Abstract

The Label Distribution Protocol (LDP), as currently defined, makes use of the TCP MD5 Signature option to protect (authentication and integrity) the LDP traffic between two adjacent LSR’s. This document specifies extensions to LDP to enable end-to-end authentication between non-adjacent LSR’s (ie not directly connected via a TCP connection) that are setting up an LSP. This mechanism also provides integrity protection of the information carried within LDP messages and protects against the malicious replay of LDP messages. The proposed mechanism requires ordered control LDP and can also be applied to CR-LDP.

1. Introduction
In its current state, LDP specifies the use of the TCP MD5 Signature option in order to provide integrity and authentication of LDP messages (in addition to protecting the TCP dialog itself). This mechanism, apart from being considered rather weak because it makes use of a simple keyed MD5 function, can only be used between two adjacent LSR’s exchanging LDP messages over a TCP connection. It does not cover situations where LSR’s exchange LDP messages via other intermediate LSR’s for setting up an LSP. It can be noted that other technologies such as IPsec or TLS could adequately be used in place of the TCP MD5 Signature option in order to provide LSR hop-by-hop security.

This document proposes a mechanism to enable such non-adjacent LSR’s to authenticate LDP messages mutually exchanged. The mechanism enables not only the receiving LER to authenticate the LDP message but also intermediate LSR’s to do so. In addition, the proposed solution also provides integrity protection of the information carried within LDP messages and anti-replay of previous messages. The proposed solution makes use of a digital signature attached to each LDP message.

As described more precisely below, this digital signature-based solution also provides data integrity and can be used to protect LDP messages in broadcast contexts (basic discovery mechanism). Finally, it also protects against replay attacks by inserting a sequence number in each LDP message.

The mechanism described herein does not provide end-to-end confidentiality, as Labelling information is not considered of a confidential nature.

It requires ordered control LDP and can be applied to CR-LDP.

A typical scenario is when, for some reason, LER A wants to set up a LSP with LER B. LER A uses ordered control LDP and sends a LDP request message to the first LSR on the path towards LER B. LER A adds authentication information to the LDP request message so that LER B can verify the identity of LER A requesting a LSP. When LER B receives this new LDP request message, it can verify the signature and hence authenticate LER A. If authentication succeeds, LER B will generate a LDP Map message (which could itself also be signed by LER B). If authentication fails, a LDP Notification message will be sent to report this authentication failure (notification which can itself be signed by LER B). The schema below illustrates this scenario, where LER B positively authenticates the LDP Request message and returns a signed LDP Map message.
2. Conventions

The keywords "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].

3. Digital Signature authentication method

This authentication method is based on the use of a digital signature that is attached to each LDP message exchanged between both LER’s. This method is bi-directional as each LER can append a signature to each LDP message it sends.

The input into the signature algorithm is basically made of parts of the LDP message that are not subject to change end-to-end, as further described in section 3.2.

Anti-replay is provided through the use of a sequence number. This sequence number is added by the message originator to the LDP message within a (new) Nonce TLV. The nonce value should take the form of an ever increasing value such as a timestamp. Because this Nonce TLV is covered by the signature value (see section 3.2.1 for details), it cannot be changed "en route" and its increasing nature prohibits replaying the same or another message with that same Nonce TLV.

3.1. Digital signature-based authentication TLV’s

3.1.1. Signature TLV

A new TLV is defined to carry the digital signature value.
<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>U-bit: the Unknown-bit must be set to '1' because intermediate LSR’s that do not know this new TLV should ignore it and process the rest of the message.</td>
</tr>
<tr>
<td>F</td>
<td>F-bit: the Forward-bit must be set to '1' because intermediate LSR’s that do not know this new TLV should transparently relay this TLV to the next LSR.</td>
</tr>
<tr>
<td></td>
<td>Length: this field indicates the total length in bytes of the following fields.</td>
</tr>
<tr>
<td></td>
<td>ID type: this field indicates the type of identifier carried in the Identifier field. Possible types are defined in 8.2.</td>
</tr>
<tr>
<td></td>
<td>ID Length: this field indicates the length in bytes of the Identifier value.</td>
</tr>
<tr>
<td></td>
<td>Identifier: this field contains an identifier of the signing entity to be used by the LSR receiver in the signature verification process (to obtain the corresponding certificate if not present in the LDP message).</td>
</tr>
<tr>
<td></td>
<td>AlgID Length: this field indicates the length in bytes of the Algorithm Identifier field.</td>
</tr>
<tr>
<td></td>
<td>Algorithm Identifier: this field contains the signature algorithm identifier. Possible algorithms are presented in section 8.1.</td>
</tr>
</tbody>
</table>
Signature value: this field contains the signature value generated as described in section 3.2.1. Its length can be deduced from the TLV Length field and other fields lengths.

3.1.2. Certificate TLV

A new TLV is defined to carry a certificate.

0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|1| Certificate (TBD)         |      Length                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Cert Type   |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

U-bit: the Unknown-bit must be set to ‘1’ because intermediate LSR’s that do not know this new TLV should ignore it and process the rest of the message.

F-bit: the Forward-bit must be set to ‘1’ because intermediate LSR’s that do not know this new TLV should transparently relay this TLV to the next LSR.

Length: this field indicates the length in bytes of the Cert Type and Certificate Value fields.

Cert Type: this field indicates the type of certificate carried in this TLV. Possible values are defined in RFC 2408 section 3.9.

Certificate Value: this field contains the certificate value.

The Certificate TLV must appear anywhere in the LDP message prior to the Signature TLV. Several Certificate TLV’s may be inserted into the LDP message if felt necessary for the LSR receiver to be able to verify the signature. Such a list of Certificate TLV’s SHOULD be inserted in a continuous sequence of TLV’s (ie no other TLV type should appear in between).

3.1.3. Nonce TLV

A new TLV is defined to carry a nonce value.
U-bit: the Unknown-bit must be set to ‘1’ because intermediate LSR’s that do not know this new TLV should ignore it and process the rest of the message.

F-bit: the Forward-bit must be set to ‘1’ because intermediate LSR’s that do not know this new TLV should transparently relay this TLV to the next LSR.

Length: this field indicates the length in bytes of the nonce value field.

Nonce value: this field contains a unique value that must increase for every message exchanged between two LSR’s. A timestamp value is an example of such an increasing counter. Both sides must keep track of the last value used by the partner to be able to verify the increasing nature of the nonce value received.

The Nonce TLV must appear prior to the Signature TLV but not necessarily next to.

3.2. Digital signature-based authentication procedure

3.2.1. Sender

When sending a LDP message, the originating LER creates and inserts a Nonce TLV. Certificate TLV’s are also inserted if felt adequate to ease the receiving LER’s signature verification. The Signature TLV is finally appended to the LDP message after having calculated the signature value. The Identifier field in the Signature TLV is used to carry the signer’s identity so that the receiving LSR can match the certificate used for signature verification with the signer’s identity.

The input into the signature algorithm depends on the type of message.

For LDP Request and Label Mapping messages, the input is the byte
stream built as described in the following figure (the Adapted Length represents the actual length of this byte stream, not the length of the real LDP Request message). The Nonce TLV prevents re-using the signature value for another LDP message.

```
+--------+-+-----------------------------+-------------------------------+
|0|      Message type           |    Adapted Length             |
+--------+-+-----------------------------+-------------------------------+
                                           Nonce TLV
                                           |
+---------------------------------------------------------------+
                                           FEC TLV
+---------------------------------------------------------------+
```

For LDP Notification messages, the input is the byte stream built as described in the following figure.

```
+--------+-+-----------------------------+-------------------------------+
|0|      Notification           |    Adapted Length             |
+--------+-+-----------------------------+-------------------------------+
                                           Nonce TLV
                                           |
+---------------------------------------------------------------+
                                           Status TLV
+---------------------------------------------------------------+
```

3.2.2. Receiver

Upon receiving a LDP message, the destination LER verifies the signature value present in the Signature TLV. Verification of the signature can be made easier thanks to the use of certificates in Certificate TLV’s if present. The LSR should take care of ensuring that any certificate taken from a Certificate TLV is still valid by consulting some up-to-date CRL (Certificate Revocation List).

The input into the signature verification algorithm is built similarly to what is described in section 3.2.1 above.

3.2.3. Intermediary

This mechanism enables intermediate LSR’s to authenticate LDP messages being relayed. The intermediate LSR can indeed easily
verify the digital signature attached to the LDP message as long as it has a copy of the sender’s public key or can find it in a repository or LDP message itself.

3.3. Discussion

This mechanism relies on the use of asymmetric public-key signature algorithms such as RSA and DSA. Each LSR must therefore possess its own key pair. Some form of Public Key Infrastructure (PKI) is also necessary to create certificates for LSR’s public keys and distribute those certificates. Depending on the scenarios, a minimal form of PKI may be to distribute the certificates within the LDP messages themselves, as the Certificate TLV allows it. LSR’s must also be preconfigured with the (possibly several) certification Authority (CA) root key(s).

This mechanism does not require a prior shared knowledge between both LSR’s (apart from having a common CA or being able to find a certification path between their two CA’s).

The anti-replay mechanism can only work if the receiver system is able to realize that the nonce value increases with each LDP message received. In case of system failure so that the last sequence number received is forgotten, there is a potential for replay.

It is a matter of local policy for a LSR (and implicitly mutual agreement between both LSR’s) to decide whether to use end-to-end authentication when sending/receiving LDP messages with a particular partner.


A new Status Code value is required in the LDP Notification message Status TLV. This code is used by an LSR to announce that a previously received LDP message has failed authentication.

The U-bit and F-bit in the Status TLV must be set to ‘1’ so as to enable transparent relay of the Status TLV by LSR’s.

The E-bit in the Status Code field of the Status TLV can be set to ‘0’ or ‘1’ depending on the local policy of the LSR.

The F-bit in the Status Code field of the Status TLV MUST be set to ‘1’, in alignment with the F-bit at the Status TLV level.

5. Editor’s notes

- the signature algorithm identifier could be defined as a simple
integer, but it then requires some IANA registration.

6. IANA considerations

New TLV types defined in this document must be assigned values by
IANA. Also, a new status code must be defined to identify
authentication failure in Notification messages.

7. Security considerations

This specification entirely deals with security issues.

8. Algorithms and identifiers

8.1. Signature algorithms

Implementations MUST support the use of DSA-with-SHA1 algorithm. The
exact encoding of the resulting signature and the algorithm
identifier (id-dsa-with-sha1) can be found in RFC 2095 section 7.2.2.

8.2. Identifiers types

The following types of signer's identifiers are defined.

<table>
<thead>
<tr>
<th>Value</th>
<th>Type 0</th>
<th>IPv4 address 1</th>
<th>IPv6 address 2</th>
<th>FQDN</th>
</tr>
</thead>
</table>

References

[LDP] Andersson et al., "LDP Specification", draft-ietf-mpls-ldp-
11.txt, August 2000.

Acknowledgments

The authors would like to thank Peter De Schrijver for having produced
an original version of this document.

Authors' Addresses

Jeremy De Clercq
Alcatel
Fr. Wellesplein 1, 2018 Antwerpen, Belgium
E-mail: jeremy.de_clercq@alcatel.be
Olivier Paridaens  
Alcatel  
Fr. Wellesplein 1, 2018 Antwerpen, Belgium  
E-mail: olivier.paridaens@alcatel.be

Yves T'joens  
Alcatel  
Fr. Wellesplein 1, 2018 Antwerpen, Belgium  
E-mail: yves.tjoens@alcatel.be

Peter De Schrijver

Full Copyright Statement

"Copyright (C) The Internet Society (date). All Rights Reserved. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."