixed key length. Also, note that the KEK_KEY_LEN includes only the actual length of the cipher key (the IV length is not included in this attribute).

5.3.5. KEK_KEY_LIFETIME

The KEK_KEY_LIFETIME class specifies the maximum time for which the KEK is valid. The GCKS may refresh the KEK at any time before the end of the valid period. The value is a four (4) octet number defining a valid time period in seconds.

5.3.6. SIG_HASH_ALGORITHM

SIG_HASH_ALGORITHM specifies the SIG payload hash algorithm. The following table defines the algorithms for SIG_HASH_ALGORITHM.

Algorithm Type	Value
RESERVED	0
SIG_HASH_MD5	1
SIG_HASH_SHA1	2
SIG_HASH_SHA256	TBD-2
SIG_HASH_SHA384	TBD-3
SIG_HASH_SHA512	TBD-4
Standards Action	3-127
Private Use	128-255
Unassigned	256-32767

The SHA hash algorithms are defined in the Secure Hash Standard[FIPS.180-2.2002].

If the SIG_ALGORITHM is SIG_ALG_ECDSA-256, SIG_ALG_ECDSA-384, or SIG_ALG_ECDSA-521 the hash algorithm is implicit in the definition, and SIG_HASH_ALGORITHM is not required to be present in a SAK Payload.

5.3.7. SIG_ALGORITHM

The SIG_ALGORITHM class specifies the SIG payload signature algorithm. Defined values are specified in the following table.

Algorithm Type	Value
RESERVED	0
SIG_ALG_RSA	1
SIG_ALG_DSS	2
SIG_ALG_ECDSS	3
SIG_ALG_RSA_PSS	TBD-6
SIG_ALG_ECDSA-256	TBD-7
SIG_ALG_ECDSA-384	TBD-8
SIG_ALG_ECDSA-521	TBD-9
Standards Action	4-127
Private Use	128-255
Unassigned	256-32767

5.3.7.1. SIG_ALG_RSA

This algorithm specifies the RSA digital signature algorithm using the EMSA-PKCS1-v1_5 encoding method, as described in [RFC3447].

5.3.7.2. SIG_ALG_DSS

This algorithm specifies the DSS digital signature algorithm as described in Section 4 of [FIPS186-3].

5.3.7.3. SIG_ALG_ECDSS

This algorithm specifies the Elliptic Curve digital signature algorithm as described in Section 5 of [FIPS186-3]. This definition is deprecated in favor of the SIG_ALG_ECDSA family of algorithms.

5.3.7.4. SIG_ALG_RSA_PSS

This algorithm specifies the RSA digital signature algorithm using the EMSA-PSS encoding method, as described in [RFC3447].

5.3.7.5. SIG_ALG_ECDSA-256

This algorithm specifies the 256-bit Random ECP Group, as described in [RFC5903]. The format of the signature in the SIG payload MUST be as specified in [RFC4754].

5.3.7.6. SIG_ALG_ECDSA-384

This algorithm specifies the 384-bit Random ECP Group, as described in [RFC5903]. The format of the signature in the SIG payload MUST be as specified in [RFC4754].

5.3.7.7. SIG_ALG_ECDSA-521

This algorithm specifies the 521-bit Random ECP Group, as described in [RFC5903]. The format of the signature in the SIG payload MUST be as specified in [RFC4754].

5.3.8. SIG_KEY_LENGTH

The SIG_KEY_LENGTH class specifies the length of the SIG payload key in bits.

5.4. Group Associated Policy

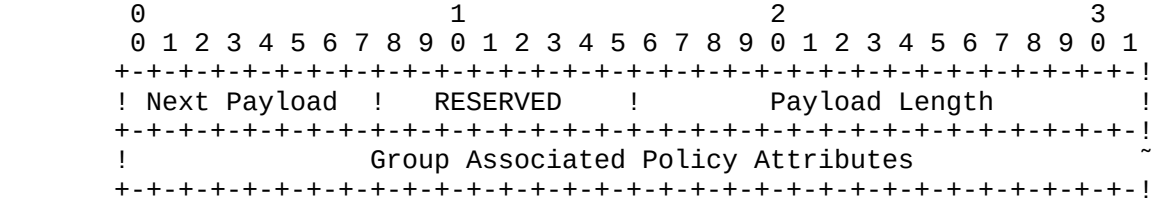
A GCKS may have group-specific policy that is not distributed in an SA TEK or SA KEK. Some of this policy is relevant to all group members, and some is sender-specific policy for a particular group member. The former can be distributed in either a GROUPKEY-PULL or GROUPKEY-PUSH exchange, whereas the latter MUST only be sent in a GROUPKEY-PULL exchange. Additionally, a group member sometimes has the need to make policy requests for resources of the GCKS in a GROUPKEY-PULL exchange. GDOI distributes this associated group policy and policy requests in the Group Associated Policy (GAP) payload.

The GAP payload can be distributed by the GCKS as part of the SA

payload. It follows any SA KEK payload, and is placed before any SA TEK payloads. In the case that group policy does not include an SA KEK, the SA Attribute Next Payload field in the SA payload MAY indicate the SA GAP payload.

The GAP payload can be optionally included by a group member in message 3 of the GROUPKEY-PULL exchange in order to make policy requests.

The SA GAP payload is defined as follows:



The SA GAP payload fields are defined as follows:

- Next Payload (1 octet) -- Identifies the next payload present in the GROUPKEY-PULL or the GROUPKEY-PUSH message. The only valid next payload type for this message is an SA TEK or zero to indicate there are no more security association attributes.

- RESERVED (1 octet) -- Must be zero.

- Payload Length (2 octets) -- Length of this payload, including the SA GAP header and Attributes.

- Group Associated Policy Attributes (variable) -- Contains attributes following the format defined in Section 3.3 of RFC 2408.

Several group associated policy attributes are defined in this memo. An GDOI implementation MUST abort if it encounters an attribute or capability that it does not understand. The values for these attributes are included in the IANA Considerations section of this memo.

5.4.1. ACTIVATION_TIME_DELAY/DEACTIVATION_TIME_DELAY

Section 4.2.1 of RFC 5374 specifies a key rollover method that requires two values be given it from the group key management protocol. The ACTIVATION_TIME_DELAY attribute allows a GCKS to set the Activation Time Delay (ATD) for SAs generated from TEKs. The ATD defines how long after receiving new SAs that they are to be

activated by the GM. The ATD value is in seconds.

The DEACTIVATION_TIME_DELAY allows the GCKS to set the Deactivation Time Delay (DTD) for previously distributed SAs. The DTD defines how long after receiving new SAs that it should deactivate SAs that are destroyed by the re-key event. The value is in seconds.

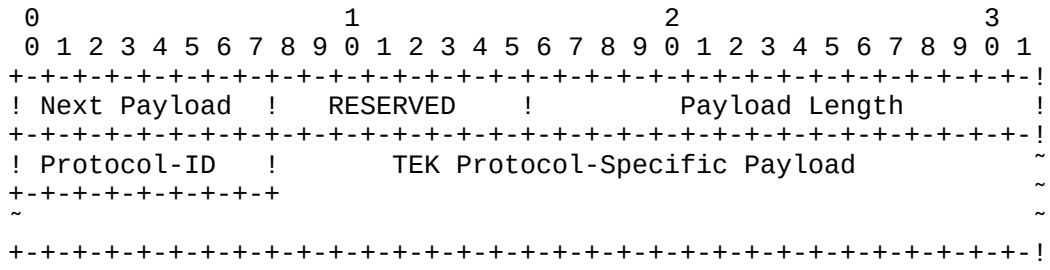
The values of ATD and DTD are independent. However, the DTD value should be larger, which allows new SAs to be activated before older SAs are deactivated. Such a policy ensures that protected group traffic will always flow without interruption.

5.4.2. SENDER_ID_REQUEST

The SENDER_ID_REQUEST attribute is used by a group member to request SIDs during the GROUPKEY-PULL message, and includes a count of how many SID values it desires.

5.5. SA TEK Payload

The SA TEK (SAT) payload contains security attributes for a single TEK associated with a group.



The SAT Payload fields are defined as follows:

o Next Payload (1 octet) -- Identifies the next payload for the GROUPKEY-PULL or the GROUPKEY-PUSH message. The only valid next payload types for this message are another SAT Payload or zero to indicate there are no more security association attributes.

o RESERVED (1 octet) -- Must be zero.

o Payload Length (2 octets) -- Length of this payload, including the TEK Protocol-Specific Payload.

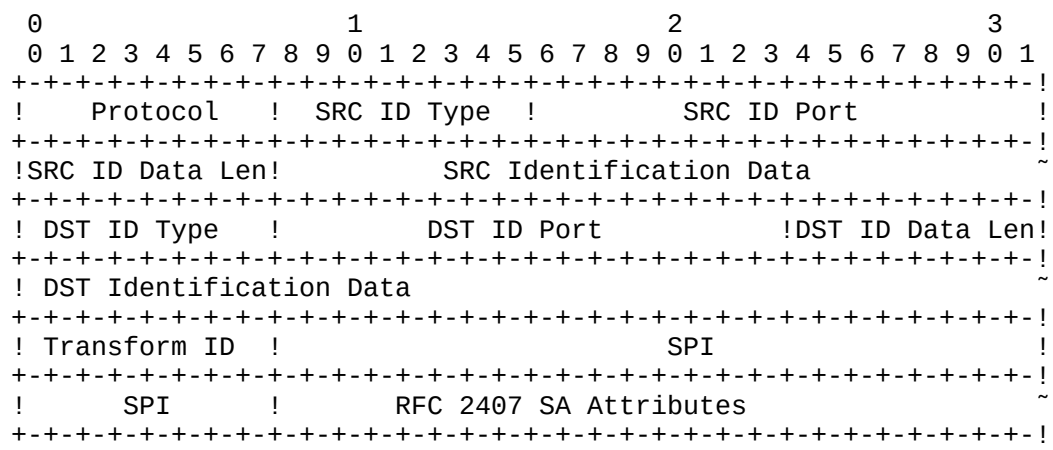
o Protocol-ID (1 octet) -- Value specifying the Security Protocol. The following table defines values for the Security Protocol

Protocol ID	Value
RESERVED	0
GDOI_PROTO_IPSEC_ESP	1
GDOI_PROTO_IPSEC_AH	TBD-5
Standards Action	3-127
Private Use	128-255

o TEK Protocol-Specific Payload (variable) -- Payload which describes the attributes specific for the Protocol-ID.

5.5.1. GDOI_PROTO_IPSEC_ESP/GDOI_PROTO_IPSEC_AH

The TEK Protocol-Specific payload for ESP and AH is as follows:



The SAT Payload fields are defined as follows:

o Protocol (1 octet) -- Value describing an IP protocol ID (e.g., UDP/TCP). A value of zero means that the Protocol field should be ignored.

o SRC ID Type (1 octet) -- Value describing the identity information found in the SRC Identification Data field. Defined values are specified by the IPsec Identification Type section in the IANA isakmpd-registry [ISAKMP-REG].

o SRC ID Port (2 octets) -- Value specifying a port associated with the source Id. A value of zero means that the SRC ID Port field should be ignored.

o SRC ID Data Len (1 octet) -- Value specifying the length of the SRC

Identification Data field.

o SRC Identification Data (variable length) -- Value, as indicated by the SRC ID Type. Set to three bytes of zero for multiple-source multicast groups that use a common TEK for all senders.

o DST ID Type (1 octet) -- Value describing the identity information found in the DST Identification Data field. Defined values are specified by the IPsec Identification Type section in the IANA isakmpd-registry [ISAKMP-REG].

o DST ID Prot (1 octet) -- Value describing an IP protocol ID (e.g., UDP/TCP). A value of zero means that the DST Id Prot field should be ignored.

o DST ID Port (2 octets) -- Value specifying a port associated with the source Id. A value of zero means that the DST ID Port field should be ignored.

o DST ID Data Len (1 octet) -- Value specifying the length of the DST Identification Data field.

o DST Identification Data (variable length) -- Value, as indicated by the DST ID Type.

o Transform ID (1 octet) -- Value specifying which ESP or AH transform is to be used. The list of valid values is defined in the IPsec ESP or IPsec AH Transform Identifiers section of the IANA isakmpd-registry [ISAKMP-REG].

o SPI (4 octets) -- Security Parameter Index for ESP.

o RFC 2407 Attributes -- ESP and AH Attributes from RFC 2407 Section 4.5. The GDOI supports all IPsec DOI SA Attributes for GDOI_PROTO_IPSEC_ESP and GDOI_PROTO_IPSEC_AH excluding the Group Description (section 4.5 of [RFC2407], which MUST NOT be sent by a GDOI implementation and is ignored by a GDOI implementation if received. The following attributes MUST be supported by an implementation supporting ESP and AH: SA Life Type, SA Life Duration, Encapsulation Mode. An implementation supporting ESP must also support the Authentication Algorithm attribute if the ESP transform includes authentication/ The Authentication Algorithm attribute of the IPsec DOI is group authentication in GDOI.

5.5.1.1. New IPsec Security Association Attributes

The Multicast Extensions to the Security Architecture for the Internet Protocol (RFC 5374) introduces new requirements for a group

key management system distributing IPsec policy. It also defines new attributes as part of the Group Security Policy Database (GSPD). These attributes describe policy that a group key management system must convey to a group member in order to support those extensions. The GDOI SA TEK payload distributes IPsec policy using IPsec security association attributes defined in [ISAKMP-REG]. This section defines how GDOI can convey the new attributes as IPsec Security Association Attributes.

5.5.1.1.1. Address Preservation

Applications use the extensions in RFC 5374 to copy the IP addresses into the outer IP header when encapsulating an IP packet as an IPsec tunnel mode packet. This allows an IP multicast packet to continue to be routed as a IP multicast packet. In order for the GDOI group member to appropriately set up the GSPD, the GCKS must provide that policy to the group member.

Depending on group policy, several address preservation methods are possible: no address preservation ("None"), preservation of the original source address ("Source-Only"), preservation of the original destination address ("Destination-Only"), or both addresses ("Source-And-Destination"). The IANA Considerations section of this memo adds the "Address Preservation" security association attribute. If this attribute is not included in a GDOI SA TEK payload provided by a GCKS, then Source-And-Destination address preservation has been defined for the SA TEK.

5.5.1.1.2. SA Direction

Depending on group policy, an IPsec SA created from an SA TEK payload may be required in one or both directions. SA TEK policy used by multiple senders is required to be installed in both the sending and receiving direction ("Symmetric"), whereas SA TEK for a single sender should only be installed in the receiving direction by receivers ("Receiver-Only") and in the sending direction by the sender ("Sender-Only"). The IANA Considerations section of this memo adds the "SA Direction" security association attribute.

An SA TEK payload that does not include the SA Direction attribute is treated as a Symmetric IPsec SA. Note that unless Symmetric may be the only value that can be meaningfully described for an SA TEK distributed in an GROUPKEY-PUSH message. Alternatively, Receiver-Only could be distributed, but group senders would need to be configured to not receive GROUPKEY-PUSH messages in order to retain their role.

5.5.2. Other Security Protocols

Besides ESP and AH, GDOI should serve to establish SAs for secure groups needed by other Security Protocols that operate at the transport, application, and internetwork layers. These other Security Protocols, however, are in the process of being developed or do not yet exist.

The following information needs to be provided for a Security Protocol to the GDOI.

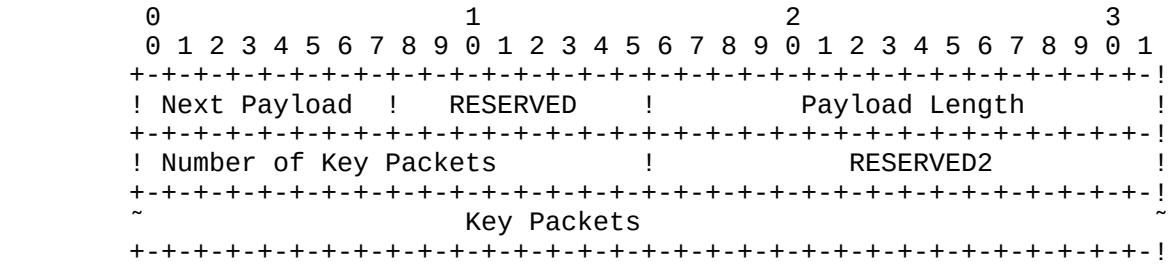
- The Protocol-ID for the particular Security Protocol
- The SPI Size
- The method of SPI generation
- The transforms, attributes and keys needed by the Security Protocol

All Security Protocols must provide the information in the bulleted list above to guide the GDOI specification for that protocol. Definitions for the support of those Security Protocols in GDOI will be specified in separate documents.

A Security Protocol MAY protect traffic at any level of the network stack. However, in all cases applications of the Security Protocol MUST protect traffic which MAY be shared by more than two entities.

5.6. Key Download Payload

The Key Download Payload contains group keys for the group specified in the SA Payload. These key download payloads can have several security attributes applied to them based upon the security policy of the group as defined by the associated SA Payload.



The Key Download Payload fields are defined as follows:

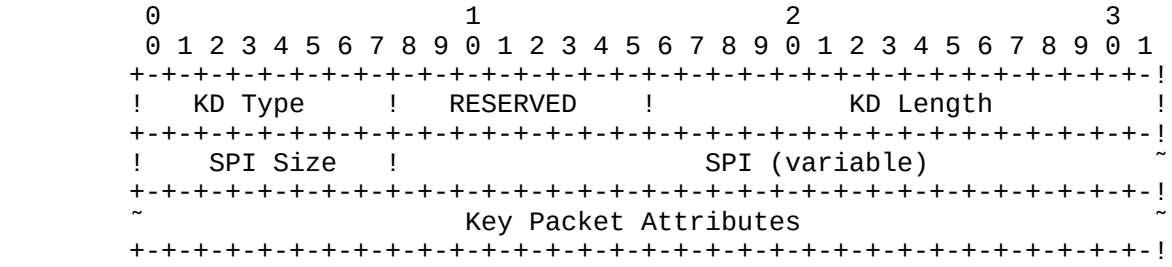
o Next Payload (1 octet) -- Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be zero.

o RESERVED (1 octet) -- Unused, set to zero.

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

o Number of Key Packets (2 octets) -- Contains the total number of both TEK and Rekey arrays being passed in this data block.

o Key Packets (variable) -- Several types of key packets are defined. Each Key Packet has the following format.



o Key Download (KD) Type (1 octet) -- Identifier for the Key Data field of this Key Packet.

Key Download Type	Value
RESERVED	0
TEK	1
KEK	2
LKH	3
SID	TBD-7
Standards Action	4-127
Private Use	128-255

"KEK" is a single key whereas LKH is an array of key-encrypting keys.

o RESERVED (1 octet) -- Unused, set to zero.

o Key Download Length (2 octets) -- Length in octets of the Key Packet data, including the Key Packet header.

o SPI Size (1 octet) -- Value specifying the length in octets of the SPI as defined by the Protocol-Id.

o SPI (variable length) -- Security Parameter Index which matches a SPI previously sent in an SAK or SAT Payload.

o Key Packet Attributes (variable length) -- Contains Key information. The format of this field is specific to the value of the KD Type field. The following sections describe the format of each KD Type.

5.6.1. TEK Download Type

The following attributes may be present in a TEK Download Type. Exactly one attribute matching each type sent in the SAT payload MUST be present. The attributes must follow the format defined in ISAKMP (Section 3.3 of [RFC2408]). In the table, attributes defined as TV are marked as Basic (B); attributes defined as TLV are marked as Variable (V).

TEK Class	Value	Type
RESERVED	0	
TEK_ALGORITHM_KEY	1	V
TEK_INTEGRITY_KEY	2	V
TEK_SOURCE_AUTH_KEY	3	V
Standards Action	4-127	
Private Use	128-255	
Unassigned	256-32767	

If no TEK key packets are included in a Registration KD payload, the group member can expect to receive the TEK as part of a Re-key SA. At least one TEK must be included in each Re-key KD payload. Multiple TEKs may be included if multiple streams associated with the SA are to be rekeyed.

When an algorithm specification specifies the format of the keying material, the value transported in the KD payload for that key is passed according to that specification. The keying material may contain information besides a key. For example, The Use of Galois/ Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP) [RFC4106] defines a salt value as part of KEYMAT.

5.6.1.1. TEK_ALGORITHM_KEY

The TEK_ALGORITHM_KEY class declares that the encryption key for this SPI is contained as the Key Packet Attribute. The encryption algorithm that will use this key was specified in the SAT payload.

In the case that the algorithm requires multiple keys (e.g., 3DES), all keys will be included in one attribute.

DES keys will consist of 64 bits (the 56 key bits with parity bit). Triple DES keys will be specified as a single 192 bit attribute (including parity bits) in the order that the keys are to be used for encryption (e.g., DES_KEY1, DES_KEY2, DES_KEY3).

5.6.1.2. TEK_INTEGRITY_KEY

The TEK_INTEGRITY_KEY class declares that the integrity key for this SPI is contained as the Key Packet Attribute. The integrity algorithm that will use this key was specified in the SAT payload. Thus, GDOI assumes that both the symmetric encryption and integrity keys are pushed to the member. HMAC-SHA1 keys will consist of 160 bits[RFC2404], HMAC-MD5 keys will consist of 128 bits[RFC2403]. HMAC-SHA2 and AES-GMAC keys will have a key length equal to the output length of the hash functions [RFC4868][RFC4543].

5.6.1.3. TEK_SOURCE_AUTH_KEY

The TEK_SOURCE_AUTH_KEY class declares that the source authentication key for this SPI is contained in the Key Packet Attribute. The source authentication algorithm that will use this key was specified in the SAT payload.

5.6.2. KEK Download Type

The following attributes may be present in a KEK Download Type. Exactly one attribute matching each type sent in the SAK payload MUST be present. The attributes must follow the format defined in ISAKMP (Section 3.3 of [RFC2408]). In the table, attributes defined as TV are marked as Basic (B); attributes defined as TLV are marked as Variable (V).

KEK Class	Value	Type
RESERVED	0	
KEK_ALGORITHM_KEY	1	V
SIG_ALGORITHM_KEY	2	V
Standards Action	3-127	
Private Use	128-255	
Unassigned	256-32767	

If the KEK key packet is included, there MUST be only one present in the KD payload.

5.6.2.1. KEK_ALGORITHM_KEY

The KEK_ALGORITHM_KEY class declares the encryption key for this SPI is contained in the Key Packet Attribute. The encryption algorithm that will use this key was specified in the SAK payload.

If the mode of operation for the algorithm requires an IV, an explicit IV MUST be included in the KEK_ALGORITHM_KEY before the actual key.

5.6.2.2. SIG_ALGORITHM_KEY

The SIG_ALGORITHM_KEY class declares that the public key for this SPI is contained in the Key Packet Attribute, which may be useful when no public key infrastructure is available. The signature algorithm that will use this key was specified in the SAK payload.

5.6.3. LKH Download Type

The LKH key packet is comprised of attributes representing different nodes in the LKH key tree.

The following attributes are used to pass an LKH KEK array in the KD payload. The attributes must follow the format defined in ISAKMP (Section 3.3 of [RFC2408]). In the table, attributes defined as TV are marked as Basic (B); attributes defined as TLV are marked as Variable (V).

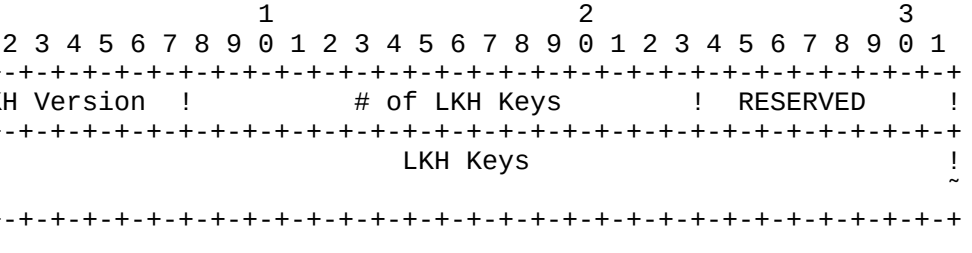
KEK Class	Value	Type
RESERVED	0	
LKH_DOWNLOAD_ARRAY	1	V
LKH_UPDATE_ARRAY	2	V
SIG_ALGORITHM_KEY	3	V
Standards Action	4-127	
Private Use	128-255	
Unassigned	256-32767	

If an LKH key packet is included in the KD payload, there must be only one present.

5.6.3.1. LKH_DOWNLOAD_ARRAY

This attribute is used to download a set of keys to a group member. It MUST NOT be included in a GROUPKEY-PUSH message KD payload if the GROUPKEY-PUSH is sent to more than the group member. If an LKH_DOWNLOAD_ARRAY attribute is included in a KD payload, there must be only one present.

This attribute consists of a header block, followed by one or more LKH keys.

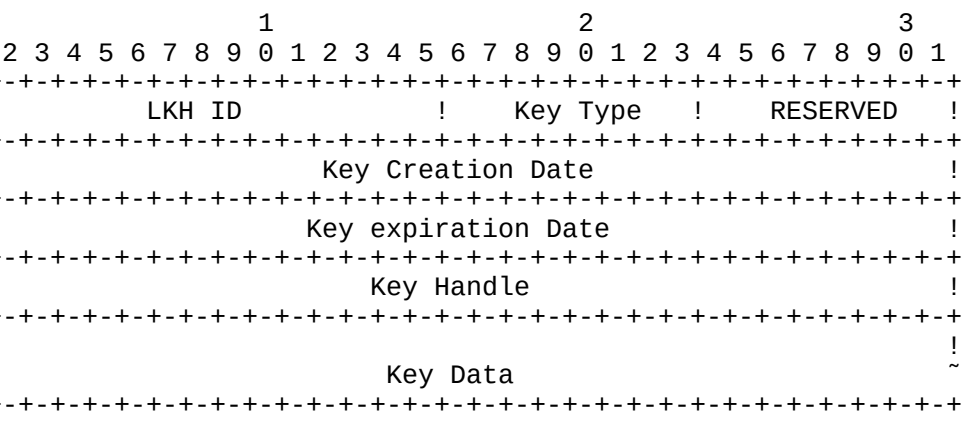


The KEK_LKH attribute fields are defined as follows:

o LKH version (1 octet) -- Version of the LKH data format. Must be one.

o Number of LKH Keys (2 octets) -- This value is the number of distinct LKH keys in this sequence.

o RESERVED (1 octet) -- Unused, set to zero. Each LKH Key is defined as follows:



o LKH ID (2 octets) -- Identity of the LKH node. A GCKS is free to choose the ID in an implementation-specific manner (e.g., the position of this key in a binary tree structure used by LKH).

o Key Type (1 octet) -- Encryption algorithm for which this key data is to be used. This value is specified in Section 5.3.3.

o RESERVED (1 octet) -- Unused, set to zero.

o Key Creation Date (4 octets) -- Time value of when this key data was originally generated. A time value of zero indicates that there is no time before which this key is not valid.

o Key Expiration Date (4 octets) -- Time value of when this key is no longer valid for use. A time value of zero indicates that this key does not have an expiration time.

o Key Handle (4 octets) -- Value assigned by the GCKS to uniquely identify a key within an LKH ID. Each new key distributed by the GCKS for this node will have a key handle identity distinct from previous or successive key handles specified for this node.

o Key Data (variable length) -- Key data, which is dependent on the

Key Type algorithm for its format. If the mode of operation for the algorithm requires an IV, an explicit IV MUST be included in the Key Data field prepended to the actual key.

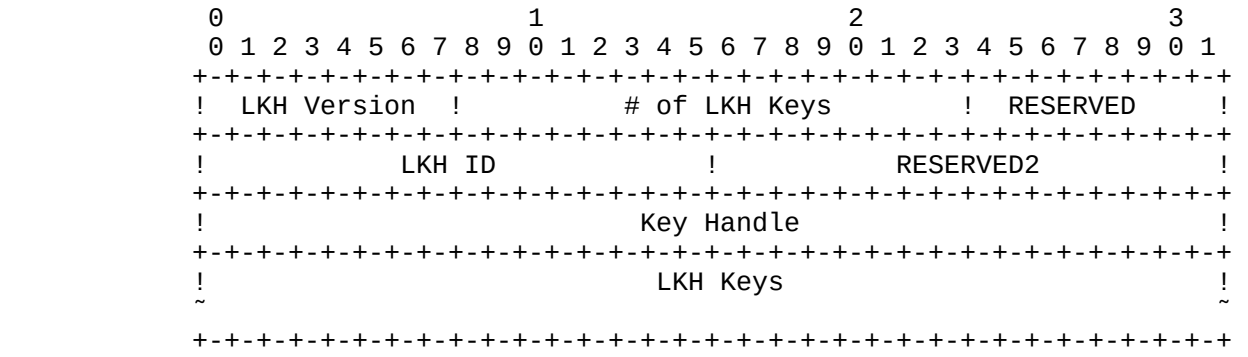
The Key Creation Date and Key expiration Dates MAY be zero. This is necessary in the case where time synchronization within the group is not possible.

The first LKH Key structure in an LKH_DOWNLOAD_ARRAY attribute contains the Leaf identifier and key for the group member. The rest of the LKH Key structures contain keys along the path of the key tree in order from the leaf, culminating in the group KEK.

5.6.3.2. LKH_UPDATE_ARRAY

This attribute is used to update the keys for a group. It is most likely to be included in a GROUPKEY-PUSH message KD payload to rekey the entire group. This attribute consists of a header block, followed by one or more LKH keys, as defined in the previous section.

There may be any number of UPDATE_ARRAY attributes included in a KD payload.



o LKH version (1 octet) -- Version of the LKH data format. Must be one.

o Number of LKH Keys (2 octets) -- Number of distinct LKH keys in this sequence.

o RESERVED (1 octet) -- Unused, set to zero.

o LKH ID (2 octets) -- Node identifier associated with the key used to encrypt the first LKH Key.

o RESERVED2 (2 octets) -- Unused, set to zero.

o Key Handle (4 octets) -- Value assigned by the GCKS to uniquely identify the key within the LKH ID used to encrypt the first LKH Key.

The LKH Keys are as defined in the previous section. The LKH Key structures contain keys along the path of the key tree in order from the LKH ID found in the LKH_UPDATE_ARRAY header, culminating in the group KEK. The Key Data field of each LKH Key is encrypted with the LKH key preceding it in the LKH_UPDATE_ARRAY attribute. The first LKH Key is encrypted under the key defined by the LKH ID and Key Handle found in the LKH_UPDATE_ARRAY header.

5.6.3.3. SIG_ALGORITHM_KEY

The SIG_ALGORITHM_KEY class declares that the public key for this SPI is contained in the Key Packet Attribute, which may be useful when no public key infrastructure is available. The signature algorithm that will use this key was specified in the SAK payload.

5.6.4. SID Download Type

This attribute is used to download one or more Sender-ID (SID) values for the exclusive use of a group member.

SID Class	Value	Type
RESERVED	0	
NUMBER_OF_SID_BITS	1	B
SID_VALUE	2	V
Standards Action	3-128	
Private Use	129-255	
Unassigned	256-32767	

Because a SID value is intended for a single group member, the SID Download type MUST NOT be distributed in a GROUPKEY_PUSH message distributed to multiple group members.

5.6.4.1. NUMBER_OF_SID_BITS

The NUMBER_OF_SID_BITS class declares how many bits of the cipher nonce in which to represent a SID value. This value is applied to each SID value distributed in the SID Download.

5.6.4.2. SID_VALUE

The SID_VALUE class declares a single SID value for the exclusive use of the a group member. Multiple SID_VALUE attributes MAY be included in a SID Download.

5.6.4.3. Group Member Semantics

The SID_VALUE attribute value distributed to the group member MUST be used by that group member as the SID field portion of the IV for all Data-Security SAs including a counter-based mode of operation distributed by the GCKS as a part of this group.

When the Sender-Specific IV (SSIV) field for any Data-Security SA is exhausted, the group member MUST no longer act as a sender on that SA using its active SID. The group member SHOULD re-register, at which time the GCKS will issue a new SID to the group member, along with either the same Data-Security SAs or replacement ones. The new SID replaces the existing SID used by this group member, and also resets the SSIV value to its starting value. A group member MAY re-register prior to the actual exhaustion of the SSIV field to avoid dropping data packets due to the exhaustion of available SSIV values combined with a particular SID value.

A group member MUST NOT process an SID Download Type KD payload present in a GROUPKEY-PUSH message.

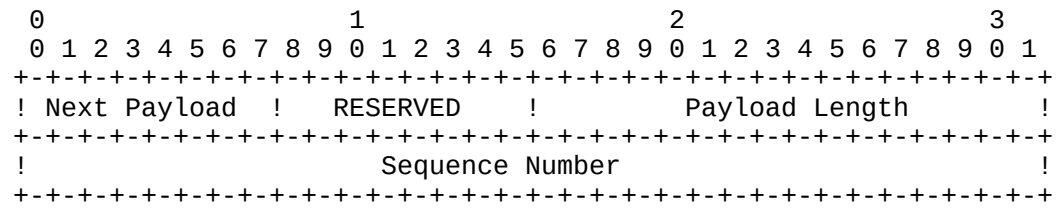
5.6.4.4. GCKS Semantics

If any KD payload includes keying material that is associated with a counter-mode of operation, an SID Download Type KD payload containing at least one SID_VALUE attribute MUST be included.

The GCKS MUST NOT send the SID Download Type KD payload as part of a GROUPKEY-PUSH message, because distributing the same sender-specific policy to more than one group member will reduce the security of the group.

5.7. Sequence Number Payload

The Sequence Number Payload (SEQ) provides an anti-replay protection for GROUPKEY-PUSH messages. Its use is similar to the Sequence Number field defined in the IPsec ESP protocol [RFC4303].



The Sequence Number Payload fields are defined as follows:

o 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 zero.

o RESERVED (1 octet) -- Unused, set to zero.

o Payload Length (2 octets) -- Length in octets of the current payload, including the generic payload header. Must be a value of 8.

o Sequence Number (4 octets) -- This field contains a monotonically increasing counter value for the group. It is initialized to zero by the GCKS, and incremented in each subsequently-transmitted message. Thus the first packet sent for a given Rekey SA will have a Sequence Number of 1. The GDOI implementation keeps a sequence counter as an attribute for the Rekey SA and increments the counter upon receipt of a GROUPKEY-PUSH message. The current value of the sequence number must be transmitted to group members as a part of the Registration SA payload. A group member must keep a sliding receive window. The window must be treated as in the ESP protocol [RFC4303] Section 3.4.3.

5.8. Nonce

The data portion of the Nonce payload (i.e., Ni_b and Nr_b included in the HASHs) MUST be a value between 8 and 128 bytes.

5.9. Delete

There are times the GCKS may want to signal to receivers to delete SAs, for example at the end of a broadcast. Deletion of keys may be accomplished by sending an ISAKMP Delete payload (Section 3.15 of [RFC2408]) as part of a GDOI GROUPKEY-PUSH message.

One or more Delete payloads MAY be placed following the SEQ payload in a GROUPKEY-PUSH message. If a GCKS has no further SAs to send to group members, the SA and KD payloads MUST be omitted from the message.

The following fields of the Delete Payload are further defined as follows:

o The Domain of Interpretation field contains the GDOI DOI.

o The Protocol-Id field contains TEK protocol id values defined in Section 5.5 of this document. To delete a KEK SA, the value of zero MUST be used as the protocol id. Note that only one protocol id value can be defined in a Delete payload. If a TEK SA and a KEK SA must be deleted, they must be sent in different Delete payloads.

There may be circumstances where the GCKS may want to start over with a clean slate. If the administrator is no longer confident in the integrity of the group, the GCKS can signal deletion of all policy of a particular TEK protocol by sending a TEK with a SPI value equal to zero in the delete payload. For example, if the GCKS wishes to remove all the KEKs and all the TEKs in the group, the GCKS SHOULD send a delete payload with a spi of zero and a protocol_id of a TEK protocol_id value, followed by another delete payload with a spi of zero and protocol_id of zero, indicating that the KEK SA should be deleted.

6. Algorithm Selection

For GDOI implementations to interoperate, they must support one or more security algorithms in common. This section specifies the security algorithm implementation requirements for standards-conformant GDOI implementations. In all cases the choices are intended to maintain at least 112 bits of security [SP.800-131].

Algorithms not referenced in this section MAY be used.

6.1. KEK

These tables list the algorithm selections for values related to the KEK.

Requirement	KEK Management Algorithm
SHOULD	LKH

Requirement	KEK Algorithm (notes)
MUST	KEK_ALG_AES with 128-bit keys
SHOULD NOT	KEK_ALG_DES (1)

Requirement	KEK Signature Hash Algorithm (notes)
MUST	SIG_HASH_SHA256
SHOULD	SIG_HASH_SHA1 (2)
SHOULD NOT	SIG_HASH_MD5 (3)

Requirement	KEK Signature Algorithm (notes)
MUST	SIG_ALG_RSA with 2048-bit keys

Notes:

(1) DES, with its small key size and corresponding security strength is of questionable security for general use

(2) The use of SIG_HASH_SHA1 as a signature hash algorithm used with GROUPKEY-PUSH messages remains safe at the time of this writing, and is a widely deployed signature hash algorithm.

(3) Although a real weakness with second preimage resistance with MD5 has not been found at the time of this writing, the security strength of MD5 has been shown to be rapidly declining over time and it’s use should be understood and carefully weighed.

6.2. TEK

The following table lists the requirements for Security Protocol support for an implementation.

Requirement	KEK Management Algorithm
MUST	GDOI_PROTO_IPSEC_ESP

7. Security Considerations

GDOI is a security association (SA) management protocol for groups of senders and receivers. This protocol performs authentication of communicating protocol participants (Group Member, Group Controller/ Key Server). It provides confidentiality of key management messages, and it provides source authentication of those messages. GDOI includes defenses against man-in-middle, connection hijacking, replay, reflection, and denial-of-service (DOS) attacks on unsecured networks. GDOI assumes the network is not secure and may be under the complete control of an attacker.

GDOI assumes that the group members and GCKS are secure even though the network is insecure. GDOI ultimately establishes keys among members of a group, which MUST be trusted to use those keys in an authorized manner according to group policy. An GDOI entity compromised by an attacker may reveal the secrets necessary to eavesdrop on group traffic and/or take the identity of a group sender, so host security is of the utmost important once. The latter threat could be mitigated by using source origin authentication in the Data-Security SAs (e.g., the use of RSA signatures [RFC4359] or TESLA [RFC4082]). The choice of Data-Security SAs is a matter of group policy and is not within the scope of this memo.

There are three phases of GDOI as described in this document: an ISAKMP Phase 1 protocol, the GROUPKEY-PULL exchange protected by the ISAKMP Phase 1 protocol, and the GROUPKEY-PUSH message. Each phase is considered separately below.

7.1. ISAKMP Phase 1

GDOI uses the Phase 1 exchanges defined in [RFC2409] to protect the GROUPKEY-PULL exchange. Therefore all security properties and considerations of those exchanges (as noted in [RFC2409]) are relevant for GDOI.

GDOI may inherit the problems of its ancestor protocols [FS00], such as identity exposure, absence of unidirectional authentication, or stateful cookies [PK01].

7.1.1. Authentication

Authentication is provided via the mechanisms defined in [RFC2409], namely Pre-Shared Keys or Public Key encryption.

7.1.2. Confidentiality

Confidentiality is achieved in Phase 1 through a Diffie-Hellman exchange that provides keying material, and through negotiation of encryption transforms.

The Phase 1 protocol will be protecting encryption and integrity keys sent in the GROUPKEY-PULL protocol. The strength of the encryption used for Phase 1 SHOULD exceed that of the keys send in the GROUPKEY-PULL protocol.

7.1.3. Man-in-the-Middle Attack Protection

A successful man-in-the-middle or connection-hijacking attack foils entity authentication of one or more of the communicating entities during key establishment. GDOI relies on Phase 1 authentication to defeat man-in-the-middle attacks.

7.1.4. Replay/Reflection Attack Protection

In a replay/reflection attack, an attacker captures messages between GDOI entities and subsequently forwards them to a GDOI entity. Replay and reflection attacks seek to gain information from a subsequent GDOI message response or seek to disrupt the operation of a GDOI member or GCKS entity. GDOI relies on the Phase 1 nonce mechanism in combination with a hash-based message authentication code to protect against the replay or reflection of previous key management messages.

7.1.5. Denial of Service Protection

A denial of service attacker sends messages to a GDOI entity to cause that entity to perform unneeded message authentication operations. GDOI uses the Phase 1 cookie mechanism to identify spurious messages prior to cryptographic hash processing. This is a "weak" form of denial of service protection in that the GDOI entity must check for good cookies, which can be successfully imitated by a sophisticated attacker. The Phase 1 cookie mechanism is stateful, and commits memory resources for cookies.

7.2. GROUPKEY-PULL Exchange

The GROUPKEY-PULL exchange allows a group member to request SAs and keys from a GCKS. It runs as a "phase 2" protocol under protection of the Phase 1 security association.

7.2.1. Authentication

Peer authentication is not required in the GROUPKEY-PULL protocol. It is running in the context of the Phase 1 protocol, which has previously authenticated the identity of the peer.

Message authentication is provided by HASH payloads in each message, where the HASH is defined to be over SKEYID_a (derived in the Phase 1 exchange), the ISAKMP Message-ID, and all payloads in the message. Because only the two endpoints of the exchange know the SKEYID_a value, this provides confidence that the peer sent the message.

7.2.2. Confidentiality

Confidentiality is provided by the Phase 1 security association, after the manner described in [RFC2409].

7.2.3. Man-in-the-Middle Attack Protection

Message authentication (described above) includes a secret known only to the group member and GCKS when constructing a HASH payload. This prevents man-in-the-middle and connection-hijacking attacks because an attacker would not be able to change the message undetected.

7.2.4. Replay Protection

A GROUPKEY-PULL message identifies its messages using a cookie pair from the Phase 1 exchange that precedes it. A GROUPKEY-PULL message with invalid cookies will be discarded. Therefore, GDOI messages that are not associated with a current GDOI session will be discarded without further processing.

Replayed GDOI messages that are associated with a current GDOI session will be decrypted and authenticated. The M-ID in the HDR identifies a session. Replayed packets will be processed according to the state machine of that session. Packets not matching that state machine will be discarded without processing.

7.2.5. Denial of Service Protection

GCKS implementations SHOULD keep a record of recently received GROUPKEY-PULL messages (e.g., a hash of the packet) and reject messages that have already been processed. This provides Denial of Service and Replay Protection of previously sent messages. An implementation MAY choose to rate-limit the receipt of GDOI messages in order to mitigate avoid overloading its computational resources.

The GCKS SHOULD NOT perform any computationally expensive tasks

before receiving a HASH with its own nonce included. The GCKS MUST NOT update the group management state (e.g., LKH key tree, SID-counter) until it receives the third message in the exchange with a valid HASH payload including its own nonce.

7.2.6. Authorization

A GCKS implementation SHOULD maintain an authorization list of authorized group members. Group members MUST specifically list each authorized GCKS in its Group Peer Authorization Database (GPAD) [RFC5374].

7.3. GROUPKEY-PUSH Exchange

The GROUPKEY-PUSH exchange is a single message that allows a GCKS to send SAs and keys to group members. This is likely to be sent to all members using an IP multicast group. This message provides an efficient rekey and group membership adjustment capability.

7.3.1. Authentication

The GROUPKEY-PULL exchange distributes a public key that is used for message authentication. The GROUPKEY-PUSH message is digitally signed using the corresponding private key held by the GCKS. This digital signature provides source authentication for the message. Thus, GDOI protects the GCKS from impersonation in group environments.

7.3.2. Confidentiality

The GCKS encrypts the GROUPKEY-PUSH message with an encryption key that was distributed in the GROUPKEY-PULL exchange or a previous GROUPKEY-PUSH exchange. The encryption key may be a simple KEK, or the result of a group management method (e.g., LKH) calculation.

7.3.3. Man-in-the-Middle Attack Protection

This combination of confidentiality and message authentication services protects the GROUPKEY-PUSH message from man-in-middle and connection-hijacking attacks.

7.3.4. Replay/Reflection Attack Protection

The GROUPKEY-PUSH message includes a monotonically increasing sequence number to protect against replay and reflection attacks. A group member will discard sequence numbers associated with the current KEK SPI that have the same or lower value as the most recently received replay number.

Implementations SHOULD keep a record (e.g., a hash value) of recently received GROUPKEY-PUSH messages and reject duplicate messages prior to performing cryptographic operations. This enables an early discard of the replayed messages.

7.3.5. Denial of Service Protection

A cookie pair identifies the security association for the GROUPKEY-PUSH message. The cookies thus serve as a weak form of denial-of-service protection for the GROUPKEY-PUSH message.

The digital signature used for message authentication has a much greater computational cost than a message authentication code and could amplify the effects of a denial of service attack on GDOI members who process GROUPKEY-PUSH messages. The added cost of digital signatures is justified by the need to prevent GCKS impersonation: If a shared symmetric key were used for GROUPKEY-PUSH message authentication, then GCKS source authentication would be impossible and any member would be capable of GCKS impersonation.

The potential of the digital signature amplifying a denial of service attack is mitigated by the order of operations a group member takes, where the least expensive cryptographic operation is performed first. The group member first decrypts the message using a symmetric cipher. If it is a validly formed message then the sequence number is checked against the replay window. Only if the sequence number is valid is the digital signature verified. Thus in order for a denial of service attack to be mounted, an attacker would need to know both the symmetric encryption key used for confidentiality, and a valid sequence number. Generally speaking this means only current group members can effectively deploy a denial of service attack.

7.4. Forward and Backward Access Control

Through GROUPKEY-PUSH, the GDOI supports group management methods such as LKH (section 5.4 of [RFC2627]) that have the property of denying access to a new group key by a member removed from the group (forward access control) and to an old group key by a member added to the group (backward access control). The concepts "forward access control" and "backward access control" have also been described as "perfect forward security" and "perfect backward security" respectively in the literature [RFC2627].

Group management algorithms providing forward and backward access control other than LKH have been proposed in the literature, including OFT [OFT] and Subset Difference [NNL]. These algorithms could be used with GDOI, but are not specified as a part of this document.

7.4.1. Forward Access Control Requirements

When group membership is altered using a group management algorithm new Data-security SAs are usually also needed. New SAs ensure that members who were denied access can no longer participate in the group.

If forward access control is a desired property of the group, new Data-security SAs MUST NOT be included in a GROUPKEY-PUSH message which changes group membership. This is required because the new Data-security SAs are not protected with the new KEK. Instead, two sequential GROUPKEY-PUSH messages must be sent by the GCKS; the first changing the KEK, and the second (protected with the new KEK) distributing the new Data-security SAs.

Note that in the above sequence, although the new KEK can effectively deny access to the group to some group members they will be able to view the new KEK policy. If forward access control policy for the group includes keeping the KEK policy secret as well as the KEK itself secret, then two GROUPKEY-PUSH messages changing the KEK must occur before the new Data-security SAs are transmitted.

If other methods of using LKH or other group management algorithms are added to GDOI, those methods MAY remove the above restrictions requiring multiple GROUPKEY-PUSH messages, providing those methods specify how forward access control policy is maintained within a single GROUPKEY-PUSH message.

7.4.2. Backward Access Control Requirements

If backward access control is a desired property of the group, a new member MUST NOT be given Data-security SAs that were used prior to it joining the group. This can be accomplished if the GCKS provides only the Rekey SA to the new member in a GROUPKEY-PULL exchange, followed by a GROUPKEY-PUSH message that both deletes current Data-security SAs, and provides new replacement Data-security SAs. The new group member will effectively join the group at such time as the existing members begin sending on the Data-security SAs.

If there is a possibility that the new group member has stored GROUPKEY-PUSH messages delivered prior to joining the group, then the above procedure is not sufficient. In this case, to achieve backward access control the GCKS needs to return a new Rekey SA to the group member in a GROUPKEY-PULL exchange rather than the existing one. The GCKS would subsequently deliver two GROUPKEY-PUSH messages. The first, intended for existing group members, distributes the new Rekey-SA to existing members. The GCKS would then deliver the second GROUPKEY-PUSH message using the new Rekey-SA that both deletes

current Data-security SAs, and provides new replacement Data-security SAs. Both preexisting and new members would process the second GROUPKEY-PUSH message, and all would be able to communicate using the new Data-security SAs.

7.5. Derivation of keying material

A GCKS distributes keying material associated with Data-Security SAs and the Rekey SA. Because these security associations are used by a set of group members, this keying material is not related to any pair wise connection, and there is no requirement in The Multicast Group Security Architecture [RFC3740] for group members to permute group keying material. Because the GCKS is solely responsible for the generation of the keying material, the GCKS MUST derive the keying material using a strong random number generator. Because there are no interoperability concerns with key generation, no method is prescribed in GDOI.

8. IANA Considerations

This memo requests IANA to make several additions to existing registries, and to add sever new GDOI registries. When the new registries are added, the following terms are to be applied as described in the Guidelines for Writing an IANA Considerations Section in RFCs [RFC5226]: Standards Action, and Private Use.

8.1. Additions to current registries

The GDOI KEK Attribute named SIG_HASH_ALGORITHM [GDOI-REG] should be assigned several new Algorithm Type values from the RESERVED space to represent the SHA-256, SHA-384, and SHA-512 hash algorithms as defined in [FIPS.180-2.2002]. The new algorithm names should be SIG_HASH_SHA256, SIG_HASH_SHA384, and SIG_HASH_SHA512 respectively and have the values of TBD-2, TBD-3, and TBD-4 respectively.

The GDOI KEK Attributed named SIG_ALGORITHM [GDOI-REG] should be assigned a new Algorithm Type value from the RESERVED space to represent the RSA PSS encoding type. The new algorithm name should be SIG_ALG_RSA_PSS, and has the value of TBD-6.

A new GDOI SA TEK type Protocol-ID type [GDOI-REG] should be assigned from the RESERVED space. The new algorithm id should be called GDOI_PROTO_IPSEC_AH, refers to the IPsec AH encapsulation, and has a value of TBD-5.

A new Next Payload Type [ISAKMP-REG] should be assigned. The new type is called "SA Group Associated Policy (GAP)", and has a value of TBD-1.

A new Key Download Type Section 5.6 should be assigned. The new type is called "SID", and has a value of TBD-7.

8.2. New registries

A new namespace should be created in the GDOI Payloads registry [GDOI-REG] to describe SA GAP Payload Values. The following rules apply to define the attributes in SA SSA Payload Values:

Attribute Type	Value	Type
RESERVED	0	
ACTIVATION_TIME_DELAY	1	B
DEACTIVATION_TIME_DELAY	2	B
SENDER_ID_REQUEST	3	B
Standards Action	4-127	
Private Use	128-255	
Unassigned	256-32767	

A new IPsec Security Association Attribute [ISAKMP-REG] defining the preservation of IP addresses is needed. The attribute class is called "Address Preservation", and it is a Basic type. The following rules apply to define the values of the attribute:

Name	Value
Reserved	0
None	1
Source-Only	2
Destination-Only	3
Source-And-Destination	4
Standards Action	5-61439
Private Use	61440-65535

A new IPsec Security Association Attribute [ISAKMP-REG] defining the SA direction is needed. The attribute class is called "SA Direction", and it is a Basic type. The following rules apply to define the values of the attribute:

Name	Value
Reserved	0
Sender-Only	1
Receiver-Only	2
Symmetric	3
Standards Action	4-61439
Private Use	61440-65535

When the SID "Key Download Type" (described in the previous section) has a set of attributes. The attributes must follow the format defined in ISAKMP (Section 3.3 of [RFC2408]). In the table, attributes defined as TV are marked as Basic (B); attributes defined as TLV are marked as Variable (V).

SID Class	Value	Type
RESERVED	0	
NUMBER_OF_SID_BITS	1	B
SID_VALUE	2	V
Standards Action	3-128	
Private Use	129-255	
Unassigned	256-32767	

8.3. Cleanup of existing registries

Several existing GDOI Payloads registries do not use the terms in RFC 5226 and/or do not describe the entire range of possible values. The following sections correct these registries.

8.3.1. Pop Algorithm

Values 4-127 are to be designated Standards Action. Values 256-32767 are to be added and designated Unassigned.

8.3.2. KEK Attributes

Values 9-127 are to added and designated Standards Action. Values 128-255 are to be added and designated Private Use. Values 256-32767 are to be added and designated Unassigned.

8.3.3. KEK_MANAGEMENT_ALGORITHM

Values 2-127 are to be designated Standards Action. Values 256-65535 are to be added and designated Unassigned.

8.3.4. KEK_ALGORITHM

Values 4-127 are to be designated Standards Action. Values 256-65535 are to be added and designated Unassigned.

8.3.5. SIG_HASH_ALGORITHM

Values 3-127 are to be designated Standards Action. Values 256-65535 are to be added and designated Unassigned.

8.3.6. SIG_ALGORITHM

Values 4-127 are to be designated Standards Action. Values 256-65535 are to be added and designated Unassigned.

8.3.7. SA TEK Payload Values

Values 2-127 are to be designated Standards Action.

8.3.8. Key Download Types

Values 4-127 are to be designated Standards Action.

8.3.9. TEK Download Type

Values 4-127 are to be added and designated Standards Action. Values 128-255 are to be added and designated Private Use. Values 256-32767 are to be added and designated Unassigned.

8.3.10. KEK Download Type

Values 3-127 are to be designated Standards Action. Values 128-255 are to be added and designated Private Use. Values 256-32767 are to be added and designated Unassigned.

8.3.11. LKH Download Type

Values 4-127 are to be designated Standards Action. Values 256-32767 are to be added and designated Unassigned.

9. Acknowledgements

This text updates RFC 3547, and the authors wish to thank Mark Baugher and Hugh Harney for their extensive contributions that led to this updated version of GDOI.

The authors are grateful to Catherine Meadows for her careful review and suggestions for mitigating the man-in-the-middle attack she had previously identified. Yoav Nir and Vincent Roca provided many useful technical and editorial comments and suggestions for improvement.

10. References

10.1. Normative References

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

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

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

[RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast Extensions to the Security Architecture for the Internet Protocol", RFC 5374, November 2008.

[RFC6054] McGrew, D. and B. Weis, "Using Counter Modes with Encapsulating Security Payload (ESP) and Authentication Header (AH) to Protect Group Traffic", RFC 6054, November 2010.

10.2. Informative References

[FIPS.180-2.2002]
National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-2, August 2002, <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>.

[FIPS186-3]
"Digital Signature Standard (DSS)", United States of America, National Institute of Science and Technology Federal Information Processing Standard (FIPS) 186-2, June 2009.

[FIPS197] "Advanced Encryption Standard (AES)", United States of America, National Institute of Science and Technology Federal Information Processing Standard (FIPS) 197, November 2001.

[FIPS46-3]
"Data Encryption Standard (DES)", United States of America, National Institute of Science and Technology Federal Information Processing Standard (FIPS) 46-3, October 1999.

[FIPS81] "DES Modes of Operation", United States of America, National Institute of Science and Technology Federal

Information Processing Standard (FIPS) 81, December 1980.

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

[GDOI-REG] Internet Assigned Numbers Authority, "Group Domain of Interpretation (GDOI) Payload Type Values", IANA Registry, December 2004, <http://www.iana.org/assignments/gdoi-payloads>.

[HD03] Hardjono, T. and L. Dondeti, "Multicast and Group Security", Artech House Computer Security Series, ISBN 1-58053-342-6, 2003.

[I-D.weis-gdoi-mac-tek] Weis, B. and S. Rowles, "GDOI Generic Message Authentication Code Policy", draft-weis-gdoi-mac-tek-02 (work in progress), March 2011.

[ISAKMP-REG] "'Magic Numbers' for ISAKMP Protocol", <http://www.iana.org/assignments/isakmp-registry>.

[MP04] Meadows, C. and D. Pavlovic, "Deriving, Attacking, and Defending the GDOI Protocol", ESORICS 2004 pp. 53-72, September 2004.

[NNL] Naor, D., Noal, M., and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers", Advances in Cryptology, Crypto '01, Springer-Verlag LNCS 2139, 2001, pp. 41-62, 2001, <http://www.wisdom.weizmann.ac.il/~naor/>.

[OFT] McGrew, D. and A. Sherman, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees", Manuscript, submitted to IEEE Transactions on Software Engineering, 1998, <http://download.nai.com/products/media/nai/misc/oft052098.ps>.

[PK01] Perlman, R. and C. Kaufman, "Analysis of the IPsec Key Exchange Standard", WET-ICE conference , 2001, <http://sec.femto.org/wetice-2001/papers/radia-paper.pdf>.

[RFC2403] Madson, C. and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", RFC 2403, November 1998.

[RFC2404] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2408] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

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

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3686] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.

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

[RFC3947] Kivinen, T., Swander, B., Huttunen, A., and V. Volpe, "Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.

[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.

[RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", RFC 4106, June 2005.

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

[RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)", RFC 4309, 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.

[RFC4430] Sakane, S., Kamada, K., Thomas, M., and J. Vilhuber, "Kerberized Internet Negotiation of Keys (KINK)", RFC 4430, March 2006.

[RFC4543] McGrew, D. and J. Viega, "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH", RFC 4543, May 2006.

[RFC4754] Fu, D. and J. Solinas, "IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 4754, January 2007.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, May 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5903] Fu, D. and J. Solinas, "Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2", RFC 5903, June 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[SP.800-131] Barker, E. and A. Roginsky, "Recommendation for the Transitioning of Cryptographic Algorithms and Key Lengths", United States of America, National Institute of Science and Technology DRAFT NIST Special Publication 800-131, June 2010.

[SP.800-38A] Dworkin, M., "Recommendation for Block Cipher Modes of Operation", United States of America, National Institute

of Science and Technology NIST Special Publication 800-38A 2001 Edition, December 2001.

Appendix A. Extending GDOI

A.1. Alternate GDOI Phase 1 protocols

This section describes a manner in which other protocols could be used as GDOI Phase 1 protocols in place of the ISAKMP Phase 1 protocol. However, they are not specified as a part of this document. A separate document MUST be written in order for another protocol to be used as a GDOI Phase 1 protocol.

Other possible phase 1 protocols are also described in [RFC4046].

Any GDOI phase 1 protocol MUST satisfy the requirements specified in Section 2 of this document.

A.1.1. IKEv2 Exchange

Version 2 of the IKE protocol (IKEv2) [RFC5996] has been published. That protocol simplifies IKE processing, and combines the two phases of IKE. An IKEv2 Phase 1 negotiates an IPsec SA during phase 1, which was not possible in IKE. However, IKEv2 also defines a phase 2 protocol. The phase 2 protocol is protected by the Phase 1, similar in concept to how IKE Quick Mode is protected by the IKE Phase 1 protocols in [RFC2409].

It would be possible to define GDOI as a phase 2 protocol protected by an IKEv2 initial exchange. Alternatively, it would be possible to define a new protocol re-using some of the IKEv2 initial exchange (e.g., IKE_SA_INIT).

A.1.2. KINK Protocol

The Kerberized Internet Negotiation of Keys (KINK) [RFC4430] has defined a method of encapsulating an IKEv1 Quick Mode [RFC2409] encapsulated in Kerberos KRB_AP_REQ and KRB_AP_REP payloads. KINK provides a low-latency, computationally inexpensive, easily managed, and cryptographically sound method of setting up IPsec security associations.

The KINK message format includes a DOI field in the KINK header. The [RFC4430] document defines the DOI for the IPsec DOI.

A new DOI for KINK could be defined which would encapsulate a GROUPKEY-PULL exchange in the Kerberos KRB_AP_REQ and KRB_AP_REP payloads. As such, GDOI would benefit from the computational efficiencies of KINK.

A.2. Supporting new SA TEK types

Not all secure multicast or multimedia applications can use IPsec ESP or AH. Many Real Time Transport Protocol applications, for example, require security above the IP layer to preserve RTP header compression efficiencies and transport-independence [RFC3550]. Alternatively, GDOI can distribute message authentication code (MAC) policy and keys for legacy applications that have defined their own security associations [I-D.weis-gdoi-mac-tek].

In order to add a new data security protocol, a new RFC MUST specify the data-security SA parameters conveyed by GDOI for that security protocol; these parameters are listed in Section 5.5.2 of this document.

Data security protocol SAs MUST protect group traffic. GDOI provides no restriction on whether that group traffic is transmitted as unicast or multicast packets.

Appendix B. GDOI Applications

GDOI can be used to distribute keys for several secure multicast applications, where different applications have different key management requirements. This section outlines two example ways that GDOI can be used. Other examples can be found in Section 10 of [HD03].

A simple application is secure delivery of periodic multicast content over an organization's IP network, perhaps a multicast video broadcast. Assuming the content delivery time frame is bounded and the group membership is not expected to change over time, there is no need for group policy to include a GROUPKEY-PUSH exchange, and there's no need for the GCKS to distribute a Re-key SA. Thus, the GDOI GCKS may only need to distribute a single set of Data-Security SAs to protect the time-bounded broadcast.

In contrast, a persistent IP multicast application (e.g., stock-ticker delivery service) may have many group members, where the group membership changes over time. A periodic change of Data-security SAs may be desirable, and the potential for change in group membership requires the use of a group management method enabling de-authorization of group members. The GDOI GCKS will distribute the current set of Data-Security SAs and a Re-key SA to registering group members. It will then deliver regularly-scheduled GROUPKEY-PUSH protocol delivering the new SAs for the group. Additionally, the group membership on the GCKS may be frequently adjusted, which will result in GROUPKEY-PUSH exchange delivering a new Rekey SAs protected by a group management method. Each GROUPKEY-PUSH may include Data-security SAs and/or a Rekey SA.

In each example the relevant policy is defined on the GCKS and relayed to group members using the GROUPKEY-PULL and/or GROUPKEY-PUSH protocols. Specific policy choices configured by the GCKS administrator depends on each application.

Appendix C. Significant Changes from RFC 3547

The following significant changes have been made from RFC 3547.

- The Proof of Possession (POP) payload was removed from the GROUPKEY-PULL exchange. It provided an alternate form of authorization, but its use was underspecified. Furthermore, Meadows and Pavlovic [MP04] discussed a man-in-the-middle attack on the POP authorization method, which would require changes to its semantics. No known implementation of RFC 3547 supported the POP payload, so it was removed. Removal of the POP payload obviated the need for the CERT payload in that exchange and it was removed as well.

- The Key Exchange Payloads (KE_I, KE_R) payloads were removed from the GROUPKEY-PULL exchange. However, the specification for computing keying material for the additional encryption function in RFC 3547 is faulty. Furthermore, it has been observed that because the GDOI registration message uses strong ciphers and provides authenticated encryption, additional encryption of the keying material in a GDOI registration message provides negligible value. Therefore, the use of KE payloads is deprecated in this memo.

- The Certificate Payload (CERT) was removed from the GROUPKEY-PUSH exchange. The use of this payload was underspecified. In all known use cases, the public key of used to verify the GROUPKEY-PUSH payload is distributed directly from the key server as part of the GROUPKEY-PULL exchange.

- Supported cryptographic algorithms were changed to meet current guidance. Implementations are required to support AES with 128-bit keys to encrypt the rekey message, and SHA-256 for cryptographic signatures. The use of DES is deprecated.

- New protocol support for AH.

- New protocol definitions were added to conform to the most recent Security Architecture for the Internet Protocol [RFC4301] and the Multicast Extensions to the Security Architecture for the Internet Protocol[RFC5374]. This includes addition of the GAP payload.

- New protocol definitions and semantics were added to support Using Counter Modes with ESP and AH to Protect Group Traffic[RFC6054].

- Specification to IANA to better clarify the use of the GDOI Payloads registry.

Authors' Addresses

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

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

Sheela Rowles
Cisco Systems
170 W. Tasman Drive
San Jose, California 95134-1706
USA

Phone: +1-408-527-7677
Email: sheela@cisco.com

Thomas Hardjono
MIT
77 Massachusetts Ave.
Cambridge, Massachusets 02139
USA

Phone: +1-781-729-9559
Email: hardjono@mit.edu

MSEC Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

S. Rowles
A. Yeung, Ed.
P. Tran
Cisco Systems

March 14, 2011

Group Key Management using IKEv2
draft-yeung-g-ikev2-02

Abstract

This document presents a new group key distribution protocol, using group key distribution RFC 3547 with IKEv2 RFC 5996. The new protocol is similar to IKEv2 in message and payload formats as well as message semantics. The protocol is in conformance with MSEC key management architecture that it contains two components: member registration and group rekeying, both downloading group security associations from the Group Controller Key Server to a member of the group.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and Overview

This document presents a group key management protocol protected by IKEv2. The group is protected by the security association derived in the mutual authentication between the group member and the group controller/key server (GCKS) using IKEv2 [RFC5996]. The GCKS downloads policy and keys after the GCKS authenticates the client. The initial exchange uses IKE_SA_INIT exchange in IKEv2. The new payloads for G-IKEv2 are added in the IKE_AUTH exchange. The result of the IKE_AUTH is that the GCKS downloads policy and keys for the group to the Group Members (GM). This document will reference the IKEv2 RFCs [5996 and 4718] but otherwise is intended to be a standalone document. [RFC3547] presented GDOI using the ISAKMP domain of interpretation. This document is updating the group security protocol to use IKEv2 without any need for a domain of interpretation, but will instead distinguish G-IKEv2 from IKEv2 by the port being used. The message semantics of IKEv2 will be maintained in that all communications consist of pairs of messages. The exception is in the case that when rekeys are issued in a multicast domain, the previous model [RFC3547] will be maintained: a multicast rekey sent by the GCKS will not expect a response from the GM. A number of payloads were deemed unnecessary since [RFC3547]. These are described in Appendix A.

1.1. Why do we need another GSA protocol?

GDOI protocol specified in [RFC3547] is protected by IKEv1 phase1 security association defined in [RFC2407], [RFC2408] and [RFC2409]; these documents are obsoleted and replaced by a new version of the IKE protocol defined in RFC 5996. G-IKEv2 provides group key management between the group member and group controller key server using the new IKEv2 protocol and inherits the following key advantages over GDOI:

1. Provide a simple mechanism for the responder to keep minimal state and avoid DOS attack from forged IP address using cookie challenge exchange.

2. Improve performance and network latency by the reduced number of initial messages to complete the G-IKEv2 protocol from (9 messages in main mode and quick mode, 6 messages in aggressive mode and quick) to 4 messages.

3. Fix cryptographic weakness with authentication HASH (ikev1 authentication HASH specified in RFC-2409 does not include all ISAKMP payloads and does not include ISAKMP header). This issue is documented at [IKE-HASH]

4. Improve protocol reliability where all unicast messages are ack'ed and sequenced.

5. Well defined behavior for error conditions to improve interoperability.

1.2. G-IKEv2 Payloads

1. IDg (group ID) - The GM requests the GCKS for membership into the group by sending its IDg payload.

2. GSA (Group Security Association) - The GCKS sends the group policy to the GM using this payload.

3. GSA KEK (Group Security Association Key Encryption Key) - The KEK Payload MAY be sent as part of the group policy to ensure that the GCKS will send rekeys using the security credentials of the KEK.

4. GSA GAP (Group Associated Policy) - The GAP payload allows for the request of sender specific information as well as the distribution of group-wise policy [Section 4.6].

5. GSA TEK (Group Security Association Traffic Encryption Key) - The GSA TEK Payload MAY be sent as part of the group policy to ensure that the GCKS will send the keying material for the group members to communicate securely amongst each other.

6. KD (Key Download) - The GCKS sends the control and data keys to the GM using the KD payload.

7. SEQ (Sequence Number Payload) - The SEQ payload provides anti-replay protection for the rekey message.

2. G-IKEv2 integration into IKEv2 protocol

The G-IKEv2 protocol provides the security mechanisms of IKEv2 (peer authentication, confidentiality, message_integrity) to protect the group negotiations required for G-IKEv2. The G-IKEv2 exchange further provides group authorization, and secure policy and key download from the GCKS to its group members.

2.1. UDP port

G-IKEv2 SHOULD use port 848 since GDOI [RFC3547] and G-IKEv2 are related protocols where both provide group key management between group member and the group controller key server. The version number in the IKEv2 header distinguishes the G-IKEv2 protocol from GDOI protocol [RFC3547].

3. G-IKEv2 Protocol

3.1. G-IKEv2 member registration and secure channel establishment

The registration protocol consists of two exchanges, GSA_INIT and GSA_AUTH. Each exchange consists of request/response pairs. The first exchange GSA_INIT is the same as IKE_SA_INIT is defined in IKEv2 [RFC5996]. This exchange negotiates cryptographic algorithms, exchanges nonces and does a Diffie-Hellman exchange between the member and the Group Controller Key Server (GCKS).

The second exchange GSA_AUTH authenticates the previous messages, exchange identities and certificates, and downloads the data security keys (TEKs) and/or group key encrypting key (KEK) or KEK array. Parts of these messages are encrypted and integrity protected with keys established through the GSA_INIT exchange, so the identities are hidden from eavesdroppers and all fields in all the messages are authenticated. The GCKS MAY authorize group members to be allowed into the group as part of the GSA_AUTH exchange. In the following descriptions, the payloads contained in the message are indicated by names as listed below.

Notation	Payload
AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
GSA	Group Security Association
HDR	IKEv2 Header
IDg	Identification - Group
IDi	Identification - Initiator
IDr	Identification - Responder
KD	Key Download
KE	Key Exchange
Ni, Nr	Nonce SA Security Association
SEQ	Sequence Number of rekey message

The details of the contents of each payload are described in Section 4. Payloads that may optionally appear will be shown in brackets, such as [CERTREQ], indicate that optionally a certificate request payload can be included.

3.1.1. GSA_INIT exchange

```
     Member (Initiator)            GCKS (Responder)
    --------------------          ------------------
     HDR, SAi1, KEi, Ni    -->
                           <--     HDR, SAr1, KEr, Nr, [CERTREQ,]
```

The group member initiates the GSA_INIT exchange to the GCKS to negotiate cryptographic algorithms, exchange nonces, and perform a Diffie-Hellman exchange in the same manner as the IKE_SA_INIT exchange.

3.1.2. GSA_AUTH exchange

The security properties of the GSA_AUTH exchange are the same as the properties of the IKE_SA_AUTH exchange. It is used to authenticate the GSA_INIT messages, exchange identities and certificates. G-IKEv2 also uses this exchange for group member registration and optionally authorization.

```
     Initiator (Member)                              Responder (GCKS)
    --------------------                            ------------------
     HDR, SK { IDi, [CERT,] [CERTREQ,] [IDr,] AUTH,
               IDg, GAP }          -->
```

After an unauthenticated secure channel is established by IKE_SA_INIT exchange, the member initiates a registration request to join a group indicated by the IDg payload. The GM MUST include the GAP payload to request resources from the GCKS. Note the GAP payload is used by the initiator to request GCKS resources, and the GAP payload is sent by the GCKS to provide group based information.

```
                                        <-- HDR, SK { IDr, [CERT,] AUTH,
                                                      [SEQ,] GSA, KD }
```

The GCKS MAY inform the group member the current value of the rekey sequence number using the SEQ payload. The first GSA_PUSH sequence number the member receives MUST be greater than SEQ value. The SEQ payload MUST be present if the GSA payload contains a GSA KEK attribute, indicating that the GCKS will be sending rekeys.

The GCKS also informs the member of the cryptographic policies of the group in the GSA payload, which contains the KEK and/or TEK policy, and/or the policy in the GAP and the authentication transforms. The KD payload contains the KEK and/or TEK keying material. The SPIs for the data traffic are also determined by the GCKS and downloaded in the GSA payload. The GSA KEK attribute contains the G-IKEv2 SPI for the Re-key SA, which is not negotiated but downloaded. The GSA TEK attribute contains a SPI as defined in Section 4.5.1 of this document. If a Re-key SA is defined in the SA payload, indicated by

the presence of the GSA KEK attribute, then the KD will contain the GSA KEK; if one or more Data-security SAs are defined in the GSA payload, the KD will contain the TEKs. The GAP payload may also provide the ATD or DTD providing specifying activation and deactivation delays for SAs generated from the TEKs. The KD payload MAY specify the sender specific information if any of the AES counter-based modes are being used to provide unique sender information to the GMs.

G-IKEv2 member registration may have a few more messages exchanged if the EAP method, cookie challenge (for DoS) and invalid KE are used.

In addition to the IKEv2 error handling, GCKS can reject the registration request when IDg is invalid or authorization fail, etc. In these cases, see Section 4.10, the IKE_AUTH response will include notify indicate errors. The member SHOULD send an IKEv2 delete using the INFORMATIONAL message exchange to bring down the authenticated IKE SA.

3.1.3. GSA_PULL Exchange

When a secure channel is already established between GM and KS, the GM registration for another group can reuse the established secure channel. In this scenario the GM will use the PULL exchange by including the desired group ID (IDg) to request data security keys (TEKs) and/or group key encrypting keys (KEKs) from the GCKS. The GM MUST also include the GAP payload to request resources from the GCKS.

```
 Initiator (Member)                          Responder (GCKS)
--------------------                       ------------------
 HDR, SK {IDg, GAP } -->

                                      <--    HDR, SK { [SEQ], GSA, KD }
```

3.1.4. IKEv2 Header Initialization

The Major Version is (2) and Minor Version number (0) according to IKEv2 [RFC5996], and maintained in this document. The G-IKEv2 GSA-INIT uses the SPI according to IKEv2 [RFC5996],section 2.6.

3.1.5. GM Registration Operations

A G-IKEv2 Initiator (GM) requesting registration contacts the GCKS using the GSA_INIT exchange and receives the response from the GCKS. This exchange is unchanged from the IKE_SA_INIT in IKEv2 protocol. Upon completion of parsing and verifying the GSA_INIT response, the GM sends the GSA_AUTH message with the IKEv2 payloads from IKE_SA_AUTH along with the Group ID and the GAP payload informing the

GCKS of the group the initiator wishes to join. The initiator determines how many Sender-ID values it would like to receive and adds the SENDER_ID_REQUEST in the GAP payload since there is a possibility the Data Security SA supports a counter mode cipher [section 3.2]. The initiator then parses the response from the GCKS authenticating the exchange using the IKEv2 payloads, then accessing the SEQ, if present, GSA, and KD.

If SEQ payload is present, the sequence number in the SEQ payload must be checked against any previously received sequence number for this group. If it is less than the previously received number, it should be considered stale and ignored. This could happen if two GSA_AUTH exchanges happened in parallel, and the sequence number changed.

The GSA is parsed providing the TEK and/or KEK and/or the GAP policy. The GSA payload contains policy describing the security protocol and cryptographic protocols used by the group. This policy describes the Re-key SA, if present, Data-security SAs, and other group policy. If the policy in the GSA payload is acceptable to the GM, it continues parsing the remaining payload. Otherwise, the GM SHOULD tear down the session after notifying the GCKS. Finally the KD is parsed providing the keying material for the TEK and/or KEK. The GM interprets the KD key packets, where each key packet includes the keying material for SAs distributed in the GSA payload. Keying material is matched by comparing the SPIs in the key packets to SPIs previously sent in the GSA payloads. Once TEK keys and policy are matched, the GM provides them to the data security subsystem, and it is ready to send or receive packets matching the TEK policy. If this group has a KEK, the KEK policy and keys are marked as ready for use, and the GM knows to expect the sequence number reset to 1 with the next Rekey SA, which will be encrypted with the new KEK attribute. The GM is now ready to receive GSA_PUSH messages.

3.1.6. GCKS Registration Operations

A G-IKEv2 GCKS passively listens for incoming requests from group members. The GCKS receives the GSA_INIT request message and responds with the GSA_INIT response and authenticates the GM with the same properties as IKEv2.

Upon receiving the GSA_AUTH message, and after authenticating the peer, the GCKS locates the group the GM wishes to join, extracts the policy for that group, and includes the SEQ payload (if the GCKS sends rekey messages), generates the policy in the GSA payload, including the GSA KEK, optionally the GAP, and/or SA TEK payloads according to GCKS policy., along with the keying material in the KD payload. The GAP payload MAY include the ATD or DTD [section 4.6.1]

if it is desired to address the activation and deactivation time delays of the TEK SA. If one or more Data Security SAs distributed in the GSA payload included a counter mode of operation, the GCKS includes at least one SID value in the KD payload, and possibly more depending on the request received in the GAP payload requesting the number of SIDs from the GM. If the GCKS desires authorization, the GCKS authorizes the peer against the specified credentials before sending the GSA_AUTH response.

If the GCKS receives a GSA PULL exchange with a request to register the same GM to another group, the GCKS will need to authorize the GM with the new group (IDg) and respond with corresponding group policy and keys. If the GCKS fails to authorize the GM, it will respond with the AUTHORIZATION_FAILED notify message.

3.2. Counter-based modes of operation

Several new counter-based modes of operation have been specified for ESP (e.g., AES-CTR [RFC3686], AES-GCM [RFC4106], AES-CCM [RFC4309], AES-GMAC [RFC4543]) and AH (e.g., AES-GMAC [RFC4543]). These counter-based modes require that no two senders in the group ever send a packet with the same Initialization Vector (IV) using the same cipher key and mode. This requirement is met in GDOI when the following requirements are met:

o The GCKS distributes a unique key for each Data-Security SA.

o The GCKS uses the method described in [RFC6054], which assigns each sender a portion of the IV space by provisioning each sender with one or more unique SID values.

When at least one Data-Security SA included in the group policy includes a counter-mode, the GCKS automatically allocates and distributes one SID to each group member acting in the role of sender on the Data-Security SA. The SID value is used exclusively by the group member to which it was allocated. The group member uses the same SID for each Data-Security SA specifying the use of a counter-based mode of operation. A GCKS MUST distribute unique keys for each Data-Security SA including a counter-based mode of operation in order to maintain a unique key and nonce usage.

A group member MUST request as many SIDs matching the number of encryption modules in which it will be installing the TEKs in the outbound direction. Alternatively, a group member MAY request more than one SID and use them serially. This could be useful when it is anticipated that the group member will exhaust their range of Data-Security SA nonces using a single SID too quickly (e.g., before the time-based policy in the TEK expires).

When group policy includes a counter-based mode of operation, a GCKS SHOULD use the following method to allocate SID values, which ensures that each SID will be allocated to just one group member.

1. A GCKS maintains an SID-counter, which records the SIDs that have been allocated. SIDs are allocated sequentially, with the first SID allocated to be zero.

2. Each time an SID is allocated, the current value of the counter is saved and allocated to the group member. The SID-counter is then incremented in preparation for the next allocation.

3. When the GCKS specifies a counter-based mode of operation in the Data Security SA, and a group member is a sender, a group member may request a count of SIDs in a GAP payload. When the GCKS receives this request, it increments the SID- counter once for each requested SID, and distributes each SID value to the group member.

4. A GCKS allocates new SID values for each GSA_PULL exchange originated by a sender, regardless of whether a group member had previously contacted the GCKS. In this way, the GCKS does not have a requirement of maintaining a record of which SID values it had previously allocated to each group member. More importantly, since the GCKS cannot reliably detect whether the group member had sent data on the current group Data-Security SAs it does not know what Data-Security counter-mode nonce values that a group member has used. By distributing new SID values, the key server ensures that each time a conforming group member installs a Data- Security SA it will use a unique set of counter-based mode nonces.

5. When the SID-counter maintained by the GCKS reaches its final SID value, no more SID values can be distributed. Before distributing any new SID values, the GCKS MUST delete the Data- Security SAs for the group, followed by creation of new Data- Security SAs, and resetting the SID-counter to its initial value.

6. The GCKS SHOULD send a GSA_PUSH message deleting all Data-Security SAs and the Rekey SA for the group. This will result in the group members initiating a new GSA_PULL exchange, in which they will receive both new SID values and new Data-Security SAs. The new SID values can safely be used because they are only used with the new Data-Security SAs. Note that deletion of the Rekey SA is necessary to ensure that group members receiving a GSA_PUSH exchange before the re-register do not inadvertently use their old SIDs with the new Data-Security SAs. Using the method above, at no time can two group members use the same IV values with the same Data-Security SA key.

3.3. G-IKEv2 group maintenance channel

The GCKS MAY send the GSA Rekey if the KEK attribute is present in the G-IKEv2 registration. Though the G-IKEv2 Rekey is optional, it plays a crucial role for large and dynamic groups. The GCKS is responsible for rekeying of the secure group per the group policy. The GCKS uses multicast to transport the rekey message. The G-IKEv2 protocol uses GSA_REKEY exchange type in G-IKEv2 header identifying it as a rekey message. This rekey message is protected by the registration exchanges.

3.3.1. G-IKEv2 GSA_PUSH exchange

The GCKS initiates the G-IKEv2 Rekey securely using IP multicast. Since multicast rekey does not require a response and it sends to multiple GMs, G-IKEv2 Rekeying SHOULD not support windowing. The anti-replay protection is supported by the SEQ payload. The GCKS Rekey message replaces the Rekey GSA KEK or KEK array, and/or creates a new Data-Security GSA TEK. The SID Download attribute [section 4.7.4] in the Key Download payload SHOULD NOT be part of the Rekey Exchange as this is sender specific information and the Rekey Exchange is group specific. The GCKS initiates the GSA_REKEY exchange as following:

```
    Members (Responder)            GCKS (Initiator)
   --------------------           ------------------
                          <--      HDR, SK { SEQ, GSA, KD, AUTH }
```

HDR is defined in Section 4.1. The SEQ payload is defined in Section 4.8. The GSA payload contains the current rekey and data security SA payloads. The GSA may contain a new data security SA and/or a new rekey SA, which, optionally contains an LKH rekey SA, Section 4.3.

The KD represents the keys for the policy sent in the GSA. If the data security SA is being refreshed in this rekey message, the IPSec keys are updated in the KD, and/or if the rekey SA is being refreshed in this rekey message, the rekey Key or the LKH KEK array is updated in the KD payload.

The AUTH payload is a signature of the hash of the message, not including the G-IKEv2 header, to ensure the integrity of the rekey message.

After adding the Signature of the above Hash to the rekey message, the current KEK encryption key encrypts all the payloads following the HDR.

3.3.2. Forward and Backward Access Control

Through G-IKEv2 rekey, the G-IKEv2 supports algorithms such as LKH that have the property of denying access to a new group key by a member removed from the group (forward access control) and to an old group key by a member added to the group (backward access control). An unrelated notion to PFS, "forward access control" and "backward access control" have been called "perfect forward security" and "perfect backward security" in the literature [RFC2627].

Group management algorithms providing forward and backward access control other than LKH have been proposed in the literature, including OFT [OFT] and Subset Difference [NNL]. These algorithms could be used with G-IKEv2, but are not specified as a part of this document.

Support for group management algorithms is supported via the KEY_MANAGEMENT_ALGORITHM attribute which is sent in the SA_KEK payload. G-IKEv2 specifies one method by which LKH can be used for forward and backward access control. Other methods of using LKH, as well as other group management algorithms such as OFT or Subset Difference may be added to G-IKEv2 as part of a later document. Any such addition MUST be due to a Standards Action as defined in [RFC2434].

3.3.3. Forward Access Control Requirements

When group membership is altered using a group management algorithm new SA_TEKs (and their associated keys) are usually also needed. New SAs and keys ensure that members who were denied access can no longer participate in the group.

If forward access control is a desired property of the group, new SA_TEKs and the associated key packets in the KD payload MUST NOT be included in a G-IKEv2 rekey message which changes group membership. This is required because the SA_TEK policy and the associated key packets in the KD payload are not protected with the new KEK. A second G-IKEv2 rekey message can deliver the new SA_TEKS and their associated keys because it will be protected with the new KEK, and thus will not be visible to the members who were denied access.

If forward access control policy for the group includes keeping group policy changes from members that are denied access to the group, then two sequential G-IKEv2 rekey messages changing the group KEK MUST be sent by the GCKS. The first G-IKEv2 rekey message creates a new KEK for the group. Group members, which are denied access, will not be able to access the new KEK, but will see the group policy since the G-IKEv2 rekey message is protected under the current KEK. A

subsequent G-IKEv2 rekey message containing the changed group policy and again changing the KEK allows complete forward access control. A G-IKEv2 rekey message MUST NOT change the policy without creating a new KEK.

If other methods of using LKH or other group management algorithms are added to G-IKEv2, those methods MAY remove the above restrictions requiring multiple G-IKEv2 rekey messages, providing those methods specify how forward access control policy is maintained within a single G-IKEv2 rekey message.

3.3.4. Deletion of SAs

There are occasions the GCKS may want to signal to receivers to delete policy at the end of a broadcast, or if group policy has changed. Deletion of keys MAY be accomplished by sending the G-IKEv2 Delete Payload [RFC4306], section 3.11 as part of the G-IKEv2 Rekey Exchange.

One or more Delete payloads MAY be placed following the HDR payload in the G-IKEv2 Rekey Exchange. The Protocol-ID field contains TEK protocol id values, defined in section 4.6 of this document. In order to delete a KEK SA, the value of zero MUST be used as the protocol id. Note that only one protocol id value can be defined in a Delete payload. If a TEK and a KEK SA must be deleted, they must be sent in different Delete payloads. Similarly, if a TEK specifying ESP and a TEK specifying AH need to be deleted, they must be sent in different Delete payloads.

When a policy delete is required the GCKS sends a rekey of the following format:

```
 Members (Responder)            GCKS (Initiator)
--------------------           ------------------
                        <--     HDR, SK { SEQ, DEL, [GSA], [KD], AUTH }
```

The GSA MAY specify the remaining active time of the remaining policy by using the DTD attribute in the GAP Payload. If a GCKS has no further SAs to send to group members, the SA and KD payloads MUST be omitted from the message. There may be circumstances where the GCKS may want to start over with a clean slate. If the administrator is no longer confident in the integrity of the group, the GCKS can signal deletion of all policy of a particular TEK protocol by sending a TEK with a SPI value equal to zero in the delete payload. For example, if the GCKS wihses to remove all the KEKs and all the TEKs in the group, the GCKS SHOULD send a delete payload with a spi of zero and a protocol_id of a TEK protocol_id value define in Section 4.5, followed by another delete payload with a spi of zero

and protocol_id of zero, indicating that the KEK SA should be deleted.

3.3.5. GSA_PUSH GCKS Operations

The GCKS may initiate a rekey message if group membership and/or policy has changed, or if the keys are about to expire. The GCKS builds the rekey message with value of the SEQ payload that is one greater than the previous rekey. If the message is using a new KEK attribute, the SEQ is reset to 1 in this message. The GSA and KD follow with the same characteristics as in the GSA Registration exchange. The AUTH payload is created by hashing the string "G-IKEv2" and the message created so far, and then digitally signed. Finally, the payloads following the HDR are encrypted using the current KEK encryption key.

3.3.6. GSA_PUSH GM Operations

The group member receives the Rekey Message from the GCKS, decrypts the message using the current KEK, validates the signature using the public key retrieved in a previous G-IKEv2 exchange, verifies the value in SEQ payload is one or more greater than that of the last GSA rekey received, and processes the GSA and KD payloads. The group member then downloads the new data security SA and/or new Rekey GSA. The parsing of the payloads is identical to the registration exchange.

If the SA payload includes Data-Security SA including a counter-modes of operation and the receiving group member is a sender for that SA, the group member uses its current SID value with the Data-Security SAs to create counter-mode nonces. If it is a sender and does not hold a current SID value, it MUST NOT install the Data-Security SAs. It MAY initiate a GSA_PULL exchange to the GCKS in order to obtain an SID value (along with current group policy).

4. Header and Payload Formats

Refer to IKEv2 [RFC5996] for existing payloads.

4.1. The G-IKEv2 Header

G-IKEv2 uses the same IKE header format as specified in RFC 5996 section 3.1.

Several new payload formats are required in the group security exchanges.

Next Payload Type	Value
Group Identification (IDg)	TBD
Group Security Association (GSA)	TBD
GSA KEK Payload (GSAK)	TBD
GSA GAP Payload (GGAP)	TBD
GSA TEK Payload (GSAT)	TBD
Key Download (KD)	TBD
Sequence Number Payload (SEQ)	TBD

New exchange types IKE_AUTH and GSA_REKEY are added to the IKEv2 [RFC5996] protocol.

Exchange Type	Value
GSA_INIT	TBD
GSA_AUTH	TBD
GSA_PULL	TBD
GSA_PUSH	TBD

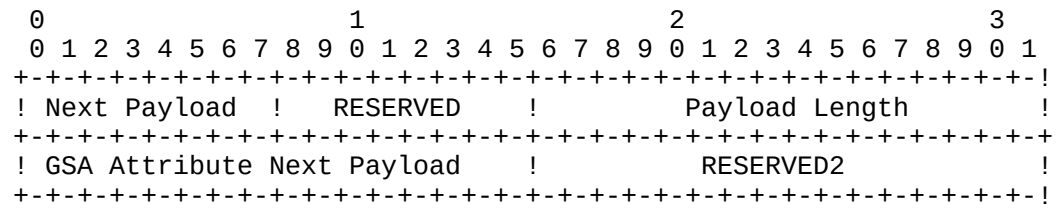
Major Version is 2 and Minor Version is 0 as in IKEv2 [RFC5996]. IKE SA initiator SPI, IKE SA responder SPI, Flags, Message Id are as specified in [RFC5996].

4.2. IDgroup Payload

The IDg Payload allows the group member to indicate which group it wants to join. The payload is constructed by using the IKEv2 [RFC5996] Identification Payload.

4.3. Group Security Association Payload

The Group Security Association payload is used by the GCKS to assert security attributes for both Re-key and Data-security SAs.



The Security Association Payload fields are defined as follows:

- Next Payload (1 octet) -- Identifies the next payload for the G-IKEv2 registrationG-IKEv2 registration or the G-IKEv2 rekey message as defined above. The next payload MUST NOT be a GSAK Payload or GSAT Payload type, but the next non-Security Association type payload.

- RESERVED (1 octet) -- Must be zero.

- Payload Length (2 octets) -- Is the octet length of the current payload including the generic header and all TEK and KEK payloads.

- GSA Attribute Next Payload (1 octet) -- Must be either a GSAK Payload or a GSAT Payload or GAP payload. See Section 4.3.1 for a description of which circumstances are required for each payload type to be present.

- RESERVED2 (2 octets) -- Must be zero.

4.3.1. Payloads following the GSA Payload

Payloads that define specific security association attributes for the KEK and/or TEKs used by the group MUST follow the GSA payload. How many of each payload is dependent upon the group policy. There may be zero or one GSA KEK Payload, zero or more GAP Payloads, and zero or more GSA TEK Payloads, where either one GSA KEK or GSA TEK payload MUST be present. When present, the order of the SA attribute payloads MUST be: KEK, GAP(s), TEK(s).

This latitude allows various group policies to be accommodated. For example if the group policy does not require the use of a Re-key SA, the GCKS would not need to send an GSA KEK attribute to the group member since all SA updates would be performed using the Registration SA. Alternatively, group policy might use a Re-key SA but choose to download a KEK to the group member only as part of the Registration SA. Therefore, the KEK policy (in the GSA KEK attribute) would not be necessary as part of the Re-key SA message GSA payload.

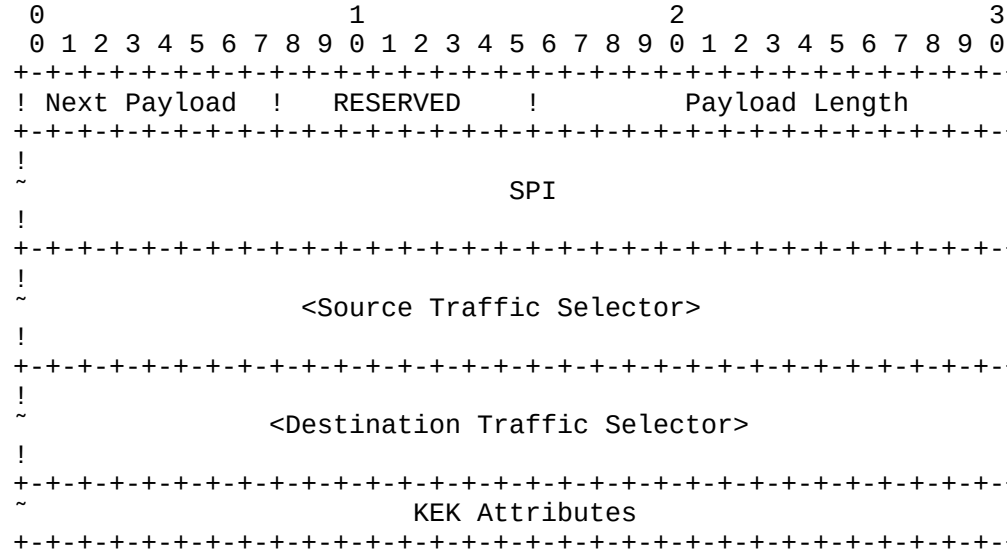
Specifying multiple GSA TEKs allows multiple sessions to be part of

the same group and multiple streams to be associated with a session (e.g., video, audio, and text) but each with individual security association policy.

A GAP payload allows for the distribution of group-wise policy, such as instructions as to when to activate and de-activate SAs.

4.4. KEK Payload

The GSA KEK (GSAK) payload contains security attributes for the KEK method for a group and parameters specific to the G-IKEv2 registration operation. The source and destination identities describe the identities used for the G-IKEv2 registration datagram.



The GSAK Payload fields are defined as follows:

- Next Payload (1 octet) -- Identifies the next payload for the G-IKEv2 registration or the G-IKEv2 rekey message. The only valid next payload types for this message are a GSA TEK Payload or zero to indicate there is no GSA TEK payload.

- RESERVED (1 octet) -- Must be zero.

- Payload Length (2 octets) -- Length of this payload, including the KEK attributes.

- SPI (16 octets) -- Security Parameter Index for the KEK. The SPI must be the IKEv2 Header SPI pair where the first 8 octets become

the "Initiator's SPI" field of the G-IKEv2 rekey message IKEv2 HDR, and the second 8 octets become the "Responder's SPI" in the same HDR. As described above, these SPIs are assigned by the GCKS.

- o Source & Destination Traffic Selectors - Substructures describing the source and destination of the identities. These identities refer to the source and destination of the next KEK rekey SA. Defined format and values are specified by IKEv2 [RFC5996], section 3.13.1.

- o KEK Attributes -- Contains KEK policy attributes associated with the group. The following sections describe the possible attributes. Any or all attributes may be optional, depending on the group policy.

4.4.1. KEK Attributes

The following attributes may be present in a GSA KEK Payload. The attributes must follow the format defined in IKEv2 [RFC5996] section 3.3.5. In the table, attributes that are defined as TV are marked as Basic (B); attributes that are defined as TLV are marked as Variable (V).

ID Class	Value	Type
RESERVED	0	
KEK_MANAGEMENT_ALGORITHM	1	B
KEK_ALGORITHM	2	B
KEK_KEY_LENGTH	3	B
KEK_KEY_LIFETIME	4	V
AUTH_HASH_ALGORITHM	5	B

The following attributes may only be included in a G-IKEv2 registration message: KEK_MANAGEMENT_ALGORITHM.

Minimum attributes that must be sent as part of an GSA KEK: KEK_ALGORITHM, KEK_KEY_LENGTH (if the cipher definition includes a variable length key), KEK_KEY_LIFETIME and AUTH_HASH_ALGORITHM (except for DSA based algorithms).

4.4.2. KEK_MANAGEMENT_ALGORITHM

The KEK_MANAGEMENT_ALGORITHM class specifies the group KEK management algorithm used to provide forward or backward access control (i.e., used to exclude group members). Defined values are specified in the following table.

KEK Management Type	Value
RESERVED	0
LKH	1
Standards Action	2-127
Private Use	128-255

4.4.3. KEK_ALGORITHM

The KEK_ALGORITHM class specifies the encryption algorithm using with the KEK. Defined values are specified in the following table.

Algorithm Type	Value
RESERVED	0
KEK_ALG_AES_CBC	1
KEK_ALG_AES_GCM	2
Standards Action	3-127
Private Use	128-255

If a KEK_MANAGEMENT_ALGORITHM is defined which defines multiple keys (e.g., LKH), and if the management algorithm does not specify the algorithm for those keys, then the algorithm defined by the KEK_ALGORITHM attribute MUST be used for all keys which are included as part of the management.

4.4.3.1. KEK_ALG_AES_CBC

This algorithm specifies AES as described in [FIPS197]. The mode of operation for AES is Cipher Block Chaining (CBC) as recommended in [SP800-38A].

4.4.3.2. KEK_ALG_AES_GCM

This algorithm specifies AES as described in [FIPS197]. The mode of operation for AES is Galois/Counter Mode (GCM) as recommended in [SP800-38D].

4.4.4. KEK_KEY_LENGTH

The KEK_KEY_LENGTH class specifies the KEK Algorithm key length (in bits).

The Group Controller/Key Server (GCKS) adds the KEK_KEY_LEN attribute to the GSA payload when distributing KEK policy to group members. The group member verifies whether or not it has the capability of using a cipher key of that size. If the cipher definition includes a fixed key length, the group member can make its decision solely using

KEK_ALGORITHM attribute and does not need the KEK_KEY_LEN attribute. Sending the KEK_KEY_LEN attribute in the GSA payload is OPTIONAL if the KEK cipher has a fixed key length.

4.4.5. KEK_KEY_LIFETIME

The KEK_KEY_LIFETIME class specifies the maximum time for which the KEK is valid. The GCKS may refresh the KEK at any time before the end of the valid period. The value is a four (4) octet number defining a valid time period in seconds.

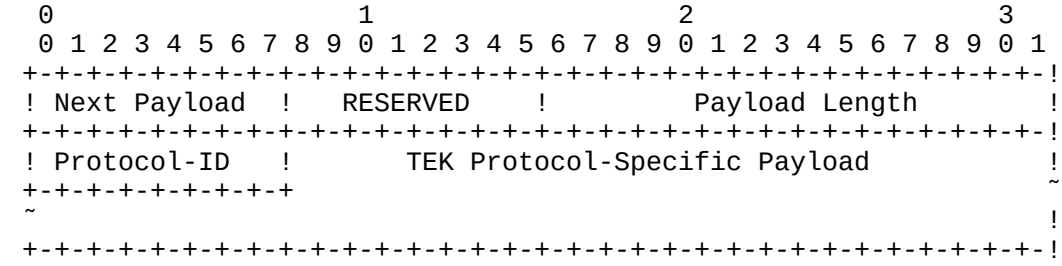
4.4.6. AUTH_HASH_ALGORITHM

AUTH_HASH_ALGORITHM specifies the AUTH payload hash algorithm. The following tables define the algorithms for AUTH_HASH_ALGORITHM.

Algorithm Type	Value
RESERVED	0
AUTH_HASH_SHA256	1
AUTH_HASH_SHA384	2
AUTH_HASH_SHA512	3
Standards Action	4-127
Private Use	128-255

4.5. GSA TEK Payload

The GSA TEK (GSAT) payload contains security attributes for a single TEK associated with a group.



The GSAT Payload fields are defined as follows:

- Next Payload (1 octet) -- Identifies the next payload for the G-IKEv2 registration or the G-IKEv2 rekey message. The only valid next payload types for this message are another GSAT Payload or zero to indicate there are no more security association attributes.

- o RESERVED (1 octet) -- Must be zero.

- o Payload Length (2 octets) -- Length of this payload, including the TEK Protocol-Specific Payload.

- o Protocol-ID (1 octet) -- Value specifying the Security Protocol. The following table defines values for the Security Protocol

Protocol ID	Value
RESERVED	0
GSA_PROTO_IPSEC_ESP	1
GSA_PROTO_IPSEC_AH	2
Standards Action	3-127
Private Use	128-255

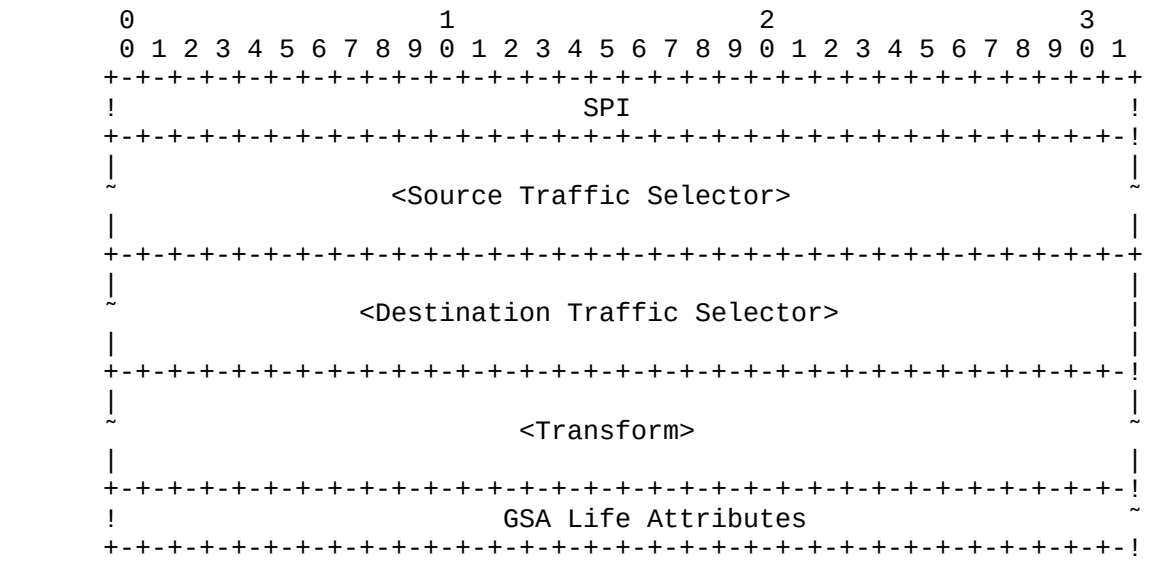
Support for the GSA_PROTO_IPSEC_AH GSA TEK is OPTIONAL.

- o TEK Protocol-Specific Payload (variable) -- Payload which describes the attributes specific for the Protocol-ID.

4.5.1. TEK ESP and AH Protocol-Specific Payload

The TEK Protocol-Specific payload contains of two traffic selectors for source and destination of the protecting traffic, SPI, Transform, and GSA Life Attributes.

The TEK Protocol-Specific payload for ESP is as follows:



The GSAT Payload fields are defined as follows:

- SPI (4 octets) -- Security Parameter Index.

- Source & Destination Traffic Selectors - The traffic selectors describe the source and the destination of the protecting traffic. The format and values are defined in IKEv2 [RFC5996], section 3.13.1.

- Transform -- A substructure specifies the transform information. The format and values are defined in IKEv2 [RFC5996], section 3.3.2.

- GSA Life Attributes -- The GSA Life Attributes are defined as below. The attributes must follow the format defined in IKEv2 [RFC5996], section 3.3.5.

Attribute Types

class	value	type
GSA Life Type	1	B
GSA Life Duration	2	V

Class Values

GSA Life Type
GSA Duration

Specifies the time-to-live for the overall security association. When the GSA expires, all keys downloaded under the association (AH or ESP) must be re-rekeyed. The life type values are:

RESERVED	0
seconds	1
kilobytes	2

Values 3-61439 are reserved to IANA and will be allocated using the Standards Action method. Values 61440-65535 are for private use. For a given Life Type, the value of the Life Duration attribute defines the actual length of the component lifetime -- either a number of seconds, or a number of Kbytes that can be protected.

If unspecified, the default value shall be assumed to be 28800 seconds (8 hours).

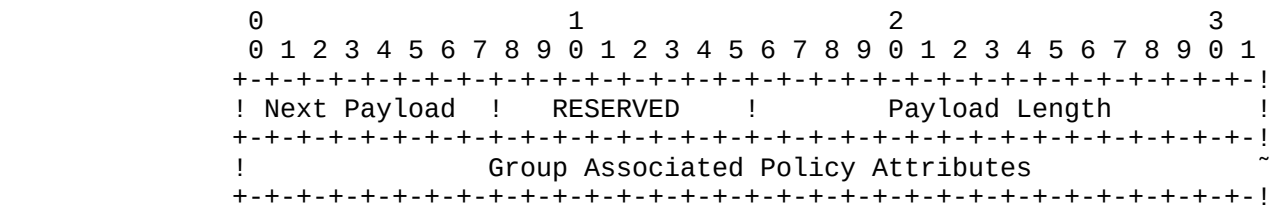
An GSA Life Duration attribute MUST always follow an GSA Life Type which describes the units of duration.

4.6. GSA Group Associated Policy Payload

[RFC3547] provides for the distribution of policy in the G-IKEv2 registration exchange in an SA payload. Policy can define G-IKEv2 rekey policy (GSA KEK) or traffic encryption policy (GSA TEK) such as IPsec policy. There is a need to distribute group policy that fits into neither category. Some of this policy is generic to the group, and some is sender-specific policy for a particular group member. The policy relevant to all group members can be distributed in the G-IKEv2 Registration exchange or the GSA Push exchange, but the sender specific information MUST only be sent in a G-IKEv2 Registration Exchange. Additionally, a group member sometimes has the need to make policy requests for resources of the GCKS in the GSA_AUTH request.

G-IKEv2 distributes this associated group policy in a new payload called the GSA Group Associated Policy (GSA SAP). The GSA GAP payload follows any GSA KEK payload, and is placed before any GSA TEK payloads. In the case that group policy does not include an GSA KEK, the GSA Attribute Next Payload field in the GSA payload MAY indicate the GSA GAP payload.

The GSA GAP payload is defined as follows:



The GSA GAP payload fields are defined as follows:

- Next Payload (1 octet) -- Identifies the next payload present in the G-IKEv2 registration or the G-IKEv2 rekey message. The only valid next payload type for this message is an GSA TEK or zero to indicate there are no more security association attributes.

- RESERVED (1 octet) -- Must be zero.

- Payload Length (2 octets) -- Length of this payload, including the GSA GAP header and Attributes.

- Group Associated Policy Attributes (variable) -- Contains attributes following the format defined in Section 3.3.5 of [RFC5996].

Several group associated policy attributes are defined below. A G-IKEv2 implementation MUST abort if it encounters an attribute or capability that it does not understand.

4.6.1. ACTIVATION_TIME_DELAY/DEACTIVATION_TIME_DELAY

Section 4.2.1 of RFC 5374 specifies a key rollover method that requires two values be given it from the group key management protocol. The ACTIVATION_TIME_DELAY attribute allows a GCKS to set the Activation Time Delay (ATD) for SAs generated from TEKs. The ATD defines how long after receiving new SAs that they are to be activated by the GM. The ATD value is in seconds.

The DEACTIVATION_TIME_DELAY allows the GCKS to set the Deactivation Time Delay (DTD) for previously distributed SAs. The DTD defines how

long after receiving new SAs that it should deactivate SAs that are destroyed by the re-key event. The value is in seconds.

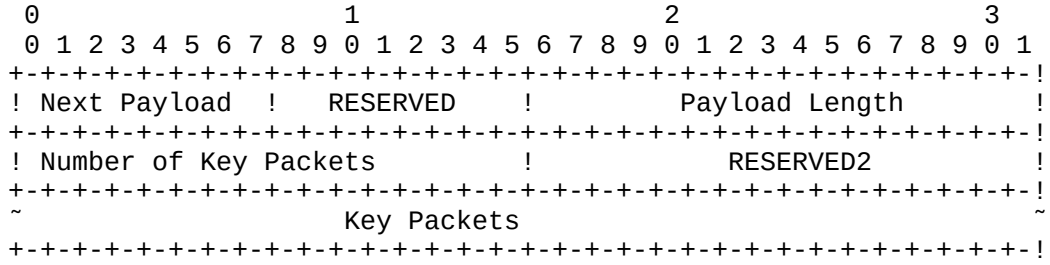
The values of ATD and DTD are independent. However, the DTD value should be larger, which allows new SAs to be activated before older SAs are deactivated. Such a policy ensures that protected group traffic will always flow without interruption.

4.6.2. Sender_ID_REQUEST

The SENDER_ID_REQUEST attribute is used by a group member to request SIDs during the G-IKEv2 Registration message, and includes a count of how many SID values it desires.

4.7. Key Download Payload

The Key Download Payload contains group keys for the group specified in the SA Payload. These key download payloads can have several security attributes applied to them based upon the security policy of the group as defined by the associated SA Payload.



The Key Download Payload fields are defined as follows:

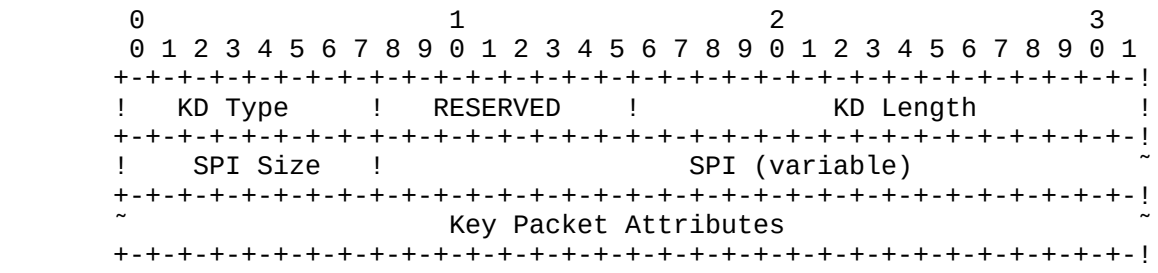
- Next Payload (1 octet) -- Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be zero.

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

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

- Number of Key Packets (2 octets) -- Contains the total number of both TEK and Rekey arrays being passed in this data block.

- Key Packets Several types of key packets are defined. Each Key Packet has the following format.



- Key Download (KD) Type (1 octet) -- Identifier for the Key Data field of this Key Packet.

Key Download Type	Value
RESERVED	0
TEK	1
KEK	2
LKH	3
SID	TBD-7
Standards Action	4-127
Private Use	128-255

"KEK" is a single key whereas LKH is an array of key-encrypting keys.

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

- Key Download Length (2 octets) -- Length in octets of the Key Packet data, including the Key Packet header.

- SPI Size (1 octet) -- Value specifying the length in octets of the SPI as defined by the Protocol-Id.

- SPI (variable length) -- Security Parameter Index which matches a SPI previously sent in an GSAK or GSAT Payload.

- Key Packet Attributes (variable length) -- Contains Key information. The format of this field is specific to the value of the KD Type field. The following sections describe the format of each KD Type.

4.7.1. TEK Download Type

The following attributes may be present in a TEK Download Type. Exactly one attribute matching each type sent in the GSAT payload MUST be present. The attributes must follow the format defined in IKEv2 (Section 3.3.5 of [RFC5996]). In the table, attributes defined as TV are marked as Basic (B); attributes defined as TLV are marked

as Variable (V).

TEK Class	Value	Type
RESERVED	0	
TEK_ALGORITHM_KEY	1	V
TEK_INTEGRITY_KEY	2	V
TEK_SOURCE_AUTH_KEY	3	V

If no TEK key packets are included in a Registration KD payload, the group member can expect to receive the TEK as part of a Re-key SA. At least one TEK must be included in each Re-key KD payload. Multiple TEKs may be included if multiple streams associated with the SA are to be rekeyed.

4.7.1.1. TEK_ALGORITHM_KEY

The TEK_ALGORITHM_KEY class declares that the encryption key for this SPI is contained as the Key Packet Attribute. The encryption algorithm that will use this key was specified in the GSAT payload.

In the case that the algorithm requires multiple keys, all keys will be included in one attribute.

4.7.1.2. TEK_INTEGRITY_KEY

The TEK_INTEGRITY_KEY class declares that the integrity key for this SPI is contained as the Key Packet Attribute. The integrity algorithm that will use this key was specified in the GSAT payload. Thus, G-IKEv2 assumes that both the symmetric encryption and integrity keys are pushed to the member. SHA256 keys will consist of 256 bits.

4.7.1.3. TEK_SOURCE_AUTH_KEY

The TEK_SOURCE_AUTH_KEY class declares that the source authentication key for this SPI is contained in the Key Packet Attribute. The source authentication algorithm that will use this key was specified in the GSAT payload.

4.7.2. KEK Download Type

The following attributes may be present in a KEK Download Type. Exactly one attribute matching each type sent in the GSAK payload MUST be present. The attributes must follow the format defined in IKEv2 (Section 3.3.5 of [RFC5996]). In the table, attributes defined as TV are marked as Basic (B); attributes defined as TLV are marked as Variable (V).

KEK Class	Value	Type
RESERVED	0	
KEK_ALGORITHM_KEY	1	V
AUTH_ALGORITHM_KEY	2	V

If the KEK key packet is included, there MUST be only one present in the KD payload.

4.7.2.1. KEK_ALGORITHM_KEY

The KEK_ALGORITHM_KEY class declares the encryption key for this SPI is contained in the Key Packet Attribute. The encryption algorithm that will use this key was specified in the GSAK payload.

If the mode of operation for the algorithm requires an Initialization Vector (IV), an explicit IV MUST be included in the KEK_ALGORITHM_KEY before the actual key.

4.7.2.2. AUTH_ALGORITHM_KEY

The AUTH_ALGORITHM_KEY class declares that the public key for this SPI is contained in the Key Packet Attribute, which may be useful when no public key infrastructure is available. The signature algorithm that will use this key was specified in the GSAK payload.

4.7.3. LKH Download Type

The LKH key packet is comprised of attributes representing different leaves in the LKH key tree.

The following attributes are used to pass an LKH KEK array in the KD payload. The attributes must follow the format defined in IKEv2 (Section 3.3.5 of [RFC5996]). In the table, attributes defined as TV are marked as Basic (B); attributes defined as TLV are marked as Variable (V).

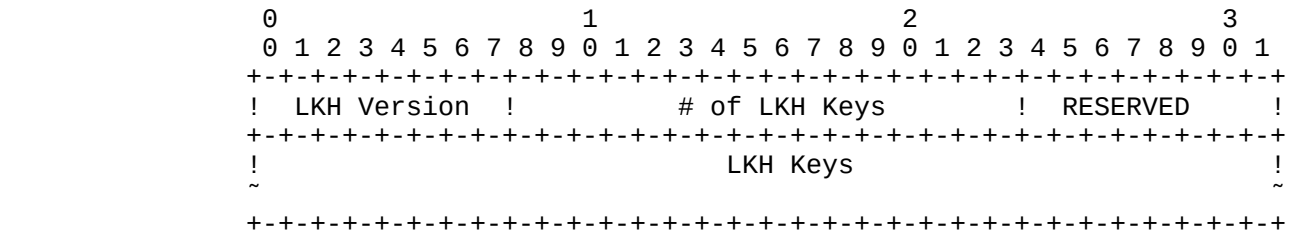
KEK Class	Value	Type
RESERVED	0	
LKH_DOWNLOAD_ARRAY	1	V
LKH_UPDATE_ARRAY	2	V
AUTH_ALGORITHM_KEY	3	V
Standards Action	4-127	
Private Use	128-255	

If an LKH key packet is included in the KD payload, there must be only one present.

4.7.3.1. LKH_DOWNLOAD_ARRAY

This attribute is used to download a set of keys to a group member. It MUST NOT be included in a IKEv2 rekey message KD payload if the IKEv2 rekey is sent to more than the group member. If an LKH_DOWNLOAD_ARRAY attribute is included in a KD payload, there must be only one present.

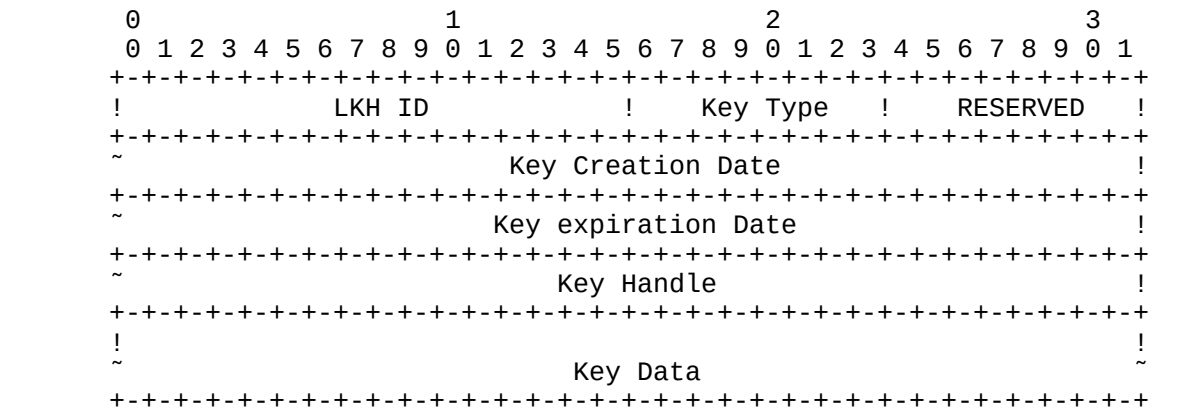
This attribute consists of a header block, followed by one or more LKH keys.



The KEK_LKH attribute fields are defined as follows:

- LKH version (1 octet) -- Contains the version of the LKH protocol which the data is formatted in. Must be one.
- Number of LKH Keys (2 octets) -- This value is the number of distinct LKH keys in this sequence.
- RESERVED (1 octet) -- Unused, set to zero.

Each LKH Key is defined as follows:



- LKH ID (2 octets) -- This is the position of this key in the binary tree structure used by LKH.

- Key Type (1 octet) -- This is the encryption algorithm for which this key data is to be used. This value is specified in Section 4.4.3.

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

- Key Creation Date (4 octets) -- This is the time value of when this key data was originally generated. A time value of zero indicates that there is no time before which this key is not valid.

- Key Expiration Date (4 octets) -- This is the time value of when this key is no longer valid for use. A time value of zero indicates that this key does not have an expiration time.

- Key Handle (4 octets) -- This is the randomly generated value to uniquely identify a key within an LKH ID.

- Key Data (variable length) -- This is the actual encryption key data, which is dependent on the Key Type algorithm for its format. If the mode of operation for the algorithm requires an Initialization Vector (IV), an explicit IV MUST be included in the Key Data field before the actual key.

The Key Creation Date and Key expiration Dates MAY be zero. This is necessary in the case where time synchronization within the group is not possible.

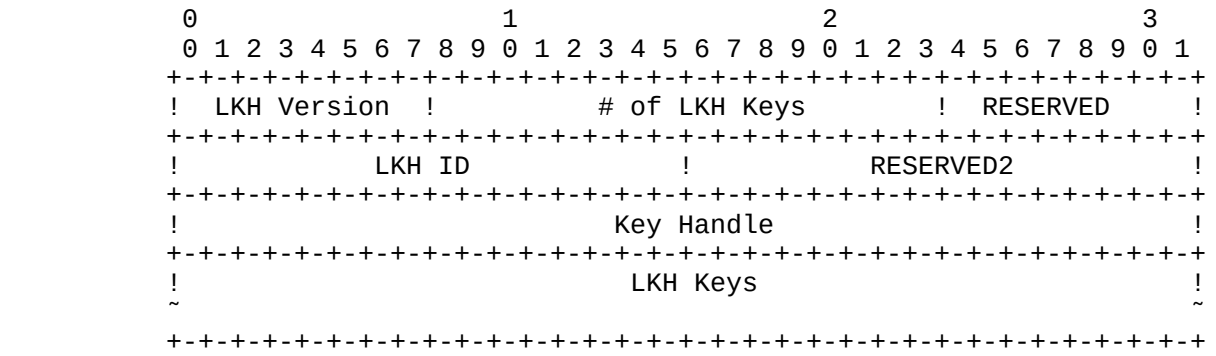
The first LKH Key structure in an LKH_DOWNLOAD_ARRAY attribute contains the Leaf identifier and key for the group member. The rest

of the LKH Key structures contain keys along the path of the key tree in order from the leaf, culminating in the group KEK.

4.7.3.2. LKH_UPDATE_ARRAY

This attribute is used to update the keys for a group. It is most likely to be included in a G-IKEv2 rekey message KD payload to rekey the entire group. This attribute consists of a header block, followed by one or more LKH keys, as defined in Section 4.7.3.1.

There may be any number of UPDATE_ARRAY attributes included in a KD payload.



- LKH version (1 octet) -- Contains the version of the LKH protocol which the data is formatted in. Must be one.

- Number of LKH Keys (2 octets) -- This value is the number of distinct LKH keys in this sequence.

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

- LKH ID (2 octets) -- This is the node identifier associated with the key used to encrypt the first LKH Key.

- RESERVED2 (2 octets) -- Unused, set to zero.

- Key Handle (4 octets) -- This is the value to uniquely identify the key within the LKH ID which was used to encrypt the first LKH key.

The LKH Keys are as defined in Section 4.7.3.1. The LKH Key structures contain keys along the path of the key tree in order from the LKH ID found in the LKH_UPDATE_ARRAY header, culminating in the group KEK. The Key Data field of each LKH Key is encrypted with the LKH key preceding it in the LKH_UPDATE_ARRAY attribute. The first

LKH Key is encrypted under the key defined by the LKH ID and Key Handle found in the LKH_UPDATE_ARRAY header.

4.7.3.3. AUTH_ALGORITHM_KEY

The AUTH_ALGORITHM_KEY class declares that the public key for this SPI is contained in the Key Packet Attribute, which may be useful when no public key infrastructure is available. The signature algorithm that will use this key was specified in the GSAK payload.

4.7.4. SID Download Type

This attribute is used to download one or use more Sender-ID (SID) values for the exclusive use of a group member.

KEK Class	Value	Type
RESERVED	0	
NUMBER_OF_SID_BITS	1	V
SID_VALUE	2	V
Standards Action	3-128	
Private Use	129-255	
Unassigned	256-32767	

Because a SID value is intended for a single group member, the SID Download type MUST NOT be distributed in a GROUPKEY_PUSH message distributed to multiple group members.

4.7.4.1. NUMBER_OF_SID_BITS

The NUMBER_OF_SID_BITS class declares how many bits of the cipher nonce in which to represent an SID value. This value applied to each SID value is distributed in the SID Download.

4.7.4.2. SID_VALUE

The SID_VALUE class declares a single SID value for the exclusive use of the a group member. Multiple SID_VALUE attributes MAY be included in a SID Download.

4.7.4.3. GM Semantics

The SID_VALUE attribute value distributed to the group member MUST be used by that group member as the SID field portion of the IV for all Data-Security SAs including a counter-based mode of operation distributed by the GCKS as a part of this group. When the Sender-Specific IV (SSIV) field for any Data-Security SA is exhausted, the group member MUST no longer act as a sender on that SA using its

active SID. The group member SHOULD re-register, at which time the GCKS will issue a new SID to the group member, along with either the same Data-Security SAs or replacement ones. The new SID replaces the existing SID used by this group member, and also resets the SSIV value to its starting value. A group member MAY re-register prior to the actual exhaustion of the SSIV field to avoid dropping data packets due to the exhaustion of available SSIV values combined with a particular SID value.

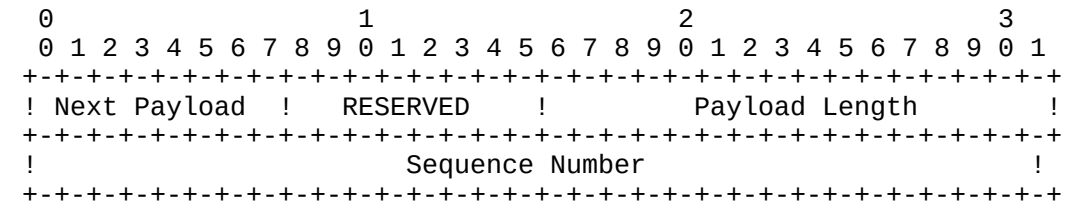
A group member MUST NOT process an SID Download Type KD payload present in a GSA-PUSH message.

4.7.4.4. GCKS Semantics

If any KD payload includes keying material that is associated with a counter-mode of operation, an SID Download Type KD payload containing at least one SID_VALUE attribute MUST be included. The GCKS MUST NOT send the SID Download Type KD payload as part of a GSA-PUSH message, because distributing the same sender-specific policy to more than one group member will reduce the security of the group.

4.8. Sequence Number Payload

The Sequence Number Payload (SEQ) provides an anti-replay protection for GSA rekey messages.



- 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 zero.

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

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

- Sequence Number (4 octets) -- The sequence number of the rekey message.

4.9. Delete Payload

There are occasions the GCKS may want to signal to receivers to delete policy at the end of a broadcast, or if policy has changed. Deletion of keys MAY be accomplished by sending an IKEv2 Delete Payload, section 3.11 of [RFC5996] as part of the G-IKEv2 Rekey Exchange.

One or more Delete payloads MAY be placed following the HDR payload in the G-IKEv2 Rekey Exchange.

The Protocol-ID field contains TEK protocol id values. In order to delete a KEK SA, the value of zero MUST be used as the protocol id. Note that only one protocol id value can be defined in a Delete payload. If a TEK and a KEK SA must be deleted, they must be sent in different Delete payloads.

4.10. Notify Payload

G-IKEv2 uses the same notify payload as specified in [RFC5996], section 3.10.

There are additional notify message types introduced by G-IKEv2 to communicate error conditions and status.

NOTIFY MESSAGES - ERROR TYPES	Value
INVALID_GROUP_ID - Indicates the group id sent during registration process is invalid.	TBD
AUTHORIZATION_FAILED - Sent in the response to IKE_AUTH message when authorization failed.	TBD

NOTIFY MESSAGES - STATUS TYPES	Value
GIKEv2_REGISTRATION - See Section 3.1.3	TBD
GIKEv2_REKEY - See Section 3.2.1	TBD

4.11. Authentication Payload

G-IKEv2 uses the same Authentication payload as specified in [RFC5996], section 3.8, to sign the rekey message.

5. Security Considerations

5.1. GSA registration and secure channel

G-IKEv2 registration exchange uses IKEv2 IKE_SA_INIT and IKE_AUTH inheriting all the security considerations documented in [RFC5996] section 5 Security Considerations, including authentication, confidentiality, protection against man-in-the middle, protection against replay/reflection attacks, and denial of service protection. In addition, G-IKEv2 brings in the capability to authorize a particular group member regardless of whether they have the IKEv2 credentials.

5.2. GSA maintenance channel

The GSA maintenance channel is cryptographically and integrity protected using the cryptographic algorithm and key negotiated in the GSA member registration exchanged.

5.2.1. Authentication/Authorization

Authentication is implicit, the public key of the identity is distributed during the registration, and the receiver of the rekey message uses that public key and identity to verify the message is come from the authorized GCKS.

5.2.2. Confidentiality

Confidentiality is provided by distributing a confidentiality key as part of the GSA member registration exchange.

5.2.3. Man-in-the-Middle Attack Protection

GSA maintenance channel is integrity protected by using digital signature.

5.2.4. Replay/Reflection Attack Protection

The GSA rekey message includes a monotonically increasing sequence number to protect against replay and reflection attacks. A group member will recognize a replayed message by comparing the sequence number to that of the last received rekey message, any rekey message contains sequence number less than or equal to the last received value SHOULD be discarded. Implementations SHOULD keep a record of recently received GSA rekey messages for this comparison.

6. IANA Considerations

6.1. New registries

A new set of registries are created for this draft.

KEK Attributes Registry, see Section 4.4.1

KEK Management Algorithm Registry, see Section 4.4.2

KEK Algorithm Registry, see Section 4.4.3

AUTH Hash Algorithm Registry, see Section 4.4.6

GSA TEK Payload Protocol ID Type Registry, see Section 4.5

GSA Life Attributes Registry, see Section 4.5

Key Download Type Registry, see Section 4.7

TEK Download Type Registry, see Section 4.7.1

KEK Download Type Registry, see Section 4.7.2

LKH Download Type Registry, see Section 4.7.3

6.2. New payload and exchange types to existing IKEv2 registry

The present document describes new IKEv2 Next Payload types, see Section 4.1

The present document describes new IKEv2 Exchanges types, see Section 4.1

The present document describes new IKEv2 Notify Payload types, see Section 4.10

6.3. Payload Types

The present document defines new ISAKMP Next Payload types. See Section 5.0 for the payloads defined in this document, including the Next Payload values defined by the IANA to identify these payloads.

6.4. New Name spaces

The present document describes many new name spaces for use in the GDOI payloads. Those may be found in subsections under Section 5.0. A new GDOI registry has been created for these name spaces.

Portions of name spaces marked "RESERVED" are reserved for IANA allocation. New values MUST be added due to a Standards Action as defined in [RFC2434].

Portions of name spaces marked "Private Use" may be allocated by implementations for their own purposes.

7. Acknowledgements

The authors thank Lakshminath Dondeti and Jing Xiang for originating the GKDP document and providing the basis behind the protocol.

The authors also thank reviewers: Brian Weis, Kavitha Kamarthy, Lewis Chen, Cheryl Madson, and Raghunandan P.

8. References

8.1. Normative References

[FIPS186-2] "Digital Signature Standard (DSS)", United States of America, National Institute of Science and Technology Federal Information Processing Standard (FIPS) 186-2, January 2001.

[FIPS197] "Advanced Encryption Standard (AES)", United States of America, National Institute of Science and Technology Federal Information Processing Standard (FIPS) 197, November 2001.

[RSA] TRSA Laboratories, "PKCS #1 v2.0: RSA Encryption Standard", 1998.

[SP800-38A] Dworkin, M., "Recommendation for Block Cipher Modes of Operation", United States of America, National Institute of Science and Technology NIST Special Publication 800-38A 2001 Edition, December 2001.

[SP800-38D] Dworkin, M., "Recommendation for Block Cipher Modes of Operation", United States of America, National Institute of Science and Technology NIST Special Publication 800-38D 2007 Edition, December 2001.

8.2. Informative References

[IKE-HASH] Kivienen, T., "Fixing IKE Phase 1 & 2 Authentication HASHs", November 2001, <http://tools.ietf.org/html/draft-ietf-ipsec-ike-hash-revised-03>.

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2408] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an

IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

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

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

[RFC3686] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.

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

[RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", RFC 4106, June 2005.

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

[RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)", RFC 4309, December 2005.

[RFC4430] Sakane, S., Kamada, K., Thomas, M., and J. Vilhuber, "Kerberized Internet Negotiation of Keys (KINK)", RFC 4430, March 2006.

[RFC4543] McGrew, D. and J. Viega, "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH", RFC 4543, May 2006.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

Appendix A. Differences between G-IKEv2 and RFC 3547

POP/CERT - The Proof of Possession and associated Certificate payloads are no longer needed since the GCKS authorization capability adequately provides the authorization.

KE Payload - The KE payload is no longer needed with the availability of newer algorithms such as AES and GCM which provide adequate protection therefore not needing the PFS capability the KE payload offers.

SIG Payload - The AUTH payload is used for the same purpose instead.

DOI/Situation - The DOI and Situation fields in the SA payload are no longer needed in the G-IKEv2 protocol as port 848 will distinguish the IKEv2 messages from the G-IKEv2 messages.

Authors' Addresses

Sheela Rowles
Cisco Systems
170 W. Tasman Drive
San Jose, California 95134-1706
USA

Phone: +1-408-527-7677
Email: sheela@cisco.com

Aldous Yeung (editor)
Cisco Systems
170 W. Tasman Drive
San Jose, California 95134-1706
USA

Phone: +1-408-853-2032
Email: cyyeung@cisco.com

Paulina Tran
Cisco Systems
170 W. Tasman Drive
San Jose, California 95134-1706
USA

Phone: +1-408-526-8902
Email: ptran@cisco.com