MSEC Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 08, 2014

S. Rowles
A. Yeung, Ed.
P. Tran
Cisco Systems
Y. Nir
Check Point Software Technologies Ltd.

November 04, 2013

Group Key Management using IKEv2
draft-yeung-g-ikev2-07

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on May 08, 2014.

Table of Contents

1. Introduction and Overview

This document presents a group key management protocol protected by IKEv2. The data communications within the group are protected by a key pushed to the group members (GMs) by the Group Controller/Key Server (GCKS) using IKEv2 [RFC5996]. The GCKS pushes policy and keys for the group to the GM after authenticating it using new payloads added in the IKE_AUTH exchange. This document references IKEv2 [RFC5996] but it intended to be a separate document. GDOI update document [RFC6407] presented GDOI using IKEv1 syntax. This document uses IKEv2 syntax. The message semantics of IKEv2 are preserved, in that all communications consists of message request-response pairs. The exception to this rule are the rekeying messages, which are sent in multicast without a response. A number of payloads were deemed unnecessary since [RFC6407] are described in Appendix A

1.1. Requirements Language

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

1.2. Why do we need another GSA protocol?

GDOI protocol specified in [RFC6407] is protected by IKEv1 phase1 security association defined in [RFC2407], [RFC2408] and [RFC2409]; these documents are obsoleted and replaced by a new version of the IKE protocol defined in RFC 5996. G-IKEv2 provides group key management between the group member and group controller key server using the new IKEv2 protocol and inherits the following key advantages over GDOI:

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

2. Improve performance and network latency by the reduced number of initial messages to complete the G-IKEv2 protocol from (10 messages in main mode and quick mode, 7 messages in aggressive mode and quick) to 4 messages.

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

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

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

1.3. 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

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

2.1. UDP port

G-IKEv2 SHOULD use port 848, the same as GDOI [RFC6407] , because they serve a similar function, and 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].

3. G-IKEv2 Protocol

3.1. G-IKEv2 member registration and secure channel establishment

The registration protocol consists of minimum two exchanges IKE_SA_INIT and GSA_AUTH; member registration may have a few more messages exchanged if the EAP method, cookie challenge (for DoS) or invalid KE are used. Each exchange consists of request/response pairs. The first exchange IKE_SA_INIT is defined in IKEv2 [RFC5996]. This exchange negotiates cryptographic algorithms, exchanges nonces and does a Diffie-Hellman exchange between the member and the Group Controller Key Server (GCKS).

The second exchange GSA_AUTH authenticates the previous messages, exchange identities and certificates, . Parts of 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 MAY authorize group members to be allowed into the group as part of the GSA_AUTH exchange. Once the GCKS accepted a group member to join a group it may downloads the data security keys (TEKs) and/or group key encrypting key (KEK) or KEK array as part of GSA_AUTH response message. In the following descriptions, the payloads contained in the message are indicated by names as listed below.

Notation	Payload
AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
GSA	Group Security Association
HDR	IKEv2 Header

IDg	Identification - Group
IDi	Identification - Initiator
IDr	Identification - Responder
KD	Key Download
KE	Key Exchange
Ni, Nr	Nonce SA Security Association

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

3.1.1. GSA_AUTH exchange

After the group member and GCKS uses IKE_SA_INIT exchange to negotiate cryptographic algorithms, exchange nonces, and perform a Diffie-Hellman exchange as defined in IKEv2 [RFC5996], the GSA_AUTH MAY be used for the GCKS to download group policy and keys to the member. 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 optionally authorization. GSA_AUTH does not include SA2, TSi and TSr since policy is not negotiated between group member and GCKS but downloaded from the GCKS to the group member.

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

After an unauthenticated secure channel is established by IKE_SA_INIT exchange, the member initiates a registration request to join a group indicated by the IDg payload. The GM can include 0 or more Notify payload. Notify payload status type SENDER_ID_REQUEST is indicate a request SIDs for Counter-based cipher from the GCKS.

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

Besides response with IDr, optional CERT, and AUTH material, the GCKS MAY also informs the member the cryptographic policies of the group in the GSA payload and key material in the KD payload. GCKS can also include Delete payload instructing the group member to delete existing SAs it might have.

In addition to the IKEv2 error handling, GCKS can reject the registration request when IDg is invalid or authorization fail, etc. In these cases, see Section 4.9, the GSA_AUTH response will include notify indicate errors. The member SHOULD delete the registration IKE SA.

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

When the group member found the policy sent by the GCKS is unacceptable, the member SHOULD send an IKEv2 delete using the INFORMATION message exchange to bring down the authenticated IKE SA.

3.1.2. GSA_REGISTRATION Exchange

When a secure channel is already established between GM and KS, 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. The GM MAY also include the Notify payload status type SENDER_ID_REQUEST to request SIDs for Counter-based cipher from the GCKS.

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

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

This exchange can also be used when the group member found the policy sent by the GCKS is unacceptable. The group member can 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 number (0) according to IKEv2 [RFC5996], and maintained in this document. The G-IKEv2

IKE_SA_INIT, GSA_AUTH and GSA_REGISTRATION use the SPI according to IKEv2 [RFC5996],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) 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 [section 3.2].

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

The GSA payload contains the security protocol and cryptographic protocols used by the group. This policy describes the Re-key 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 tear down the session after notifying the GCKS. Finally the KD is parsed providing the keying material for the TEK and/or KEK. The GM interprets the KD key packets, where each key packet includes the keying material for SAs distributed in the GSA payload. Keying material is matched by comparing the SPIs in the key packets to SPIs previously 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 has to have 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. The GCKS receives the IKE_SA_INIT request, select the IKE proposal, generates nonce and DH to include them in the IKE_SA_INIT response.

Upon receiving the GSA_AUTH request, and after authenticate the group member using the same properties as IKEv2, the GCKS locates the group the GM wishes to join, extracts the policy for that group. If the GCKS policy desires authorization, the GCKS authorizes the group member against the specified credentials before preparing to send GSA_AUTH response. The GSA_AUTH response MAY include group policy in GSA payload and keys in the KD payload. If the GCKS policy include 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 attribute KEK_MESSAGE_ID specify the starting message id the GCKS will be using when sending the GSA_REKEY message to the group member. This message id is used prevent replay attacks of the GSA_REKEY message and will be increasing 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 KD TEK. GSA GAP MAY also included to provide the ATD and /or DTD [section 4.6.1] 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 corresponding group policy and keys. If the GCKS fails to authorize the GM, it will respond with the AUTHORIZATION_FAILED notify message.

3.2. Counter-based modes of operation

Several new counter-based modes of operation have been specified for ESP (e.g., AES-CTR [RFC3686], AES-GCM [RFC4106], AES-CCM [RFC4309], AES-GMAC [RFC4543]) and AH (e.g., AES-GMAC [RFC4543]). These counter-based modes require that no two senders in the group ever send a packet with the same Initialization Vector (IV) using the same cipher key and mode. This requirement is met in 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-mode, the GCKS automatically allocates and distributes one SID to each group member acting in the role of sender on the Data-Security SA. The SID value is used exclusively by the group member to which it was allocated. The group member uses the same SID for each Data-Security SA specifying the use of a counter-based mode of operation. A GCKS MUST distribute unique keys for each Data-Security SA including a counter-based mode of operation in order to maintain a unique key and nonce usage.

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

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

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

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

3. When the GCKS specifies a counter-based mode of operation in the Data Security SA, and a group member is a sender, a group member may request a count of SIDs during registration in a NOTIFY payload information 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 does not have a requirement of maintaining a record of which SID values it had

previously allocated to each group member. More importantly, since the GCKS cannot reliably detect whether the group member had sent data on the current group Data-Security SAs it does not know what Data-Security counter-mode nonce values that a group member has used. By distributing new SID values, the key server ensures that each time a conforming group member installs a Data- Security SA it will use a unique set of counter-based mode nonces.

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

6. The GCKS SHOULD send a GSA_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.

3.3. G-IKEv2 group maintenance channel

The GCKS indicates that it will be delivering group rekey messages when the KEK poilcy and keys are present in the G-IKEv2 GSA and KD payloads. Though the G-IKEv2 Rekey is optional, it plays a crucial role for large and dynamic groups. The GCKS is responsible for rekeying of the secure group per the group policy. The GCKS uses multicast to transport the rekey message. The G-IKEv2 protocol uses GSA_REKEY exchange type in G-IKEv2 header identifying it as a rekey message. This rekey message is protected by the registration exchanges.

3.3.1. G-IKEv2 GSA_REKEY exchange

The GCKS initiates the G-IKEv2 Rekey securely using IP multicast. Since multicast rekey does not require a response and it sends to multiple GMs, G-IKEv2 rekeying MUST NOT support 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 [section 4.7.4] in the Key Download payload 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 group member in the KEK attribute KEK_MESSAGE_ID during registration; this message id will be increasing 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 data security SA and/or a new rekey SA, which, optionally contains an LKH rekey SA, Section 4.3.

The KD represents the keys for the policy 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.

The Delete payload is included to instruct the GM to delete existing SAs.

The AUTH payload is a signature of the hash of the entire GSA_REKEY message before it has been encrypted.

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

3.3.2. Forward and Backward Access Control

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

access control" have been called "perfect forward security" and "perfect backward security" in the literature [RFC2627].

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

Support for group management algorithms is supported via the KEY_MANAGEMENT_ALGORITHM attribute which is sent in the 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. Any such addition MUST be due to a Standards Action as defined in [RFC2434].

3.3.3. Forward Access Control Requirements

When group membership is altered using a group management algorithm new 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 keys because it will be protected with the new KEK, and thus will not be visible to the members who were denied access.

If forward access control policy for the group includes keeping group policy changes from members that are denied access to the group, then two sequential G-IKEv2 rekey messages changing the group KEK MUST be sent by the GCKS. The first G-IKEv2 rekey message creates a new KEK for the group. Group members, which are denied access, will not be able to access the new KEK, but will see the group policy since the G-IKEv2 rekey message is protected under the current KEK. A subsequent G-IKEv2 rekey message containing the changed group policy and again changing the KEK allows complete forward access control. A G-IKEv2 rekey message MUST NOT change the policy without creating a new KEK.

If other methods of using LKH or other group management algorithms are added to G-IKEv2, those methods MAY remove the above restrictions

requiring multiple G-IKEv2 rekey messages, providing those methods specify how forward access control policy is maintained within a single G-IKEv2 rekey message.

3.3.4. Deletion of SAs

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

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

```
 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 SA and KD payloads MUST be omitted from the message. There may be circumstances where the GCKS may want to start over with a clean slate. If the administrator is no longer confident in the integrity of the group, the GCKS can signal deletion of all policy of a particular TEK protocol by sending a TEK with a SPI value equal to zero in the delete payload. For example, if the GCKS wihses to remove all the KEKs and all the TEKs in the group, the GCKS SHOULD send a delete payload with a spi of zero and a protocol_id of a TEK protocol_id value define in Section 4.5, followed by another delete payload with a spi of zero and protocol_id of zero, indicating that the KEK SA should be deleted.

3.3.5. GSA_REKEY GCKS Operations

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

3.3.6. GSA_REKEY GM Operations

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

Anti-replay protection is achieved when the group member rejects GSA_REKEY message which has 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 the subsequence GSA_REKEY message to be greater than the last valid GSA_REKEY message it received.

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

4. Header and Payload Formats

Refer to IKEv2 [RFC5996] for existing payloads.

4.1. The G-IKEv2 Header

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

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

Next Payload Type	Value
Group Identification (IDg)	TBD
Group Security Association (GSA)	TBD
Key Download (KD)	TBD

New exchange types GSA_AUTH, GSA_REGISTRATION and GSA_REKEY are added to the IKEv2 [RFC5996] protocol.

Exchange Type	Value
GSA_AUTH	TBD
GSA_REGISTRATION	TBD
GSA_REKEY	TBD

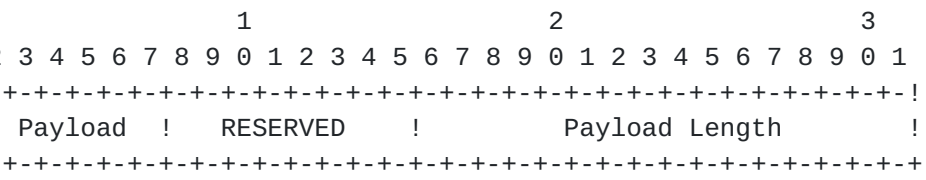
Major Version is 2 and Minor Version is 0 as in IKEv2 [RFC5996]. IKE SA initiator SPI, IKE SA responder SPI, Flags, Message Id are as specified in [RFC5996].

4.2. IDgroup Payload

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

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



The Security Association Payload fields are defined as follows:

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

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

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

4.3.1. GSA policy

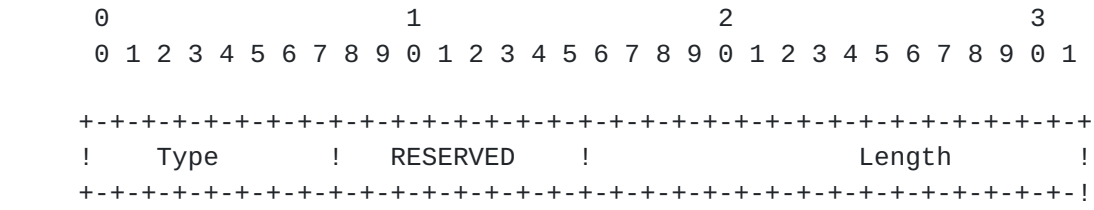
Following GSA generic payload header are GSA policies for group rekeying (KEK) and/or data traffic SAs (TEK). There may be zero or one GSA KEK policy, zero or more GAP policy, and zero or more GSA TEK policies, where either one GSA KEK or GSA TEK payload MUST be present.

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

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

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

Policies following the GSA payload has common header

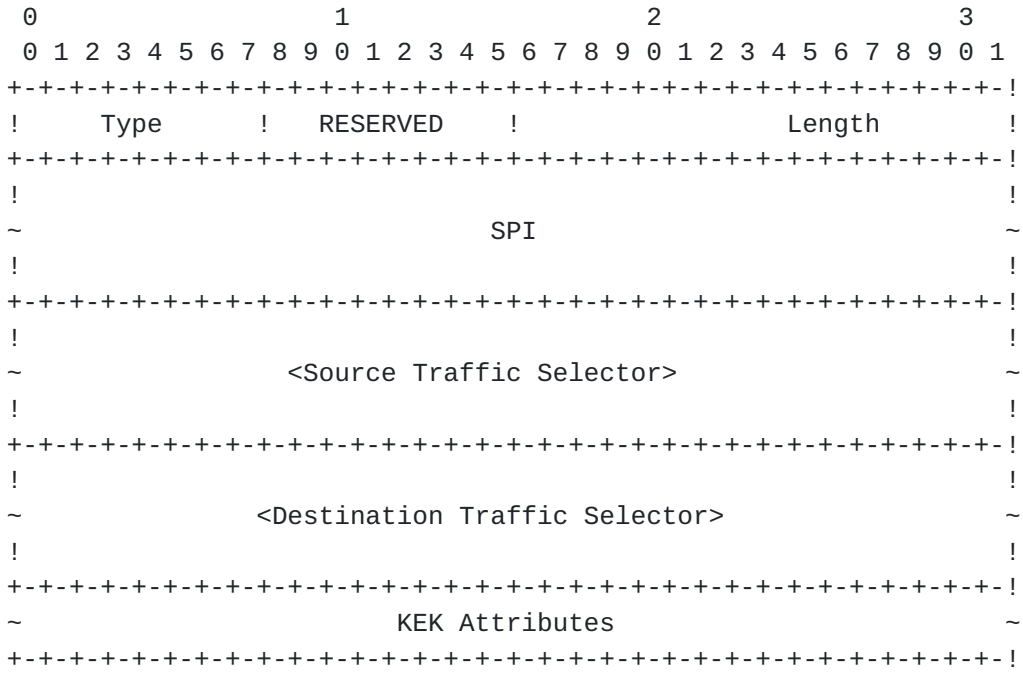


Type is defined as follow:

0 - RESERVED
1 - KEK
2 - GAP
3 - TEK
4-240 - RESERVED
241-255 - private and experimental

4.4. KEK Policy

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



The GSAK Payload fields are defined as follows:

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

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

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

- SPI (16 octets) -- Security Parameter Index for the KEK. The SPI must be the IKEv2 Header SPI pair where the first 8 octets become the "Initiator's SPI" field of the G-IKEv2 rekey message IKEv2 HDR, and the second 8 octets become the "Responder's SPI" in the same HDR. As described above, these SPIs are assigned by the GCKS.

- Source & Destination Traffic Selectors - Substructures describing the source and destination of the identities. These identities refer to the source and destination of the next KEK rekey SA.

Defined format and values are specified by IKEv2 [RFC5996], section 3.13.1.

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

4.4.1. KEK Attributes

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

ID Class	Value	Type
RESERVED	0	
KEK_MANAGEMENT_ALGORITHM	1	B
KEK_ENCR_ALGORITHM	2	B
KEK_KEY_LENGTH	3	B
KEK_KEY_LIFETIME	4	V
KEK_INTEGRITY_ALGORITHM	5	B
KEK_AUTH_METHOD	6	B
KEK_AUTH_ALGORITHM	7	B
KEK_MESSAGE_ID	8	B

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

Minimum attributes that must be sent as part of an GSA KEK: KEK_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_ALGORITHM (except for DSA based algorithms).

4.4.2. KEK_MANAGEMENT_ALGORITHM

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

KEK Management Type	Value
RESERVED	0

LKH	1
Standards Action	2-127
Private Use	128-255

4.4.3. KEK_ENCR_ALGORITHM

The KEK_ENCR_ALGORITHM class specifies the encryption algorithm using with the KEK. This is the same as IKEv2 encryption algorithm defined in [RFC5996] section 3.3.2. 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 the management.

4.4.4. KEK_KEY_LENGTH

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

The Group Controller/Key Server (GCKS) adds the KEK_KEY_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 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.

4.4.5. KEK_KEY_LIFETIME

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

4.4.6. KEK_INTEGRITY_ALGORITHM

KEK_INTEGRITY specifies the integrity algorithm. This integrity algorithm is specified in IKEv2 RFC 5996 section 3.3.2.

4.4.7. KEK_AUTH_METHOD

KEK_AUTH_METHOD specifies the method of authentication used. This is the same as IKEv2 Auth Method specified in IKEv2 RFC 5996 section 3.8

4.4.8. KEK_AUTH_ALGORITHM

KEK_AUTH_ALGORITHM specifies the hash algorithm uses to sign the AUTH payload as defined in IKEv2 [RFC5996] section 3.8 for RSA Digital Signature. The following tables define the algorithms for KEK_AUTHALGORITHM.

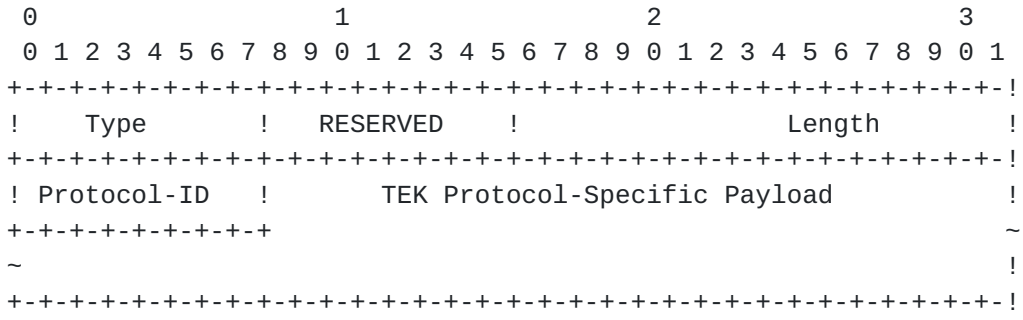
Algorithm Type	Value
RESERVED	0
AUTH_HASH_SHA256	1
AUTH_HASH_SHA384	2
AUTH_HASH_SHA512	3
Standards Action	4-127
Private Use	128-255

4.4.9. KEK_MESSAGE_ID

KEK_MESSAGE_ID define the start message id to be used by the GCKS in the GSA_REKEY message. Message ID is 4 octets unsigned integer.

4.5. GSA TEK Policy

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



The GSAT Payload fields are defined as follows:

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

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

- 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

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

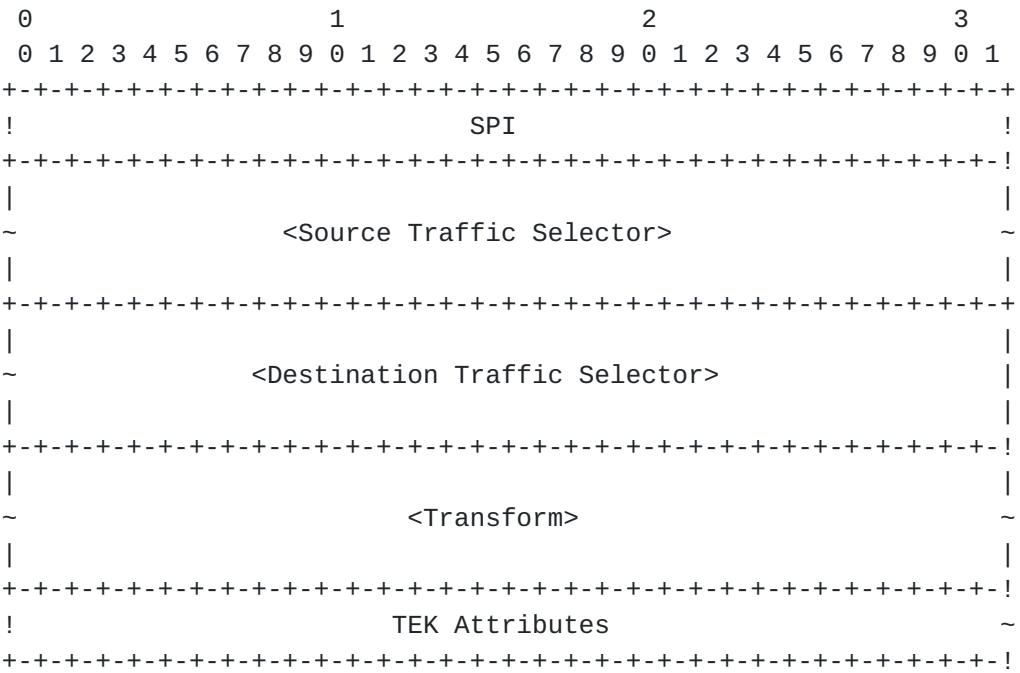
Support for the GSA_PROTO_IPSEC_AH GSA TEK is OPTIONAL.

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

4.5.1. TEK ESP and AH Protocol-Specific Policy

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

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



The GSAT Policy fields are defined as follows:

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

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

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

- TEK Attributes -- Contains TEK 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. [RFC5996], section 3.3.5.

Attribute Types

class	value	type
Life Duration	1	V

Specifies the time-to-live for the overall security association. When the TEK expires, all keys downloaded under the association (AH or ESP) must be re-rekeyed. The life duration attribute defines the actual length of the component lifetime in secondsthat can be protected.

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

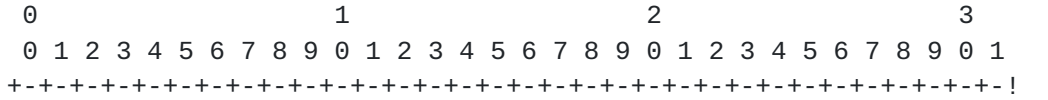
Transport mode	2	B

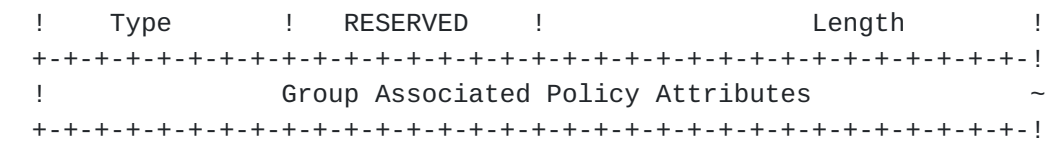
Specifies to use transport mode rather than tunnel mode

4.6. 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:





The GSA GAP payload fields are defined as follows:

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

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

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

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

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

4.6.1. ACTIVATION_TIME_DELAY/DEACTIVATION_TIME_DELAY

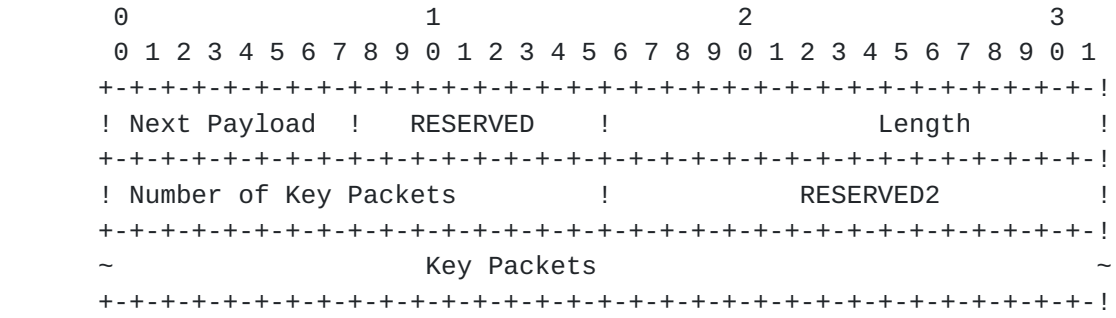
Section 4.2.1 of RFC 5374 specifies a key rollover method that requires two values be given it from the group key management protocol. The ACTIVATION_TIME_DELAY attribute allows a GCKS to set the Activation Time Delay (ATD) for SAs generated from TEKs. The ATD defines how long after receiving new SAs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

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

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

4.7. Key Download Payload

The Key Download Payload contains 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.



The Key Download Payload fields are defined as follows:

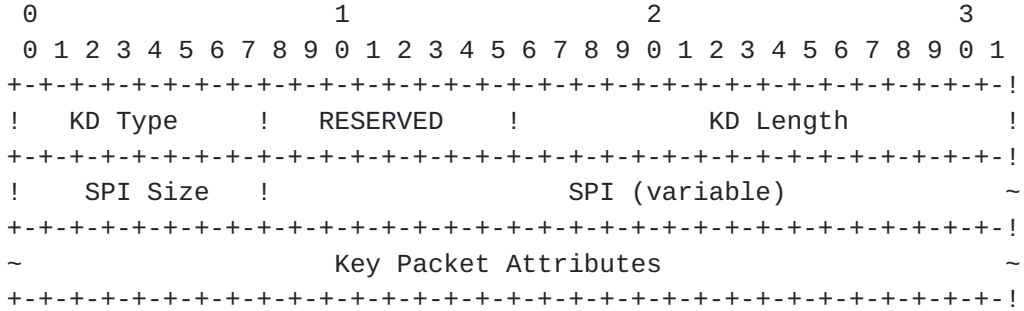
- Next Payload (1 octet) -- Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be zero.

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

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

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

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



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

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

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

- RESERVED (1 octet) -- Unused, set to zero.
- Key Download Length (2 octets) -- Length in octets of the Key Packet data, including the Key Packet header.
- SPI Size (1 octet) -- Value specifying the length in octets of the SPI as defined by the Protocol-Id.
- SPI (variable length) -- Security Parameter Index which matches a SPI previously sent in an GSAK or GSAT Payload.
- Key Packet Attributes (variable length) -- Contains Key information. The format of this field is specific to the value of the KD Type field. The following sections describe the format of each KD Type.

4.7.1. TEK Download Type

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

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

If no TEK key packets are included in a Registration KD payload, the group member can expect to receive the TEK as part of a Re-key SA.

At least one TEK must be included in each Re-key KD payload. Multiple TEKs may be included if multiple streams associated with the SA are to be rekeyed.

4.7.1.1. TEK_ALGORITHM_KEY

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

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

4.7.1.2. TEK_INTEGRITY_KEY

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

4.7.1.3. TEK_SOURCE_AUTH_KEY

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

4.7.2. KEK Download Type

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

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

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

4.7.2.1. KEK_ALGORITHM_KEY

The KEK_ALGORITHM_KEY class declares key for this SPI is contained in the Key Packet Attribute. The encryption and integrity algorithm that will use this key were specified in the GSAK payload. Then encryption and integrity keys can be derived using the following formula

{ SK_ai | SK_ar | SK_ei | SK_er } = prf+(KEK key, "G-IKEv2 REKEY" | SPIi | SPIr)

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

4.7.2.2. KEK_AUTH_KEY

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

4.7.3. LKH Download Type

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

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

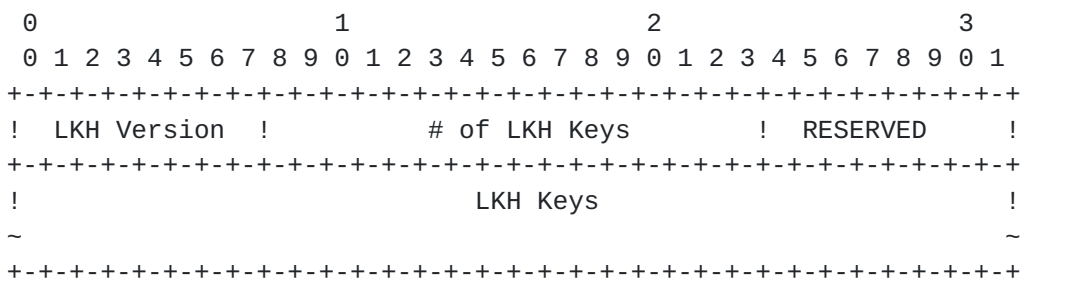
KEK Class	Value	Type
RESERVED	0	
LKH_DOWNLOAD_ARRAY	1	V
LKH_UPDATE_ARRAY	2	V
AUTH_ALGORITHM_KEY	3	V
Standards Action	4-127	
Private Use	128-255	

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

4.7.3.1. LKH_DOWNLOAD_ARRAY

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

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



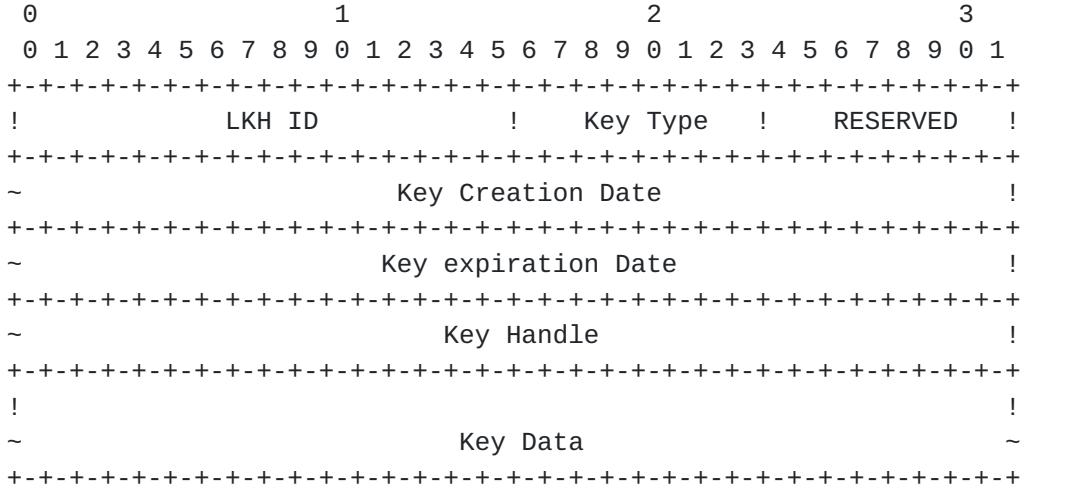
The KEK_LKH attribute fields are defined as follows:

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

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

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

Each LKH Key is defined as follows:



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

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

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

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

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

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

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

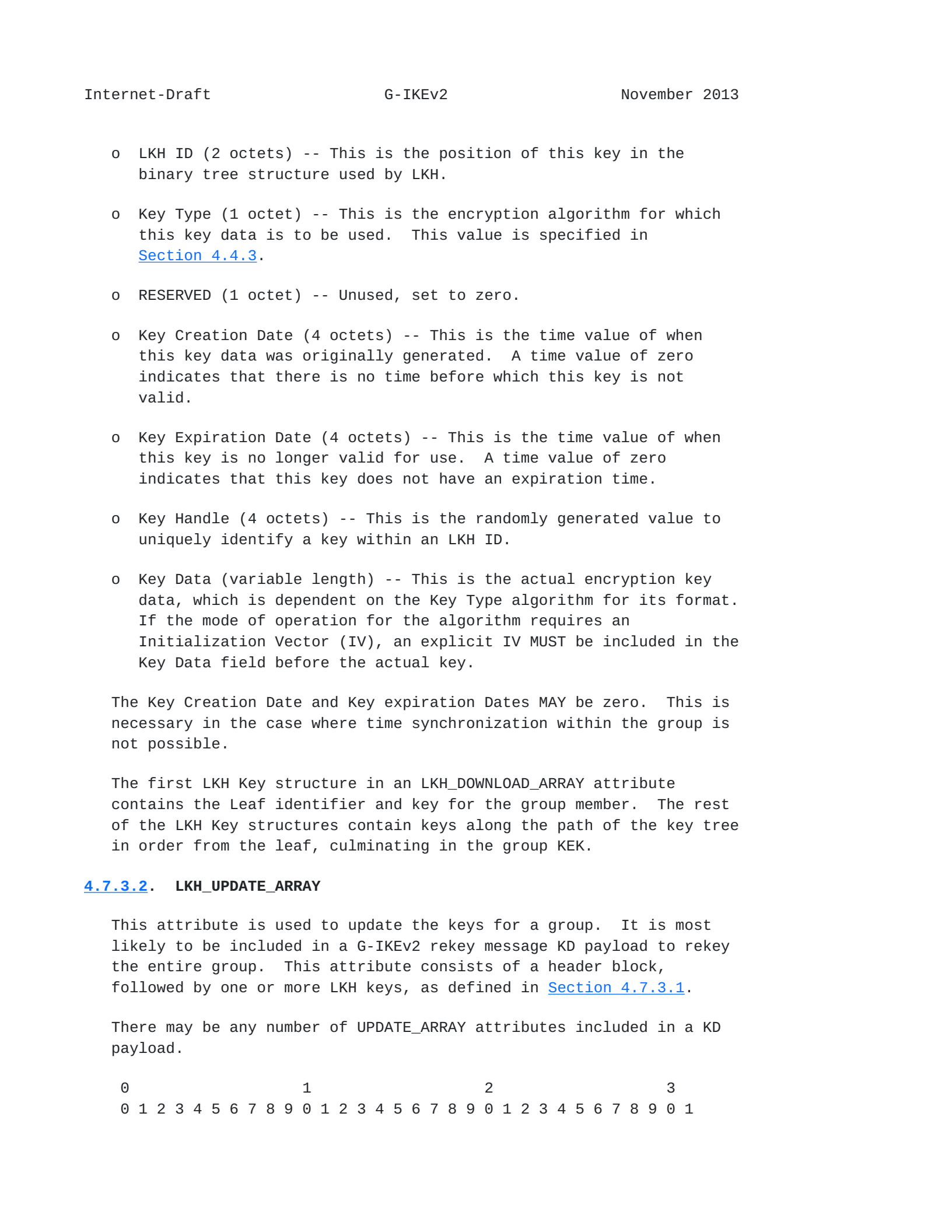
The Key Creation Date and Key expiration Dates MAY be zero. This is necessary in the case where time synchronization within the group is not possible.

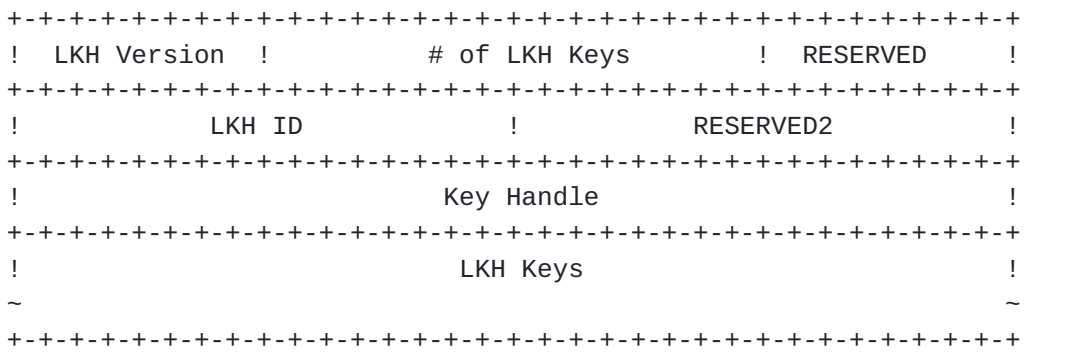
The first LKH Key structure in an LKH_DOWNLOAD_ARRAY attribute contains the Leaf identifier and key for the group member. The rest of the LKH Key structures contain keys along the path of the key tree in order from the leaf, culminating in the group KEK.

4.7.3.2. LKH_UPDATE_ARRAY

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

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





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

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

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

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

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

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

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

4.7.4. SID Download Type

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

KEK Class	Value	Type
RESERVED	0	
NUMBER_OF_SID_BITS	1	V
SID_VALUE	2	V
Standards Action	3-128	

Private Use	129-255
Unassigned	256-32767

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

4.7.4.1. NUMBER_OF_SID_BITS

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

4.7.4.2. SID_VALUE

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

4.7.4.3. GM Semantics

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

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

4.7.4.4. GCKS Semantics

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

4.8. Delete Payload

There are occasions the GCKS may want to signal to receivers to delete policy at the end of a broadcast, or if policy has changed. Deletion of keys MAY be accomplished by sending an IKEv2 Delete Payload, section 3.11 of [RFC5996] as part of the 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 1 for KEK SA, 2 for TEK AH or 3 for TEK ESP. Note that only one protocol id value can be defined in a Delete payload. If a TEK and a KEK SA must be deleted, they must be sent in different Delete payloads. Similarly, if a TEK specifying ESP and a TEK specifying AH need to be deleted, they must be sent in different Delete payloads.

4.9. Notify Payload

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

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

NOTIFY MESSAGES - ERROR TYPES	Value
INVALID_GROUP_ID -	TBD

Indicates the group id sent during registration process is invalid.

AUTHORIZATION_FAILED -	TBD

Sent in the response to GSA_AUTH message when authorization failed.

NOTIFY MESSAGES - STATUS TYPES	Value
SENDER_REQUEST_ID -	TBD

Sent in GSA_AUTH or GSA_REGISTRATION to request SIDs from GCKS. The data includes a count of how many SID values it desires.

4.10. Authentication Payload

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

5. Security Considerations

5.1. GSA registration and secure channel

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

5.2. GSA maintenance channel

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

5.2.1. Authentication/Authorization

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

5.2.2. Confidentiality

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

5.2.3. Man-in-the-Middle Attack Protection

GSA maintenance channel is integrity protected by using digital signature.

5.2.4. Replay/Reflection Attack Protection

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

6. IANA Considerations

6.1. New registries

A new set of registries are created for this draft.

KEK Attributes Registry, see Section 4.4.1

KEK Management Algorithm Registry, see Section 4.4.2

GSA TEK Payload Protocol ID Type Registry, see Section 4.5

TEK Attributes Registry, see Section 4.5

Key Download Type Registry, see Section 4.7

TEK Download Type Registry, see Section 4.7.1

KEK Download Type Registry, see Section 4.7.2

LKH Download Type Registry, see Section 4.7.3

SID Download Type Registry, see Section 4.7.4

6.2. New payload and exchange types to existing IKEv2 registry

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

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

The present document describes new IKEv2 Notify Payload types, see Section 4.9

6.3. Payload Types

The present document defines new IKEv2 Next Payload types. See Section 4.0 for the payloads defined in this document, including the Next Payload values defined by the IANA to identify these payloads.

6.4. New Name spaces

The present document describes many new name spaces for use in the G-IKEv2 payloads. Those may be found in subsections under Section 4.0. A new G-IKEv2 registry has been created for these name spaces.

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

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

7. Acknowledgements

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

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

8. References

8.1. Normative References

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

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

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

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

8.2. Informative References

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

[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>.

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

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

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

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

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

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

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

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

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

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

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

Appendix A. Differences between G-IKEv2 and RFC 6407

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

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

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

SEQ Payload - The SEQ payload is no longer needed since IKEv2 header has message id which is used to prevent message replay attacks.

Authors' Addresses

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

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

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

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

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

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

Yoav Nir
Check Point Software Technologies Ltd.
5 Hasolelim St.
Tel Aviv 67897
Israel

Email: ynir@checkpoint.com