

INTERNET DRAFT
Intended Status: Informational

T. Polk
NIST
R. Housley
Vigil Security
November 8, 2010

Expires: May 12, 2011

Routing Authentication Using A Database of Long-Lived Cryptographic Keys
[draft-polk-saag-rtg-auth-keytable-05.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>

Abstract

This document describes the application of a database of long-lived cryptographic keys to establish session-specific cryptographic keys to support authentication services in routing protocols. Keys may be established between two peers for pair-wise communications, or between groups of peers for multicast traffic.

INTERNET DRAFT

November 8, 2010

Table of Contents

1	Introduction	2
1.1	Terminology	2
2	Architecture and Design	3
3	Pair-wise Application	3
4	Identifier Mapping	5
4.1	Selected Range Reservation	6
4.2	Protocol Specific Mapping Tables	6
5	Database Maintenance	6
6	Worked Examples	6
6.1	Worked Example: TCP-AO	7
6.1.1	Setup	7
6.1.2	Protocol Operation: Xp Initiates a Connection	8
6.1.3	Protocol Operation: Yp Initiates a Connection	9
6.2	Worked Example: IS-IS	9
6.2.1	Setup	10
6.2.2	Protocol Operations	14
6.2.2.1	Sending a Hello Message	14
6.2.2.2	Receiving a Hello Message	15
6.2.2.3	Generating a Link State PDU	15
6.2.2.4	Receiving a Link State PDU	16
6.2.2.5	Sending a Sequence Number PDU	16
6.2.2.6	Receiving a Sequence Number PDU	16
7	Security Considerations	16
8	IANA Considerations	17
9	IANA Considerations	17
10	References	17
10.1	Normative References	17
10.2	Informative References	17
	Author's Addresses	18
	Full Copyright Statement	19

[1](#) Introduction

This document describes the application of a database of long-lived cryptographic keys, as defined in [\[KEYTAB\]](#), to establish session-specific cryptographic keys to provide authentication services in routing protocols. Keys may be established between two peers for pair-wise communications, or between groups of peers for multicast traffic.

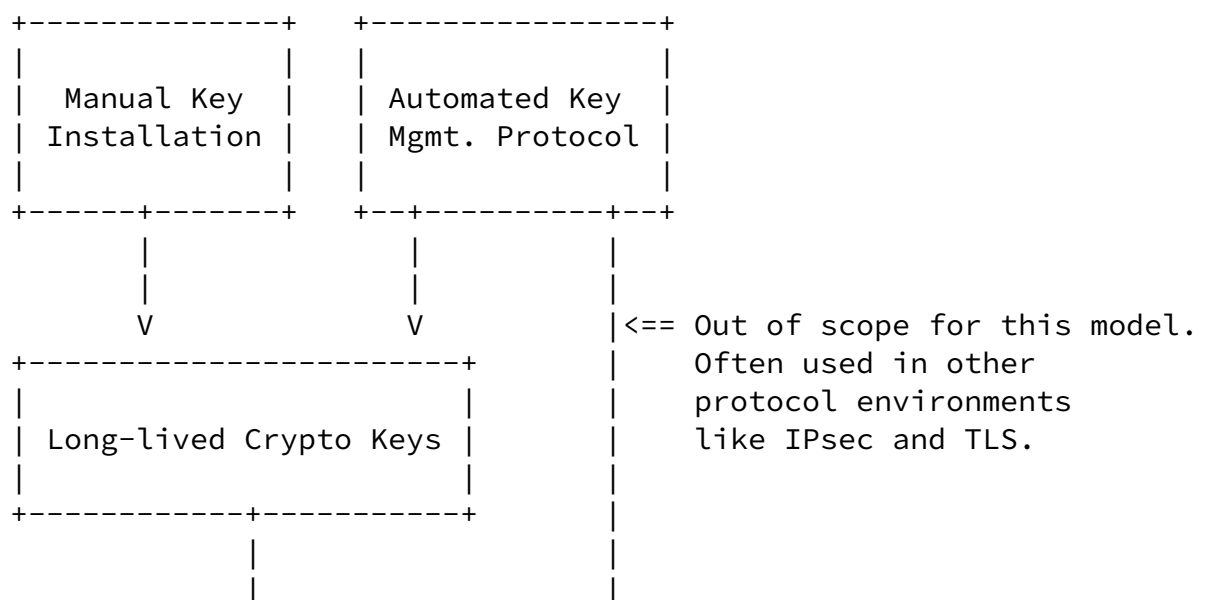
1.1 Terminology

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

2 Architecture and Design

Figure 1 illustrates the establishment and use of cryptographic keys for authentication in routing protocols. Long-lived cryptographic keys are inserted in a database manually. In the future, we anticipate an automated key management protocol to insert these keys in the database. (While this future environment conceivably includes automated key management protocols to negotiate short-lived cryptographic session keys, such keys are out of scope for this database.) The structure of the database of long-lived cryptographic keys is described in [[KEYTAB](#)].

The cryptographic keying material for individual sessions is derived from the keying material stored in the database of long-lived cryptographic keys. A key derivation function (KDF) and its inputs are named in the database of long-lived cryptographic keys; session specific values based on the routing protocol are input the the KDF. Protocol specific key identifiers may be assigned to the cryptographic keying material for individual sessions if needed.



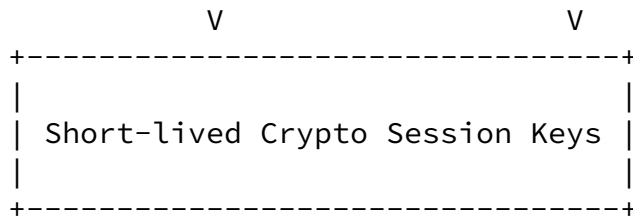
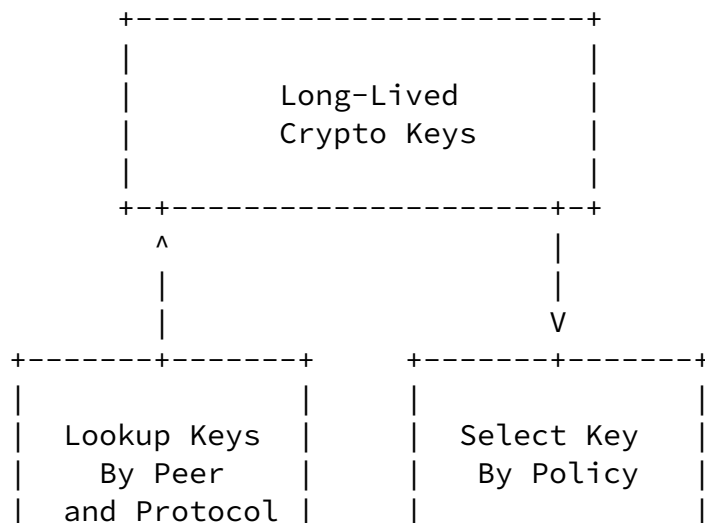


Figure 1. Cryptographic key establishment and use.

3 Pair-wise Application

Figure 2 illustrates how the long-lived cryptographic keys are accessed and employed when an entity wishes to establish a protected session with a peer. As one step in the initiation process, the initiator requests the set of long term keys associated with the peer for the particular protocol. If the set contains more than one key, the initiator selects one long-term key based on the local policy. The long-term key is provided as an input, along with session-specific information (e.g., ports or initial counters), to a key derivation function. The result is session-specific key material which is used to generate cryptographic authentication.

Where the initiator is establishing a multicast session, the Peer in the key request identifies the set of systems that will receive this information.



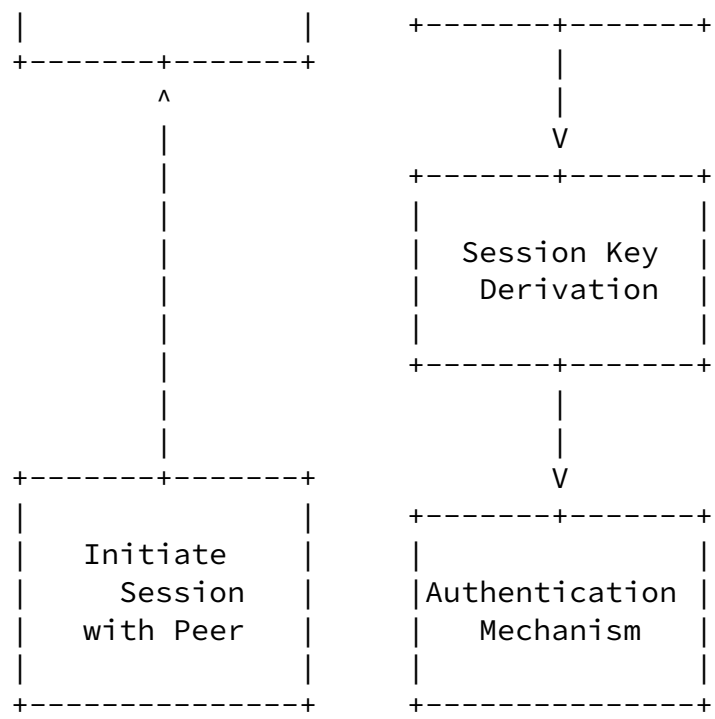
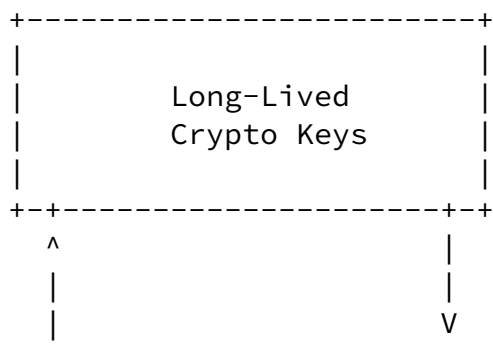


Figure 2. Session Initiation

Figure 3 illustrates how the long-lived cryptographic keys are accessed and employed when an entity receives a request establish a protected session with a peer. As step one in the session establishment process, the receiver extracts the keyID for the long-term keyID from the received data. The receiver then requests the specified long-term key from the table. The long-term key is provided as an input, along with session-specific information (e.g., ports or initial counters), to a key derivation function. The result is session-specific key material which is used to verify the cryptographic authentication information.



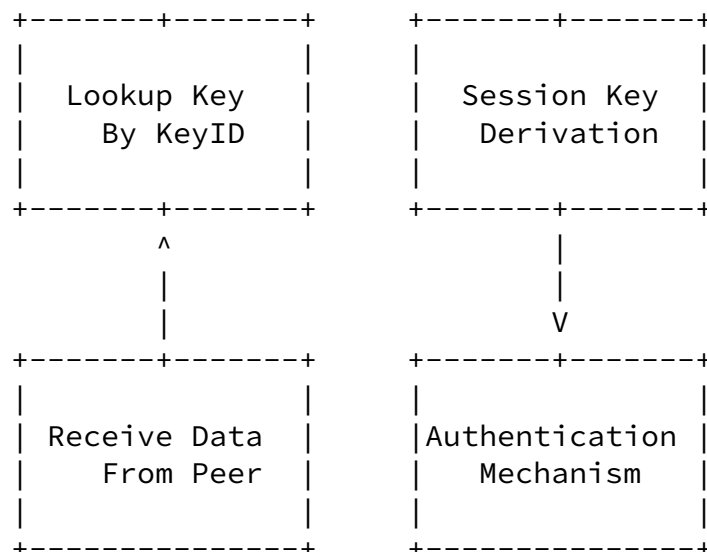


Figure 3. Session Acceptance

4 Identifier Mapping

[KEYTAB] specifies a 16-bit identifier, but protocols already exist with key identifiers of various sizes. Where the identifiers are of different sizes, an extra mapping step may be required. Note that mapping mechanisms are local – that is, different mapping mechanisms could be employed on different peers.

In practice, the mapping process need only be applied to the LocalKeyID, whose value must be unique in the context of the database, as defined in [KEYTAB]. Uniqueness is not required for the PeerKeyID, so mapping is generally restricted to truncation. Mapping would only be needed to expand PeerKeyID's value beyond 16 bits.

4.1 Selected Range Reservation

Where a protocol uses an index of less than 16 bits, a selected range of the local index space can be reserved for a particular protocol. For example, consider two protocols P1 and P2 that each use 8 bit key identifiers. Without identifier mapping these protocols would share the space {0x0000 through 0x00ff} which would limit the pair of protocols to 256 keys in total. By reserving the ranges {0x7f00 through 0x7fff} and {0x7e00 through 0x7eff} for P1 and P2

respectively permits each protocol to use the full 256 key identifiers and establishes an unambiguous mapping for the protocol key identifiers and local table identifiers.

When an initiator selects a key from the set in the table, the given key identifier needs to be masked or shifted to the on-the-wire range. Before requesting a specific key, the receiver would use a shim layer to map the on-the-wire identifier into the reserved range.

[4.2](#) Protocol Specific Mapping Tables

Each protocol can also maintain a simple mapping table with two fields: the 16 bit index and the protocol specific value:

KEYTAB index (16 bits) | Protocol specific index (8 bits)

In this case, the host system would maintain separate mapping tables for protocols P1 and P2.

[5](#) Database Maintenance

The previous sections focus upon installing and using the cryptographic keys in the database. A mechanism or mechanisms to remove unneeded keys is also needed to ensure that the key material up-to-date. [[KEYTAB](#)] provides mechanisms for expiration of entries; such key management could be performed in a fully automated fashion. Other reasons for key removal, such as severing a business relationship, or deciding a long lived key has been compromised before its expiration date, would inherently require a manual key removal process.

[6](#) Worked Examples

[6.1](#) Worked Example: TCP-AO

This section describes the way a TCP-AO implementation could use the database. [[tcpao](#)] TCP-AO protocol is an example where the key identifier is limited to 8 bits, so an identifier mapping is needed.

We will assume two peers Xp and Yp. Xp employs the range reservation method for mapping and has reserved the range {0x7f00 ... 0x7fff} for

LocalKeyIDs for TCP-A0, mapping to {0x00 ... 0xff}. Yp employs a protocol specific mapping table in its TCP-A0 implementation.

The following subsections describe how peers Xp and Yp make use of the database of long-lived cryptographic keys when Xp and Yp respectively initiate a session. (Note: Rollover to new sessions keys during a session is described in [[tcpao](#)].)

6.1.1 Setup

The owners of Xp and Yp determine a need for authenticated communication using TCP-A0. They decide to use AES-CMAC-128 for authentication, so a 128 bit key is needed. They decide to use the same key for both directions (inbound and outbound), and that the key will be available from 12/31/2010 through 12/31/2011. Through an out-of-band channel, the administrators establish the shared secret:

0x0123456789ABCDEF0123456789ABCDEF

Peer Xp selects the first available TCP-A0 identifier in the reserved range, which is 0x7f05 and maps to an eight-bit identifier 0x05. Peer Yp selects the next available TCP-A0 identifier, 0x12, and the next available LocalKeyID, which is 0x0107. Peer Yp also adds an entry to its TCP-A0 mapping table mapping the LocalKeyID to the TCP-A0 identifier, as shown in Figure 5:

LocalKeyID	TCP-A0 identifier
-----	-----
0x001a	0x01
0x004d	0x02
...	...
0x0107	0x12

Figure 5. Protocol Specific KeyID Mapping Table for TCP-A0

After exchanging the TCP-A0 identifiers, the peers have sufficient information to establish their [[KEYTAB](#)] entries. Peer Xp's [[KEYTAB](#)] entry is shown as Figure 6:

LocalKeyID 0x7f05

PeerKeyID 0x0012

KDF	????
KDFInputs	none
AlgID	AES-CMAC-128
Key	0x0123456789ABCDEF0123456789ABCDEF
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	yp.example.com
Protocol	TCP-A0

Figure 6. Key Table Entry on Xp

Peer Yp's [\[KEYTAB\]](#) entry is shown as Figure 6:

LocalKeyID	0x0107
PeerKeyID	0x0005
KDF	????
KDFInputs	none
AlgID	AES-CMAC-128
Key	0x0123456789ABCDEF0123456789ABCDEF
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	xp.example.com
Protocol	TCP-A0

Figure 7. Key Table Entry on Yp

[6.1.2](#) Protocol Operation: Xp Initiates a Connection

Peer Xp wishes to initiate a connection with Peer Yp.

- (1) Xp performs a key lookup for {Peer=Yp, Protocol=TCP-A0}, and the entry with LocalKeyID 0x7f05 is returned.
- (2) The LocalKeyID 0x7f05 is range mapped by Xp to the TCP-A0 identifier 0x05.
- (3) Xp performs the session key derivation using the mechanism specified for the TCP-A0 protocol in [\[ao-crypto\]](#).
- (4) Xp generates the AES-CMAC-128 MACs for the outgoing traffic using the derived key, and asserts the key identifier 0x05 in the packets.
- (5) Yp receives a protected packet from Xp, and extracts the key identifier 0x05.
- (6) Yp performs a key lookup for {Peer=Xp, Protocol=TCP-A0, PeerKeyID=0x05}, and the entry with LocalKeyID 0x0107 is returned.
- (7) Yp performs the session key derivation using the mechanism specified for the TCP-A0 protocol in [\[ao-crypto\]](#).
- (8) Yp verifies the MACs for the incoming traffic using the derived

key.

[6.1.3](#) Protocol Operation: Yp Initiates a Connection

Where Peer Yp establishes the connection, the same process is followed, except that the range mapping process from step (2) is replaced by a table lookup.

[6.2](#) Worked Example: IS-IS

This section describes the way an IS-IS implementation supporting the IS-IS generic cryptographic authentication mechanism could use the database. [[isis](#)] [[rfc1195](#)] [[rfc5310](#)] IS-IS is an interior gateway protocol (IGP) that can be used to support IP as well as OSI.

IS-IS routers are grouped into "areas". Routers establish adjacencies with their neighboring routers and share link state information through flooding. Information shared within an area is termed Level 1 information, and information shared between areas is termed level 2 information. An IS-IS router can be Level 1, Level 2, or both (designated as Level 1/2). Level 1 routers only form Level 1 adjacencies with other Level 1 or Level 1/2 routers within their own area. Level 2 or Level 1/2 routers can form adjacencies with other Level 2 or Level 1/2 routers in other areas as well as their own area.

An IS-IS deployment can have multiple Level 1 areas; Level 1 areas are differentiated by area addresses that are unique within the IS-IS deployment. (An IS-IS deployment has only a single Level 2 domain which is formed from all the Level 2 and Level 1/2 routers within the routing domain, irrespective of their area addresses.)

The IS-IS protocol supports routers that are connected by LANs and point-to-point links. Level 1 and Level 2 messages on a LAN are differentiated by the broadcast address. Point-to-Point links may be configured as Level 1, Level 2, or both.

This worked example describes how an IS-IS router, denoted Rp, makes use of the database for the following eight cases:

- * sending a LAN IS to IS Hello PDU
- * receiving a LAN IS to IS Hello PDU
- * sending a Point-to-Point IS to IS Hello PDU
- * receiving a Point-to-Point IS to IS Hello PDU
- * sending a Link State Packet
- * receiving a Link State Packet
- * sending sequence number PDUs

* receiving sequence number PDUs

INTERNET DRAFT

November 8, 2010

In this example, Rp is a Level 1/2 router. Rp has two LAN interfaces; on the first interface (eth0) Rp is connected to other Level 1 routers; on the second interface (eth1) Rp is connected to both other Level 1 and Level 2 routers by a LAN. Rp is also connected to one additional Level 1 router, Rq, by a point-to-point link (ppp1). The Level 1 area that Rp participates in has an area address of:

0x4922

The IS-IS protocol supports routers that are connected by LANs and point-to-point links. Level 1 and Level 2 messages on a LAN are differentiated by the broadcast address. The implementation will use the following multicast addresses:

Level 1: 01-80-C2-00-00-14

Level 2: 01-80-C2-00-00-15

The authentication mechanism specified in [RFC 5310](#) uses a 16 bit key identifier which matches the key table, so the identifier can be used directly.

In this example, an interior router Rp makes use of the database of long-lived cryptographic keys to manage its IS-IS long-term keys. Rp participates in both Level 1 and Level 2.

(For this example, we will use a single area address for each area. Note that multiple area addresses can be supported for each area.)

In addition to the area addresses that specify the set of recipients, six octet system IDs are used to uniquely identify the sender. The system ID is required to be unique within the area, and in practice is derived from a MAC address. Rp has the following system ID

0x123456

The Network Entity Title (or NET) is constructed from the system ID and the area. Rp has the following NET:

Level 1 Area: 0x4922123456

[6.2.1](#) Setup

The owners of the IS-IS system determine a need for authenticated communication between the interior gateways. They decide to use HMAC-SHA1 for authentication with 128 bit keys.

For routers that only participate in Level 1, there are two long-term keys: one for hello traffic, and a second for link state PDUs. For

routers that participate in both Level 1 and Level 2, two additional long-term keys are required: again, the two keys are used to protect hellos and LSPs, respectively. The owners decide these keys will be available from 12/31/2010 through 12/31/2011. Through an out-of-band channel, the administrators establish the following shared secrets:

- * a pairwise key for each point-to-point link to protect hello messages;
- * a multicast key for each broadcast LAN interface for each Level to protect hello messages;
- * a multicast key for LSP and sequence number packets for each Level 1 area; and
- * a multicast key for LSP and sequence number packets for the Level 2 domain.

Since Rp will send Level 1 hellos on two LANs and a point-to-point link, and Level 2 hellos on one LAN, it will be configured with four IS-IS hello keys. These keys are specified in Figures 8 through 11, respectively.

```
Level 1 hello traffic: 0x0123456789ABCDEF0123456789ABCDEF
Level 1 link state PDUs: 0x123456789ABCDEF0123456789ABCDEF0
Level 2 hello traffic: 0x23456789ABCDEF0123456789ABCDEF01
Level 2 link state PDUs: 0x3456789ABCDEF0123456789ABCDEF012
```

Since the three LAN hello keys are for multicast traffic, the leading bit of the LocalKeyID is required to be 1. PeerkeyID is set to group. There is a pairwise key for the point-to-point hellos (in Figure

10), Since there is no concept of a session, key diversification is not needed. This implies there is no kdf or kdf inputs, and the long-term key is used directly to protect the messages. The algorithm id indicates hmac sha1, and the direction is both inbound and outbound.

The key generator selects the first available IS-IS identifier. For a new implementation, any value may be selected. Otherwise, the key identifiers can not collide with currently assigned values for IS-IS keys. Since Rp participates at both Level 1 and Level 2, Rp installs all four keys. Rp's [KEYTAB](#) entries are shown as Figures 8 through 11:

LocalKeyID	0x7101
PeerKeyID	group
KDF	none

KDFInputs	none
AlgID	HMAC-SHA-1
Key	0x0123456789ABCDEF0123456789ABCDEF
Interface	eth0
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	0x4922
Protocol	IS-IS Hello L1

Figure 8. Key Table Entry on Rp for Level 1 LAN Hellos on eth0

(use ppp1)

LocalKeyID	0x7102
PeerKeyID	0x7102
KDF	none
KDFInputs	none
AlgID	HMAC-SHA-1
Key	0x123456789ABCDEF0123456789ABCDEF0
Interface	eth1
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	0x4922

Protocol IS-IS Hello L1

Figure 9. Key Table Entry on Rp for Level 1 LAN Hellos on eth1

LocalKeyID	0x0003
PeerKeyID	0x0105
KDF	none
KDFInputs	none
AlgID	HMAC-SHA-1
Key	0x23456789ABCDEF0123456789ABCDEF01
Interface	ppp1
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	0x4922
Protocol	IS-IS Hello L1

Figure 10. Key Table Entry on Rp for Level 1 point-to-point Hellos

LocalKeyID	0x7103
PeerKeyID	group
KDF	none
KDFInputs	none

AlgID	HMAC-SHA-1
Key	0x3456789ABCDEF0123456789ABCDEF012
Interface	eth1
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	0x4922
Protocol	IS-IS Hello L2

Figure 11. Key Table Entry on Rp for Level 2 Hellos on eth1

Rp also requires two multicast keys for flooding Link State Packets and transmitting Sequence number packets. The first key is shared throughout the Level 1 Area 0x4922; the second key is shared amongst the routers in the Level 2 domain. Rp's [\[KEYTAB\]](#) entries for the two multicast LSP/sequence number packet keys are shown as Figures 12 and 13:

LocalKeyID	0x7104
PeerKeyID	group
KDF	none
KDFInputs	none
AlgID	HMAC-SHA-1
Key	0x456789ABCDEF0123456789ABCDEF0123
Interface	*
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	0x4922
Protocol	IS-IS LSP L1

Figure 12. Key Table Entry on Rp for Level 1 LSPs and Sequence Number packets

LocalKeyID	0x7105
PeerKeyID	group
KDF	none
KDFInputs	none
AlgID	HMAC-SHA-1
Key	0x56789ABCDEF0123456789ABCDEF01234
Interface	*
Direction	both
NotBefore	12/31/2010
NotAfter	12/31/2011
Peers	IS-IS L2
Protocol	IS-IS LSP L2

Figure 13. Key Table Entry on Rp for Level 1 LSPs and Sequence Number

packets

[6.2.2](#) Protocol Operations

The following subsections describe how an IS-IS router makes use of the database for the following four cases:

- * sending a Hello message
- * receiving a Hello message
- * sending a Link State Packet
- * receiving a Link State Packet
- * sending a sequence number PDU

* receiving a sequence number PDU

[6.2.2.1](#) Sending a Hello Message

Rp wishes to send a Hello message. Because Rp is configured with three Level 1 interfaces, and one Level 2 interface, four different hello messages will be transmitted. Each message is protected with the key IS-IS Hello key for that interface and level.

For each LAN interface:

- (1) Rp performs a key lookup for the interface (e.g., eth0 or eth1) with the protocol "IS-IS Hello L1".
- (2) Rp parses the key entry and determines the algorithm attribute (in this example, the algorithm attribute is always HMAC-SHA1).
- (3) Rp constructs the outgoing LAN Hello PDU. If replay protection is a concern, Rp includes a timestamp with the local time. (The timestamp would be contained in a new TLV. Such a TLV has not been specified at this time.)
- (4) Rp generates the SHA1-HMAC for the outgoing LAN Hello using the long-term key, and asserts the appropriate key identifier in the [RFC 5310](#) authentication mechanism TLV.
- (5) Rp transmits the Hello message on the LAN interface using the Level 1 multicast MAC address.

For the point-to-point HELLO:

- (1) Rp performs a key lookup for the interface (ppp1) and protocol "IS-IS Hello L1".
- (2) Rp parses the key entry and determines the algorithm attribute (i.e., HMAC-SHA1).
- (3) Rp constructs the outgoing point-to-point Hello PDU. If replay protection is a concern, Rp includes a timestamp with the local time.
- (4) Rp generates the SHA1-HMAC for the outgoing point-to-point LAN Hello using the long-term key, and asserts the key identifier in the [RFC 5310](#) authentication mechanism TLV.
- (5) Rp transmits the Hello message over the point-to-point link.

[6.2.2.2](#) Receiving a Hello Message

Rp processes hello messages by the following algorithm:

- (1) Rp parses the [RFC 5310](#) authentication mechanism TLV and performs a key lookup using the included PeerKeyID.
- (2) Rp parses the key entry and
 - (a) Rp verifies the keyID is associated with this interface. If the interface does not match, the sender or receiver is misconfigured. An alarm is triggered and the hello is discarded. Otherwise, continue with (2)(b).
 - (b) Rp determines the algorithm attribute (in this case, HMAC-SHA1).
- (3) Rp calculates the SHA1-HMAC and compares it to the value in the Hello. If the HMACs do not match, the message is discarded. (Otherwise proceed to step 4.)
- (4) Rp checks the timestamp state for the sender. (If the timestamp value is NULL, proceed to 6. If there is a timestamp value for this sender, proceed to step 7).
- (5) Rp extracts the timestamp, if any, and compares it to the value in the Hello. If the timestamp is earlier than the stored timestamp, or no timestamp was present, the Hello message is discarded. If the timestamp is later than the stored timestamp, update the stored value and process the Hello message.
- (6) Process the hello message.

[Note that there is no different in processing for LAN or Point-to-point hellos.]

[6.2.2.3](#) Generating a Link State PDU

Rp wishes to send a link state PDU to the other routers. To perform this task, Rp constructs two separate LSPs, protected by its Level 1 and Level 2 LSP keys. The LSPs are transmitted to each neighbor that has formed an adjacency with Rp as appropriate. (Level 1 LSPs are ONLY transmitted over links which have a Level 1 adjacency, and similarly Level 2 LSPs only over links which have Level 2 adjacencies.)

- (1) Rp performs a key lookup for protocol "IS-IS L1 Flood". (The entry with PeerKeyID 0x7104 is returned.)
- (2) Rp parses the key entry and determines the algorithm attribute (HMAC-SHA1).
- (3) Rp constructs the Level 1 link state PDU. Note that this includes a sequence number.
- (4) Rp generates the appropriate MAC for the outgoing LSP using the long-term key, and asserts the key identifier 0x7104 in the [RFC 5310](#)

authentication mechanism TLV.

(5) Rp transmits the LSP to all current L1 neighboring adjacencies.

The process is repeated for Level 2, beginning with a key lookup for protocol "IS-IS L2 Flood". Note that the Level 2 link state PDU constructed in step (3) will contain different information than the Level 1 LSP.

[6.2.2.4](#) Receiving a Link State PDU

Rp processes incoming link state PDUs by the following algorithm:

(1) Rp parses the [RFC 5310](#) authentication mechanism TLV and performs a key lookup using the PeerKeyID.

(2) Rp parses the key entry and determines the algorithm attribute (HMAC-SHA1)

(3) Rp calculates the SHA1-HMAC and compares it to the value in the link state PDU. If the HMACs do not match, the message is discarded. (Otherwise proceed to step 4.)

(4) Rp performs IS-IS processing to ensure the message is fresh (e.g., checks the sequence number for the sender.) If Rp already has fresher information, Rp will discard the packet, then construct an LSP with the fresher information and forward it to the sender. Otherwise, perform step 5.

(5) Rp forwards the verified Link State PDU to all neighbors with the same level except the neighbor that transmitted the PDU. (That is, Level 1 Link State PDUs are forwarded to Level 1 neighbors; Level 2 Link State PDUs are forwarded to Level 2 neighbors.)

[6.2.2.5](#) Sending a Sequence Number PDU

The cryptographic process for protecting a Sequence Number PDU is the same as those specified for LSPs in 6.2.2.3. Note that there is no difference when sending partial or full link state PDUs.

[6.2.2.6](#) Receiving a Sequence Number PDU

The cryptographic process for authenticating a Sequence Number PDU is the same as those specified for LSPs in 6.2.2.4.

[7](#) Security Considerations

The "hello" message processing examples assume the existence of a timestamp extension to provide replay protection. Sequence numbers for hello messages would provide an alternative solution; the authors selected a timestamp since this imposes no state on the sender. Time

synchronization is not needed to achieve replay protection; receivers

INTERNET DRAFT

November 8, 2010

that desire replay protection simply retain the timestamp from the previous hello for comparison.

By requiring an IS-IS router to begin using timestamps immediately upon key change, or not at all, step (x) in 6.2.2.2 could have been omitted. By verifying that previous messages did not have a timestamp, a receiver prevents replay of a past hello message that did not include timestamps that was protected with the current key.

The timestamp was omitted from the point-to-point hello in the example based on an assumption of physically protected media. If that is not the case, the timestamp could be included in these messages as well.

[8](#) IANA Considerations

This document requires no actions by IANA.

[9](#) IANA Considerations

Mike Shand was amazingly patient and helpful, demystifying and explaining IS-IS. The authors are grateful for his assistance. Any remaining mistakes in [section 6.2](#) are the responsibility of the authors, of course!

[10](#) References

[10.1](#) Normative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [KEYTAB] R. Housley and Polk, T. "Database of Long-Lived Cryptographic Keys", [draft-housley-saag-crypto-key-table-04.txt](#), October 2010.

[10.2](#) Informative References

[tcpao] J. Touch, Mankin A., and Bonica R. "The TCP Authentication Option", [draft-ietf-tcpm-tcp-auth-opt-08.txt](#), October 2009.

[ao-crypto] Lebovitz, G., "Cryptographic Algorithms, Use, & Implementation Requirements for TCP Authentication Option", [draft-lebovitz-ietf-tcpm-tcp-ao-crypto-02.txt](#),

July 2009.

[rfc1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", [RFC 1195](#), December 1990.

[isis] International Organization for Standardization, "Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO/IEC 10589:2002, Second Edition, Nov 2002.

[rfc5310] M. Bhatia, Manral, V., Li, T., Atkinson, R., White, R. and Fanto, M. "IS-IS Generic Cryptographic Authentication", [RFC 5310](#), February 2009

Author's Addresses

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

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

INTERNET DRAFT

November 8, 2010

Full Copyright Statement

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

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) 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.

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

