TCP Authentication Option Master Key Tuple negotiation in IKEv2

draft-mahesh-karp-rkmp-02

Abstract

This document describes a mechanism to secure TCP-based pairwise Routing Protocol (RP) associations using the IKEv2 Key Management Protocol (KMP) integrated with TCP-AO. Included are extensions to IKEv2 and its Security Associations to enable its key negotiation to support TCP-AO.

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 April 25, 2013.
1. Introduction

Existing routing protocols using unicast communication model (e.g., BGP, LDP, RSVP-TE) have cryptographic authentication mechanisms that use a key shared between the routers on the both sides of the model to protect routing message exchanges between the routers. Unicast key management today is limited to statically configuring master keys in individual routers. This document defines a mechanism to secure TCP-based pairwise Routing Protocol (RP) associations using IKEv2 [RFC5996], allowing network devices to automatically exchange key material related information between the network devices.

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

When two routers running a routing protocol have not authenticated each other yet, and before sending out any routing protocol packets the two routers need to perform mutual authentication using their provisioned credentials. If successful, two routers negotiate the key material to secure the routing protocol execution.

1.1. Terminology

Here are some terms that we will be using throughout the document.

TBD

1.2. Acronyms and Abbreviations

The following acronyms and abbreviations are used throughout this document.

IKE   Internet Key Exchange Protocol
IKEv2 Internet Key Exchange Protocol Version 2
RP    Routing Protocol
SA    Security Association

2. Overview
2.1. Types of Keys

The keys adopted in RKMP are listed as follows:

- **PSK (Pre-Shared Key):** PSKs are pair-wise unique keys used for authenticating one router to the other one during the initial exchange. These keys are configured by some mechanism such as manual configuration or a management application outside of the scope of RKMP.

- **Seed key:** Refers to value derived from SKEYSEED that is used to derive new keys (e.g., for TCP-AO).

- **Protocol master key:** A protocol master key is the key exported by RKMP for use by a routing protocol such as BGP. This is the key that is shared in the key table between the routing protocol and RKMP.

- **Transport key:** A transport key is the key used to integrity protect routing messages in a protocol such as BGP. In today’s routing protocol cryptographic authentication mechanisms the transport key can be the same as the protocol master key.
3. Protocol Exchanges

The exchange of private keying material between two network devices using a dedicated key management protocol is a requirement as articulated in [I-D.ietf-karp-routing-tcp-analysis]. 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.

In the following figures, the notations contained in the message are defined as follows.

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

Acronyms Used in Protocol Exchange

3.1. IKE_SA_INIT

A network device desiring to negotiate a TCP-AO MKT to a peer initiates an IKE_SA_INIT exchange defined in Internet Key Exchange Protocol Version 2 [RFC5996]. The IKE_SA_INIT exchange is a two-message exchange that allows the network devices to negotiate cryptographic algorithms, exchange nonces, and do a Diffe-Hellman (DH) exchange, for their routing protocols, after which protocols on these network devices can communicate privately. Note that at this point the network devices have not identified their peer. For the details of this exchange, refer to IKE_SA_INIT in Internet Key Exchange Protocol Version 2 [RFC5996].
3.2. IKE_AUTH

Next, the network devices perform an IKE_AUTH exchange defined in RFC 5996. The SA payloads contain the security policies for a TCP-AO MKT (as defined in Section 4), and the TS payloads contains traffic selectors as defined in [RFC5996]. For the details of the exchange please refer to IKE_AUTH in RFC 5996.

In the IKE_AUTH exchange, the Initiator proposes one or more sets of policies for a TCP-AO MKT in the SAi2. The SA payload indicates that TCP-AO MKT policy is being proposed, and the TS payloads represent the traffic selectors for the particular routing protocol that will use the TCP-AO MKT (e.g., BGP or LDP). 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 the further use. 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 policy and keying material for another routing protocol. Either case will require the use of the IKEv2 CREATE_CHILD_SA exchange as defined in RFC 5996.
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 key material for a newly executed routing
   protocol based on an existing SA

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 ]}

Figure 4: IKEv2 CREATE_CHILD_SA Exchange

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 RFC 5996.

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, ] ... }

Figure 5: IKEv2 INFORMATIONAL Exchange

4. 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.
4.1. Security Association Payload

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" [IKEV2-PROTOCOL-IDS]. This memo proposes adding a new Protocol Identifier to the table, with a Protocol Name of "TCP_AO" and a value of TBD1.

The Security Association (SA) payload contains a list of Proposals, which describe one or more sets of policy that a router 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 section), 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 it’s 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.

4.1.1. Transforms Substructures

Each Proposal has a list of Transform (T) substructures, each of which describe a particular set of cryptographic policy choices. A TCP-AO proposal uses the INTEG transform to negotiate the MKT Message Authentication Code (MAC) algorithm. Cryptographic Algorithms for the TCP Authentication Option (TCP-AO) [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) so no 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-AO, as required in RFC 5926 (see Section 3.2 where the "KDF_Alg" is a fixed element of a MAC algorithm definition for TCP-AO).

A TCP-AO 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]
The TCP-AO KeyID that is sent in the SPI field of an IKEv2 proposal. A KeyID for TCP-AO 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 in a responder’s Proposal is not the same as the 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-AO protocol.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Mandatory Types</th>
<th>Optional Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP-AO</td>
<td>INTEG, TCP</td>
<td>D-H</td>
</tr>
</tbody>
</table>

Figure 7: Mandatory and Optional Transforms

4.1.2. Example Proposal Exchange

Figure 8 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-AO Master Key Tuple (MKT) policy being negotiated.
SA Payload

| Proposal #1 ( Proto ID = TCP-AO(TBD1), 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 8: Example Initiator SA Payload for TCP-AO

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-AO(TBD1), SPI size = 1,  |
| 2 transforms, SPI = 0x11 ) |
|--- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 ) |
|--- Transform TCP ( Name = PROTECT_OPTIONS ) |

Figure 9: Example Responder SA Payload for TCP-AO

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."

4.2. Derivation of TCP-AO Keying Material

Each TCP-AO MAC algorithm specification in Section 3.2 of [RFC5926] defines the number of bits <n> needed by the MAC algorithm. The first <n> bits of KEYMAT (according to Section 2.17 of [RFC5996]) are used as the key for the negotiated MAC algorithm.

4.3. Notify and Delete Payloads

A Notify Payload ([RFC5996] Section 3.10) or Delete Payload ([RFC5996] Section 3.11) contains a Protocol ID field. The Protocol ID is set to TCP_AO (TBD1) when a notify message is relevant to the TCP-AO KeyID value contained in the SPI field.
5. Operation Details

5.1. General

IKEv2 is used to dynamically derive key 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 key 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 key 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 4. The procedure to exchange key information is explained in Section 4. Once the key 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 4.

The master key data received from IKEv2 peers is stored in the separate Key Management Database known as KMDB. KMDB follows the guidelines inDatabase 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 RKMP Peer. KMDB is also used to notify the routing protocols about the key updates. Typically key 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. Key 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.
5.2. Initial Key Specific Data Exchange

Routing protocols informs 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 RKMP peering with the neighbor and starts an initial IKE_SA_INIT exchange explained in Section 3.1 followed by the RP_AUTH exchanged 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.

5.3. Key Selection, Rollover and Protocol Interaction

The procedure for key selection and rollover exchange has been 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.

6. Key Management Database (KMDB)

Protocol interaction between RKMP and its client routing protocols is typically done using KMDB. Routing protocols update KMDB by installing a new Key related information or purging an existing Key specific information. As part of the KMDB update, IKEv2 initiates peering connections with its appropriate IKEv2 peers to announce the updated key related information. IKEv2 may also receive an updated key related information from its peers which gets installed in KMDB. Whenever IKEv2 updates KMDB with updated key information from its peers, it notifies client routing protocols of its updates.

7. IANA Considerations

TBD

8. Security Considerations

TBD

9. Acknowledgements

During the development of TCP-AO, Gregory Lebovitz noted that a
protocol based on an IKEv2 exchange would be a good automated key
management method for deriving a TCP-AO master key. Joe Touch
provided many helpful comments.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

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


[RFC5926] Lebovitz, G. and E. Rescorla, "Cryptographic Algorithms
for the TCP Authentication Option (TCP-AO)", RFC 5926,
June 2010.

"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.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-03 (work in progress),
June 2012.

[I-D.ietf-karp-routing-tcp-analysis]
Jethanandani, M., Patel, K., and L. Zheng, "Analysis of
BGP, LDP, PCEP and MSDP Issues According to KARP Design
Guide", draft-ietf-karp-routing-tcp-analysis-05 (work in
progress), October 2012.

[IKEV2-PARAMS]
"Internet Key Exchange Version 2 (IKEv2) Parameters",
<http://www.iana.org/assignments/ikev2-parameters/>
ikev2-parameters.xml>.

[IKEV2-PROTOCOL-IDS]

[IKEV2-TRANSFORM-TYPES]

Authors’ Addresses

Mahesh Jethanandani
Ciena Corporation
1741 Technology Drive
San Jose, CA  95110
USA
Phone: +1 (408) 436-3313
Fax: Email: mjethanandani@gmail.com
URI:

Brian Weis
Cisco Systems
170 W. Tasman Drive
San Jose, California  95134
USA
Phone: +1 (408) 526-4796
Fax: Email: bew@cisco.com
URI: