Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 2, 2014

L. Zheng
M. Chen
Huawei Technologies
M. Bhatia
Alcatel-Lucent

August 29, 2013

LDP Hello Cryptographic Authentication
draft-ietf-mpls-ldp-hello-crypto-auth-02.txt

Abstract

This document introduces a new optional Cryptographic Authentication TLV that LDP can use to secure its Hello messages. It secures the Hello messages against spoofing attacks and some well known attacks against the IP header. This document describes a mechanism to secure the LDP Hello messages using National Institute of Standards and Technology (NIST) Secure Hash Standard family of algorithms.

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on March 2, 2014.

Copyright Notice

Table of Contents

1. Introduction

The Label Distribution Protocol (LDP) [RFC5036] sets up LDP sessions that runs between LDP peers. The peers could either be directly connected at the link level or could be multiple hops away. An LDP Label Switching Router (LSR) could either be configured with the identity of its peers or could discover them using LDP Hello messages. These messages are sent encapsulated in UDP addressed to "all routers on this subnet" or to a specific IP address. Periodic Hello messages are also used to maintain the relationship between LDP peers necessary to keep the LDP session active.

Unlike other LDP messages, the Hello messages are sent using UDP and not TCP. This implies that these messages do not use the security mechanisms defined for TCP [RFC5926]. Besides a note that some configuration may help protect against bogus discovery messages, [RFC5036] does not really provide any security mechanism to protect the Hello messages.

Spoofing a Hello packet for an existing adjacency can cause the valid adjacency to time out and in turn can result in termination of the associated session. This can occur when the spoofed Hello specifies a smaller Hold Time, causing the receiver to expect Hellos within this smaller interval, while the true neighbor continues sending Hellos at the previously agreed lower frequency. Spoofing a Hello packet can also cause the LDP session to be terminated directly, which can occur when the spoofed Hello specifies a different Transport Address, other than the previously agreed one between neighbors. Spoofed Hello messages have been observed and reported as a real problem in production networks [I-D.ietf-karp-routing-tcp-analysis].

[RFC5036] describes that the threat of spoofed Basic Hellos can be reduced by accepting Basic Hellos only on interfaces to which LSRs that can be trusted are directly connected, and ignoring Basic Hellos not addressed to the "all routers on this subnet" multicast group. Spoofing attacks via Extended Hellos are a potentially more serious threat. An LSR can reduce the threat of spoofed Extended Hellos by filtering them and accepting only those originating at sources permitted by an access list. However, filtering using access lists requires LSR resource, and does not prevent IP-address spoofing.

This document introduces a new Cryptographic Authentication TLV which is used in LDP Hello message as an optional parameter. It enhances the authentication mechanism for LDP by securing the Hello message against spoofing attack. It also introduces a cryptographic sequence number carried in the Hello messages that can be used to protect against replay attacks. The LSRs could be configured to only accept

Hello messages from specific peers when authentication is in use.

5-1

Using this Cryptographic Authentication TLV, one or more secret keys (with corresponding key IDs) are configured in each system. For each LDP Hello packet, the key is used to generate and verify a HMAC Hash that is stored in the LDP Hello packet. For cryptographic hash function, this document proposes to use SHA-1, SHA-256, SHA-384, and SHA-512 defined in US NIST Secure Hash Standard (SHS) [FIPS-180-3]. The HMAC authentication mode defined in NIST FIPS 198 is used [FIPS-198]. Of the above, implementations MUST include support for at least HMAC-SHA-256 and SHOULD include support for HMAC-SHA-1 and MAY include support for either of HMAC-SHA-384 or HMAC-SHA-512.

5-2

2. Cryptographic Authentication TLV

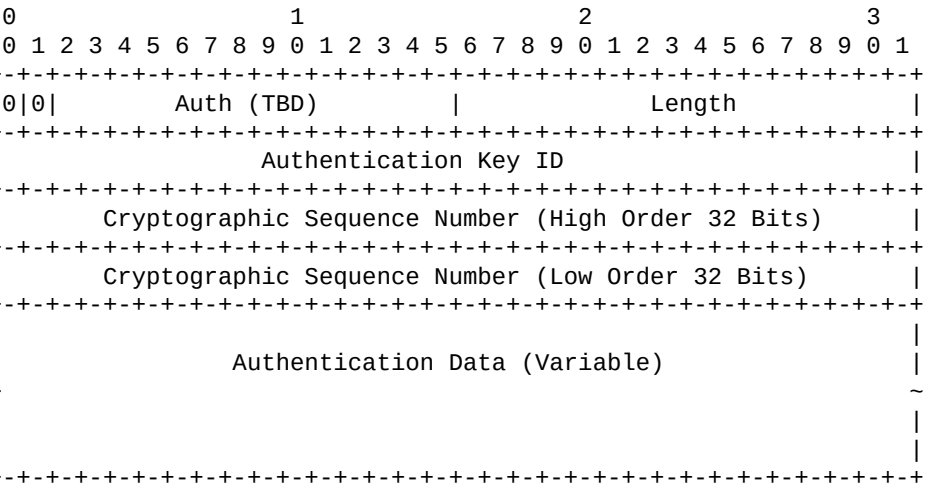
2.1. Optional Parameter for Hello Message

[RFC5036] defines the encoding for the Hello message. Each Hello message contains zero or more Optional Parameters, each encoded as a TLV. Three Optional Parameters are defined by [RFC5036]. This document defines a new Optional Parameter: the Cryptographic Authentication parameter.

Optional Parameter	Type
IPv4 Transport Address	0x0401 (RFC5036)
Configuration Sequence Number	0x0402 (RFC5036)
IPv6 Transport Address	0x0403 (RFC5036)
Cryptographic Authentication	0x0404 (this document, TBD by IANA)

The Cryptographic Authentication TLV Encoding is described in section 2.2.

2.2. Cryptographic Authentication TLV Encoding



- Type: TBD, Cryptographic Authentication

- Length: Specifying the length in octets of the value field.

- Auth Key ID: 32 bit field that identifies the algorithm and the secret key used to create the message digest carried in LDP payload.

6-1

6-2

- Cryptographic Sequence Number: 64-bit strictly increasing sequence number that is used to guard against replay attacks. The 64-bit sequence number MUST be incremented for every LDP Hello packet sent by the LDP router. Upon reception, the sequence number MUST be greater than the sequence number in the last LDP Hello packet accepted from the sending LDP neighbor. Otherwise, the LDP packet is considered a replayed packet and dropped.

LDP routers implementing this specification MUST use available mechanisms to preserve the sequence number's strictly increasing property for the deployed life of the LDP router (including cold restarts). One mechanism for accomplishing this could be to use the high-order 32 bits of the sequence number as a wrap/boot count that is incremented anytime the LDP router loses its sequence number state. Techniques such as sequence number space partitioning described above or non-volatile storage preservation can be used but are really beyond the scope of this specification. Sequence number wrap is described in Section 2.3.

- Authentication Data:

This field carries the digest computed by the Cryptographic Authentication algorithm in use. The length of the Authentication Data varies based on the cryptographic algorithm in use, which is shown as below:

Auth type	Length
HMAC-SHA1	20 bytes
HMAC-SHA-256	32 bytes
HMAC-SHA-384	48 bytes
HMAC-SHA-512	64 bytes

2.3. Sequence Number Wrap

When incrementing the sequence number for each transmitted LDP packet, the sequence number should be treated as an unsigned 64-bit value. If the lower order 32-bit value wraps, the higher order 32-bit value should be incremented and saved in non-volatile storage. If by some chance the LDP router is deployed long enough that there is a possibility that the 64-bit sequence number may wrap, all keys, independent of key distribution mechanism, MUST be reset to avoid the possibility of replay attacks. Once the keys have been changed, the higher order sequence number can be reset to 0 and saved to non-volatile storage.

3. Cryptographic Authentication Procedure

As noted earlier, the Auth Key ID maps to the authentication algorithm and the secret key used to generate and verify the message digest. This specification discusses the computation of LDP Cryptographic Authentication data when any of the NIST SHS family of algorithms is used in the Hashed Message Authentication Code (HMAC) mode.

The currently valid algorithms (including mode) for LDP Cryptographic Authentication include:

HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512

Of the above, implementations of this specification MUST include support for at least HMAC-SHA-256 and SHOULD include support for HMAC-SHA-1 and MAY also include support for HMAC-SHA-384 and HMAC-SHA-512.

Implementations of this standard MUST use HMAC-SHA-256 as the default authentication algorithm.

9-1

4. Cross Protocol Attack Mitigation

In order to prevent cross protocol replay attacks for protocols sharing common keys, the two octet LDP Cryptographic Protocol ID is appended to the authentication key prior to use. Other protocols using cryptographic authentication as specified herein MUST similarly append their respective Cryptographic Protocol IDs to their keys in this step. Refer to IANA Considerations (Section 8).

10-1

5. Cryptographic Aspects

In the algorithm description below, the following nomenclature, which is consistent with [FIPS-198], is used:

H is the specific hashing algorithm (e.g. SHA-256).

K is the Authentication Key from the LDP security association.

Ks is a Protocol Specific Authentication Key obtained by appending Authentication Key (K) with the two-octet LDP Cryptographic Protocol ID appended.

Ko is the cryptographic key used with the hash algorithm.

B is the block size of H, measured in octets rather than bits.

Note that B is the internal block size, not the hash size.

For SHA-1 and SHA-256: B == 64

For SHA-384 and SHA-512: B == 128

L is the length of the hash, measured in octets rather than bits.

XOR is the exclusive-or operation.

Opad is the hexadecimal value 0x5c repeated B times.

Ipad is the hexadecimal value 0x36 repeated B times.

Apad is a value which is the same length as the hash output or message digest. In case of IPv4, the first 4 octets contain the IPv4 source address followed by the hexadecimal value 0x878FE1F3 repeated (L-4)/4 times. In case of IPv6, the first 16 octets contain the IPv6 source address followed by the hexadecimal value 0x878FE1F3 repeated (L-16)/4 times. This implies that hash output is always a length of at least 16 octets.

5.1. Preparing the Cryptographic Key

The LDP Cryptographic Protocol ID is appended to the Authentication Key (K) yielding a Protocol Specific Authentication Key (Ks). In this application, Ko is always L octets long. While [RFC2104] supports a key that is up to B octets long, this application uses L as the Ks length consistent with [RFC4822], [RFC5310], [RFC5709] and [RFC6506]. According to [FIPS-180-3], Section 3, keys greater than L

octets do not significantly increase the function strength. Ks is computed as follows:

If the Protocol Specific Authentication Key (Ks) is L octets long, then Ko is equal to Ks. If the Protocol Specific Authentication Key (Ks) is more than L octets long, then Ko is set to H(Ks). If the Protocol Specific Authentication Key (Ks) is less than L octets long, then Ko is set to the Protocol Specific Authentication Key (Ks) with zeros appended to the end of the Protocol Specific Authentication Key (Ks) such that Ko is L octets long.

5.2. Computing the Hash

First, the Authentication Data field in the Cryptographic Authentication TLV is filled with the value Apad. Then, to compute HMAC over the Hello packet it performs:

H(Ko XOR Opad || H(Ko XOR Ipad || (Hello Packet)))

Hello Packet refers to the LDP Hello packet excluding the IP header.

When XORing Ko and Ipad, or XORing Ko and Opad, Ko must be padded with zeros to the length of Ipad and the Opad.

5.3. Result

The resultant Hash becomes the Authentication Data that is sent in the Authentication Data field of the Cryptographic Authentication TLV. The length of the Authentication Data field is always identical to the message digest size of the specific hash function H that is being used.

This also means that the use of hash functions with larger output sizes will also increase the size of the LDP packet as transmitted on the wire.

12-1

6. Processing Hello Message Using Cryptographic Authentication

6.1. Transmission Using Cryptographic Authentication

Prior to transmitting Hello message, the Length in the Cryptographic Authentication TLV header is set as per the authentication algorithm that is being used. It is set to 24 for HMAC-SHA-1, 36 for HMAC-SHA-256, 52 for HMAC-SHA-384 and 68 for HMAC-SHA-512.

The Auth Key ID field is set to the ID of the current authentication key. The HMAC Hash is computed as explained in Section 3. The resulting Hash is stored in the Authentication Data field prior to transmission. The authentication key MUST NOT be carried in the packet.

6.2. Receipt Using Cryptographic Authentication

The receiving LSR applies acceptability criteria for received Hellos using cryptographic authentication. If the Cryptographic Authentication TLV is unknown to the receiving LSR, the received packet MUST be discarded according to Section 3.5.1.2.2 of [RFC5036].

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the cryptographic sequence number in the LDP packet is less than or equal to the last sequence number received from the same neighbor, the LDP packet MUST be discarded.

Before the receiving LSR performs any processing, it needs to save the values of the Authentication Data field. The receiving LSR then replaces the contents of the Authentication Data field with Apad, computes the Hash, using the authentication key specified by the received Auth Key ID field, as explained in Section 3. If the locally computed Hash is equal to the received value of the Authentication Data field, the received packet is accepted for other normal checks and processing as described in [RFC5036]. Otherwise, if the locally computed Hash is not equal to the received value of the Authentication Data field, the received packet MUST be discarded.

7. Security Considerations

Section 1 of this document describes the security issues arising from the use of unauthenticated LDP Hello messages. In order to address those issues, it is RECOMMENDED that all deployments use the Cryptographic Authentication TLV to authenticate the Hello messages.

The quality of the security provided by the Cryptographic Authentication TLV depends completely on the strength of the cryptographic algorithm in use, the strength of the key being used, and the correct implementation of the security mechanism in communicating LDP implementations. Also, the level of security provided by the Cryptographic Authentication TLV varies based on the authentication type used.

It should be noted that the authentication method described in this document is not being used to authenticate the specific originator of a packet but is rather being used to confirm that the packet has indeed been issued by a router that has access to the Authentication Key.

Deployments SHOULD use sufficiently long and random values for the Authentication Key so that guessing and other cryptographic attacks on the key are not feasible in their environments. Furthermore, it is RECOMMENDED that Authentication Keys incorporate at least 128 pseudo-random bits to minimize the risk of such attacks. In support of these recommendations, management systems SHOULD support hexadecimal input of Authentication Keys.

The mechanism described herein is not perfect and does not need to be perfect. Instead, this mechanism represents a significant increase in the effort required for an adversary to successfully attack the LDP Hello protocol while not causing undue implementation, deployment, or operational complexity.

8. IANA Considerations

The IANA is requested to as assign a new TLV from the "Multiprotocol Label Switching Architecture (MPLS) Label Switched Paths (LSPs) Parameters - TLVs" registry, "TLVs and sub-TLVs" sub- registry.

Value	Meaning	Reference
TBD	Cryptographic Authentication TLV	this document (sect 3.2)

The IANA is also requested to as assign value from the "Authentication Cryptographic Protocol ID", registry under the "Keying and Authentication for Routing Protocols (KARP) Parameters" category.

Value	Meaning	Reference
TBD	LDP Cryptographic Protocol ID	this document (sect 4)

9. Acknowledgements

The authors would like to thank Liu Xuehu for his work on background and motivation for LDP Hello authentication. The authors also would like to thank Adrian Farrel, Eric Rosen, Sam Hartman, Eric Gray, Kamran Raza and Acee Lindem for their valuable comments.

We would also like to thank the authors of RFC 5709 and RFC 6506 from where we have taken most of the cryptographic computation procedures from.

10. References

10.1. Normative References

[FIPS-180-3]
"Secure Hash Standard (SHS), FIPS PUB 180-3", October 2008.

[FIPS-198]
"The Keyed-Hash Message Authentication Code (HMAC), FIPS PUB 198", March 2002.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

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

[RFC4822] Atkinson, R. and M. Fanto, "RIPv2 Cryptographic Authentication", RFC 4822, February 2007.

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

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

[RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, October 2009.

[RFC6506] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 6506, February 2012.

10.2. Informative References

[I-D.ietf-karp-routing-tcp-analysis]
Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP and MSDP Issues According to KARP Design Guide", draft-ietf-karp-routing-tcp-analysis-07 (work in progress), April 2013.

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

Authors' Addresses

Lianshu Zheng
Huawei Technologies
China

Email: vero.zheng@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
China

Email: mach.chen@huawei.com

Manav Bhatia
Alcatel-Lucent
India

Email: manav.bhatia@alcatel-lucent.com

4-1 Sep 27, 2013, 3:04 PM
Nit: sec1, para 1 1st sentence : c/runs/run

4-2 Sep 27, 2013, 3:04 PM
Minor- sec 1, para 2: start the second sentence with "Since the Hello messages are sent with UDP and not TCP..."

4-3 Sep 27, 2013, 3:04 PM
Minor - sec 1, para 2: change "Besides a note that some configuration may help protect against bogus discovery messages, [RFC5036] does not really provide any security mechanism to protect the Hello messages." To "While some configuration guidance is given in [RFC5036] to help protect against false discovery messages, it does not provide an explicit security mechanism to protect the Hello messages."

4-4 Sep 27, 2013, 3:04 PM
Minor - global: c/Spoofing/Falsifying/

4-5 Sep 27, 2013, 3:04 PM
Minor - sec 1, para 3: please give an example of a falsified Hello with a different transport address.

4-6 Sep 27, 2013, 3:04 PM
Nit - sec 1, para 4: remove first "that" in the fist sentence

4-7 Sep 27, 2013, 3:04 PM
Minor - sec 1 para 4, 1st sentence and global: remove "Basic". The text is introducing a new term of "Basic Hellos" that doesn't exist in RFC 5036. You may be referring to "Link Hellos". See RFC 5036 sec 2.4.

4-8 Sep 27, 2013, 3:04 PM
Minor - sec 1 para 4 & global: remove "Extended" and replace with "Targeted". The text is introducing a new term of "Extended Hellos" that doesn't exist in RFC 5036. You may be referring to "Targeted Hellos". See RFC 5036 sec 2.4.

4-9 Sep 27, 2013, 3:04 PM
Nit - sec 1, para 5, sentence 1: c/message/messages/

5-1 Sep 27, 2013, 3:04 PM
Nit - global: refer consistently to "Hello message"s vs "Hellos" and "Hello packets".

5-2 Sep 27, 2013, 3:04 PM
Major - Sec 1, para 6, last sentence: remove thisMinor - sec 1 para 4, 1st sentence and global: remove "Basic". The text is introducing a new term of "Basic Hellos" that doesn't exist in RFC 5036. You may be referring to "Link Hellos". See RFC 5036 sec 2.4. last sentence, it is redundant with section 3.

6-1 Sep 27, 2013, 3:04 PM
Minor - sec 2.1, 1st para: remove 3rd sentence. Adds no value.

6-2 Sep 28, 2013, 1:29 PM
Major- s2.2, Auth Key ID: this field is conveying the algorithm AND the the secret key. It should be atomic, I.e., the algorithm ID should be one field the key another.

Major - s2.2, Auth Key ID: the composition of this field is not clear to the casual reader. Exactly how is the algorithm identified and what portion is the algorithm I'd and what part is the secret key?

7-1 Sep 28, 2013, 1:29 PM
Nit - s2.2, p6: remove "really" in the penultimate sentence.

7-2 Sep 28, 2013, 1:29 PM
Major - s2.2, p6: what are the "available mechanisms" that are being required? I.e., if some one was to test compliance to the requirements of this draft what would they test for this? Without more specifics on how to meet the sequence number requirement, the strictly increasing requirement in the paragraph above is sufficient.

7-3 Sep 28, 2013, 1:29 PM
Major - s2.3, p1: it should be clearly stated that the high order 32 bits are incremented on device boot, and low order 32 bit wrap. "This is currently on given subtly in the previous section as "wrap/boot".

7-4 Sep 28, 2013, 1:29 PM
Nit - s2.3, p1: "If by some chance..." Should start a new paragraph,

8-1 Sep 28, 2013, 1:29 PM
Major - s3, p1, sentence 1: "... the Auth Key ID maps to the authentication algorithm and the secret key used to..." Maps how exactly? As noted in earlier comments it seems as those this field should be broken into two atomic fields Auth Algorithm and Auth Key.

8-2 Sep 28, 2013, 1:29 PM
Nit - s3, p4: might be easier to see requirements as a list.
"Of the above, implementations of this specification:
- MUST include support for at least HMAC-SHA-256
- SHOULD include support for HMAC-SHA-1
- MAY include support for HMAC-SHA-384 or HMAC-SHA-512 or both."

8-3 Sep 28, 2013, 1:29 PM
Nit- s3,p3:remove this paragraph/list, it is superfluous, with the paragraph below and section 2.2 last paragraph. Further the word "includes" implies the list is not exhaustive. i.e., I can use MD5 if I wish.

Question- s3,p3: Are other authentication hashes prohibited? If not how are they identified and encoded?

9-1 Sep 28, 2013, 1:29 PM
Major- s4: this is difficult to follow.
- the section makes reference to "the two octet LDP Cryptographic Protocol ID" this field is not previously mentioned in this document. What is the reference to this ID? Where is it defined?

- The section makes reference to the "authentication key". Is this the same as the Authentication Key ID in section 2.2?

-"Other protocols using cryptographic authentication as specified herein MUST similarly append their respective Cryptographic Protocol IDs to their keys in this step." This document only specifies one protocol using crypto authentication, I.e., LDP What are the others being referred to?

- This text first uses the term "Cryptographic Protocol ID". What is this? Where is it defined? From the IANA considerations it is clearer this refers to a KARP mechanism. Please add some explanation of how this mechanism fits into the KARP framework.

- could use more explanation or a reference to additional information on the cross protocol attack problem being addressed.

- reference to "protocols sharing common keys," what does this mean? Please elaborate. Perhaps an example would help.

10-1 Sep 28, 2013, 1:29 PM

General- the RtgDir

Note - s5: I don't see any routing protocol issues with this. It should be reviewed for security and crypto issues.

12-1 Sep 28, 2013, 1:29 PM

Nit - s6.1, p1, 1st sentence: add "the" after "transmitting"
Nit - s6.1, p1, last sentence: make this a list
Nit - s6.1, p2, 1st sentence: make this a separate paragraph