

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2012

M. Jethanandani
B. Weis
K. Patel
Cisco Systems
July 1, 2011

Key Management for Pairwise Routing Protocol
draft-mahesh-karp-kmprp-00

Abstract

This document defines an automated method of Key Management for routing protocol that need pair-wise keys.

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 January 2, 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

Internet-Draft

kmprrp

July 2011

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
2.	Protocol Exchanges	3
2.1.	RP_INIT	4
2.2.	RP_AUTH	4
2.3.	RP_ADD	5
2.4.	INFORMATIONAL	5
3.	Header and Payload Formats	5
3.1.	Security Association Payload	6
3.1.1.	Proposal Substructure	6
3.1.1.1.	Transforms Substructures	8
3.1.1.1.1.	KMPRP	8
3.1.1.1.2.	TCP-A0 Transforms	8
4.	Operation Details	10
4.1.	General	10
4.2.	Initial Key Specific Data Exchange	11
4.3.	Key Specific Data Rollover Exchange	11
5.	Key Management Database (KMDB)	11
6.	Protocol Interaction	12
7.	IANA Considerations	12
8.	Security Considerations	12
9.	Acknowledgements	12
10.	References	12
10.1.	Normative References	12
10.2.	Informative References	13
	Authors' Addresses	13

Internet-Draft

kmprp

July 2011

[1.](#) Introduction

Key management today is limited to statically configuring master keys in individual routers. This document extends currently defined IKEv2 protocol to define an automated method of Key Management for Pairwise Routing Protocol (KMPPR) that allows network devices to automatically exchange key material related information between the network devices.

[2.](#) 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.

+-----+-----+ Notation Payload +-----+-----+	
AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
D	Delete
HDR	KMPPR 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

Jethanandani, et al.

Expires January 2, 2012

[Page 3]

Internet-Draft

kmprp

July 2011

[2.1.](#) 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 Diffie-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](#)].

Peer (Initiator)		Peer (Responder)
-----		-----
HDR, SAi1, KEi, Ni	-->	
	<--	HDR, SAR1, KEr, Nr, [CERTREQ,]
		RP_INIT

[2.2.](#) 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](#).

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

In the RP_AUTH exchange, the Initiator proposes one or more sets of policies for one routing protocol in the SArpi. The Responder returns the one policy contained in SArpi that it accepts. Based on this policy, appropriate keying material is derived from the existing shared keying material. At the successful conclusion of the RP_AUTH exchange, the initiator and responder have agreed upon a single set of policy and keying material for a particular routing protocol.

[2.3](#). 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](#).

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

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](#).

[2.4.](#) 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

[3.](#) 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.

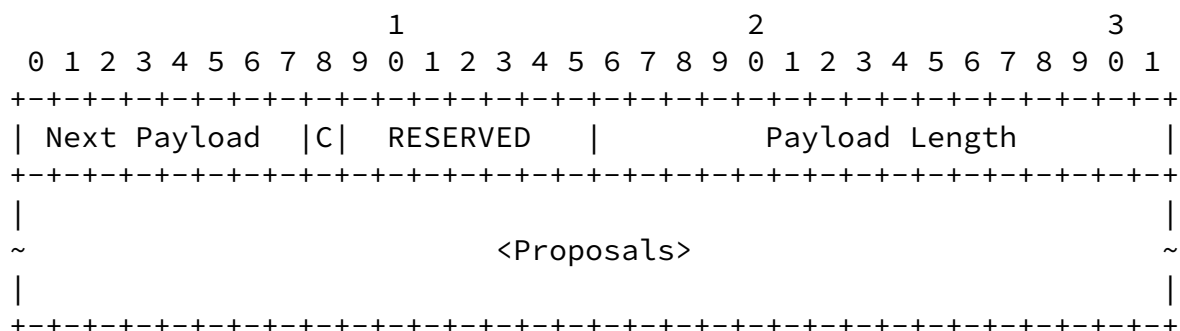
[3.1.](#) Security Association Payload

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 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.

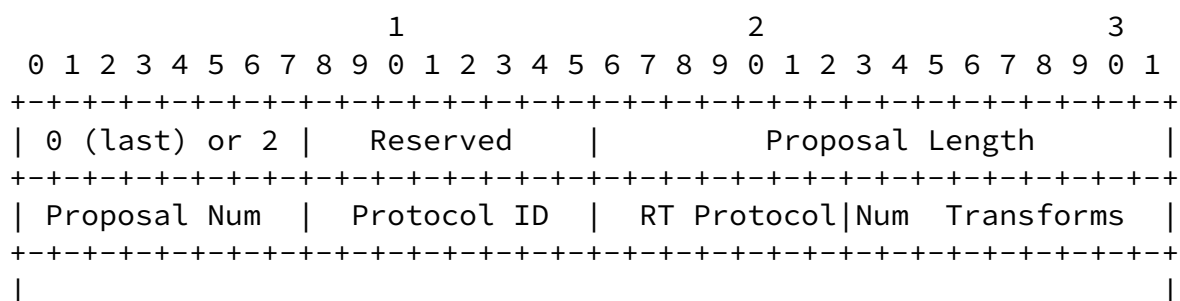


Security Association Payload

The Security Association Payload fields are defined as in [RFC 5996](#).

3.1.1. 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.



receipt.

- o Transform Length (2 octets) - The length (in octets) of the Transform Substructure including Header and Attributes.
- o 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.
- o 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](#)].

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

Authentication Algorithm

- o 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

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

```

+---+---+---+---+---+---+
|0|X|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.

Internet-Draft

kmprrp

July 2011

- o 0 (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 N_i and N_r 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).

$$\langle \text{TCP-AO master key} \rangle = \text{prf}+(SK_d, N_i \mid N_r)$$

[4.](#) Operation Details

[4.1.](#) General

KMPRRP 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 KMPRRP 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 KMPRRP neighbor adjacency between them. This is typically done by the Initiator (KMPRRP speaker) initiating a KMPRRP RP_INIT exchange mentioned in the [section 2.1](#) towards its KMPRRP 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.

[4.2.](#) 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.

[4.3.](#) 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.

[5.](#) 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.

[6.](#) 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.

[7.](#) 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.

[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.

Many protocol definitions and protocol formats come from [RFC 5996](#), either by reference or inclusion.

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.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", [RFC 5925](#), June 2010.

Jethanandani, et al. Expires January 2, 2012 [Page 12]

Internet-Draft kmprp July 2011

- [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.ietf-karp-crypto-key-table] Housley, R. and T. Polk, "Database of Long-Lived Symmetric Cryptographic Keys", [draft-ietf-karp-crypto-key-table-01](#) (work in progress), May 2011.

[I-D.ietf-karp-routing-tcp-analysis]

Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Security According to KARP Design Guide", [draft-ietf-karp-routing-tcp-analysis-00](#) (work in progress), June 2011.

[IKEV2-PARAMS]

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

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:

Jethanandani, et al. Expires January 2, 2012

[Page 13]

Internet-Draft

kmprp

July 2011

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: