

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 21, 2011

S. Hartman  
Painless Security  
D. Zhang  
Huawei  
October 18, 2010

Multicast Router Key Management Protocol (MRKMP)  
draft-hartman-karp-mrkmp-00.txt

Abstract

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

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, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Terminology . . . . .	3
1.2. Relationship to IKEv2 . . . . .	3
1.3. Relationship to GDOI . . . . .	4
2. Overview . . . . .	5
2.1. Types of Keys . . . . .	5
2.1.1. Key Encryption Key . . . . .	6
2.1.2. Protocol Keys . . . . .	6
2.2. GCKS Election . . . . .	7
2.3. Initial Exchange . . . . .	8
2.4. Group Join Exchange . . . . .	8
2.5. Group Key Management . . . . .	9
3. GCKS Election . . . . .	10
3.1. A new GCKS is Elected . . . . .	11
3.2. Merging Partitioned Networks . . . . .	11
4. Key Download Payload . . . . .	13
5. Initial Exchange Details . . . . .	14
6. Group Management Unicast Exchanges . . . . .	15
6.1. Group Join Exchange . . . . .	15
7. Group Key Management Operation . . . . .	16
7.1. General operation . . . . .	16
7.2. Out of Sequence Space . . . . .	16
7.3. Changing the Active GCKS . . . . .	16
8. Interface to Routing Protocol . . . . .	17
8.1. Joining a Group . . . . .	17
8.2. Priority Adjustment . . . . .	17
8.3. Leaving a Group . . . . .	17
9. Security Considerations . . . . .	19
10. Acknowledgements . . . . .	20
11. Informative References . . . . .	21
Authors' Addresses . . . . .	22

## 1. Introduction

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

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

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

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

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

### 1.1. Terminology

One key terminology question to answer is the definition of group. It appears that as used in this document, the term group corresponds to a routing protocol instance on a single link. However, this needs to be confirmed with TE routing protocols and with PIM. If that works out then a more precise term than group should be used in this document.

### 1.2. Relationship to IKEv2

IKEv2 [RFC5996] 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, 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 of 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.

For these reasons, this protocol borrows concepts and payloads from IKEv2 but does not normatively depend on the IKEv2 specification.

### 1.3. Relationship to GDOI

The IPsec Group Domain of Interpretation (GDOI) [RFC3547] provides a protocol that is structurally very similar to this one. As specified, IKE can be used to provide phase 1 authentication to a GCKS. After that, GDOI provides phase 2 messages to establish key-encryption keys and traffic keys. Key management operations can be accomplished via GDOI messages sent to the group after the phase 2 exchange.

GDOI is defined for IKE not for IKEv2. In addition, GDOI's phase 2 uses its own hashing mechanism and nonce mechanism to provide integrity protection and replay protection. Like IKE, GDOI has significant complexity to support phase 2 identities that are different than the phase 1 identity. GDOI requires a GCKS to have a signature key used to sign GDOI messages when the rekey protocol is used. Since attacks caused by members of the group masquerading as the GCKS are out of scope, this is significant unnecessary complexity in the protocol.

This protocol can be thought of as a simplified GDOI based on IKEv2 rather than IKE. However, integrity and replay mechanisms are taken from IKEv2. Support for phase 2 identities is removed as unneeded complexity. Security for the group key management messages is provided using symmetric primitives rather than asymmetric signatures. Phase 1 authentication will often still involve asymmetric signatures.

## 2. Overview

MRKMP is composed of several parts. There is an initial exchange used to establish a shared key with a GCKS and authenticate the identities of both parties. Unicast key management exchanges provide the ability to join a group or request updates to the group; group joins can also be combined with the initial exchange. There is an election protocol used by routers to determine which router will act as the GCKS; this protocol is not integrity protected, but a GCKS confirms its role when a member uses the unicast exchange to join the group. Finally, a GCKS uses multicast exchanges to update parameters of the group. This section briefly describes each of these parts of MRKMP. The later sections in the document describe the details of the protocols.

### 2.1. Types of Keys

MRKMP manipulates several different types of symmetric keys:

**preshared:** Preshared keys are one mechanism for authenticating one router to another during the initial exchange. These keys are configured by some mechanism such as manual configuration or a management application outside of the scope of MRKMP. A single preshared key can be used for all members of a group. Alternatively each pair of routers can have a different preshared key.

**peer key management key:** Routers share a key with the GCKS that is a result of the `mrkmp_init` exchange.

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

**protocol master key:** A protocol master key is the key exported by MRKMP for use by a routing protocol such as OSPF or IS-IS. The Protocol master key is the key that would be manually configured if a routing protocol is used without key management.

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

#### 2.1.1. Key Encryption Key

When a router wishes to join a group, the router performs the mrkmp\_init and mrkmp\_auth exchange with a GCKS. During this process the router can establish an association with a specific group. Part of that association will be delivery of a KEK and associated parameters.

Group key management messages are sent to a group address not unicast to an individual peer. The group key management messages are protected using the KEK. The group key management messages need to provide both integrity and confidentiality protection using the KEK.

As part of establishing the association, the router joining the group is given an expiration time for the KEK. A group key management message may establish a new KEK with new parameters.

From time to time, a GCKS may wish to either force early expiration of a KEK or allow a KEK to expire. Protocol master keys are permitted to be valid for somewhat longer than the KEK that created them so as to avoid disrupting routing when this happens. When a KEK is retired or expires without being replaced by a new KEK announced in the old KEK, group members need to perform a new initial exchange to the GCKS. This is useful for example if a router is no longer authorized to be part of the group.

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

#### 2.1.2. Protocol Keys

Current routing protocols directly use the protocol master key to integrity protect messages. One advantage for this approach is that the initial hello messages used for discovery and capability exchange can be protected using the same mechanism as other messages. Typically a sequence number is used for replay detection. Without

changing the key, the existing protocols are vulnerable to a number of serious denial of service attacks from replays.

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

Another approach would be to separate the protocol master key from the transport keys. For example the transport key used by a given peer could be a fresh key derived from the protocol master key and nonces announced by that peer. Some mechanism would need to make sure that the peer's announcement of its nonce was fresh; this mechanism would almost certainly involve some form of interaction with the router wishing to guarantee freshness. There are two key advantages of this separation between transport keys and protocol master keys. The first is that the interaction between the MRKMP and routing protocol can be simplified significantly. The second is that even when manually configured protocol master keys are used, replay and adequate DOS protection can be achieved.

## 2.2. GCKS Election

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

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

### 2.3. Initial Exchange

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

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

Then the identities of each party are cryptographically verified. This can be done using a preshared key or symmetric keys. Other mechanisms may be added as a future extension.

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

### 2.4. Group Join Exchange

The primary purpose of the unicast MRKMP messages is to get an initiator the information it needs to join a group and participate in a routing protocol. The initiator indicates what group it wants to join. XXX we need to discuss group naming--if MRKMP is limited to a subnet this may be as simple as saying that initiator wants to join the OSPF group or the IS-IS group.

The responder performs several checks. First, the responder confirms that the responder is currently acting as GCKS for the group in question. Then, the responder confirms that the initiator is permitted to join the group. If these checks pass, then the responder provides a key download payload to the initiator encrypted



in the peer key management key. As discussed in Section 2.1.2, the GCKS MUST change the protocol master key if a router was part of the group under the current protocol master key and reboots. In this case, the GCKS SHOULD provide the new and old protocol master key to the initiator, setting the validity times for the old key to permit reception but not transmission. The GCKS MUST use the mechanism in the next section to flood the new key to the rest of the group.

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

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

## 2.5. Group Key Management

The GCKS shares a KEK with all members of a group. The GCKS can send a multicast message to the group to update the set of protocol master keys, update the KEK, or retire the KEK and request new group join exchanges.

Typically the protocol master key is changed only when needed to provide replay protection or when the KEK changes. The KEK changes whenever a new GCKS is elected or whenever it is administratively desirable to change the keys. For example if an employee leaves an organization it might be desirable to change the KEKs. A KEK is retired whenever forward security is desired: whenever the authorization of who is permitted to be in a group changes and the GCKS needs to make sure that the router is no longer participating. Most authorization changes such as removing a router from service do not require forward security in practical deployments.

### 3. GCKS Election

The GCKS election process selects a single router on a link to act as GCKS for a group. Similar with other popular announcer electing mechanisms (e.g., VRRP, HSRP), in MRKMP, only GCKSes use multicast to periodically send Advertisement messages. Such advertisements can be used as heart beat packets to indicate the aliveness of GCKSes. In addition, a state machine with three states (Initial, GCKS, and Member) is specified for GCKS election. When a router is initially connected to a multicast network, its state is set as Initial. The router then sends a multicast initial advertisement, if a GCKS is working on the network, it will reply the router with an advertisement using unicast. After receiving the advertisement from the GCKS, the router will try to register with the GCKS using the initial exchange, and then the state of the router is transferred to Member. Note that when the router receives the advertisement it does not have the traffic distributed in the group. Thus, the integrity of the unicast advertisement does not have to be protected. After a certain period, if the router still does not receive any advertisement from a GCKS or other group members, the router then believe there is no other group member on the network and set its state as GCKS. If during the period the router does not receive any advertisement from a GCKS but receives advertisements from other routers on the network, router believes that the group is involved in a GCKS election process. Apart from the initialization of a multicast network, the fail-over of a GCKS can also trigger an election process. For instance, if a router does not receive the heart beat advertisement for a certain period, it will transfer its state to Initial and try to elect a new one. In a GCKS electing process, a router has to stay in the Initial state until a new GCKS is allocated. Particularly, the router first sends its initial advertisement with its priority and waits for a certain period. During the period, if a router receives an initial advertisement which consists of a lower priority, the router then sends the advertisement again with a limited rate. After period, if the router does not find any router with a higher priority, it announces itself as the GCKS. If two routers have the same priority, the one with the lowest IP source address used for messages on the link will be the GCKS. After a router transfer its state to GCKS, it will reply to the initial advertisements from other routers with GCKS advertisements, even when the initial advertisements consist of properties priorities than its priority. This approach guarantees that a GCKS will not be changed frequently after it has been elected. After receiving the GCKS advertisement of the new elected GCKS, other routers transfer their states to Member. However, if a GCKS G1 receives a GCKS advertisement from another router G2 and G2 is a more preferred GCKS, G1 follows the procedure in Section 3.2.

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

If a node transitions to GCKS state, it performs the procedure in Section 3.1.

### 3.1. A new GCKS is Elected

### 3.2. Merging Partitioned Networks

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

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

The more preferred GCKS generates a new key download payload including a KEK and the union of all the protocol master keys. The GCKS SHOULD mark the existing protocol master keys as expiring for usage in transmitted packets in a relatively short time. The GCKS SHOULD introduce a new protocol master key. This key download payload is returned to the less preferred GCKS and is sent out in the current KEK using a group key management message.

The less preferred GCKS sends the received key download payload encrypted in its existing KEK. XXX how many retransmits. After all retransmissions of this payload the less preferred GCKS sets its state to member.

As a result of this procedure, members learn the protocol master keys of both GCKSes and converge on a single KEK and GCKS. Changing the protocol master keys during a merge is important for protocols that use the protocol master key as a transport key. The new GCKS does not know which routers have joined the group with the other GCKS.

Therefore, it could not correctly detect one of these routers rebooting and change the protocol master key at that point. If the key is changed as part of the merge, replays are handled.

#### 4. Key Download Payload

What all is actually in the message you get at the end of phase 2 and that is sent out periodically during group key management

For the KEK, this needs to include the key itself, the algorithm (presumably drawn from the IKEv2 symmetric algorithms), key ID, group ID and the four lifetimes.

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

By four lifetimes we mean receive start, send start, send end and receive end. It's important that a key can be flooded out to all potential receivers before it is used for sending.

## 5. Initial Exchange Details

## 6. Group Management Unicast Exchanges

### 6.1. Group Join Exchange

If a router receives a group join exchange for a group for which it is not the GCKS, it MUST return a notification. If it knows the GCKS for the group then it returns MRKMP\_WRONG\_GCKS including the address of the GCKS in the notification payload. The initiator tries the group join exchange (probably with a new initial exchange) with the indicated router. If the responder does not know the GCKS for the group, either because it is not a member of the group or because its GCKS election state is initial, it returns the MRKMP\_GCKS\_UNKNOWN notification. If the responder is not trying to be a member of this group or has seen a more preferred GCKS advertisement in the election process then the potential\_candidate bit is clear, otherwise it is set. The initiator sets its GCKS election state to initial when receiving this notification. If the potential candidate bit is set in the notification then the initiator will accept GCKS election advertisements from the responder. If the potential candidate bit is clear, then the initiator will discard GCKS election advertisements from the responder until BLACKLIST\_TIMEOUT seconds have elapsed or until the initiator successfully joins the group.

## 7. Group Key Management Operation

Group key management messages are multicast from the GCKS to the group. The message contains the key identifier of a KEK, as well as encrypted/integrity-protected payloads. Inside the encrypted/integrity-protected payloads is a monotonically increasing sequence number, and payloads specific to the message being sent. Group members **MUST** ignore a message with a sequence number that is the same or less than the sequence number of the most recent message they have received.

### 7.1. General operation

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

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

### 7.2. Out of Sequence Space

### 7.3. Changing the Active GCKS



## 8. Interface to Routing Protocol

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

### 8.1. Joining a Group

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

### 8.2. Priority Adjustment

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

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

### 8.3. Leaving a Group

If a routing protocol terminates on an interface, MRKMP needs to be notified that group is no longer joined. MRKMP MUST stop participating in the GCKS election process, stop monitoring for key

management messages and if the current router is a GCKS, stop acting in that role.

## 9. Security Considerations

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

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

## 10. Acknowledgements

This draft is the result of a design discussion held after the IETF 78 KARMP meeting. The authors, David McGrew, Brian Weis and Gregory Lebovitz all contributed to the design meeting.

## 11. Informative References

- [RFC2627] Wallner, D., Harder, E., and R. Agee, "Key Management for Multicast: Issues and Architectures", RFC 2627, June 1999.
- [RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 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: Informational  
Expires: April 28, 2011

S. Hartman  
Painless Security  
D. Zhang  
Huawei  
October 25, 2010

Operations Model for Router Keying  
draft-hartman-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 28, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.



## Table of Contents

1. Introduction . . . . .	3
2. Requirements notation . . . . .	4
3. Breakdown of KARP configuration . . . . .	5
3.1. Integrity of the Key Table . . . . .	6
3.2. Management of Key Table . . . . .	6
3.3. Protocol Limitations from the Key Table . . . . .	7
3.4. VRFs . . . . .	7
4. Credentials and Authorization . . . . .	8
4.1. Preshared Keys . . . . .	9
4.2. Asymmetric Keys . . . . .	11
4.3. Public Key Infrastructure . . . . .	11
4.4. The role of Central Servers . . . . .	12
5. Grouping Peers Together . . . . .	13
6. Administrator Involvement . . . . .	15
6.1. Enrollment . . . . .	15
6.2. Handling Faults . . . . .	15
7. Upgrade Considerations . . . . .	17
8. Related Work . . . . .	18
9. Security Considerations . . . . .	19
10. Acknowledgments . . . . .	20
11. References . . . . .	21
11.1. Normative References . . . . .	21
11.2. Informative References . . . . .	21
Authors' Addresses . . . . .	22

## 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. This model would need to be expanded in cases where keys are configured per-area or per-domain. 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. 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:

- o The cryptographic algorithms are valid for the protocol.
- o The key derivation function is valid for the protocol.
- o 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.
- o 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.

### 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 one peer. This is an interesting security property: it is impossible to identify the peer sending a message cryptographically; it is only possible to identify a group of peers using cryptographic means. 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. However, for the process of authorization, it would be more complicated to identify peers this way and would not gain a security benefit in most deployments.

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. Their naming and authorization may differ. 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 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:

- o 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.
- o A peer is a router that this router might wish to communicate with. Peers may be identified by names or keys.
- o 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-01 (work in progress),  
September 2010.
- [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-03 (work in progress),  
July 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] Bellare, 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



INTERNET DRAFT  
Internet Engineering Task Force (IETF)  
Intended Status: Standards Track

Expires: 13 April 2010

R. Housley  
Vigil Security  
T. Polk  
NIST  
13 October 2010

Database of Long-Lived Symmetric Cryptographic Keys  
<draft-housley-saag-crypto-key-table-04.txt>

Status of this Memo

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

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

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

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

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

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

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 peersKeyID 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.

#### 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 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.

#### 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".

#### 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.

**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.

**NotBefore**

The NotBefore field specifies the earliest date and time in Universal Coordinated Time (UTC) at which this key should be considered for use. 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.

**NotAfter**

The NotAfter field specifies the latest date and time at which this key should be considered for use. 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 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 H2 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 H2;
- (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.

#### 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

No IANA actions are required.

{{{ RFC Editor: Please remove this section prior to publication. }}}}

## 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.

[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  
Internet Draft  
Intended status: Informational  
Expires: March, 2011

G. Lebovitz  
Juniper  
M. Bhatia  
Alcatel-Lucent  
September 2010

Keying and Authentication for Routing Protocols (KARP)  
Design Guidelines

draft-ietf-karp-design-guide-01.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on March 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described





in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

In the March of 2006 the IAB held a workshop on the topic of "Unwanted Internet Traffic". The report from that workshop is documented in RFC 4948 [RFC4948]. Section 8.2 of RFC 4948 calls for [t]ightening the security of the core routing infrastructure." Four main steps were identified for improving the security of the routing infrastructure. One of those steps was "securing the routing protocols' packets on the wire." One mechanism for securing routing protocol packets on the wire is the use of per-packet cryptographic message authentication, providing both peer authentication and message integrity. Many different routing protocols exist and they employ a range of different transport subsystems. Therefore there must necessarily be various methods defined for applying cryptographic authentication to these varying protocols. Many routing protocols already have some method for accomplishing cryptographic message authentication. However, in many cases the existing methods are dated, vulnerable to attack, and/or employ cryptographic algorithms that have been deprecated. 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. The overall roadmap reflects the input of both the security area and routing area in order to form a jointly agreed upon and prioritized work list for the effort.

## 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]

1. Introduction.....	3
2. Categorizing Routing Protocols.....	4
2.1. Category: Message Transaction Type.....	4
2.2. Category: Peer vs Group Keying.....	5
3. Consider the future existence of a KMP.....	6
3.1. Consider Asymmetric Keys.....	6
3.2. Cryptographic Keys Life Cycle.....	7
4. RoadMap.....	8
4.1. Work Phases on any Particular Protocol.....	8
4.2. Work Items Per Routing Protocol.....	11
5. Routing Protocols in Categories.....	12
6. Gap Analysis.....	16
7. Security Considerations.....	18
7.1. Use Strong Keys.....	19
7.2. Internal vs. External Operation.....	20
7.3. Unique versus Shared Keys.....	20
7.4. Out-of-Band External Configuration vs. Peer-to-Peer Key Management.....	22
8. Acknowledgments.....	24
9. IANA Considerations.....	24
10. References.....	25
10.1. Normative References.....	25
10.2. Informative References.....	25

## 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:

- o More secure mechanisms and practices for operating routers. This work is being addressed in the OPSEC Working Group.
- o 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. This work should be addressed through liaisons with those running the IRR's globally.

- o Specifications for cryptographic validation of routing

Expires March 2011

[Page 3]

message content. This work will likely be addressed in the SIDR Working Group.

- o Securing the routing protocols' packets on the wire

This document addresses the last bullet, securing the packets on the wire of the routing protocol exchanges.

Readers must refer to the [I-D.ietf-karp-threats-req] 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

For the purpose of this security roadmap definition, we will categorize the routing protocols into groups and have design teams focus on the specification work within those groupings. It is believed that the groupings will have like requirements for their authentication mechanisms, and that reuse of authentication mechanisms will be greatest within these grouping. The work items placed on the roadmap will be defined and assigned based on these categorizations. It is also hoped that, down the road in the Phase 2 work, we can create one Key Management Protocol (KMP) per category (if not for several categories) so that the work can be easily leveraged by the various Routing Protocol teams. 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. When configured properly, a KMP will enforce the key freshness policy of two 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 categorization defines four 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] [RFC3036], 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.

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 consumption by 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 of the fact that 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, <http://www.ietf.org/html.charters/msec-charter.html>) and may be helpful to review, but is not directly applicable.

### 2.2. Category: Peer vs Group Keying

The second axis of categorization groups protocols by the keying mechanism that will be necessary for distributing session keys to the actual Routing Protocol transports. They are:

#### Peer keying

One router sends the keying messages directly and only to one other router, such that a one-to-one, unique keying security association (SA) is established between the two routers. This would be employed by protocols like BGP, BFD, LDP, etc.

#### 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 between 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 KMP

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

However, when conducting the design work on a manual keyed version of a routing protocol's authentication, consideration must be made for the eventual use of a KMP. In particular, design teams must consider what parameters would need to be handed down to the Routing Protocol by the KMP.

Consider: some sort of security association identifier (e.g. IPsec ESP's SPI, or TCP-AO's KeyID), key life times which may be represented either in bytes or seconds, the cryptographic algorithms being used, the keys themselves, and the direction of the keys (i.e. receiveKey, sendKey).

#### 3.1. Consider Asymmetric Keys

The use of asymmetric keys can be a very powerful way to authenticate machine peers as are found in routing protocol peer exchanges. If generated on the machine, and never moved off the machine, these keys will be very secret, and will not be subject to change if an administrator leaves the organization. Since the keys are totally random, and very long, 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 for extremely valuable keys like the root key pair used by a certifying 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. Elliptic Curve Cryptography [RFC4492] (ECC) appears to be secure with shorter keys than those needed by other asymmetric key algorithms. NIST guidelines 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 over the SSH [RFC4252] and [RFC4253]. As such, they will also have the ability to generate and store an asymmetric key pair, because this is the commonly used method that users authenticate the SSH service when connecting to the router for management sessions.

Once 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 either raw keys, the more easily administrable self-signed certificate format, or a PKI issued certificate credential.

Regardless which form we eventually standardize, the proof of this identity presentation can be as simple as the SHA-1 fingerprint, which is represented in a very human readable and transferable form of 20 pairs of ASCII characters. More complexly, but also more securely, 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 much shorter. In other words, the entire list for a given organization's router ID & SHA-1 fingerprints could easily be held in a router's configuration file, uploaded, downloaded and move about at will. And it doesn't matter who sees or gains access to these fingerprint strings, because they are meant to be distributed publicly.

### 3.2. Cryptographic Keys Life Cycle

Cryptographic keys should have a limited lifetime and must change when an operator who had access to them leaves. Using the key chains also does not help as one still has to change all the keys in the keychain 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 adjacencies. So the first step is to fix all routing protocols so that they 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 his or her 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 systems security motivations for doing the same.

On the other hand, where manual key distribution methods are subject to human error and frailty, more frequent 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 get 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 with 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 till the cryptographic keying material for individual sessions is derived from the keying material stored in the database of long-lived cryptographic keys as described in [I-D.housley-saag-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 work phase for any protocol would be a two step process where the first would be to fix the manual key management procedures that currently exists within the routing protocols today using modern cryptography algorithms, key agility and then later move to an automated key management mechanism. This is like a crawl, walk and run process. In order to deliver that to the operators in a way that we could complete these action items a little bit a time and make some

incremental advance over what is currently deployed in the wild, we believe that it is therefore useful to cleanly separate the key management protocol from the routing transport subsystem mechanism. 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 parameters" so that it can refer to some module written by people good in security and who will be maintaining it over the time 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 somewhat regular basis will be operationally easy, causing no service interruption.

Achieving the above described end-state using manual keys may only be pragmatic in very small deployments. In larger deployments, this end state will be much more operationally difficult to reach with only manual keys. Thus, there will be a need for key life cycle management, in the form of a key management protocol, or KMP. We expect that the two forms, manual key usage and KMP usage, 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:

Expires March 2011

[Page 9]



1. Enhance the Routing Protocol's current authentication mechanism. 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 keying framework. It is understood and accepted that the existing keying 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-req] for the list of the requirements for Phase 1 work.
2. Develop an automated keying framework. The second phase will focus on the development of an automated keying framework to facilitate unique pair-wise (or perhaps 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 a number of attacks such as inter-connection replay and two-time pads. 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 key identifiers, derive Traffic Keys, determine re-keying, synchronize their keying state, signal various keying events, notify with error messages, etc.

The framework for any one Routing Protocol will fall under, and be able to leverage, the generic framework described in

[I-D.ietf-karp-framework]

## 4.2. Work Items Per Routing Protocol

Each Routing Protocol will have a team (the [Routing\_Protocol]-KARP team) working on incrementally improving their Routing Protocol's security. These teams will have the following main work items:

## PHASE 1:

## Characterize the RP

Assess the Routing Protocol to see what authentication mechanisms it has today. Does it need 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 to modern, strong security algorithms and mechanisms, per the Requirements document [I-D.ietf-karp-threats-req]. The goal here is to determine what is needed for the Routing Protocol alone to be used securely with at least manual keys.

## 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 a document(s)

Expires March 2011

[Page 11]

#### KMP Analysis

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

#### Gap Analysis

Enumerate the requirements for this protocol to move from its current security state to its optimal state.

#### Define, Assign, Design

Create a deliverables list of the design and specification work, with milestones. Define owners. Do 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 to the Routing Protocol.

There will also be a team formed to work on the base framework mechanisms for each of the main categories, i.e. the blocks and API's represented in [I-D.ietf-karp-framework].

### 5. Routing Protocols in Categories

This section groups the Routing Protocols into like 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 and MSDP

The Routing Protocols that fall into the category of the one-to-one peering messages, and will use peer keying protocols. BGP [RFC4271] and MSDP [RFC3618] are transmitted over TCP, while LDP [RFC5036] uses UDP. 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 is already occurring in the TCPM Working Group, on the TCP-AO [RFC5925] document, as well as its cryptography-helper document, TCP-AO-CRYPTO [RFC5926]. However, this cannot be used for LDP as LDP runs 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, ISIS, and RIP

The Routing Protocols that fall into the category Group keying with one-to-many peering messages includes OSPF [RFC2328], ISIS [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, or IGP's. The keying mechanisms and use 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 aliveness detection mechanism, Bidirectional Forwarding Detection (BFD) 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 with respect to the sequence number scheme that is generally deployed to protect against the 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 authenticity 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, per interface or per neighbor.

RSVP relies on per peer authentication mechanism, where each hop authenticates its neighbor with a shared key or 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. 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 generating the RSVP messages can, in particular, 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 RSVP messages intact or confidential, depending on whether authentication or encryption (or both) is used. This means only that the message has not been altered or seen by another, non-trusted node. Implicitly each node trusts each other node with which it has a trust relationship established via the mechanisms here to adhere to the protocol specifications laid out by the various standards. Note that in any group keying scheme like GDOI a node trusts all the other members of the group.

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 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, folks MAY make use of the security features already defined for RSVP-TE [RFC3209]. This may require some coordination between the domains to share the keys (see [RFC2747] and [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 only be working on RSVP-TE as the native RSVP lies outside the scope of the WG charter.

#### PIM-SM and PIM-DM

Finally, the multicast protocols of PIM-SM [RFC4601] and PIM-DM [RFC3973] will be handled 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 occurred in RFC 4601 [RFC4601], and this work is being 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, which is why this roadmap is being created.

## 6. Gap Analysis

The [I-D.ietf-karp-threats-req] document lists the generic requirements for the security and authentication mechanisms that must exist for the various routing and signaling 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-req]. This document contains detailed descriptions of the threat analysis for routing protocol authentication in general. Note that it will not contain all the authentication-related threats for any one routing protocol, or category of routing protocol. The design team must conduct a threat analysis to determine if specific threats beyond those in the [I-D.ietf-karp-threats-req] document exist, and to describe those threats.

The [I-D.ietf-karp-threats-req] document also contains many requirements around security matters. The different routing protocol design teams must walk through each section of the requirements and determine one by one how their protocol either does or does not address 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 one set of cryptography algorithms or constructions 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 is capable of supporting algorithms other than its defaults, in order to adapt to future discoveries.

Ideally, authentication MUST be performed on routing protocols packets oblivious to the order in which they have arrived, so that it does not get influenced by packets loss and reordering.

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

Design teams while defining the new authentication and security mechanisms MUST design in such a manner that the routing protocol authentication mechanism remains oblivious of 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.

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. 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.

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.

## 7. 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 in a necessary system of security improvements.

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.

## 7.1. 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.

In addition to using a strong key/PSK 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 per-connection by a KDF. The burden for doing so is understandable much higher than for using the same static Traffic Key across all peering routers. This is why use of a KMP network-wide increases peer-wise security so greatly, because now each set of peers can enjoy a unique Traffic Key, and if an attacker sitting between two routers learns or guesses the Traffic Key for that connection, she doesn't gain access to all the other connections as well.

However, whenever using manual keys, it is best to design a system where a given 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 MAC operation are peer-wise unique, it provides inter-connection replay protection, and, if the per-message MAC covers some connection counter, intra-connection replay protection.

Note that in the composition of 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/deployers, a specification may not want to force the deployer to enter a PSK of exactly 16 bytes. Instead, a specification may call for a sub-key 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 sub-key routine would act as a key extractor to derive a second 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] as listed above. We simply attempted to "put a fence around stupidity", in as much as possible.

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.

## 7.2. Internal vs. External Operation

The designers 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 utmost security mechanisms enabled in all deployments, 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 between 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. The 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.

## 7.3. 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 unscalable.

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 across their entire network. It is common to see the same key in use for years, and that being the same key that was entered when authentication was originally configured, or the routing gear deployed.

The goal for designers is to create authentication mechanisms that are easy for the operators to deploy and manage, and still use unique keys between peers (or small groups on multi-access links), and within between sessions. 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 (or few) key, compared to the inconvenience they would experience if they deployed the same authentication mechanism using unique pair-wise 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 will 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 keys 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 his job easier. Using the same key across the network makes the attack vulnerability surface more penetrable than unique keys. 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 designers may need to use shared keys in order to solve the given problem space. For example, a multicast packet is sent once but then observed and consumed by several routing neighbors. If unique keys were used per neighbor, the benefit of multicast would be erased because the casting peer would have to create a different announcement packet/stream for each listening peer. Though this may be desired and acceptable in some small amount of use cases, it is not the norm. Shared group keys are an acceptable solution here, and much work has been done already in this area (see MSEC working group).

#### 7.4. Out-of-Band External Configuration vs. Peer-to-Peer Key Management

This section discusses the security and use case considerations for keys placed on devices through out-of-band configurations versus through one routing peer-to-peer key management protocol exchanges. Note, when we say here "Peer-to-Peer KMP" we do not mean in-band to the Routing Protocol. Instead, we mean that the exchange occurs in-line, over IP, between the two routing peers directly. In peer-to-peer KMP the peers handle the key and security association negotiation between themselves directly, whereas in an out-of-band configuration system the keys are placed onto the device through some other configuration or management method or interface.

An example of an out-of-band external mechanism could be an administrator who makes a remote management connection (e.g. using SSH) to a router and manually enters the keying information -- like the algorithm, the key(s), the lifetimes, etc. Another example could be an OSS system which inputs the same information via 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 mechanism include: lack of scale-ability, complexity and speed of changing if a 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 defined by a human. Those keys then need to be placed into the OSS system, manually, and

the OSS system then needs to push the change -- often during a very limited change window -- into the relevant devices. If there are multiple organizations involved in these connections, this process is greatly complicated.

The benefits 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 of control. Operators have mechanisms in place for this already. In small environments with few routers, a manual system is not difficult to employ.

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

The benefits of a peer-to-peer KMP are several. It results in key(s) that are privately generated, and not recorded permanently anywhere. Since the traffic keys used in a particular connection are not a fixed part of a device configuration no steal-able data exists anywhere else in the operator's systems which can be stolen, e.g. in the case of a terminated or turned employee. If a server or other data store is stolen or compromised, the thieves gain no access to current traffic keys. They may gain access to key derivation material, like a PSK, but not current traffic keys in use. In this example, these PSKs can be updated into 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 would likely not even 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 re-keys operations occur without any operator involvement or oversight. This keeps keys fresh.

The drawbacks to using a KMP are few. First, a KMP requires more cryptographic processing for the router at the very beginning of a connection. This will add some minor start-up time to connection establishment versus a purely manual key approach. Once a connection with traffic keys have been established via a KMP, the performance is the same in the KMP and the out-of-band case. KMPs also add another layer of

protocol and configuration complexity which can fail or be mis-configured. 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.

The desired end goal for KARP WG is develop a strong peer-to-peer KMP as an Out-of-band external Key Management protocol is out of scope.

Within this there are two approaches for key management:

The first, is to use an Out-of-band Key Management protocol that runs independent of the routing and the signaling protocols. It could run on its own port and could use its own transport. When the routing protocols need a key, they would contact the local instance of this key management protocol and request a key. This instance generates a key which is delivered to the routing protocols for them to use for authenticating their protocol packets. This Key Management protocol could either be an existing key management protocol like ISAKMP/IKE, GKMP, etc. which is extended for the routing protocols, or could be a new one, designed and written from scratch.

The second, is to define an In-band Key Management protocol where the existing routing protocols are extended to incorporate 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.

## 8. 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, Michael Barnes and Vishwas Manral for their comments on the draft.

## 9. IANA Considerations

This document places no requests to IANA.

#### 10.1. 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.

#### 10.2. 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 1997.
- [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.
- [RFC3036] Andersson, L., et. al, "LDP Specification", RFC 3036, January 2001.
- [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.
- [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.

- [RFC5151] Farrel, A., et. al., "Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", February 2008.
- [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.
- [I-D.ietf-karp-threats-req] Lebovitz, G., "KARP Threats and Requirements", Work in Progress, February 2010.
- [I-D.ietf-karp-framework] Lebovitz, G., "Framework for Cryptographic Authentication of Routing Protocol Packets on the Wire", Work in Progress, February 2010.
- [I-D.housley-saag-crypto-key-table] Housley, R. and Polk, T., "Database of Long-Lived Cryptographic Keys" , Work in Progress, September 2009
- [I-D.weis-gdoi-mac-tek] Weis, B. and S. Rowles, "GDOI Generic Message Authentication Code Policy", Work in Progress, July 2008.
- [IRR] Merit Network Inc , "Internet Routing Registry Routing Assets Database", 2006, <http://www.irr.net/>.

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

Phone:  
Email: gregory.ietf@gmail.com

Manav Bhatia  
Alcatel-Lucent  
Bangalore  
India

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

KARP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 12, 2011

G. Lebovitz  
Juniper Networks, Inc.  
M. Bhatia  
Alcatel-Lucent  
R. White  
Cisco Systems  
October 9, 2010

The Threat Analysis and Requirements for Cryptographic Authentication of  
Routing Protocols' Transports  
draft-ietf-karp-threats-reqs-01

Abstract

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

This document has two main parts - the first describes the threat analysis for attacks against routing protocols' transports and the second enumerates the requirements for addressing the described threats. This document, along with the KARP design guide and KARP framework documents, will be used by KARP design teams for specific protocol review and overhaul. This document reflects the input of both the IETF's Security Area and Routing Area in order to form a jointly agreed upon guidance.

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 12, 2011.

#### Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	4
1.1. Terminology . . . . .	4
1.2. Requirements Language . . . . .	7
1.3. Scope . . . . .	7
1.4. Incremental Approach . . . . .	8
1.5. Goals . . . . .	9
1.6. Non-Goals . . . . .	12
1.7. Audience . . . . .	12
2. Threats . . . . .	14
2.1. Threats In Scope . . . . .	14
2.2. Threats Out of Scope . . . . .	16
3. Requirements for Phase 1 of a Routing Protocol Transport's Security Update . . . . .	18
4. Security Considerations . . . . .	23
5. IANA Considerations . . . . .	24
6. Acknowledgements . . . . .	25
7. Change History (RFC Editor: Delete Before Publishing) . . . .	26
8. References . . . . .	27
8.1. Normative References . . . . .	27
8.2. Informative References . . . . .	27
Authors' Addresses . . . . .	30

## 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 "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:

- o More secure mechanisms and practices for operating routers. This work is being addressed in the OPSEC Working Group.
- o 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. This work should be addressed through liaisons with those running the IRR's globally.
- o Specifications for cryptographic validation of routing message content. This work will likely be addressed in the SIDR Working Group.
- o Securing the routing protocols' packets on the wire

This document addresses the last item in the list above, securing the the transmission of routing protocol packets on the wire, or rather securing routing protocol transport. This effort is referred to as Keying and Authentication for Routing Protocols, or "KARP". This document specifically addresses the threat analysis for per packet routing protocol transport authentication, and the requirements for protocols to mitigate those threats.

This document is one of three that together form the guidance and instructions for KARP design teams working to overhaul routing protocol transport security. The other two are the KARP Design Guide [I-D.ietf-karp-design-guide] and the KARP Framework [I-D.ietf-karp-framework].

### 1.1. Terminology

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

#### PSK (Pre-Shared Key)

A key used by both peers in a secure configuration. Usually exchanged out-of-band prior to a first connection.

#### Routing Protocol

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

#### PRF

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

#### KDF (Key derivation function)

A KDF is a function in which an input key and other input data is used to generate (or derive) keying material that can be employed by cryptographic algorithms. The key that is input to a KDF is called a key derivation key. KDFs can be used to generate one or more keys from either (i) a uniformly random or pseudorandom seed value or (ii) a Diffie-Hellman shared secret or (iii) a non-uniform random source or (iv) a passphrase.

#### Identifier

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

#### Identity Proof

Once the form of identity is decided, then there must be a cryptographic proof of that identity, that the peer really is who they assert themselves to be. Proof of identity can be arranged between the peers in a few ways, for example pre-shared keys, raw asymmetric keys, or a more user-friendly representation of



assymmetric keys, such as a certificate. Certificates can be used in a way requiring no additional supporting systems -- e.g. public keys for each peer can be maintained locally for verification upon contact. Certificate management can be made more simple and scalable with the use of minor additional supporting systems, as is the case with self-signed certificates and a flat file list of "approved thumbprints". Self-signed certificates will have somewhat lower security properties than Certificate Authority signed certificates. The use of these different identity proofs vary in ease of deployment, ease of ongoing management, startup effort, ongoing effort and management, security strength, and consequences from loss of secrets from one part of the system to the rest of the system. For example, they differ in resistance to a security breach, and the effort required to remediate the whole system in the event of such a breach. The point here is that there are options, many of which are quite simple to employ and deploy.

#### SA (Security Association)

The parameters and keys that together form the required information for processing secure sessions between peers. Examples of items that may exist in an SA include: Identifier, PSK, Traffic Key, cryptographic algorithms, key lifetimes.

#### KMP (Key Management Protocol)

A protocol used between peers for creation, distribution and maintenance of secret keys. It determines how secret keys are generated and made available to both the parties. If session or traffic keys are being used, KMP is responsible for generating them and determining when they should be renewed.

A KMP is helpful because it negotiates unique, pair wise, random keys without administrator involvement. It also negotiates as mentioned earlier several of the SA parameters required for the secure connection, including key life times. It keeps track of those lifetimes using counters, and negotiates 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.

#### KMP Function

Any actual KMP used in the general KARP solution framework

#### Peer Key

Keys that are used between peers as the identity proof. These keys may or may not be connection specific, depending on how they were established, and what form of identity and identity proof is being used in the system. This would generally be given by the KMP that would later be used to derive fresh traffic keys.

#### Traffic Key

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

Definitions of items specific to the general KARP framework are described in more detail in the KARP Framework [I-D.ietf-karp-framework] document.

### 1.2. Requirements Language

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

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

### 1.3. Scope

Three basic services (or techniques) may be employed in order to secure any piece of data as it is transmitted over the wire: privacy, authentication, and message integrity. The focus for this effort, and the scope for this roadmap document, will be message authentication and packet integrity only. This work explicitly excludes, at this point in time, privacy services. Non-repudiation is also excluded as a goal at this time. Since the objective of most routing protocols is to broadly advertise the routing topology, routing messages are commonly sent in the clear; confidentiality is not normally required for routing protocols. However, ensuring that

routing peers truly are the trusted peers expected, and that no rogue peers or messages can compromise the stability of the routing environment is critical, and thus our focus. Privacy and non-repudiation may be addressed in future work.

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

Therefore, the scope of this roadmap of work includes:

- o Making use of existing routing protocol transport security mechanisms, where they exist, and enhancing or updating them as necessary for modern cryptographic best practices
- o Developing a framework for using automatic key management in order to ease deployment, lower cost of operation, and allow for rapid responses to security breaches
- o Specifying the automated key management protocol that may be combined with the bits-on-the-wire mechanisms.

This document does not contain protocol specifications. Instead, it defines the areas where protocol specification work is needed and sets a direction, a set of requirements, and a relative priority for addressing that specification work.

There are a set of threats to routing protocols that are considered in-scope for this document, and a set considered out-of- scope. These are described in detail in the Threats (Section 2) section below.

#### 1.4. Incremental Approach

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

out places where agreement has been reached on such trade offs.

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

This gives the vendors and implementors the guidance they need to tune their software and hardware appropriately over time. Last, some security mechanisms require the build out of other operational support systems, and this will take time. An example where these three reasons are at play in an incremental improvement roadmap is seen in the improvement of BGP's [RFC4271] security via the update of the TCP Authentication Option (TCP-AO) [I-D.ietf-tcpm-tcp-auth-opt] effort. It would be ideal, and reflect best common security practice, to have a fully specified key management protocol for negotiating TCP-AO's authentication material, using certificates for peer authentication in the keying.

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

#### 1.5. Goals

The goals and general guidance for the KARP work follow:

1. Provide authentication and integrity protection for packets on the wire of existing routing protocols
2. Deliver a path to incrementally improve security of the routing infrastructure as explained in the earlier sections.

3. The deployability of the improved security solutions on currently running routing infrastructure equipment. This begs the consideration of the current state of processing power available on routers in the network today.
4. Operational deployability - A solutions acceptability will also be measured by how deployable the solution is by common operator teams using common deployment processes and infrastructures. I.e. We will try to make these solutions fit as well as possible into current operational practices or router deployment. This will be heavily influenced by operator input, to ensure that what we specify can -- and, more importantly, will -- be deployed once specified and implemented by vendors. Deployment of incrementally more secure routing infrastructure in the Internet is the final measure of success. Measurably, we would like to see an increase in the number of surveyed respondents who report deploying the updated authentication mechanisms anywhere across their network, as well as a sharp rise in usage for the total percentage of their network's routers.

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

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

- B. For external peers, difficulty of coordination with the other organization is an issue. Once they find the correct contact at the other organization (not always so easy), the coordination function is serialized and on a per peer/AS basis. The coordination is very cumbersome and tedious to execute in practice.
  - C. Keys must be changed at precisely the same time, or at least within 60 seconds (as supported by two major vendors) in order to limit connectivity outage duration. This is incredibly difficult to do, operationally, especially between different organizations.
  - D. Relatively low priority compared to other operational issues.
  - E. Lack of staff to implement the changes device by device.
  - F. There are three use cases for operational peering at play here: peers and interconnection with other operators, Internal BGP and other routing sessions within a single operator, and operator-to-customer-CPE devices. All three have very different properties, and all are reported as cumbersome. One operator reported that the same key is used for all customer premise equipment. The same operator reported that if the customer mandated, a unique key could be created, although the last time this occurred it created such an operational headache that the administrators now usually tell customers that the option doesn't even exist, to avoid the difficulties. These customer-unique keys are never changed, unless the customer demands so. The main threat at play here is that a terminated employee from such an operator who had access to the one (or few) keys used for authentication in these environments could easily wage an attack -- or offer the keys to others who would wage the attack -- and bring down many of the adjacencies, causing destabilization to the routing system.
- 5. Whatever mechanisms we specify need to be easier than the current methods to deploy, and should provide obvious operational efficiency gains along with significantly better security and threat protection. This combination of value may be enough to drive much broader adoption.
  - 6. Address the threats enumerated above in the "Threats" section (Section 2) for each routing protocol, along a roadmap. Not all threats may be able to be addressed in the first specification update for any one protocol. Roadmaps will be defined so that both the security area and the routing area agree on how the

threats will be addressed completely over time.

7. Create a re-usable architecture, framework, and guidelines for various IETF working teams who will address these security improvements for various Routing Protocols. The crux of the KARP work is to re-use that framework as much as possible across relevant Routing Protocols. For example, designers should aim to re-use the key management protocol that will be defined for BGP's TCP-AO key establishment for as many other routing protocols as possible. This is but one example.
8. Bridge any gaps between IETF's Routing and Security Areas by recording agreements on work items, roadmaps, and guidance from the Area leads and Internet Architecture Board (IAB, [www.iab.org](http://www.iab.org)).

#### 1.6. Non-Goals

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

- o Privacy of the packets on the wire. Once this roadmap is realized, we may revisit work on privacy.
- o Message content validity (routing database validity). This work is being addressed in other IETF efforts, like SIDR.

#### 1.7. Audience

The audience for this document includes:

- o Routing Area working group chairs and participants - These people are charged with updates to the Routing Protocol specifications. Any and all cryptographic authentication work on these specifications will occur in Routing Area working groups, with close partnership with the Security Area. Co- advisors from Security Area may often be named for these partnership efforts.
- o Security Area reviewers of routing area documents - These people are delegated by the Security Area Directors to perform reviews on routing protocol specifications as they pass through working group last call or IESG review. They will pay particular attention to the use of cryptographic authentication and corresponding security mechanisms for the routing protocols. They will ensure that incremental security improvements are being made, in line with this roadmap.
- o Security Area engineers - These people partner with routing area authors/designers on the security mechanisms in routing protocol

specifications. Some of these security area engineers will be assigned by the Security Area Directors, while others will be interested parties in the relevant working groups.

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



## 2. Threats

In RFC4949 [RFC4949], a threat is defined as a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. This section defines the threats that are in scope for this roadmap, and those that are explicitly out of scope. This document leverages the "Generic Threats to Routing Protocols" model, RFC 4593 [RFC4593], capitalizes terms from that document, and offers a terse definition of those terms. (More thorough description of routing protocol threats sources, motivations, consequences and actions can be found in RFC 4593 [RFC4593] itself). The threat listings below expand upon these threat definitions.

### 2.1. Threats In Scope

The threats that will be addressed in this roadmap are those from OUTSIDERS, attackers that may reside anywhere in the Internet, have the ability to send IP traffic to the router, may be able to observe the router's replies, and may even control the path for a legitimate peer's traffic. These are not legitimate participants in the routing protocol. Message authentication and integrity protection specifically aims to identify messages originating from OUTSIDERS.

The concept of OUTSIDERS can be further refined to include attackers who are terminated employees, and those sitting on-path.

- o On-Path - attackers with control of a network resource or a tap along the path of packets between two routers. An on-path outsider can attempt a man-in-the-middle attack, in addition to several other attack classes. A man-in-the-middle (MitM) attack occurs when an attacker who has access to packets flowing between two peers tampers with those packets in such a way that both peers think they are talking to each other directly, when in fact they are actually talking to the attacker only. Protocols conforming to this roadmap will use cryptographic mechanisms to prevent a man-in-the-middle attacker from situating himself undetected.
- o Terminated Employees - in this context, those who had access router configuration that included keys or keying material like pre-shared keys used in securing the routing protocol. Using this material, the attacker could send properly MAC'd spoofed packets appearing to come from router A to router B, and thus impersonate an authorized peer. The attacker could then send false traffic that changes the network behavior from its operator's design. The goal of addressing this source specifically is to call out the case where new keys or keying material becomes necessary very quickly, with little operational expense, upon the termination of

such an employee. This grouping could also refer to any attacker who somehow managed to gain access to keying material, and said access had been detected by the operators such that the operators have an opportunity to move to new keys in order to prevent an attack.

These attack actions are in scope for this roadmap:

- o Spoofing - when an unauthorized device assumes the identity of an authorized one. Spoofing can be used, for example, to inject malicious routing information that causes the disruption of network services. Spoofing can also be used to cause a neighbor relationship to form that subsequently denies the formation of the relationship with the legitimate router.
- o Falsification - an action whereby an attacker sends false routing information. To falsify the routing information, an attacker has to be either the originator or a forwarder of the routing information. Falsification may occur by an originator, or a forwarder, and may involve overclaiming, misclaiming, or mistatement of network resource reachability. We must be careful to remember that in this work we are only targeting falsification from outsiders as may occur from tampering with packets in flight. Falsification from BYZANTINES (see the Threats Out of Scope section (Section 2.2) below) are not addressed by the KARP effort.
- o Interference - when an attacker inhibits the exchanges by legitimate routers. The types of interference addressed by this work include:
  - A. Adding noise
  - B. Replaying out-dated packets
  - C. Inserting messages
  - D. Corrupting messages
  - E. Breaking synchronization
  - F. Changing message content
- o DoS attacks on transport sub-systems - This includes any other DoS attacks specifically based on the above attack types. This is when an attacker sends spoofed packets aimed at halting or preventing the underlying protocol over which the routing protocol runs, for example halting a BGP session by sending a TCP FIN or RST packet. Since this attack depends on spoofing, operators are

encouraged to deploy proper authentication mechanisms to prevent such attacks.

- o DoS attacks using the authentication mechanism - This includes an attacker sending packets which confuse or overwhelm a security mechanism itself. An example is initiating an overwhelming load of spoofed authenticated route messages so that the receiver needs to process the MAC check, only to discard the packet, sending CPU levels rising. Another example is when an attacker sends an overwhelming load of keying protocol initiations from bogus sources. All other possible DoS attacks are out of scope (see next section).
- o Brute Force Attacks Against Password/Keys - This includes either online or offline attacks where attempts are made repeatedly using different keys/passwords until a match is found. While it is impossible to make brute force attacks on keys completely unsuccessful, proper design can make such attacks much harder to succeed. For example, the key length should be sufficiently long so that covering the entire space of possible keys is improbable using computational power expected to be available 10 years out or more. Using per session keys is another widely used method for reducing the number of brute force attacks as this would make it difficult to guess the keys.

## 2.2. Threats Out of Scope

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

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

- o Sniffing - passive observation of route message contents in flight
- o Falsification by Byzantine sources - unauthorized message content by a legitimate authorized source.
- o Interference due to:
  - A. Not forwarding packets - cannot be prevented with cryptographic authentication
  - B. Delaying messages - cannot be prevented with cryptographic authentication

- C. Denial of receipt - cannot be prevented with cryptographic authentication
- D. Unauthorized message content - the work of the IETF's SIDR working group (<http://www.ietf.org/html.charters/sidr-charter.html>).
- E. Any other type of DoS attack. For example, a flood of traffic that fills the link ahead of the router, so that the router is rendered unusable and unreachable by valid packets is NOT an attack that this work will address. Many other such examples could be contrived.

### 3. Requirements for Phase 1 of a Routing Protocol Transport's Security Update

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

1. Clear definitions of which elements of the transmission (frame, packet, segment, etc.) are protected by the authentication mechanism
2. Strong algorithms, and defined and accepted by the security community, MUST be specified. The option should use algorithms considered accepted by the IETF's Security community, which are considered appropriately safe. The use of non-standard or unpublished algorithms SHOULD BE avoided.
3. Algorithm agility for the cryptographic algorithms used in the authentication MUST be specified, i.e. more than one algorithm MUST be specified and it MUST be clear how new algorithms MAY be specified and used within the protocol. This requirement exists in case one algorithm gets broken suddenly. Research to identify weakness in algorithms is constant. Breaking a cipher isn't a matter of if, but when it will occur. It's highly unlikely that two different algorithms will be broken simultaneously. So, if two are supported, and one gets broken, we can use the other until we get a new one in place. Having the ability within the protocol specification to support such an event, having algorithm agility, is essential. Mandating two algorithms provides both a redundancy, and a mechanism for enacting that redundancy when needed. Further, the mechanism MUST describe the generic interface for new cryptographic algorithms to be used, so that implementers can use algorithms other than those specified, and so that new algorithms may be specified and supported in the future.
4. Secure use of simple PSKs, offering both operational convenience as well as building something of a fence around stupidity, MUST be specified.
5. Inter-connection replay protection. Packets captured from one session MUST NOT be able to be re-sent and accepted during a later session. In OSPF parlance, or other non TCP based

protocols, two routers have a session up if they are able to exchange protocol packets. In OSPF, a session between two routers is called an adjacency only if the neighbor FSM is in ExStart or a higher state. An OSPF session between two routers must go through two main stages of two-way connectivity and LSDB synchronization before an OSPF adjacency is fully established.

6. Intra-connection replay protection. Packets captured during a session MUST NOT be able to be re-sent and accepted during that same session, to deal with long-lived connections. The design teams may thus want to provide a sufficiently large sequence number space for providing intra-connection replay protection. Additionally, replay mechanisms MUST work correctly even in the presence of Routing Protocol packet prioritization by the router.
7. A change of security parameters REQUIRES, and even forces, a change of session traffic keys
8. Intra-connection re-keying which occurs without a break or interruption to the current peering session, and, if possible, without data loss, MUST be specified. Keys need to be changed periodically, for operational privacy (e.g. when an administrator who had access to the keys leaves an organization) and for entropy purposes, and a re-keying mechanism enables the deployers to execute the change without productivity loss.
9. Efficient re-keying SHOULD be provided. The specification SHOULD support rekeying during a connection without the need to expend undue computational resources. In particular, the specification SHOULD avoid the need to try/compute multiple keys on a given packet.
10. Prevent DoS attacks as those described as in-scope in the threats section Section 2.1 above.
11. Default mechanisms and algorithms specified and defined are REQUIRED for all implementations.
12. For backward compatibility reasons manual keying MUST be supported.
13. Architecture of the specification SHOULD consider and allow for future use of a KMP.
14. The authentication mechanism in the Routing Protocol MUST be decoupled from the key management system used. It MUST be obvious how the keying material was obtained, and the process

for obtaining the keying material MUST exist outside of the Routing Protocol. This will allow for the various key generation methods, like manual keys and KMPs, to be used with the same Routing Protocol mechanism.

15. Convergence times of the Routing Protocols SHOULD NOT be materially affected. Materially here is defined as anything greater than a 5% convergence time increase. Note that convergence is different than boot time. Also note that convergence time has a lot to do with the speed of processors used on individual routing peers, and this processing power increases by Moore's law over time, meaning that the same route calculations and table population routines will decrease in duration over time. Therefore, this requirement should be considered only in terms of total number of messages that must be exchanged, and less for the computational intensity of processing any one message. Alternatively this can be simplified by saying that the new mechanisms should only result in a minimal increase in the number of routing protocol messages passed between the peers.
16. The changes or addition of security mechanisms SHOULD NOT cause a refresh of route updates or cause additional route updates to be generated.
17. Router implementations provide prioritized treatment to certain protocol packets. For example, OSPF HELLO messages and ACKs are prioritized for processing above other OSPF packets. The authentication mechanism SHOULD NOT interfere with the ability to observe and enforce such prioritization. Any effect on such priority mechanisms MUST be explicitly documented and justified. Replay mechanisms provided by the routing protocols MUST work even if certain protocol packets are offered prioritized treatment.
18. The authentication mechanism does not provide message confidentiality, but SHOULD NOT preclude the possibility of confidentiality support being added in the future.
19. Routing protocols MUST only send minimal information regarding the authentication mechanisms and the parameters in its protocol packets to avoid exposing the information to parties on the path.
20. In most routing protocols (OSPF, ISIS, BFD, RIP, etc), all speakers share the same key on a broadcast segment. Possession of the key itself is used for identity validation and no other identity check is used. This opens a window for an attack where

the sender can masquerade as some other neighbor. Routing protocols SHOULD thus use some other information besides the key to validate a neighbor. One could look at [I-D.ietf-opsec-routing-protocols-crypto-issues] for details on such attacks.

21. Routing protocols that rely on the IP header (or information beyond the routing protocol payload) to identify the neighbor which originated the packet must either protect the IP header or provide some other means to identify the neighbor. [I-D.ietf-opsec-routing-protocols-crypto-issues] describes some attacks that are based on this.
22. The new security and authentication mechanisms MUST support incremental deployment. It will not be feasible to deploy a new Routing Protocol authentication mechanism throughout the network instantaneously. It also may not be possible to deploy such a mechanism to all routers in a large autonomous system (AS) at one time. Proposed solutions SHOULD support an incremental deployment method that provides some benefit for those who participate. Because of this, there are several requirements that any proposed KARP mechanism should consider.
  - A. The Routing Protocol security mechanism MUST enable each router to configure use of the security mechanism on a per-peer basis where the communication is one-on-one.
  - B. The new KARP mechanism MUST provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment. Message formatting in a fully secured environment MAY be handled in a non-backward compatible fashion though care must be taken to ensure that routing protocol packets can traverse intermediate routers which don't support the new format.
  - C. In an environment where both secured and non-secured systems are interoperating a mechanism MUST exist for secured systems to identify whether an originator intended the information to be secured.
  - D. In an environment where secured service is in the process of being deployed a mechanism MUST exist to support a transition free of service interruption (caused by the deployment per se).



23. The introduction of mechanisms to improve routing authentication and security may increase the processing performed by a router. Since most of the currently deployed routers do not have hardware to accelerate cryptographic operations, these operations could impose a significant processing burden under some circumstances. Thus proposed solutions should be evaluated carefully with regard to the processing burden they may impose, since deployment may be impeded if network operators perceive that a solution will impose a processing burden which either provokes substantial capital expense, or threatens to destabilize routers.
24. Given the high number of routers that would require the new authentication mechanisms in a typical ISP deployment, solutions can increase their appeal by minimizing the burden imposed on all routers in favor of confining significant work loads to a relatively small number of devices. Optional features or increased assurance that provokes more pervasive processing load MAY be made available for deployments where the additional resources are economically justifiable.
25. The new authentication and security mechanisms should not rely on systems external to the routing system (the equipment that is performing forwarding). In order to ensure the rapid initialization and/or return to service of failed nodes it is important to reduce reliance on these external systems to the greatest extent possible. Therefore, proposed solutions SHOULD NOT require connections to external systems, beyond those directly involved in peering relationships, in order to return to full service. It is however acceptable for the proposed solutions to require post initialization synchronization with external systems in order to fully synchronize the security information.

#### 4. Security Considerations

This document is mostly about security considerations for the KARP efforts, both threats and requirements for solving those threats. More detailed security considerations were placed in the Security Considerations section of the KARP Design Guide [I-D.ietf-karp-design-guide] document.

## 5. IANA Considerations

This document has no actions for IANA.

## 6. Acknowledgements

The majority of the text for version -00 of this document was taken from draft-lebovitz-karp-roadmap, authored by Gregory Lebovitz.

## 7. Change History (RFC Editor: Delete Before Publishing)

[NOTE TO RFC EDITOR: this section for use during I-D stage only.  
Please remove before publishing as RFC.]

kmart-00-00 original rough rough rough draft for review by routing  
and security AD's

karp-threats-reqs-00-

o removed all the portions that will be covered in either  
draft-ietf-karp-design-guide or draft-ietf-karp-framework

## 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.
- [RFC4593] Barbir, A., Murphy, S., and Y. Yang, "Generic Threats to Routing Protocols", RFC 4593, October 2006.
- [RFC4948] Andersson, L., Davies, E., and L. Zhang, "Report from the IAB workshop on Unwanted Traffic March 9-10, 2006", RFC 4948, August 2007.

### 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-01 (work in progress),  
September 2010.
- [I-D.ietf-karp-framework]  
Atwood, W. and G. Lebovitz, "Framework for Cryptographic Authentication of Routing Protocol Packets on the Wire",  
draft-ietf-karp-framework-00 (work in progress),  
February 2010.
- [I-D.ietf-opsec-routing-protocols-crypto-issues]  
Jaeggli, J., Hares, S., Bhatia, M., Manral, V., and R. White, "Issues with existing Cryptographic Protection Methods for Routing Protocols",  
draft-ietf-opsec-routing-protocols-crypto-issues-07 (work in progress), August 2010.
- [ISR2008] McPherson, D. and C. Labovitz, "Worldwide Infrastructure Security Report", October 2008,  
<[http://www.arbornetworks.com/dmdocuments/ISR2008\\_US.pdf](http://www.arbornetworks.com/dmdocuments/ISR2008_US.pdf)>.
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, November 1998.
- [RFC3562] Leech, M., "Key Management Considerations for the TCP MD5

Signature Option", RFC 3562, July 2003.

- [RFC3618] Fenner, B. and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC 3618, October 2003.
- [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.
- [RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, 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.
- [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.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [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.

- [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.



Authors' Addresses

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

Email: [gregory.ietf@gmail.com](mailto:gregory.ietf@gmail.com)

Manav Bhatia  
Alcatel-Lucent  
Bangalore,  
India

Phone:  
Email: [manav.bhatia@alcatel-lucent.com](mailto:manav.bhatia@alcatel-lucent.com)

Russ White  
Cisco Systems  
USA

Phone:  
Email: [russ@cisco.com](mailto:russ@cisco.com)



KARP Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 21, 2011

X. Liang, Ed.  
H. Wang  
Y. Wei  
ZTE Corporation  
C. Wan  
Southeast University  
October 18, 2010

Automated Security Association Management for Routing Protocols  
draft-liang-karp-auto-sa-management-rp-00

Abstract

This document discusses automated security association (SA) management for routing protocols, which includes SA establishment and SA maintenance for routing protocols, and also discusses two candidate solutions of automated SA management that are based on IKEv2 and ISAKMP respectively.

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, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Conventions Used in This Document . . . . .	3
2. Automated SA Management for Routing Protocols . . . . .	3
2.1. RP SA Attributes/Parameters/Components/Contents . . . . .	4
2.2. RP SA Format . . . . .	4
2.3. Secure Channel . . . . .	4
2.4. RP SA Negotiation . . . . .	4
2.5. RP SA Creation/Generation . . . . .	5
2.6. RP SA Distribution and Delivery . . . . .	5
2.7. RP SA Deletion, Update, and Rekeying . . . . .	5
3. Candidate Solutions . . . . .	6
3.1. IKEv2 Extensions . . . . .	6
3.1.1. Extending SA Payload to Support PR SA Management . . . . .	6
3.1.2. Adding New Payload to Support RP SA Management . . . . .	8
3.1.3. Adding New Exchange Type to Support RP SA Management . . . . .	9
3.2. ISAKMP Extensions . . . . .	9
4. Security considerations . . . . .	10
5. IANA Considerations . . . . .	10
6. Acknowledgement . . . . .	10
7. References . . . . .	10
7.1. Normative references . . . . .	10
7.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

The draft [I-D.wei-karp-analysis-rp-sa] has shown the diversity of SA of routing protocols, and then one problem arises -- how to manage these diverse SAs of routing protocols automatically? An automated key management protocol (KMP) is desired to manage SAs of routing protocols. When considering KMP design for routing protocols, automated SA management is the main function that KMP should provide for routing protocols. In some sense and to some extent, KMP for routing protocols is automated SA management for routing protocols essentially.

The following sections discuss automated SA management for routing protocols based on [I-D.wei-karp-analysis-rp-sa] , and also discuss the candidate solutions based on IKEv2 [RFC4306] [RFC4307] [RFC4718] [RFC5996] and ISAKMP [RFC2408], respectively.

### 1.1. 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 RFC2119 [RFC2119].

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

## 2. Automated SA Management for Routing Protocols

An SA is a simplex "connection" that affords security services to the traffic carried by it [RFC4301]. In the routing protocol context, an SA of routing protocol is a set of cryptographic algorithms and other security parameters to be used to protect routing protocol message, i.e., to perform routing protocol message authentication and integrity protection (see [I-D.ietf-karp-framework] and [I-D.ietf-karp-design-guide]). The document uses RP SA to represent routing protocol security association, i.e., security association for routing protocols, and uses RP SAs to represent its plural form. Since manual SA management is vulnerable to a variety of security issues as discussed in [I-D.ietf-karp-threats-reqs], automated SA management is the desired solution to be considered.

To achieve automated SA management for routing protocols, the following items to be considered are identified and defined.

### 2.1. RP SA Attributes/Parameters/Components/Contents

In this document, the four words -- attribute, parameter, component and content, refer to the same meaning to RP SA, and among them, attribute is preferred in RFC 2408 [RFC2408]. The attributes of RP SA is different from that of IPsec [RFC4301], since RP SA is dedicated to routing protocols. The draft [I-D.wei-karp-analysis-rp-sa] has identified the attributes of RP SA, i.e., Key ID, authentication algorithm, authentication key, life time, sequence number, etc. In order to distinguish different RP SAs, routing protocol ID, message type ID of routing protocol, which is related to message transaction type, and etc., may also be taken into consideration. Attention should be paid to the direction attribute of SA. In IPsec, the SA is simplex, while in routing protocols, the direction attribute of SA is not defined, and it seems that the direction attribute of RP SA depends on how to use the traffic key by routing protocol message. If the communicating peers use different traffic keys in both directions, RP SA is simplex; if the communicating peers use the same traffic key in both directions, RP SA is duplex; and vice versa.

### 2.2. RP SA Format

An agreed on RP SA format should be formed to achieve interoperation between/among communicating peers. It is about how RP SA to be constructed, e.g., the header format and the payload format of RP SA. The RP SA format could support as much as possible, if not all, the RP SA attributes.

### 2.3. Secure Channel

RP SA is shared information between/among involving peers, and the attributes of RP SA that will be transferred should be protected via secure channel. Secure channel is needed to be established before RP SA is involved in the communication between/among peers, for example, RP SA negotiation, RP SA distribution, etc. Generally, the secure channel protection provides encryption and message authentication and anti-replay for RP SA.

### 2.4. RP SA Negotiation

In RFC 2408 [RFC2408], the need for negotiation was stated, and the main reason for SA negotiation is the diverse security requirements and security services of different networks. To achieve common supported security functionality for interoperation and cooperation between/among communicating peers for routing protocols, RP SA negotiation is desired in automated SA management for routing protocols. In other words, the diverse configurations and security

functionalities of routers and their deployed routing protocols are the main reason that makes RP SA negotiation desirable, and automation requirement of SA management makes RP SA negotiation necessary. The RP SA negotiation procedures and payloads that will be transferred should be identified and defined in automated SA management for routing protocols. What attributes of RP SA will be negotiated is dependent on specific routing protocol and its message transaction type, etc.

## 2.5. RP SA Creation/Generation

Creation and generation of RP SA have the same meaning in this document. If the RP SA attributes and format are defined, and the negotiation of needed attributes of RP SA is finished successfully, then automated SA management takes the task of RP SA creation according to corresponding RP SA attributes, format, and specific routing protocol, etc., obtained via negotiation. In this creation process, one or more databases may be involved, e.g., cryptography algorithms database, cryptography functions database, and key databases, or one database composed of these three things.

## 2.6. RP SA Distribution and Delivery

This may involve two scenarios, that is, RP SA is distributed to other peers, and RP SA is delivered to routing protocol served by KMP. The former scenario may be a group SA for routing protocol that is distributed to all the group member peers, and this kind of distribution should be under the protection of secure channel. The later scenario may be that the RP SA is delivered to the corresponding routing protocol directly, or to a key store, which then will serve the corresponding routing protocol with the RP SA.

## 2.7. RP SA Deletion, Update, and Rekeying

Since RP SA is time relevant, and some routing protocol messages are updated periodically, RP SA deletion, update, and rekeying are very important. The life time or life cycle attribute of RP SA is a key factor that effects RP SA deletion, update, and rekeying. If life time is expired, then the RP SA can be deleted or updated or rekeyed. Deletion means to remove RP SA from corresponding store, update means to assign new values to attributes of RP SA, which may be by the means of negotiation, and rekeying means to create a new RP SA totally, which may involve using different cryptography algorithms, different key, etc. Attention should be paid to adjacencies bouncing problem during RP SA deletion, update, and rekeying. Roughly speaking though, update and rekeying indicate the same meaning, and both involve deletion of old RP SA.

### 3. Candidate Solutions

According to the design spirit of KMP for routing protocols [I-D.ietf-karp-design-guide], it is best to reuse existing technology/mechanisms to solve problems. In this sense, IKEv2 and ISAKMP are good candidates for automated SA management for routing protocols, since they are existing and mature protocols for key management that is evolving along time, and they provide some flexible and extendable mechanism that can be exploited. Taking into consideration the above items discussed in section 2, IKEv2 and ISAKMP can be extended to support automated SA management for routing protocols. The details of extending IKEv2 is discussed in section 3.1, and the details of extending ISAKMP is discussed in section 3.2. Both extensions are focusing on peer-to-peer communication at the time being, and the extension to support group RP SA will be discussed in later update version of the document.

#### 3.1. IKEv2 Extensions

IKEv2 is dedicated to IPsec, specifically ESP [RFC4303] [RFC4305] [RFC4835] and/or AH [RFC2402] [RFC4302] [RFC4305] [RFC4835], to establish and maintain SAs for two communicating peers, and cannot be applied to routing protocols as automated SA management for routing protocols directly. IKEv2 can be extended and made adaptive to routing protocols. Specifically, IKEv2 can be extended to support attributes of RP SA, to provide unified format for RP SA, to establish secure channel for RP SA negotiation and distribution if applicable, to negotiate RP SA between/among communicating peers, to create RP SA, to distribute RP SA if applicable, and to update and rekey RP SA.

Three ways of IKEv2 extension to support RP SA management are considered, that is, extending SA payload, adding new payload, and adding new exchange type.

##### 3.1.1. Extending SA Payload to Support PR SA Management

The SA payload of IKEv2 has proposals substructure which has transforms substructure, which has Transform Attributes in turn to describe key length for encryption algorithm, etc., to support the SA for ESP and/or AH of IPsec. The following changes to SA payload can be made to adapt for RP SA.

- o Extend Protocol ID field of proposal substructure to include routing protocols, and take the ID values from those reserved to IANA, i.e., 4-200, e.g., assign OSPFv2 [RFC2328] the value 3.



- o Match SPI field in proposal substructure to Key ID of RP SA, and extend SPI Size (in octet) field of proposal substructure to include routing protocols, e.g., assign OSPFv2 the value 1, since the length of Key ID is 8 bits.
- o Extend Transform Type 3 in transform substructure to be also used in routing protocols, and extend Transform ID of Transform Type 3 to include authentication algorithms used by routing protocols, and take the ID values from those reserved to IANA, i.e., 6-1023, e.g., assign AUTH\_HMAC\_SHA\_256, which stands for HMAC-SHA-256 [RFC5709] as authentication algorithm, the value 8.
- o Extend Attribute Type in transform substructure to include key length and life time attributes for routing protocols. As to the Attribute Type 14 Key Length (in bits), it is defined for encryption algorithm only, and can be extended to use in authentication algorithm in case of negotiation of key length of authentication algorithm. As to the life time attributes, they can be defined as a new Attribute Type and assigned the Attribute Type values from those reserved to IANA, i.e., 18-16383, e.g., Start Time can be defined as one new Attribute Type, and its Attribute Type Value can be assigned 18.

The above extensions together can support attributes and format of RP SA. The extended SA can be denoted as SAe, where !oe!+/- means extension.

The procedures to establish secure channel and negotiate RP SA can be as follows:

Initiator	Responder
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> HDR, SAi, KEi, Ni --&gt; </div> <div style="width: 45%; text-align: right;"> &lt;-- HDR, SAR, KEr, Nr, [CERTREQ] </div> </div>	

The above IKE\_SA\_INIT exchange results in IKE\_SA, which will be used to encrypt and authenticate the subsequent exchange content in brace following SK, hence, the secure channel is established.

Initiator	Responder
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAe, TSi, TSr} --&gt; </div> <div style="width: 45%; text-align: right;"> &lt;-- HDR, SK {IDr, [CERT,] AUTH, SAe, TSi, TSr} </div> </div>	

In the above IKE\_AUTH exchange, Initiator offers a list of proposals in the extended payload SAe, and the Responder chooses one, put it in the SAe, and send to the Initiator. Then Initiator hands the SA to routing protocol being served. Hence the RP SA negotiation is finished.

As to the Authentication Key in RP SA, it can be calculated by Pseudo-random Function (PRF) with the inputs of Nonce from both peers and from KE (Key Exchange) payload that will be used in Diffie-Hellman exchange.

### 3.1.2. Adding New Payload to Support RP SA Management

Alternatively, a new payload to support attributes and format of RP SA can be created in IKEv2, for example, it may be named SARP, which stands for security association of routing protocol, and take the Next Payload Type value, say 49, from the reserved to IANA value 49-127. The Next Payload Type of the original SA payload in IKEv2 is 33.

The structure of SARP can be similar to that of SA, with the following differences:

- o Add Length of Life Time field and the Life Time field in the Proposal Substructure.
- o Replace SPI Size field with Length of Key ID field, and Key ID field substitutes the SPI field.
- o Define Transform Type to include Pseudo-random Function (PRF), Integrity Algorithm (INTEG), and Sequence Numbers (SN), etc., which are used by routing protocols.
- o Define Transform ID for PRF that will be used by routing protocols.
- o Define Transform ID for INTEG that will be used by routing protocols.
- o Define Key Length of PRF or/and INTEG that will be used by routing protocols in case the length of key can be negotiated.

As to the Authentication Key in RP SA, it can be calculated by PRF with the inputs from Nonce payload and KE payload of both peers.

The procedures to negotiate RP SA using the above new payload SARP are the same as that of extending SA payload above, with SAe payload substituted by SARP payload.

The benefit gained by adding new payload to support RP SA management is a bit fast processing for automated SA management.

### 3.1.3. Adding New Exchange Type to Support RP SA Management

New exchange type can also be defined to do RP SA negotiation. For example, IKE\_RP\_AUTH exchange can be defined dedicated to RP SA negotiation, and take the value, say 38, from the reserved to IANA value 38-239, as its Exchange Type value. Note that the Exchange Type of IKE\_AUTH is 35. The payloads involved in IKE\_RP\_AUTH exchange may be similar with that in IKE\_AUTH, and the major difference may be the SA payload. The extended SA payload SAe or the new added payload SARP can be used in this exchange, replacing the SA payload in IKE\_AUTH exchange. In this way, i.e., with a dedicated exchange, the negotiation of RP SA can be speeded up to some extent.

Extending SA payload and adding new payload can be used independently or even hybridly if applicable, and can also be combined with the new added exchange type defined in section 3.1.3, to finish the RP SA negotiation, creation, and distribution if applicable.

## 3.2. ISAKMP Extensions

ISAKMP [RFC2408] defines procedures and packet formats to establish, negotiate, modify and delete SA. SAs contain all the information required for execution of various network security services, such as the IP layer services (such as header authentication and payload encapsulation), transport or application layer services, or self-protection of negotiation traffic. ISAKMP is intended to support the negotiation of SAs for security protocols at all layers of the network stack. However, ISAKMP provides a framework but not define them [RFC2409]. This section tries to discuss ISAKMP extensions and definitions to serve routing protocol.

The following changes can be made in ISAKMP to support automated SA management for routing protocols:

- o Extend DOI (Domain of Interpretation) field of SA payload to indicate the subsequent payloads are used to negotiate RP SA, e.g., define a new DOI and assign it the value 3 if applicable, and denote it as SARPDOI, which means DOI for routing protocol security association.
- o Extend Security Protocol Identifiers of proposal for RP SA, e.g., define OSPFv2 as PROTO\_OSPFv2 and assign it the value 5 if applicable. In this way, the RP SA established for OSPFv2 is shown.

- o Match SPI field in proposal substructure to Key ID of RP SA, and extend SPI Size (in octet) field of proposal substructure to include routing protocols, e.g., assign OSPFv2 the value 1, since the length of Key ID is 8 bits.
- o Extend Transform Identifiers to define transform for specific routing protocol, e.g., define new transforms OSPFv2\_MD5 and OSPFv2\_SHA for OSPFv2, which means OSPFv2 using MD5 and SHA-1 as authentication algorithm respectively, and assign them the value 2 and 3 respectively.
- o Extend Attribute Type to support attributes of RP SA, e.g., define Start Time for life time of OSPFv2, and assign it the value 4.

Alternatively, DOI (Domain of Interpretation ) can be extended in this way -- extend DOI field of SA payload to indicate the subsequent payloads will be used to negotiate RP SA, e.g., define SPFv2 DOI and assign it the value 3.

#### 4. Security considerations

To be completed.

#### 5. IANA Considerations

To be completed.

#### 6. Acknowledgement

To be completed.

#### 7. References

##### 7.1. Normative references

- [RFC2408] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

- [RFC4307] Schiller, J., "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)", RFC 4307, December 2005.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

## 7.2. 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-01 (work in progress), September 2010.
- [I-D.ietf-karp-framework]  
Atwood, W. and G. Lebovitz, "Framework for Cryptographic Authentication of Routing Protocol Packets on the Wire", draft-ietf-karp-framework-00 (work in progress), February 2010.
- [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.
- [I-D.wei-karp-analysis-rp-sa]  
Wei, Y., Wang, H., and X. Liang, "Analysis of Security Association for Current Routing Protocol", draft-wei-karp-analysis-rp-sa-00 (work in progress), July 2010.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.
- [RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.
- [RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC4305] Eastlake, D., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4305, December 2005.
- [RFC4718] Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines", RFC 4718, October 2006.
- [RFC4835] Manral, V., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4835, April 2007.
- [RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, October 2009.

#### Authors' Addresses

Xiaoping Liang (editor)  
ZTE Corporation  
No. 6, HuashenDa Road, Yuhuatai District  
Nanjing, Jiangsu 210012  
China

Phone: +86 25 52877610  
Email: liang.xiaoping@zte.com.cn

Hongyan Wang  
ZTE Corporation  
No. 6, HuashenDa Road, Yuhuatai District  
Nanjing, Jiangsu 210012  
China

Phone: +86 25 52877993  
Email: wang.hongyan4@zte.com.cn

Yinxing Wei  
ZTE Corporation  
No. 6, HuashenDa Road, Yuhuatai District  
Nanjing, Jiangsu 210012  
China

Phone: +86 25 52877993  
Email: wei.yinxing@zte.com.cn

Changsheng Wan  
Southeast University  
No. 2, Sipailou, Radio department, Southeast University  
Nanjing, Jiangsu 210096  
China

Phone: +86 25 83795822-866  
Email: wanchangsheng@seu.edu.cn

