KARP Working Group
Internet Draft
Intended status: Informational
Expires: June 16, 2012

G. Lebovitz
M. Bhatia
Alcatel-Lucent

December 13, 2011

Keying and Authentication for Routing Protocols (KARP) Design Guidelines

draft-ietf-karp-design-guide-10.txt

Status of this Memo

Copyright Notice

Abstract

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

Conventions used in this document

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

Table of Contents

1. Introduction....3
2. Categorizing Routing Protocols....5
   2.1. Category: Message Transaction Type....5
   2.2. Category: Peer vs Group Keying....6
3. Consider the future existence of a Key Management Protocol....7
   3.1. Consider Asymmetric Keys....7
   3.2. Cryptographic Keys Life Cycle....8
4. RoadMap....9
   4.1. Work Phases on any Particular Protocol....9
   4.2. Work Items Per Routing Protocol....11
5. Routing Protocols in Categories....13
6. Supporting Incremental Deployment....17
7. Denial of Service Attacks....17
8. Gap Analysis....18
9. Security Considerations....21
   9.1. Use Strong Keys....21
   9.2. Internal vs. External Operation....22
   9.3. Unique versus Shared Keys....23
   9.4. Key Exchange Mechanism....24
10. Acknowledgments....27
11. IANA Considerations....27
12. References....27
   12.1. Normative References....27
   12.2. Informative References....27

1. Introduction

In March 2006 the Internet Architecture Board (IAB) held a workshop on the topic of "Unwanted Internet Traffic". The report from that workshop is documented in RFC 4948 [RFC4948]. Section 8.1 of that document states that "A simple risk analysis would suggest that an ideal attack target of minimal cost but maximal disruption is the core routing infrastructure." Section 8.2 calls for "[t]ightening the security of the core routing infrastructure." Four main steps were identified for that tightening:

- Increased security mechanisms and practices for operating routers.

- Cleaning up the Internet Routing Registry repository [IRR], and securing both the database and the access, so that it can be used for routing verifications.

- Specifications for cryptographic validation of routing message content.

- Securing the routing protocols' packets on the wire

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

This document addresses the last bullet, securing the packets on the wire of the routing protocol exchanges. Thus, it is concerned with guidelines for describing issues and techniques for protecting the messages between directly communicating peers. This may overlap with, but is strongly distinct from, protection designed to ensure that routing information is properly authorized relative to sources of this information. Such authorizations are provided by other mechanisms and are outside the scope of this document and work that relies on it.

This document uses the terminology "on the wire" to talk about the information used by routing systems. This term is widely used in IETF RFCs, but is used in several different ways. In this document, it is used to refer both to information exchanged between routing protocol instances, and to underlying protocols that may also need to be protected in specific circumstances. Other documents that will analyze individual protocols will need to indicate how they use the term "on the wire".

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

KARP will focus on an abstraction for keying information that describes the interface between routing protocols, operators and automated key management. Conceptually when routing protocols send or receive messages they will look up the key to use in this abstract key table. Conceptually, there will be an interface for a routing protocol to make requests of automated key management when it is being used; when keys become available they will be made available in the key table. There is no requirement that this abstraction be used for implementation; the abstraction serves the needs of standardization and management. Specifically as part of the KARP work plan:

1) KARP will design the key table abstraction, the interface between key management protocols and routing protocols and possibly security protocols at other layers.

2) For each routing protocol, KARP will define the mapping between how the protocol represents key material and the protocol independent key table abstraction. When routing protocols share a common mechanism for authentication, such as the TCP Authentication Option, the same mapping is likely to be reused between protocols. An implementation may be able to move much of the keying logic into code related to this shared authentication primitive rather than code specific to routing protocols.

3) When designing automated key management for both symmetric keys and group keys, we will only use the abstractions designed in point 1 above to communicate between automated key management and routing protocols.

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

2. Categorizing Routing Protocols

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

2.1. Category: Message Transaction Type

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

One-to-One

One peer router directly and intentionally delivers a route update specifically to one other peer router. Examples are BGP [RFC4271], LDP [RFC5036], BFD [RFC5880] and RSVP-TE [RFC3209] [RFC3473] [RFC4726] [RFC5151]. Point-to-point modes of both IS-IS [RFC1195] and OSPF [RFC2328], when sent over both traditional point-to-point links and when using multi-access layers, may both also fall into this category.

One-to-Many

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

Multicast

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

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

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

2.2. Category: Peer vs Group Keying

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

Peer keying

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

Group Keying

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

3. Consider the future existence of a Key Management Protocol

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

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

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

3.1. Consider Asymmetric Keys

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

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

Many routers have the ability to be remotely managed using Secure Shell (SSH) Protocol [RFC4252] and [RFC4253]. As such, routers will also have the ability to generate and store an asymmetric key pair, because this is the common authentication

method employed by SSH when an administrator connects to a router for management sessions.

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

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

3.2. Cryptographic Keys Life Cycle

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

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

from compromised keys.

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

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

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

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

4. RoadMap

4.1. Work Phases on any Particular Protocol

It is believed that improving security for any routing protocol will be a two phase process. The first phase would be to modify routing protocols to support modern cryptography algorithms and key agility. The second phase would be to design and move to an automated key management mechanism. This is like a crawl, walk and run process. In order for operators to accept these phases, we believe that the key management protocol should be clearly separated from the routing transport. This would mean that the

routing transport subsystem is oblivious to how the keys are

derived, exchanged and downloaded as long as there is something that it can use. It is like having a routing protocol configuration switch that requests the security module for the "KARP security parameters" so that it can refer to some module written, maintained, and operated by security experts and insert those parameters in the routing exchange.

The desired end state for the KARP work contains several items. First, the people desiring to deploy securely authenticated and integrity validated packets between routing peers have the tools specified, implemented and shipping in order to deploy. These tools should be fairly simple to implement, and not more complex than the security mechanisms to which the operators are already accustomed. (Examples of security mechanisms to which router operators are accustomed include: the use of asymmetric keys for authentication in SSH for router configuration, the use of pre-shared keys (PSKs) in TCP MD5 for BGP protection, the use of self-signed certificates for HTTPS access to device Web-based user interfaces, the use of strongly constructed passwords and/or identity tokens for user identification when logging into routers and management systems.) While the tools that we intend to specify may not be able to stop a deployment from using "foobar" as an input key for every device across their entire routing domain, we intend to make a solid, modern security system that is not too much more difficult than that. In other words, simplicity and deployability are keys to success. The Routing Protocols will specify modern cryptographic algorithms and security mechanisms. Routing peers will be able to employ unique, pair-wise keys per peering instance, with reasonable key lifetimes, and updating those keys on a regular basis will be operationally easy, causing no service interruption.

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

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

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

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

4.2. Work Items Per Routing Protocol

Each Routing Protocol will have a team (the [Routing_Protocol]-

KARP team) working on incrementally improving the security of a

Routing Protocol. These teams will have the following main work items:

PHASE 1:

Characterize the RP

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

Define Optimal State

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

Gap Analysis

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

Transition and Deployment Considerations

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

Define, Assign, Design

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

PHASE 2:

KMP Analysis

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

Gap Analysis

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

Define, Assign, Design

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

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

5. Routing Protocols in Categories

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

BGP, LDP, PCEP and MSDP

These Routing Protocols fall into the category of the one-to-one peering messages, and will use peer keying protocols. BGP [RFC4271], PCEP [RFC5440] and MSDP [RFC3618] messages are transmitted over TCP, while LDP [RFC5036] uses both UDP and TCP. A team will work on one mechanism to cover these TCP unicast protocols. Much of the work on the Routing Protocol update for its existing authentication mechanism

has already occurred in the TCPM Working Group, on the TCP-

AO [RFC5925] document, as well as its cryptography-helper document, TCP-AO-CRYPTO [RFC5926]. However, TCP-AO cannot be used for discovery exchanges carried in LDP as those are carried over UDP. A separate team might want to look at LDP. Another exception is the mode where LDP is used directly on the LAN. The work for this may go into the Group keying category (along with OSPF) as mentioned below.

OSPF, IS-IS, and RIP

The Routing Protocols that fall into the category Group Keying *with one-to-many peering) includes OSPF [RFC2328], IS-IS [RFC1195] and RIP [RFC2453]. Not surprisingly, all these routing protocols have two other things in common. First, they are run on a combination of the OSI datalink layer 2, and the OSI network layer 3. By this we mean that they have a component of how the routing protocol works which is specified in Layer 2 as well as in Layer 3. Second, they are all internal gateway protocols(IGPs). The keying mechanisms will be much more complicated to define for these than for a one-to-one messaging protocol.

BFD

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

RSVP and RSVP-TE

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

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

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

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

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

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

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

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

with Fast Reroute, since the merge point (MP) and point of local repair (PLR) should also share the key.

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

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

PIM-SM and PIM-DM

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

One possible solution proposes a region-wide "master" key server (possibly replicated), and one "local" key server per speaking router. There is no issue with propagating the messages outside the link, because link-local messages, by definition, are not forwarded. This solution is offered only as an example of how work may progress; further discussion should occur in this work team. Specification of a link-local protection mechanism for PIM-SM is defined in [RFC4601], and this mechanism has been updated in PIM-SM-

LINKLOCAL [RFC5796]. However, the KMP part is completely

unspecified, and will require work outside the expertise of the PIM working group to accomplish, another example of why this roadmap is being created.

6. Supporting Incremental Deployment

It is imperative that the new authentication and security mechanisms defined support incremental deployment, as it is not feasible to deploy a new routing protocol authentication mechanism throughout the network instantaneously. One of the goals of KARP WG is to add incremental security to existing mechanisms rather than replacing them. Delivering better deployable solutions to which vendors and operators can migrate to is more important than getting a perfect security solution. It may also not be possible to deploy such a mechanism to all routers in a large AS at one time. This means that the designers must work on this aspect of authentication mechanism for the routing protocol that they are working on. The mechanisms must provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment.

7. Denial of Service Attacks

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

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

validation.

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

Particular care needs to be paid to design of automated key management schemes. It is often desirable to force a party attempting to authenticate to do work and to maintain state until that work is done. That is, the initiator of the authentication should maintain the cost of any state required by the authentication for as long as possible. This also helps when an attacker send an overwhelming load of keying protocol initiations from bogus sources.

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

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

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

8. Gap Analysis

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

To start, design teams must review the "Threats and Requirements for Authentication of Routing Protocols" document [I-D.ietf-karp-threats-reqs]. This document contains detailed

authentication and integrity in general. Note that it will not contain all the authentication-related threats for any one routing protocol, or category of routing protocols. The design team must conduct a protocol-specific threat analysis to determine if threats beyond those in the [I-D.ietf-karp-threats-reqs] document arise in the context of the protocol (group), and to describe those threats.

The [I-D.ietf-karp-threats-reqs] document also contains many security requirements. Each routing protocol design team must walk through each section of the requirements and determine one by one how its protocol either does or does not relate to each requirement.
Examples include modern, strong cryptographic algorithms, with at least one such algorithm listed as a MUST; algorithm agility; secure use of simple PSKs; intra-connection replay protection; inter-connection replay protection, etc.

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

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

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

Care should also be taken to ensure that the routing protocol authentication scheme has algorithm agility (i.e., it is

capable of supporting algorithms other than its defaults).

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

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

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

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

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

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

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

9. Security Considerations

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

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

9.1. Use Strong Keys

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

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

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

However, whenever using manual keys, it is best to design a system where a given pre-shared key (PSK) will be used in a

KDF, mixed with connection-specific material, in order to

generate session unique -- and therefore peer-wise -- Traffic Keys. Doing so has the following advantages: the Traffic Keys used in the per-message authentication mechanism are peer-wise unique, it provides inter-connection replay protection, and, if the per-message authentication mechanism covers some connection counter, intra-connection replay protection.

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

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

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

9.2. Internal vs. External Operation

Design teams must consider whether the protocol is an internal routing protocol or an external one, i.e. does it primarily run between peers within a single domain of control or between two different domains of control? Some protocols may be used in both cases, internally and externally, and as such various modes of authentication operation may be required for the same protocol. While it is preferred that all routing exchanges run with the best security mechanisms enabled in all deployment contexts, this exhortation is greater for those protocols running on inter-domain point-to-point links, and greatest for those on shared access link layers with several different domains interchanging together, because the volume of attackers are greater from the outside. Note however that the consequences of internal attacks maybe no less severe -- in

fact they may be quite a bit more severe -- than an external

attack. An example of this internal versus external consideration is BGP which has both EBGP and IBGP modes. Another example is a multicast protocol where the neighbors are sometimes within a domain of control and sometimes at an inter-domain exchange point. In the case of PIM-SM running on an internal multi-access link, it would be acceptable to give up some security to get some convenience by using a group key among the peers on the link. On the other hand, in the case of PIM-SM running over a multi-access link at a public exchange point, operators may favor security over convenience by using unique pair-wise keys for every peer. Designers must consider both modes of operation and ensure the authentication mechanisms fit both.

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

9.3. Unique versus Shared Keys

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

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

One goal for designers is to create authentication and integrity mechanisms that are easy for operators to deploy and manage, and still use unique keys between peers (or small groups on multi-access links), and for different sessions among the same peers. Operators have the impression that they NEED one key shared across the network, when in fact they do not. What they need is the relative convenience they experience from

deploying cryptographic authentication with one key (or a few

keys), compared to the inconvenience they would experience if they deployed the same authentication mechanism using unique pairwise keys. An example is BGP Route Reflectors. Here operators often use the same authentication key between each client and the route reflector. The roadmaps defined from this guidance document should allow for unique keys to be used between each client and the peer, without sacrificing much convenience. Designers should strive to deliver peer-wise unique keying mechanisms with similar ease-of-deployment properties as today's one-key method.

Operators must understand the consequences of using the same key across many peers. One argument against using the same key is that if the same key that is used in multiple devices then a compromise of any one of the devices will expose the key. Also since the same key is supported on many devices this is known by many people which affects its distribution to all of the devices.

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

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

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

9.4. Key Exchange Mechanism

This section discusses the security and use case considerations for key exchange for routing protocols. Two options exist: an out-of-band mechanism or a KMP. An out-of-band mechanism involves operators configuring keys in the device through a configuration tool or management method (e.g., SNMP, NETCONF). A KMP is an automated protocol that exchanges key without operator intervention. KMPs can occur either in-band to the

routing protocol or out-of-band to the routing protocol (i.e., a different protocol).

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

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

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

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

There are several benefits of a peer-to-peer KMP versus centrally managed and distributing keys. It results in key(s) that are privately generated, and need not be recorded permanently anywhere. Since the traffic keys used in a particular connection are not a fixed part of a device configuration no security sensitive data exists anywhere else in the operator's systems which can be stolen, e.g. in the case

of a terminated or turned employee. If a server or other data

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

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

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

Within this constraint there are two approaches for a KMP:

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

The second approach is to define an In-band KMP extension for existing routing protocols putting the key management

mechanisms inside the protocol itself. In this case, the key

management messages would be carried within the routing protocol packets, resulting in very tight coupling between the routing protocols and the key management protocol.

10. Acknowledgments

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

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

11. IANA Considerations

This document places no requests to IANA.

12. References

12.1. Normative References

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

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

12.2. Informative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Author's Addresses

Gregory M. Lebovitz
California
USA 95003

Phone:
Email: gregory.ietf@gmail.com

Manav Bhatia
Alcatel-Lucent
Bangalore
India

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