TRILL Workgroup
Internet-Draft
Intended status: Standards Track
Expires: May 19, 2023

D. Eastlake
Futurewei Technologies
D. Zhang
Huawei Technologies

November 20, 2022

Simple Group Keying Protocol (SGKP)
<draft-ietf-trill-group-keying-11.txt>

Abstract

This document specifies a simple general group keying protocol that provides for the distribution of shared secret keys to group members and the management of such keys. It assumes that secure pairwise keys can be created between any two group members.

Status of This Memo

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

Distribution of this document is unlimited. Comments should be sent to the authors or the TRILL mailing list: trill@ietf.org.

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

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

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

Table of Contents

1. Introduction

This document specifies a simple general group keying protocol that provides for the distribution of shared secret keys to group members and for the management of such keys. It assumes that secure pairwise keys can be created between any two group members.

A companion document specifies two profiles for the use of this group keying protocol in a case using DTLS and a case using IPsec payload formats. It is anticipated that there will be other uses for this group keying protocol.

1.1 Terminology and Acronyms

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

This document uses the following terms and acronyms:

AES - Advanced Encryption Standard.

DTLS - Datagram Transport Level Security [RFC9147].

GKd - A distinguished station in a group that is in charge of which group key (Section 2) is in use.

GKs - Stations in a group other than GKd (Section 2).

IS-IS - Intermediate System to Intermediate System [RFC7176].

keying material - The set of a Key ID, a secret key, and a cypher suite.

QoS - Quality of Service.

RBridge - An alternative term for a TRILL switch.

TRILL - Transparent Interconnection of Lots of Links or Tunneled Routing in the Link Layer [RFC6325] [RFC7780].

TRILL switch - A device that implements the TRILL protocol [RFC6325] [RFC7780], sometimes referred to as an RBridge.

2. Simple Group Keying Protocol

This section gives an overview of the assumptions and capabilities of the Simple Group Keying Protocol (SGKP) that provides shared secret group keys. Further details of the messages used for this protocol are give in Section 3.

Any particular use of this protocol will require profiling giving further details and specifics for that use. For example, the envelope used for addressing and transmitting the messages of this protocol must be specified for any particular use. This protocol is not suitable for discovery messages but is intended for use between members of a group that have already established or can establish pair-wise security.

2.1 Assumptions

The following are assumed:

- All pairs of stations in the group can engage in pairwise communication with unicast messages and each can groupcast a message to the other group members.
- At any particular time, there is a distinguished station GKd in the group that is in charge of keying for the groupcast data messages to be sent to the group. The group wide shared secret keys established by GKd are referred to herein as "dynamic" keys.
- Pairwise keying has been negotiated between GKd and each other station GKs1, GKs2, ... GKsN in the group. These keys are referred to in this protocol as "pairwise" keys.
- There are one or more keys, other than the dynamic or pairwise keys, which are already in place at all group member stations and may be present at other stations. These are referred to as "stable" keys.

When keying material is stored by a station, it is accompanied by a "use flag" indicating whether or not that keying material is usable for groupcast transmissions.

2.2 Group Keying Procedure Overview

GKd sends unicast keying messages to the other stations in the group and they respond as specified below and in further detail in the particular use profiles of SGKP. All such keying messages MUST be encrypted and authenticated using the pairwise keys as further specified in the use profile.

Typically, GKd sends a keying message to each GKs with keying material. After successful acknowledgement of receipt from each GKs, GKd sends a keying message to each GKs instructing it to use the dynamic key GKd has set. It would be common for GKd to set a new dynamic key at each GKs while an older dynamic key is in use so that GKd can more promptly roll over to the new dynamic key when appropriate.

To avoid an indefinite build up of keying material at a GKs, keys have a lifetime specified by GKd and GKd can send a message deleting a key. (GKd can also send a message indicating that a key is no longer to be used but leaving it set.) Should the space available at a GKs for keying material be exhausted, on receipt of a Set Key keying message from GKd for a new key ID, GKs discards a dynamic key it has and originates a Delete Key message to the source of that dynamic key.

2.3 Transmission and Receipt of Group Data Messages

If a group has only one member, transmission of data between group members is a moot question and any messages that would be so transmitted if the group had more members are discarded.

If a group has only two members, then pairwise security is used between them.

When a group has more than two members and a station in the group transmits a data message to the group, if the transmitter has one or more keys set by GKd that it has been instructed to use, it uses one of those keys and its associated cypher suite to secure the data message and then groupcasts it to all other membes of the group. If it has no such key, then it uses serial unicast to send the data message to each other member of the group, negotiating pairwise keys with them if it does not already have such pairwise keys. Thus, it is a responsibility of GKd not to authorize the use of a groupcast key until it knows that all the GKs have that key.

When a station in the group receives data that has been groupcast to the group, if the receiver has the key referenced by the data message the receiver decrypts and verifies it. If verification fails or if the receiver does not have the required key, the receiver discards the data message. Thus whether GKs has been directed to "use" a key by GKd is relevant only to transmission, not reception.

2.4 Changes in Group Membership or GKd

When a new station joins the group, GKd SHOULD send that station the currently in-use group key and instruct it to use that key and MAY send it other keys known to the group members and intended for future use.

If GKd detects that one or more stations that were members of the group are no longer members of the group, it SHOULD generate and distribute a new group key to the remaining group members, instruct them to use this new key, and delete from them any old keys known to the departed group member station(s) or at least instructing them to dis-use such old keys that are marked for use; however, in the case of groups with large and/or highly dynamic membership, where a station might frequently leave and then rejoin, it may, as a practical matter, be necessary to rekey less frequently.

A new group member can become GKd due to the previous GKd leaving the group or a configuration change or the like. A GKs MUST NOT use keying material for transmission that was set by a station that it determines is not GKd. To avoid a gap in service, a station that is not GKd MAY set keying material at other stations in the group; however, such a non-GKd station cannot set the use flag for any such keying material. It is RECOMENDED that the second highest priority station to be GKd set such keying material at all other stations in the group. Should a station run out of room for keying material, it SHOULD discard keying material set by a station with lower priority to be GKd before discarding keying material set by a higher priority station and among keys set by GKd is SHOULD discard the lest recently used first.

3. Group Keying Messages

Keying messages start with a Version number. This document specifies Version zero.

Keying messages are structured, as shown in Figure 3.1 below, as

- a Version number,
- a Response flag,
- a Key ID length,
- the Key ID of a stable key,
- a group keying use profile identifier,
- possible padding,
- a key wrap algorithm specifier, and finally
- a key wrapped vector of additional fields wrapped using a key derived from the stable key identified.

Keying messages are always sent unicast and encrypted and authenticated with the appropriate pairwise key, all as further specified for the particular use profile. It will typically be possible for GKd to calculate the keying message once, including the wrapping under a key derived from the stable key, then send that message to various GKs using the different pairwise keys for each GKs.

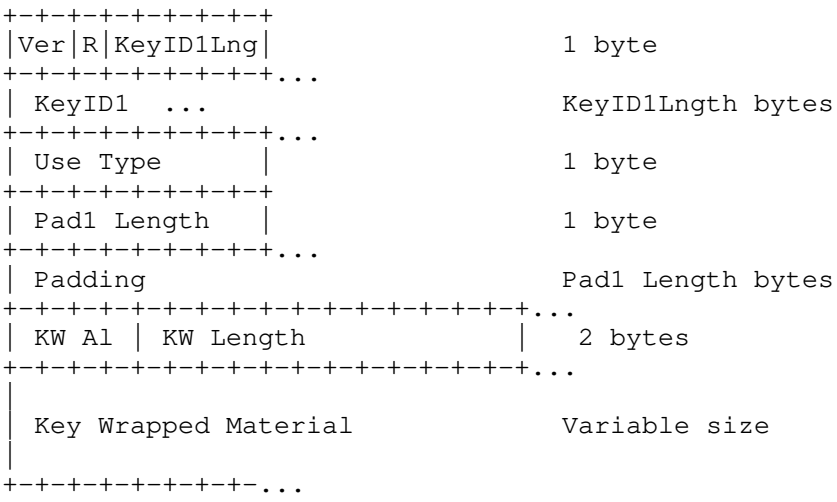

Figure 3.1. Keying Message Structure

The fields in Figure 3.1 are as follows:

Ver - Group Keying protocol version. This document specifies version zero.

R - Response flag. If set to one, indicates a response message. If set to zero, indicated a request or no-op message.

KeyID1Lngth, KeyID1 - KeyID1 identifies the stable key wrapping key (also known as the Key Encrypting Key (KEK)) as further specified in the use profile. KeyID1Lngth is a 5-bit field that gives the length of KeyID1 in bytes minus 1 as an unsigned integer.

Use Type - Specifies the particular group security use profile such as one of the two profiles in [SGKPuses]. See Section 5, Item 3.

Pad1 Length, Pad1 - Padding to obscure the non-padded message size. Pad1 Length may be from 0 to 255 and gives the length of the padding as an unsigned integer. Each byte of padding MUST be equal to Pad1 Length. For example, 3 bytes of padding with length is 0x03030303.

KW Algorithm - An unsigned integer 4-bit field specifying the Key Wrap Algorithm. See Section 5, Item 4.

KW Length - An unsigned integer 12-bit field that gives the length of the Key Wrapped Material in octets as an unsigned integer.

Key Wrapped Material - The output of the designated Key Wrapping Algorithm on the message vector of fields using the designated stable key.

The vector of fields contained within the key wrapping is specified for the various keying messages in subsections below. The contents of this wrapped vector are protected by the key wrapping as well as being authenticated and super-encrypted by the pairwise keyed security used for sending the overall keying message. The probability that the stable key used for key wrapping is the same as the outer message pairwise key MUST be insignificant (less that 1 in 2**64).

Each group keying message contains, in the key wrapped vector of fields, a message type and a message ID set by the sender of a request. These fields are returned in the corresponding response to assist in the matching of response to requests, except that there is no response required to the No-Op message.

If no response is received to a request (other than a No-Op request) for an amount of time configurable in milliseconds from 1 to (2**15 - 1), the request is re-transmitted with the same message ID. These retries can occur up to a configurable number of times from 1 to 8. Unless otherwise provided in the particular use profile, the default response delay threshold is 200 milliseconds and the default maximum

number of retries is 3.

Keying messages are sent with a priority/QoS configurable on a per device per use type basis. The default priority/QoS is specified in the use profile.

Since the minimum length of the Key Wrapped Material is 16 bytes, the minimum valid length of a keying message before pairwise security is 22 bytes, even if KeyID1 Length and Pad1 Length are zero. All multi-byte fields are in network order, that is, with the most significant byte first. The maximum valid length before pairwise security is 6 (fixed bytes) + 32 (max KeyID1) + 255 (max padding) + 4095 (max KW material) = 4388 bytes. However, the Key Wrapped Material is usually quite compact. To reduce possible problems with MTU, fragmentation, etc., keying messages SHOULD NOT exceed 1000 bytes in length.

3.1 Set Key Message

The structure of the wrapped vector of fields for the Set Key keying message is as show in Figure 3.2. A recipient automatically determines the overall length provided for this vector of fields inside the key wrapping as a byproduct of the process of key unwrapping.

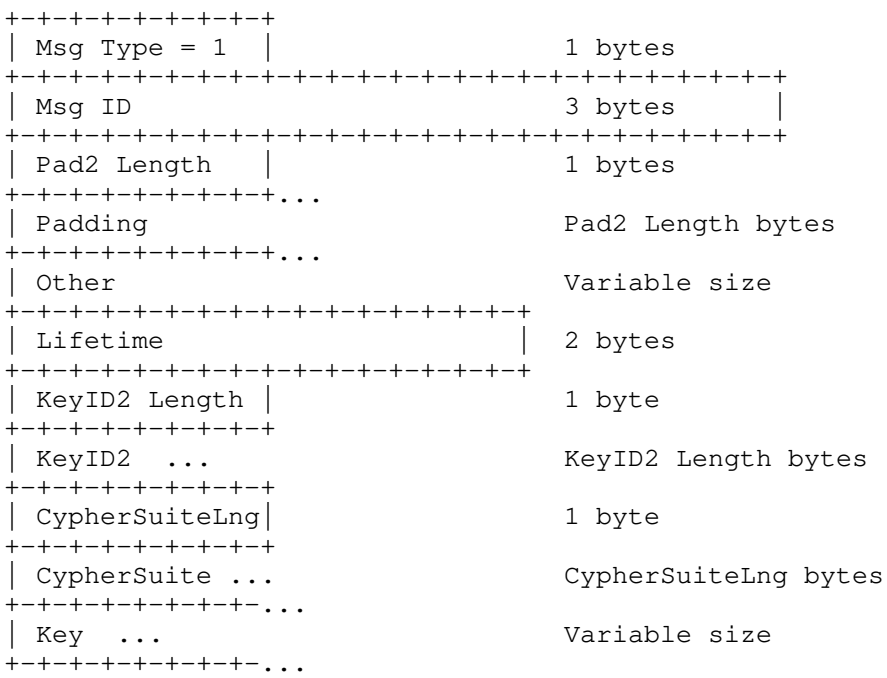

Figure 3.2. Set Key Message Inner Structure

The fields are as follows:

Msg Type = 1 for Set Key message

Msg ID - A 3 byte quantity to be included in the corresponding response message to assist in matching requests and responses. Msg ID zero has a special meaning in responses and MUST NOT be used in a Set Key message or any other group keying request message.

Pad2 Length, Pad2 - Padding to obscure the size of the unpadded wrapped data. Pad2 Length may be from 0 to 255 and gives the length of the padding as an unsigned integer. Each byte of padding MUST be equal to Pad1 Length. For example, 2 bytes of padding with length byte is 0x020202.

Other - Additional information if specified in the use profile. If Other information in this message is not mentioned in the use profile, there is none and this portion of the wrapped information is null. If a use profile specifies Other information it must be possible to determine its length so that following fields can be properly parsed and so that the size of the Key field can be deduced; for example, Other could begin with a length byte.

Lifetime - A 2-byte unsigned integer. After that number of seconds plus one second, the key and associated information being set MUST be discarded. Unless otherwise specified for a particular use profile of this group keying protocol, the default Lifetime is 10,000 seconds or about 2 hours and 46 minutes.

KeyID2 Length, KeyID2 - KeyID2 identifies the group key and associated information being set as further specified in the use profile. KeyID2 Length is an unsigned byte that gives the length of KeyID2 in bytes.

CypherSuiteLng, CypherSuite - CypherSuite identifies the cypher suite associated with the key being set as further specified in the use profile. CypherSuite Length is an unsigned byte the gives the length of CypherSuite in bytes.

Key - This is the actually group shared secret keying material being set. Its length is deduced from the overall length of the vector of fields (found by the key unwrap operation) and the length of the preceding fields.

Keying material and associated cypher suite are indexed under the Key ID and the identity of the station that sent the information. This identity is normally the address of that station as specified in the use profile.

If GKs already has a dynamic key set under KeyID2, the key's value and associated cypher suite are compared with those in the Set Key messages. If they are the same, the only receiver action is to update the Lifetime information associated with KeyID2 and send a Response message. If they are different, the lifetime, cypher suite, and key (and possibly Other material) are replaced, the use flag is cleared, and a Response message sent.

3.2 Use, Delete, Disuse, or Deleted Key Messages

The structure of the wrapped material for the Use Key, Delete Key, Disuse Key, and Deleted Key keying messages are the same as each other except for the message type and are shown in Figure 3.3

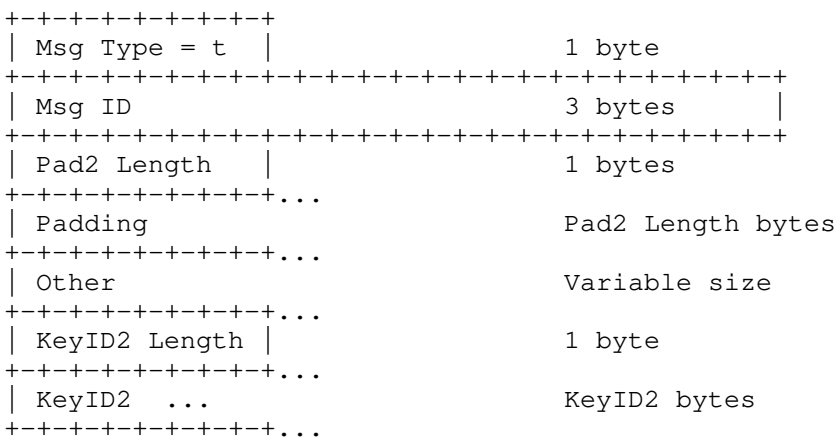

Figure 3.3. Use, Delete, Disuse, or Deleted Key Message

The Msg Type field specifies the particular message as follows:

Msg Type	Message
2	Use Key
3	Delete Key
4	Disuse Key
5	Deleted Key

The remaining fields are as specified in Section 3.1. KeyID2 indicates the key to be used, deleted, for which use should cease, or which has been deleted, depending on the message type.

It is RECOMMENDED that these messages be padded so as to be the same length as a typical Set Key message.

The Delete Key is sent by a station believing itself to be GKd instructing some GKs to delete a key. When a GKs spontaneously deletes a key, it sends a Deleted Key message to the station from which it received the key. The message types for Delete Key and Deleted Key are different to minimize confusion in corner cases such as the GKd changing while messages are in flight. The Msg ID used in a Deleted Key message is created by the sending GKs from a space of Msg IDs associated with that GKs, which space is independent of the Msg IDs used in requests originated by GKd.

3.3 Response Message

The structure of the wrapped material for the Response group keying message is as show below in Figure 3.4. A response message is

indicated by the R bit in the first byte of the message outside the key wrapping.

A response MUST NOT be sent due to the receipt of a response. The R bit is outside of the key wrapping so that this rule can be enforced even in case unwrapping is not possible.

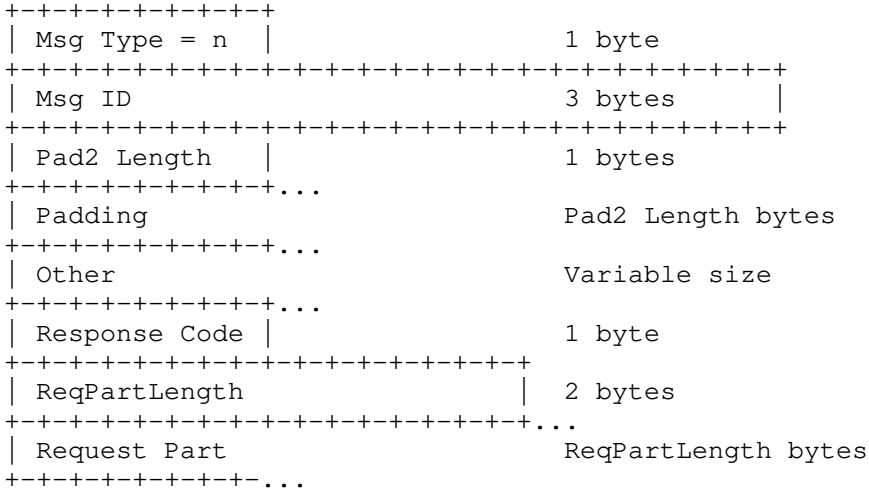

Figure 3.4. Response Message Inner Structure

Except as specified below, the fields are as specified for the Key Set message in Section 3.1.

Msg Type, Msg ID - The content of these field is copied from the message in reply to which this Response message is sent unless there is an error that stops the replying station from determining them; in that case the special value zero is used for the Msg Type and Msg ID. Errors where the Msg Type and ID could not be determined are indicated by a Response Code with its high order bit set to one, that is, the 0b1xxxxxxx bit set.

Response Code - An unsigned byte giving the response as enumerated in in Section 3.3.1. Any Response Code other than a success indicates that the receiver took no action on the request other than sending an error Response message.

ReqPartLength, Request Part: It is usually usefully to include some or all of the request message in error responses.

- If the Response Code high order two bits are zero, the request succeeded and ReqPartLength MUST be set to zero so Request Part will be null.

- If the Response Code high order two bits are zero one (0b01xxxxxx), then there was an error in the part of the request inside the key wrapping but the unwrap process was successful. ReqPartLength is the length of the request message material included in the Request Part field. The included request material is from the unwrapped vector of fields started with the Msg Type byte.
- If the Response Code high order bit is one (the 0b1xxxxxxx is set), then there was an error parsing the material outside the key wrap or an error in the unwrapping process. ReqPartLength is the length of the request message part included in the Request Part field. The included part of the request starts with the first byte of the message (the byte containing the version, response flag, and KeyID1 Length). The key wrapped material in the response message will still be wrapped.

3.3.1 Response Codes

The high order two bits of the Response Code have meaning as shown in Table 3.1.

Top 2 Bits	Category
0b00	Success
0b01	key wrap contents
0b10/11	Outside of key wrap contents

Figure 3.1 Categories of Response Codes

Response Decimal	Response Hex	Meaning
0	0x00	Success
1	0x01	Success and the key at an existing key ID was changed
2-47	0x02-0x2F	Unassigned
48-63	0x30-0x3F	Reserved for special success codes defined in use profiles
64	0x40	Malformed inner fields (see Note 2 below)
65	0x41	Unknown or zero Msg Type in a request
66	0x42	Zero Msg ID in a request
68	0x43	Invalid length KeyID2
69	0x44	Unknown KeyID2
70	0x45	Invalid length CypherSuite
71	0x46	Unknown CyperSuite

72	0x47	Bad Key (see Note 3 below)
73-111	0x49-0x6F	Unassigned
112-127	0x70-0x7F	Reserved for error codes defined in use profiles and related to the key wrapped contents
128	0x80	Malformed message (see Note 1 below)
129	0x81	Invalid length KeyID1
130	0x82	Unknown KeyID1
131	0x83	Unknown Use Type
131	0x84	Key unwrap fails test for constant (e.g., AES test 1, see Section 3 [RFC5649]).
132	0x85	Key unwrap fails message length versus wrapped size test (e.g., AES test 2, see Section 3 [RFC5649]).
133	0x86	Key unwrap fails test for value of padding (e.g., AES test3, see Section 3 [RFC5649]).
134-175	0x86-0x7F	Unassigned
176-191	0xB0-0xBF	Reserved for error codes defined in use profiles and related to parts of message outside the key wrap contents
192	0xC0	No keys set
193	0xC1	Referenced key unknown
194	0xC2	Referenced key known but use flag not set
195-255	0xC3-0xFF	Reserved

Response Code Notes:

Note 1 Message is too short or too long, key wrapped material is too short, Padding bytes are not the required value, or similar fundamental message format problems.

Note 2 The key wrapped inner vector of fields is too short or too long, Padding bytes are not the required value, or similar fundamental vector of fields format problems.

Note 3 Key is not a valid length for CypherSuite or other internal checks on key (for example, parity bits in a 64 bit DES key (not that you should be using DES)) fail when they should be correct.

Figure 3.2 Response Codes

3.4 No-Op Message

The No-Op message is a dummy message intended for use in disguising metadata deducible from keying message transmissions. It requires no

response although a recipient can always decide to send a No-Op message to a station from which it has received such a message. The vector of fields inside the key wrap is as follows:

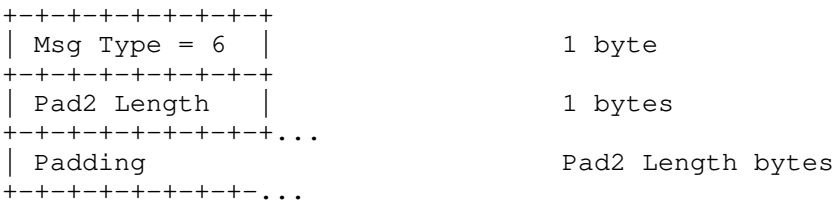

Figure 3.5. No-Op Message Inner Structure

The Msg Type is set to 6 to indicate a No-Op message.

Pad2 Length and Padding are as specified in Section 3.1. It is RECOMMENDED that Pad2 Length in a No-Op message be such as to make its length the same as the length of a typical Set Key message.

4. Security Considerations

This section gives some general security considerations of this group keying protocol as distinguished from security considerations of a particular use profile.

The method by which the stations in the group discover each other is specified in the group keying use profile. GKd controls group access and generally learns whatever it needs to know about GKs during the pairwise authentication and pairwise keying process.

The group keying provided by this protocol is shared secret keying. This means that data messages can only be authenticated as coming from some group member but not as coming from a specific group member. If this level of authentication is insufficient, GKd can simply not set keys or not set them as usable. This will force all stations in the group that are configured to use security for multi-destination transmissions to the group to serially unicast data to the other group members using pairwise keying.

The content value of padding fields in the Group Keying protocol is fixed so that it cannot be used as a covert channel. It might still be possible to use the length of padding as a covert channel.

5. IANA Considerations

IANA is requested to perform the following actions:

1. Establish a protocol parameters web page for "Group Keying Protocol Parameters" with the initial registries on that page as specified below in this section.

2. Establish a "Message Type" registry on the Group Keying Protocol Parameters page as follows:

   Name: Message Types
   Registration Procedure: IETF Review
   Reference: [this document]

Type	Description	Reference
0	Reserved	[This document]
1	Set Key	[This document]
2	Use Key	[This document]
3	Delete Key	[This document]
4	Disuse Key	[This document]
5	Deleted Key	[This document]
6	No-Op	[This document]
7-250	Unassigned	
251-254	Private Use	[This document]
255	Reserved	[This document]

3. Establish a "Group Keying Use Profile" registry on the Group Keying Protocol Parameters page as follows:

   Name: Group Keying Use Profiles
   Registration Procedure: IETF Review
   Reference: [This document]

Profile	Description	Reference(s)
0	Reserved	[This document]
1	Extended RBridge Channel	[SGKPuses]
2	TRILL over IP	[SGKPuses]
3-250	Unassigned	
251-254	Private Use	[This document]
255	Reserved	[This document]

4. Establish a "Key Wrap Algorithm" registry on the Group Keying Protocol Parameters page as follows:

Name: Key Wrap Algorithms
Registration Procedure: IETF Review
Reference: [This document]

Code	Algorithm	References
0	-	Reserved
1	AES	[This document][RFC5649]
2	ChaCha	[This document][ChaChaKW]
3-16	-	Reserved

5. Establish a "Response Code" registry on the Group Keying Protocol Parameters page as show below taking entries from the Response Code table in Section 3.3.1 above. In the table of values, the Reference column should be "[This document]" except where the Meaning is "Unassigned" or "Reserved".

Name: Response Codes
Registration Procedure: IETF Review
Reference: [This document]
Note: The top two bits of the Response Code indicate a category as specified in Section 3.3.1 of [this document].

Response Decimal	Response Hex	Meaning	Reference
0	0x00	Success	[this document]
...	...	...	
255	0xFF	Reserved	

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>.

[RFC5649] - Housley, R. and M. Dworkin, "Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm", RFC 5649, DOI 10.17487/RFC5649, September 2009, <https://www.rfc-editor.org/info/rfc5649>.

[RFC6325] - Perlman, R., Eastlake 3rd, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, DOI 10.17487/RFC6325, July 2011, <https://www.rfc-editor.org/info/rfc6325>.

[RFC7176] - Eastlake 3rd, D., Senevirathne, T., Ghanwani, A., Dutt, D., and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC 7176, May 2014, <https://www.rfc-editor.org/info/rfc7176>.

[RFC7780] - Eastlake 3rd, D., Zhang, M., Perlman, R., Banerjee, A., Ghanwani, A., and S. Gupta, "Transparent Interconnection of Lots of Links (TRILL): Clarifications, Corrections, and Updates", RFC 7780, DOI 10.17487/RFC7780, February 2016, <https://www.rfc-editor.org/info/rfc7780>.

[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>.

[RFC9147] - Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", RFC 9147, DOI 10.17487/RFC9147, April 2022, <https://www.rfc-editor.org/info/rfc9147>.

[SGKPuses] - D. Eastlake, D. Zhang, "Simple Group Keying Protocol TRILL Use Profiles", draft-ietf-trill-link-gk-profiles, work in progress.

[ChaChaKW] - D. Eastlake, "CHA CHA 20 Key Wrap with Padding Algorithm", draft-eastlake-chacha20-key-wrap, work in progress.

Informative References

None.

Acknowledgements

The contributions of the following are hereby gratefully acknowledged:

TBD

Authors' Addresses

Donald E. Eastlake, 3rd
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL 32703 USA

Phone: +1-508-333-2270
EMail: d3e3e3@gmail.com

Dacheng Zhang
Huawei Technologies

Email: dacheng.zhang@huawei.com