Abstract

This document analyzes the threats applicable for Intermediate system to Intermediate system (IS-IS) routing protocol and security gaps according to the KARP Design Guide. This document also provides specific requirements to address the gaps with both manual and auto key management protocols.

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 September 13, 2012.

Copyright Notice

Copyright (c) 2012 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
Internet-Draft KARP IS-IS security gap analysis March 2012

described in the Simplified BSD License.

Table of Contents

1. Introduction .............................................. 3
   1.1. Requirements Language ............................... 3
   1.2. Acronyms ............................................ 3
2. Current State .............................................. 3
   2.1. Key Usage ........................................... 4
      2.1.1. Sub network Independent ......................... 4
      2.1.2. Sub network dependent ......................... 4
   2.2. Key Agility ......................................... 5
   2.3. Security Issues ...................................... 5
      2.3.1. Replay Attacks ................................. 5
      2.3.2. Spoofing Attacks ............................... 6
      2.3.3. DoS Attacks .................................... 7
   3.1. Manual Key Management ............................... 7
   3.2. Key Management Protocols ............................ 8
4. IANA Considerations ....................................... 9
5. Security Considerations ................................... 9
6. Acknowledgements ......................................... 9
7. References ................................................ 10
   7.1. Normative References ............................... 10
   7.2. Informative References ............................. 10
Authors’ Addresses ............................................ 11
1. Introduction

This document analyzes the current state of Intermediate system to Intermediate system (IS-IS) protocol according to the requirements set forth in [RFC6518] for both manual and key management protocols. With currently published work, IS-IS meets some of the requirements expected from a manually keyed routing protocol. Integrity protection is expanded with more cryptographic algorithms and also limited algorithm agility (HMAC-SHA family) is provided with [RFC5310]. Basic form of Intra-connection re-keying capability is provided by the specification [RFC5310] with some gaps as explained in Section 3.

This draft summarizes the current state of cryptographic key usage in IS-IS protocol and several previous efforts to analyze IS-IS security. This includes base IS-IS specification [RFC1195], [RFC5304], [RFC5310] and the OPSEC working group document [RFC6039]. Authors would like to acknowledge all the previous work done in the above documents.

This document also analyzes applicability of various threats as described in [ietf-karp-threats-reqs] to IS-IS, lists gaps and provides specific recommendations to thwart the applicable threats for both manual keying and for auto key management mechanisms.

1.1. 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].

1.2. Acronyms

KMP - Key Management Protocol (auto key management)
MKM - Manual Key management Protocols
NONCE - Number Once
SA - Security Association

2. Current State

IS-IS is specified in International Standards Organization (ISO) 10589, with extensions to support Internet Protocol version 4 (IPv4) described in [RFC1195]. The specification includes an authentication
mechanism that allows for any authentication algorithm and also specifies the algorithm for clear text passwords. Further [RFC5304] extends the authentication mechanism to work with HMAC-MD5 and also modifies the base protocol for more effectiveness. [RFC5310] provides algorithm agility, with new generic crypto authentication mechanism (CRYPTO_AUTH) for IS-IS. The CRYPTO_AUTH also introduces Key ID mechanism that map to unique IS-IS Security Associations (SAs).

The following sections describe the current authentication key usage for various IS-IS messages, current key change methodologies and the various potential security threats.

2.1. Key Usage

IS-IS can be provisioned with a per interface, peer-to-peer key for IS-IS HELLO PDUs (IIH) and a group key for Link State PDUs (LSPs) and Sequence number PDUs (SNPs). IIH packets also can use the group key used for LSPs and SNPs.

2.1.1. Sub network Independent

Link State PDUs, Complete and partial Sequence Number PDUs come under sub network Independent messages. For protecting Level-1 SNPs and Level-1 LSPs, provisioned Area Authentication key is used. Level-2 SNPs as well as Level-2 LSPs use the provisioned domain authentication key.

Since authentication is performed on the LSPs transmitted by an IS, rather than on the LSP packets transmitted to a specific neighbor, it is implied that all the ISes within a single flooding domain must be configured with the same key in order for authentication to work correctly. This is also true for SNP packets, though they are limited to link local scope in broadcast networks.

2.1.2. Sub network dependent

IS-IS HELLO PDUs use the Link Level Authentication key, which may be different from that of Link State PDUs (LSPs) and Sequence number PDUs (SNPs). This could be particularly true for point-to-point links. In broadcast networks it is possible to provision the same common key used for LSPs and SNPs, to protect IIH messages. This allows neighbor discovery and adjacency formation with more than one neighbor on the same physical interface.
2.2. Key Agility

Key roll over without effecting the routing protocols operation is critical for effective key management protocol integration.

Current HMAC-MD5 crypto authentication as defined in [RFC5304], suggests a transition mode, so that ISes use a set of keys when verifying the authentication value, to allow key changes. This approach will allow changing the authentication key manually without bringing down the adjacency and without dropping any control packet. But, this can increase the load on control plane for the key transition duration as each control packet may have to be verified by more than one key and also allows to mount a potential Denial of Service (DoS) attack in the transition duration.

The above situation is improved with the introduction of Key ID mechanism as defined in [RFC5310]. With this, the receiver determines the active security association (SA) by looking at the Key ID field in the incoming PDU and need not try with other keys, when the integrity check or digest verification fails. But, neither Key co-ordination across the group nor exact key change mechanism is clearly defined. [RFC5310] says: "Normally, an implementation would allow the network operator to configure a set of keys in a key chain, with each key in the chain having a fixed lifetime. The actual operation of these mechanisms is outside the scope of this document."

2.3. Security Issues

The following section analyzes various security threats possible, in the current state for IS-IS protocol.

2.3.1. Replay Attacks

Replaying a captured protocol packet to cause damage is a common threat for any protocol. Securing the packet with cryptographic authentication information alone can not mitigate this threat completely.

In intra-session replay attacks a secured protocol packet of the current session is replayed, can cause damage, if there is no other mechanism to confirm this is a replay packet. In inter-session replay attacks, captured packet from one of the previous session can be replayed to cause the damage. IS-IS packets are vulnerable to both these attacks, as there is no sequence number verification for IIH packets and SNP packets. Also with current manual key management periodic key changes across the group are done rarely. Thus the intra-connection and inter-connection replay requirements are not met.
IS-IS specifies the use of the HMAC-MD5 [RFC5304] and HMAC-SHA-1 family in [RFC5310], to protect IS-IS packets. An adversary could replay old IIHs or replay old SNPs that would cause churn in the network or bring down the adjacencies.

1. At the time of adjacency bring up an IS sends IIH packet with empty neighbor list (TLV 6) and with the authentication information as per provisioned authentication mechanism. If this packet is replayed later on the broadcast network, all ISes in the broadcast network can bounce the adjacency to create a huge churn in the network.

2. Today Link State PDUs (LSPs) have intra-session replay protection as LSP header contains 32-bit sequence number which is verified for every received packet against the local LSP database. But, if the key is not changed, an adversary can cause an inter-session replay attack by replaying a old LSP with higher sequence number and fewer prefixes or fewer adjacencies. This forces the receiver to accept and remove the routes from the routing table, which eventually causes traffic disruption to those prefixes.

3. In point-to-point (P2P) networks, if a old Complete Sequence Number packet (CSNP) is replayed this can cause LSP flood in the network. Similarly a replayed Partial Sequence Number packet (PSNP) can cause LSP flood in the broadcast network.

2.3.2. Spooﬁng Attacks

IS-IS shares the same key between all neighbors in an area or in a domain to protect the LSP, SNP packets and in broadcast networks even IIH packets. False advertisement by a router is not within scope of the KARP work. However, given the wide sharing of keys as described above, there is a signiﬁcant risk that an attacker can compromise a key from one device, and use it to falsely participate in the routing, possibly even in a very separate part of the network. Possession of the key itself is used as authentication check and there is no identity check separately. Spooﬁng occurs when an illegitimate device assumes the identity of a legitimate one. An attacker can use spooﬁng as a means for launching various types of attacks. For example:

1. The attacker can send out unrealistic routing information that might cause the disruption of network services such as block holes.

2. A rogue system having access to the common key used to protect the LSP, can send an LSP, setting the Remaining Lifetime ﬁeld to zero, and ﬂooding it thereby initiating a purge. Subsequently,
this also can cause the sequence number of all the LSPs to increase quickly to max out the sequence number space, which can cause an IS to shut down for MaxAge + ZeroAgeLifetime period to allow the old LSPs to age out in other ISes of the same flooding domain.

2.3.3. DoS Attacks

Denial-of-service (DoS) attacks using the authentication mechanism is possible and an attacker can send packets which can overwhelm the security mechanism itself. An example is initiating an overwhelming load of spoofed but integrity protected protocol packets, so that the receiver needs to process the integrity check, only to discard the packet. This can cause significant CPU usage. DoS attacks are not generally preventable with in the routing protocol. As the attackers are often remote, the DoS attacks are more damaging to area-scoped or domain-scoped packet receivers than link-local scoped packet receivers.

3. Gap Analysis and Security Requirements

This section outlines the differences between the current state of the IS-IS routing protocol and the desired state as specified in KARP Design Guidelines [RFC6518]. The section focuses on where IS-IS protocol fails to meet general requirements as specified in the threats and requirements document.

This section also describes security requirements that should be met by IS-IS implementations that are secured by manual as well as auto key management protocols.

3.1. Manual Key Management

1. With CRYPTO_AUTH specification [RFC5310], IS-IS packets can be protected with HMAC-SHA family of cryptographic algorithms. The specification provides the limited algorithm agility (SHA family). By using Key IDs, it also conceals the algorithm information from the protected control messages.

2. Even though both intra and inter session replay attacks are best prevented by deploying key management protocols with frequent key change capability, basic constructs for sequence number should be there in the protocol messages. So, some basic or extended sequence number mechanism should be in place to protect IIH packets and SNP packets. The sequence number should be increased for each protocol packet. This allows mitigation of some of the replay threats as mentioned in Section 2.3.1.
3. Any common key mechanism with keys shared across a group of routers is susceptible to spoofing attacks caused by a malicious router. Separate authentication check (apart from the integrity check to verify the digest) with digital signatures as described in [RFC2154], can effectively nullify this attack. But this approach was never deployed and one can only assume due to operational considerations at that time. The alternative approach to thwart this threat would be by using the keys from the group key management protocol. As the group key(s) are generated by authenticating the member ISes in the group first, and then periodically rekeyed, per packet identity or authentication check may not be needed.

4. In general DoS attacks may not be preventable with mechanism from routing protocols itself. But some form of Admin controlled lists (ACLs) at the forwarding plane can reduce the damage. There are some other forms the DoS attacks common to any protocol are not in scope as per the section 2.2 in [I-D.ietf-karp-threats-reqs].

As discussed in Section 2.2, though Key ID mechanism in [RFC5310] helps, better key co-ordination mechanism for key roll over is desirable even with manual key management. But, it fell short of specifying exact mechanism other than using key chains. The specific requirements:

a. Keys SHOULD be able to change without affecting the established adjacency and even better without any control packet loss.

b. Keys SHOULD be able to change without effecting the protocol operations, for example, LSP flooding should not be help for a specific Key ID availability.

c. Any proposed mechanism SHOULD also be further incrementally deployable with key management protocols.

3.2. Key Management Protocols

In broadcast deployments, the keys used for protecting IS-IS protocols messages can, in particular, be group keys. A mechanism, similar to as described in [I-D.wels-gdoi-mac-tek] can be used to distribute group keys to a group of ISes in Level-1 area or Level-2 domain, using GDOI [I-D.ietf-msec-gdoi-update]. There are also similar approaches with IKEv2 ([RFC5996]) based GDOI, to routing protocols as described in [I-D.hartman-karp-mrkmp].

If a group key is used, the authentication granularity becomes group membership of devices, not peer authentication between devices.
Group key management protocol deployed SHOULD be capable of supporting rekeying support.

In some deployments, where IS-IS point-to-point (P2P) mode is used for adjacency bring-up, sub network dependent messages (IIHs) can use a different key shared between the two point-to-point peers, while all other messages use a group key. When group keying mechanism is deployed, even the P2P IIHs can be protected with the common group keys. This approach facilitates one key management mechanism instead of both pair-wise keying and group keying protocols to be deployed together.

As mentioned earlier, effective key change capability with in the routing protocol which allows key roll over without impacting the routing protocol operation, is one of the requirements for deploying any group key mechanism. Once such mechanism is in place with deployment of group key management protocol, IS-IS can be protected from various threats not limited to intra and inter session replay attacks and spoofing attacks.

Specific use of crypto tables [I-D.ietf-karp-crypto-key-table] should be defined for IS-IS protocol.

4. IANA Considerations

This document defines no new namespaces.

5. Security Considerations

This document is mostly about security considerations of IS-IS protocol, lists potential threats and security requirements for solving those threats. This document does not introduce any new security threats for IS-IS protocol. For more detailed security considerations please refer the Security Considerations section of the KARP Design Guide [RFC6518] document as well as KARP threat document [I-D.ietf-karp-threats-reqs]

6. Acknowledgements

Authors would like to thank Joel Halpern for encouraging us to come up with this document and giving valuable review comments.

7. References

7.1. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and

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

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R.,
and M. Fanto, "IS-IS Generic Cryptographic

7.2. Informative References

[I-D.hartman-karp-mrkmp]
Hartman, S., Zhang, D., and G. Lebovitz, "Multicast Router
Key Management Protocol (MaRK)",
draft-hartman-karp-mrkmp-02 (work in progress), July 2011.

[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-threats-reqs]
Lebovitz, G., Bhatia, M., and R. White, "The Threat
Analysis and Requirements for Cryptographic Authentication
of Routing Protocols’ Transports",
draft-ietf-karp-threats-reqs-03 (work in progress),
June 2011.

[I-D.ietf-msec-gdoi-update]
Weis, B., Rowles, S., and T. Hardjono, "The Group Domain
of Interpretation",
draft-ietf-msec-gdoi-update-11 (work in progress),
August 2011.

[I-D.weis-gdoi-mac-tek]
Weis, B. and S. Rowles, "GDOI Generic Message
Authentication Code Policy",
draft-weis-gdoi-mac-tek-02
(work in progress), March 2011.

[RFC2154] Murphy, S., Badger, M., and B. Wellington, "OSPF with

Key Management", BCP 107, RFC 4107, June 2005.


Authors’ Addresses

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

Phone: 408 750-5678
Email: uma.chunduri@ericsson.com

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

Phone: 408 750-5210
Email: albert.tian@ericsson.com

Wenhu Lu
Ericsson Inc.,
300 Holger Way,
San Jose, California  95134
USA

Email: wenhu.lu@ericsson.com
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. A Gatekeeper (GK) mechanism is introduced to allow TCP-AO to coordinate with IKEv2 without fundamental modification to either. This document also introduces extensions to IKEv2 and its Security Associations to enable its key negotiation to support TCP-AO. The document also includes a summary of IKEv2 authentication methods available for peer authentication for use in protecting routing protocols.

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 September 13, 2012.

Copyright Notice

Copyright (c) 2012 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. Requirements Language .................................... 3
   1.2. Acronyms ................................................. 4
2. Motivation and Overview ........................................ 4
3. The Gatekeeper ................................................. 5
   3.1. RP interface to the Gatekeeper ............................ 6
   3.2. KMP interface to Gatekeeper ............................... 7
   3.3. TCP-AO interface to Gatekeeper ............................ 8
4. Extensions required for IKEv2 ................................. 8
   4.1. Non IPsec DOI ........................................... 9
      4.1.1. Security Association Extensions ....................... 9
   4.2. Simple Traffic Selectors Negotiation ..................... 10
5. IANA Considerations ........................................... 10
6. Security Considerations ....................................... 10
7. Acknowledgements ............................................. 11
8. Appendix A .................................................... 11
   8.1. BGP Multi Session and transport level differentiation .. 11
   8.2. Applicable Authentications methods ....................... 12
      8.2.1. Symmetric key based authentication ................... 12
      8.2.2. Asymmetric key based authentication .................. 12
      8.2.3. EAP based authentication ............................ 13
9. References .................................................... 14
   9.1. Normative References .................................... 14
   9.2. Informative References .................................. 14
Authors’ Addresses .............................................. 16
1. Introduction

A security threat analysis for TCP-based routing protocols (BGP [RFC4271], PCEP [RFC5440], MSDP [RFC3618] and LDP [RFC5036]) is detailed in [ietf-karp-routing-tcp-analysis]. The KARP design guide [RFC6518] suggests various requirements and options for obtaining keys to protect the routing protocols and recommends using a Key Management Protocol (KMP) to automate key establishment, as well as rekeying to continuously protect the routing protocols.

This document analyzes the TCP-based pairwise Routing Protocol (RP) requirements needed to integrate the IKEv2 [RFC5996] KMP together with TCP-AO [RFC5925] to protect routing protocols.

This document introduces a new Gatekeeper module, which provides a common interface and minimizes the changes for all routing protocols (BGP, PCEP, MSDP and LDP) to be integrated with KMP. The Gatekeeper modules does the SA management and interaction with KMP as well as TCP-AO protocol. The purpose of the Gatekeeper is to act as a shim between IKEv2 and TCP-AO, so that TCP-AO and the Gatekeeper together act like IPsec to IKEv2 (since IKEv2 is designed to tightly interact with IPsec). This document defines this common interface between all TCP-based pairwise routing protocols with Gatekeeper in [RFC5996]. Gatekeeper interface with IKEv2 KMP and the TCP-AO for session parameter negotiation, key establishment and rekeying is also defined in [RFC5996].

Currently IKEv2 can establish only Security Association (SA) for IPsec. A few extensions are needed for IKEv2 to establish SA for TCP-based routing protocols when TCP-AO is used for protection. Section 4 discusses the summary of extensions required for IKEv2 protocol for key establishment, traffic selectors negotiation and SA establishment to support the keying and parameters needed by TCP-AO.

One of the services provided by IKEv2 KMP is peer authentication. This happens before traffic keys are established between IKEv2 peers. As IKEv2 KMP provides a variety of authentications methods; Section 8.2 discusses various Symmetric, Asymmetric and EAP based KMP authentication options available. The goal of Section 8.2 is to summarize vastly scattered information for choosing the right authentication method by operators for peer authentication with low operational overhead and yet secure mechanism especially suitable for routing environments.

1.1. 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].

1.2. Acronyms

BGP - Border Gateway Protocol
EAP - Extensible Authentication Protocol
GKR - Gatekeeper Record
IKEv2 - Internet Key Exchange Protocol Version 2
IPsec - Security Architecture for the Internet Protocol
KDF - Key Derivation Function
KMP - Key Management Protocol (auto key management)
LDP - Label Distribution Protocol
MKT - Master Key Tuples as defined in TCP-AO
MSDP - Multicast Source Discovery Protocol
PCEP - Path Computation Element Communication Protocol
RP - Routing Protocol
SA - Security Association
TCP-AO - TCP Authentication Option

2. Motivation and Overview

IKEv2 assumes IPsec triggers new SA requests, manages SA timers and rekeys SAs as needed. TCP-AO assumes an external key manager, which could support functions like Master key triggering, SA timers, and rekey triggering to get all the parameters required including Master key to protect the TCP session. To bridge the gap between IKEv2 and TCP-AO, this document defines a Gatekeeper module as described in Section 3.

The motivation of this document is to offload Security Association (SA) management and to provide a generic and common interface for all TCP-based RPs to integrate with KMPs in general and specifically with
IKEv2 KMP.

The following diagram depicts the Gatekeeper module interfaces with all protocols involved i.e., TCP-based RPs, IKEv2 KMP, and TCP-AO.

```
+-------------+ RP Session Configuration +-------------+
| TCP Based RPs |                        | TCP-AO     |
| (BGP/LDP/PCEP |                        |            |
| /MSDP)       |                        |            |
+-------------+                        +-------------+
                   |                       |
                   | Gatekeeper             |
                   |                        |
                   |                        |
                   |  Trigger Negotiated    |
                   |  Parameters            |
                   |<------------------------|
                   |                        |
                   | MKT Provisioning       |
                   |------------------------|
                   +-----------------------+

Figure 1: KARP KMP: Using IKEv2 with TCP-AO
```

In Figure 1, before initiating the TCP connection, all TCP-based RPs communicate the provisioned configuration to Gatekeeper module. The Gatekeeper then issues a corresponding request with all the proposed alternatives at the RP to the IKEv2 KMP, so IKEv2 can negotiate the needed parameters. These negotiated parameters are used to provision MKTs in the TCP-AO, as well as to establish timers and other needed state local to the Gatekeeper. Gatekeeper then maintains the KMP SAs and initiates rekey triggers as needed to provision new MKTs for the long-lived TCP sessions protected by TCP-AO. The Gatekeeper installs these new keys in TCP-AO consistent with TCP-AO’s support for key changes.

3. The Gatekeeper

TCP-AO has a different model of security associations and key management than IPsec. IKEv2 is designed to support IPsec’s model. This document introduces the Gatekeeper to enable IKEv2 to support key and parameter negotiation that can be used in TCP-AO, as identified in Section 2.
The Gatekeeper maintains a Gatekeeper record (GKR) to keep track of TCP-AO MKTs. For long-lived TCP connections MKTs can be rolled over by rekeying creating new MKTs and installing them in TCP-AO. The GKR can be viewed as a superset of MKT; it also maintains and tracks the lifetime of the provisioned MKT, and includes other per-connection parameters needed by TCP-AO, such as algorithm, key length, etc. [RFC5926].

The next section defines the Gatekeeper module interface between TCP-based RPs (BGP, LDP, MSDP, PCEP), interface with IKEv2 and TCP-AO.

3.1. RP interface to the Gatekeeper

When a routing protocol is configured to use TCP-AO with KMP (by not specifying the keys or through some other means), TCP connection identifiers, all configured Message Authentication Code (MAC) algorithms, all configured Key Derivation Function (KDF) parameters, rekey lifetime and the TCP option flag (i.e., all additional parameters specified in [RFC5926]) are provisioned in the Gatekeeper record.

If the same routing protocol (RP) needs differentiate transport sessions to differently securing separate TCP connections between the same endpoints, TCP connection identifiers either the full socket pair (i.e., local IP address, remote IP address, local TCP port, and remote TCP port) or partial socket pair values (as indicated with wildcards) need to be provisioned. GKR s SHOULD thus support full or partial socket pair specification.

In general, a full socket pair is not needed for negotiating the TCP-AO MKT with KMP. As specified in Section 3.1 of TCP-AO [RFC5925], socket pair values can be partially specified using ranges, masks, wildcards, or any other suitable indication. These provisioned socket pair parameters are supplied to KMP as context in which to negotiate traffic selectors for which the MKT or Master key should be used in TCP-AO.

For more details on cases where a full socket pair is needed before opening the connection, please refer Section 8.1. Provisioning of the Gatekeeper record SHOULD be done before opening the TCP connection. From the RP interface, the record created in Gatekeeper only contains only the RP’s connection information, and this information is given to KMP (IKEv2) to obtain the negotiated parameters to provision the MKT to protect the underlying TCP session by [RFC5925].
3.2. KMP interface to Gatekeeper

IKEv2 expects an external trigger that contains the information required to negotiate security associations. There needs to be a way to trigger the KMP to initiate negotiation with all the provisioned parameters of a Gatekeeper record by a TCP-based RP. A similar trigger is also required to rekey, to maintain the negotiated SAs for long-lived connections. The purpose of this section is to define a common interface between the Gatekeeper and the IKEv2 KMP that can be used by all TCP-based RPs.

The following are the details of the interface between KMP and GK:

1. At the time of a new connection, a trigger to the KMP occurs to negotiate the session-specific parameters with the needed information on MAC algorithm, KDF parameter, Traffic Selectors, and the TCP option flag from the Gatekeeper record. The GK at the peer is expected to have similar provisioning in place for responding to the received KMP request.

2. A KMP session identifier, provided by a successful key negotiation by the KMP, needs to be stored and should be used when the Gatekeeper make decision based on the lifetime to rekey the existing session.

3. MKT IDs (as specified in Section 3.1 of TCP-AO [RFC5925]) require a SendID and a RecvID for each MKT, mutually agreed by the connection endpoints. These 1-byte quantities need to be negotiated by the KMP with the peer to populate in the MKT.

4. A KMP-negotiated MAC algorithm, MKT connection identifiers (negotiated traffic selectors) and optionally lifetime for traffic keys for each session, need to be populated in MKT.

5. KMP-negotiated KDF parameters for each session used to generate traffic keys from master keys to be populated in MKT.

6. IKEv2 does not negotiate rekey lifetime and rekeying is based on local operator policy. The Gatekeeper adds this capability, tracking the key lifetime provisioned at TCP-based RP and explicitly triggering the KMP to rekey when indicated. This rekey trigger then creates a new MKT for the underlying TCP connection. Implementations can proactively negotiate a new MKT Master Key before the lifetime of the current Master key expires.
3.3. TCP-AO interface to Gatekeeper

TCP-AO expects an external entity to provision its MKTs in order to protect TCP sessions. The Gatekeeper module provides this function so that all TCP-based RPs can benefit from this common interface.

The following are the details of the interface between TCP-AO and the GK:

1. After getting the negotiated parameters and mutually authenticated Master keys from the KMP, the Gatekeeper inserts a corresponding MKT and parameters into TCP-AO. The session-specific parameters include negotiated Connection identifiers, MAC algorithms, KDFs, KeyIDs, the TCP option flag and the Master Key given by the KMP.

2. MKT IDs (as specified in Section 3.1 of TCP-AO [RFC5925]) require a SendID and a RecvID for each MKT, which are mutually agreed by the connection endpoints. These 1-byte quantities need to be part of the MKT when the KMP key(s) are populated in MKT.

3. For long-lived TCP sessions, the Gatekeeper removes the old MKTs from TCP-AO after rekeying the corresponding new MKTs, to continuously protect the underlying TCP sessions.

4. In general, restarted TCP sessions can use existing MKT in TCP-AO i.e., IKEv2 need not be retriggered, since new key and parameter negotiation is not needed due to the protection already provided by TCP-AO (refer Section 5.3.1 of TCP-AO [RFC5925]). However, if GKR and hence TCP-AO MKT is created with full socket pair (in other words without using ranges, masks, wildcards for socket pair values, for the cases as specified in Section 8.1), then IKEv2 needs to be retriggered to get the new master key for the corresponding restarted TCP session.

4. Extensions required for IKEv2

There can be two ways to derive a KMP that is suitable for TCP-based routing protocols:

a. Create a new KMP for routing protocols, e.g., based on IKEv2 (as proposed in [mahesh-karp-rkmp]).

b. Extend IKEv2 to be suitable for TCP-based routing protocols.

In this section, we would like to explore option (b).
This section summarizes the extensions required for IKEv2 to negotiate non-IPsec SAs for TCP-based routing protocols. The authors acknowledge that some of the items below are already discussed in KARP WG, but the details presented here are different.

Routing protocols that use this extended IKEv2 KMP can continuously benefit from the new authentication methods and any other new features which might be added to [RFC5996].

4.1. Non IPsec DOI

IKEv2 is designed for performing mutual authentication with the peers and establishing and maintaining Security Associations for IPsec. IKEv2 defines the IKE_AUTH and CREATE_CHILD_SA exchanges, consisting of payloads and processing guidelines for IPsec Domain of Interpretation (DOI); this need to be generalized to exchange other protocol-specific parameters to be useful for RPs.

IKEv2 is designed to be extensible with additional parameters. The extensions proposed here can be deployed within that context, running over the existing IKEv2 port number and using existing IKEv2 tunneling mechanisms where needed.

The current IKEv2 CREATE_CHILD_SA exchange can be used to rekey the IKE SA and the master key. This document does not propose any changes or extensions to re-establishing IKE SA through the CREATE_CHILD_SA exchange.

4.1.1. Security Association Extensions

The IKEv2 Security Association (SA) payload is used to negotiate attributes of a Security Association. This payload contains multiple proposals, as configured in the routing protocol. Possible extensions to be made are:

1. A new Protocol ID, to be added in the proposal substructure with TCP-AO as new protocol (IANA-TBD).

2. An Integrity Algorithm (INTEG), as defined in the transform substructure needs to be mandated for the new TCP-AO Protocol.

3. Authentication algorithms and associated parameters as defined in [RFC5926] should be extended to the current list in IKEv2 Transform Type 3 (Integrity Algorithm), for TCP-AO usage (IANA-TBD).

4. The Diffie-Hellman group (D-H) transform type can be used for TCP-AO proposal as an optional transform.
5. A new transform type is needed to represent the KDF for traffic key derivation by TCP-AO (IANA-TBD) and also needs to be mandated for the new TCP-AO Protocol. KDF algorithms and associated parameters as defined in [RFC5926] should be listed for this new transform in IKEv2 for TCP-AO usage (IANA-TBD).

6. A new transform type is needed to represent the TCP-AO KeyIDs (IANA-TBD). The Initiator KeyID represents the SendID and the Responder KeyID represents the RecvID in the TCP-AO MKT.

7. A new transform type needs to be created to indicate TCP options coverage by TCP-AO (IANA-TBD).

8. The valid transform types (as defined in Section 3.3.3 of [RFC5996]) for TCP-AO with mandatory and optional types need to be listed.

9. Attribute negotiation rules need to be extended for TCP-AO protocol.

4.2. Simple Traffic Selectors Negotiation

The Traffic Selectors defined in IKEv2 [RFC5996] have huge potential to negotiate the particular traffic to be secured, agreeable to both initiator and responder. For a routing protocol SA, traffic selectors negotiation presents a simple case and does not require any changes. A single connection or multiple connections with different source ports can be negotiated with a single CREATE_CHILD_SA exchange. The IP Protocol ID in the traffic selector field (as defined in Section 3.13.1 of [RFC5996]) can always be TCP for the routing protocol SAs.

The above is an attempt to summarize the brief list of changes in this approach and this section will be revisited.

5. IANA Considerations

This document requests that IANA to allocate new parameters as described in Section 4.1.1.

6. Security Considerations

This document does not introduce any new security threats for IKEv2 [RFC5996] or TCP-AO [RFC5925]. For more detailed security considerations please refer the Security Considerations section of the KARP Design Guide [RFC6518] document as well as KARP threat
7. Acknowledgements

The authors would like to thank Joel Halpern for his initial discussions and providing feedback on the document. The authors also thank Tero Kivinin and Dan Harkins for reviewing the document and Ron Bonica for his initial requirement discussions. Thanks to Sam Hartman for his KARP working group discussions on this topic.

The Gatekeeper module is originally proposed by Joe Touch.

8. Appendix A

8.1. BGP Multi Session and transport level differentiation

[ietf-idr-bgp-multisession] describes MP-BGP, which uses multiple TCP sessions between a pair of BGP speakers. Each TCP session is used to exchange routes related by some session-based attribute, such as AFI/SAFI. The reason transport level distinction is required could be because of operator policy. Though it is less likely to see different MAC/KDF parameters for each of these sessions, it is possible rekey lifetimes or TCP option flags for TCP-AO can be different for each of these AFI/SAFI based sessions.

If transport level separation is required for all sessions between a pair of BGP speakers, a unique and full socket pair (i.e., a local IP address, a remote IP address, a local TCP port, and a remote TCP port) MUST be known before establishing a TCP connection. The full socket pair is required for both unique MKT creation in TCP-AO, as well as for the KMP to negotiate unique Master keys for each connection.

The use of different IP addresses to differentiate connections in multi session BGP is discouraged in [ietf-idr-bgp-multisession] and the destination port is always BGP. As a result, the only option for transport level differentiation is by knowing the source port of the connection being initiated. This is required to negotiate unique KMP SAs by the Gatekeeper, as well as to configure unique TCP-AO MKTs for each TCP connection. How source port lock-down is done is beyond the scope of this document (this is an implementation issue) and this can be achieved in many different ways before making the TCP connection.

The Gatekeeper interface, defined in Section 3, is oblivious to this issue and can well accommodate this requirement.
8.2. Applicable Authentications methods

One advantage that IKEv2 provides is the largest selection of key management and parameter coordination authentication methods suitable for various environments. The goal of this section is to look at various KMP authentication options available and recommend suitable options for use in negotiating keys and other parameters for routing protocol protection.

As some of the authentication mechanisms are optional in IKEv2, one mandatory authentication mechanism from the list below needs to be selected for routing environments to ensure inter-operability and quicker adoption. This section attempts to summarize the available options and constraints surrounding the options.

8.2.1. Symmetric key based authentication

IKEv2 [RFC5996] allows for authentication of the IKEv2 peers using a symmetric pre-shared key. For symmetric pre-shared key peer authentication, deployments need to consider the following as per [RFC5996):

1. Deriving a shared secret from a password, name, or other low-entropy source is not secure. These sources are subject to dictionary and social-engineering attacks, among others.

2. The pre-shared key should not be derived solely from a user-chosen password without incorporating another source of randomness.

3. If password-based authentication is used for bootstrapping the IKE_SA, then one of the EAP methods as described in Section 8.2.3 needs to be used.

One of the IPsecME WG charter goals is to provide IKEv2 [RFC5996] a secure password authentication mechanism which is protected against off-line dictionary attacks, without requiring the use of certificates or Extensible Authentication Protocol (EAP), even when using the low-entropy shared secrets. There are couple of documents which try to address this issue and the work is still in progress.

8.2.2. Asymmetric key based authentication

Another peer authentication mechanism for IKEv2 uses is asymmetric key certificates or public key signatures. This approach relies on a Public Key Infrastructure using X.509 (PKIX) Certificates. If this can be deployed for IKEv2 peer authentication, it will be one of the most secure authentication mechanisms. With this authentication
Apart from RSA and DSS digital signatures for public key authentication provided by IKEv2, [RFC4754] introduces Elliptic Curve Digital Signature Algorithm (ECDSA) signatures. ECDSA provides additional benefits including computational efficiency, small signature sizes, and minimal bandwidth compared to other available digital signature methods.

8.2.3. EAP based authentication

In addition to supporting authentication using shared secrets and public key signatures, IKEv2 also supports authentication based on the Extensible Authentication Protocol (EAP), defined in [RFC3748]. EAP is an authentication framework that supports multiple authentication mechanisms. IKEv2 provides EAP authentication because public key signatures and shared secrets are not flexible enough to meet the requirements of many deployment scenarios. For KARP KMP, EAP-Only Authentication in IKEv2 as specified in [RFC5998] can be explored.

By using EAP, IKEv2 KMP can leverage existing authentication infrastructure and credential databases, because EAP allows users to choose a method suitable for existing credentials. Routing protocols today use password-based pre-shared keys to integrity protect the routing protocol messages. The same pre-shared key can be used to bootstrap the KMP and as a potential authentication key in KMP. With appropriate password based EAP methods, stronger keys can be generated without using certificates.

For authenticating the nodes running routing protocols, EAP and the IKEv2 endpoints are co-located (so no separate EAP server required). When EAP is deployed, authenticating the IKEv2 responder using both EAP and public key signatures could be redundant. EAP methods that offer mutual authentication and key agreement can be used to provide responder authentication in IKEv2 completely based on EAP.

Section 4 of [RFC5998] lists safe EAP methods to support EAP_ONLY_AUTHENTICATION. For routing protocols deployment, because an EAP server is co-located with IKEv2 responder, channel binding capability of the selected EAP method is irrelevant. Various qualified mutual authentication methods are listed in [RFC5998]; of these, a password based methods [RFC4746], [RFC5931], [RFC6124] can offer potential EAP alternative for pre-shared key only authentication.

From the list above, Encrypted Key Exchange (EKE) as described in
[RFC6124] is relatively lightweight and provides mutual authentication. This method also offers secure and robust authentication, even with an operator provisioned weak password in the presence of a strong adversary.

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

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

Phone: +1 (408) 750-5678
Email: uma.chunduri@ericsson.com

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

Phone: +1 (408) 750-5210
Email: albert.tian@ericsson.com

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

Phone: +1 (310) 448-9151
Email: touch@isi.edu
Multicast Router Key Management Protocol (MaRK)
draft-hartman-karp-mrkmp-04

Abstract

Several routing protocols engage in one-to-many communication. In order to authenticate these communications using symmetric cryptography, a group key needs to be established. This specification defines a group protocol for establishing and managing such 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 September 13, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.
# Table of Contents

1. Introduction .................................................. 4  
   1.1. Terminology ............................................. 4  
   1.2. Relationship to IKEv2 .................................. 4  
   1.3. Relationship to GDOI .................................... 5  
2. Overview ....................................................... 5  
   2.1. Types of Keys ............................................ 5  
      2.1.1. Key Encryption Key ................................. 6  
      2.1.2. Protocol Master Keys .............................. 7  
   2.2. GCKS Election .......................................... 8  
   2.3. Initial Exchange ....................................... 9  
   2.4. Group Join Exchange ................................... 9  
   2.5. Group Key Management .................................. 10  
3. GCKS Election .................................................. 10  
   3.1. A new GCKS is Elected .................................. 12  
      3.1.1. Parameters, Timers, and Events .................. 12  
      3.1.2. Initial ............................................. 14  
      3.1.3. Validate ........................................... 15  
      3.1.4. GCKS2 ............................................... 16  
      3.1.5. GCKS ............................................... 17  
      3.1.6. Member ............................................. 18  
      3.1.7. Follower ........................................... 18  
   3.2. Merging Partitioned Networks ............................ 19  
   3.3. Operations on Receiving a Packet ....................... 20  
4. Key Download Payload ........................................... 21  
5. Initial Exchanges Details ..................................... 21  
6. Group Management Unicast Exchanges ............................ 22  
   6.1. Group Join Exchange ................................... 22  
7. Group Key Management Operation ................................ 22  
   7.1. General operation ..................................... 22  
   7.2. Out of Sequence Space .................................. 22  
   7.3. Changing the Active GCKS ............................... 23  
   7.4. Reboot Cases .......................................... 23  
8. Interface to Routing Protocol .................................. 23  
   8.1. Joining a Group ........................................ 23  
   8.2. Priority Adjustment ................................... 24  
   8.3. Leaving a Group ....................................... 24  
   8.4. Out of Sequence Space .................................. 24  
9. Security Considerations ....................................... 24  
10. Acknowledgements ............................................ 26  
11. References .................................................. 26  
   11.1. Normative References .................................. 26  
   11.2. Informative References ............................... 26  
Authors’ Addresses ............................................... 27
1. Introduction

Many routing protocols such as OSPF [RFC2328] and IS-IS [RFC1142] use a one-to-many or multicast model of communications. The same message is sent to a number of recipients.

These protocols have cryptographic authentication mechanisms that use a key shared among all members of a communicating group in order to protect messages sent within that group. From a security standpoint, all routers in a group are considered equal. Protecting against a misbehaving router that is part of the group is out of scope for this protocol.

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.

The members of a group elect a Group Controller/Key Server (GCKS). Potentially any member of the group may act as a GCKS. Since protecting against misbehaving routers is out of scope, there is no need to protect against an entity that is not currently the GCKS impersonating the GCKS.

To prove membership in the group, a router authenticates using its provisioned credentials to the current GCKS. If successful, the router is given the current key material for the group. Group size is relatively small and need for forced eviction of members is rare. If a GCKS needs to evict a member, then it can simply re-authenticate with the existing members and provide them new key material.

1.1. Terminology

GCKS (Group Controller/Key Server): a GCKS is a particular group member which establishes security associations among other authorized group members which it serves.

group: a group specified in this document is a set of routers, called group members, which are located on a single broadcast domain/ link/ NBMA segment and use a one-to-many or multicast model of communication.

1.2. Relationship to IKEv2

IKEv2 [RFC4306] 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.
IKEv2 is expanded to support authentication of routers in [I-D.mahesh-karp-rkmp]. That specification describes how IKEv2 can be used for unicast routing protocols. This specification is part of expanding that work to cover multicast routing.

1.3. Relationship to GDOI

[RFC3547] provides a protocol that is structurally very similar to this one. As specified, IKE can be used to provide phase 1 authentication to a GCKS. After that, GDOI provides phase 2 messages to establish key-encryption keys and traffic keys. After the phase 2 exchange, additional key management operations can be accomplished via GDOI messages sent within the group.

In [I-D.yeung-g-ikev2] a group management approach is defined for IKEv2. This approach is extended in [I-D.tran-karp-mrmp] to provide for management of routing messages. This specification acts as a companion to that specification, providing an election protocol and some of the interactions with routing protocols.

2. Overview

2.1. Types of Keys

MaRK manipulates several different types of symmetric keys:

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

Peer key management key: Routers share a key with the GCKS that is a result of the RP_INIT exchange.

KEK (Key Encryption Key): A KEK is a key used to encrypt group key management messages to the current members of a group. A KEK is learned as the product of establishing an MaRK association or through a group key management message encrypted in a previous KEK. A KEK has an explicit expiration but may also be retired by a message encrypted in the KEK sent by the GCKS.

Protocol master key: A protocol master key is the key exported by MaRK for use by a routing protocol such as OSPF or IS-IS. The Protocol master key is the key that would be manually configured if a routing protocol is used without key management. This key is distinguished from the ‘transport key’ (see next) in that this
Protocol Master Key may be used in a cryptographic operation in order to derive a specific transport key.

Transport key: A transport key is the key used to integrity protect routing messages in a protocol such as IS-IS or OSPF. In today’s routing protocol cryptographic authentication mechanisms the transport key is the same as the protocol master key. A disadvantage of this approach is that replay prevention is challenging with this design. Ideally some key derivation step would be used to establish a fresh transport key among all the participants in the group.

2.1.1. Key Encryption Key

When a router wishes to join a group, the router performs the RP_INIT and RP_AUTH exchange with a GCKS. If the exchanges are successful, the router can establish an association with a specific group. Part of that association will be delivery of a KEK and associated parameters.

Group key management messages are sent to a group address rather than unicast to an individual peer. The authenticity, integrity and confidentiality of group key management messages need to be protected with the KEK.

As part of establishing the association, the router joining the group is given an valid period (which is identified by a start time point and an expire time point) for the KEK. A group key management message may establish a new KEK with new parameters.

From time to time, a GCKS may wish to either force early expiration of a KEK or allow a KEK to expire. Protocol master keys are permitted to be valid for somewhat longer than the KEK that created them so as to avoid disrupting routing when this happens. When a KEK is retired or expires without being replaced by a new KEK announced in the old KEK, the group members delete that KEK. Unless local policy configuration dictates otherwise, the group member will perform a new initial exchange to the GCKS in order to establish a new KEK. This solution is useful for enforcing "forward security" in the cases where a router is no longer authorized to be part of the group. That is, only valid group members can obtain the new KEK while the ones which have leaven the group will be rejected.

Other mechanisms such as LKH (section 5.4 [RFC2627]) could be used to permit removal of a group member while avoiding new initial authentications. However these mechanisms come at a complexity cost that is not justified for a small number of routers participating in a single multicast link.
2.1.2. Protocol Master Keys

Current routing protocols directly use the protocol master key to protect the integrity of messages. One advantage for this approach is that the initial hello messages used for discovery and capability exchange can be protected using the same mechanism as other messages. Typically a sequence number is used for replay detection. Without changing the key, the existing protocols are vulnerable to a number of serious denial of service attacks from replays.

The MaRK can solve this replay problem by changing the protocol master key whenever a peer is about to exhaust its sequence number space or whenever a peer loses information about what sequence numbers it used. This could potentially involve changing the protocol master key whenever a router reboots that was part of the group using the current protocol master key. Since key changes will not disrupt active adjacencies and can be accomplished relatively quickly, this is not expected to be a huge problem. Note that after one key change, others routers can boot without causing additional key changes; a flurry of key changes would not be required if several routers reboot near each other.

Another approach would be to separate the protocol master key from the transport keys. For example the transport key used by a given peer could be a fresh key derived from the protocol master key and nonces announced by that peer. Some secure mechanism would be provisioned to enable one to confirm that the peer’s announcement of its nonce was fresh and authentic; this mechanism would almost certainly involve some form of interaction with the router wishing to guarantee freshness in order to resistant, e.g., replay attacks. There are two key advantages of this separation between transport keys and protocol master keys. The first is that the interaction between the MaRK and routing protocol can be simplified significantly. The second is that even when manually configured protocol master keys are used, replay and adequate DOS protection can be achieved.

A simple compare between the keys described in this section is provided in the following table.
<table>
<thead>
<tr>
<th>Keys</th>
<th>KMP usage vs. RP usage</th>
<th>Bootstrapping vs. Traffic</th>
<th>Group vs Pair-Wise</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Shared Keys</td>
<td>KMP usage</td>
<td>Bootstrapping</td>
<td>Pair-Wise</td>
<td>Distributed in an out-of-band way</td>
</tr>
<tr>
<td>Key Encryption Key</td>
<td>KMP usage</td>
<td>Bootstrapping</td>
<td>Group</td>
<td>For GCKS to distribute protocol master keys</td>
</tr>
<tr>
<td>Protocol Master Key</td>
<td>KMP usage or Both</td>
<td>Bootstrapping or Both</td>
<td>Group</td>
<td>Used by group members to secure routing packets or generate traffic keys</td>
</tr>
<tr>
<td>Transport Key</td>
<td>RP usage</td>
<td>Traffic</td>
<td>Group</td>
<td>Used by group members to secure routing packets</td>
</tr>
</tbody>
</table>

2.2. GCKS Election

Before a MaRK system actually starts working, the routers in the multicast group need to elect a GCKS so that they can obtain cryptographic keys to secure subsequent exchanges of routing information. MaRK specifies an election protocol that dynamically assigns the responsibility of key management to one of the group members. Note that there are already announcer-electing mechanisms provided in some routing protocols (e.g., OSPF and IS-IS). However, much involvement between a MaRK system and a routing protocol implementation will be introduced if the MaRK system reuses the announcer-electing mechanism for the election of the GCKS. The state machine of the routing protocol also has to be modified. For instance, in OSPF, after a DR has been elected, routers need to halt their OSPF executions, and carry out the initial exchange to authenticate the DR and collect the keys for subsequent communications. After this step, the routers need to re-start their OSPF state machines so as to exchange routing information. As a consequence of such cases, an individual GCKS electing solution within MaRK is preferable.

Each router has a GCKS priority. Higher priorities are more preferred GCKSes. As discussed in Section 8, the routing protocol can influence the GCKS election protocol by manipulating the priority so that it is likely that the same router will be the announcer for the routing protocol and the GCKS. Even if two different routers are elected as the announcer and GCKS, then the routing protocol and MaRK
will function correctly.

A key design goal of the election protocol is to maximize the chance that some router permitted to take on the role of GCKS will be elected to that role even when attackers are injecting messages into the election process. The election process can be attacked to cause a router other than the most preferred router to be elected.

2.3. Initial Exchange

The initial exchange is based on IKEv2’s IKE_SA_INIT and IKE_SA_AUTH exchanges. During this exchange, an initiating router attempts to authenticate to the router it believes is a GCKS for a group that the initiating router wants to join. Messages are unicast from the initiator to the responding GCKS. Unicast MaRK messages form a request/response protocol; the party sending the messages is responsible for retransmissions.

The initial exchange provides capability negotiation, specifically including supported cryptographic suites for the key management protocol. Identification of the initiator and responder is also exchanged. A symmetric key is established to protect integrity, confidentiality and authenticity of the subsequent key management messages. While routing security does not typically require confidentiality, the key management protocol does because keys are exchanged and these must be protected.

Then the identities of each party are cryptographically verified. This can be done using, e.g., a preshared key, asymmetric keys or self-signing certificates. Other mechanisms may be added as a future extension.

The authentication exchange also provides an opportunity to join a group as part of the initial exchange. In the typical case, a router can obtain the needed key material for a group in two round-trips.

2.4. Group Join Exchange

The primary purpose of the unicast MaRK messages is to get an initiator the information it needs to join a group and participate in a routing protocol. The initiator can contact a GCKS to apply to join a group that the GCKS manages. In the case a GCKS manages multiple groups concurrently, the initiator can additionally provide a group identifier to indicate which particular group it intends to join.

The responder performs several checks. First, the responder confirms that the responder is currently acting as GCKS for the group in
question. Then, the responder confirms that the initiator is permitted to join the group. If these checks pass, then the responder provides a key download payload to the initiator encrypted in the peer key management key. As discussed in Section 2.1.2, the GCKS MUST change the protocol master key if a router was part of the group under the current protocol master key and reboots. In this case, the GCKS SHOULD provide the new and old protocol master key to the initiator, setting the validity times for the old key to permit reception but not transmission. The GCKS MUST use the mechanism in the next section to flood the new key to the rest of the group.

A group association created by this exchange may last beyond the unicast MaRK association used to create it. Once membership in a group is established, resources are not required to maintain the unicast association with the GCKS.

2.5. Group Key Management

After the establishment of a group, a KEK is shared by the GCKS and all the other group members. Using the KEK, the GCKS can securely send multicast messages to the group in order to, for example, update the set of protocol master keys, revoke the KEK, or initiate new group join exchanges.

Typically, a protocol master key may be changed for the purpose of replay protection or as a result of KEK update. The KEK needs to be updated whenever a new GCKS is elected or whenever it is administratively desirable to change the keys. For example, after an employee leaves an organization it might not be wise to keep using the KEKs (and any other keys) that the employee has accessed. A KEK update is also required whenever forward security is desired: whenever the authorization of who is permitted to be in a group changes and the GCKS needs to make sure that the router is no longer participating. Most authorization changes such as removing a router from service do not require forward security in practical deployments.

3. GKCS Election

After a successful GCKS election process, a single router is selected to act as the GCKS for a group. Similar with other popular announcer electing mechanisms (e.g., VRRP, HSRP), in MaRK, only GCKSes use multicast to periodically send Advertisement messages. Such advertisements can be used as heart beat packets to indicate the aliveness of GCKSes. In addition, a state machine with six states (Initial, Validate, GCKS, GCKS2, Follower, and Member) is specified for GCKS election. When a router is initially connected to a
multicast network, its state is set as Initial. The router then sends a multicast initial advertisement. If a GCKS is working on the network, it will reply to the router with an advertisement. After receiving the advertisement from the GCKS, the router will try to register with the GCKS using the initial exchange. Typically this registration will succeed, and the state of the router is transferred to Member. After a certain period, if the router still does not receive any advertisement from a GCKS or other group members, the router then believes there is no other group member on the network and sets its state as GCKS. If during the period the router does not receive any advertisement from a GCKS but receives advertisements from other more preferred routers on the network, the router believes that the group is involved in a GCKS election process. The router then puts these routers into its candidate list. When the timer to end the Initial state expires, the router tries to authenticate the most preferred router in the candidate list and validate whether it can be a GCKS. If the validation result is positive, the router then transfers its state to Member, and the router being validated transfers its state to GCKS.

In the absence of attacks, this process functions similar to designated router election protocols in existing routing protocols. Because the election process happens before group keys are established, the initial election process is not integrity-protected. An attacker can inject fake GCKS announcements or initial announcements from fake routers that are more preferred than any router actually in the group. Such attacks can create a denial of service situation. If the election process does not converge within the expected time, or if an authentication attempt fails, then the group is probably under attack. A new state called GCKS2 is introduced. A router permitted to be the GCKS can enter the GCKS2 state after failing to validate a received announcement in the expected time. GCKS2 is used to increase the convergence speed while the system is under attack. If an initial router receives a GCKS2 announcement, the initial router can authenticate and validate the sender, and transfer its own state to Follower, similar to how it would respond to a GCKS announcement. GCKS2 routers attempt to validate each other and to use the resulting security keys to establish a router to act as GCKS. The GCKS2 state does not generate protocol master keys: until the election result in a GCKS only keying material needed for the election is produced. In the subsequent election, the router will wait for the election results from its GCKS2 router until its GCKS2 end timer expires. In this way, the authenticated entities generate a tree structure and avoid generating large amount of KEKs and protocol master keys when a adversary keeps sending fake GCKS announcements to disrupt election.

Apart from the initialization of a multicast group, the fail-over of
a GCKS can also trigger an election process. For instance, if a router does not receive the heart beat advertisement for a certain period, it will transfer its state to Initial and try to elect a new one. In a GCKS electing process, a router has to stay in the Initial state until a new GCKS is allocated. Particularly, the router first sends its initial advertisement with its priority and waits for a certain period. During the period, if a router receives an initial advertisement which consists of a lower priority, the router then sends the advertisement again with a limited rate. After period, if the router does not find any router with a higher priority, it announces itself as the GCKS. If two routers have the same priority, the one with the lowest IP source address used for messages on the link will be the GCKS. After a router transfers its state to GCKS, it will reply to the initial advertisements from other routers with GCKS advertisements, even when the initial advertisements consist of higher priorities than its priority. This approach guarantees that a GCKS will not be changed frequently after it has been elected. After receiving the GCKS advertisement of the new elected GCKS, other routers transfer their states to Member. However, if a GCKS G1 receives a GCKS advertisement from another router G2 and G2 is a more preferred GCKS, G1 follows the procedure in Section 3.2.

If a node in state member fails to perform an initial exchange with the router it believes to be GCKS, it resets its state to initial but ignores advertisements from that router. This way an attacker cannot disrupt communications indefinitely by masquerading as a GCKS.

3.1. A new GCKS is Elected

This section is a detailed description of the election process.

In the following discussion, the packets are identified by all upper case characters.

3.1.1. Parameters, Timers, and Events

Before going into detailed discussion, several parameters are introduced:

- Initial_Anno_Interval, which is the time interval between INITIAL_ANNOUNCEMENTS).

- Initial_End_Interval, which is the time interval to transfer the state of a router from Initial to GCKS/Validate if it does not receive any GCKS or GCKS2 announcement on the link).

- Validate_End_Interval, which is the time interval for a router to transfer its state from Validate to GCKS2 if it does not find any
other more preferred router).

- GCKS.Down.Interval, which is the time interval for a Member router to declare a GCKS router is down).

- GCKS2.Down.Interval, which is the time interval for a Follower router to declare a GCKS2 router is down).

- GCKS2.End.Interval, which is the time interval for a router to transfer its state from GCKS2 to GCKS if it does not find any other more preferred router).

- GCKS.Anno.Interval, which is the time interval between GCKS.ANNOUNCEMENTS).

- GCKS2.Anno.Interval, which is the time interval between GCKS2.ANNOUNCEMENTS).


During an election process, a MaRK router may have to deal with following types of events:

- X.Anno.Received: an X.ANNOUNCEMENT is received.

- Requester.Validated: have authenticated and validated against a some router who believes we should be a GCKS or GCKS2.

- GCKS.Validated: a remote entity has been authenticated and validated to be a GCKS router.

- GCKS2.Validated: a remote entity has been authenticated and validated to be a GCKS2 router.
- Referral_Validated: have authenticated and validated against a candidate who is not a GCKS router but knows one is.

- Referral2_Validated: have authenticated and validated against a candidate who knows a GCKS2 router.

- Authentication/Validation_Failed: the remote entity fails in the authentication or cannot be either a GCKS/GCKS2 or a referral.

- X_Timer_Expired: the timer of type X expired.

- KEK_Expired: we have no valid KEK.

### 3.1.2. Initial

The timers utilized in this state are Initial_Anno_Timer and Initial_End_Timer.

#### On entry:

- Send an INITIAL_ANNOUNCEMENT.

- Set the Initial_Anno_Timer with Initial_Anno_Interval.

- Set the Initial_End_Timer with Initial_End_Interval.

#### Events:

- Initial_Anno_Timer_Expired: send an INITIAL_ANNOUNCEMENT and reset the Initial_Anno_Timer.

- Initial_Anno_Received: if the sender of the announcement is more preferred, add the entity into the candidate list; if less preferred, send an INITIAL_ANNOUNCEMENT with a limited rate.

- GCKS_Anno_Received: add the sender of the announcement to the candidate list; set the Validate_End_Timer with the remaining period of Initial_End_Interval; transfer to validate.

- GCKS2_Anno_Received: add the sender of the announcement to candidate list; set the Validate_End_Timer with the remaining period of Initial_End_Interval; transfer to validate.

- Requester_Validated: If the requester is looking for a GCKS router and the local policy permits, transfer the state to GCKS2 and set GCKS2_End_Interval to time remaining on Initial_End_timer.
3.1.3. Validate

The timer utilized in this state is Validate_End_Timer.

Entering this state means that there is a router which potential could be a GCKS. The purpose of this state is to confirm that it is able to establish a security association with that router and that router’s policy permits it to be a GCKS for this group. The two normal paths through the state machine are Initial leading to GCKS for the most preferred router and Initial leading to Validate leading to Member for other routers.

On entry:

- Authenticate and validate the most preferred entry in the candidate list.
- If Validate_End_timer has more time than Validate_end_interval, set Validate_End_timer to Validate_End_interval.

Events:

- GCKS_Validated: transfer the state to Member.
- GCKS2_Validated: Transfer the state to Follower.
- Referral_Validated: perform the authentication/validation on the recommended node; move the referring from the candidate list to the black list for Blacklist_Interval.
- Referral2_Validated: perform the authentication/validation on the recommended node; move the referring node from the candidate list to the black list for Blacklist_Interval.
- Requester_Validated: If the requester is looking for a GCKS/GCKS2 router and the local policy permits, transfer the state to GCKS2.
- Validation_Failed: move the router being validated from the candidate list to black list for Blacklist_interval.
- Initial_Anno_Received: if the sender of the announcement is more preferred, add the router into the candidate list; if less preferred, send an INITIAL_ANNOUNCEMENT with a limited rate.
3.1.4. GCKS2

The timers utilized in this state include GCKS2_Anno_Timer and GCKS2_End_Timer.

When a router transfers its state from Validate to GCKS2, it is indicated that there has been some authentication/validation problem or another node is behaving in a manner inconsistent with the election state. In this case, the purpose of the GCKS2 state is to establish sufficient security keys to integrity protect the election process. In addition, it is possible for a router to enter this state during normal operations if the router being elected GCKS gets an authentication request before Initial_End_timer expires. In this case, the router will transfer its state to GCKS if no more preferred GCKS candidate is found within a limited period.

On entry:

- Send an GCSK2_ANNOUNCEMENT.
- Set the GCKS2_Anno_Timer with GCKS2_Anno_Interval.
- Set the GCKS2_End_Timer with GCKS2_End_Interval unless it was set on entry transferring from Initial.

Events:

- GCKS_Anno_Received: add to candidate list; start authentication/validation.
- GCKS2_Anno_Received: if more preferred, add to candidate list, start authentication/validation. If less preferred, send GCKS2_ANNOUNCEMENT if rate limiting is permitted.
- GCKS_Validated: Transfer to member state; flood KEK to the associated followers.
- GCKS2_Validated: Transfer the state to Follower; flood KEK to the associated followers.
o Referral_Validated: Perform authentication and validation on the recommended node; move the referring node from the candidate list to the black list for Blacklist_Interval.

o Referral2_Validated: if the recommended GCKS2 is more preferred, perform authentication and validation on the recommended node; move the referring from the candidate list to the black list for Blacklist_Interval.

o Requester_Validated: if the requester is looking for a GCKS2, distribute KEK.

o Validation_Failed: move the router being validated from the candidate list to black list for Blacklist_interval.

o GCKS2_End_Timer_Expired: transition the state to GCKS.

o GCKS2_Announce_Timer_Expired: send a GCKS2_Announcement.

3.1.5. GCKS

The timer utilized in this state is GCKS_Announce_Timer.

On entry:

o Send a GCKS_Announcement.

o Set the GCKS_Announce_Timer with GCKS_Announce_Interval.

o Generate protocol keys; if needed, generate KEK.

Events:

o GCKS_Announce_Timer_Expired: send a GCKS_Announcement.

o Initial_Announce_Received: send an GCKS_Announcement immediately if the rate limiting is permitted.

o GCKS2_Announce_Received: send an GCKS_Announcement immediately if the rate limiting is permitted.

o GCKS_Announce_Received: if the sender is more preferred, add to candidate list and start authentication/validation; Otherwise, send an GCKS_Announcement immediately if the rate limiting is permitted.

o GCKS_Validated: start network merging operations as what is illustrated in Section 3.2.
3.1.6. Member

The timer utilized in this state is GCKS_Down_Timer.

On entry:

- Set the GCKS_Down_Timer with GCKS_Down_Interval.

Events:

- GCKS_Down_Timer_Expired: Transfer the state into Initial.
- GCKS_Anno_Received: reset GCKS_Down_Timer.
- Requester_Validated: if the requester is legal, recommend the GCKS router to it.

3.1.7. Follower

The timer utilized in this state is GCKS2_Down_Timer.

On entry:

- Set the GCKS2_Down_Timer with GCKS2_Down_Interval.

Events:

- GCKS2_Down_Timer_Expired: Transfer the state into Initial.
- GCKS2_Anno_Received: reset GCKS2_Down_Timer.
- GCKS_Anno_Received: Add the announcer to the candidate list and start validation.
- Requester_Validated: if the requester is legal, recommend the GCKS2 router to it.
- GCKS_Validated: Transfer the state to member.

The following diagram illustrates the rules of transiting the states introduced this section.
3.2. Merging Partitioned Networks

Whenever a GCKS finds that a more preferred router is also acting as a GCKS for the same group, then the group is partitioned. Typically if there is already an active GCKS for a group, even if a more preferred router joins the group, the GCKS will not change. Two situations can result in multiple GCKSes active for a group. The first is that members of the group do not share common authentication credentials. The second is that the group was previously partitioned so that some nodes could not see election messages from other nodes. After the problem resulting in the partition is fixed, then both active GCKSes will see each others election announcements. The group needs to merge.

The less preferred GCKS performs a unicast mark_merge_sa unicast key management message to the more preferred GCKS. In this message the less preferred GCKS includes its key download payload, so the more preferred GCKS learns the protocol master keys of the less preferred GCKS.

The more preferred GCKS generates a new key download payload including a KEK and the union of all the protocol master keys. The GCKS SHOULD mark the existing protocol master keys as expiring for usage in transmitted packets in a relatively short time. The GCKS SHOULD introduce a new protocol master key. This key download payload is returned to the less preferred GCKS and is sent out in the current KEK using a group key management message.
The less preferred GCKS sends the received key download payload encrypted in its existing KEK and retransmit the message for several times according to its local policy. After all retransmissions of this payload the less preferred GCKS sets its state to member.

As a result of this procedure, members learn the protocol master keys of both GCKSes and converge on a single KEK and GCKS. Changing the protocol master keys during a merge is important for protocols that use the protocol master key as a transport key. The new GCKS does not know which routers have joined the group with the other GCKS. Therefore, it could not correctly detect one of these routers rebooting and change the protocol master key at that point. If the key is changed as part of the merge, replays are handled.

3.3. Operations on Receiving a Packet

When a router attempts to join an election process, it may have a valid KEK. For instance, when a GCKS cannot work properly, the routers on the link need to transfer their state to Initial and raise an election to find a new valid GCKS. If there is still a valid KEK shared by the router, they can use the KEK to secure the packets transmitted during the election until a new KEK is distributed by the new GCKS. A router holding the valid KEK is regarded to be more preferred than a router which doesn’t have the key. By using the KEK, it is able to prevent an attacker from disturbing the election process by broadcasting fake announcements. Therefore, after an initial router does not find any more preferred router holding the valid key, it then can transfer its state to GCKS directly.

Therefore, the operations on receiving a packet are as follows:

- Check the blacklist. If the sender of the packet is on the blacklist, discard the packet.
- If the state is GCKS, accept the packet and generate an event. GCKS announcements need to be excepted in GCKS state for merges to work.
- If there is a KEK that is not expired, check the packet integrity against any matching KEK.
- If no KEK matches or if the integrity fails to validate, discard the packet.
- If there is no KEK at all or the KEK integrity check passed, process the packet and generate an event.

It is notable this approach limits the scope of the election within
the routers managed by the failed GCKS. If there are routers newly accessing the link during the election, no router with a KEK will process their packets. However these routers can process packets from routers with the KEK. In many cases one of the routers with a KEK will be elected GCKS and the other routers can authenticate and join. In the worst case, two independent GCKSes will be elected and then merge.

4. Key Download Payload

What all is actually in the message you get at the end of phase 1 exchange (the RP_AUTH Exchange) and that is sent out periodically during group key management.

For the KEK, this needs to include the key itself, the algorithm (presumably drawn from the IKEv2 symmetric algorithms), key ID, group ID transmit start time, receive start time, and expire time.

The protocol master keys include the key, an algorithm ID, the key ID and the lifetimes.

5. Initial Exchanges Details

Similar with [I-D.tran-karp-mrmp], in MRKMP, when two routers needs to authenticate each other, they need to perform the initial exchanges defined in [I-D.mahesh-karp-rkmp]. For example, when a router intends to join a group, it needs to firstly perform a RP Initial (RP_INIT) Exchange with the GCKS of the group. RP_INIT is identical to the IKE_SA_INIT exchange defined in Internet Key Exchange Protocol Version 2 [RFC5996], after which the router and the GCKS 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].

<table>
<thead>
<tr>
<th>Router</th>
<th>GCKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SA1, KE1, N1</td>
<td>HDR, SA1, KE1, NR, [CERTREQ],</td>
</tr>
<tr>
<td>---&gt;</td>
<td>&lt;---</td>
</tr>
</tbody>
</table>

The router and the GCKS then needs to perform an RP_AUTH exchange defined in [I-D.mahesh-karp-rkmp]. At the successful conclusion of the exchange, the router is adopted as a group member and obtains keying material (e.g., the KEK and protocol master key) to securely...
communicate with other group members.

```
Router                                GCKS
--------------------                  ------------------
HDR, SK {IDi, [CERT,] [CERTREQ,]
[IDr,] AUTH, SArpi}       -->
                         <- HDR, SK {IDr, [CERT,] AUTH,
                                        SArpr}
RP_AUTH
```

6. Group Management Unicast Exchanges

6.1. Group Join Exchange

If a router receives a group join exchange for a group for which it is not the GCKS, it MUST return a notification. If it knows the GCKS for the group then it returns MaRK_WRONG_GCKS including the address of the GCKS or GCKS2 in the notification payload along with an indication of whether the router is a GCKS or GCKS2. The initiator tries the group join exchange (probably with a new initial exchange) with the indicated router. If the responder does not know the GCKS for the group, either because it is not a member of the group or because its GCKS election state is initial, it returns the MaRK_GCKS_UNKNOWN notification.

7. Group Key Management Operation

7.1. General operation

Periodically the GCKS will send out an update message encrypted in the current KEK including the current group key download payload and parameters. If a new KEK is about to be valid for receiving messages, this is included. Any protocol master keys that are valid for sending or receiving SHOULD be included.

If a previous KEK is still valid for sending, then an update message is sent encrypted in the old KEK. This message MUST include the new KEK. This message SHOULD include the protocol master keys.

7.2. Out of Sequence Space

A member of a group can also use the unicast exchange to request a GCKS to change a protocol master key, on the occasions, for example, where the member is going to exhaust its sequence space of the associated routing protocol. For protocols where the protocol master key is the same as the transport key, it is critical that no two
messages be sent by the same router with the same sequence number and protocol master key. The sequence number space is finite. So if a router is running low on available sequence space it needs to request a new protocol master key be generated.

7.3. Changing the Active GCKS

When a GCKS finds a more preferred router announcing itself as a GCKS, it will forward its privilege to another one in the following conditions. The operations are introduced in Section 3.2.

When a GCKS cannot work properly, it will just stop sending the GCKS_ANNOUNCEMENT. Then after a certain time period, a new GCKS election process will be raised.

7.4. Reboot Cases

After a reboot, a router in a group will lost the state information about the group (e.g., protocol master keys, traffic keys, the sequence numbers used by GCKS). Therefore, the router needs to find and authenticate the GCKS, and apply to join the group. If the GCKS finds that the router is already a group member, the GCKS will update the transport keys (and the protocol master keys if necessary) used in the group first in order to avoid inter-session replay attacks.

8. Interface to Routing Protocol

This section describes signaling between MaRK and the routing protocol. The primary communication between these protocols is that MaRK populates rows in the key table making protocol master keys available to the routing protocol. However additional signaling is also required from the routing protocol to MaRK. This section discusses that signaling. All required communication from MaRK to the routing protocol can be accomplished by manipulating the key table. However an implementation MAY wish to signal MaRK failures to the routing protocol in order to provide consistent management feedback.

8.1. Joining a Group

When a routing protocol instance wishes to begin communicating on a multicast group, it signals a group join event to MaRK. This event includes the identity of the group as well as this router’s priority for being a GCKS for the group. When MaRK receives this event, it starts MaRK for this group and attempts to find a GCKS.
8.2. Priority Adjustment

It is desirable that the GCKS function track the functions within a routing protocol. For example for protocols such as OSPF that designate a router on a link to manage adjacencies for that link, it would be desirable for the GCKS role to be assigned to that router. The routing protocol provides a priority input to the GCKS election process. Initially the routing protocol should map any priority mechanism within the routing protocol to the GCKS election procedure so that routers favored as announcer for a link will also be favored as a GCKS.

However, the routing protocol SHOULD also dynamically manipulate the GCKS election priority based on what happens within the routing protocol. The router actually elected as the announcer SHOULD have a GCKS election priority higher than any other group member. Typically, by the time the routing protocol is able to elect an announcer, a GCKS will already be chosen. However, if a GCKS election is triggered when the routing protocol is already operational, then the election can choose the routing protocol’s announcer.

8.3. Leaving a Group

If a routing protocol terminates on an interface, MaRK implementation on the router needs to be notified that group is no longer joined. MaRK MUST stop participating in the GCKS election process, stop monitoring for key management messages and if the current router is a GCKS, stop acting in that role.

8.4. Out of Sequence Space

If a routing protocol is running out its sequence space, the MaRK implementation on the router needs to be notified. The MaRK implementation then needs to contact the GCKS to request the update of the transport keys (and the protocol master keys if necessary).

9. Security Considerations

This protocol is intended to protect against attackers who are not properly authorized mounting a integrity or availability attack on the system. All parties who are authorized to be part of a given group are equivalent; group members impersonating each other, impacting availability or integrity are all out of scope for this threat model. Protecting confidentiality of key material against parties not authorized for membership in a given group is in scope as it would directly lead to an attack on integrity or availability.
Protecting confidentiality of group policy or routing data is not required. Attackers are assumed to be able to insert and observe packets. Even if attackers can modify and suppress packets, integrity should not be impacted. Minimizing the availability impact against attackers who can modify and suppress packets is strongly desirable, although there are limits to this defense. It is important that a member of one group not be able to impact another group.

Significant complexity results from the election protocol. In order to support arbitrary authentication mechanisms including preshared keys, the election protocol itself is not signed. At least before group keys are established, the election protocol is not integrity protected. Later authentication can establish integrity, but managing availability attacks on the election protocol requires significant analysis.

An attacker who can suppress packets sent to the group can create a denial of service condition. One attack is to suppress GCKS election packets and cause two routers to believe they are both the GCKS for the group. If the least preferred router never hears the GCKS advertisement from the more preferred router, then the group will remain partitioned. Such an attacker is likely to be able to mount more direct denial of service, for example suppressing the actual routing protocol packets.

The election protocol has been designed to try and resist denial of service conditions. However, the election protocol maintains state in the form of a candidate list and black list. An attacker can consume state by generating fake election announcements. An implementation can discard state if it has insufficient resources. However, if legitimate routers are discarded from the candidate list, the protocol may take longer to converge or may not converge. If entries are removed from the black list, then more resources may be spent on attackers. So the solution has some residual denial of service possibilities. The election protocol requires significant analysis to confirm it meets its design goals.

The security of the election protocol depends on the denial of service resistance of the authentication protocol. It is important that an attacker not be able to cause an authentication to fail by injecting a packet. So, rather than failing an authentication if a bad packet is received, an implementation needs to wait and see if a good packet appears in some timeout.

The security of the system as a whole depends on the pair-wise security between the router currently in the GCKS role and the other routers in the group. Since any router can potentially act as GCKS,
the pair-wise security between all members of the group is critical
to the security of the system. In practical deployments, information
used by the router acting as GCKS to authorize a member joining the
group will be configured by some management application. In these
deployments, the security of the system depends on the management
application correctly maintaining this information on all routers
potentially in the group.

10. Acknowledgements

The funding for Sam Hartman’s work on this document is provided by
Huawei.

XXX add the list of people in the lunch time group unless they are
willing to be listed as authors.

11. References

11.1. Normative References

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

[RFC2409]  Harkins, D. and D. Carrel, "The Internet Key Exchange
(IKE)", RFC 2409, November 1998.


11.2. Informative References

Hartman, "Key Management for Pairwise Routing Protocol",
draft-mahesh-karp-rkmp-01 (work in progress), March 2012.

Router Key Management", draft-tran-karp-mrmp-01 (work in
progress), March 2012.

[I-D.yeung-g-ikev2]  Rowles, S., Yeung, A., Tran, P., and Y. Nir, "Group Key
Management using IKEv2", draft-yeung-g-ikev2-04 (work in progress), March 2012.


Authors’ Addresses

Sam Hartman
Painless Security

Email: hartmans-ietf@mit.edu

Dacheng Zhang
Huawei Technologies co. ltd
Huawei Building No.3 Xinxi Rd., Shang-Di Information Industrial Base Hai-Dian District, Beijing
China

Email: zhangdacheng@huawei.com

Gregory Lebovitz
Juniper Networks, Inc.
1194 North Mathilda Ave.
Sunnyvale, California  94089-1206
USA

Email: gregory.ietf@gmail.com
Database of Long-Lived Symmetric Cryptographic Keys
<draft-ietf-karp-crypto-key-table-02.txt>

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

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.

Abstract

This document specifies the information contained in a database of long-lived cryptographic keys used by many different security
protocols. The database design supports both manual and automated key management. In many instances, the security protocols do not directly use the long-lived key, but rather a key derivation function is used to derive a short-lived key from a long-lived key.

1. Introduction

This document specifies the information that needs to be included in a database of long-lived cryptographic keys. This conceptual database is designed to support both manual key management and automated key management. The intent is to allow many different implementation approaches to the specified cryptographic key database.

Security protocols such as TCP-AO [RFC5925] are expected to use an application program interface (API) to select a long-lived key from the database. In many instances, the long-lived keys are not used directly in security protocols, but rather a key derivation function is used to derive short-lived key from the long-lived keys in the database. In other instances, security protocols will directly use the long-lived key from the database. The database design supports both use cases.

2. Conceptual Database Structure

The database is characterized as a table, where each row represents a single long-lived symmetric cryptographic key. Each key should only have one row; however, in the (hopefully) very rare cases where the same key is used for more than one purpose, multiple rows will contain the same key value. The columns in the table represent the key value and attributes of the key.

To accommodate manual key management, then formatting of the fields has been purposefully chosen to allow updates with a plain text editor.

The table has the following columns:

- **LocalKeyId**
  - LocalKeyId is a 16-bit integer in hexadecimal. The LocalKeyId can be used by a peer to identify this entry in the database. For pairwise keys, the most significant bit in LocalKeyId is set to zero, and the integer value must be unique among all the pairwise keys in the database. For group keys, the most significant bit in LocalKeyId is set to one, but collisions among group key identifiers must be accommodated.
PeerKeyID
For pairwise keys, the PeerKeyID field is a 16 bit integer in hexadecimal provided by the peer. If the peer has not yet provided this value, the PeerKeyID is set to "unknown". For group keying, the PeerKeyID field is set to "group", which easily accommodates group keys generated by a third party. If the protocol associated with this key uses a keyname instead of a numeric identifier, the PeerKeyID field is set to "null". (Note that some protocols include keynames and numeric identifiers.)

KeyName
The KeyName field is a variable length text field that identifies the key material. If the value has not yet been established, the KeyName field is set to the special value "unknown". If the protocol associated with the key does not use keynames, the KeyName field is set to "null".

Peers
The Peers field identifies the peer system or set of systems that have this key configured in their own database of long-lived keys. For pairwise keys, the database on the peer system LocalKeyID field will contain the value specified in the PeerKeyID field in the local database. For group keying, the Peers field names the group, not the individual systems that comprise the group.

Interfaces
The Interfaces field identifies the set of physical and/or virtual interfaces for which it is appropriate to use this key. When the long-lived value in the Key field is intended for use on any interface, the Interfaces field is set to "all".

Protocol
The Protocol field identifies a single security protocol where this key may be used to provide cryptographic protection. This protocol establishes a registry for this field; the registry also specifies the contents of the following field, ProtocolSpecificInfo, for each registered protocol.

ProtocolSpecificInfo
The ProtocolSpecificInfo field contains a variable length binary object with any protocol specific values. From the perspective of the database, this is an opaque object. The type and contents of the subfields are specified as part of the IANA registration for the Protocol field value.
KDF
The KDF field indicates which key derivation function is used to generate short-lived keys from the long-lived value in the Key field. When the long-lived value in the Key field is intended for direct use, the KDF field is set to "none". This document establishes an IANA registry for the values in the KDF field to simplify references in future specifications.

KDFInputs
The KDFInputs field is used when supplementary public or private data is supplied to the KDF. For protocols that do not require additional information for the KDF, the KDFInputs field is set to "none". The Protocol field will determine the format of this field if it is not "none".

AlgID
The AlgID field indicates which cryptographic algorithm to be used with the security protocol for the specified peer. The algorithm may be an encryption algorithm and mode (such as AES-128-CBC), an authentication algorithm (such as HMAC-SHA1-96 or AES-128-CMAC), or any other symmetric cryptographic algorithm needed by a security protocol. If the KDF field contains "none", then the long-lived key is used directly with this algorithm, otherwise the derived short-lived key is used with this algorithm. When the long-lived key is used to generate a set of short-lived keys for use with the security protocol, the AlgID field identifies a ciphersuite rather than a single cryptographic algorithm. This document establishes an IANA registry for the values in the AlgID field to simplify references in future specifications.

Key
The Key is a hexadecimal string representing a long-lived symmetric cryptographic key. The size of the Key depends on the KDF and the AlgID. For example, a KDF=none and AlgID=AES128 requires a 128-bit key, which is represented by 32 hexadecimal digits.

Direction
The Direction field indicates whether this key may be used for inbound traffic, outbound traffic, or both. The supported values are "in", "out", and "both", respectively. The Protocol field will determine which of these values are valid.

SendNotBefore
The NotBefore field specifies the earliest date and time in Universal Coordinated Time (UTC) at which this key should be considered for use when sending traffic. The format is
YYYYMMDDHHSSZ, where four digits specify the year, two digits specify the month, two digits specify the day, two digits specify the hour, and two digits specify the minute. The "Z" is included as a clear indication that the time is in UTC.

SendNotAfter
The NotAfter field specifies the latest date and time at which this key should be considered for use when sending traffic. The format is the same as the NotBefore field.

RcvNotBefore
The NotBefore field specifies the earliest date and time in Universal Coordinated Time (UTC) at which this key should be considered for use when processing received traffic. The format is YYYYMMDDHHSSZ, where four digits specify the year, two digits specify the month, two digits specify the day, two digits specify the hour, and two digits specify the minute. The "Z" is included as a clear indication that the time is in UTC.

RcvNotAfter
The NotAfter field specifies the latest date and time at which this key should be considered for use when processing received traffic. The format is the same as the NotBefore field.

Note that some security protocols use a KeyID value of zero for special purposes, so care is needed if this KeyID value is included in the table.

3. Key Selection and Rollover

When a system desires to protect a unicast protocol data unit for a remote system H using security protocol P via interface I, the local system selects a long-lived key at time T from the database, any key that satisfies the following conditions may be used:

(1) the Peer field includes H;
(2) the PeerKeyID field is not "unknown";
(3) the Protocol field matches P;
(4) the Interfaces field includes I;
(5) the Direction field is either "out" or "both"; and
(6) NotBefore <= T <= NotAfter.
The value in the PeerKeyID field is used to identify the selected key to the remote system H.

Group key selection is different than pairwise key selection. When a system desires to protect a multicast protocol data unit for a group of systems G using security protocol P via interface I, the local system selects a long-lived key at time T from the database, any key that satisfies the following conditions may be used:

1. the Peer field includes the multicast group G;
2. the PeerKeyID field is "group";
3. the Protocol field matches P;
4. the Interfaces field includes I;
5. the Direction field is either "out" or "both"; and

The value in the LocalKeyID field is used to identify the selected key since all of the systems in the group G use the same identifier.

During algorithm transition, multiple entries may exist associated with different cryptographic algorithms or ciphersuites. Systems should support selection of keys based on algorithm preference.

In addition, multiple entries with overlapping use periods are expected to be employed to provide orderly key rollover. In these cases, the expectation is that systems will transition to the newest key available. To meet this requirement, this specification recommends supplementing the key selection algorithm with the following differentiation: select the long-lived key specifying the most recent time in the NotBefore field.

When a system participates in a security protocol, a sending peer system H has selected a long-lived key and the LocalKeyID is included in the protocol control information. When retrieving the long-lived key (for direct use or for key derivation), the local system should confirm the following conditions are satisfied before use:

1. the Peer field includes H;
2. the Protocol field matches P;
3. the Interface field includes I;
(4) the Direction field is either "in" or "both"; and
(5) NotBefore <= T <= NotAfter.

Note that the key usage is loosely bound by the times specified in
the NotBefore and NotAfter fields. New security associations should
not be established except within the period of use specified by these
fields, while allowing some grace time for clock skew. However, if a
security association has already been established based on a
particular long-lived key, exceeding the lifetime does not have any
direct impact. Implementations of protocols that involve long-lived
security association should be designed to periodically interrogate
the database and rollover to new keys without tearing down the
security association.

For group keying, the local system should confirm the following
conditions are satisfied before use:
(1) the Peer field includes the multicast group G;
(2) the PeerKeyID field is "group";
(3) the Protocol field matches P;
(4) the Interface field includes I;
(5) the Direction field is either "in" or "both"; and
(6) NotBefore <= T <= NotAfter.

As long as a key remains in the database, the key may be used for
received traffic. Any key that is unacceptable for received traffic
needs to be removed from the database.

4. Operational Considerations

If usage periods for long-lived keys do not overlap and system clocks
are inconsistent, it is possible to construct scenarios where systems
cannot agree upon a long-lived key. When installing a series of keys
to be used one after the other (sometimes called a key chain),
operators should configure the NotAfter field of the preceding key to
be several days after the NotBefore field of the subsequent key to
ensure that clock skew is not a concern.

For group keys, the most significant bit in LocalKeyID must be set to
one. Collisions among group key identifiers can be avoided by
subdividing the remaining 15 bits of the LocalKeyID field into an
identifier of the group key generator and an identifier assigned by
that generator.

5. Security Considerations

Management of encryption and authentication keys has been a significant operational problem, both in terms of key synchronization and key selection. For example, current guidance [RFC3562] warns against sharing TCP MD5 keying material between systems, and recommends changing keys according to a schedule. The same general operational issues are relevant for the management of other cryptographic keys.

It is recognized in [RFC4107] that automated key management is not viable in some situations. The conceptual database specified in this document is intended to accommodate both manual key management and automated key management. A future specification to automatically populate rows in the database is envisioned.

Designers should recognize the warning provided in [RFC4107]:

Automated key management and manual key management provide very different features. In particular, the protocol associated with an automated key management technique will confirm the liveness of the peer, protect against replay, authenticate the source of the short-term session key, associate protocol state information with the short-term session key, and ensure that a fresh short-term session key is generated. Further, an automated key management protocol can improve interoperability by including negotiation mechanisms for cryptographic algorithms. These valuable features are impossible or extremely cumbersome to accomplish with manual key management.

6. IANA Considerations

This specification defines three registries.

6.1. KeyTable Protocols

This document requests establishment of a registry called "KeyTable Protocols". The following subsection describes the registry; the second subsection provides initial values for IEEE 802.1X.

6.1.1. KeyTable Protocols Registry Definition

All assignments to the KeyTable Protocols registry are made on a Specification Required basis per Section 4.1 of [RFC5226].
Each registration entry must contain the three fields:

- protocol name (unique within the registry);
- specification; and
- protocol specific values.

6.1.2. KeyTable Protocols Registry Initial Values

protocol name: IEEE 802.1X

specification: IEEE Std 802.1X-2010, "IEEE Standard for Local and Metropolitan Area Networks -- Port-Based Network Access Control".

protocol specific values: there are two:

- A Key Management Domain (KMD).
  A string of up to 253 UTF-8 characters that names the transmitting authenticator’s key management domain.

- A Network Identifier (NID).
  A string of up to 100 UTF-8 characters that identifies a network service. The NID can also be null, indicating the key is associated with a default service.

6.2. KeyTable KDFs

This document requests establishment of a registry called "KeyTable KDFs". The remainder of this section describes the registry.

All assignments to the KeyTable KDFs registry are made on a First Come First Served basis per Section 4.1 of RFC 5226.

6.3. KeyTable AlgIDs

This document requests establishment of a registry called "KeyTable AlgIDs". The remainder of this section describes the registry.

All assignments to the KeyTable KDFs registry are made on a First Come First Served basis per Section 4.1 of RFC 5226.

7. Acknowledgments

This document reflects many discussions with many different people over many years. In particular, the authors thank Jari Arkko, Ran Atkinson, Ron Bonica, Ross Callon, Lars Eggert, Pasi Eronen, Adrian Farrel, Sam Hartman, Gregory Lebovitz, Sandy Murphy, Eric Rescorla, Mike Shand, Dave Ward, and Brian Weis for their insights.
8. Informational References


Authors’ Addresses

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA
EMail: housley@vigilsec.com

Tim Polk
National Institute of Standards and Technology
100 Bureau Drive, Mail Stop 8930
Gaithersburg, MD 20899-8930
USA
EMail: tim.polk@nist.gov
Abstract

Different routing protocols exist and each employs its own mechanism for securing the protocol packets on the wire. While most already have some method for accomplishing cryptographic message authentication, in many cases the existing methods are dated, vulnerable to attack, and employ cryptographic algorithms that have been deprecated. The "Keying and Authentication for Routing Protocols" (KARP) effort aims to overhaul and improve these mechanisms.

This document does not contain protocol specifications. Instead, it defines the areas where protocol specification work is needed and a set of requirements for KARP design teams to follow. RFC 6518, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines" is a companion to this document; KARP design teams will use them together to review and overhaul routing protocols. These two documents reflect the input of both the IETF’s Security Area and Routing Area in order to form a mutually agreeable work plan.

This document has three main parts. The first part provides an overview of the KARP effort. The second part lists the threats from RFC 4593, Generic Threats To Routing Protocols, that are in scope for attacks against routing protocols’ transport systems, including any mechanisms built into the routing protocols themselves, which accomplish packet authentication. The third part enumerates the requirements that routing protocol specifications must meet when addressing those threats for RFC 6518’s "Work Phase 1", the update to a routing protocol’s existing transport security.

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
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 September 10, 2012.

Copyright Notice

Copyright (c) 2012 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 ................................................. 4  
   1.1. Terminology ........................................ 5  
   1.2. Requirements Language ............................... 8  

2. KARP Effort Overview ....................................... 9  
   2.1. KARP Scope .......................................... 9  
   2.2. Incremental Approach ................................. 10  
   2.3. Goals ................................................ 11  
   2.4. Non-Goals ........................................... 13  
   2.5. Audience ............................................. 14  

3. Threats ..................................................... 15  
   3.1. Threat Sources ....................................... 15  
      3.1.1. OUTSIDERS ...................................... 15  
      3.1.2. Stolen Keys ..................................... 16  
      3.1.2.1. Terminated Employee .......................... 17  
   3.2. Threat Actions In Scope ............................... 18  
   3.3. Threat Actions Out of Scope ........................... 19  

4. Requirements for KARP Work Phase 1, the Update to a  
   Routing Protocol’s Existing Transport Security .......... 21  

5. Security Considerations .................................... 27  

6. IANA Considerations ........................................ 28  

7. Acknowledgements .......................................... 29  

8. References ................................................ 30  
   8.1. Normative References ................................ 30  
   8.2. Informative References .............................. 30  

Authors’ Addresses ........................................... 32
1. Introduction

In March 2006 the Internet Architecture Board (IAB) held a workshop on the topic of "Unwanted Internet Traffic". The report from that workshop is documented in [RFC4948]. Section 8.1 of that document states "A simple risk analysis would suggest that an ideal attack target of minimal cost but maximal disruption is the core routing infrastructure." Section 8.2 calls for "[t]ightening the security of the core routing infrastructure." Four main steps were identified for that tightening:

- Create secure mechanisms and practices for operating routers.
- Clean up the Internet Routing Registry repository (IRR), and securing both the database and the access, so that it can be used for routing verification.
- Create specifications for cryptographic validation of routing message content.
- Secure the routing protocols' packets on the wire

The first bullet is being addressed in the OPSEC working group. The second bullet should be addressed through liaisons with those running the IRR's globally. The third bullet is being addressed in the SIDR working group.

This document addresses the last item in the list above, securing the transmission of routing protocol packets on the wire, or rather securing the routing protocols' transport systems, including any mechanisms built into the routing protocols themselves which accomplish packet authentication. This effort is referred to as Keying and Authentication for Routing Protocols, or "KARP". KARP is concerned with issues and techniques for protecting the messages and their contents between directly communicating peers. This may overlap with, but is strongly distinct from, protection designed to ensure that routing information is properly authorized relative to sources of information. Such assurances are provided by other mechanisms and are outside the scope of this document and work that relies on it.

This document is one of two that together form the guidance and instructions for KARP design teams working to overhaul routing protocol transport security. The other document is the KARP Design Guide [RFC6518].

This document does not contain protocol specifications. Instead, its goal is to define the areas where protocol specification work is
needed and to provide a set of requirements for KARP design teams to follow as they tackle [RFC6518], Section 4.1’s "Work Phase 1", the update to a routing protocol’s existing transport security.

This document has three main parts. The first part, found in Section 2, provides an overview of the KARP effort. Section 3 lists the threats from [RFC4593], Generic Threats To Routing Protocols, that are in scope for routing protocols’ transport systems’ per packet authentication. Therefore, this document does not contain a complete threat model; it simply points to the parts of the governing threat model that KARP design teams must address, and explicitly states which parts are out of scope for KARP design teams. Section 4 enumerates the requirements that routing protocol specifications must meet when addressing those threats related to KARP’s "Work Phase 1", the update to a routing protocol’s existing transport security. ("Work Phase 2", a framework and usage of a KMP, will be addressed in a future requirements document).

This document uses the terminology "on the wire" to refer to the information used by routing protocols’ transport systems. This term is widely used in IETF RFCs, but is used in several different ways. In this document, it is used to refer both to information exchanged between routing protocol instances, and to underlying protocols that may also need to be protected in specific circumstances. Individual protocol analysis documents will need to be more specific in their usage."

1.1. Terminology

Within the scope of this document, the following words, when beginning with a capital letter, or spelled in all capitals, hold the meanings described to the right of each term. If the same word is used uncapsulated, then it is intended to have its common English definition.

Identifier

The type and value used by a peer of an authenticated message exchange to signify who it is to another peer. The Identifier is used by the receiver as an index into a table containing further information about the peer that is required to continue processing the message, for example a Security Association (SA) or keys.

Identity Authentication
Once the identity is decided, then there must be a cryptographic proof of that identity, that the peer really is who it asserts to be. Proof of identity can be arranged among peers in a few ways, for example symmetric and asymmetric pre-shared keys, or an asymmetric key contained in a certificate. Certificates can be used in ways that requires no additional supporting systems external to the routers themselves. An example of this would be using self signed certificates and a flat file list of "approved thumbprints". The use of these different identity authentication mechanisms vary in ease of deployment, ease of ongoing management, startup effort, ongoing effort and management, security strength, and consequences from loss of secrets from one part of the system to the rest of the system. For example, they differ in resistance to a security breach, and the effort required to remediate the whole system in the event of such a breach. The point here is that there are options, many of which are quite simple to employ and deploy.

KDF (Key derivation function)

A KDF is a function in which an input key and other input data is used to generate (or derive) keying material that can be employed by cryptographic algorithms. The key that is input to a KDF is called a key derivation key. KDFs can be used to generate one or more keys from either (i) a truly random or pseudorandom seed value or (ii) result of the Diffie-Hellman exchange or (iii) a non-uniform random source or (iv) a pre-shared key which may or may not be memorable by a human.

KMP (Key Management Protocol)

A protocol to establish a shared symmetric key between a pair (or a group) of users. It determines how secret keys are generated and made available to both the parties. If session or traffic keys are being used, KMP is responsible for generating them and determining when they should be renewed.

A KMP is helpful because it negotiates unique, random keys without administrator involvement. It also negotiates, as mentioned earlier, several of the SA parameters required for the secure connection, including key life times. It keeps track of those lifetimes, and negotiates new keys and parameters before they expire, again, without administrator interaction. Additionally, in the event of a security breach, changing KMP authentication credentials will immediately cause a rekey to occur for the Traffic Keys, and new Traffic Keys will be installed and used in the current connection.
KMP Function

Any actual KMP used in the general KARP solution framework

Peer Key

Keys that are used among peers as a basis for identifying one another. These keys may or may not be connection-specific, depending on how they were established, and what forms of identity and identity authentication mechanism used in the system. A peer key generally would be provided by a KMP that would later be used to derive fresh traffic keys.

PRF

In cryptography, a pseudorandom function, abbreviated PRF, is a collection of efficiently-computable functions which emulate a random oracle in the following way: No efficient algorithm can distinguish (with significant advantage) between a function chosen randomly from the PRF family and a random oracle (a function whose outputs are determined at random). Informally, a PRF takes a secret key and a set of input values and produces random-seeming output values for each input value.

PSK (Pre-Shared Key)

A key used to communicate with one or more peers in a secure configuration. Always distributed out-of-band prior to a first connection.

Routing Protocol

When used with capital "R" and "P" in this document the term refers the Routing Protocol for which work is being done to provide or enhance its peer authentication mechanisms.

SA (Security Association)

A relationship established between two or more entities to enable them to protect data they exchange. Examples of items that may exist in an SA include: Identifier, PSK, Traffic Key, cryptographic algorithms, key lifetimes.

Traffic Key
The key (or one of a set of keys) used for protecting the routing protocol traffic. Since the traffic keys used in a particular connection are not a fixed part of a device configuration no data exists anywhere else in the operator’s systems which can be stolen, e.g. in the case of a terminated or turned employee. If a server or other data store is stolen or compromised, the thieves gain no access to current traffic keys. They may gain access to key derivation material, like a PSK, but not current traffic keys in use.

1.2. 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 RFC2119 [RFC2119].

When used in lower case, these words convey their typical use in common language, and are not to be interpreted as described in RFC2119 [RFC2119].
2. KARP Effort Overview

2.1. KARP Scope

Three basic services may be employed in order to secure any piece of data as it is transmitted over the wire: confidentiality, authenticity, or integrity. The focus for the KARP working group will be message authentication and message integrity only. This work explicitly excludes, at this point in time, privacy services. Non-repudiation is also excluded as a goal at this time. Since the objective of most routing protocols is to broadly advertise the routing topology, routing protocol packets are commonly sent in the clear; confidentiality is not normally required for routing protocols. However, ensuring that routing peers are authentically identified, and that no rogue peers or unauthenticated packets can compromise the stability of the routing environment is critical, and thus our focus. Confidentiality and non-repudiation may be addressed in future work.

OSPF [RFC5709], IS-IS [RFC5310], LDP [RFC5036], and RIP [RFC2453] [RFC4822] already have existing mechanisms for cryptographically authenticating and integrity checking the messages on the wire. Products with these mechanisms have been produced, code has been written, and both have been optimized for these existing security mechanisms. Rather than turn away from these mechanisms, this document aims to enhance them, updating them to modern and secure levels.

Therefore, the scope of KARP’s roadmap of work includes:

- Making use of existing routing protocol transport security mechanisms, where they exist, and enhancing or updating them as necessary for modern cryptographic best practices. [RFC6518], Section 4.1 labels this KARP’s "Work Phase 1."

- Developing a framework for using automatic key management in order to ease deployment, lower cost of operation, and allow for rapid responses to security breaches. [RFC6518], Section 4.1 labels this KARP’s "Work Phase 2."

- Specifying an automated key management protocol that may be combined with the bits-on-the-wire mechanisms. [RFC6518], Section 4.1 labels this KARP’s "Work Phase 2."

Neither this document nor [RFC6518] contain protocol specifications. Instead, they define the areas where protocol specification work is needed and set a direction, a set of requirements, and priorities for addressing that specification work.
There are a set of threats to routing protocols that are considered in-scope for KARP, and a set considered out-of-scope. These are described in detail in the Threats (Section 3) section below.

2.2. Incremental Approach

The work also serves as an agreement between the Routing Area and the Security Area about the priorities and work plan for incrementally delivering the above work. The principle of "crawl, walk, run" will be employed. Thus routing protocol authentication mechanisms may not go immediately from their current state to a state reflecting the best possible, most modern security practices. This point is important as there will be times when the best-security-possible will give way to vastly-improved-over-current-security-but-admittedly-not-yet-best-security-possible, in order that incremental progress toward a more secure Internet may be achieved. As such, this document will call out places where agreement has been reached on such trade offs.

Incremental steps will need to be taken for a few very practical reasons. First, there are a considerable number of deployed routing devices in operating networks that will not be able to run the most modern cryptographic mechanisms without significant and unacceptable performance penalties. The roadmap for any one routing protocol MUST allow for incremental improvements on existing operational devices. Second, current routing protocol performance on deployed devices has been achieved over the last 20 years through extensive tuning of software and hardware elements, and is a constant focus for improvement by vendors and operators alike. The introduction of new security mechanisms affects this performance balance. The performance impact of any incremental step of security improvement will need to be weighed by the community, and introduced in such a way that allows the vendor and operator community a path to adoption that upholds reasonable performance metrics. Therefore, certain specification elements may be introduced carrying the "SHOULD" guidance, with the intention that the same mechanism will carry a "MUST" in a future release of the specification.

This approach gives the vendors and implementors the guidance they need to tune their software and hardware appropriately over time. Last, some security mechanisms require the build out of other operational support systems, and this will take time. An example where these three reasons are at play in an incremental improvement roadmap is seen in the improvement of BGP's [RFC4271] security via the TCP Authentication Option (TCP-AO) [RFC5925] effort. It would be ideal, and reflect best common security practice, to have a fully specified key management protocol for negotiating TCP-AO's keying material, e.g., using certificates for peer authentication. However,
in the spirit of incremental deployment, we will first address issues like cryptographic algorithm agility, replay attacks, TCP session resetting in the base TCP-AO protocol before we layer key management on top of it.

However, in the spirit of incremental deployment, we will first address issues like cryptographic algorithm agility, replay attacks, TCP session resetting in the base TCP-AO protocol before we layer key management on top of it.

2.3. Goals

The goals and general guidance for the KARP work follow.

1. Provide authentication and integrity protection for messages on the wire of existing routing protocols.

2. Define a path to incrementally improve security of the routing infrastructure as explained in the earlier sections.

3. Ensure that the improved security solutions on currently running routing infrastructure equipment are deployable. This begs the consideration of the current state of processing power available on routers in the network today.

4. Operational deployability - A solution’s acceptability will also be measured by how deployable the solution is by common operator teams using common deployment processes and infrastructures. Specifically, we will try to make these solutions fit as well as possible into current operational practices and router deployment. This will be heavily influenced by operator input, to ensure that what we specify can -- and, more importantly, will -- be deployed once specified and implemented by vendors.

Deployment of incrementally more secure routing infrastructure in the Internet is the final measure of success. Measurably, we would like to see an increase in the number of surveyed respondents who report deploying the updated authentication and integrity mechanisms in their networks, as well as a sharp rise in usage for the total percentage of their network's routers.

Interviews with operators show several points about routing security. First, over 70% of operators have deployed transport connection protection via TCP-MD5 [RFC3562] on their exterior Border Gateway Protocol (eBGP) [ISR2008] sessions. Over 55% also deploy TCP-MD5 on their interior Border Gateway Protocol (iBGP) connections, and 50% make use of TCP-MD5 offered on some other internal gateway protocol (IGP). The survey states that "a considerable increase was observed over previous editions of the
survey for use of TCP MD5 with external peers (eBGP), internal peers (iBGP) and MD5 extensions for IGPs." Though the data is not captured in the report, the authors believe anecdotally that of those who have deployed TCP-MD5 somewhere in their network, only about 25-30% of the routers in their network are deployed with the authentication enabled. None report using IPsec [RFC4301] to protect the routing protocol, and this was a decline from the few that reported doing so in the previous year’s report. From our personal conversations with operators, of those using MD5, almost all report using one, manually-distributed key throughout the entire network. These same operators report that the single key has not been changed since it was originally installed, sometimes five or more years ago. When asked why, particularly for the case of protecting BGP sessions using TCP MD5, the following reasons are often given:

A. Changing the keys triggers a TCP reset, and thus bounces the links/adjacencies, undermining Service Level Agreements (SLAs).

B. For external peers, the difficulty of coordination with the other organization is an issue. Once they find the correct contact at the other organization (not always so easy), the coordination function is serialized and on a per peer/AS basis. The coordination is very cumbersome and tedious to execute in practice.

C. Keys must be changed at precisely the same time, or at least within 60 seconds (as supported by two major vendors) in order to limit connectivity outage duration. This is incredibly difficult to do, operationally, especially between different organizations.

D. Key change is perceived as a relatively low priority compared to other operational issues.

E. Lack of staff to implement the changes on a device-by-device basis.

F. There are three use cases for operational peering at play here: peers and interconnection with other operators, iBGP, and other routing sessions within a single operator, and operator-to-customer devices. All three have very different properties, and all are reported as cumbersome. One operator reported that the same key is used for all customer premise equipment (CPE). The same operator reported that if the customer mandated it, a unique key could be created, although the last time this occurred it created such an operational
headache that the administrators now usually tell customers that the option doesn’t even exist, to avoid the difficulties. These customer-unique keys are never changed, unless the customer demands so. The main threat at play here is that a terminated employee from such an operator who had access to the one (or several) keys used for authentication in these environments could easily wage an attack. Alternatively, the operator could offer the keys to others who would wage the attack. In either case, the attacker could then bring down many of the adjacencies, causing destabilization to the routing system.

5. Whatever mechanisms KARP specifies need to be easier to deploy than the current methods, and should provide obvious operational efficiency gains along with significantly better security and threat protection. This combination of value may be enough to drive much broader adoption.

6. Address the threats enumerated below in the "Threats" section (Section 3) for each routing protocol. Not all threats may be able to be addressed in the first specification update for any one protocol. Roadmaps will be defined so that both the security area and the routing area agree on how the threats will be addressed completely over time.

7. Create a re-usable architecture, framework, and guidelines for various IETF working groups who will address these security improvements for various Routing Protocols. The crux of the KARP work is to re-use the architecture, guidelines and the framework as much as possible across relevant Routing Protocols. For example, designers should aim to re-use the key management protocol that will be defined for BGP’s TCP-AO key establishment for as many other routing protocols as possible.

8. Bridge any gaps between IETF’s Routing and Security Areas by recording agreements on work items, roadmaps, and guidance from the cognizant Area Directors and the Internet Architecture Board (IAB).

2.4. Non-Goals

The following two goals are considered out-of-scope for this effort:

- Confidentiality of the packets on the wire. Once this roadmap is realized, we may revisit work on privacy.

- Message content validity (routing database validity). This work is being addressed in other IETF efforts, like SIDR.
2.5. Audience

The audience for this document includes:

- Routing Area working group chairs and participants - These people are charged with updates to the Routing Protocol specifications. Any and all cryptographic authentication work on these specifications will occur in Routing Area working groups, with close partnership with the Security Area. Co-advisors from the Security Area may often be named for these partnership efforts.

- Security Area reviewers of routing area documents - These people are delegated by the Security Area Directors to perform reviews on routing protocol specifications as they pass through working group last call or IESG review. They will pay particular attention to the use of cryptographic authentication and newly specified security mechanisms for the routing protocols. They will ensure that incremental security improvements are being made, in line with this roadmap.

- Security Area engineers - These people partner with routing area authors/designers on the security mechanisms in routing protocol specifications. Some of these security area engineers will be assigned by the Security Area Directors, while others will be interested parties in the relevant working groups.

- Operators - The operators are a key audience for this work, as the work is considered to have succeeded only if operators deploy the technology, presumably due to a perception of significantly improved security value coupled with relative similarity to deployment complexity and cost. Conversely, the work will be considered a failure if the operators do not care to deploy it, either due to lack of value or perceived (or real) over-complexity of operations. As a result, the GROW and OPSEC WGs should be kept squarely in the loop as well.
3. Threats

In this document we will use the definition of "threat" as defined in RFC4949 [RFC4949]: "a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm."

This section defines the threats that are in scope for the KARP effort. It also lists those threats that are explicitly out of scope for the KARP effort.

This document leverages the "Generic Threats to Routing Protocols" model, [RFC4593]. Specifically, the threats below were derived by reviewing [RFC4593], analyzing the KARP problem space relative to it, and simply listing the threats that are applicable to the KARP design teams’ work. This document categorizes [RFC4593] threats into those in scope and those out of scope for KARP. Each in-scope threat is discussed below, and its applicability to the KARP problem space is described. As such, the below text intentionally does not constitute a self-standing, complete threat analysis, but rather describes the applicability of the existing threat analysis [RFC4593] relevant to KARP.

Note: terms from [RFC4593] appear capitalized below -- e.g. OUTSIDERS -- so as to make explicit the term’s origin, and to enable rapid cross referencing to the source RFC.

For convenience, a terse definition of most [RFC4593] terms is offered here. Those interested in a more thorough description of routing protocol threat sources, motivations, consequences and actions will want to read [RFC4593] before continuing here.

3.1. Threat Sources

3.1.1. OUTSIDERS

One of the threats that will be addressed in this roadmap are those where the source is an OUTSIDER. An OUTSIDER attacker may reside anywhere in the Internet, have the ability to send IP traffic to the router, may be able to observe the router’s replies, and may even control the path for a legitimate peer’s traffic. OUTSIDERS are not legitimate participants in the routing protocol. The use of message authentication and integrity protection specifically aims to identify packets originating from OUTSIDERS.

KARP design teams will consider two specific use cases of OUTSIDERS: those on-path, and those off-path.
3.1.1. Threat Sources

o On-Path - These sources have control of a network resource or a tap that sits along the path of packets between the two routing peers. A "Man-in-the-Middle" (MitM) is an on-path attacker. From this vantage point, the attacker can conduct either active or passive attacks. An active attack occurs when the attacker actually places packets on the network as part of the attack. One active MitM attack relevant to KARP, an active wiretapping attack, occurs when the attacker tampers with packets moving between two legitimate router peers in such a way that both peers think they are talking to each other directly, when in fact they are actually talking to the attacker only. Protocols conforming to this roadmap will use cryptographic mechanisms to detect MitM attacks and reject packets from such attacks (i.e. treat them as not authentic). Passive on-path attacks occur when the attacker silently gathers data and analyses it to gain advantage. Passive activity by an on-path attacker may often eventually lead to an active attack.

o Off-Path - These sources sit on some network outside of that over which runs the packets between two routing peers. The source may be one or several hops away. Off-path attackers can launch active attacks, such as SPOOFING or denial-of-service (DoS) attacks, to name a few.

3.1.2. Stolen Keys

This threat source exists when an unauthorized entity somehow manages to gain access to keying material. Using this material, the attacker could send packets that pass the authenticity checks based on message authentication codes (MACs). The resulting traffic might appear to come from router A to router B, and thus the attacker could impersonate an authorized peer. The attacker could then adversely affect network behavior by sending bogus messages that appear to be authentic. The attack source possessing the stolen keys could be on-path, off-path, or both.

The obvious mitigation for stolen keys is to change the keys currently in use by the legitimate routing peers. This mitigation can be either reactive or pro-active. Reactive mitigation occurs when keys are changed only after having discovered that the previous keys fell into the possession of unauthorized users. The stolen keys, reactive mitigation case is highlighted here in order to explain a common operational situation where new keying material will become necessary with little or no advanced warning. In such a case new keys must be able to be installed and put into use very quickly, and with little operational expense. Pro-active mitigation occurs when an operator assumes that unauthorized possession will occur from time to time without being discovered, and the operator moves to new
keying material in order to cut short, or make nonexistent, an attacker's window of opportunity to use the stolen keys effectively.

In KARP, we can address the attack source with stolen keys by creating specifications that make it practical for the operator to quickly change keys without disruption to the routing system, and with minimal operational overhead. Operators can further mitigate the stolen keys case by habitually changing keys.

3.1.2.1. Terminated Employee

A terminated employee is an important example of a "stolen keys" threat source to consider. Staff attrition is a reality in routing operations, and so regularly causes the potential for a threat source. The threat source risk arises when a network operator who had been granted access to keys ceases to be an employee. If new keys are deployed immediately, the situation of a terminated employee can become a "stolen keys, pro-active" case, as described above, rather than a "stolen keys, reactive" case.

On one hand, terminated employees could be considered INSIDERS rather than OUTSIDERS, because at one point in time they were authorized to have the keys. On the other hand, they aren’t really a BYZANTINE attacker, which is defined to be an attack from an INSIDER, a legitimate router. Further, once terminated, the authorization granted to the terminated employee regarding the keys is revoked. If they maintain possession of the keys they are acting in an unauthorized way. If they go on to use those keys to launch an attack they are definitely acting in an unauthorized way. In this way the terminated employee becomes an OUTSIDER at the point of termination, they cease to be legitimate participants in the routing system. It behooves the operator to change the keys, to enforce the revocation of authorization of the old keys, in order to minimize the threat source’s window of opportunity.

Regardless of whether one considers a terminated employee an "insider" or an OUTSIDER, it is important to consider them a threat source, study the use case, and address the threats therein. In such a case within the KARP context, new keys must be able to be installed and made operational in the routing protocols very quickly, with zero impact to the routing system, and with little operational expense.

The threat source of the terminated employee and/or the detected-stolen-keys drives the requirement for quick and easy key rollover. The threat actions associated with these sources are mitigated if the operator has mechanisms in place (both inherent in the protocol, as well as built into their management systems) that allow them to roll the keys quickly with minimal impact to the routing system, at low
operational cost.

3.2. Threat Actions In Scope

These ATTACK ACTIONS are in scope for KARP:

- **SPOOFING** - when an unauthorized device assumes the identity of an authorized one. SPOOFING can be used, for example, to inject malicious routing information that causes the disruption of network services. SPOOFING can also be used to cause a neighbor relationship to form that subsequently denies the formation of the relationship with the legitimate router.

- **DoS attacks at the transport layer** - This is an example of SPOOFING. It can also be an example of FALSIFICATION and INTERFERENCE (see below). It occurs when an attacker sends spoofed packets aimed at halting or preventing the underlying protocol over which the routing protocol runs. For example, BGP running over TLS will still not solve the problem of being able to send a spoofed TCP FIN or TCP RST and causing the BGP session to go down. Since this attack depends on spoofing, operators are encouraged to deploy proper authentication mechanisms to prevent such attacks. Specification work should ensure that Routing Protocols can operate over transport sub-systems in a fashion that is resilient to such DoS attacks.

- **FALSIFICATION** - an action whereby an attacker sends false routing information. To falsify the routing information, an attacker has to be either the originator or a forwarder of the routing information. FALSIFICATION may occur by an ORIGINATOR, or a FORWARDER, and may involve OVERCLAIMING, MISCLAIMING, or MISTATEMENT of network resource reachability. We must be careful to remember that in this work we are only targeting FALSIFICATION from OUTSIDERS as may occur from tampering with packets in flight, or sending entirely false messages. FALSIFICATION from BYZANTINES (see the Threats Out of Scope section below) are not addressed by the KARP effort.

- **INTERFERENCE** - when an attacker inhibits the exchanges by legitimate routers. The types of INTERFERENCE addressed by this work include:

  A. **ADDING NOISE**

  B. **REPLAYING OUT-DATED PACKETS**

  C. **INSERTING MESSAGES**
D. CORRUPTING MESSAGES

E. BREAKING SYNCHRONIZATION

F. Changing message content

- DoS attacks using the authentication mechanism - This includes an attacker sending packets that confuse or overwhelm a security mechanism itself. An example is initiating an overwhelming load of spoofed routing protocol packets that contain a MAC, so that the receiver needs to spend the processing cycles to check the MAC, only to discard the spoofed packet, consuming substantial CPU resources. Another example is when an attacker sends an overwhelming load of keying protocol initiations from bogus sources.

- Brute Force Attacks Against Password/Keys - This includes either online or offline attacks where attempts are made repeatedly using different keys/passwords until a match is found. While it is impossible to make brute force attacks on keys completely unsuccessful, proper design can make such attacks much harder to succeed. For example, the key length should be sufficiently long so that covering the entire space of possible keys is improbable using computational power expected to be available 10 years out or more. Using per session keys is another widely used method for reducing the number of brute force attacks as this would make it difficult to guess the keys.

3.3. Threat Actions Out of Scope

Threats from BYZANTINE sources -- faulty, misconfigured, or subverted routers, i.e., legitimate participants in the routing protocol -- are out of scope for this roadmap. Any of the attacks described in the above section (Section 2.1) that may be levied by a BYZANTINE source are therefore also out of scope, e.g. FALSIFICATION, or unauthorized message content by a legitimate authorized peer.

In addition, these other attack actions are out of scope for this work:

- SNIFFING - passive observation of route message contents in flight. Data privacy, as achieved by data encryption, is the common mechanism for preventing SNIFFING. While useful, especially to prevent the gathering of data needed to perform an off-path packet injection attack, data encryption is out-of-scope for KARP.
INTERFERENCE due to:

A. NOT FORWARDING PACKETS - cannot be prevented with cryptographic authentication. Note: If sequence numbers with sliding windows are used in the solution (as is done, for example, in IPsec’s ESP [RFC4303] and BFD [RFC5880]), a receiver can at least detect the occurrence of this attack.

B. DELAYING MESSAGES - cannot be prevented with cryptographic authentication. Note: Timestamps can be used to detect delays.

C. DENIAL OF RECEIPT - cannot be prevented with cryptographic authentication

D. UNAUTHORIZED MESSAGE CONTENT - the work of the IETF’s SIDR working group (http://www.ietf.org/html.charters/sidr-charter.html).

E. DoS attacks not involving the routing protocol. For example, a flood of traffic that fills the link ahead of the router, so that the router is rendered unusable and unreachable by valid packets is NOT an attack that KARP will address. Many such examples could be contrived.
4. Requirements for KARP Work Phase 1, the Update to a Routing Protocol’s Existing Transport Security

The KARP Design Guide [RFC6518], Section 4.1 describes two distinct work phases for the KARP effort. This section addresses requirements for the first work phase only, "Work Phase 1", the update to a routing protocol’s existing transport security. "Work Phase 2", a framework and usage of a KMP, will be addressed in a future requirements document.

The following list of requirements SHOULD be addressed by a KARP Work Phase 1 security update to any Routing Protocol (according to section 4.1 of the KARP Design Guide [RFC6518] document). IT IS RECOMMENDED that any Work Phase 1 security update to a Routing Protocol contain a section of the specification document that describes how each of the below requirements are met. It is further RECOMMENDED that justification be presented for any requirements that are NOT addressed.

1. Clear definitions of which elements of the transmitted data (frame, packet, segment, etc.) are protected by the authentication mechanism

2. Strong cryptographic algorithms, as defined and accepted by the IETF security community, MUST be specified. The use of non-standard or unpublished algorithms SHOULD BE avoided.

3. Algorithm agility for the cryptographic algorithms used in the authentication MUST be specified, i.e. more than one algorithm MUST be specified and it MUST be clear how new algorithms MAY be specified and used within the protocol. This requirement exists because research identifying weaknesses in cryptographic algorithms can cause the security community to reduce confidence in some algorithms. Breaking a cipher isn’t a matter of if, but when it will occur. Having the ability to specify alternate algorithms (algorithm agility) within the protocol specification to support such an event is essential. Mandating two algorithms provides both a redundancy, and a mechanism for enacting that redundancy when needed. Further, the mechanism MUST describe the generic interface for new cryptographic algorithms to be used, so that implementers can use algorithms other than those specified, and so that new algorithms may be specified and supported in the future.

4. Secure use of PSKs, offering both operational convenience and a baseline level of security, MUST be specified.
5. Routing protocols should be able to detect and reject replayed messages. For non TCP based protocols like OSPF [RFC2328], IS-IS [RFC1195], etc., two routers are said to have a session up if they are able to exchange protocol packets. Packets captured from one session must not be able to be re-sent and accepted during a later session. Additionally, replay mechanisms must work correctly even in the presence of routing protocol packet prioritization by the router.

A. There is a specific case of replay attack combined with spoofing that must be addressed. In several routing protocols (e.g., OSPF [RFC2328], IS-IS [RFC1195], BFD [RFC5880], RIP [RFC2453], etc.), all speakers share the same key (K) on a broadcast segment. The ability to run a MAC operation with K is used for identity validation, and (currently) no other identity validation check is performed. Assume there are four routers using authentication on a LAN, R1 – R4. Also assume attacker "Z", who is NOT a legitimate neighbor, is observing and recording packets on the same LAN segment. Z captures a packet from R1, and changes the source IP, spoofing it to that of R2, then sends the packet on the LAN. Z does not have K, but in this case it does not matter because R1 already performed the MAC operation, and Z simply re-uses that MAC. R3 and R4 will process the packet as if coming from R2, the MAC check will return valid, and they will update their route tables accordingly. R3 and R4 have confirmed that the MAC was created by someone holding K, but not that it was actually sent by R2. This is a well known attack with known solutions. Some string must be added into the MAC operation that uniquely identifies the sender. Said string must also be located in the packet such that if that string were to be altered after the MAC operation, it would be detected by the receiver. Examples of solutions used in other protocols include sequence numbers with sliding acceptance windows, time stamps, IP header info (SRC, DST), unique identifiers which are temporarily bound to an IP Address.

6. A change of security parameters Requires, and even forces, a change of session traffic keys. The specific security parameters for the various routing protocols will differ, and will be defined by each protocols design team. Some examples may include: master key, key lifetime, cryptographic algorithm, etc. If one of these configured parameters changes, then a new session traffic key must immediately be established using the updated parameters. The routing protocol security mechanisms MUST support this behavior.
7.  Intra-session re-keying which occurs without a break or interruption to the current routing session, and, if possible, without data loss, MUST be specified. Keys need to be changed periodically, for operational confidentiality (e.g. when an administrator who had access to the keys leaves an organization) and for entropy purposes, and a re-keying mechanism enables the operators to execute the change without productivity loss.

8.  Efficient re-keying SHOULD be provided. The specification SHOULD support rekeying during a session without needing to try/compute multiple keys on a given packet. The rare exception will occur if a routing protocols design team can find no other way to re-key and still adhere to the other requirements in this section.

9.  New mechanisms must resist DoS attacks described as in-scope in Section 3.2. Routers protect the control plane by implementing mechanisms to filter completely or rate limit traffic not required at the control plane level (i.e., unwanted traffic). Typically line rate packet filtering capabilities look at information at or below the IP and transport (TCP or UDP) headers, but do not include higher layer information. Therefore the new mechanisms shouldn’t hide nor encrypt the information carried in the IP and transport layers in control plane packets.

10.  Mandatory cryptographic algorithms and mechanisms MUST be specified for a routing protocol. Further, the protocol specification MUST define default security mechanism settings for all implementations to use when no explicit configuration is provided. To understand the need for this requirement, consider the case where a routing protocol mandates 3 different cryptographic algorithms for a MAC operation. If company A implements algorithm 1 as the default for this protocol, while company B implements algorithm 2 as the default, then two operators who enable the security mechanism with no explicit configuration other than a PSK will experience a connection failure. It is not enough that each implementation implement the 3 mandatory algorithms; one default must further be specified in order to gain maximum out-of-the-box interoperability.

11.  For backward compatibility reasons manual keying MUST be supported.

12.  Architecture of the specification SHOULD consider and allow for future use of a KMP.
13. The authentication mechanism in the Routing Protocol MUST be decoupled from the key management system used. It MUST be obvious how the keying material was obtained, and the process for obtaining the keying material MUST exist outside of the Routing Protocol. This will allow for the various key generation methods, like manual keys and KMPs, to be used with the same Routing Protocol mechanism.

14. Convergence times of the Routing Protocols SHOULD NOT be materially affected. "Materially" is defined here as anything greater than a 5% increase in convergence time. Changes in the convergence time will be immediately verifiable by convergence performance test beds already in use by most router vendors and service providers. Note that convergence is different than boot time. Also note that convergence time has a lot to do with the speed of processors used on individual routing peers, and this processing power increases by Moore’s law over time, meaning that the same route calculations and table population routines will decrease in duration over time. Therefore, this requirement should be considered only in terms of total number of protocol packets that must be exchanged, and less for the computational intensity of processing any one message. Alternatively this can be simplified by saying that the new mechanisms should only result in a minimal increase in the number of routing protocol packets passed between the peers.

15. The changes to or addition of security mechanisms SHOULD NOT cause a refresh of route advertisements or cause additional route advertisements to be generated.

16. Router implementations provide prioritized treatment for certain protocol packets. For example, OSPF HELLO packets and ACKs are prioritized for processing above other OSPF packets. The security mechanism SHOULD NOT interfere with the ability to observe and enforce such prioritization. Any effect on such priority mechanisms MUST be explicitly documented and justified. Replay protection mechanisms provided by the routing protocols MUST work even if certain protocol packets are offered prioritized treatment.

17. Routing protocols MUST only send minimal information regarding the authentication mechanisms and the parameters in its protocol packets. One reason for this is to keep the Routing Protocols as clean and focused as possible, and load security negotiations into the future KMP as much as possible. Another reason is to avoid exposing any security negotiation information unnecessarily to possible attackers on the path.
18. Routing protocols that rely on the IP header (or information separate from routing protocol payload) to identify the neighbor that originated the packet, MUST either protect the IP header or provide some other means to authenticate the neighbor. [RFC6039] describes some attacks that are based on this.

19. Every new KARP-developed security mechanisms MUST support incremental deployment. It will not be feasible to deploy a new Routing Protocol authentication mechanism throughout a network instantaneously. It also may not be possible to deploy such a mechanism to all routers in a large autonomous system (AS) at one time. Proposed solutions MUST support an incremental deployment method that provides some benefit for those who participate. Because of this, there are several requirements that any proposed KARP mechanism should consider.

A. The Routing Protocol security mechanism MUST enable each router to configure use of the security mechanism on a per-peer basis where the communication is peer-to-peer (unicast).

B. Every new KARP-developed security mechanism MUST provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment. Message formatting in a fully secured environment MAY be handled in a non-backward compatible fashion though care must be taken to ensure that routing protocol packets can traverse intermediate routers that don’t support the new format.

C. In an environment where both secured and non-secured systems are interoperating, a mechanism MUST exist for secured systems to identify whether a peer intended the messages to be secured.

D. In an environment where secured service is in the process of being deployed, a mechanism MUST exist to support a transition free of service interruption (caused by the deployment per se).

20. The introduction of mechanisms to improve routing security may increase the processing performed by a router. Since most of the currently deployed routers do not have hardware to accelerate cryptographic operations, these operations could impose a significant processing burden under some circumstances. Thus proposed solutions should be evaluated carefully with regard to the processing burden they may impose, since deployment may be impeded if network operators perceive that a
solution will impose a processing burden which either incurs substantial capital expense, or threatens to destabilize routers.

21. Given the number of routers that would require the new authentication mechanisms in a typical ISP deployment, solutions can increase their appeal by minimizing the burden imposed on all routers in favor of confining significant work loads to a relatively small number of devices. Optional features or increased assurance that engenders more pervasive processing loads MAY be made available for deployments where the additional resources are economically justifiable.

22. New authentication and security mechanisms should not rely on systems external to the routing system (the equipment that is performing forwarding) in order for the routing system to function. In order to ensure the rapid initialization and/or return to service of failed nodes it is important to reduce reliance on these external systems to the greatest extent possible. Proposed solutions SHOULD NOT require connections to external systems, beyond those directly involved in peering relationships, in order to return to full service. It is however acceptable for the proposed solutions to require post initialization synchronization with external systems in order to fully synchronize the security information.

If authentication and security mechanisms rely on systems external to the routing system, then there MUST be one or more options available to avoid circular dependencies. For example, if it is not acceptable to have the operation of OSPF within a routing domain depend upon correct operation of a security protocol, have correct operation of the security protocol depend upon the ability to exchange IP packets between remote systems, and have the ability to exchange IP packets between remote systems depend upon correct operation of the same instance of OSPF within the same routing domain. However, it is okay to have operation of multicast routing and/or inter-domain routing depend upon operation of a security protocol, which depends upon exchange of IP packets between remote systems, which depends upon the correct operation of OSPF for unicast routing. Similarly it would be okay to have the operation of OSPF depend upon a security protocol, which in turn uses an out of band network to exchange information with remote systems.
5. Security Considerations

This document is mostly about security considerations for the KARP efforts, both threats and requirements for addressing those threats. More detailed security considerations were placed in the Security Considerations section of the KARP Design Guide [RFC6518] document.

Spoofing by a Legitimate Neighbor - In several routing protocols (e.g.) all speakers share the same key, a group key, on a broadcast segment. Possession of the group key itself is used for identity validation, and no other identity check is used. Under these conditions an attack exists where one neighbor (e.g. Router 1, or R1) can masquerade as a different neighbor, R2, by sending spoofed packets using R2 as the source IP address. When other neighbors, R3 and R4, receive these packets, they will calculate the MAC successfully, and process its contents as if it originated from R2. SPOOFING this way, the attacker can succeed in several different types of attacks, including FALSIFICATION and INTERFERENCE. The source of such an attack is a BYZANTINE actor, since the attack originates from a legitimate actor in the routing system, and such sources are out of scope for KARP. This type of attack has been well documented in the group keying problem space, and it’s non-trivial to solve. The common method used to prevent this type of attack is to use a unique key for each sender rather than a group key. Other solutions exist within the group keying realm, but they come with significant increases in complexity and computational intensity. KARP protocol design teams should consider this attack and determine, based on costs and benefits, if a plausible solution can be employed, and document the decision, either way.
6. IANA Considerations

This document has no actions for IANA.
7. Acknowledgements

The majority of the text for version -00 of this document was taken from "Roadmap for Cryptographic Authentication of Routing Protocol Packets on the Wire", draft-lebovitz-karp-roadmap, authored by Gregory M. Lebovitz.

Brian Weis provided significant assistance in handling the many comments that came back during IESG review.

We would like to thank the following people for their thorough reviews and comments: Brian Weis, Yoshifumi Nishida, Stephen Kent, Vishwas Manral.

Author Gregory M. Lebovitz was employed at Juniper Networks, Inc. for the majority of the time he worked on this document, though not at the time of its publishing. Thus Juniper sponsored much of this effort.
8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Gregory Lebovitz
Aptos, California  95003
USA

Email: gregory.ietf@gmail.com

Manav Bhatia
Alcatel-Lucent
Bangalore,
India

Phone:
Email: manav.bhatia@alcatel-lucent.com
Key Management for Pairwise Routing Protocol
draft-mahesh-karp-rkmp-01

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 September 9, 2012.

Copyright Notice
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 Router Key Management Protocol (RKMP) that largely makes use of currently defined IKEv2 [RFC5996] protocol and extends it to allow 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. Pre-shared 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

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

SKEYSEED: 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 Section 1.2 of IKEv2 [RFC5996], and refers to the value derived from SKEYSEED (defined in Section 2.14 of IKEv2 [RFC5996]). SK_d is used to derive new keys (e.g., for TCP-AO) as follows:

<TCP-AO master key> = prf+(SK_d, Ni | Nr)

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
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).
o 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.

o 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

The IKE_SA_INIT exchange defined in Internet Key Exchange Protocol Version 2 [RFC5996] is used in RKMP. The IKE_SA_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              HttpHeaders, SAr1, KEr, Nr, [CERTREQ,]
IKE_SA_INIT

3.2. IKE_AUTH

Next, the network devices perform an IKE_AUTH exchange defined in RFC 5996. However, the SA payloads contain the routing protocol specific security policies rather than IPsec policies (SAi2, SAr2 defined in RFC 5996), and the TS payloads contains routing protocol specific traffic selectors. Policy definitions for routing protocols is 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, SAi2, TSi, TSr}     HttpHeaders, SK {IDr, [CERT,] AUTH,
                                    SAr2, TSi, TSr}
IKE_AUTH

In the IKE_AUTH exchange, the Initiator proposes one or more sets of policies for one routing protocol in the SAi2. 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]}

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 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.
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 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.

This memo defines new Proposal substructure definitions, which allow protocol participants to exchange proposals for routing protocol policy. Figure 2 defines new Protocol IDs that can be negotiated within an IKEv2 SA payload.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Protocol ID</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP-AO</td>
<td>TBD-1</td>
<td>RFC 5925</td>
</tr>
<tr>
<td>LDP Discovery Key</td>
<td>TBD-2</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Protocol IDs

The following section describes the SA Payload Transforms Substructures that are to be used with these Protocol IDs.
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. 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. TCP-AO

The TCP-AO [RFC5925] transform payload is specified to negotiate the TCP-AO policies and contains the following fields.

```
+----------------------------------+
| 0 (last) or 3 | RESERVED    |        Transform Length       |
| 0 (last) or 3 | SendID     | Auth Alg       |     KDF       |     Flags     |
+----------------------------------+
```

Figure 3: 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. Initial values are defined in Cryptographic Algorithms for the TCP Authentication Option [RFC5926]. Parenthetical names in Figure 4 refer to the values assigned in the Cryptographic Algorithms for TCP-AO Registration [TCP-AO-REG].
Table of Auth Algs

<table>
<thead>
<tr>
<th>Auth Alg</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>0</td>
</tr>
<tr>
<td>HMAC-SHA-1-96 (SHA1)</td>
<td>1</td>
</tr>
<tr>
<td>AES-128-CMAC-96 (AES128)</td>
<td>2</td>
</tr>
<tr>
<td>Standards Action</td>
<td>3-140</td>
</tr>
<tr>
<td>Private Use</td>
<td>241-255</td>
</tr>
</tbody>
</table>

Figure 4: TCP-AO 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].

<table>
<thead>
<tr>
<th>KDF</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>0</td>
</tr>
<tr>
<td>KDF_HMAC_SHA1</td>
<td>1</td>
</tr>
<tr>
<td>KDF_AES_128_CMAC</td>
<td>2</td>
</tr>
<tr>
<td>Standards Action</td>
<td>3-240</td>
</tr>
<tr>
<td>Private Use</td>
<td>241-255</td>
</tr>
</tbody>
</table>

Figure 5: TCP-AO Key Derivation Functions

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

```
+----------------+
| O | 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.

- **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).

\[
<TCP-AO master key> = prf+(SK_d, Ni | Nr)
\]
4.1.1.2. LDP Discovery Key

TBD

4.2. Traffic Selector Payload

The Traffic Selector (TS) payload allows an RP peer to identify packet flows that are to be protected with the policy in the SA payload. Unlike IPsec, routing protocols have well-defined flows, and there is no need to specify them to the specificity of IPsec policy. The document defines a new TS type for routing protocols as shown in Figure 6. The TS payload defined in this document includes only the routing protocol identifier that is to be protected.

<table>
<thead>
<tr>
<th>TS Type</th>
<th>Rtg. Prot. ID</th>
<th>Selector Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6

- TS Type (1 octet) - TBD-3 for all routing protocols
- Rtg. Prot. ID (1 octet) - Specifies the routing protocol identifier for the current negotiation.

<table>
<thead>
<tr>
<th>Routing (RT) Protocol</th>
<th>Protocol ID</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>BGP</td>
<td>1</td>
<td>RFC 4271</td>
</tr>
<tr>
<td>LDP</td>
<td>2</td>
<td>RFC 5036</td>
</tr>
<tr>
<td>MSDP</td>
<td>3</td>
<td>RFC 3618</td>
</tr>
<tr>
<td>PIM PORT</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>PCEP</td>
<td>5</td>
<td>RFC 5440</td>
</tr>
<tr>
<td>Unassigned</td>
<td>6-240</td>
<td></td>
</tr>
<tr>
<td>Private Use</td>
<td>240-255</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Routing Protocol IDs

Exchanges including traffic selectors (i.e., IKE_AUTH, CREATE_CHILD_SA) include two TS payloads, one for the initiator policy and one for the responder policy. In the case of RPs the policy is symmetric and both payloads contain the same routing protocol ID value.
5. Operation Details

5.1. General

RKMP 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 RKMP 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 RKMP neighbor adjacency between them. This is typically done by the Initiator (RKMP speaker) initiating a RKMP RP_INIT exchange mentioned in the section 2.1 towards its RKMP peer. As part of RP_INIT exchange, RKMP will send a message to the RKMP peer’s IKEv2 port. 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 RKMP speaker, the RKMP neighbor adjacency may be torn down or kept around as explained in section 3.

The master key data received from RKMP peers are 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 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 inform RKMP 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 RKMP initiates RKMP peering with the neighbor and starts initial RKMP 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 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, RKMP initiates peering connections with its appropriate RKMP peers to announce the updated key related information. RKMP may also receive an updated key related information from its peers which gets installed in KMDB. Whenever RKMP updates KMDB with updated key information from its peers, it notifies client routing protocols of its updates.

7. IANA Considerations

New Protocol-IDs (as described in Figure 2) are to be allocated in the IKEv2 Security Protocol Identifiers registry.

A new Traffic Selector Type (as described in Figure 7) is to be allocated in the IKEv2 Traffic Selector Types registry.

Several new registries are to be defined as part of a new RKMP Protocol Registry. These are described in Figure 4, Figure 5, and Figure 7.
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.

10. References

10.1. Normative References


10.2. Informative References


[IKEV2-PARMS]

[TCP-AO-REG]

Authors’ Addresses

Mahesh Jethanandani
Independent

Phone: 
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:

Sam Hartman
Painless Security

Phone:
Fax:
Email: hartmans@painless-security.com
URI:
The Use of G-IKEv2 for Multicast Router Key Management
draft-tran-karp-mrmp-01

Abstract

The G-IKEv2 key management protocol protects group traffic, usually in the form of IP multicast communications between a set of network devices. This memo defines extensions to G-IKEv2 allowing it to protect the communications of routing protocols using a one-to-many or many-to-many communications model.

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 September 10, 2012.

Copyright Notice

Copyright (c) 2012 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. Requirements Language .................................. 3
2. Overview of Group Key Management ............................. 3
3. Exchanges ...................................................... 4
4. Header and Payload Formats ................................... 5
   4.1. Group Security Association Payload ....................... 5
       4.1.1. GSA TEK Payload .................................... 6
5. IANA Considerations ............................................ 8
6. Security Considerations ....................................... 9
7. Acknowledgements ................................................ 9
8. References ..................................................... 9
   8.1. Normative References ..................................... 9
   8.2. Informative References ................................... 9
Authors’ Addresses ............................................... 9
1.  Introduction

The G-IKEv2 protocol [I-D.yeung-g-ikev2] has been defined to distribute group policy and keys to a group of network devices. It uses IKEv2 protocols, incorporating payloads similar to GDOI [RFC6407]. This memo describes a mode of using G-IKEv2 to protect routing protocols using one-to-many communication models (e.g., OSPF, PIM), and is known as G-IKEv2 for Multicast Router Key Management (G-IKEv2-MRKM).

1.1.  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].

2.  Overview of Group Key Management

When a group of network devices need to communicate using multicast communications, the devices need to share keying material and the policy associated with that keying material. A group key management (GKM) protocol is used to securely distribute this keying material and associated policy. Typically each network device (also known as a group member (GM)) needing to participate in the group "register" to a group controller/key server (GCKS), during which mutual authentication and authorization occur. The GCKS also distributes current group policy and keying material to the group member over an authenticated and encrypted session. When G-IKEv2 is used, this is achieved in four messages: two to setup the encrypted session (identical to the IKEv2 [RFC5996] IKE_SA_INIT protocol, and two to authenticate, authorize, and distribute group policy to the GM (similar in construction to the IKEv2 IKE_AUTH protocol).

A GKM protocol typically uses a "rekey" protocol to efficiently distribute replacement keying material and associated policy to GMs. However, this is primarily an optimization for scalability. This optimization is less desirable for a small group of routing protocol participants. In this memo, we describe how the group can utilize the registration protocol for both initial keying and rekeying purposes.

G-IKEv2-MRKM is a GKM use case where a group of network routers participating in a routing protocol using a multicast transport act as GMs. Which device takes the role of GCKS is not specified by this memo, but operationally it is most reliable for one of the GMs to take that role. Additionally, for routing protocol reliability the routers may require the ability to use redundant key servers and to
elect a key server from available group members. This memo does not address redundancy, however the protocols in this memo should be compatible with the redundancy strategy described in [I-D.hartman-karp-mrkmp].

3. Exchanges

The exchange of private keying material between two network devices using a dedicated key management protocol is a common requirement. There is no need to define an entirely new protocol for routing protocols having this requirement when existing mature protocol exchanges and methods have been vetted. This memo extends the G-IKEv2 protocol exchanges, state machine, and policy definitions.

The following two exchanges enable the group member to register to the key server to get the policy, traffic selector and keys used to communicate with others group member.

The IKE_SA_INIT exchange is a two-message exchange allows the group member and key server devices to negotiate cryptographic algorithms, exchange nonces, and do a Diffie-Hellman exchange [DH]. At the conclusion of the IKE_SA_INIT, the group member (e.g., router) and key server can exchange private messages. For the details of this exchange, refer to IKE_SA_INIT in RFC 5996.

<table>
<thead>
<tr>
<th>Group Member (Initiator)</th>
<th>Key Server (Responder)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SAi1, KEi, Ni</td>
<td>HDR, SAr1, KEr, Nr, [CERTREQ],</td>
</tr>
</tbody>
</table>

Next, the group member and key server devices perform a GSA_AUTH, which is substantially the same as the IKE_AUTH exchange defined in RFC 5996, except that the SA, TS1, TSr payloads in IKE_AUTH are not used. Policy and traffic selectors are pushed from the key server to group member using new payloads GSA and KD. For the details of the rest of the exchange please refer to Section 4 of [I-D.yeung-g-ikev2]. Section 4 of this document includes additional GSA definitions specifically for the purpose of protecting routing protocol traffic.

<table>
<thead>
<tr>
<th>Group Member (Initiator)</th>
<th>Key Server (Responder)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, IDg}</td>
<td>HDR, SK {IDr, [CERT,] AUTH, GSA, KD}</td>
</tr>
</tbody>
</table>
In the GSA_AUTH exchange, the group member sends the identification of the group to which it wants to join or register. The key server authenticates and authorizes the group member and pushes the policy, traffic selector in GSA payload, and the key in the KD payload to the group member. At the successful conclusion of the GSA_AUTH exchange, the group member has policy and keying material to securely communicate with others group members that also registered with the key server. With this IKEv2 SA established between GM and KS, the GM can request for policy and keys of an additional group using the GSA_CLIENT_SERVER exchange. In the GSA_CLIENT_SERVER exchange, the GM will send group ID that it wants to join, where the key server response will include policy (GSA) and key material (KD).

Group Member (Initiator)                  Key Server (Responder)
----------------------------------------                  ---------------
HDR, SK {IDg}                  -->            HDR, SK { GSA, KD, [D]}
               <--

Once a GSA_AUTH has completed the group member and key server MAY destroy the G-IKEv2 SA. However when the number of group members is small, as is usually the case for routing protocol participants, it is RECOMMENDED for them to maintain the G-IKEv2 association SA for the key server to notify group members that they should re-register in order to obtain new group policy. This notify exchange replaces a separate rekey mechanism optimized for large group.

In some cases, a GCKS may need to change the group policy and/or rekey before current keys expire. In cases where the G-IKEv2 rekey protocol is not used the GCKS can send an INFORMATIONAL exchange with a Notify payload directing the group member to re-register using a REGISTER_REQUESTED status notify message (value TBD). This event triggers a GSA_CLIENT_SERVER exchange, as described above. The response to GSA_CLIENT_SERVER from the KS may include Delete payloads instructing the group member to delete old SAs.

4. Header and Payload Formats

The protocol defined in this memo uses a HDR defined in RFC5996. GSA exchange types and payloads described in this section are added to same IANA registry containing G-IKEv2 definitions.

4.1. Group Security Association Payload

The Group Security Association (GSA) payload contains one or more sets of policy that a router is willing to use to protect a routing protocol. It is identical to the GSA payload described in Section 4.3 of [I-D.yeung-g-ikev2]. This memo makes no changes to this
payload.

4.1.1. GSA TEK Payload

One of GSA types is the Traffic Encryption Key (TEK) policy. The TEK describes the Traffic Encryption Policy. This document define new protocol ID of TEK protocol specific payload for routing protocol OSPFv2, OSPFv3 and PIM.

4.1.1.1. TEK OSPFv2 Protocol-Specific Payload

TEK OSPFv2 Protocol Specific Payload contains SPI, the authentication algorithm and key lifetime.

The TEK OSPF protocol specific payload is defined as follows:

<table>
<thead>
<tr>
<th>Protocol ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
</tr>
<tr>
<td>GSA_PROTO_IPSEC_ESP</td>
<td>1</td>
</tr>
<tr>
<td>GSA_PROTO_IPSEC_AH</td>
<td>2</td>
</tr>
<tr>
<td>GSA_PROTO_OSPFv2</td>
<td>TBD</td>
</tr>
<tr>
<td>GSA_PROTO_OSPFv3</td>
<td>TBD</td>
</tr>
<tr>
<td>GSA_PROTO_PIM</td>
<td>TBD</td>
</tr>
</tbody>
</table>
SPI - (1 octet) Secure Parameter Index will be used in OSPFv2 header as Key ID (RFC 2328, Appendix D)

Auth algo - (2 octets) Authentication Algorithm
- Keyed-MD5 (defined in RFC 2328, Appendix D)
- HMAC-SHA-1 (defined in RFC 5709, Section 3)
- HMAC-SHA-256 (defined in RFC 5709, Section 3)
- HMAC-SHA-384 (defined in RFC 5709, Section 3)
- HMAC-SHA-512 (defined in RFC 5709, Section 3)

TEK Attribute - Key lifetime, define in
(draft-yeung-g-ikev2-03, section 4.5)

4.1.1.2. TEK OSPFv3 and PIM IPsec Protocol-Specific Payload

OSPFv3 and PIM IPSEC protocol specific payload similar to GIKEv2 TEK payload for ESP and AH. This payload doesn’t include the traffic selector as protocol-ID value in the GSA TEK payload already indicate OSPFv3 or PIM traffic, because the traffic selectors are well known and unchanging values. However it should be noted that a site requiring alternative transforms can use the G-IKEv2 definitions for ESP and AH to define this policy.
SPI (4 octets) - Secure Parameter Index

Transform - Same as G-IKEv2 TEK transform defined in
(draft-yeung-g-ikev2-03, section 4.5)
Where transform type can be 1 (Encryption Algorithm)
for ESP and/or 3 (Integrity Algorithm) for AH.

<table>
<thead>
<tr>
<th>Description</th>
<th>Trans. Used In</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
</tr>
<tr>
<td>Encryption Algorithm (ENCR)</td>
<td>1   ESP</td>
</tr>
<tr>
<td>Integrity Algorithm (INTEG)</td>
<td>3   AH, optional in ESP</td>
</tr>
<tr>
<td>Extended Sequence Numbers (ESN)</td>
<td>5   AH and ESP</td>
</tr>
</tbody>
</table>

Transform Type 1 (Encryption Algorithm)

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
<th>Defined In</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCR_NULL</td>
<td>11</td>
<td>(RFC2410)</td>
</tr>
<tr>
<td>ENCR_AES_CBC</td>
<td>12</td>
<td>(RFC3602)</td>
</tr>
</tbody>
</table>

Transform Type 3 (Integrity Algorithm)

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
<th>Defined In</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>0</td>
<td>Defined In</td>
</tr>
<tr>
<td>AUTH_HMAC_MD5_96</td>
<td>1</td>
<td>(RFC2403)</td>
</tr>
<tr>
<td>AUTH_HMAC_SHA1_96</td>
<td>2</td>
<td>(RFC2404)</td>
</tr>
</tbody>
</table>

TEK Attribute - Key lifetime, define in
(draft-yeung-g-ikev2-03, section 4.5)

5. IANA Considerations

G-IKEv2 [I-D.yeung-g-ikev2] defines a new registry. This memo adds the following definitions to this registry. (TBD)
6. Security Considerations

This document describes a use case of group key management using G-IKEv2. The security considerations in [I-D.yeung-g-ikev2] directly apply to this memo.

7. Acknowledgements

Sam Hartman and Dacheng Zhang previously published the MRKMP protocol [I-D.hartman-karp-mrkmp], which includes many operations and protocol elements in common with this memo.

8. References

8.1. Normative References

[I-D.yeung-g-ikev2]
Rowles, S., Yeung, A., Tran, P., and Y. Nir, "Group Key Management using IKEv2", draft-yeung-g-ikev2-03 (work in progress), July 2011.


8.2. Informative References


[I-D.hartman-karp-mrkmp]

Authors’ Addresses

Paulina Tran  
Cisco Systems  
170 Tasman Drive  
San Jose, California CA  
USA

Phone: +1 (408) 526-8902  
Fax:  
Email: ptran@cisco.com  
URI:  

Brian Weis  
Cisco Systems  
170 Tasman Drive  
San Jose, California CA  
USA

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