Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 16, 2011

M. Bhatia
Alcatel-Lucent

April 14, 2011

Analysis of Protocol Independent Multicast Sparse Mode (PIM-SM) Security According to KARP Design Guide
draft-bhatia-karp-pim-gap-analysis-00

Abstract

This document analyzes Protocol Independent Multicast Sparse Mode (PIM-SM) according to the guidelines set forth in the KARP Design Guide.

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 October 16, 2011.

1. Introduction

This document performs the initial analysis of the current state of Protocol Independent Multicast Sparse Mode (PIM-SM) [RFC4601] according to the requirements of [I-D.ietf-karp-design-guide]

[RFC5796] describes mechanisms to authenticate the PIM-SM link-local messages using the IP security (IPsec) Encapsulating Security Payload (ESP) [RFC4303] or (optionally) the Authentication Header (AH) [RFC4302] .

This document specifies manual key management as mandatory to implement, i.e., that all implementations MUST support, and provides the necessary structure for an automated key management protocol that the PIM routers may use.

However, some gaps remain between the current state and the requirements for manually keyed routing security expressed in the [I-D.ietf-karp-threats-reqs] document. This document explores these gaps and proposes directions for addressing the gaps.

2. Current State and Gap Analysis

[RFC5796] describes how IPsec can be used to secure and authenticate PIM-SM protocol packets. It mandates the use of manual keying and optionally provides support for an automated group key management mechanism. However, it leaves the procedures for implementing automated group key management to other documents and does not discuss how this can be done.

[RFC5796] uses manually configured keys, rather than some automated key management protocol , since no suitable key management mechanism is available at this time. This is because PIM-SM adjacencies are formed on a one-to-many basis and most key management mechanisms are designed for a one-to-one communication model. Since [RFC5796] uses manual keying it clearly states that it provides no protection against both inter-session and intra-session replay attacks. This can be exploited in several ways.

Since multiple PIM-SM routers can exist on a single link, it would be

worth noting that setting up IPsec Security Associations (SAs) manually can be a very tedious process. The routers might not even support IPsec, rendering automatic key negotiation either impractical (in those platforms where an extra license has to be obtained for using IPsec) or infeasible (in those platforms where IPsec support is not available at all).

While I don't yet see a need to prioritize certain PIM-SM packets over the others, it should be noted that this would be extremely difficult to achieve since PIM-SM uses IPsec for its security and authentication.

[RFC4601] requires all PIM-SM routers to configure an IPsec Security Association (SA) when sending PIM Register packets to each Rendezous Point (RP). This can become highly unscalable as the number of RPs increase or in case of Anycast-RP [RFC4610] deployment where each PIM-SM router close to the source will need to establish IPsec tunnels to all PIM-SM routers in the Anycast-RP set.

Similarly, the Security Policy Database at each Rendezvous Point should be configured to choose an SA to use when sending Register-Stop messages. Because Register-Stop messages are unicast to the destination DR, a different SA and a potentially unique SPI are required for each DR.

In order to simplify the management problem, [RFC4601] suggests using the same authentication algorithm and authentication parameters, regardless of the sending RP and regardless of the destination DR. While this alleviates the management problem by some extent it still requires a unique SA on each DR which can result in a significant scaling issue as the size of the PIM-SM network grows.

In order to encourage deployment of PIM-SM security, an authentication option is required that does not have the deployment challenges of IPsec. We thus need an authentication mechanism alternate to IPsec as part of the first phase of the KARP design guide where we secure the routing protocols using manual keying.

The new mechanism should work for both the Unicast and Multicast PIM-SM routing exchanges. It should also provide both inter-session and intra-session replay protection that has been spelled out in the [I-D.ietf-karp-threats-reqs] document.

3. Security Considerations

TBD

4. IANA Considerations

This document places no new request to IANA

5. Acknowledgements

I would like to thank Stig Venaas and Bill Atwood for reviewing and providing feedback on this draft.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC5796] Atwood, W., Islam, S., and M. Siami, "Authentication and Confidentiality in Protocol Independent Multicast Sparse Mode (PIM-SM) Link-Local Messages", RFC 5796, March 2010.

6.2. Informative References

[I-D.ietf-karp-design-guide] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", draft-ietf-karp-design-guide-02 (work in progress), March 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-01 (work in progress), October 2010.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol",

RFC 4306, December 2005.

[RFC4610] Farinacci, D. and Y. Cai, "Anycast-RP Using Protocol Independent Multicast (PIM)", RFC 4610, August 2006.

Author's Address

Manav Bhatia
Alcatel-Lucent
India

Email: manav.bhatia@alcatel-lucent.com

Working Group
Internet-Draft
Intended status: Informational
Expires: April 26, 2012

U. Chunduri
A. Tian
W. Lu
Ericsson Inc.,
October 24, 2011

KARP IS-IS security gap analysis
draft-chunduri-karp-is-is-gap-analysis-00

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.

Table of Contents

1. Introduction

This document analyzes the current state of Intermediate system to Intermediate system (IS-IS) protocol according to the requirements set forth in [I-D.ietf-karp-design-guide] 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 it self 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 [I-D.ietf-karp-design-guide]. 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.weis-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 [I-D.ietf-karp-design-guide] 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 dual environments", RFC 1195, December 1990.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, October 2008.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

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-design-guide]
Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", draft-ietf-karp-design-guide-02 (work in progress), March 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 Digital Signatures", RFC 2154, June 1997.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010.

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

Working Group U. Chunduri
Internet-Draft A. Tian
Intended status: Informational Ericsson Inc.,
Expires: April 27, 2012 October 25, 2011

Using IKEv2 with TCP-AO
draft-chunduri-karp-using-ikev2-with-tcp-ao-00

Abstract

This document analyzes the TCP based pairwise Routing Protocol (RP) requirements for IKEv2 Key Management Protocol (KMP). This document discusses the various authentication methods available for peer authentication in IKEv2 KMP and the specific Security Association (SA) requirements for IKEv2 protocol to protect the TCP based pairwise RPs.

Table of Contents

1. Introduction

Threat analysis for TCP based routing protocols (BGP [RFC4271], PCEP [RFC5440], MSDP [RFC3618] and LDP [RFC5036]) is detailed in [ietf-karp-routing-tcp-analysis]. KARP design guide [ietf-karp-design-guide] suggests various requirements and options for getting keys to protect the routing protocols and recommends using KMP to automate the key establishment and rekeying to protect the routing protocols.

This document analyzes the TCP based pairwise Routing Protocol (RP) requirements for IKEv2[RFC5996] Key Management Protocol (KMP).

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 raft of authentications methods, Section 2 discusses various Symmetric, Asymmetric and EAP based KMP authentication options available for all TCP based routing protocols. This document also provides guidelines for designing suitable approach for routing environments.

This document analyzes one approach, which minimizes the changes for routing protocols (BGP, PCEP, MSDP and LDP) to be integrated with KMP. This document defines the interface between all TCP based pairwise routing protocols and the TCP-AO [RFC5925]. The interface between IKEv2 KMP and the TCP-AO for session parameter negotiation, key establishment and rekeying is also defined in Section 3.

Currently IKEv2 can establish only Security Association (SA) for IP Security (IPSec). Few extensions are needed for IKEv2 to establish SA for TCP based routing protocols that use TCP-AO. Section 4 discusses a brief summary of the extensions required for IKEv2 protocol for key establishment, traffic selectors negotiation and Security Association (SA) establishment for TCP based routing protocols.

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

EAP - Extensible Authentication Protocol

KDF - Key Derivation Function

KMP - Key Management Protocol (auto key management)

MKM - Manual Key management Protocols

NONCE - Number Once

2. Applicable Authentications methods

One advantage that IKEv2 provides is the largest selection of authentication methods suitable for various environments. The goal of this section is to look at various KMP authentications options available and recommend suitable options for deployment with routing protocols.

As some of the authentication mechanisms are optional in IKEv2, one mandatory authentication mechanism from the list below need 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.

2.1. Symmetric key based authentication

IKEv2 [RFC5996] allow for authentication of the IKEv2 peers using a symmetric pre-shared key. For symmetric pre-shared key based peer authentication, the 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 for bootstrapping the IKE_SA, then one of the EAP methods as described in Section 2.3 need 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.

2.2. Asymmetric key based authentication

Another peer authentication mechanism for IKEv2 is with asymmetric key certificates or public key signatures. This approach will use the 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 option, there is no need for out-of-band shared key between the peers for mutual 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.

2.3. EAP based authentication

In addition to supporting authentication using shared secrets and public key signatures, IKEv2 also supports authentication based on Extensible Authentication Protocol (EAP), defined in [RFC3748]. EAP is an authentication framework that supports multiple authentication mechanisms. IKEv2 provides EAP authentication since it was recognized that 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, since EAP allows users to choose a method suitable for existing credentials. Routing protocols today use password based pre-shared key 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 (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, as 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] and out of these, password based methods [RFC4746], [RFC5931], [RFC6124] can offer potential EAP alternative for pre-shared key only authentication.

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

3. Interfaces

Section 1.2 of TCP-AO [RFC5925] says "..we recommend the use of IPsec and IKE, especially where IKE's level of existing support for parameter negotiation, session key negotiation, or rekeying are desired." - but such interface is not defined. As IKEv2 [RFC5996] is being discussed as the potential KMP for routing protocols, this section defines the interface between IKEv2 KMP and TCP-AO. This section also analyzes the interface between TCP based routing protocols (BGP, LDP, MSDP, PCEP) and the TCP-AO module.

3.1. RP interface to TCP-AO

When a routing protocol is configured to use KMP (by not specifying the keys or through some other means), configured authentication algorithms and rekey life time is provisioned in the TCP-AO MKT. This can be achieved at the time of opening the socket. With this, the MKT created in TCP-AO contains all the configured information other than the keys to protect the underlying session.

3.2. TCP-AO interface to KMP

There needs to be a way to trigger the KMP to initiate negotiation with provisioned parameters, to rekey and to maintain the negotiated sessions. In this section, we define a common interface between TCP-AO and KMP that can be used by all TCP based routing protocols. (An alternative approach is to define an interface for each routing protocols to trigger KMP directly. This alternative is not of scope for this document.)

Following are the details of the interface between TCP-AO and KMP:

1. When the first SYN packet on the session is initiated, a trigger to negotiate the session specific parameters with all provisioned authentication algorithms and optionally key lifetime should be given to KMP.

2. A KMP session identifier need to be stored and should be used for rekeying the existing session.

3. MKT IDs as specified in Section 3.1 of TCP-AO [RFC5925], requires 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 MKT when the KMP key(s) are populated in MKT.

4. KMP negotiated authentication algorithm and optionally life time for traffic keys for each session, need to be populated in MKT.

5. Trigger may also be needed at the time of rekeying any particular session. Implementations can pro-actively negotiate new traffic keys before the life time of current traffic keys expire.

4. Extensions required for IKEv2 protocol

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

a. To create a new KMP for routing protocols based on IKEv2 as proposed in [mahesh-karp-rkmp].

b. Extend IKEv2 to make it 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. Authors acknowledge, some of the items below are already discussed in KARP WG but the details presented here are different.

Routing protocols by deploying extended IKEv2 KMP, can continuously benefit from the new authentication methods and any other new features which might be added.

4.1. Non IPSec DOI

IKEv2 is designed for performing mutual authentication with the peer and establishing and maintaining Security Associations for IPSec. IKEv2 defined IKE_AUTH and CREATE_CHILD_SA exchange, consist of payloads, and processing guidelines for IPSec Domain of Interpretation (DOI) and this need to be generalized to exchange other protocol specific parameters.

IKEv2 CREATE_CHILD_SA exchange today can also be used to rekey the IKE SA and the master key. This document do not propose any changes

or extensions to re-establishing IKE SA through CREATE_CHILD_SA exchange.

4.1.1. Security Association Extensions

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

1. Protocol ID, to be added in the proposal substructure with TCP-AO as new protocol.

2. Integrity Algorithm (INTEG), defined in the transform substructure need to be mandated for the new TCP-AO Protocol. Authentication algorithms as defined in [RFC5926] should be extended to the current list in IKEv2.

3. New transform type need to be created to represent the TCP-AO KeyIDs. Initiator KeyID represents the SendID and the Responder KeyID represents the RecvID in the TCP-AO MKT.

4. Diffie-Hellman group (D-H) transform type can be used for TCP_AO proposal as an optional transform.

5. Valid transform types for TCP-AO with mandatory and optional types need to be listed. Attribute negotiation rules need to be extended for TCP-AO protocol.

4.2. Simple Traffic Selectors Negotiation

The Traffic Selectors defined in IKEv2 [RFC5996] has huge potential to negotiate the particular traffic to be secured, agreeable to both initiator and responder. But for routing protocol SA, traffic selectors negotiation present a simple case and does not require any changes. A single connection or multiple connections with a different source port to be protected, can be negotiated with one 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 with the approach and this section will be revisited further.

5. IANA Considerations

This document defines no new namespaces.

6. Security Considerations

This document does not introduce any new security threats for IKEv2 [RFC5925] or TCP-AO [RFC5996] protocol. For more detailed security considerations please refer the Security Considerations section of the KARP Design Guide [I-D.ietf-karp-design-guide] document as well as KARP threat document [I-D.ietf-karp-threats-reqs].

7. Acknowledgements

The authors would like to thank Joel Halpern for initial discussions and providing feedback on the document.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

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

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC5998] Eronen, P., Tschofenig, H., and Y. Sheffer, "An Extension for EAP-Only Authentication in IKEv2", RFC 5998, September 2010.

8.2. Informative References

[I-D.ietf-karp-design-guide]
Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", draft-ietf-karp-design-guide-02 (work in progress), March 2011.

[I-D.ietf-karp-routing-tcp-analysis]
Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Security According to KARP Design

Guide", draft-ietf-karp-routing-tcp-analysis-00 (work in progress), June 2011.

[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.mahesh-karp-rkmp]
Jethanandani, M., Weis, B., Patel, K., Zhang, D., and S. Hartman, "Key Management for Pairwise Routing Protocol", draft-mahesh-karp-rkmp-00 (work in progress), October 2011.

[RFC3618] Fenner, B. and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4746] Clancy, T. and W. Arbaugh, "Extensible Authentication Protocol (EAP) Password Authenticated Exchange", RFC 4746, November 2006.

[RFC4754] Fu, D. and J. Solinas, "IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 4754, January 2007.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5931] Harkins, D. and G. Zorn, "Extensible Authentication Protocol (EAP) Authentication Using Only a Password", RFC 5931, August 2010.

[RFC6124] Sheffer, Y., Zorn, G., Tschofenig, H., and S. Fluhrer, "An EAP Authentication Method Based on the Encrypted Key Exchange (EKE) Protocol", RFC 6124, February 2011.

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

INTERNET-DRAFT
Internet Engineering Task Force (IETF)
Intended Status: Standards Track
Expires: 30 April 2012

R. Housley
Vigil Security
T. Polk
NIST

30 October 2011

Database of Long-Lived Symmetric Cryptographic Keys
<draft-ietf-karp-crypto-key-table-02.txt>

Status of this Memo

Copyright Notice

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 wit 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, ProtcolSpecificInfo, 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

(6) NotBefore <= T <= NotAfter.

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 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

[RFC3562] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", RFC 3562, July 2003.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", RFC 4107, BCP 107, June 2005.

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

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

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

KARP Working Group G. Lebovitz
Internet Draft
Intended status: Informational M. Bhatia
Expires: February, 2012 Alcatel-Lucent
October 2011

Keying and Authentication for Routing Protocols (KARP) Design Guidelines

draft-ietf-karp-design-guide-07.txt

Status of this Memo

Copyright Notice

Abstract

This document is one of a series concerned with defining a roadmap of protocol specification work for the use of modern cryptographic mechanisms and algorithms for message authentication in routing protocols. In particular, it defines the framework for a key management protocol that may be used to create and manage session keys for message authentication and integrity.

Conventions used in this document

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]

Table of Contents

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 RFC 4948 [RFC4948]. Section 8.1 of that document states that "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:

- Increased security mechanisms and practices for operating routers.
- Cleaning up the Internet Routing Registry repository [IRR], and securing both the database and the access, so that it can be used for routing verifications.
- Specifications for cryptographic validation of routing message content.
- Securing 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 bullet, securing the packets on the wire of the routing protocol exchanges. Thus, it is concerned with guidelines for describing issues and techniques for protecting the messages 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 this information. Such assurances are provided by other mechanisms and are outside the scope of this document and work that relies on it.

This document uses the terminology "on the wire" to talk about the information used by routing 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. Other documents that will analyze individual protocols will need to indicate how they use the term "on the wire".

The term "routing transport" is used to refer to the layer that exchanges the routing protocols. This can be TCP, UDP, or even direct link level messaging in the case of some routing protocols. The term is used here to allow a referent for discussing both common and disparate issues that affect or interact with this dimension of the routing systems. The term is used here to refer generally to the set of mechanisms and exchanges underneath the routing protocol, whatever that is in specific cases.

Readers must refer to the [I-D.ietf-karp-threats-reqs] for a clear definition of the scope, goals, non goals and the audience for the design work being undertaken in KARP WG.

2. Categorizing Routing Protocols

This document places the routing protocols into two categories according to their requirements for authentication. We hope these categories will allow design teams to focus on security mechanisms for a given category. Further, we hope that the each protocol in the group will be able to reuse the authentication mechanism. It is also hoped that, down the road we can create one Key Management Protocol (KMP) per category (if not for several categories) so that the work can be easily leveraged by for use in the various Routing Protocol groupings. KMPs are useful for allowing simple, automated updates of the traffic keys used in a base protocol. KMPs replace the need for humans, or OSS routines, to periodically replace keys on running systems. It also removes the need for a chain of manual keys to be chosen or configured on such systems. When configured properly, a KMP will enforce the key freshness policy among peers by keeping track of the key lifetime and negotiating a new key at the defined interval.

2.1. Category: Message Transaction Type

The first category defines three types of messaging transactions used on the wire by the base Routing Protocol. They are:

One-to-One

One peer router directly and intentionally delivers a route update specifically to one other peer router. Examples are BGP [RFC4271], LDP [RFC5036], BFD [RFC5880] and RSVP-TE [RFC3209] [RFC3473] [RFC4726] [RFC5151]. Point-to-point modes

of both IS-IS [RFC1195] and OSPF [RFC2328], when sent over both traditional point-to-point links and when using multi-access layers, may both also fall into this category.

One-to-Many

A router peers with multiple other routers on a single network segment -- i.e. on link local -- such that it creates and sends one route update message which is intended for multiple peers. Examples would be OSPF and IS-IS in their broadcast, non-point-to-point mode and Routing Information Protocol (RIP) [RFC2453].

Multicast

Multicast protocols have unique security properties because they are inherently group-based protocols and thus have group keying requirements at the routing level where link-local routing messages are multicasted. Also, at least in the case of PIM-SM [RFC4601], some messages are sent unicast to a given peer(s), as is the case with router-close-to-sender and the "Rendezvous Point". Some work for application layer message security has been done in the Multicast Security working group (MSEC) and may be helpful to review, but is not directly applicable.

These categories affect both the routing protocol view of the communication, and the actual message transfer. As a result, some message transaction types for a few routing protocols, may be mixtures, for example using broadcast where multicast might be expected, or using unicast to deliver what looks to the routing protocol like broadcast or multicast.

This may include any semantics of the communication that impact the routing protocol, such as when source identity or path properties of the communication path are used by the routing algorithm, e.g., as when BGP infers routing table entry quality from the persistence of the TCP connection over which they are received.

Protocol security analysis documents produced in KARP need to pay attention both to the semantics of the communication, and the techniques that are used for the message exchanges.

2.2. Category: Peer vs Group Keying

The second category is the keying mechanism that will be used to distribute the session keys to the routing transports.They are:

Peer keying

One router sends the keying messages only to one other router, such that a one-to-one, uniquely keyed security association (SA) is established between the two routers (e.g., BGP, BFD and LDP).

Group Keying

One router creates and distributes a single keying message to multiple peers. In this case a group SA will be established and used among multiple peers simultaneously. Group keying exists for protocols like OSPF [RFC2328], and also for multicast protocols like PIM-SM [RFC4601].

3. Consider the future existence of a Key Management Protocol

When it comes time for the KARP WG to design a re-usable model for a Key Management Protocol (KMP), [RFC4107] should be consulted.

When conducting the design work on a manually-keyed version of a routing protocol's authentication mechanism, consideration must be made for the eventual use of a KMP. In particular, design teams must consider what parameters would need to be handed to the routing protocols by a KMP.

Examples of parameters that might need to be passed are: a security association identifier (e.g. IPsec SPI, or TCP-AO's KeyID), a key lifetime (which may be represented either in bytes or seconds), the cryptographic algorithms being used, the keys themselves, and the directionality of the keys (i.e., receive versus the sending keys)

3.1. Consider Asymmetric Keys

The use of asymmetric keys can be a very powerful way to authenticate machine peers as used in routing protocol peer exchanges. If generated on the machine, and never moved off the machine, these keys will not need to be changed if an administrator leaves the organization. Since the keys are random they are far less susceptible to off-line dictionary and guessing attacks.

An easy and simple way to use asymmetric keys is to start by having the router generate a public/private key pair. At the time of this writing, the recommended key size for algorithms based on integer factorization cryptography like RSA is 1024 bits and 2048 bits for extremely valuable keys like the root key pair used by a certification authority. It is believed that a 1024-bit RSA key is equivalent in strength to 80-bit symmetric keys and 2048-bit RSA keys to 112-bit symmetric keys [RFC3766]. Elliptic Curve Cryptography [RFC4492] (ECC) appears to be secure with shorter keys than those needed by other asymmetric key algorithms. NIST guidelines [NIST-800-57] state that ECC keys should be twice the length of equivalent strength symmetric key algorithms. Thus, a 224-bit ECC key would roughly have the same strength as a 112-bit symmetric key.

Many routers have the ability to be remotely managed using Secure Shell (SSH) Protocol [RFC4252] and [RFC4253]. As such, routers will also have the ability to generate and store an asymmetric key pair, because this is the common authentication method employed by SSH when an administrator connects to a router for management sessions.

Once an asymmetric key pair is generated, the KMP generating security association parameters and keys for routing protocol may use the machine's asymmetric keys for the identity proof. The form of the identity proof could be raw keys, the more easily administrable self-signed certificate format, or a PKI-issued [RFC5280] certificate credential.

Regardless which form we eventually standardize, the proof of this identity presentation can be as simple as a strong hash, which could be represented in a human readable and transferable form of some pairs of ASCII characters. More complex, but also more secure, the identity proof could be verified through the use of a PKI system's revocation checking mechanism, (e.g. Certificate Revocation List (CRL) or OCSP responder). If the SHA-1 fingerprint is used, the solution could be as simple as loading a set of neighbor routers' peer ID strings into a table and listing the associated fingerprint string for each ID string. In most organizations or peering points, this list will not be longer than a thousand or so routers, and often the list will be much shorter. In other words, the entire list for a given organization's router ID and hash could be held in a router's configuration file, uploaded, downloaded and moved about at will. And it doesn't matter who sees or gains access to these fingerprints, because they can be distributed publicly as it needn't be kept secret.

3.2. Cryptographic Keys Life Cycle

Cryptographic keys should have a limited lifetime and may need to be changed when an operator who had access to them leaves. Using a key chain also does not help as one still has to change all the keys in the key chain when an operator having access to all those keys leaves the company. Additionally, key chains will not help if the routing transport subsystem does not support rolling over to the new keys without bouncing the routing sessions and adjacencies. So the first step is to fix the routing stack so that routing protocols can change keys without breaking or bouncing the adjacencies.

An often cited reason for limiting the lifetime of a key is to minimize the damage from a compromised key. It could be argued that it is likely a user will not discover an attacker has compromised the key if the attacker remains "passive" and thus relatively frequent key changes will limit any potential damage from compromised keys.

Another threat against the long-lived key is that one of the systems storing the key, or one of the users entrusted with the key, will be subverted. So, while there may not be cryptographic motivations of changing the keys, there could be system security motivations for rolling the key.

Although manual key distribution methods are subject to human error and frailty, more frequent manual key changes might actually increase the risk of exposure as it is during the time that the keys are being changed that they are likely to be disclosed. In these cases, especially when very strong cryptography is employed, it may be more prudent to have fewer, well controlled manual key distributions rather than more frequent, poorly controlled manual key distributions. In general, where strong cryptography is employed, physical, procedural, and logical access protection considerations often have more impact on the key life than do algorithm and key size factors.

For incremental deployments we could start by associating life times with the send and the receive keys in the key chain for the long-lived keys. This is an incremental approach that we could use until the cryptographic keying material for individual sessions is derived from the keying material stored in a database of long-lived cryptographic keys as described in [I-D.ietf-karp-crypto-key-table]. A key derivation function (KDF) and its inputs are also specified in the database of long-lived cryptographic keys; session-specific values based on the routing protocol are input to the KDF. Protocol-specific

key identifiers may be assigned to the cryptographic keying material for individual sessions if needed.

The long-lived cryptographic keys used by the routing protocols can be either inserted manually in a database or can make use of an automated key management protocol to do this.

4. RoadMap

4.1. Work Phases on any Particular Protocol

It is believed that improving security for any routing protocol will be a two phase process. The first phase would be to modify routing protocols to support modern cryptography algorithms and key agility. The second phase would be to design and move to an automated key management mechanism. This is like a crawl, walk and run process. In order for operators to accept these phases, we believe that the key management protocol should be clearly separated from the routing transport. This would mean that the routing transport subsystem is oblivious to how the keys are derived, exchanged and downloaded as long as there is something that it can use. It is like having a routing protocol configuration switch that requests the security module for the "KARP security parameters" so that it can refer to some module written, maintained, and operated by security experts and insert those parameters in the routing exchange.

The desired end state for the KARP work contains several items. First, the people desiring to deploy securely authenticated and integrity validated packets between routing peers have the tools specified, implemented and shipping in order to deploy. These tools should be fairly simple to implement, and not more complex than the security mechanisms to which the operators are already accustomed. (Examples of security mechanisms to which router operators are accustomed include: the use of asymmetric keys for authentication in SSH for router configuration, the use of pre-shared keys (PSKs) in TCP MD5 for BGP protection, the use of self-signed certificates for HTTPS access to device Web-based user interfaces, the use of strongly constructed passwords and/or identity tokens for user identification when logging into routers and management systems.) While the tools that we intend to specify may not be able to stop a deployment from using "foobar" as an input key for every device across their entire routing domain, we intend to make a solid, modern security system that is not too much more difficult than that. In other words, simplicity and deployability are keys to success. The Routing Protocols will specify modern cryptographic algorithms and security mechanisms. Routing peers will be able to employ unique, pair-wise keys per peering

instance, with reasonable key lifetimes, and updating those keys on a regular basis will be operationally easy, causing no service interruption.

Achieving the above described end-state using manual keys may be pragmatic only in very small deployments. However, manual keying in larger deployments will be too burdensome for operators. Thus, the second goal is to support key life cycle management with a KMP. We expect that both manual and automated key management will co-exist in the real world.

In accordance with the desired end state just described, we define two main work phases for each Routing Protocol:

1. Enhance the Routing Protocol’s current authentication mechanism(s). This work involves enhancing a Routing Protocol’s current security mechanisms in order to achieve a consistent, modern level of security functionality within its existing key management framework. It is understood and accepted that the existing key management frameworks are largely based on manual keys. Since many operators have already built operational support systems (OSS) around these manual key implementations, there is some automation available for an operator to leverage in that way, if the underlying mechanisms are themselves secure. In this phase, we explicitly exclude embedding or creating a KMP. Refer to [I-D.ietf-karp-threats-reqs] for the list of the requirements for Phase 1 work.

2. Develop an automated key management framework. The second phase will focus on the development of an automated keying framework to facilitate unique pair-wise (group-wise, where applicable) keys per peering instance. This involves the use of a KMP. The use of automatic key management mechanisms offers a number of benefits over manual keying. Most importantly it provides fresh traffic keying material for each session, thus helping to prevent inter-connection replay attacks. A KMP is also helpful because it negotiates unique, pair wise, random keys without administrator involvement. It negotiates several SA parameters like algorithms, modes, and parameters required for the secure connection, thus providing interoperability between endpoints with disparate capabilities and configurations. In addition it could also include negotiating the key life times. The KMP can thus keep track of those lifetimes using counters, and can negotiate new keys and parameters before they expire, again, without administrator interaction. Additionally, in the event of a breach, changing the KMP key will immediately cause a rekey to occur for the Traffic Key, and those new Traffic Keys will be installed and used in the current connection. In summary, a KMP provides a protected channel between the peers through which they can negotiate and pass important data required to exchange proof of identities, derive Traffic Keys, determine re-keying, synchronize their keying state, signal various keying events, notify with error messages, etc.

4.2. Work Items Per Routing Protocol

Each Routing Protocol will have a team (the [Routing_Protocol]-KARP team) working on incrementally improving the security of a Routing Protocol. These teams will have the following main work items:

PHASE 1:

Characterize the RP

Assess the Routing Protocol to see what authentication and integrity mechanisms it has today. Does it needs significant improvement to its existing mechanisms or not? This will include determining if modern, strong security algorithms and parameters are present and if the protocol supports key agility without bouncing adjacencies.

Define Optimal State

List the requirements for the Routing Protocol's session key usage and format to contain modern, strong security algorithms and mechanisms, per the Requirements document [I-D.ietf-karp-threats-reqs]. The goal here is to determine what is needed for the Routing Protocol to be used securely with at least manual key management.

Gap Analysis

Enumerate the requirements for this protocol to move from its current security state, the first bullet, to its optimal state, as listed just above.

Transition and Deployment Considerations

Document the operational transition plan for moving from the old to the new security mechanism. Will adjacencies need to bounce? What new elements/servers/services in the infrastructure will be required? What is an example work flow that an operator will take? The best possible case is if the adjacency does not break, but this may not always be possible.

Define, Assign, Design

Create a deliverables list of the design and specification work, with milestones. Define owners. Release one or more documents.

PHASE 2:

KMP Analysis

Review requirements for KMPs. Identify any nuances for this particular routing protocol's needs and its use cases for a KMP. List the requirements that this Routing Protocol has for being able to be used in conjunction with a KMP. Define the optimal state and check how easily it can be decoupled from the KMP.

Gap Analysis

Enumerate the requirements for this protocol to move from its current security state to its optimal state, with respect to the key management.

Define, Assign, Design

Create a deliverables list of the design and specification work, with milestones. Define owners. Generate the design and document work for a KMP to be able to generate the Routing Protocol's session keys for the packets on the wire. These will be the arguments passed in the API to the KMP in order to bootstrap the session keys for the Routing Protocol.

There will also be a team formed to work on the base framework mechanisms for each of the main categories.

5. Routing Protocols in Categories

This section groups the Routing Protocols into categories, according to attributes set forth in Categories Section (Section 2). Each group will have a design team tasked with improving the security of the Routing Protocol mechanisms and defining the KMP requirements for their group, then rolling both into a roadmap document upon which they will execute.

BGP, LDP, PCEP and MSDP

These Routing Protocols fall into the category of the one-to-one peering messages, and will use peer keying protocols. BGP [RFC4271], PCEP [RFC5440] and MSDP [RFC3618] messages are transmitted over TCP, while LDP [RFC5036] uses both UDP and TCP. A team will work on one mechanism to cover these TCP unicast protocols. Much of the work on the Routing Protocol update for its existing authentication mechanism has already occurred in the TCPM Working Group, on the TCP-AO [RFC5925] document, as well as its cryptography-helper document, TCP-AO-CRYPTO [RFC5926]. However, TCP-AO cannot be used for discovery exchanges carried in LDP as those are carried over UDP. A separate team might want to look at LDP. Another exception is the mode where LDP is used directly on the LAN. The work for this may go into the Group keying category (along with OSPF) as mentioned below.

OSPF, IS-IS, and RIP

The Routing Protocols that fall into the category Group Keying *with one-to-many peering) includes OSPF [RFC2328], IS-IS [RFC1195] and RIP [RFC2453]. Not surprisingly, all these routing protocols have two other things in common. First, they are run on a combination of the OSI datalink layer 2, and the OSI network layer 3. By this we mean that they have a component of how the routing protocol works

which is specified in Layer 2 as well as in Layer 3. Second, they are all internal gateway protocols(IGPs). The keying mechanisms will be much more complicated to define for these than for a one-to-one messaging protocol.

BFD

Because it is less of a routing protocol, per se, and more of a peer liveness detection mechanism, Bidirectional Forwarding Detection (BFD) [RFC5880] will have its own team. BFD is also different from the other protocols covered here as it works on millisecond timers and would need separate considerations to mitigate the potential for DoS attacks. It also raises interesting issues [RFC6039] with respect to the sequence number scheme that is generally deployed to protect against replay attacks as this space can rollover quite frequently because of the rate at which BFD packets are generated.

RSVP and RSVP-TE

The Resource reSerVation Protocol [RFC2205] allows hop-by-hop authentication of RSVP neighbors, as specified in [RFC2747]. In this mode, an integrity object is attached to each RSVP message to transmit a keyed message digest. This message digest allows the recipient to verify the identity of the RSVP node that sent the message, and to validate the integrity of the message. Through the inclusion of a sequence number in the scope of the digest, the digest also offers replay protection.

[RFC2747] does not dictate how the key for the integrity operation is derived. Currently, most implementations of RSVP use a statically configured key, on a per interface or per neighbor basis.

RSVP relies on a per peer authentication mechanism, where each hop authenticates its neighbor using a shared key or a certificate.

Trust in this model is transitive. Each RSVP node trusts explicitly only its RSVP next hop peers, through the message digest contained in the INTEGRITY object [RFC2747]. The next hop RSVP speaker in turn trusts its own peers and so on. See also the document "RSVP security properties" [RFC4230] for more background.

The keys used for protecting the RSVP messages can be group keys (for example distributed via GDOI [RFC3547], as discussed in [I-D.weis-gdoi-mac-tek]).

The trust an RSVP node has to another RSVP node has an explicit and an implicit component. Explicitly the node trusts the other node to maintain the integrity (and, optionally confidentiality) of RSVP messages depending on whether authentication or encryption (or both) are used. This means that the message has not been altered or its contents seen by another, non-trusted node. Implicitly each node trusts the other node to maintain the level of protection specified within that security domain Note that in any group key management scheme, like GDOI, each node trusts all the other members of the group with regard to data origin authentication.

RSVP TE [RFC3209] [RFC3473] [RFC4726] [RFC5151] is an extension of the RSVP protocol for traffic engineering. It supports the reservation of resources across an IP network and is used for establishing MPLS label switch paths (LSPs), taking into consideration network constraint parameters such as available bandwidth and explicit hops. RSVP-TE signaling is used to establish both intra and inter-domain TE LSPs.

When signaling an inter-domain RSVP-TE LSP, operators may make use of the security features already defined for RSVP-TE [RFC3209]. This may require some coordination between domains to share keys ([RFC2747],[RFC3097]), and care is required to ensure that the keys are changed sufficiently frequently. Note that this may involve additional synchronization, should the domain border nodes be protected with Fast Reroute, since the merge point (MP) and point of local repair (PLR) should also share the key.

For inter-domain signaling for MPLS-TE, the administrators of neighboring domains must satisfy themselves as to the existence of a suitable trust relationship between the domains. In the absence of such a relationship, the administrators should decide not to deploy inter-domain signaling, and should disable RSVP-TE on any inter-domain interfaces.

KARP will currently be working only on RSVP-TE as the native RSVP lies outside the scope of the WG charter.

PIM-SM and PIM-DM

Finally, the multicast protocols PIM-SM [RFC4601] and PIM-DM [RFC3973] will be grouped together. PIM-SM multicasts routing information (Hello, Join/Prune, Assert) on a link-local basis, using a defined multicast address. In addition, it specifies unicast communication for exchange of information (Register, Register-Stop) between the router closest to a group sender and the "rendezvous point" (RP). The RP is typically not "on-link" for a particular router. While much work has been done on multicast security for application-layer groups, little has been done to address the problem of managing hundreds or thousands of small one-to-many groups with link-local scope. Such an authentication mechanism should be considered along with the router-to-Rendezvous Point authentication mechanism. The most important issue is ensuring that only the "authorized neighbors" get the keys for (S,G), so that rogue routers cannot participate in the exchanges. Another issue is that some of the communication may occur intra-domain, e.g. the link-local messages in an enterprise, while others for the same (*,G) may occur inter-domain, e.g. the router-to-Rendezvous Point messages may be from one enterprise's router to another.

One possible solution proposes a region-wide "master" key server (possibly replicated), and one "local" key server per speaking router. There is no issue with propagating the messages outside the link, because link-local messages, by definition, are not forwarded. This solution is offered only as an example of how work may progress; further discussion should occur in this work team. Specification of a link-local protection mechanism for PIM-SM is defined in [RFC4601], and this mechanism has been updated in PIM-SM-LINKLOCAL [RFC5796]. However, the KMP part is completely unspecified, and will require work outside the expertise of the PIM working group to accomplish, another example of why this roadmap is being created.

6. Supporting Incremental Deployment

It is imperative that the new authentication and security mechanisms defined support incremental deployment, as it is not feasible to deploy a new routing protocol authentication mechanism throughout the network instantaneously. One of the goals of KARP WG is to add incremental security to existing mechanisms rather than replacing them. Delivering better deployable solutions to which vendors and operators can migrate to is more important than getting a perfect security solution. It may also not be possible to deploy such a mechanism to all

routers in a large AS at one time. This means that the designers must work on this aspect of authentication mechanism for the routing protocol that they are working on. The mechanisms must provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment.

7. Denial of Service Attacks

Denial of Service (DoS) attacks must be kept in mind when designing KARP solutions. [I-D.ietf-karp-threats-reqs] describes DoS attacks that are in scope for the KARP work. Protocol designers should ensure that the new cryptographic validation mechanisms must not provide an attacker with an opportunity for DoS attacks. Cryptographic validation, while typically cheaper than signing, is still an incremental cost. If an attacker can force a system to validate many packets multiple times then this could be a potential DoS attack vector. On the other hand, if the authentication procedure is itself quite CPU intensive, then overwhelming the CPU with multiple bogus packets can bring down the system. In this case, the authentication procedure itself aids the DoS attack.

There are some known techniques to reduce the cryptographic computation load. Packets can include non cryptographic consistency checks. For example, [RFC5082] provides a mechanism that uses the IP header to limit the attackers that can inject packets that will be subject to cryptographic validation. In the design phase II, once an automated key management protocol is developed, it may be possible to determine the peer IP addresses that are valid participants. Only the packets from the verified sources could be subject to cryptographic validation.

Protocol designers must ensure that device never needs to check incoming protocol packets using multiple keys, as this can overwhelm the CPU, leading to a DoS attack. KARP solutions should indicate the checks that are appropriate prior to performing cryptographic validation. KARP solutions should indicate where information about valid neighbors can be used to limit the scope of the attacks.

Particular care needs to be paid to design of automated key management schemes. It is often desirable to force a party attempting to authenticate to do work and to maintain state until that work is done. That is, the initiator of the authentication should maintain the cost of any state required by the authentication for as long as possible. This also helps

when an attacker send an overwhelming load of keying protocol initiations from bogus sources.

Another important class of attack is denial of service against the routing protocol where an attacker can manipulate either the routing protocol or cryptographic authentication mechanism to disrupt routing adjacencies.

Without KARP solutions, many routing protocols are subject to disruption simply by injecting an invalid packet or a packet for the wrong state. Even with cryptographic validation, replay attacks are often a vector where a previously valid packet can be injected to create a denial of service. KARP solutions should prevent all cases where packet replays or other packet injections by an outsider can disrupt routing sessions.

Some residual denial of service risk is always likely. If an attacker can generate a large enough number of packets, the routing protocol can get disrupted. Even if the routing protocol is not disrupted, the loss rate on a link may rise to a point where claiming that traffic can successfully be routed across the link will be inaccurate.

8. Gap Analysis

The [I-D.ietf-karp-threats-reqs] document lists the generic requirements for the security mechanisms that must exist for the various routing protocols that come under the purview of KARP. There will be different design teams working for each of the categories of routing protocols defined.

To start, design teams must review the "Threats and Requirements for Authentication of Routing Protocols" document [I-D.ietf-karp-threats-reqs]. This document contains detailed descriptions of the threat analysis for routing protocol authentication and integrity in general. Note that it will not contain all the authentication-related threats for any one routing protocol, or category of routing protocols. The design team must conduct a protocol-specific threat analysis to determine if threats beyond those in the [I-D.ietf-karp-threats-reqs] document arise in the context of the protocol (group), and to describe those threats.

The [I-D.ietf-karp-threats-reqs] document also contains many security requirements. Each routing protocol design team must walk through each section of the requirements and determine one by one how its protocol either does or does not relate to each requirement.

Examples include modern, strong cryptographic algorithms, with at least one such algorithm listed as a MUST; algorithm agility; secure use of simple PSKs; intra-connection replay protection; inter-connection replay protection, etc.

When doing the gap analysis we must first identify the elements of each routing protocol that we wish to protect. In case of protocols riding on top of IP, we might want to protect the IP header and the protocol headers, while for those that work on top of TCP, it will be the TCP header and the protocol payload. There is patently value in protecting the IP header and the TCP header if the routing protocols rely on these headers for some information (for example, identifying the neighbor which originated the packet).

Then there will be a set of Cryptography requirements that we might want to look at. For example, there must be at least on set of cryptographic algorithms (MD5, SHA, etc.) or constructions (HMAC, etc.) whose use is supported by all implementations and can be safely assumed to be supported by any implementation of the authentication option. The design teams should look for this for the protocol that they are working on. If such algorithms or constructions are not available then some should be defined to support interoperability by having a single default.

Design teams must ensure that the default cryptographic algorithms and constructions supported by the routing protocols are accepted by the community. This means that the protocols must not rely on non-standard or ad-hoc hash functions, keyed-hash constructions, signature schemes, or other functions, and must use published and standard schemes.

Care should also be taken to ensure that the routing protocol authentication scheme has algorithm agility (i.e., it is capable of supporting algorithms other than its defaults).

Ideally, the authentication mechanism should not be affected by packet loss and reordering.

Design teams should ensure that their protocols authentication mechanism is able to accommodate rekeying. This is essential since it is well known that keys must periodically be changed. Also what the designers must ensure is that this rekeying event should not affect the functioning of the routing protocol. For example, OSPF rekeying requires coordination among the adjacent routers, while IS-IS requires coordination among routers in the entire domain.

If new authentication and security mechanisms are needed then the design teams must design in such a manner that the routing protocol authentication mechanism remains oblivious to how the keying material is derived. This decouples the authentication mechanism from the key management system that is employed.

Design teams should also note that many routing protocols require prioritized treatment of certain protocol packets and authentication mechanisms should honor this.

Not all routing protocol authentication mechanisms provide support for replay attacks, and the design teams should identify such authentication mechanisms and work on them so that this can get fixed. The design teams must look at the protocols that they are working on and see if packets captured from the previous/stale sessions can be replayed.

What might also influence the design is the rate at which the protocol packets are originated. In case of protocols like BFD, where packets are originated at millisecond intervals, there are some special considerations that must be kept in mind when defining the new authentication and security mechanisms.

The designers should also consider whether the current authentication mechanisms impose considerable processing overhead on a router that’s doing authentication. Most currently deployed routers do not have hardware accelerators for cryptographic processing and these operations can impose a significant processing burden under some circumstances. The proposed solutions should be evaluated carefully with regard to the processing burden that they will impose, since deployment may be impeded if network operators perceive that a solution will impose a processing burden which either entails substantial capital expenses or threatens to destabilize the routers.

9. Security Considerations

As mentioned in the Introduction, RFC4948 [RFC4948] identifies additional steps needed to achieve the overall goal of improving the security of the core routing infrastructure. Those include validation of route origin announcements, path validation, cleaning up the IRR databases for accuracy, and operational security practices that prevent routers from becoming compromised devices. The KARP work is but one step needed to improve core routing infrastructure.

The security of cryptographic-based systems depends on both the strength of the cryptographic algorithms chosen and the strength of the keys used with those algorithms. The security also depends on the engineering of the protocol used by the system to ensure that there are no non-cryptographic ways to bypass the security of the overall system.

9.1. Use Strong Keys

Care should be taken to ensure that the selected key is unpredictable, avoiding any keys known to be weak for the algorithm in use. [RFC4086] contains helpful information on both key generation techniques and cryptographic randomness.

Care should also be taken when choosing the length of the key. [RFC3766] provides some additional information on asymmetric and symmetric key sizes and how they related to system requirements for attack resistance.

In addition to using a key of appropriate length and randomness, deployers of KARP protocols should use different keys between different routing peers whenever operationally possible. This is especially true when the Routing Protocol takes a static Traffic Key as opposed to a Traffic Key derived on a per-connection basis using a KDF. The burden for doing so is understandably much higher than for using the same static Traffic Key across all peering routers. Depending upon the specific KMP it can be argued that generally using a KMP network-wide increases peer-wise security. Consider an attacker that learns or guesses the Traffic Key used by two peer-routers: if the Traffic key is only used between those two routers, then the attacker has only compromised that one connection not the entire network.

However, whenever using manual keys, it is best to design a system where a given pre-shared key (PSK) will be used in a KDF, mixed with connection-specific material, in order to generate session unique -- and therefore peer-wise -- Traffic Keys. Doing so has the following advantages: the Traffic Keys used in the per-message authentication mechanism are peer-wise unique, it provides inter-connection replay protection, and, if the per-message authentication mechanism covers some connection counter, intra-connection replay protection.

Note that certain key derivation functions (e.g. KDF_AES_128_CMAC, as used in TCP-AO [RFC5926], the pseudorandom function (PRF) used in the KDF may require a key of a certain fixed size as an input.

For example, AES_128_CMAC requires a 128 bit (16 byte) key as the seed. However, for convenience to the administrators, a specification may not want to require the entry of a PSK of exactly 16 bytes. Instead, a specification may call for a key prep routine that could handle a variable length PSK, one that might be less or more than 16 bytes (see [RFC4615], section 3, as an example). That key prep routine would derive a key of exactly the required length and thus suitable as a seed to the PRF. This does NOT mean that administrators are safe to use weak keys. Administrators are encouraged to follow [RFC4086] [NIST-800-118]. We simply attempted to "put a fence around stupidity", as much as possible as its hard to imagine administrators putting in a password that is, say 16 bytes in length.

A better option, from a security perspective, is to use some representation of a device-specific asymmetric key pair as the identity proof, as described in section "Unique versus Shared Keys" section.

9.2. Internal vs. External Operation

Design teams must consider whether the protocol is an internal routing protocol or an external one, i.e. does it primarily run between peers within a single domain of control or between two different domains of control? Some protocols may be used in both cases, internally and externally, and as such various modes of authentication operation may be required for the same protocol. While it is preferred that all routing exchanges run with the best security mechanisms enabled in all deployment contexts, this exhortation is greater for those protocols running on inter-domain point-to-point links, and greatest for those on shared access link layers with several different domains interchanging together, because the volume of attackers are greater from the outside. Note however that the consequences of internal attacks maybe no less severe -- in fact they may be quite a bit more severe -- than an external attack. An example of this internal versus external consideration is BGP which has both EBGP and IBGP modes. Another example is a multicast protocol where the neighbors are sometimes within a domain of control and sometimes at an inter-domain exchange point. In the case of PIM-SM running on an internal multi-access link, it would be acceptable to give up some security to get some convenience by using a group key among the peers on the link. On the other hand, in the case of PIM-SM running over a multi-access link at a public exchange point, operators may favor security over convenience by using unique pair-wise keys for every peer. Designers must consider

both modes of operation and ensure the authentication mechanisms fit both.

Operators are encouraged to run cryptographic authentication on all their adjacencies, but to work from the outside in, i.e. EBGP links are a higher priority than the IBGP links because they are externally facing, and, as a result, more likely to be targeted in an attack.

9.3. Unique versus Shared Keys

This section discusses security considerations regarding when it is appropriate to use the same authentication key inputs for multiple peers and when it is not. This is largely a debate of convenience versus security. It is often the case that the best secured mechanism is also the least convenient mechanism. For example, an air gap between a host and the network absolutely prevents remote attacks on the host, but having to copy and carry files using the "sneaker net" is quite inconvenient and does not scale.

Operators have erred on the side of convenience when it comes to securing routing protocols with cryptographic authentication. Many do not use it at all. Some use it only on external links, but not on internal links. Those that do use it often use the same key for all peers in a network. It is common to see the same key in use for years, e.g., the key was entered when authentication mechanisms were originally configured, or the routing gear was deployed.

One goal for designers is to create authentication and integrity mechanisms that are easy for operators to deploy and manage, and still use unique keys between peers (or small groups on multi-access links), and for different sessions among the same peers. Operators have the impression that they NEED one key shared across the network, when in fact they do not. What they need is the relative convenience they experience from deploying cryptographic authentication with one key (or a few keys), compared to the inconvenience they would experience if they deployed the same authentication mechanism using unique pairwise keys. An example is BGP Route Reflectors. Here operators often use the same authentication key between each client and the route reflector. The roadmaps defined from this guidance document should allow for unique keys to be used between each client and the peer, without sacrificing much convenience. Designers should strive to deliver peer-wise unique keying mechanisms with similar ease-of-deployment properties as today’s one-key method.

Operators must understand the consequences of using the same key across many peers. Unique keys are more secure than shared keys because they reduce both the attack target size and the attack consequence size. In this context, the attack target size represents the number of unique routing exchanges across a network that an attacker may be able to observe in order to gain security association credentials, i.e. crack the keys. If a shared key is used across the entire internal domain of control, then the attack target size is very large. The larger the attack target, the easier it is for the attacker to gain access to analysis data, and greater the volume of analysis data he can access in a given time frame, both of which make the job easier. Using the same key across the network makes the attack vulnerability surface more penetrable than unique keys.

The above attack can be mitigated to a certain extent by using strong keys. Another argument against using the same key is that if the same key that is used in multiple devices then a compromise of any one of the devices will expose the key. Also since the same key is supported on many devices this is known by many people which affects its distribution to all of the devices.

Consider also the attack consequence size, the amount of routing adjacencies that can be negatively affected once a breach has occurred, i.e., once the keys have been acquired by the attacker.

Again, if a shared key is used across the internal domain, then the consequence size is the whole network. Ideally, unique key pairs would be used for each adjacency.

In some cases use of shared keys is needed because of the problem space. For example, a multicast packet is sent once but then consumed by several routing neighbors. If unique keys were used per neighbor, the benefit of multicast would be erased because sender would have to create a different announcement packet for each receiver. Though this may be desired and acceptable in some small number of use cases, it is not the norm. Shared (i.e., group) keys are an acceptable solution here, and much work has been done already in this area (see MSEC working group).

9.4. Key Exchange Mechanism

This section discusses the security and use case considerations for key exchange for routing protocols. Two options exist: an out-of-band mechanism or a KMP. An out-of-band mechanism involves operators configuring keys in the device through a

configuration tool or management method (e.g., SNMP, NETCONF). A KMP is an automated protocol that exchanges key without operator intervention. KMPs can occur either in-band to the routing protocol or out-of-band to the routing protocol (i.e., a different protocol).

An example of an out-of-band configuration mechanism could be an administrator who makes a remote management connection (e.g. using SSH) to a router and manually enters the keying information, e.g., the algorithm, the key(s), the key lifetimes, etc. Another example could be an OSS system that inputs the same information using a script over an SSH connection, or by pushing configuration through some other management connection, standard (Netconf-based) or proprietary.

The drawbacks of an out-of-band configuration mechanism include: lack of scaleability, complexity, and speed of changing if a security breach is suspected. For example, if an employee who had access to keys was terminated, or if a machine holding those keys was believed to be compromised, then the system would be considered insecure and vulnerable until new keys were generated and distributed. Those keys then need to be placed into the OSS system, and the OSS system then needs to push the new keys -- often during a very limited change window -- into the relevant devices. If there are multiple organizations involved in these connections, because the protected connections are inter-domain, this process is very complicated.

The principle benefit of out-of-band configuration mechanism is that once the new keys/parameters are set in OSS system, they can be pushed automatically to all devices within the OSS's domain. Operators have mechanisms in place for this already for managing other router configuration data. In small environments with few routers, a manual system is not difficult to employ.

We further define a peer-to-peer KMP as using cryptographically protected identity verification, session key negotiation, and security association parameter negotiation between the two routing peers. The KMP among peers may also include the negotiation of parameters, like cryptographic algorithms, cryptographic inputs (e.g. initialization vectors), key life-times, etc.

There are several benefits of a peer-to-peer KMP versus centrally managed and distributing keys. It results in key(s) that are privately generated, and need not be recorded permanently anywhere. Since the traffic keys used in a particular connection are not a fixed part of a device

configuration no security sensitive 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 limited or no access to current traffic keys. They may gain access to key derivation material, like a PSK, but may not be able to access the current traffic keys in use. In this example, these PSKs can be updated in the device configurations (either manually or through an OSS) without bouncing or impacting the existing session at all. In the case of using raw asymmetric keys or certificates, instead of PSKs, the data theft (from the data store) would likely not result in any compromise, as the key pairs would have been generated on the routers, and never leave those routers. In such a case no changes are needed on the routers; the connections will continue to be secure, uncompromised. Additionally, with a KMP regular rekey operations occur without any operator involvement or oversight. This keeps keys fresh.

There are a few drawbacks to using a KMP. First, a KMP requires more cryptographic processing for the router at the beginning of a connection. This will add some minor start-up time to connection establishment versus a purely manual key management approach. Once a connection with traffic keys has been established via a KMP, the performance is the same in the KMP and the out-of-band configuration case. KMPs also add another layer of protocol and configuration complexity which can fail or be misconfigured. This was more of an issue when these KMPs were first deployed, but less so as these implementations and operational experience with them has matured.

One of the goals for KARP is to develop a KMP; an out-of-band configuration protocol for key exchange is out of scope.

Within this constraint there are two approaches for a KMP:

The first, is to use a KMP that runs independent of the routing and the signaling protocols. It would run on its own port and use its own transport (to avoid interfering with the routing protocol that it is serving). When a routing protocol needs a key, it would contact the local instance of this key management protocol and request a key. The KMP generates a key that is delivered to the routing protocol for it to use for authenticating and integrity verification of the routing protocol packets. This KMP could either be an existing key management protocol like ISAKMP/IKE, GKMP, etc., extended for the routing protocols, or it could be a new KMP, designed for the routing protocol context.

The second approach is to define an In-band KMP extension for existing routing protocols putting the key management mechanisms inside the protocol itself. In this case, the key management messages would be carried within the routing protocol packets, resulting in very tight coupling between the routing protocols and the key management protocol.

10. Acknowledgments

Much of the text for this document came originally from draft-lebovitz-karp-roadmap, authored by Gregory M. Lebovitz.

We would like to thank Sam Hartman, Eric Rescorla, Russ White, Sean Turner, Stephen Kent, Stephen Farrell, Adrian Farrel, Russ Housley, Michael Barnes and Vishwas Manral for their comments on the draft.

11. IANA Considerations

This document places no requests to IANA.

12. References

12.1. Normative References

[RFC2119] Bradner, S.,"Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4948] Andersson, L., et. al, "Report from the IAB workshop on Unwanted Traffic March 9-10, 2006", RFC 4948, August 2007.

12.2. Informative References

[RFC1195] Callon, R. , "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", RFC 1195, December 1990.

[RFC2205] Braden, R., et. al, "Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification", RFC 2205, September 1197.

[RFC2328] Moy, J., "OSPF Version 2", RFC 2328, April 1998.

[RFC2453] Malkin, G., "RIP Version 2", RFC 2453, November 1998.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.

[RFC3097] Braden, R, and Zhang, L., "RSVP Cryptographic Authentication -- Updated Message Type Value", RFC 3097, April 2001

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3618] Fenner, B. and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003.

[RFC3766] Orman, H. and Hoffman, P., "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys", RFC 3766, April 2004.

[RFC3973] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, June 2005.

[RFC4230] Tschofenig, H. and R. Graveman, "RSVP Security Properties", RFC 4230, December 2005.

[RFC4252] Ylonen, T., et. al, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.

[RFC4253] Ylonen, T., et. al, "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006

[RFC4271] Rekhter, Y., Li, T. and Hares, S.,"A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4492] Blake-Wilson, S., "Elliptical Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, May 2006

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas,"Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[RFC4615] Song, J., Poovendran, R., Lee, J., and T. Iwata, "The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4615, August 2006.

[RFC4726] Farrel, A., et. al.,"A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, November 2006.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

[RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P. and Pignataro, C.,"The Generalized TTL Security Mechanism (GTSM)" , RFC 5082, October 2007

[RFC5151] Farrel, A., et. al.,"Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 5151, February 2008.

[RFC5280] Cooper, D., et. al.,"Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008

[RFC5440] Vasseur, J.P. and Le Roux, J.L., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009

[RFC5796] Atwood, W., Islam, S., and M. Siami, "Authentication and Confidentiality in PIM-SM Link-local Messages", RFC 5796, March 2010.

[RFC5880] Katz, D. and Ward, D., "Bidirectional Forwarding Detection", RFC 5880, June 2010.

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

[RFC5926] Lebovitz, G., "Cryptographic Algorithms, Use and Implementation Requirements for TCP Authentication Option", RFC 5926, June 2010.

[RFC6039] Manral, V., Bhatia, M., Jaeggli, J. and White, R.,"Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010

[I-D.ietf-karp-threats-reqs] Lebovitz, G., "KARP Threats and Requirements", Work in Progress, October 2010.

[I-D.ietf-karp-crypto-key-table] Housley, R. and Polk, T., "Database of Long-Lived Symmetric Cryptographic Keys" , Work in Progress, May 2011

[I-D.weis-gdoi-mac-tek] Weis, B. and S. Rowles, "GDOI Generic Message Authentication Code Policy", Work in Progress, June 2010.

[IRR] Merit Network Inc , "Internet Routing Registry Routing Assets Database", 2006, http://www.irr.net/.

[NIST-800-57] US National Institute of Standards & Technology, "Recommendation for Key Management Part 1: General (Revised)", March 2007

[NIST-800-118] US National Institute of Standards & Technology, "Guide to Enterprise Password Management (Draft)", April 2009

Author’s Addresses

Gregory M. Lebovitz
California
USA 95003

Phone:
Email: gregory.ietf@gmail.com

Manav Bhatia
Alcatel-Lucent
Bangalore
India

Phone:
Email: manav.bhatia@alcatel-lucent.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 25, 2012

S. Hartman
Painless Security
D. Zhang
Huawei

October 23, 2011

Operations Model for Router Keying
draft-ietf-karp-ops-model-01.txt

Abstract

Developing an operational and management model for routing protocol security that works across protocols will be critical to the success of routing protocol security efforts. This document discusses issues and begins to consider development of these models.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on April 25, 2012.

Copyright Notice

Table of Contents

1. Introduction

The KARP working group is designing improvements to the cryptographic authentication of IETF routing protocols. These improvements include improvements to how integrity functions are handled within each protocol as well as designing an automated key management solution.

This document discusses issues to consider when thinking about the operational and management model for KARP. Each implementation will take its own approach to management; this is one area for vendor differentiation. However, it is desirable to have a common baseline for the management objects allowing administrators, security architects and protocol designers to understand what management capabilities they can depend on in heterogeneous environments. Similarly, designing and deploying the protocol will be easier with thought paid to a common operational model. This will also help with the design of NetConf schemas or MIBs later.

2. Requirements notation

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].

3. Breakdown of KARP configuration

There are multiple ways of structuring configuration information. One factor to consider is the scope of the configuration information. Several protocols are peer-to-peer routing protocols where a different key could potentially be used for each neighbor. Other protocols require the same group key to be used for all nodes in an administrative domain or routing area. In other cases, the same group key needs to be used for all routers on an interface, but different group keys can be used for each interface.

Within situations where a per-interface, per-area or per-peer key can be used for manually configured long-term keys, that flexibility may not be desirable from an operational standpoint. For example consider OSPF [RFC2328]. Each OSPF link needs to use the same authentication configuration, including the set of keys used for reception and the set of keys used for transmission, but may use different keys for different links. The most general management model would be to configure keys per link. However for deployments where the area uses the same key it would be strongly desirable to configure the key as a property of the area. If the keys are configured per-link, they can get out of sync. In order to support generality of configuration and common operational situations, it would be desirable to have some sort of inheritance where default configurations are made per-area unless overridden per-interface.

As described in [I-D.housley-saag-crypto-key-table], the cryptographic keys are separated from the interface configuration into their own configuration store. This document should specify how key selection interacts with the key table. One possible approach would be to assume that all keys that permit use on a given interface would be used on that interface with no additional configuration steps. If this model is adopted then the key table draft should be expanded to permit specification of domains and areas as well. It’s not clear why "all" is permitted as an interface specification in this model; it seems unlikely that it would be desirable to use the same set of keys for two different instances of an IGP or across autonomous system boundaries.

Another model is that the interface specification in the key table is a restriction that limits keys on top of other configuration enabling them. Then a set of keys from the key table is attached to an interface, area or routing domain using an additional configuration step. This avoids the previous problems at the expense of significant complexity of configuration.

Operational Requirements: KARP MUST support configuration of keys at the most general scope for the underlying protocol; protocols

supporting per-peer keys MUST permit configuration of per-peer keys, protocols supporting per-interface keys MUST support configuration of per-interface keys, and so on. KARP MUST NOT permit configuration of an inappropriate key scope. For example, configuration of separate keys per interface MUST NOT be supported for a protocol requiring per-area keys.

3.1. Integrity of the Key Table

The routing key table [I-D.housley-saag-crypto-key-table] provides a very general mechanism to abstract the storage of keys for routing protocols. To avoid misconfiguration and simplify problem determination, the router MUST verify the internal consistency of entries added to the table. At a minimum, the router MUST verify:

- The cryptographic algorithms are valid for the protocol.
- The key derivation function is valid for the protocol.
- The direction is valid for the protocol; for example protocols that require the same session key be used in both directions MUST have a direction of both.
- The peer and interface specification is consistent with the protocol.

Other checks are possible. For example the router could verify that if a key is associated with a peer, that peer is a configured peer for the specified protocol. However, this may be undesirable. It may be desirable to load a key table when some peers have not yet been configured. Also, it may be desirable to share portions of a key table across devices even when their current configuration does not require an adjacency with a particular peer in the interest of uniform configuration or preparing for fail-over.

3.2. Management of Key Table

Several management operations will be quite common. For service provider deployments the configuration management system can simply update the key table. However, for smaller deployments, efficient management operations are important.

As part of adding a new key it is typically desirable to set an expiration time for an old key. The management interface SHOULD provide a mechanism to easily update the expiration time for a current key used with a given peer or interface. Also when adding a key it is desirable to push the key out to nodes that will need it, allowing use for receiving packets then later enabling transmit.

This can be accomplished automatically by providing a delay between when a key becomes valid for reception and transmission. However, some environments may not be able to predict when all the necessary changes will be made. In these cases having a mechanism to enable a key for sending is desirable.

3.3. Protocol Limitations from the Key Table

The format of the key table imposes a few limitations on routing protocols. The first is that the key ID is 16 bits; some routing protocols have 32-bit key identifiers. A key mapping table as discussed in 4.1 of [I-D.polk-saag-rtg-auth-keytable] could be used to map to the larger key identifier. However it's probably desirable to either decide that only 16 bits of the key ID space is to be used or to expand the identifier space in the key table. From a management standpoint we need to make concrete requirements around whether a key ID is per-protocol or whether subspaces in the key ID space are reserved for each protocol. This is necessary so that implementations from different vendors can be managed consistently.

The second requirement that the key table places is that the key ID is scoped fairly broadly. At least within some protocols such as OSPF, the key ID might only need to be unique per-link or per-peer. That is, packets sent on two different interfaces could use key ID 32 even if the keys were different for these interfaces. An implementation could use the interface and the key ID as a lookup to find the right key. However, the key table draft requires that a key ID be sufficient to look up a key, meaning that the key ID is a globally scoped identifier. There is nothing wrong with this restriction, but it does need to be noted when assigning key IDs for a domain.

Consideration is required for how an automated key management protocol will assign key IDs for group keys. All members of the group may need to use the same key ID. This requires careful coordination of global key IDs. Interactions with the peer key ID field may make this easier; this requires additional study.

Automated key management protocols also assign keys for single peers. If the key ID is global and needs to be coordinated between the receiver and transmitter, then there is complexity in key management protocols.

3.4. VRFs

Many core and enterprise routers support multiple routing instances. For example a router serving multiple VPNs is likely to have a forwarding/routing instance for each of these VPNs. We need to

decide how the key table and other configuration information for KARP interacts with this. The obvious first-order answer is that each routing instance gets its own key table. However, we need to consider how these instances interact with each other and confirm this makes sense.

4. Credentials and Authorization

Several methods for authentication have been proposed for KARP. The simplest is preshared keys used directly as traffic keys. In this mode, the traffic integrity keys are directly configured. This is the mode supported by today's routing protocols.

As discussed in [I-D.polk-saag-rtg-auth-keytable], preshared keys can be used as the input to a key derivation function (KDF) to generate traffic keys. For example the TCP Authentication Option (TCP-AO) [RFC5925] derives keys based on the initial TCP session state. Typically a KDF will combine a long-term key with public inputs exchanged as part of the protocol to form fresh session keys. a KDF could potentially be used with some inputs that are configured along with the long-term key. Also, it's possible that inputs to a KDF will be private and exchanged as part of the protocol, although this will be uncommon in KARP's uses of KDFs.

Preshared keys could also be used by an automated key management protocol. In this mode, preshared keys would be used for authentication. However traffic keys would be generated by some key agreement mechanism or transported in a key encryption key derived from the preshared key. This mode may provide better replay protection. Also, in the absence of active attackers, key agreement strategies such as Diffie-Hellman can be used to produce high-quality traffic keys even from relatively weak preshared keys.

Public keys can be used for authentication. The design guide [I-D.ietf-karp-design-guide] describes a mode in which routers have the hashes of peer routers' public keys. In this mode, a traditional public-key infrastructure is not required. The advantage of this mode is that a router only contains its own keying material, limiting the scope of a compromise. The disadvantage is that when a router is added or deleted from the set of authorized routers, all routers that peer need to be updated. Note that self-signed certificates are a common way of communicating public-keys in this style of authentication.

Certificates signed by a certification authority or some other PKI could be used. The advantage of this approach is that routers may not need to be directly updated when peers are added or removed. The disadvantage is that more complexity and cost is required.

Each of these approaches has a different set of management and operational requirements. Key differences include how authorization is handled and how identity works. This section discusses these differences.

4.1. Preshared Keys

In the protocol, manual preshared keys are either unnamed or named by a small integer (typically 16 or 32 bits) key ID. Implementations that support multiple keys for protocols that have no names for keys need to try all possible keys before deciding a packet cannot be validated [RFC4808]. Typically key IDs are names used by one group or peer.

Manual preshared keys are often known by a group of peers rather than just oneother peer. This is an interesting security property: unlike with digitally signed messages or protocols where symmetric keys are known only to two parties, it is impossible to identify the peer sending a message cryptographically. However, it is possible to show that the sender of a message is one of the parties who knows the preshared key. Within the routing threat model the peer sending a message can be identified only because peers are trusted and thus can be assumed to correctly label the packets they send. This contrasts with a protocol where cryptographic means such as digital signatures are used to verify the origin of a message. As a consequence, authorization is typically based on knowing the preshared key rather than on being a particular peer. Note that once an authorization decision is made, the peer can assert its identity; this identity is trusted just as the routing information from the peer is trusted. Doing an additional check for authorization based on the identity included in the packet would provide little value: an attacker who somehow had the key could claim the identity of an authorized peer and an attacker without the key should be unable to claim the identity of any peer. Such a check is not required by the KARP threat model: inside attacks are not in scope.

Preshared keys used with key derivation function similarly to manual preshared keys. However to form the actual traffic keys, session or peer specific information is combined with the key. From an authorization standpoint, the derivation key works the same as a manual key. An additional routing protocol step or transport step forms the key that is actually used.

Preshared keys that are used via automatic key management have not been specified for KARP. Their naming and authorization may differ from existing uses of preshared keys in routing protocols. In particular, such keys may end up being known only by two peers. Alternatively they may also be known by a group of peers. Authorization could potentially be based on peer identity, although it is likely that knowing the right key will be sufficient. There does not appear to be a compelling reason to decouple the authorization of a key for some purpose from authorization of peers holding that key to perform the authorized function.

Care needs to be taken when symmetric keys are used for multiple purposes. Consider the implications of using the same preshared key for two interfaces: it becomes impossible to cryptographically distinguish a router on one interface from a router on another interface. So, a router that is trusted to participate in a routing protocol on one interface becomes implicitly trusted for the other interfaces that share the key. For many cases, such as link-state routers in the same routing area, there is no significant advantage that an attacker could gain from this trust within the KARP threat model. However, distance-vector protocols, such as BGP and RIP, permit routes to be filtered across a trust boundary. For these protocols, participation in one interface might be more advantageous than another. Operationally, when this trust distinction is important to a deployment, different keys need to be used on each side of the trust boundary. Key derivation can help prevent this problem in cases of accidental misconfiguration. However, key derivation cannot protect against a situation where a system was incorrectly trusted to have the key used to perform the derivation. To the extent that there are multiple zones of trust and a routing protocol is determining whether a particular router is within a certain zone, the question of untrusted actors is within the scope of the routing threat model.

Key derivation can be part of a management solution to a desire to have multiple keys for different zones of trust. A master key could be combined with peer, link or area identifiers to form a router-specific preshared key that is loaded onto routers. Provided that the master key lives only on the management server and not the individual routers, trust is preserved. However in many cases, generating independent keys for the routers and storing the result is more practical. If the master key were somehow compromised, all the resulting keys would need to be changed. However if independent keys are used, the scope of a compromise may be more limited.

More subtle problems with key separation can appear in protocol design. Two protocols that use the same traffic keys may work together in unintended ways permitting one protocol to be used to attack the other. Consider two hypothetical protocols. Protocol A starts its messages with a set of extensions that are ignored if not understood. Protocol B has a fixed header at the beginning of its messages but ends messages with extension information. It may be that the same message is valid both as part of protocol A and protocol B. An attacker may be able to gain an advantage by getting a router to generate this message with one protocol under situations where the other protocol would not generate the message. This hypothetical example is overly simplistic; real-world attacks exploiting key separation weaknesses tend to be complicated and involve specific properties of the cryptographic functions involved.

The key point is that whenever the same key is used in multiple protocols, attacks may be possible. All the involved protocols need to be analyzed to understand the scope of potential attacks.

Key separation attacks interact with the KARP operational model in a number of ways. Administrators need to be aware of situations where using the same manual traffic key with two different protocols (or the same protocol in different contexts) creates attack opportunities. Design teams should consider how their protocol might interact with other routing protocols and describe any attacks discovered so that administrators can understand the operational implications. When designing automated key management or new cryptographic authentication within routing protocols, we need to be aware that administrators expect to be able to use the same preshared keys in multiple contexts. As a result, we should use appropriate key derivation functions so that different cryptographic keys are used even when the same initial input key is used.

4.2. Asymmetric Keys

Outside of a PKI, public keys are expected to be known by the hash of a key or (potentially self-signed) certificate. The Session Description Protocol provides a standardized mechanism for naming keys (in that case certificates) based on hashes (section 5 [RFC4572]). KARP SHOULD adopt this approach or another approach already standardized within the IETF rather than inventing a new mechanism for naming public keys.

A public key is typically expected to belong to one peer. As a peer generates new keys and retires old keys, its public key may change. For this reason, from a management standpoint, peers should be thought of as associated with multiple public keys rather than as containing a single public key hash as an attribute of the peer object.

Authorization of public keys could be done either by key hash or by peer identity. Performing authorizations by peer identity should make it easier to update the key of a peer without risk of losing authorizations for that peer. However management interfaces need to be carefully designed to avoid making this extra level of indirection complicated for operators.

4.3. Public Key Infrastructure

When a PKI is used, certificates are used. The certificate binds a key to a name of a peer. The key management protocol is responsible for exchanging certificates and validating them to a trust anchor.

Authorization needs to be done in terms of peer identities not in terms of keys. One reason for this is that when a peer changes its key, the new certificate needs to be sufficient for authentication to continue functioning even though the key has never been seen before.

Potentially authorization could be performed in terms of groups of peers rather than single peers. An advantage of this is that it may be possible to add a new router with no authentication related configuration of the peers of that router. For example, a domain could decide that any router with a particular keyPurposeID signed by the organization's certificate authority is permitted to join the IGP. Just as in configurations where cryptographic authentication is not used, automatic discovery of this router can establish appropriate adjacencies.

Assuming that potentially self-signed certificates are used by routers that wish to use public keys but that do not need a PKI, then PKI and the infrastructureless mode of public-key operation described in the previous section can work well together. One router could identify its peers based on names and use certificate validation. Another router could use hashes of certificates. This could be very useful for border routers between two organizations. Smaller organizations could use public keys and larger organizations could use PKI.

4.4. The role of Central Servers

An area to explore is the role of central servers like RADIUS or directories. As discussed in the design-guide, a system where keys are pushed by a central management system is undesirable as an end result for KARP. However central servers may play a role in authorization and key rollover. For example a node could send a hash of a public key to a RADIUS server.

If central servers do play a role it will be critical to make sure that they are not required during routine operation or a cold-start of a network. They are more likely to play a role in enrollment of new peers or key migration/compromise.

Another area where central servers may play a role is for group key agreement. As an example, [I-D.liu-ospfv3-automated-keying-req] discusses the potential need for key agreement servers in OSPF. Other routing protocols that use multicast or broadcast such as IS-IS are likely to need a similar approach.

5. Grouping Peers Together

One significant management consideration will be the grouping of management objects necessary to determine who is authorized to act as a peer for a given routing action. As discussed previously, the following objects are potentially required:

- Key objects are required. Symmetric keys may be preshared. Asymmetric public keys may be used directly for authorization as well. During key transitions more than one key may refer to a given peer. Group preshared keys may refer to multiple peers.

- A peer is a router that this router might wish to communicate with. Peers may be identified by names or keys.

- Groups of peers may be authorized for a given routing protocol.

Establishing a management model is difficult because of the complex relationships between each set of objects. As discussed there may be more than one key for a peer. However in the preshared key case, there may be more than one peer for a key. This is true both for group security association protocols such as an IGP or one-to-one protocols where the same key is used administratively. In some of these situations, it may be undesirable to explicitly enumerate the peers in the configuration; for example IGP peers are auto-discovered for broadcast links but not for non-broadcast multi-access links.

Peers may be identified either by name or key. If peers are identified by key it is probably strongly desirable from an operational standpoint to consider any peer identifiers or name to be a local matter and not require the names or identifiers to be synchronized. Obviously if peers are identified by names (for example with certificates in a PKI), identifiers need to be synchronized between the authorized peer and the peer making the authorization decision.

In many cases peers will explicitly be identified. In these cases it is possible to attach the authorization information (keys or identifiers) to the peer’s configuration object. Two cases do not involve enumerating peers. The first is the case where preshared keys are shared among a group of peers. It is likely that this case can be treated from a management standpoint as a single peer representing all the peers that share the keys. The other case is one where certificates in a PKI are used to introduce peers to a router. In this case, rather than configuring peers, , the router needs to be configured with information on what certificates represent acceptable peers.

Another consideration is what routing protocols share peers. For example it may be common for LDP peers to also be peers of some other routing protocol. Also, RSVP-TE may be associated with some TE-based IGP. In some of these cases it would be desirable to use the same authorization information for both routing protocols.

In order to develop a management model for authorization, the working group needs to consider several questions. What protocols support auto-discovery of peers? What protocols require more configuration of a peer than simply the peer's authorization information and network address? What management operations are going to be common as security information for peers is configured and updated? What operations will be common while performing key transitions or while migrating to new security technologies?

6. Administrator Involvement

One key operational question is what areas will administrator involvement be required. Likely areas where involvement may be useful includes enrollment of new peers. Fault recovery should also be considered.

6.1. Enrollment

One area where the management of routing security needs to be optimized is the deployment of a new router. In some cases a new router may be deployed on an existing network where routing to management servers is already available. In other cases, routers may be deployed as part of connecting or creating a site. Here, the router and infrastructure may not be available until the router has securely authenticated. This problem is similar to the problem of getting initial configuration of routing instances onto the router. However, especially in cases where asymmetric keys or per-peer preshared keys are used, the configuration of other routers needs to be modified to bring up the security association. Also, there has been discussion of generating keys on routers and not allowing them to leave devices. This also impacts what strategies are possible. For example this might mean that routers need to be booted in a secure environment where keys can be generated, and public keys copied to a management server to push out the new public key to potential peers. Then, the router needs to be packaged, moved to where it will be deployed and set up.Alternatives are possible; it is critical that we understand how what we propose impacts operators.

We need to work through examples with operators familiar with specific real-world deployment practices and understand how proposed security mechanisms will interact with these practices.

6.2. Handling Faults

Faults may interact with operational practice in at least two ways. First, security solutions may introduce faults. For example if certificates expire in a PKI, previous adjacencies may no longer form. Operational practice will require a way of repairing these errors. This may end up being very similar to deploying a router that is connecting a new site as the security fault may have partitioned the network. However, unlike a new deployment, the event is unplanned. Strategies such as configuring a router and shipping it to a site may not be appropriate for recovering a fault even though they may be more useful for new deployments.

Monitoring will play a critical role in avoiding security faults such as certificate expiration. However, the protocols MUST still have

adequate operational mechanisms to recover from these situations. Also, some faults, such as those resulting from a compromise or actual attack on a facility are inherent and may not be prevented.

A second class of faults is equipment faults that impact security. For example if keys are stored on a router and never moved from that device, failure of a router implies a need to update security provisioning on the replacement router and its peers.

To address these operational considerations, we should identify circumstances surrounding recovery from today’s faults and understand how protocols will impact mechanisms used today.

7. Upgrade Considerations

It needs to be possible to deploy automated key management in an organization without either having to disable existing security or disrupting routing. As a result, it needs to be possible to perform a phased upgrade from manual keying to automated key management.

For peer-to-peer protocols such as BGP, this is likely to be relatively easy. First, code that supports automated key management needs to be loaded on both peers. Then the adjacency can be upgraded. The configuration can be updated to switch to automated key management when the second router reboots.

The situation is more complicated for multicast protocols. It's probably not reasonable to bring down an entire link to reconfigure it as using automated key management. Two approaches should be considered. One is to support key table rows from the automated key management and manually configured for the same link at the same time. Coordinating this may be tricky. Another possibility is for the automated key management protocol to actually select the same traffic key that is being used manually

8. Related Work

Discuss draft-housley-saag-*, draft-polk-saag-*, the discussions in the KARP framework, etc.

9. Security Considerations

This document does not define a protocol. It does discuss the operational and management implications of several security technologies.

10. Acknowledgments

Funding for Sam Hartman’s work on this memo is provided by Huawei.

The authors would like to thank Gregory Lebovitz, Russ White and Bill Atwood for valuable reviews.

11. References

11.1. Normative References

[I-D.housley-saag-crypto-key-table]
Housley, R. and T. Polk, "Database of Long-Lived Symmetric Cryptographic Keys", draft-housley-saag-crypto-key-table-04 (work in progress), October 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[I-D.ietf-karp-design-guide]
Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", draft-ietf-karp-design-guide-07 (work in progress), October 2011.

[I-D.liu-ospfv3-automated-keying-req]
Liu, Y., "OSPFv3 Automated Group Keying Requirements", draft-liu-ospfv3-automated-keying-req-01 (work in progress), July 2007.

[I-D.polk-saag-rtg-auth-keytable]
Polk, T. and R. Housley, "Routing Authentication Using A Database of Long-Lived Cryptographic Keys", draft-polk-saag-rtg-auth-keytable-05 (work in progress), November 2010.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, July 2006.

[RFC4808] Bellovin, S., "Key Change Strategies for TCP-MD5", RFC 4808, March 2007.

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

Authors' Addresses

Sam Hartman
Painless Security

Email: hartmans-ietf@mit.edu

Dacheng Zhang
Huawei

Email: zhangdacheng@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2012

M. Jethanandani
B. Weis
K. Patel
Cisco Systems
D. Zhang
Huawei
S. Hartman
Painless Security

October 19, 2011

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

Abstract

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

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on April 21, 2012.

Copyright Notice

Table of Contents

1. Introduction

Existing routing protocols using unicast communication model (e.g., BGP, LDP, RSVP-TE) have cryptographic authentication mechanisms that use a key shared between the routers on the both sides of the model to protect routing message exchanges between the routers. Unicast key management today is limited to statically configuring master keys in individual routers. This document extends currently defined IKEv2 [RFC5996] protocol to define a Router Key Management Protocol (RKMP) that allows network devices to automatically exchange key material related information between the network devices.

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

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

1.1. Terminology

1.2. Acronyms and Abbreviations

The following acronyms and abbreviations are used throughout this document.

IKE Internet Key Exchange Protocol

IKEv2 Internet Key Exchange Protocol Version 2

SA Security Association

1.3. Relationship to IKEv2

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

Unlike IKE, IKEv2 is explicitly designed for IPsec. The document does not separate handling aspects of the protocol that would be needed for IPsec from those that apply to general key management. IPsec specific rules are combined with more general requirements.

While concepts and protocol payloads can be used in a different key management protocol, the current structure of IKEv2 does not provide a mechanism for applying IKEv2 to a domain of interpretation other than IPsec. In addition, the complexity required in the IKE specification when compared to IKEv2 suggests that the generality of IKE may not be worth the complexity cost.

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

2. Overview

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

2.1. Types of Keys

The keys adopted in RKMP are listed as follows:

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

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

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

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

3. Protocol Exchanges

The exchange of private keying material between two network devices using a dedicated key management protocol is a requirement as articulated in [I-D.ietf-karp-routing-tcp-analysis]. There is no need to define an entirely new protocol for this purpose, when existing mature protocol exchanges and methods have been vetted.

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

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

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

Acronyms Used in Protocol Exchange

3.1. RP_INIT

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

```
      Peer (Initiator)                  Peer (Responder)
      --------------------              ------------------
      HDR, SAi1, KEi, Ni        -->
                                <--     HDR, SAr1, KEr, Nr, [CERTREQ,]

                                  RP_INIT
```

3.2. RP_AUTH

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

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

                                  RP_AUTH
```

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

3.3. RP_ADD

The network devices may then destroy the state associated with the RP SA, continuing to use the RP policy and keying material, or they may choose to retain them for the further use. If both the network devices choose to retain them, they may use the RP SA to subsequently agree upon replacement policy for the same RP, or agree upon policy and keying material for another routing protocol. Either case will require the use of the RP Additional Exchange (RP_ADD), similar to the IKEv2 CREATE_CHILD_SA exchange as defined in RFC 5996.

A RP_ADD exchange therefore can be triggered in order to

1. Rekey an antique protocol master key and establish a new equivalent one

2. Generate needed key material for a newly executed routing protocol based on an existing SA

3. Rekey an RMKP_SA and establish a new equivalent RMKP_SA

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

RP_ADD

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

In the RP_ADD exchange, the SA payloads in the RP_ADD exchange are used identically as in the RP_AUTH exchange. For details on the rest of the exchange, refer to the CREATE_CHILD_SA exchange as defined in RFC 5996.

3.4. INFORMATIONAL

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

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

INFORMATIONAL

4. Header and Payload Formats

The protocol defined in this memo uses a HDR identical to the Generic Payload Header defined in section 3.2 of RFC 5996. The new exchanges defined in this memo are not used with IKEv2. A new IANA registry is

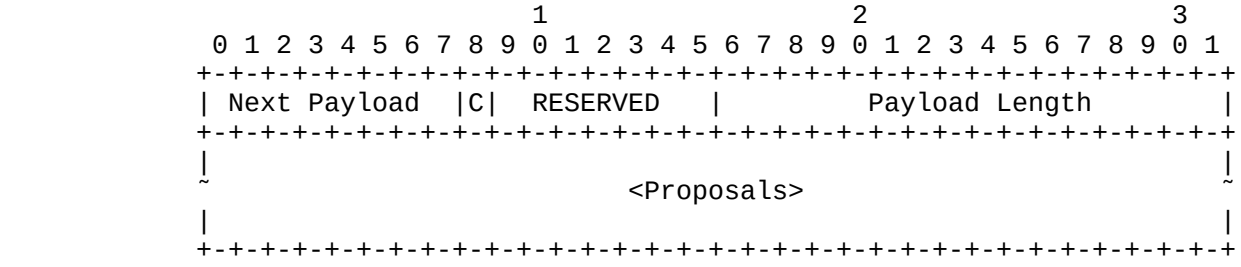
to be created to identify the RP exchange types and payloads described in this section.

4.1. Security Association Payload

The Security Association (SA) payload contains a list of Proposals, which describe one or more sets of policy that a router is willing to use to protect a routing protocol. It is identical to the SA payload described in RFC 5996, and the details of the fields are described there.

In the Initiator's message, the SArpi payload contains a list of Proposal payloads (as defined in the next section), each of which contains a single set of policy that can be applied to the packets described in the Traffic Selector (TS) payloads in the same exchange. For example, the TS payloads may describe a set of IP addresses and ports which are a BGP connection, and the SA payload contains a list of proposals describing what policy the router is willing to use to protect that BGP traffic. Each set of policy is given a particular "Proposal Number" uniquely identifying this set of policy.

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



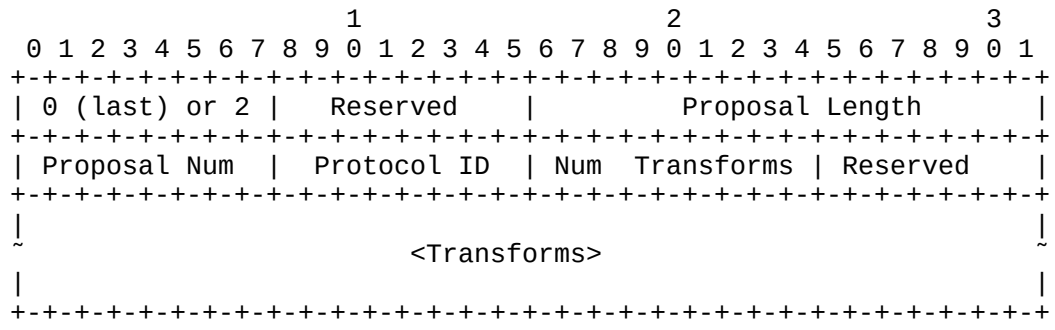
Security Association Payload

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

4.1.1. Proposal Substructure

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

cryptographic choices offered.



Proposal Payload

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

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

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

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

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

Protocol	Protocol ID	Reference
RKMP	1	
TCP-AO	2	RFC 5925
LDP Discovery Key	3	TBD
Standards Action	4-128	
Private Use	129-255	

Protocol ID

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

o Transforms (variable) - One or more transform substructures.

4.1.1.1. Transforms Substructures

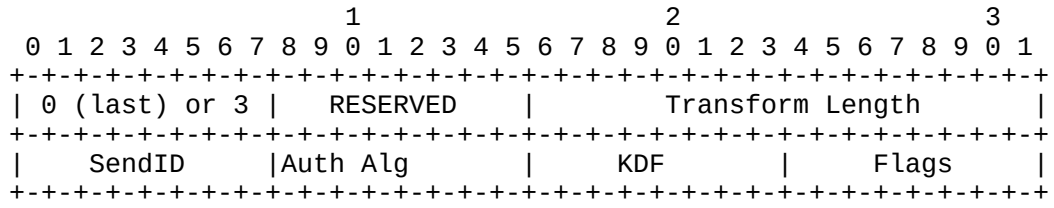
Each Proposal has a list of Transform (T) substructures, each of which describe a particular set of cryptographic policy choices. This is useful for an initiator to propose multiple cryptographic choices for the same policy described in its associated Proposal payload.

4.1.1.1.1. RKMP

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

4.1.1.1.2. TCP-AO Transforms

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



TCP-AO Transforms

o 0 (last) or 3 (more) (1 octet) - Specifies whether this is the last Transform Substructure in the Proposal.

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

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

o SendID (1 octet) - The TCP-AO KeyID that the sender will use to represent this Transform. The KeyID will be used to generate the keys independently on each network device at the end of the exchange.

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

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

Authentication Algorithm

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

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

Key Derivation Functions

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

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.

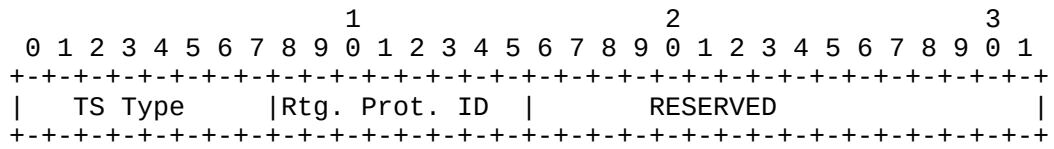
o O (Options) - This bit indicates whether or not TCP Options are to be included in the bytes protected by the authentication calculation. This bit is set to indicate that TCP Options are to be ignored and cleared to indicate that TCP Options are protected.

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)

4.2. Traffic Selector Payload

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



o TS Type (1 octet) - 1 for all routing protocols

o Rtg. Prot. ID (1 octet) - Specifies the routing protocol identifier for the current negotiation.

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

Routing Protocol

5. Operation Details

5.1. General

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

A routing protocol causes KMPRP to execute when it needs key material to establish neighbor adjacency. This can be as a result of the routing protocol neighbor being configured, neighbor changed or updated, a local rekey policy decision, or some other event dictated

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

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

5.2. Initial Key Specific Data Exchange

Routing protocols informs KMPRP of its new neighbor adjacency. It does so by creating a local entry in KMDB which consists of a Security algorithm, Key specific information, routing protocol client and the routing protocol neighbor. Upon a successful creation of such an entry KMPRP initiates KMPRP peering with the neighbor and starts initial KMPRP RP_INIT exchange explained in section 2.1 followed by the RP_AUTH exchanged explained in section 2.2. Once the key related information is successfully exchanged, KMDB may invoke the routing protocol client to provide key specific information updates if any.

5.3. Key Specific Data Rollover Exchange

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

6. Key Management Database (KMDB)

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

7. Protocol Interaction

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

8. IANA Considerations

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

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

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

9. Security Considerations

TBD

10. Acknowledgements

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

Many protocol definitions and protocol formats come from RFC 5996,

either by reference or inclusion.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

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

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

11.2. Informative References

[DH] Diffie, W. and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22 n. 6, June 1977.

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

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

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

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

Authors' Addresses

Mahesh Jethanandani
Cisco Systems
170 Tasman Drive
San Jose, California CA
USA

Phone: +1 (408) 527-8230
Fax:
Email: mjethanandani@gmail.com
URI:

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

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

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

Phone: _1 (408) 526-7183
Fax:
Email: keyupate@cisco.com
URI:

Dacheng Zhang
Huawei
Beijing,
China

Phone:
Fax:
Email: zhangdacheng@huawei.com
URI:

Sam Hartman
Painless Security

Phone:
Fax:
Email: hartmans@painless-security.com
URI:

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2012

P. Tran
B. Weis
Cisco Systems

October 24, 2011

The Use of G-IKEv2 for Multicast Router Key Management
draft-tran-karp-mrmp-00

Abstract

The G-IKEv2 key management protocol protects group traffic, usually in the form of IP multicast communications between a set of network devices. G-IKEv2 can be used 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 April 26, 2012.

Copyright Notice

Table of Contents

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 re-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-IKEv2for 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_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 primarly an optimization for scalability. Because there are likely to be network devices communicating in a routing protocol, this protocol is less desirable. 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 multicast routing protocol act as GMs. The choice of a GCKS is not restricted by this memo, but operationally it is most reliable for one of the GMs to take that role.

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 makes use of the G-IKEv2 protocol exchanges, state machine, and policy definitions whenever possible. However, as G-IKEv2 was developed exclusively for the use of IPsec, these protocol exchanges are incorporated by reference into the present key protocol definitions, and are exchanged using a different exchange type Group Initial Exchange (GSA_INIT) and Group member Authentication exchange (GSA_AUTH) [I-D.yeung-g-ikev2]. The use of exchange type will clearly differentiate this protocol from IKEv2.

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 GSA_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 GSA_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.

```
      Group Member(Initiator)                Key Server (Responder)
     ---------------------                  ----------------------
      HDR, SAi1, KEi, Ni      -->
                              <--      HDR, SAr1, KEr, Nr, [CERTREQ,]
```

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

```
      Group Member (Initiator)              Key Server (Responder)
      --------------------                  ----------------------
       HDR, SK {IDi, [CERT,] [CERTREQ,]
           [IDr,] AUTH, IDg}          -->
                                      <-- HDR, SK {IDr, [CERT,] AUTH,
                                                   SEQ], GSA, KD}
```

In the GSA_AUTH exchange, the group member sends the group identification that it wants to join or register to. 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 GSA_PULL exchange. In GSA_PULL exchange, the GM will send group ID that it wants to join, the key server response will include sequence (SEQ), policy (GSA) and key material (KD).

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

The group member and key server need to maintain the group association SA (GSA) to further communicate securely or the registration process above need to be repeated. Before the policy and traffic key expired, the key server can use multicast GSA_REKEY message to PUSH the fresh policy and keys to all its group members. This exchange is protected by the KEK policy and key sent from the key server to group member during registration. The key server will include sequence (SEQ), policy (GSA), and keying material (KD) payloads in the rekey message. A rekey message might be necessary if a key lifetime is about to expire, due to concern that a key might have been compromised, or some other reason.

```
      Group Member (Initiator)          Key Server (Responder)
      --------------------              ----------------------
                               <-   HDR, SK {SEQ, GSA, KD, AUTH}
```

The KS can delete group member by sending Delete (D) payloads in the GSA_REKEY message . The Delete (D) payloads are defined in Section 4 of [I-D.yeung-g-ikev2].

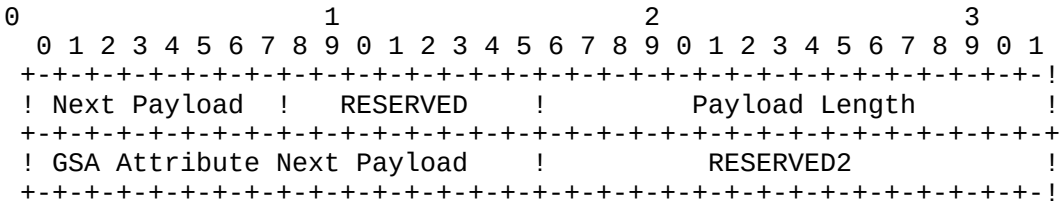
```
   Group Member (Initiator)              Server (Responder)
   ------------------                    ------------------

                                 <--    HDR, SK {[D,] ... , AUTH}
```

4. Header and Payload Formats

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

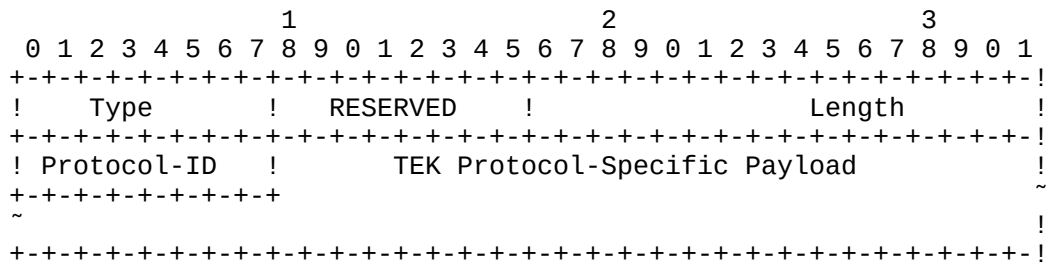
4.1. Group Security Association Payload

The Group Security Assocation (GSA) payload contains one or more sets of policy that a router is willing to use to protect a routing protocol. It is identitcal 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 attribute "next payload" types is the Traffic Encryption Key (TEK) payload. The TEK payload describes the Traffic Encryption Policy. This document define new protocol ID of TEK protocol specific payload for routing protocol OSPFv2, OSPFv3 and PIM.

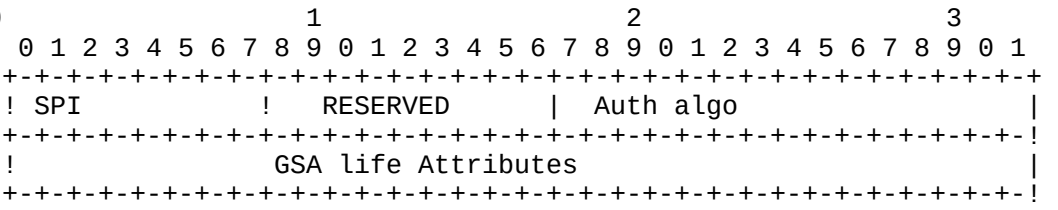


Protocol ID	Value
RESERVED	0
GSA_PROTO_IPSEC_ESP	1
GSA_PROTO_IPSEC_AH	2
GSA_PROTO_OSPFv2	TBD
GSA_PROTO_OSPFv3	TBD
GSA_PROTO_PIM	TBD

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:



SPI - (1 octet) Secure Parameter Index will be used in OSPFv2 header as Key ID (RFC 2328, Appendix D)

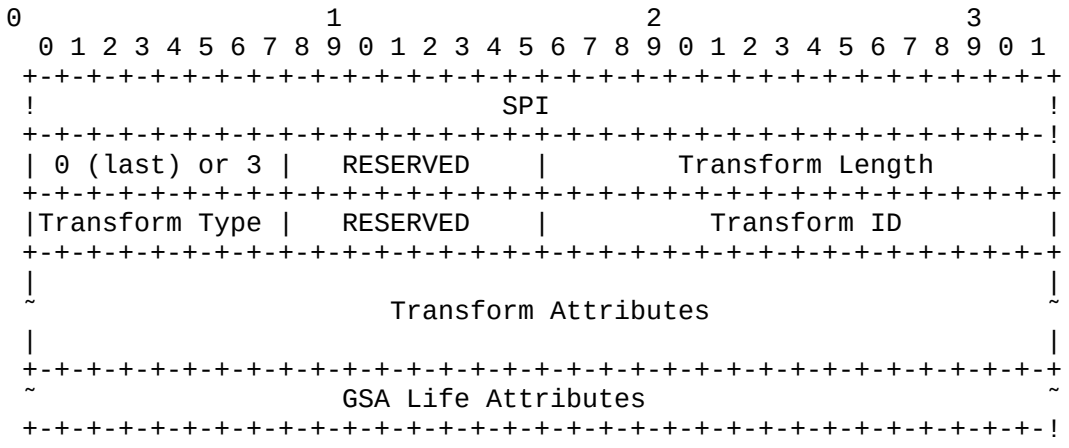
Auth algo - (2 octets) Authentication Algorithm

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)

GSA Life 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 similiar 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.



SPI (4 octets) - Secure Parameter Index

Transform - Same sa G-IKEv2 TEK transform define 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.

Description	Trans. Type	Used In
Encryption Algorithm (ENCR)	1	ESP
Integrity Algorithm (INTEG)	3	AH, optional in ESP
Extended Sequence Numbers (ESN)	5	AH and ESP

Transform Type 1 (Encryption Algorithm)

Name	Number	Defined In
ENCR_NULL	11	(RFC2410)
ENCR_AES_CBC	12	(RFC3602)

Transform Type 3 (Integrity Algorithm)

Name	Number	Defined In
NONE	0	
AUTH_HMAC_MD5_96	1	(RFC2403)
AUTH_HMAC_SHA1_96	2	(RFC2404)

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

5. IANA Considerations

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. Normative References

[I-D.hartman-karp-mrkmp]
Hartman, S. and D. Zhang, "Multicast Router Key Management Protocol (MRKMP)", draft-hartman-karp-mrkmp-01 (work in progress), March 2011.

[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.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen,

"Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

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: