

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 20, 2014

M. Jethanandani
Ciena Corporation
B. Weis
K. Patel
Cisco Systems
D. Zhang
Huawei
S. Hartman
Painless Security
U. Chunduri
A. Tian
Ericsson Inc.
J. Touch
USC/ISI
November 16, 2013

Negotiation for Keying Pairwise Routing Protocols in IKEv2
draft-mahesh-karp-rkmp-05

Abstract

This document describes a mechanism to secure the routing protocols which use unicast to transport their signaling messages. Most of such routing protocols are TCP-based (e.g., BGP and LDP), and the TCP Authentication Option (TCP-AO) is primarily employed for securing the signaling messages of these routing protocols. There are also two exceptions: BFD which is over UDP or MPLS, and RSVP-TE which is over IP (but employs an integrated approach to protecting the signaling messages instead of using IPsec). The proposed mechanism secures pairwise TCP-based Routing Protocol (RP) associations, BFD associations and RSVP-TE associations using the IKEv2 Key Management Protocol (KMP) integrated with TCP-AO, BFD, and RSVP-TE respectively. Included are extensions to IKEv2 and its Security Associations to enable its key negotiation to support TCP-AO, BFD, and RSVP-TE.

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](#)].

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 20, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Terminologies	3
1.2.	Acronyms and Abbreviations	4
2.	Overview	4
2.1.	Types of Keys	5
3.	Protocol Exchanges	6
3.1.	IKE_SA_INIT	6
3.2.	IKE_AUTH	7
3.3.	CREATE_CHILD_SA	7
3.4.	INFORMATIONAL	8
4.	Operation Details	8
4.1.	General	9
4.2.	Initial Key Specific Data Exchange	9
4.3.	Key Selection, Rollover and Protocol Interaction	10
5.	Key Management Database	10
6.	Header and Payload Formats	10
6.1.	Header and Payload Formats for TCP-A0	11
6.1.1.	Security Association Payload for TCP-A0	11
6.1.1.1.	Transforms Substructures for TCP-A0	11

6.1.1.2.	Example Proposal Exchange	13
6.1.2.	Derivation of TCP-A0 Keying Material	13
6.2.	Security Association Payload for BFD	14
6.2.1.	Transforms Substructures for BFD Authentication . . .	14
6.3.	Security Association Payload for RSVP-TE	15
6.3.1.	Transforms Substructures for RSVP-TE Authentication .	15
6.4.	Notify and Delete Payloads	16
7.	IANA Considerations	16
8.	Security Considerations	17
9.	Acknowledgements	17
10.	References	17
10.1.	Normative References	17
10.2.	Informative References	18
Authors' Addresses	19

[1.](#) Introduction

Existing routing protocols using unicast pairwise communication model (e.g., BGP, LDP, RSVP-TE, and BFD) have cryptographic authentication mechanisms that use a key shared between the network devices (devices for short) on the both sides of the model to protect routing message exchanges between endpoints. The unicast key management for these protocols today is limited to statically configured master keys in individual network devices. This document defines a mechanism to secure such pairwise Routing Protocol (RP) associations using IKEv2 [[RFC5996](#)], allowing network devices to automatically exchange keying material related information between the network devices. To benefit the discussion, it is implied that the routing protocols mentioned in the remainder of this memo use unicast pair-wise communication model, unless otherwise mentioned.

This memo assumes that network devices need to be provisioned with some credentials for a one-to-one authentication protocol. Any method for a pairwise security protocol specified for use with IKEv2 is applicable.

When two network devices running a routing protocol have not yet established a secure association, the two endpoints need to select a KMP solution that meets their mutual requirements and use that KMP solution to establish the required security before sending out any routing protocol packets. The KMP solution typically enables the network devices to perform mutual authentication using their provisioned credentials and to agree upon certain keying material as the result of an successful authentication. The keying material then can be applied to secure the routing protocol.

[1.1.](#) Terminologies

This section lists the key terminologies used throughout the memo.

Network Device: In this memo, a router or any other type of device participating in routing protocols is referred to as a network device.

Key Management Database (KDDB): A KDDB is a conceptual database which locates in the middle of a key management protocol and a routing protocol to provide the long-term key management service. Therefore, the RP and the KMP need not to cooperate directly.

1.2. Acronyms and Abbreviations

The following acronyms and abbreviations are used throughout this memo.

IKEv2 Internet Key Exchange Protocol Version 2

RP Routing Protocol

SA Security Association

KMP Key Management Protocol

2. Overview

As illustrated in Figure 1, this work makes use the state machine of IKEv2. Assume a network device and its peer device are in State 1. That is, the device has not authenticated its peer device and does not have the keys to secure the routing protocol packets which it would like to exchange with the peer. Before sending any routing protocol packets, the two devices need to perform a IKE_SA_INIT exchange. If the IKE_SA_INIT exchange succeeds, both network devices are transferred to State 2 where they have agreed upon certain keying material but have decided how to use the material to derive keys to secure routing protocols. To achieve this objective, the two network devices perform an IKE_AUTH exchange, in which both endpoints try to authenticate each other and generate security associations for the routing protocol they intend to support. If the IKE_AUTH exchange succeeds, the network devices transfer their state to State 3 where both endpoints are authenticated and keys for securing the routing protocols are generated. If the endpoints intend to generate new SAs for routing protocols by using the keying material already generated, they can just perform an CREATE_CHILD_SA exchange. A discussion in more details can be found in [Section 4](#).

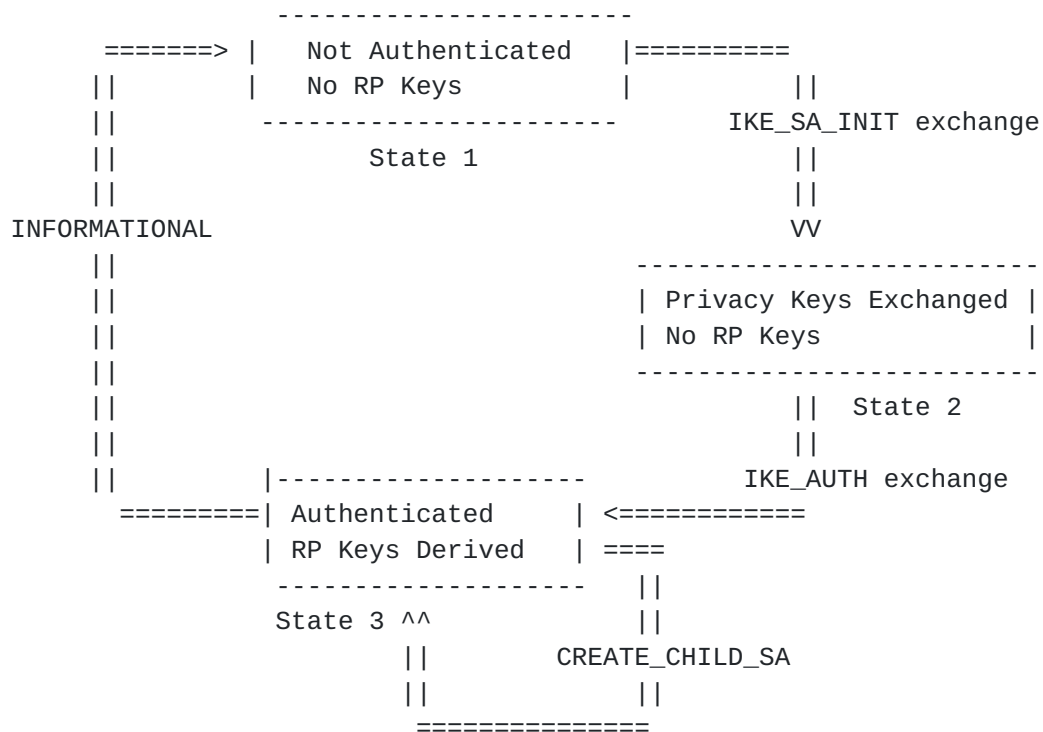


Figure 1: State Diagram

2.1. Types of Keys

Three types of keys mentioned the discussion of this memo are listed as follows:

- o PSK (Pre-Shared Key) : a PSK is a pair-wise unique key, which can be used for securing the routing protocol exchanges or be used for authenticating a network device by a KMP. These keys are configured by some mechanism such as manual configuration or a management application outside of the scope of KMP.
- o Protocol master key: A protocol master key is a key exported by a KMP for use by a routing protocol. This is the key that is shared in the KMDB between the routing protocol and KMP. A routing protocol may use a protocol master key directly or derive traffic keys from it.
- o Traffic key: A traffic key is the key actually used to protect the integrity of the routing messages exchanged in a routing protocol. In existing cryptographic authentication mechanisms for routing protocols, the traffic key can be the same as or derived from the protocol master key. If there is no KMP provided, a traffic key can be the same as or derived from a pre-shared key.

3. Protocol Exchanges

The KARP analysis in BGP, LDP, PCEP, and MSDP indicates that all of these routing protocols need a dedicated key management protocol[RFC6952] to confidentially exchange keying material between endpoints. There is no need to define an entirely new protocol for this purpose, when existing mature protocol exchanges and methods have been vetted. This draft makes use of the IKEv2 protocol exchanges, state machine, and policy definitions to define a dedicated key management protocol.

The notations contained in the IKEv2 message are defined as follows.

+-----+-----+	
Notation	Payload
+-----+-----+	
AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
D	Delete
HDR	IKEv2 Header (not a payload)
IDi	Identification - Initiator
IDr	Identification - Responder
KE	Key Exchange
Ni, Nr	Nonce
N	Notify
SA	Security Association
SK	Encrypted and Authenticated
TSi	Traffic Selector - Initiator
TSr	Traffic Selector - Responder
+-----+-----+	

Acronyms Used in Protocol Exchange

3.1. IKE_SA_INIT

A network device desiring to negotiate a key and other associated parameters for a pair-wise routing protocol to a peer initiates an IKE_SA_INIT exchange defined in IKEv2 [RFC5996]. The IKE_SA_INIT exchange is a two-message exchange that allows the network devices to negotiate cryptographic algorithms, exchange nonce information, and do a Diffie-Hellman (DH) [DH] exchange, for their routing protocols, after which protocols on these network devices can communicate privately. Note that at the end of a IKE_SA_INIT exchange the endpoints on the both sides have not authenticated each other yet. For the details of this exchange, refer to IKE_SA_INIT in IKEv2 [RFC5996].


```

Peer (Initiator)                                Peer (Responder)
-----
HDR, SAI1, KEi, Ni    -->
<--                  HDR, SAR1, KEr, Nr, [CERTREQ,]

                        IKE_SA_INIT

```

Up to this step, this work introduces no change to IKEv2.

3.2. IKE_AUTH

Next, the network devices perform an IKE_AUTH exchange defined in IKEv2 [[RFC5996](#)]. The SA payloads contain the security policies for a key and the associated parameters (as defined in Header and Payload Formats ([Section 6](#))), and the TS payloads contains traffic selectors as defined in IKEv2 [[RFC5996](#)]. For the details of the exchange please refer to IKE_AUTH in IKEv2 [[RFC5996](#)].

```

Peer (Initiator)                                Peer (Responder)
-----
HDR, SK {IDi, [CERT,] [CERTREQ,]
[IDr,] AUTH, SAI2, TSi, TSr}    -->
<--                  HDR, SK {IDr, [CERT,] AUTH,
                        SAR2, TSi, TSr}

                        IKE_AUTH

```

In the IKE_AUTH exchange, the Initiator proposes one or more sets of policies for the key used for securing a routing protocol in the SAI2. The SA payload indicates that the supported policies associated with the key are being proposed. The Responder returns the one policy contained in SAR2 that it accepts. Based on this policy, appropriate keying material is derived from the existing shared keying material. At the successful conclusion of the IKE_AUTH exchange, the initiator and responder have agreed upon a single set of policy and keying material for a particular routing protocol.

3.3. CREATE_CHILD_SA

The network devices may then destroy the state associated with the IKEv2 SA, continuing to use the RP policy and keying material, or they may choose to retain them for further usages. Note that this policy differs from IKEv2/IPsec, where the deletion of the IKEv2 SA necessitates the deletion of the IPsec SAs. If both the network devices choose to retain them, they may use the IKEv2 SA to subsequently agree upon replacement policy for the same RP, or agree upon the policy and keying material for another routing protocol.

Either case will require the use of the IKEv2 CREATE_CHILD_SA exchange as defined in IKEv2 [[RFC5996](#)].

A CREATE_CHILD_SA exchange therefore can be triggered in order to

1. Rekey an antique RP master key and establish a new equivalent one,
2. Generate needed keying material for a newly executed routing protocol based on an existing SA, or
3. Rekey an IKEv2 SA and establish a new equivalent IKEv2 SA.

Peer (Initiator)	Peer (Responder)
-----	-----
HDR, SK {[N], SA, Ni, [KEi], [TSi, TSr]}	-->
	<-- HDR, SK {SA, Nr, [KEr], [TSi, TSr]}

CREATE_CHILD_SA

A CREATE_CHILD_SA exchange MAY be initiated by either end of the SA after the initial exchanges are completed. All messages in a CREATE_CHILD_SA exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the initial exchange.

For details on the exchange, refer to the CREATE_CHILD_SA exchange as defined in IKEv2 [[RFC5996](#)].

3.4. INFORMATIONAL

The IKEv2 INFORMATIONAL exchange is also useful for deleting specific IKEv2 SAs or sending status information. The Notify (N) and Delete (D) payloads are as those defined by IKEv2 [[IKEV2-PARAMS](#)]. For example, if the Responder refused to accept one of Proposals sent by the Initiator, it would return an INFORMATIONAL exchange of type NO_PROPOSAL_CHOSEN instead of the response to CREATE_CHILD_SA.

Peer (Initiator)	Peer (Responder)
-----	-----
HDR, SK {[N,] [D,] ... }	-->
	<-- HDR, SK {[N,] [D,] ... }

INFORMATIONAL

4. Operation Details

4.1. General

IKEv2 is used to dynamically derive keying material information between the two network devices trying to establish or maintain a routing protocol neighbor adjacency. Typically network devices running the routing protocols establish neighbor adjacencies at the routing protocol level. These routing protocols may run different security algorithms that provide transport level security for the protocol neighbor adjacencies. Depending on the security algorithm used, the routing protocols are configured with security algorithm specific keys that are either long term keys or short term session keys. These keys are specific to the security algorithms used to enforce transport level security for the routing protocols.

A routing protocol causes IKEv2 to execute when it needs keying material to establish neighbor adjacency. This can be as a result of the routing protocol neighbor being configured, neighbor changed or updated, a local rekey policy decision, or some other event dictated by the implementation. The keying material would allow the network devices to then independently generate the same key and establish an IKEv2 session between them. This is typically done by the Initiator (IKEv2 speaker) initiating an IKEv2 IKE_SA_INIT exchange mentioned in the [section 2.1](#) towards its IKEv2 peer. As part of IKEv2_INIT exchange, IKEv2 will send a message to the peer's IKEv2 port. The format of the message is explained in [Section 6](#). The procedure to exchange key information is explained in [Section 6](#). Once the keying material information is successfully exchanged by both of the IKEv2 speakers, the IKEv2 neighbor adjacency may be torn down or kept around as explained in [Section 6](#).

The master key data received from IKEv2 peers is stored in the separate Key Management Database known as KMDB. KMDB follows the guidelines in Database of Long Lived Symmetric Cryptographic Keys [[I-D.ietf-karp-crypto-key-table](#)], and each entry consists of Key specific information, Security algorithm to which the Key is applicable to, Routing Protocol Clients of interest, and the announcing KMP Peer. KMDB is also used to notify the routing protocols about the key updates. Typically keying material information is exchanged whenever a routing protocol is about to create a new neighbor adjacency. This is considered as an Initial Key exchange mode. Keying material information is also exchanged to refresh existing key data on an already existing neighbor adjacency. This is considered as Key rollover exchange mode. The following sections describes their detail behavior.

4.2. Initial Key Specific Data Exchange

Routing protocols inform IKEv2 of its new neighbor adjacency. It does so by creating a local entry in KMDB which consists of a Security algorithm, Key specific information, routing protocol client and the routing protocol neighbor. Upon a successful creation of such an entry IKEv2 initiates KMP peering with the neighbor and starts an initial IKE_SA_INIT exchange explained in [Section 3.1](#) followed by the RP_AUTH exchange explained in [Section 3.2](#). Once the key related information is successfully exchanged, KMDB may invoke the routing protocol client to provide key specific information updates if any.

4.3. Key Selection, Rollover and Protocol Interaction

A routing protocol may need to perform the key selection and rollover in cooperation with KMDB. Such a procedure is described in [Section 3](#) of Database of Long-Lived Symmetric Cryptographic Keys [[I-D.ietf-karp-crypto-key-table](#)]. Details of how RP interact with KMDB and deals with multiple keys during rollover are also described in that section. When a routing protocol uses TCP-AO to secure its message exchanges, conditions could be a little more complex. Typically, a TCP-AO implementation has its own key tables. TCP-AO may only carry out key management operations on the key tables if the key information maintained in KMDB needs not to be updated. In [[I-D.chunduri-karp-using-ikev2-with-tcp-ao](#)], a Gatekeeper (GK) mechanism is provided to orchestrate the key management operations on the TCP-AO key tables and KMDB.

5. Key Management Database

Protocol interaction between KMP and its client routing protocols is typically done using KMDB. Routing protocols may be able to update KMDB by performing key selection and rollover operations. During a key selection, if there is no appropriate key found in the conceptual database, as a part of the KMDB update, IKEv2 is initiated to connect with its appropriate IKEv2 peer so as to generate a new key. When a key needs to be revoked, it is also the responsibility of IKEv2 to inform its peer to guarantee the synchronization of the databases on the both sides. In addition, when a key is obsoleted for some reasons when it is being used by a client routing protocol, the routing protocol may need to be informed of this update. For the routing protocols which use TCP-AO to secure their message exchanges, a Gatekeeper mechanism is provided to trigger the update of keys and manage the key revocation [[I-D.chunduri-karp-using-ikev2-with-tcp-ao](#)].

6. Header and Payload Formats

The protocol defined in this memo uses IKEv2 payload definitions. However, new security policy definitions are described to support security transforms and policy defined by routing protocol documents.

6.1. Header and Payload Formats for TCP-A0

6.1.1. Security Association Payload for TCP-A0

The TCP Authentication Option (TCP-AO) [[RFC5925](#)] is primarily intended for BGP and other TCP-based routing protocols. In order for IKEv2 to negotiate TCP-AO policy, a new Security Protocol Identifier needs to be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers' for ISAKMP Protocol [[IKEV2-PROTOCOL-IDS](#)]. This memo proposes adding a new Protocol Identifier to the table, with a Protocol Name of "TCP_A0" and a value of 6.

The Security Association (SA) payload contains a list of Proposals, which describe one or more sets of policies that a network device is willing to use to protect a routing protocol. In the Initiator's message, the SAi2 payload contains a list of Proposal payloads (as defined in the next sections), each of which contains a single set of policy that can be applied to the packets described in the Traffic Selector (TS) payloads in the same exchange. Each set of policy is given a particular "Proposal Number" uniquely identifying this set of policy.

The responder includes a single Proposal payload in its SA policy, which denotes the choice it has made amongst the initiator's list of Proposals. Any attributes of a selected transform MUST be returned unmodified as explained in IKEv2 [[RFC5996](#)] [section 3.3.6](#). The initiator of an exchange MUST check that the accepted offer is consistent with one of its proposals, and if not MUST terminate the exchange.

6.1.1.1. Transforms Substructures for TCP-A0

Each Proposal has a list of Transform (T) substructures, each of which describe a particular set of cryptographic policy choices. A TCP-A0 proposal uses the INTEG transform to negotiate the MKT Message Authentication Code (MAC) algorithm. Cryptographic Algorithms for TCP-A0 [RFC5926] describes HMAC-SHA-1-96, AES-128-CMAC-96, which map to the existing INTEG transform IDs of AUTH_HMAC_SHA1_96 and AUTH_AES_CMAC_96 respectively. The use of each INTEG algorithm implies the use of a specific KDF (deriving session keys from a master key), and so the choice of a particular INTEG transform ID also specifies the required KDF transform. This will be true for every transform ID used with TCP-A0, as required in RFC 5926 (see Section 3.2 where the "KDF_Alg" is a fixed element of a MAC algorithm definition for TCP-A0).

A TCP-A0 proposal also requires a new type of transform, which describes whether TCP options are to be protected by the integrity algorithm. This memo proposes adding a new Transform Type in the IANA registry for "Transform Type Values" [IKEV2-TRANSFORM-TYPES]

+-----+-----+-----+-----+-----+	
Number	Name
+-----+-----+-----+-----+-----+	
0	Options Not Integrity Protected
1	Options Integrity Protected
+-----+-----+-----+-----+-----+	

Figure 2: Transform Type 6 - TCP Authentication Option Transform IDs

The TCP-A0 KeyID is sent in the SPI field of an IKEv2 proposal. A KeyID for TCP-A0 has the same purpose as an IPsec SPI value, so it is natural to place it in this portion of the proposal. If the KeyID values in a responder's Proposal does not mach the KeyID values initiator's Proposal, then they have chosen to use different KeyID values to represent the same master key and associated proposal policy. This is consistent with how IPsec uses the SPI value, and the semantic of initiator and responder using different SendIDs is supported by RFC 5925.

The following table shows the Transforms that can be negotiated for a TCP-A0 protocol.

Protocol	Mandatory Types	Optional Types

TCP-A0	INTEG, TCP	D-H

Figure 3: Mandatory and Optional Transforms for TCP-A0

6.1.1.2. Example Proposal Exchange

Figure 4 shows an example of IKEv2 SA Payload including a single Proposal sent in the first message of an IKE_AUTH or CREATE_CHILD_SA exchange. It indicates a willingness to use either of the two MAC algorithms defined in [RFC 5926](#), and is willing to either protect TCP options or not. The SPI value represents the new SendID it is associating with the TCP-A0 Master Key Tuple (MKT) policy being negotiated.

SA Payload

```
|
+--- Proposal #1 ( Proto ID = TCP-A0(T6), SPI size = 1,
|               4 transforms,      SPI = 0x01 )
|
+--- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
+--- Transform INTEG ( Name = AUTH_AES_CMAC_96 )
+--- Transform TCP ( Name = PROTECT_OPTIONS )
+--- Transform TCP ( Name = NO_PROTECT_OPTIONS )
```

Figure 4: Example Initiator SA Payload for TCP-A0

The responder will record the SPI value to be the RecvID of the MKT. It chooses its own SendID value, one of each Transform type, and returns this policy in the response message. For example, if the responder chose HMAC-SHA-1-96 and chose to protect the TCP options, the corresponding SA payload would be:

SA Payload

```
|
+--- Proposal #1 ( Proto ID = TCP-A0(6), SPI size = 1,
|               2 transforms,      SPI = 0x11 )
|
+--- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
+--- Transform TCP ( Name = PROTECT_OPTIONS )
```

Figure 5: Example Responder SA Payload for TCP-A0

In this example, the Proposal responder chose to use a different SPI value (0x11) as its SendID. This is possible because [Section 2.2 of \[RFC5925\]](#) declares that "KeyID values MAY be the same in both directions of a connection, but do not have to be and there is no special meaning when they are."

6.1.2. Derivation of TCP-A0 Keying Material

Each TCP-A0 MAC algorithm specification in [Section 3.2](#) of Crypto for TCP-A0 [[RFC5926](#)] defines the Key_Length as a number of bits <n> needed as keying material for the MAC algorithm.

6.2. Security Association Payload for BFD

In order for IKEv2 to negotiate BFD authentication policy, a new Security Protocol Identifier needs to be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers' for ISAKMP Protocol [[IKEV2-PROTOCOL-IDS](#)]. This memo proposes adding a new Protocol Identifier to the table, with a Protocol Name of "BFD" and a value of 7.

6.2.1. Transforms Substructures for BFD Authentication

The base BFD specification [[RFC5880](#)] defines five authentication mechanisms: Password, Keyed MD5, Meticulous Keyed MD5, Keyed SHA1, and Meticulous Keyed SHA1. Because Password does not use keys, the support of this mechanism is out of the scope of this work. In the other four mechanisms, Keyed MD5 and Meticulous Keyed MD5 use MD5 as the Message Authentication Code (MAC) algorithm, while Keyed SHA1 and Meticulous Keyed SHA1 use SHA1. In [[I-D.ietf-bfd-generic-crypto-auth](#)], a generic authentication mechanism and a generic meticulous authentication mechanism which can support various MAC algorithms is proposed.

Therefore, a BFD proposal also requires a new type of transform to identify the type of BFD authentication. This memo proposes adding a new Transform Type in the IANA registry for "Transform Type Values" [[IKEV2-TRANSFORM-TYPES](#)]

+-----+-----+-----+-----+-----+	
Number	Name
+-----+-----+-----+-----+-----+	
0	Base Authentication
1	Base Meticulous Authentication
2	Generic Authentication
3	Generic Meticulous Authentication
+-----+-----+-----+-----+-----+	

Figure 6: Transform Type 7 - BFD Authentication Option Transform IDs

Base Authentication in Figure 6 indicates the keyed (MD5 or SHA-1) authentication mechanism defined in the base BFD specification [[RFC5880](#)]. Base Meticulous Authentication indicates the meticulous keyed (MD5 or SHA-1) authentication mechanism defined in the base BFD specification. Generic Authentication and Generic Meticulous Authentication indicate the generic keyed authentication and the

generic keyed meticulous authentication mechanisms defined in [\[I-D.ietf-bfd-generic-crypto-auth\]](#) respectively.

A BFD proposal uses INTEG transforms to negotiate Message Authentication Code (MAC) algorithms. In the base BFD [\[RFC5880\]](#), keyed MD5 and keyed SHA-1 are adopted. The two algorithms can be identified using existing INTEG transform IDs of AUTH_HMAC_MD5_96 and AUTH_HMAC_SHA1_96 respectively. In [\[I-D.ietf-bfd-hmac-sha\]](#), it is specified that a BFD using the authentication mechanisms defined in [\[I-D.ietf-bfd-generic-crypto-auth\]](#) MUST support HMAC-SHA-256 which can be identified using existing INTEG transform IDs of AUTH_HMAC_SHA2_256_128 [\[RFC4868\]](#).

The BFD KeyID is sent in the SPI field of an IKEv2 proposal. Note that according to [\[RFC5880\]](#), the length of KeyID is 8 bits.

Because in BFD the transport key is the same as the protocol master key, no KDF needs to be negotiated.

The following figure shows the Transforms that can be negotiated for a BFD implementation.

Protocol	Mandatory Types	Optional Types

BFD	BFD, INTEG	D-H

Figure 7: Mandatory and Optional Transforms for BFD

6.3. Security Association Payload for RSVP-TE

In order for IKEv2 to negotiate RSVP-TE authentication policy, a new Security Protocol Identifier needs to be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers' for ISAKMP Protocol [\[IKEV2-PROTOCOL-IDS\]](#). This memo proposes adding a new Protocol Identifier to the table, with a Protocol Name of "RSVP-TE" and a value of 8.

6.3.1. Transforms Substructures for RSVP-TE Authentication

In the authentication mechanism for RSVP-TE [\[RFC2747\]](#), only HMAC-MD5 is mandated. Therefore, no INTEG transform needs to be included in a RSVP-TE proposal.

A RSVP-TE proposal requires a new type of transform, which indicates whether the integrity handshake (which is used to collect the latest sequence number associated with a key ID) is permitted. This memo proposes adding a new Transform Type in the IANA registry for "Transform Type Values" [\[IKEV2-TRANSFORM-TYPES\]](#)

Number	Name
0	Not Allowed
1	Allowed

Figure 8: Transform Type 8 - RSVP-TE Transform IDs

The RSVP-TE KeyID is sent in the SPI field of an IKEv2 proposal.

The following figure shows the Transforms that can be negotiated for a RSVP-TE implementation.

Protocol	Mandatory Types	Optional Types
RSVP-TE	RSVP-TE,	D-H

Figure 9: Mandatory and Optional Transforms for BFD

6.4. Notify and Delete Payloads

A Notify Payload (IKEv2 [\[RFC5996\] Section 3.10](#)) or Delete Payload (IKEv2 [\[RFC5996\] Section 3.11](#)) contains a Protocol ID field. The Protocol ID is set to TCP_A0 (6) when a notify message is relevant to the TCP-A0 KeyID value contained in the SPI field. Similarly, the Protocol ID is set to BFD (7) when a notify message is relevant to the BFD KeyID value contained in the SPI field, and the Protocol ID is set to RSVP-TE (8) when a notify message is relevant to the RSVP-TE KeyID value contained in the SPI field.

7. IANA Considerations

In order for IKEv2 to negotiate TCP-A0 authentication policies, a new Security Protocol Identifier needs to be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers' for ISAKMP Protocol [\[IKEV2-PROTOCOL-IDS\]](#). IANA is requested to add a new Protocol Identifier to the table, with a Protocol Name of "TCP-A0" and a value of 6. A TCP-A0 proposal also requires a new type of transform, which describes whether TCP options are to be protected by the integrity algorithm. This memo proposes adding a new Transform Type 6 for this transform in the IANA registry for "Transform Type Values".

In order for IKEv2 to negotiate BFD authentication policies, a new Security Protocol Identifier needs to be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers' for ISAKMP Protocol [\[IKEV2-PROTOCOL-IDS\]](#). IANA is requested to add a new

Protocol Identifier to the table, with a Protocol Name of "BFD" and a value of 7. A BFD proposal also requires a new type of transform, which identifies the type of BFD authentication mechanism. This memo proposes adding a new Transform Type 7 in the IANA registry for "Transform Type Values".

In order for IKEv2 to negotiate RSVP-TE authentication policies, a new Security Protocol Identifier needs to be defined in the IANA registry for "IKEv2 Security Protocol Identifiers" Magic Numbers' for ISAKMP Protocol [[IKEV2-PROTOCOL-IDS](#)]. IANA is requested to add a new Protocol Identifier to the table, with a Protocol Name of "RSVP-TE" and a value of 8. A RSVP-TE proposal requires a new type of transform, which indicates whether the integrity handshake (which is used to collect the latest sequence number associated with a key ID) is permitted. This memo proposes adding a new Transform Type 8 in the IANA registry for "Transform Type Values".

8. Security Considerations

TBD

9. Acknowledgements

During the development of TCP-A0, Gregory Lebovitz noted that a protocol based on an IKEv2 exchange would be a good automated key management method for deriving a TCP-A0 master key.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", [RFC 2747](#), January 2000.
- [RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", [RFC 4868](#), May 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", [RFC 5880](#), June 2010.

- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", [RFC 5925](#), June 2010.
- [RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-A0)", [RFC 5926](#), June 2010.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", [RFC 5996](#), September 2010.

10.2. Informative References

- [DH] Diffie, W. and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22 n. 6, June 1977.
- [I-D.chunduri-karp-using-ikev2-with-tcp-ao]
Chunduri, U., Tian, A., and J. Touch, "A framework for RPs to use IKEv2 KMP", [draft-chunduri-karp-using-ikev2-with-tcp-ao-05](#) (work in progress), July 2013.
- [I-D.ietf-bfd-generic-crypto-auth]
Bhatia, M., Manral, V., and D. Zhang, "BFD Generic Cryptographic Authentication", [draft-ietf-bfd-generic-crypto-auth-05](#) (work in progress), October 2013.
- [I-D.ietf-bfd-hmac-sha]
Zhang, D., Bhatia, M., and V. Manral, "Authenticating BFD using HMAC-SHA-2 procedures", [draft-ietf-bfd-hmac-sha-04](#) (work in progress), October 2013.
- [I-D.ietf-karp-crypto-key-table]
Housley, R., Polk, T., Hartman, S., and D. Zhang, "Database of Long-Lived Symmetric Cryptographic Keys", [draft-ietf-karp-crypto-key-table-09](#) (work in progress), October 2013.
- [IKEV2-PARAMS]
, "Internet Key Exchange Version 2 (IKEv2) Parameters", , <<http://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xml>>.
- [IKEV2-PROTOCOL-IDS]
, "'Magic Numbers' for ISAKMP Protocol", , <<http://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xml#ikev2-parameters-18>>.

[IKEV2-TRANSFORM-TYPES]

, "'Magic Numbers' for ISAKMP Protocol", , <<http://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xml#ikev2-parameters-3>>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", [RFC 6952](#), May 2013.

[TCP-A0-REG]

, "Internet Key Exchange Version 2 (IKEv2) Parameters", , <<http://www.iana.org/assignments/tcp-parameters/tcp-parameters.xml>>.

Authors' Addresses

Mahesh Jethanandani
Ciena Corporation
3939 North First Street
San Jose, CA 95134
USA

Phone: +1 (408) 904-2160
Email: mjethanandani@gmail.com

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

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

Keyur Patel
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Phone: +1 (408) 526-7183
Email: keyupate@cisco.com

Dacheng Zhang
Huawei
Beijing
China

Email: zhangdacheng@huawei.com

Sam Hartman
Painless Security

Email: hartmans@painless-security.com

Uma Chunduri
Ericsson Inc.
300 Holger Way
San Jose, California 95134
USA

Email: uma.chunduri@ericsson.com

Albert Tian
Ericsson Inc.
300 Holger Way
San Jose, California 95134
USA

Email: albert.tian@ericsson.com

Joe Touch
USC/ISI
4676 Admiralty Way
Marina del Rey, California 90292-6695
USA

Email: touch@isi.edu

