Group Key Management using IKEv2
draft-yeung-g-ikev2-14

Abstract

   This document presents a set of IKEv2 exchanges that comprise a group
   key management protocol. The protocol is in conformance with the
   Multicast Security (MSEC) key management architecture, which contains
   two components: member registration and group rekeying. Both
   components require a Group Controller/Key Server to download IPsec
   group security associations to authorized members of a group. The
   group members then exchange IP multicast or other group traffic as
   IPsec packets. This document obsoletes RFC 6407.

Table of Contents

1. Introduction and Overview

   A group key management protocol provides IPsec keys and policy to a
   set of IPsec devices which are authorized to communicate using a
   Group Security Association (GSA) defined in [RFC3740]. The data
   communications within the group (e.g., IP multicast packets) are
   protected by a key pushed to the group members (GMs) by the Group
   Controller/Key Server (GCKS). This document presents a set of IKEv2
   [RFC7296] exchanges that comprise a group key management protocol.

   A GM begins a "registration" exchange when it first joins the group.
   With G-IKEv2, the GCKS authenticates and authorizes GMs, then pushes
   policy and keys used by the group to the GM. G-IKEv2 includes two
   "registration" exchanges. The first is the GSA_AUTH exchange (
   Section 3.1.1), which follows an IKE_SA_INIT exchange. The second is
   the GSA_REGISTRATION exchange (Section 3.1.2), which a GM can use
   within an established IKE SA. Group rekeys are accomplished using
   either the GSA_REKEY exchange (a single message distributed to all
   GMs, usually as a multicast message), or as a GSA_INBAND_REKEY
   exchange delivered individually to group members using existing IKE
   SAs).

   Large and small groups may used different sets of these protocols.
   When a large group of devices are communicating, the GCKS is likely
   to use the GSA_REKEY message for efficiency. This is shown in
   Figure 1. (Note: For clarity, IKE_SA_INIT is omitted from the
   figure.)

```
                          +--------+
           +------------->|  GCKS  |<-------------+
           |              +--------+              |
           |                  |   ^               |
           |                  |   |               |
           |                  | GSA_AUTH          |
           |                  |   or              |
           |                  | GSA_REGISTRATION  |
           |                  |   |               |
       GSA_AUTH              |   |           GSA_AUTH
          or          GSA_REKEY |              or
    GSA_REGISTRATION           |   |        GSA_REGISTRATION
           |                  |   |               |
           |   +-----------+-----------------+    |
           |   |           |   |           |  |   |
           v   v           v   v           v  v
       +-------+       +--------+       +-------+
       |  GM   |  ...  |  GM    |  ...  |  GM   |
       +-------+       +--------+       +-------+
          ^               ^               ^
          |               |               |
          +-------ESP-------+-------ESP------+
```
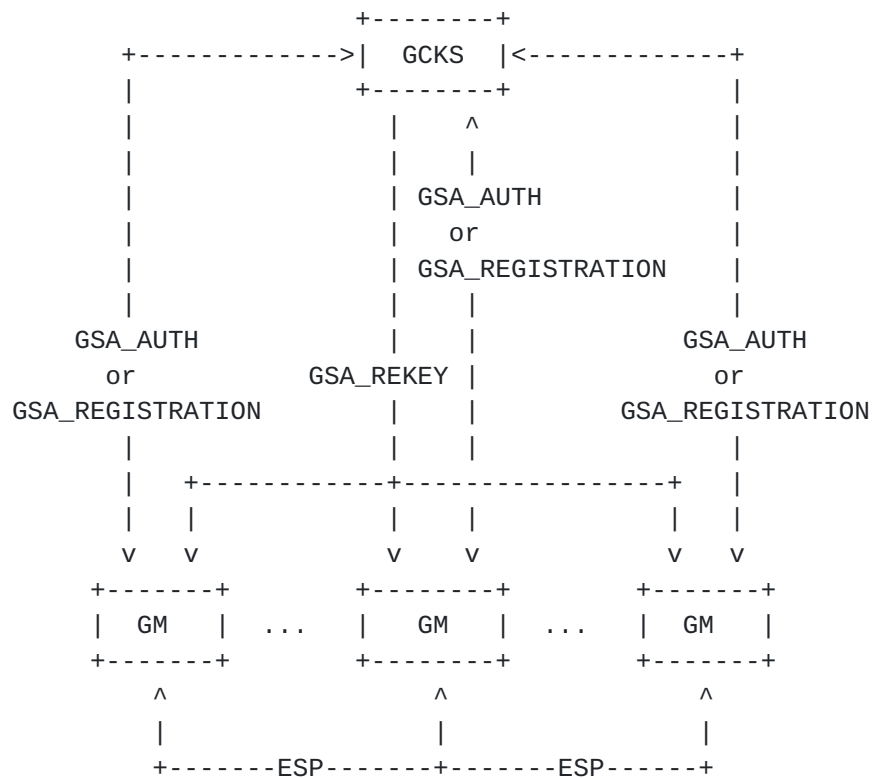
                  Figure 1: G-IKEv2 used in large groups

   Alternatively, a small group may simply use the GSA_AUTH as a
   registration protocol, where he GCKS issues rekeys using the
   GSA_INBAND_REKEY within the same IKEv2 SA. The GCKS is also likely
   to be a GM in a small group (as shown in Figure 2.)

```
                    GSA_AUTH, GSA_INBAND_REKEY
         +-------------------------------------------------+
         |                                                 |
         |          GSA_AUTH, GSA_INBAND_REKEY        |
         |   +----------------------------+             |
         |   |                             |             |
         |   | GSA_AUTH, GSA_INBAND_REKEY  |             |
         |   |    +--------+               |             |
         v   v    v        v               v             v
       +---------+    +----+          +----+        +----+
       | GCKS/GM |    | GM |          | GM |        | GM |
       +---------+    +----+          +----+        +----+
           ^            ^               ^             ^
           |            |               |             |
           +----ESP-----+------ESP-------+-----ESP-----+
```
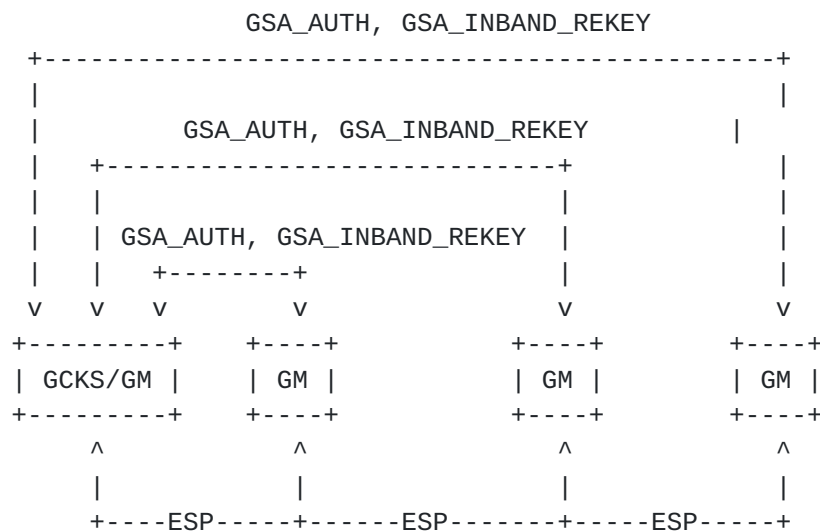
Figure 2: G-IKEv2 used in small groups

IKEv2 message semantics are preserved in that all communications
consists of message request-response pairs. The exception to this
rule is the GSA_REKEY exchange, which is a single message delivering
group updates to the GMs.

G-IKEv2 conforms with the The Multicast Group (MEC) Security
Architecture [RFC3740], and the Multicast Security (MSEC) Group Key
Management Architecture [RFC4046]. G-IKEv2 replaces GDOI [RFC6407],
which defines a similar group key management protocol using IKEv1
[RFC2409] (since deprecated by IKEv2). When G-IKEv2 is used, group
key management use cases can benefit from the simplicity, increased
robustness and cryptographic improvements of IKEv2 (see Appendix A of
   [RFC7296].

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in BCP
14 [RFC2119] [RFC8174] when, and only when, they appear in all
capitals, as shown here.

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

2. G-IKEv2 integration into IKEv2 protocol

   G-IKEv2 uses the security mechanisms of IKEv2 (peer authentication,
   confidentiality, message integrity) to ensure that only authenticated
   devices have access to the group policy and keys. The G-IKEv2
   exchange further provides group authorization, and secure policy and
   key download from the GCKS to GMs.

   It is assumed that readers are familiar with the IKEv2 protocol, so
   this document skips many details that are described in [RFC7296].

2.1. UDP port

   G-IKEv2 SHOULD use port 848, the same as GDOI [RFC6407], because they
   serve a similar function. They can use the same ports, just as IKEv1
   and IKEv2 can share port 500. The version number in the IKEv2 header
   distinguishes the G-IKEv2 protocol from GDOI protocol [RFC6407].
   G-IKEv2 MAY also use port the IKEv2 ports (500, 4500), which would
   provide a better integration with IKEv2.

3. G-IKEv2 Protocol

3.1. G-IKEv2 member registration and secure channel establishment

   The registration protocol consists of a minimum of two message
   exchanges, IKE_SA_INIT and GSA_AUTH; member registration may have a
   few more messages exchanged if the EAP method, cookie challenge (for
   DoS protection) or negotiation of Diffie-Hellman group is included.
   Each exchange consists of request/response pairs. The first exchange
   IKE_SA_INIT is defined in IKEv2 [RFC7296]. This exchange negotiates
   cryptographic algorithms, exchanges nonces and does a Diffie-Hellman
   exchange between the group member (GM) and the Group Controller/Key
   Server (GCKS).

   The second exchange GSA_AUTH authenticates the previous messages,
   exchanges identities and certificates. These messages are encrypted
   and integrity protected with keys established through the IKE_SA_INIT
   exchange, so the identities are hidden from eavesdroppers and all
   fields in all the messages are authenticated. The GCKS SHOULD
   authorize group members to be allowed into the group as part of the
   GSA_AUTH exchange. Once the GCKS accepts a group member to join a
   group it will download the data security keys (TEKs) and/or group key
   encrypting key (KEK) or KEK array as part of the GSA_AUTH response
   message. In the following descriptions, the payloads contained in
   the message are indicated by names as listed below. Payloads defined

as part of other IKEv2 extensions MAY also be included in these
exchanges.

```
    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
    SAg           Security Association - GM Supported Transforms
```

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], to indicate that a certificate request
payload can optionally be included.

3.1.1. GSA_AUTH exchange

After the group member and GCKS use the IKE_SA_INIT exchange to
negotiate cryptographic algorithms, exchange nonces, and perform a
Diffie-Hellman exchange as defined in IKEv2 [RFC7296], the GSA_AUTH
exchange MUST complete before any other exchanges can be done. The
security properties of the GSA_AUTH exchange are the same as the
properties of the IKE_AUTH exchange. It is used to authenticate the
IKE_SA_INIT messages, exchange identities and certificates. G-IKEv2
also uses this exchange for group member registration and
authorization. Even though the IKE_AUTH does contain the SA2, TSi,
and TSr payload the GSA_AUTH does not. They are not needed because
policy is not negotiated between the group member and the GCKS, but
instead downloaded from the GCKS to the group member.

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

After the IKE_SA_INIT exchange completes, the group member initiates
a GSA_AUTH request to join a group indicated by the IDg payload. The
GM MAY include an SAg payload declaring which Transforms that it is
willing to accept, and also MAY include the Notify payload status

type SENDER_ID_REQUEST to request SIDs for a Counter-based cipher
from the GCKS.

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

The GCKS responds with IDr, optional CERT, and AUTH material as if it
were an IKE_AUTH. It also informs the group member of the
cryptographic policies of the group in the GSA payload and the key
material in the KD payload. The GCKS can also include a Delete (D)
payload instructing the group member to delete existing SAs it might
have as the result of a previous group member registration.

In addition to the IKEv2 error handling, the GCKS can reject the
registration request when the IDg is invalid or authorization fails,
etc. In these cases, see Section 4.10, the GSA_AUTH response will
not include the GSA and KD, but will include a Notify payload
indicating errors. If the group member included an SAg payload, and
the GCKS chooses to evaluate it, and it detects that that group
member cannot support the security policy defined for the group, then
the GCKS SHOULD return a NO_PROPOSAL_CHOSEN. When the GCKS indicates
errors, and the group member cannot resolve the errors, the group
member MUST delete the registration IKE SA.

    Initiator (Member)   Responder (GCKS)
   --------------------   ------------------
                              <--   HDR, SK { N }

If the group member finds the policy sent by the GCKS is
unacceptable, the member SHOULD notify the GCKS by sending IDg and
the Notify type NO_PROPOSAL_CHOSEN as shown below.

    Initiator (Member)   Responder (GCKS)
   --------------------   ------------------
    HDR, SK {IDg [N,]}   -->

                              <--   HDR, SK {}

3.1.2. GSA_REGISTRATION Exchange

When a secure channel is already established between a GM and the
GCKS, the GM registration for a group can reuse the established
secure channel. In this scenario the GM will use the
GSA_REGISTRATION exchange by including the desired group ID (IDg) to
request data security keys (TEKs) and/or group key encrypting keys
(KEKs) from the GCKS. If the group member includes an SAg payload,
and the GCKS chooses to evaluate it, and it detects that group member
cannot support the security policy defined for the group, then the
GCKS SHOULD return a NO_PROPOSAL_CHOSEN. The GM MAY also include the

Notify payload status type SENDER_ID_REQUEST to request SIDs for a
Counter-based cipher from the GCKS. The GCKS response payloads are
created and processed as in the GSA_AUTH reply.

```
     Initiator (Member)                    Responder (GCKS)
    --------------------                  ------------------
    HDR, SK {IDg, [SAg, ][N ] } -->

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

This exchange can also be used if the group member finds the policy
sent by the GCKS is unacceptable. The group member SHOULD notify the
GCKS by sending IDg and the Notify type NO_PROPOSAL_CHOSEN, as shown
below. The GCKS MUST unregister the group member.

```
     Initiator (Member)                    Responder (GCKS)
    --------------------                  ------------------
    HDR, SK {IDg [N,]}      -->

                            <--         HDR, SK {}
```

3.1.3. IKEv2 Header Initialization

The Major Version is (2) and Minor Version is (0) according to IKEv2
[RFC7296], and maintained in this document. The G-IKEv2 IKE_SA_INIT,
GSA_AUTH and GSA_REGISTRATION use the IKE SPI according to IKEv2
[RFC7296], section 2.6.

3.1.4. GM Registration Operations

A G-IKEv2 Initiator (GM) requesting registration contacts the GCKS
using the IKE_SA_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 IKE_SA_INIT response,
the GM sends the GSA_AUTH message with the IKEv2 payloads from
IKE_AUTH (without the SAi2, TSi and TSr payloads) along with the
Group ID informing the GCKS of the group the initiator wishes to
join. The initiator MAY specify how many Sender-ID values it would
like to receive in the Notify payload status type, SENDER_ID_REQUEST,
in case the Data Security SA supports a counter mode cipher (see
Section 3.3).

An initiator may be limited in the types of Transforms that it is
able or willing to use, and may find it useful to inform the GCKS
which Transforms that it is willing to accept. It can OPTIONALLY
include an SAg payload, which can include ESP and/or AH Proposals.

Each Proposal contains a list of Transforms that it is willing to
support for that protocol. A Proposal of type ESP can include ENCR,
INTEG, and ESN Transforms. A Proposal of type AH can include INTEG,
and ESN Transforms. The SPI length of each Proposal in an SAg MUST
be zero, and the SPI field is null. Generally, a single Proposal of
each type will suffice, because the group member is not negotiating
Transform sets, simply alerting the GCKS to restrictions it may have.

Upon receiving the GSA_AUTH response, the initiator parses the
response from the GCKS authenticating the exchange using the IKEv2
method, then processes the GSA and KD.

The GSA payload contains the security policy and cryptographic
protocols used by the group. This policy describes the Rekey SA
(KEK), if present, Data-security SAs (TEK), and other group policy
(GAP). If the policy in the GSA payload is not acceptable to the GM,
it SHOULD notify the GCKS with a NO_PROPOSAL_CHOSEN Notify payload
(see Section 3.1.1 and Section 3.1.2). 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 included 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.

The GSA KEK policy MUST include KEK attribute KEK_MESSAGE_ID with a
Message ID. The Message ID in the KEK_MESSAGE_ID attribute MUST be
checked against any previously received Message ID 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 Message ID changed. This
KEK_MESSAGE_ID is used by the GM to prevent GSA_REKEY message replay
attacks. The first GSA_REKEY message that the GM receives from the
GCKS must have a Message ID greater or equal to the Message ID
received in the KEK_MESSAGE_ID attribute.

3.1.5. GCKS Registration Operations

A G-IKEv2 GCKS passively listens for incoming requests from group
members. When the GCKS receives an IKE_SA_INIT request, it selects
an IKE proposal and generates a nonce and DH to include them in the
IKE_SA_INIT response.

Upon receiving the GSA_AUTH request, the GCKS authenticates the group
member using the same procedures as in the IKEv2 IKE_AUTH. The GCKS
then authorizes the group member according to group policy before
preparing to send the GSA_AUTH response. If the GCKS fails to

authorize the GM, it will respond with an AUTHORIZATION_FAILED notify
message.

The GSA_AUTH response will include the group policy in the GSA
payload and keys in the KD payload. If the GCKS policy includes a
group rekey option, this policy is constructed in the GSA KEK and the
key is constructed in the KD KEK. The GSA KEK MUST include the
KEK_MESSAGE_ID attribute, specifying the starting Message ID the GCKS
will use when sending the GSA_REKEY message to the group member.
This Message ID is used to prevent GSA_REKEY message replay attacks
and will be increased each time a GSA_REKEY message is sent to the
group. The GCKS data traffic policy is included in the GSA TEK and
keys are included in the KD TEK. The GSA GAP MAY also be included to
provide the ATD and/or DTD (Section 4.7.1) specifying activation and
deactivation delays for SAs generated from the TEKs. 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
Notify payload status type SENDER_ID_REQUEST requesting the number of
SIDs from the group member.

If the GCKS receives a GSA_REGISTRATION exchange with a request to
register a GM to a group, the GCKS will need to authorize the GM with
the new group (IDg) and respond with the corresponding group policy
and keys. If the GCKS fails to authorize the GM, it will respond
with the AUTHORIZATION_FAILED notification.

If a group member includes an SAg in its GSA_AUTH or GSA_REGISTRATION
request, the GCKS MAY evaluate it according to an implementation
specific policy.

o The GCKS could evaluate the list of Transforms and compare it to
   its current policy for the group. If the group member did not
   include all of the ESP or AH Transforms in its current policy,
   then it could return a NO_PROPOSAL_CHOSEN Notification.

o The GCKS could store the list of Transforms, with the goal of
   migrating the group policy to a different Transform when all of
   the group members indicate that they can support that Transform.

o The GCKS could store the list of Transforms and adjust the current
   group policy based on the capabilities of the devices as long as
   they fall within the acceptable security policy of the GCKS.

[3.1.6](). Interaction with IKEv2 protocols

[3.1.6.1](). Session Resumption

G-IKEv2 is compatible with and can use IKEv2 Session Resumption [[RFC5723]()] except that a GM would include the initial ticket request in a GSA_AUTH exchange instead of an IKE_AUTH exchange.

[3.1.6.2](). Postquantum Preshared Keys for IKEv2

G-IKEv2 can take advantage of the protection provided by Postquantum Preshared Keys (PPK) for IKEv2 [[I-D.ietf-ipsecme-qr-ikev2]()]. However, the current PPK draft leaves the initial IKE SA susceptible to quantum computer (QC) attacks. It suggests that for applications using IKEv2 to be QC-secure, an immediate IKE SA rekey should take place followed by a GSA_REGISTRATION exchange.

[3.2](). Group Maintenance Channel

The GCKS is responsible for rekeying the secure group per the group policy. Rekeying is an operation whereby the GCKS provides replacement TEKs and KEK, deleting TEKs, and/or excluding group members. The GCKS may initiate a rekey message if group membership and/or policy has changed, or if the keys are about to expire. Two forms of group maintenance channels are provided in G-IKEv2 to push new policy to group members.

GSA_REKEY The GSA_REKEY exchange is an exchange initiated by the GCKS, where the rekey policy is usually delivered to group members using IP multicast as a transport. This is valuable for large and dynamic groups, and where policy may change frequently and an scalable rekeying method is required. When the GSA_REKEY exchange is used, the IKEv2 SA protecting the member registration exchanges is terminated, and group members await policy changes from the GCKS via the GSA_REKEY exchange.

GSA_INBAND_REKEY The GSA_INBAND_REKEY exchange is a rekey method using the IKEv2 SA that was setup to protecting the member registration exchange. This exchange allows the GCKS to rekey without using an independent GSA_REKEY exchange. The GSA_INBAND_REKEY exchange is useful when G-IKEv2 is used with a small group of cooperating devices.

[3.2.1](). GSA_REKEY exchange

The GCKS initiates the G-IKEv2 Rekey securely, usually using IP multicast. Since this rekey does not require a response and it sends to multiple GMs, G-IKEv2 rekeying MUST NOT support IKE SA windowing.

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
in the Key Download payload (defined in Section 4.8.4) MUST 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 { GSA, KD, [D,] AUTH }
```

HDR is defined in Section 4.1. The Message ID in this message will
start with the same value the GCKS sent to the group members in the
KEK attribute KEK_MESSAGE_ID during registration; this Message ID
will be increased each time a new GSA_REKEY message is sent to the
group members.

The GSA payload contains the current rekey and data security SAs.
The GSA may contain a new rekey SA and/or a new data security SA,
which, optionally contains an LKH rekey SA, Section 4.4.

The KD payload contains the keys for the policy included 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.

A Delete payload MAY be included to instruct the GM to delete
existing SAs.

The AUTH payload is included to authenticate the GSA_REKEY message
using a method defined in the IKEv2 Authentication Method IANA
registry [IKEV2-IANA]. The method SHOULD be a digital signature
authentication scheme to ensure that the message was originated from
an authorized GCKS. A Shared Key Integrity Code SHOULD NOT be used
as it doesn't provide source origin authentication (although a small
group may not require source origin authentication). During group
member registration, the GCKS sends the authentication key in the GSA
KEK payload, KEK_AUTH_KEY attribute, which the group member uses to
authenticate the key server. Before the current Authentication Key
expires, the GCKS will send a new KEK_AUTH_KEY to the group members
in a GSA_REKEY message. The AUTH key that is used in the rekey
message may not be the same as the authentication key used in
GSA_AUTH. Typically a rekey message is sent as multicast and
received by all group members, therefore the same AUTH key is
distributed to all group members.

   After adding the AUTH payload to the rekey message, the current KEK
   encryption key is used to encrypt all of the payloads following the
   HDR.

3.2.1.1. GSA_REKEY GCKS Operations

   The GCKS builds the rekey message with a Message ID value that is one
   greater than the value included in the previous rekey. If the
   message is using a new KEK attribute, the Message ID is reset to 1 in
   this message. The GSA, KD, and D payloads follow with the same
   characteristics as in the GSA Registration exchange. The AUTH
   payload is the final payload added to the message. It is created by
   hashing the string "G-IKEv2" and the message created so far, and then
   is digitally signed. Finally, the content of the Encrypted payload
   is encrypted and authenticated using the current KEK keys.

   Because GSA_REKEY messages are not acknowledged and could be
   discarded by the network, one or more GMs may not receive the
   message. To mitigate such lost messages, during a rekey event the
   GCKS SHOULD transmit several GSA_REKEY messages with the new policy.
   When re-transmitting a GSA_REKEY a GCKS needs to ensure that TEK and
   KEK time-to-live lifetimes are still the correct values. If the
   lifetimes in a re-transmitted message are stale, they will represent
   an artificially lengthened lifetime, possibly resulting in GMs with
   unsynchronized TEK and KEK lifetimes.

3.2.1.2. GSA_REKEY GM Operations

   When a group member receives the Rekey Message from the GCKS it
   decrypts the message using the current KEK, validates the signature
   using the public key retrieved in a previous G-IKEv2 exchange,
   verifies the Message ID, 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 parsing done in
   the registration exchange.

   Replay protection is achieved by a group member rejecting a GSA_REKEY
   message which has a Message ID smaller than the current Message ID
   that the GM is expecting. The GM expects the Message ID in the first
   GSA_REKEY message it receives to be equal or greater than the message
   id it receives in the KEK_MESSAGE_ID attribute. The GM expects the
   message ID in subsequent GSA_REKEY messages to be greater than the
   last valid GSA_REKEY message ID it received.

   If the GSA payload includes a 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_REGISTRATION exchange to the
GCKS in order to obtain an SID value (along with current group
policy).

If the GM receives a notification that a Data-Security SA is about to
expire (such as a "soft lifetime" expiration as described in
[Section 4.4.2.1 of [RFC4301]](#)), it SHOULD initiate a registration to
the GCKS. This registration serves as a request for current SAs, and
will result in the download of replacement SAs, assuming the GCKS
policy has created them.

[3.2.1.3](#). Forward and Backward Access Control

Through the G-IKEv2 rekey, 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 are supported via the
KEY_MANAGEMENT_ALGORITHM attribute which is sent in the GSA KEK
policy. 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.

[3.2.1.3.1](#). Forward Access Control Requirements

When group membership is altered using a group management algorithm
new GSA TEKs (and their associated keys) are usually also needed.
New GSAs 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 GSA
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 GSA 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 GSA TEKS and their

associated key packets 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 the forward access control policy is maintained within a
single G-IKEv2 rekey message.

3.2.1.4. Fragmentation

IKE fragmentation [RFC7383] can be used to perform fragmentation of
large GSA_REKEY messages, however when the GSA_REKEY message is
emitted as an IP multicast packet there is a lack of response from
the GMs. This has the following implications.

o Policy regarding the use of IKE fragmentation is implicit. If a
   GCKS detects that all GMs have negotiated support of IKE
   fragmentation in IKE_SA_INIT, then it MAY use IKE fragmentation on
   large GSA_REKEY exchange messages.

o The GCKS must always use IKE fragmentation based on a known
   fragmentation threshold (unspecified in this memo), as there is no
   way to check if fragmentation is needed by first sending
   unfragmented messages and waiting for response.

o PMTU probing cannot be performed due to lack of GSA_REKEY response
   message.

3.2.2. GSA_INBAND_REKEY exchange

When the IKEv2 SA protecting the member registration exchange is
maintained while group member participates in the group, the GCKS can
use the GSA_INBAND_REKEY exchange to individually provide policy
updates to the group member.

```
   Member (Responder)              GCKS (Initiator)
  --------------------            ------------------
                            <-- HDR, SK { GSA, KD, [D,] }

            HDR, SK {} -->
```

Because this is an IKEv2 exchange, the HDR is treated as defined in
[RFC7296].

3.2.2.1. GSA_INBAND_REKEY GCKS Operations

The GSA, KD, and D payloads are built in the same manner as in a
registration exchange.

3.2.2.2. GSA_INBAND_REKEY GM Operations

The GM processes the GSA, KD, and D payloads in the same manner as if
they were received in a registration exchange.

3.2.3. Deletion of SAs

There are occasions when the GCKS may want to signal to group members
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 [RFC7296], section 3.11 as part of the GSA_REKEY
Exchange as shown below.

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

The GSA MAY specify the remaining active time of the remaining policy
by using the DTD attribute in the GSA GAP. If a GCKS has no further
SAs to send to group members, the GSA 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 the policies 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 defined in Section 4.6,
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. 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 G-IKEv2 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-based mode of operation, 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 unique key and nonce usage.

   During registration, the group member 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 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 the 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 zero as the
   first allocated SID.

   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 a group member may request a count of SIDs during
registration in a Notify payload information of type SEND_ID_REQUEST.
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_REGISTRATION
exchange originated by a sender, regardless of whether a group member
had previously contacted the GCKS. In this way, the GCKS is not
required to 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_REKEY message deleting all Data-
Security SAs and the Rekey SA for the group. This will result in the
group members initiating a new GSA_REGISTRATION 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_REKEY 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. Header and Payload Formats

Refer to IKEv2 [RFC7296] for existing payloads. Some payloads used
in G-IKEv2 exchanges are not aligned to 4-octet boundaries, which is
also the case for some IKEv2 payloads (see Section 3.2 of [RFC7296]).

4.1. The G-IKEv2 Header

G-IKEv2 uses the same IKE header format as specified in RFC 7296
section 3.1.

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

```
            Next Payload Type                  Value
            -----------------                  -----
            Group Identification (IDg)          50
            Group Security Association (GSA)     51
            Key Download (KD)                    52
```

New exchange types GSA_AUTH, GSA_REGISTRATION and GSA_REKEY are added
to the IKEv2 [RFC7296] protocol.

```
                 Exchange Type           Value
                 -------------           -----
                 GSA_AUTH                  39
                 GSA_REGISTRATION          40
                 GSA_REKEY                 41
                 GSA_INBAND_REKEY         TBD
```

Major Version is 2 and Minor Version is 0 as in IKEv2 [RFC7296]. IKE
SA Initiator's SPI, IKE SA Responder's SPI, Flags, Message ID, and
Length are as specified in [RFC7296].

4.2. Group Identification (IDg) Payload

The IDg Payload allows the group member to indicate which group it
wants to join. The payload is constructed by using the IKEv2
Identification Payload (section 3.5 of [RFC7296]). ID type ID_KEY_ID
MUST be supported. ID types ID_IPV4_ADDR, ID_FQDN, ID_RFC822_ADDR,
ID_IPV6_ADDR SHOULD be supported. ID types ID_DER_ASN1_DN and
ID_DER_ASN1_GN are not expected to be used.

4.3. Security Association - GM Supported Transforms (SAg)

The SAg payload declares which Transforms a GM is willing to accept.
The payload is constructed using the format of the IKEv2 Security
Association payload (section 3.3 of [RFC7296]). The Payload Type for
SAg is identical to the SA Payload Type.

4.4. Group Security Association Payload

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|   RESERVED  |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Security Association Payload fields are defined as follows:

o Next Payload (1 octet) -- Identifies the next payload type for the
   G-IKEv2 registration or the G-IKEv2 rekey message.

o Critical (1 bit) -- Set according to [RFC7296].

o RESERVED (7 bits) -- 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 policies.

4.4.1. GSA Policy

Following the GSA generic payload header are GSA policies for group
rekeying (KEK), data traffic SAs (TEK) and/or Group Associated Policy
(GAP). There may be zero or one GSA KEK policy, zero or one GAP
policies, and zero or more GSA TEK policies, where either one GSA KEK
or GSA TEK payload MUST be present.

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

Specifying multiple GSA TEKs allows multiple related data streams
(e.g., video, audio, and text) to be associated with a session, but
each protected with an individual security association policy.

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

Policies following the GSA payload use the following common header.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |   RESERVED    |              Length           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
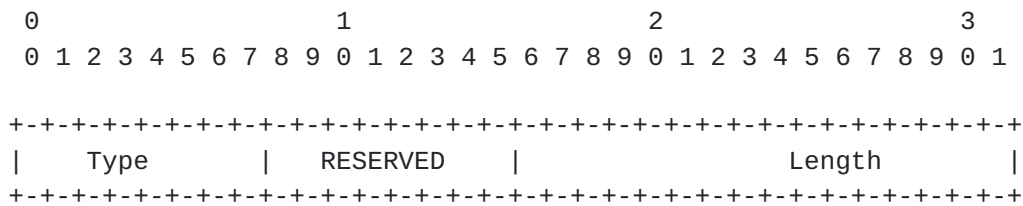
Type is defined as follows:

```
                      ID Class        Value
                      --------        -----
                      RESERVED          0
                      KEK               1
                      GAP               2
                      TEK               3
                      Expert Review   4-127
                      Private Use    128-255
```

4.5. KEK Policy

The GSA KEK policy contains security attributes for the KEK method
for a group and parameters specific to the G-IKEv2 registration
operation. The source and destination traffic selectors describe the
network identities used for the rekey messages.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Type = 1   !   RESERVED    !              Length           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                             SPI                               ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                    <Source Traffic Selector>                  ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                 <Destination Traffic Selector>                ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                  <Transform Substructure List>                ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                        KEK Attributes                         ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
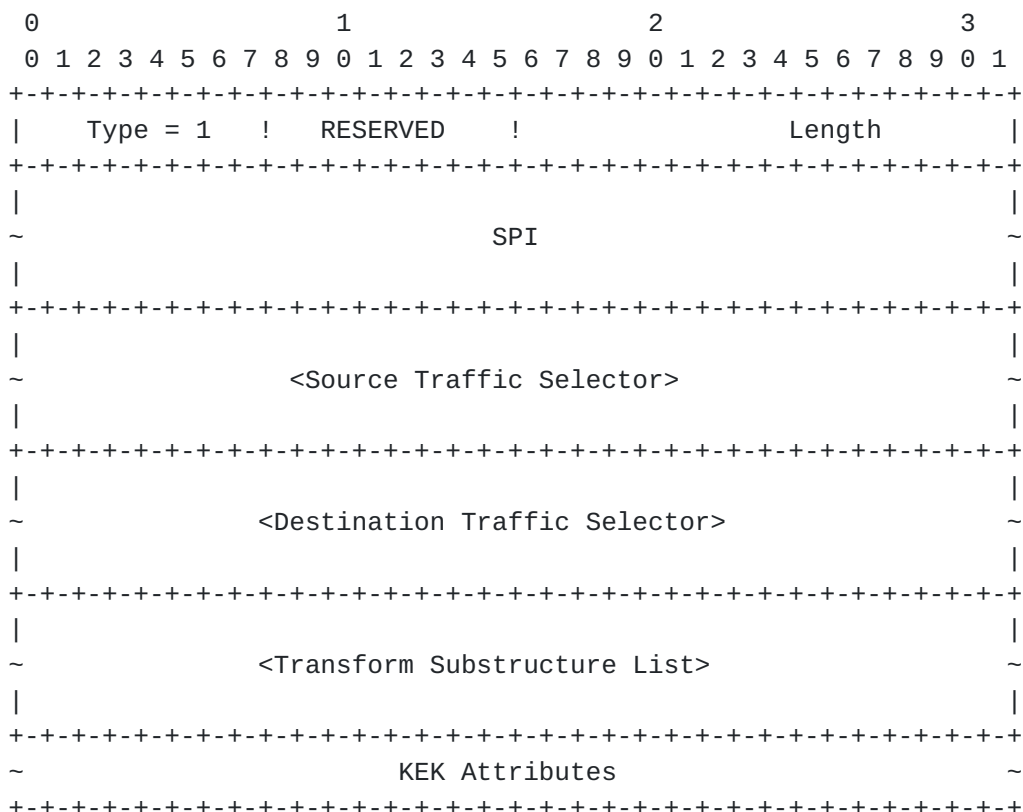
The GSA KEK Payload fields are defined as follows:

o Type = 1 (1 octet) -- Identifies the GSA payload type as KEK in
   the G-IKEv2 registration or the G-IKEv2 rekey message.

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

o Length (2 octets) -- Length of this structure including KEK
   attributes.

o SPI (16 octets) -- Security Parameter Index for the rekey message.
   The SPI must be the IKEv2 Header SPI pair where the first 8 octets
   become the "Initiator's SPI" field in 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 network identities. These
   identities refer to the source and destination of the next KEK
   rekey SA. Defined format and values are specified by IKEv2
   [RFC7296], section 3.13.1.

o Transform Substructure List -- A list of Transform Substructures
   specifies the transform information. The format is defined in
   IKEv2 [RFC7296], section 3.3.2, and values are described in the
   IKEv2 registries [IKEV2-IANA]. Valid Transform Types are ENCR,
   INTEG. The Last Substruc value in each Transform Substructure
   will be set to 3 except for the last one in the list, which is set
   to 0.

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.

4.5.1. KEK Attributes

The following attributes may be present in a GSA KEK policy. The
attributes must follow the format defined in the IKEv2 [RFC7296]
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). The terms Reserved, Unassigned, and Private Use are to
be applied as defined in [RFC8126]. The registration procedure is
Expert Review.

```
            ID Class                   Value    Type
            --------                   -----    ----
            Reserved                     0
            KEK_MANAGEMENT_ALGORITHM     1        B
            Reserved                     2
            Reserved                     3
            KEK_KEY_LIFETIME             4        V
            Reserved                     5
            KEK_AUTH_METHOD              6        B
            KEK_AUTH_HASH               7        B
            KEK_MESSAGE_ID              8        V
            Unassigned              9-16383
            Private Use          16384-32767
```

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_ENCR_ALGORITHM, KEK_KEY_LENGTH (if the cipher definition includes
a variable length key), KEK_MESSAGE_ID, KEK_KEY_LIFETIME,
KEK_INTEGRITY_ALGORITHM, KEK_AUTH_METHOD, and KEK_AUTH_HASH (except
for DSA based algorithms).

4.5.1.1. KEK_MANAGEMENT_ALGORITHM

The KEK_MANAGEMENT_ALGORITHM attribute 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. The terms Reserved, Unassigned,
and Private Use are to be applied as defined in [RFC8126]. The
registration procedure is Expert Review.

```
            KEK Management Type              Value
            -------------------              -----
            Reserved                           0
            LKH                                1
            Unassigned                      2-16383
            Private Use                  16384-32767
```

4.5.1.2. KEK_ENCR_ALGORITHM

The KEK_ENCR_ALGORITHM attribute specifies the encryption algorithm
used with the KEK. This value is a value from the IKEv2 Transform
Type 1 - Encryption Algorithm Transform IDs registry[IKEV2-IANA]. 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_ENCR_ALGORITHM attribute MUST be used for all keys which are
included as part of this KEK management.

4.5.1.3. KEK_KEY_LENGTH

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

The Group Controller/Key Server (GCKS) adds the KEK_KEY_LENGTH
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 the KEK_ENCR_ALGORITHM attribute and does not
need the KEK_KEY_LENGTH attribute. Sending the KEK_KEY_LENGTH
attribute in the GSA payload is OPTIONAL if the KEK cipher has a
fixed key length.

4.5.1.4. KEK_KEY_LIFETIME

The KEK_KEY_LIFETIME attribute 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.

4.5.1.5. KEK_INTEGRITY_ALGORITHM

The KEK_INTEGRITY attribute specifies the integrity algorithm used to
protect the rekey message. This integrity algorithm is a value from
the IKEv2 Transform Type 3 - Integrity Algorithm Transform IDs
registry [IKEV2-IANA].

4.5.1.6. KEK_AUTH_METHOD

The KEK_AUTH_METHOD attribute specifies the method of authentication
used. This value is from the IKEv2 Authentication Method registry
[IKEV2-IANA].

4.5.1.7. KEK_AUTH_HASH

The KEK_AUTH_HASH attribute specifies the hash algorithm used to
generate the AUTH key to authenticate GSA_REKEY messages. Hash
algorithms are defined in IANA registry IKEv2 Hash Algorithms
[IKEV2-IANA]. This attribute can be used by a group member to
determine in advance if it supports the algorithm used in the rekey
message.

4.5.1.8. KEK_MESSAGE_ID

   The KEK_MESSAGE_ID attribute defines the initial Message ID to be
   used by the GCKS in the GSA_REKEY messages. The Message ID is a 4
   octet unsigned integer in network byte order.

4.6. GSA TEK Policy

   The GSA TEK policy contains security attributes for a single TEK
   associated with a group.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Type = 3   |    RESERVED   |              Length           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Protocol-ID  |        TEK Protocol-Specific Payload           |
   +-+-+-+-+-+-+-+-+-+                                              ~
   ~                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
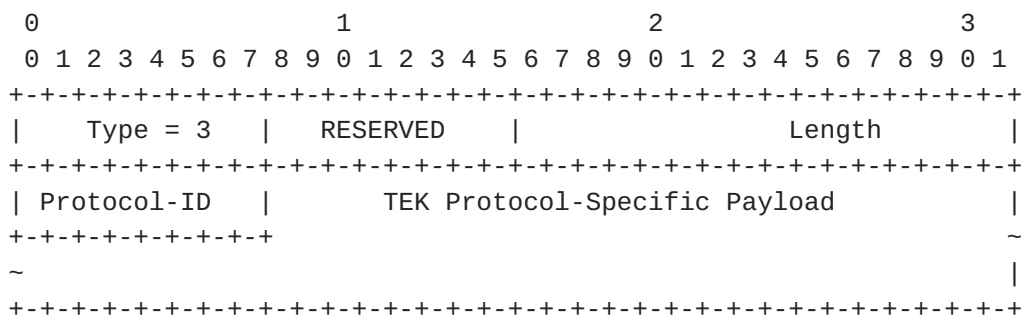
   The GSA TEK Payload fields are defined as follows:

   o Type = 3 (1 octet) -- Identifies the GSA payload type as TEK in
      the G-IKEv2 registration or the G-IKEv2 rekey message.

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

   o Length (2 octets) -- Length of this structure, 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.
      Support for the GSA_PROTO_IPSEC_AH GSA TEK is OPTIONAL. The terms
      Reserved, Unassigned, and Private Use are to be applied as defined
      in [RFC8126]. The registration procedure is Expert Review.

```
            Protocol ID                     Value
            -----------                     -----
            Reserved                          0
            GSA_PROTO_IPSEC_ESP               1
            GSA_PROTO_IPSEC_AH                2
            Unassigned                      3-127
            Private Use                    128-255
```

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

4.6.1. TEK ESP and AH Protocol-Specific Policy

   The TEK Protocol-Specific policy contains two traffic selectors one
   for the source and one for the destination of the protected traffic,
   SPI, Transforms, and Attributes.

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

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              SPI                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                              |
   ~                  <Source Traffic Selector>                   ~
   |                                                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                              |
   ~                <Destination Traffic Selector>                ~
   |                                                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                              |
   ~                 <Transform Substructure List>                ~
   |                                                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                         TEK Attributes                       ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   The GSA TEK Policy fields are defined as follows:

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

   o Source & Destination Traffic Selectors - The traffic selectors
      describe the source and the destination of the protected traffic.
      The format and values are defined in IKEv2 [RFC7296], section
      3.13.1.

   o Transform Substructure List -- A list of Transform Substructures
      specifies the transform information. The format is defined in
      IKEv2 [RFC7296], section 3.3.2, and values are described in the
      IKEv2 registries [IKEV2-IANA]. Valid Transform Types for ESP are
      ENCR, INTEG, and ESN. Valid Transform Types for AH are INTEG and
      ESN. The Last Substruc value in each Transform Substructure will
      be set to 3 except for the last one in the list, which is set to
      0. A Transform Substructure with attributes (e.g, the ENCR Key
      Length), they are included within the Transform Substructure as
      usual.

o TEK Attributes -- Contains the TEK policy attributes associated
   with the group, in the format defined in Section 3.3.5 of
   [RFC7296]. All attributes are optional, depending on the group
   policy.

Attribute Types are as follows. The terms Reserved, Unassigned, and
Private Use are to be applied as defined in [RFC8126]. The
registration procedure is Expert Review.

```
         ID Class                    Value    Type
         --------                    -----    ----
         Reserved                      0
         TEK_KEY_LIFETIME              1        V
         TEK_MODE                      2        B
         Unassigned                 3-16383
         Private Use              16384-32767
```
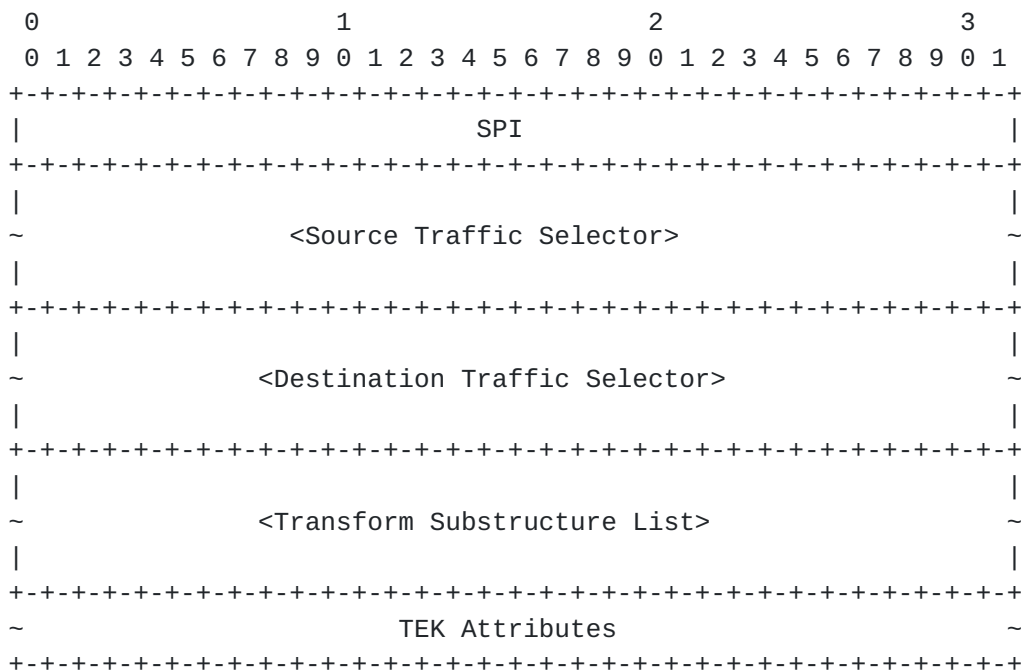
It is NOT RECOMMENDED that the GCKS distribute both ESP and AH
Protocol-Specific Policies for the same set of Traffic Selectors.

4.6.1.1. TEK_KEY_LIFETIME

The TEK_KEY_LIFETIME attribute specifies the maximum time for which
the TEK is valid. When the TEK expires, the AH or ESP security
association and all keys downloaded under the security association
are discarded. The GCKS may refresh the TEK 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. If unspecified the default value of 28800 seconds (8 hours)
shall be assumed.

4.6.1.2. TEK_MODE

The value of 0 is used for tunnel mode and 1 for transport mode. In
the absence of this attribute tunnel mode will be used.

4.7. GSA Group Associated Policy

Group specific policy that does not belong to rekey policy (GSA KEK)
or traffic encryption policy (GSA TEK) can be distributed to all
group member using GSA GAP (Group Associated Policy).

The GSA GAP payload is defined as follows:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |    Type = 2   !   RESERVED    !              Length           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  ~              Group Associated Policy Attributes               ~
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The GSA GAP payload fields are defined as follows:

o Type = 2 (1 octet) -- Identifies the GSA payload type as GAP in
   the G-IKEv2 registration or the G-IKEv2 rekey message.

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

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

o Group Associated Policy Attributes (variable) -- Contains
   attributes following the format defined in Section 3.3.5 of
   [RFC7296].

Attribute Types are as follows. The terms Reserved, Unassigned, and
Private Use are to be applied as defined in [RFC8126]. The
registration procedure is Expert Review.

```
            Attribute Type          Value        Type
            --------------          -----        ----
            Reserved                  0
            ACTIVATION_TIME_DELAY     1            B
            DEACTIVATION_TIME_DELAY   2            B
            Unassigned              3-16383
            Private Use            16384-32767
```

4.7.1. ACTIVATION_TIME_DELAY/DEACTIVATION_TIME_DELAY

Section 4.2.1 of RFC 5374 specifies a key rollover method that
requires two values be provided to group members. 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 it should deactivate SAs that are
destroyed by the rekey 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.

4.8. Key Download Payload

   The Key Download Payload contains the group keys for the group
   specified in the GSA 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 GSA Payload.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Next Payload  |C|  RESERVED   |                Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Number of Key Packets         |            RESERVED2          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-!
   ~                       Key Packets                            ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
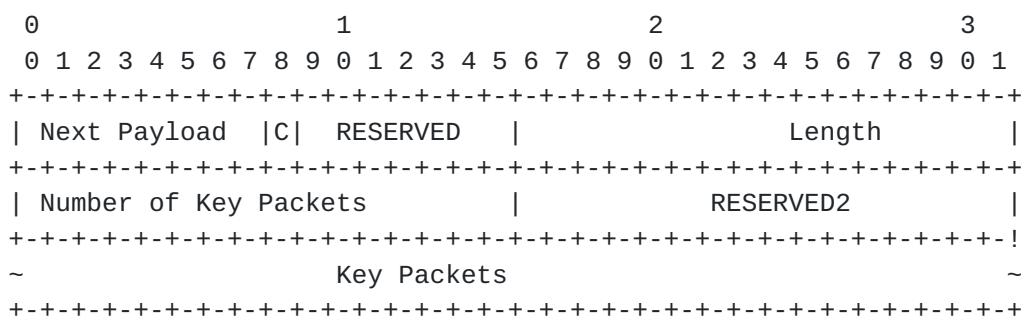
   The Key Download Payload fields are defined as follows:

   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 Critical (1 bit) -- Set according to [RFC7296].

   o RESERVED (7 bits) -- 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
      Key Packets passed in this data block.

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   KD Type     |   RESERVED    |            KD Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   SPI Size    |                    SPI (variable)             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Key Packet Attributes                      ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
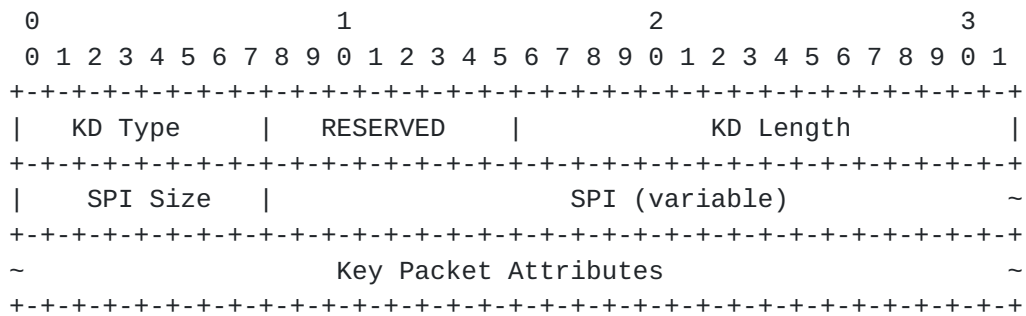
o Key Download (KD) Type (1 octet) -- Identifier for the Key Data
   field of this Key Packet. In the following table the terms
   Reserved, Unassigned, and Private Use are to be applied as defined
   in [RFC8126]. The registration procedure is Expert Review.

```
                 Key Download Type        Value
                 -----------------        -----
                 Reserved                   0
                 TEK                        1
                 KEK                        2
                 LKH                        3
                 SID                        4
                 Unassigned               5-127
                 Private Use             128-255
```

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 GSA KEK or GSA TEK 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.

4.8.1. TEK Download Type

The following attributes may be present in a TEK Download Type.
Exactly one attribute matching each type sent in the GSA TEK payload
MUST be present. The attributes must follow the format defined in
IKEv2 (Section 3.3.5 of [RFC7296]). In the table, attributes defined
as TV are marked as Basic (B); attributes defined as TLV are marked

as Variable (V). The terms Reserved, Unassigned, and Private Use are
to be applied as defined in [RFC8126]. The registration procedure is
Expert Review.

```
            TEK Class                 Value      Type
            ---------                 -----      ----
            Reserved                    0
            TEK_ALGORITHM_KEY           1          V
            TEK_INTEGRITY_KEY           2          V
            Unassigned                3-16383
            Private Use             16384-32767
```

It is possible that the GCKS will send no TEK key packets in a
Registration KD payload (as well as no corresponding GSA TEK payloads
in the GSA payload), after which the TEK payloads will be sent in a
rekey message. At least one TEK MUST be included in each Rekey KD
payload.

4.8.1.1. TEK_ALGORITHM_KEY

The TEK_ALGORITHM_KEY class contains encryption keying material for
the corresponding SPI. This keying material will be used with the
encryption algorithm specified in the GSA TEK payload, and according
to the IPsec transform describing that encryption algorithm. The
keying material is treated equivalent to IKEv2 KEYMAT derived for
that IPsec transform. If the encryption algorithm requires a nonce
(e.g., AES-GCM), the nonce is chosen as shown in Section 3.2.

4.8.1.2. TEK_INTEGRITY_KEY

The TEK_INTEGRITY_KEY class declares that the integrity key for the
corresponding SPI is contained in the Key Packet Attribute. Readers
should refer to [IKEV2-IANA] for the latest values.

4.8.2. KEK Download Type

The following attributes may be present in a KEK Download Type.
Exactly one attribute matching each type sent in the GSA KEK payload
MUST be present. The attributes must follow the format defined in
IKEv2 (Section 3.3.5 of [RFC7296]). In the table, attributes defined
as TV are marked as Basic (B); attributes defined as TLV are marked
as Variable (V). The terms Reserved, Unassigned, and Private Use are
to be applied as defined in [RFC8126]. The registration procedure is
Expert Review.

```
                    KEK Class                 Value      Type
                    ---------                 -----      ----
                    Reserved                    0
                    KEK_ENCR_KEY                1         V
                    KEK_INTEGRITY_KEY           2         V
                    KEK_AUTH_KEY                3         V
                    Unassigned                4-16383
                    Private Use              16384-32767
```

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

4.8.2.1. KEK_ENCR_KEY

   The KEK_ENCR_KEY class declares that the encryption key for the
   corresponding SPI is contained in the Key Packet Attribute. The
   encryption algorithm that will use this key was specified in the GSA
   KEK payload.

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

4.8.2.2. KEK_INTEGRITY_KEY

   The KEK_INTEGRITY_KEY class declares the integrity key for this SPI
   is contained in the Key Packet Attribute. The integrity algorithm
   that will use this key was specified in the GSA KEK payload.

4.8.2.3. KEK_AUTH_KEY

   The KEK_AUTH_KEY class declares that the authentication key for this
   SPI is contained in the Key Packet Attribute. The signature
   algorithm that will use this key was specified in the GSA KEK
   payload. An RSA public key format is defined in RFC 3447,
   Section A.1.1. DSS public key format is defined in RFC 3279
   Section 2.3.2. For ECDSA Public keys, use format described in RFC
   5480 Section 2.2.

4.8.3. 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 [RFC7296]). In the table, attributes defined as TV
   are marked as Basic (B); attributes defined as TLV are marked as

   Variable (V). The terms Reserved, Unassigned, and Private Use are to
   be applied as defined in [RFC8126]. The registration procedure is
   Expert Review.

```
                LKH Download Class          Value      Type
                ------------------          -----      ----
                Reserved                      0
                LKH_DOWNLOAD_ARRAY            1          V
                LKH_UPDATE_ARRAY              2          V
                Unassigned                  3-16383
                Private Use               16384-32767
```

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

4.8.3.1. LKH_DOWNLOAD_ARRAY

   The LKH_DOWNLOAD_ARRAY class is used to download a set of LKH 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 one 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.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         # of LKH Keys         |            RESERVED           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                            LKH Keys                           ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
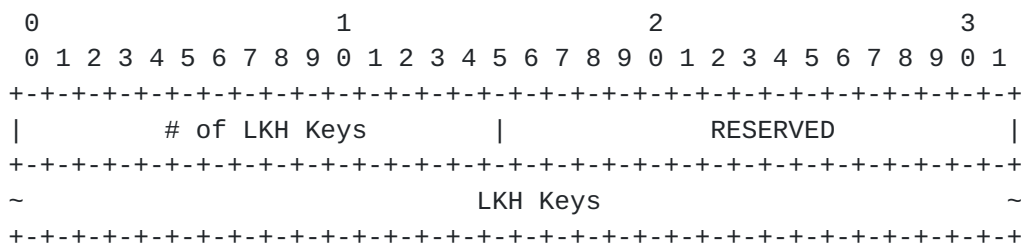
   The KEK_LKH attribute fields are defined as follows:

   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:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!            LKH ID             |            Encr Alg           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Key Handle                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          Key Data                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

o Encr Alg (2 octets) -- This is the encryption algorithm for which
   this key data is to be used. This value is specified in
   Section 4.5.1.2.

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

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

o Key Data (variable length) -- This is the actual encryption key
   data, which is dependent on the Encr Alg 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 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 the order starting from the leaf, culminating in the group KEK.

4.8.3.2. LKH_UPDATE_ARRAY

The LKH_UPDATE_ARRAY class is used to update the LKH 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 4.8.3.1.

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

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           # of LKH Keys           |           LKH ID          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Key Handle                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                           LKH Keys                            ~
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

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

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

The LKH Keys are as defined in Section 4.8.3.1. The LKH Key
structures contain keys along the path of the key tree in the 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.

4.8.4. SID Download Type

The SID attribute is used to download one or more Sender-ID (SID)
values for the exclusive use of a group member. The terms Reserved,
Unassigned, and Private Use are to be applied as defined in
[RFC8126]. The registration procedure is Expert Review.

```
        SID Download Class          Value      Type
        ------------------          -----      ----
        Reserved                      0
        NUMBER_OF_SID_BITS            1          B
        SID_VALUE                     2          V
        Unassigned                 3-16383
        Private Use              16384-32767
```

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

4.8.4.1. 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 is applied to
each SID value distributed in the SID Download.

4.8.4.2. SID_VALUE

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

4.8.4.3. 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 NOT 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 ignore an SID Download Type KD payload present in
a GSA-REKEY message, otherwise more than one GM may end up using the
same SID.

4.8.4.4. 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_REKEY message,
because distributing the same sender-specific policy to more than one
group member will reduce the security of the group.

4.9. Delete Payload

There are occasions when the GCKS may want to signal to group members
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 [RFC7296] as part of the GSA_AUTH or

GSA_REKEY Exchange. One or more Delete payloads MAY be placed
following the HDR payload in the GSA_AUTH or GSA_REKEY Exchange.

The Protocol ID MUST be 41 for GSA_REKEY Exchange, 2 for AH or 3 for
ESP. Note that only one protocol id value can be defined in a Delete
payload. If a TEK and a KEK SA for GSA_REKEY Exchange 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.

There may be circumstances where the GCKS may want to reset the
policy and keying material for the group. The GCKS can signal
deletion of all policy of a particular TEK by sending a TEK with a
SPI value equal to zero in the delete payload. In the event that the
administrator is no longer confident in the integrity of the group
they may wish to remove all KEK and all the TEKs in the group. This
is done by having the GCKS send a delete payload with a SPI of zero
and a Protocol-ID of AH or ESP to delete all TEKs, followed by
another delete payload with a SPI value of zero and Protocol-ID of
KEK SA to delete the KEK SA.

4.10. Notify Payload

G-IKEv2 uses the same Notify payload as specified in [RFC7296],
 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 -                      45
Indicates the group id sent during the registration process is
invalid.

AUTHORIZATION_FAILED -                  46
Sent in the response to a GSA_AUTH message when authorization failed.

NOTIFY messages - status types         Value
----------------------------------------------------------------------
SENDER_ID_REQUEST   -                   16429
Sent in GSA_AUTH or GSA_REGISTRATION to request SIDs from the GCKS.
The data includes a count of how many SID values it desires.
```

4.11. Authentication Payload

   G-IKEv2 uses the same Authentication payload as specified in
   [RFC7296], 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 protocols,
   inheriting all the security considerations documented in [RFC7296]
   section 5 Security Considerations, including authentication,
   confidentiality, protection against man-in-the-middle, protection
   against replay/reflection attacks, and denial of service protection.
   The GSA_AUTH and GSA_REGISTRATION exchanges also take advantage of
   those protections. 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 came
   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 a 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 Message ID
   number to that of the last received rekey message, any rekey message

containing a Message ID number less than or equal to the last
received value MUST 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 should be created for G-IKEv2, on a new page
titled Group Key Management using IKEv2 (G-IKEv2) Parameters. The
following registries should be placed on that page. The terms
Reserved, Expert Review and Private Use are to be applied as defined
in [RFC8126].

GSA Policy Type Registry, see Section 4.4.1

KEK Attributes Registry, see Section 4.5.1

KEK Management Algorithm Registry, see Section 4.5.1.1

GSA TEK Payload Protocol ID Type Registry, see Section 4.6

TEK Attributes Registry, see Section 4.6

Key Download Type Registry, see Section 4.8

TEK Download Type Attributes Registry, see Section 4.8.1

KEK Download Type Attributes Registry, see Section 4.8.2

LKH Download Type Attributes Registry, see Section 4.8.3

SID Download Type Attributes Registry, see Section 4.8.4

6.2. New payload and exchange types added to the existing IKEv2 registry

The following new payloads and exchange types specified in this memo
have already been allocated by IANA and require no further action,
other than replacing the draft name with an RFC number.

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 notification types, see
[Section 4.10](#)

7. Acknowledgements

The authors thank Lakshminath Dondeti and Jing Xiang for first
exploring the use of IKEv2 for group key management and providing the
basis behind the protocol. Mike Sullenberger and Amjad Inamdar were
instrumental in helping resolve many issues in several versions of
the document.

8. Contributors

The following individuals made substantial contributions to early
versions of this memo.

    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
    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

        Yoav Nir
        Dell EMC
        9 Andrei Sakharov St
        Haifa 3190500
        Israel

        Email: ynir.ietf@gmail.com

9. References

9.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC2627] Wallner, D., Harder, E., and R. Agee, "Key Management for
              Multicast: Issues and Architectures", RFC 2627,
              DOI 10.17487/RFC2627, June 1999,
              <https://www.rfc-editor.org/info/rfc2627>.

   [RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security
              Architecture", RFC 3740, DOI 10.17487/RFC3740, March 2004,
              <https://www.rfc-editor.org/info/rfc3740>.

   [RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm,
              "Multicast Security (MSEC) Group Key Management
              Architecture", RFC 4046, DOI 10.17487/RFC4046, April 2005,
              <https://www.rfc-editor.org/info/rfc4046>.

   [RFC4301] Kent, S. and K. Seo, "Security Architecture for the
              Internet Protocol", RFC 4301, DOI 10.17487/RFC4301,
              December 2005, <https://www.rfc-editor.org/info/rfc4301>.

   [RFC6054] McGrew, D. and B. Weis, "Using Counter Modes with
              Encapsulating Security Payload (ESP) and Authentication
              Header (AH) to Protect Group Traffic", RFC 6054,
              DOI 10.17487/RFC6054, November 2010,
              <https://www.rfc-editor.org/info/rfc6054>.

   [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T.
              Kivinen, "Internet Key Exchange Protocol Version 2
              (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October
              2014, <https://www.rfc-editor.org/info/rfc7296>.

   [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for
              Writing an IANA Considerations Section in RFCs", BCP 26,
              RFC 8126, DOI 10.17487/RFC8126, June 2017,
              <https://www.rfc-editor.org/info/rfc8126>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

   [I-D.ietf-ipsecme-qr-ikev2]
              Fluhrer, S., McGrew, D., Kampanakis, P., and V. Smyslov,
              "Postquantum Preshared Keys for IKEv2", draft-ietf-
              ipsecme-qr-ikev2-04 (work in progress), July 2018.

   [IKEV2-IANA]
              IANA, "Internet Key Exchange Version 2 (IKEv2)
              Parameters", February 2016,
              <http://www.iana.org/assignments/ikev2-parameters/
              ikev2-parameters.xhtml#ikev2-parameters-7>.

   [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>.

   [RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange
              (IKE)", RFC 2409, DOI 10.17487/RFC2409, November 1998,
              <https://www.rfc-editor.org/info/rfc2409>.

   [RFC3686] Housley, R., "Using Advanced Encryption Standard (AES)
              Counter Mode With IPsec Encapsulating Security Payload
              (ESP)", RFC 3686, DOI 10.17487/RFC3686, January 2004,
              <https://www.rfc-editor.org/info/rfc3686>.

   [RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode
              (GCM) in IPsec Encapsulating Security Payload (ESP)",
              RFC 4106, DOI 10.17487/RFC4106, June 2005,
              <https://www.rfc-editor.org/info/rfc4106>.

   [RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM
              Mode with IPsec Encapsulating Security Payload (ESP)",
              RFC 4309, DOI 10.17487/RFC4309, December 2005,
              <https://www.rfc-editor.org/info/rfc4309>.

   [RFC4543] McGrew, D. and J. Viega, "The Use of Galois Message
              Authentication Code (GMAC) in IPsec ESP and AH", RFC 4543,
              DOI 10.17487/RFC4543, May 2006,
              <https://www.rfc-editor.org/info/rfc4543>.

   [RFC5723] Sheffer, Y. and H. Tschofenig, "Internet Key Exchange
              Protocol Version 2 (IKEv2) Session Resumption", RFC 5723,
              DOI 10.17487/RFC5723, January 2010,
              <https://www.rfc-editor.org/info/rfc5723>.

   [RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain
              of Interpretation", RFC 6407, DOI 10.17487/RFC6407,
              October 2011, <https://www.rfc-editor.org/info/rfc6407>.

   [RFC7383] Smyslov, V., "Internet Key Exchange Protocol Version 2
              (IKEv2) Message Fragmentation", RFC 7383,
              DOI 10.17487/RFC7383, November 2014,
              <https://www.rfc-editor.org/info/rfc7383>.

Appendix A. Use of LKH in G-IKEv2

   Section 5.4 of [RFC2627] describes the LKH architecture, and how a
   GCKS uses LKH to exclude group members. This section clarifies how
   the LKH architecture is used with G-IKEv2.

A.1. Group Creation

   When a GCKS forms a group, it creates a key tree as shown in the
   figure below. The key tree contains logical keys (represented as
   numbers in the figure) and a private key shared with only a single GM
   (represented as letters in the figure). Note that the use of numbers
   and letters is used for explanatory purposes; in fact, each key would
   have an LKH ID, which is two-octet identifier chosen by the GCKS.
   The GCKS may create a complete tree as shown, or a partial tree which
   is created on demand as members join the group. The top of the key
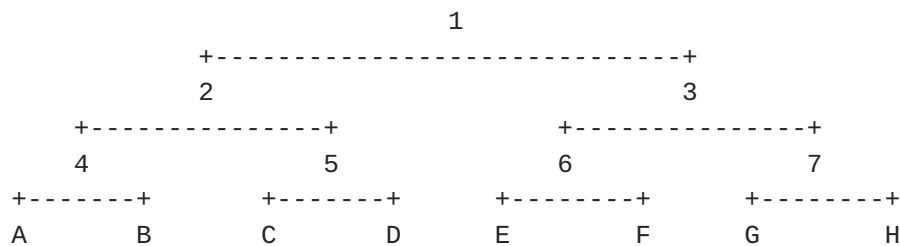   tree (i.e., "1" in Figure 3) is used as the KEK for the group.

```
                                  1
                  +-------------------------------+
                  2                               3
            +---------------+             +---------------+
            4               5             6               7
         +-------+       +-------+     +--------+     +--------+
         A       B       C       D     E        F     G        H
```

Figure 3: Initial LKH tree

When GM "A" joins the group, the GCKS provides an LKH_DOWNLOAD_ARRAY
in the KD payload of the GSA_AUTH or GSA_REGISTRATION exchange.
Given the tree shown in figure above, the LKH_DOWNLOAD_ARRAY will
contain four LKH Key payloads, each containing an LKH ID and Key
Data. If the LKH ID values were chosen as shown in the figure, four
LKH Keys would be provided to GM "A", in the following order: A, 4,
2, 1. When GM "B" joins the group, it would also be given four LKH
Keys in the following order: B, 4, 2, 1. And so on, until GM "H"
joins the group and is given H, 7, 3, 1.

A.2. Group Member Exclusion

If the GKCS has reason to believe that a GM should be excluded, then
it can do so by sending a GSA_REKEY exchange that includes a set of
LKH_UPDATE_ARRAY attributes in the KD payload. Each LKH_UPDATE_ARRAY
contains a set of LKH Key payloads, in which every GM other than the
excluded GM will be able to determine a set of new logical keys,
which culminate in a new key "1". The excluded GM will observe the
set of LKH_UPDATE_ARRAY attributes, but cannot determine the new
logical keys because each of the "Key Data" fields is encrypted with
a key held by other GMs. The GM will hold no keys to properly
decrypt any of the "Key Data" fields, including key "1" (i.e., the
new KEK). When a subsequent GSA_REKEY exchange is delivered by the
GCKS and protected by the new KEK, the excluded GM will no longer be
able to see the contents of the GSA_REKEY, including new TEKs that
will be delivered to replace existing TEKs. At this point, the GM
will no longer be able to participate in the group.

In the example below, new keys are represented as the number followed
by a "prime" symbol (e.g., "1" becomes "1'"). Each key is encrypted
by another key. This is represented as "{key1}key2", where key2
encrypts key1. For example, "{1'}2' states that a new key "1'" is
encrypted with a new key "2'".

If GM "B" is to be excluded, the GCKS will need to include three
LKH_UPDATE_ARRAY attributes in the GSA_REKEY message. The order of
the attributes does not matter; only the order of the keys within
each attribute.

o One will provide GM "A" with new logical keys that are shared with
   B: {4'}A, {2'}4', {1'}2'

o One will provide all GMs holding key "5" with new logical keys:
   {2'}5, {1'}2'

o One will provide all GMs holding key "3" with a new KEK: {1'}3

Each GM will look at each LKH_UPDATE_ARRAY attribute and observe an
LKH ID which is present in an LKH Key delivered to them in the
LKH_DOWNLOAD_ARRAY they were given. If they find a matching LKH ID,
then they will decrypt the new key with the logical key immediately
preceding that LKH Key, and so on until they have received the new 1'
key.

The resulting key tree from this rekey event would would be shown in
Figure 4.

```
                            1'
             +------------------------------+
             2'                            3
      +---------------+           +---------------+
      4'              5           6               7
    +---+         +-------+   +--------+     +--------+
    A       B     C       D   E        F     G        H
```
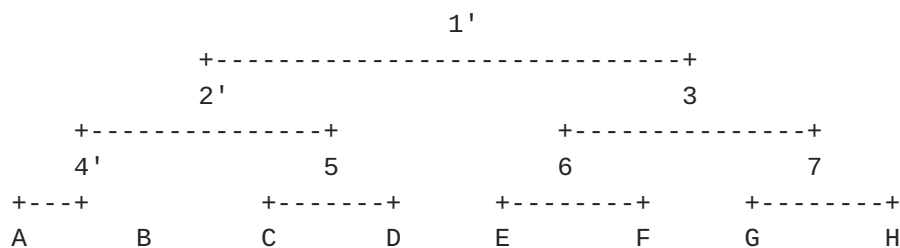
              Figure 4: LKH tree after B has been excluded

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

   Valery Smyslov
   ELVIS-PLUS
   PO Box 81
   Moscow (Zelenograd) 124460
   Russian Federation

   Phone: +7 495 276 0211
   Email: svan@elvis.ru