Abstract

When running routing protocols such as BGP or RSVP-TE, two routers need to exchange routing messages in a unicast (one-to-one) fashion. In order to authenticate these messages using symmetric cryptography, a secret key needs to be established. This document defines a Router Key Management Protocol for establishing and managing such keys for routing protocols.

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 21, 2012.

Copyright Notice
Copyright (c) 2011 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. Terminology ...................................... 3
  1.2. Acronyms and Abbreviations ....................... 3
  1.3. Relationship to IKEv2 ............................. 3
2. Overview ............................................... 4
  2.1. Types of Keys ..................................... 4
3. Protocol Exchanges ..................................... 4
  3.1. RP_INIT ........................................ 5
  3.2. RP_AUTH ........................................ 6
  3.3. RP_ADD .......................................... 6
  3.4. INFORMATIONAL .................................. 7
4. Header and Payload Formats ........................... 7
  4.1. Security Association Payload .................... 8
     4.1.1. Proposal Substructure ....................... 8
        4.1.1.1. Transforms Substructures ............... 10
           4.1.1.1.1. RKMP .......................... 10
           4.1.1.1.2. TCP-AO Transforms ............... 10
  4.2. Traffic Selector Payload ........................ 12
5. Operation Details ..................................... 12
  5.1. General .......................................... 12
  5.2. Initial Key Specific Data Exchange ............. 13
  5.3. Key Specific Data Rollover Exchange ........... 13
6. Key Management Database (KMDB) ...................... 14
7. Protocol Interaction .................................. 14
8. IANA Considerations .................................. 14
9. Security Considerations .............................. 14
10. Acknowledgements .................................... 14
11. References .......................................... 15
  11.1. Normative References ......................... 15
  11.2. Informative References ....................... 15
Authors’ Addresses ...................................... 16
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 extends currently defined IKEv2 [RFC5996] protocol to define a Router Key Management Protocol (RKMP) that allows network devices to automatically exchange key material related information between the network devices.

RKMP assumes that routers need to be provisioned with some credentials for a one-to-one authentication protocol. Preshared keys or asymmetric keys and an authorization list are expected to be common deployments.

If 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

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
SA    Security Association

1.3. Relationship to IKEv2

IKEv2 provides a protocol for authenticating IPsec security associations between two peers. It currently provides no group keying. IKEv2 is attractive as a basis for this protocol because while it is much simpler than IKE [RFC2409], it provides all the needed flexibility in one-to-one authentication.

Unlike IKE, IKEv2 is explicitly designed for IPsec. The document does not separate handling aspects of the protocol that would be needed for IPsec from those that apply to general key management. IPsec specific rules are combined with more general requirements.
While concepts and protocol payloads can be used in a different key
management protocol, the current structure of IKEv2 does not provide
a mechanism for applying IKEv2 to a domain of interpretation other
than IPsec. In addition, the complexity required in the IKE
specification when compared to IKEv2 suggests that the generality of
IKE may not be worth the complexity cost.

This protocol borrows concepts and payloads from IKEv2 but does not
normatively depend on the IKEv2 specification.

2. Overview

[Need an overview of how RKMP works, maybe a protocol flow picture
and/or state machine picture. This would be a preface to the actual
protocol descriptions in Section 3.]

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. However, as IKEv2 was developed exclusively for the use of IPsec, these protocol exchanges are incorporated by reference into the present key protocol definitions, and are exchanged using a dedicated UDP port number (TDB - IANA). The use of a dedicated UDP port will clearly differentiate this protocol from IKEv2.

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>KMPRP 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. RP_INIT

The RP Initial Exchange (RP_INIT) is identical to the IKE_SA_INIT exchange defined in Internet Key Exchange Protocol Version 2 [RFC5996]. The RP_INIT exchange is a two-message exchange that allows the network devices to negotiate cryptographic algorithms, exchange nonces, and do a Diffe-Hellman (DH) [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. RP_AUTH

Next, the network devices perform a RP Authentication exchange (RP_AUTH), which is substantially the same as the IKE_AUTH exchange defined in RFC 5996, except that the SA payload contains policy specific to the routing protocol security policy (labeled SArpi and SArpr) rather than IPsec policy (SAi2, SAr2 defined in RFC 5996). The SArpi and SArpr payloads are described in Section 3; for the details of the rest of the exchange please refer to IKE_AUTH in RFC 5996.

3.3. RP_ADD

The network devices may then destroy the state associated with the RP SA, continuing to use the RP policy and keying material, or they may choose to retain them for the further use. If both the network devices choose to retain them, they may use the RP 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 RP Additional Exchange (RP_ADD), similar to the IKEv2 CREATE_CHILD_SA exchange as defined in RFC 5996.

A RP_ADD exchange therefore can be triggered in order to
1. Rekey an antique protocol 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 RMKP_SA and establish a new equivalent RMKP_SA

Peer (Initiator)                      Peer (Responder)
--------------------                  ------------------
HDR, SK {SArpi, Ni, [KEi },       -->  HDR, SK {SArpr, Nr, [KEr },
      TS}      <--     HDR, SK {SArpr, Nr, [KEr },
                      TS}

RP_ADD

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

In the RP_ADD exchange, the SA payloads in the RP_ADD exchange are used identically as in the RP_AUTH exchange. For details on the rest of 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 RP 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 RP_ADD.

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

INFORMATIONAL

4. Header and Payload Formats

The protocol defined in this memo uses a HDR identical to the Generic Payload Header defined in section 3.2 of RFC 5996. The new exchanges defined in this memo are not used with IKEv2. A new IANA registry is
to be created to identify the RP exchange types and payloads described in this section.

4.1. Security Association Payload

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. It is identical to the SA payload described in RFC 5996, and the details of the fields are described there.

In the Initiator’s message, the SArpi 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. For example, the TS payloads may describe a set of IP addresses and ports which are a BGP connection, and the SA payload contains a list of proposals describing what policy the router is willing to use to protect that BGP traffic. 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.

Security Association Payload

The Security Association Payload fields are defined as in RFC 5996.

4.1.1. Proposal Substructure

The Proposal (P) substructure of the Security Association Payload contains an identification for the set of policy choices, the security protocol offered in the proposal, and details of the
cryptographic choices offered.

<table>
<thead>
<tr>
<th>Proposal Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>0  (last) or 2</td>
</tr>
<tr>
<td>Proposal Num</td>
</tr>
<tr>
<td>~</td>
</tr>
</tbody>
</table>

Proposal Payload

- 0 (last) or 2 (more) (1 octet) - Specifies whether this is the last Proposal Substructure in the SA.

- RESERVED (1 octet) - MUST be sent as zero; MUST be ignored on receipt.

- Proposal Length (2 octets, unsigned integer) - Length of this proposal, including all transforms and attributes that follow.

- Proposal Num (1 octet) - When a proposal is made, the first proposal in an SA payload MUST be 1, and subsequent proposals MUST be one more than the previous proposal (indicating an OR of the two proposals). When a proposal is accepted, the proposal number in the SA payload MUST match the number on the proposal sent that was accepted.

- Protocol ID (1 octet) - Specifies the protocol identifier for the current negotiation.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Protocol ID</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>RKMP</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TCP-AO</td>
<td>2</td>
<td>RFC 5925</td>
</tr>
<tr>
<td>LDP Discovery Key</td>
<td>3</td>
<td>TBD</td>
</tr>
<tr>
<td>Standards Action</td>
<td>4-128</td>
<td></td>
</tr>
<tr>
<td>Private Use</td>
<td>129-255</td>
<td></td>
</tr>
</tbody>
</table>

Protocol ID

- Num Transforms (1 octet) - Specifies the number of transforms in this proposal.
4.1.1.1. Transforms Substructures

Each Proposal has a list of Transform (T) substructures, each of which describe a particular set of cryptographic policy choices. This is useful for an initiator to propose multiple cryptographic choices for the same policy described in its associated Proposal payload.

4.1.1.1.1. RKMP

This transform payload is used to negotiate policy to protect the RKMP exchanges. The Transforms are identical to the Transforms specified to negotiate IKE policy in Section 3.3.2 of IKEv2 [RFC5996].

4.1.1.1.2. TCP-AO Transforms

The TCP-AO [RFC5925] transform payload contains the following fields.

```
  1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 3 |   RESERVED    |        Transform Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    SendID     |Auth Alg       |     KDF       |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP-AO Transforms

- 0 (last) or 3 (more) (1 octet) - Specifies whether this is the last Transform Substructure in the Proposal.
- RESERVED (1 octet) - MUST be sent as zero; MUST be ignored on receipt.
- Transform Length (2 octets) - The length (in octets) of the Transform Substructure including Header and Attributes.
- SendID (1 octet) - The TCP-AO KeyID that the sender will use to represent this Transform. The KeyID will be used to generate the keys independently on each network device at the end of the exchange.
- Auth Alg (1 octet) - The Authentication algorithm defined as a part of this Transform. Values are defined in Cryptographic Algorithms for the TCP Authentication Option [RFC5926].
Internet-Draft  rkmp  October 2011

Auth Alg | ID
---------|-----
HMAC-SHA-1-96  | 1
AES-128-CMAC-96 | 2
Standards Action | 3-128
Private Use | 129-255

Authentication Algorithm

- KDF (1 octet) - The KDF defined as a part of this Transform. Values are defined in Cryptographic Algorithms for the TCP Authentication Option [RFC5926].

  KDF | ID
   |-----
KDF_HMAC_SHA1 | 1
KDF_AES_128_CMAC | 2
Standards Action | 3-128
Private Use | 129-255

Key Derivation Functions

- Flags (1 octet) - Indicates specific options for TCP-AO. The bits are as follows:

   ++++++++---+-
   |O|X|X|X|X|X|X|
   ++++++++---+-

   In the description below, a bit being ‘set’ means its value is ‘1’, while ‘cleared’ means its value is ‘0’. ‘X’ bits MUST be cleared when sending and MUST be ignored on receipt.

   - O (Options) - This bit indicates whether or not TCP Options are to be included in the bytes protected by the authentication calculation. This bit is set to indicate that TCP Options are to be ignored and cleared to indicate that TCP Options are protected.

When a TCP-AO transform is chosen, keying material for the TCP-AO master key is generated as follows, where Ni and Nr are unique to this exchange. The value SK_D is defined in [RFC 5996], and refers to the value derived from SKEYSEED that is used to derive new keys (e.g., for TCP-AO).

   \[
   \text{<TCP-AO master key> = prf+}\left( SK_d, N_i \mid N_r \right)
   \]
4.2. Traffic Selector Payload

The Traffic Selector (TS) payload definition is the same as defined in Section 3.13 of IKEv2 [RFC5996]. The TS types for routing protocols would be defined as follows.

```
|   TS Type     |Rtg. Prot. ID  |       RESERVED                |
+----------------------|----------------+-----------------------------+
```

- TS Type (1 octet) - 1 for all routing protocols
- Rtg. Prot. ID (1 octet) - Specifies the routing protocol identifier for the current negotiation.

Routing (RT) Protocol Protocol ID Reference
---------------------------------------------
BGP 1 RFC 4271
LDP 2 RFC 5036
MSDP 3 RFC 3618
PIM PORT 4 RFC 5440
PCEP 5

5. Operation Details

5.1. General

KMPRP 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 KMPRP 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 a KMPRP neighbor adjacency between them. This is typically done by the Initiator (KMPRP speaker) initiating a KMPRP RP_INIT exchange mentioned in the section 2.1 towards its KMPRP peer. As part of RP_INIT exchange, KMPRP will send a message to the KMPRP peer’s well known KMPRP UDP port [TBD] by IANA. The format of the message is explained in section 3. The procedure to exchange key information is explained in section 3. Once the key material information is successfully exchanged by both the KMPRP speaker, the KMPRP neighbor adjacency may be torn down.

The master key data received from KMPRP peers are stored in the separate Key Management Database known as KMDB. KMDB follows the guidelines in [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 KMPRP 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 KMPRP 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 KMPRP initiates KMPRP peering with the neighbor and starts initial KMPRP RP_INIT exchange explained in section 2.1 followed by the RP_AUTH exchanged explained in section 2.2. 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 Specific Data Rollover Exchange

Key rollover exchange may be initiated at a pre-configured time interval or as part of a manual configuration and is outside the scope of this document. The procedure of Key Rollover exchange is exactly same as the Initial Key specific data exchange described above.
6. Key Management Database (KMDB)

Protocol interaction between KMPRP 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, KMPRP initiates peering connections with its appropriate KMPRP peers to announce the updated key related information. KMPRP may also receive an updated key related information from its peers which gets installed in KMDB. Whenever KMPRP updates KMDB with updated key information from its peers, it notifies client routing protocols of its updates.

7. Protocol Interaction

Routing protocols could end up with multiple keys when updated by KMDB. Typically, routing protocols should use the keys till the point its peers have transitioned to a new key. Once the peers have transitioned to a new key, routing protocols could put the old keys on timers and eventually free them. The reason to put them on timer and not free them right away is to ensure that all out of order packets in TCP are handled correctly.

8. IANA Considerations

A new UDP port number will need to be assigned for systems that want to implement this protocol.

A new IANA registry is to be created to identify the RP exchange types and payloads.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

TBD

10. 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.

Many protocol definitions and protocol formats come from RFC 5996,
either by reference or inclusion.

11. References

11.1. Normative References


11.2. Informative References


Authors’ Addresses

Mahesh Jethanandani
Cisco Systems
170 Tasman Drive
San Jose, California CA
USA
Phone: +1 (408) 527-8230
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:

Keyur Patel
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA
Phone: +1 (408) 526-7183
Fax:
Email: keyupate@cisco.com
URI:

Dacheng Zhang
Huawei
Beijing,
China
Phone:
Fax:
Email: zhangdacheng@huawei.com
URI: