Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 10, 2016

M. Boucadair
C. Jacquenet
Orange

March 9, 2016

LISP Mapping Service Discovery at Large
draft-boucadair-lisp-idr-ms-discovery-01

Abstract

Locator/ID Separation Protocol (LISP) operation relies upon a mapping mechanism that is used by ingress/egress Tunnel Routers (xTR) to forward traffic over the LISP network. The ability of dynamically discovering the Map-Resolver and Map-Server entities that provide such mapping services is meant to facilitate global LISP operation (automatic discovery of Map-Resolvers and Map-Servers).

This document specifies a BGP Extended Communities attribute that can be used to dynamically discover LISP Mapping Systems of different domains.

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 September 10, 2016.

Table of Contents

1. Introduction

Locator/ID Separation Protocol (LISP, [RFC6830]) operation relies upon a mapping mechanism that is used by ingress/egress Tunnel Routers (xTR) to forward traffic over the LISP network. The ability of dynamically discovering the Map-Resolver and Map-Server entities that provide such mapping services is meant to facilitate global LISP operation (automatic discovery of Map-Resolvers and Map-Servers).

Within this document, a Mapping System provides the LISP mapping service [RFC6833]. Map-Resolvers, Map-Servers, and other components may be part of a Mapping System such as authorization, subscription profiles, etc. These components are considered as black boxes; only the external behavior of the Mapping System is in scope.

Distinct LISP mapping systems may emerge if LISP users or network operators who solicit or manage the Mapping System want to avoid some region-centric systems, for example, or if they want to position themselves as a core provider of the Mapping System. The lack of

clear policies of the management and operation of the LISP Mapping Systems may encourage such practices.

This document assumes a hierarchy in the Mapping System organisation for business, governance, control, and regulatory purposes in particular. In such contexts, the document assumes that a Mapping System may maintain a portion of or a global mapping table.

Because of its experimental nature of LISP and the various platforms LISP operation relies upon (like the platforms used by the LISP mapping systems) should encourage innovation by testing new services that may take advantage of LISP in inter-domain deployment scenarios without requiring the participation of all LISP-enabled domains. Such approach is also meant to avoid any risk of freezing LISP developments.

Because the design and operation of a consistent Mapping System is critical for the adoption of LISP at large scale, this document advocates for means to dynamically discover other Mapping Systems that are open to cooperate in inter-domain LISP deployment scenarios, typically .

Deploying LISP for inter-domain use cases may raise the following issues:

Issue#1: A LISP domain may need to discover available Mapping Systems so that it can rely upon them to extend the reachability scope.

Issue#2: Various Mapping Systems can be deployed over the Internet. These Mapping Systems need to interconnect to extend the reachability scope and avoid pressure on PTR (Proxy Tunnel Router) devices. Also, various Mapping Systems encourage the enforcement of policies that aim at optimizing LISP forwarding: for example, policies that consist in avoiding the solicitation of specific domains/ASes.

Issue#3: Distinct flavors of Mapping Systems may be deployed. These mappings may not rely on the same database mapping system (e.g., NERD, ALT, CONS, etc.). As such, a clear interface to ease interconnection between these realms is needed. Standard solutions to discover Mapping Systems capabilities are likely to ease the interconnection of Mapping Systems.

Issue#4: Security concerns may arise during the discovery of the available Mapping Systems: for example, a given Mapping System may deny access from another domain, or available Mapping Systems need to make sure that they are entitled to exchange information with

one another or that an xTR of a given LISP network is entitled to solicit a mapping system of another LISP network, etc.

An efficient and scalable deployment of LISP within an inter-domain context for traffic engineering purposes, in particular, relies heavily on the availability of an inter-domain mapping system that spans several domains. From this perspective, the success of a global LISP adoption and deployment will mainly depend on how LISP-enabled domains will graft to existing mapping systems that can guarantee a global reachability scope. To minimize the risk of a fragmented Mapping System that would jeopardize the overall efficiency of an inter-domain LISP routing system, there is a need to encourage and facilitate the coordination of participating Mapping Systems.

This document relies on extended BGP communities [RFC4360] to advertise that a given domain supports the LISP Mapping Service. A contact IPv4 address and/or IPv6 address are also included in the attribute so that remote LISP Mapping Systems or LISP domains may initiate negotiation cycles for the sake of LISP Mapping System Interconnection or subscription to the Mapping Service offered by that Mapping System.

Section 3 specifies a solution for the discovery of LISP Mapping Systems that are deployed in distinct administrative domains. This BGP-based solution assumes that domains that support a LISP Mapping Service will use the BGP Extended Communities attribute to inform other domains about the support of the service. EIDs that can be serviced with LISP will be tagged accordingly. Note that an EID can be serviced by multiple Mapping Systems. Remote LISP Mapping Systems will rely upon that BGP-based advertising capability to discover the existence and the status of other Mapping Systems. Once a Mapping System is discovered, a local Mapping System can establish an interconnection agreement with that remote Mapping System. The contact IP address provided as part of the BGP Extended Communities attribute will be used to contact a remote Mapping System to request for further LISP-related capabilities, possibly negotiate an interconnection agreement and, consequently, extend the scope of the networks that can be serviced using LISP. Also, leaf LISP-aware networks can rely upon the information carried in the BGP Extended Communities attribute to discover Mapping Systems that may be solicited to invoke their mapping service. Subscription cycles may then be considered.

2. Rationale

This document focuses on the discovery of LISP Mapping Systems that are deployed in distinct administrative domains.

The rationale is as follows:

1. Announce: Domains that support a LISP Mapping Service will use the BGP Extended Communities attribute (Section 3) to inform other domains about the support of the service. EIDs that can be serviced with LISP can be tagged accordingly. Note that an EID can be serviced by multiple Mapping Systems.

2. Discover: Remote LISP Mapping Systems will rely upon that BGP-based advertising capability (Section 3) to discover the existence of other Mapping Systems.

3. Negotiate/Interconnect/Invoke: The contact IP address provided as part of the BGP Extended Communities attribute (Section 3) will be used to contact a remote Mapping System to request for further LISP-related capabilities, possibly negotiate an interconnection agreement and, consequently, extend the scope of the networks that can be serviced using LISP.

4. Negotiate/Subscribe/Invoke: Also, leaf LISP-aware networks can rely upon the information carried in the BGP Extended Communities attribute to discover Mapping Systems that may be solicited to invoke their mapping service. Subscription cycles may then be considered.

Only the first two steps are in scope of this document; the remaining steps can be achieved by other means such as [I-D.boucadair-connectivity-provisioning-protocol].

3. LISP Mapping System Target BGP Extended Community

The LISP Mapping System Target Community identifies one or more Mapping System contact points that can receive mapping system interconnect and/or subscription requests. These contact points are identified with IPv4 and/or IPv6 addresses.

The LISP Mapping System Target Community is of an extended type. As such, the behavior specified in Section 6 of [RFC4360] applies to the LISP Mapping System Target Community.

The presence of this community is an explicit indication that associated networks can be managed by a LISP Mapping System that is reachable at the addresses carried in the attribute.

This document reuses the Transitive IPv4-Address-Specific Extended Community [RFC4360] and Transitive IPv6-Address-Specific Extended Community [RFC5701] for the purpose of this document. Dedicated sub-types are to be allocated (see Section 5).

The Global Administrator field MUST be set to an IP address of the Mapping System. This address MUST be configured on the originating BGP speaker.

The "Local Administrator" field of the LISP Mapping System Target Community is used to encode an identifier of the Mapping System. Considerations about the assignment of globally unique identifiers to LISP Mapping Systems are out of scope. A configurable parameter may be supported by BGP implementations to provide the value carried in the "Local Administrator" field. If no identifier is configured on the originating BGP speaker, the "Local Administrator" field MUST be set to 0.

4. Security Considerations

This document does not introduce any additional security issues other than those discussed in [RFC4360] and [RFC5701].

5. IANA Considerations

According to [RFC7153], this document requests the assignment of a sub-type in the "0x00-0xbf" range from the Transitive IPv4-Address-Specific Extended Community Sub-Types registry available at http://www.iana.org/assignments/bgp-extended-communities/bgp-extended-communities.xml#trans-ipv4:

Type Value	Name	Reference
TBA	LISP Mapping System Target	[This-Document]

Also, this document requests the assignment of a sub-type from the Transitive IPv6-Address-Specific Extended Community Types registry available at http://www.iana.org/assignments/bgp-extended-communities/bgp-extended-communities.xml#trans-ipv6:

Type Value	Name	Reference
TBA	LISP Mapping System Target	[This-Document]

6. Acknowledgments

This work is partly funded by ANR LISP-Lab project #ANR-13-INFR-009-X.

7. References

7.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <http://www.rfc-editor.org/info/rfc4360>.

[RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <http://www.rfc-editor.org/info/rfc5701>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC7153] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", RFC 7153, DOI 10.17487/RFC7153, March 2014, <http://www.rfc-editor.org/info/rfc7153>.

7.2. Informative references

[I-D.boucadair-connectivity-provisioning-protocol] Boucadair, M., Jacquenet, C., Zhang, D., and P. Georgatsos, "Connectivity Provisioning Negotiation Protocol (CPNP)", draft-boucadair-connectivity-provisioning-protocol-10 (work in progress), September 2015.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <http://www.rfc-editor.org/info/rfc6833>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet
Orange
Rennes 35000
France

EMail: christian.jacquenet@orange.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 17, 2016

M. Boucadair
C. Jacquenet
Orange

March 16, 2016

IANA Registry for LISP Packet Type Allocations
draft-boucadair-lisp-type-iana-00

Abstract

This document defines a registry for LISP Packet Type allocations. It also specifies a shared LISP message type for experimentation purposes.

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 September 17, 2016.

Table of Contents

1. Introduction

The Locator/ID Separation Protocol (LISP, [RFC6830]) base specification defines a set of primitives that are identified with a packet type code. Several extensions have been proposed to add more LISP functionalities. For example, new message types are proposed in [I-D.ietf-lisp-ddt], [I-D.zhao-lisp-mn-extension], [I-D.boucadair-lisp-bulk], [I-D.boucadair-lisp-subscribe], or [I-D.boucadair-lisp-ms-assisted-forwarding]. It is expected that additional LISP extensions will be proposed in the future.

In order to ease the tracking of LISP message types, this document proposes to create a "LISP Packet Types" IANA registry (see Section 4).

Because of the limited type space [RFC6830], this document specifies a shared LISP message type for experimentation purposes and proposes a procedure for registering LISP experiment identifiers (see Section 2) that make use of additional LISP capabilities associated with this message type. Concretely, one single LISP message type code is dedicated to experiments; experiment IDs are used to uniquely identify a given LISP experimental message. These identifiers are selected by the author(s) of the corresponding LISP specification that introduces a new experimental message type.

2. LISP Message Type for Experimentations

Figure 1 depicts a common LISP experimental message type. The type field MUST be set to 15 (see Section 4).

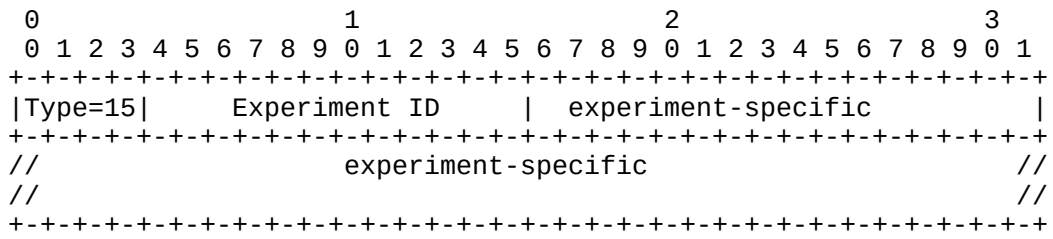


Figure 1: Common LISP Experimental Message Type

The "Experiment ID" field conveys a unique identifier that is assigned on a First Come, First Served (FCFS) basis [RFC5226]. These identifiers are registered with IANA (see Section 4.1).

The exact structure of the 'experiment-specific' portion of the message is specified in the corresponding specification document.

3. Security Considerations

This document does not introduce any additional security issues other than those discussed in [RFC6830].

4. IANA Considerations

IANA is requested to create a new protocol registry for LISP Packet Types, numbered 0-15. The registry must be initially populated with the following values ([RFC6830]):

Message	Code
Reserved	0
LISP Map-Request	1
LISP Map-Reply	2
LISP Map-Register	3
LISP Map-Notify	4
LISP Encapsulated Control Message	8
LISP Experimental Message	15

The values in the ranges 5-7 and 9-14 can be assigned via Standards Action [RFC5226].

The value 15 is reserved for Experimental Use [RFC5226].

4.1. Experiment IDs

IANA is requested to create a "LISP Experimental Message Experiment Identifiers" registry.

Entries are assigned on a First Come, First Served (FCFS) basis [RFC5226].

IANA should impose no requirements on making a registration other than indicating the desired experiment ID and providing a point of contact. Providing a short description (together with an acronym, if relevant) of the foreseen usage of the experimental message is encouraged.

5. Acknowledgments

This work is partly funded by ANR LISP-Lab project #ANR-13-INFR-009-X.

The shared experiment ID is inspired by [RFC6994].

6. References

6.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

6.2. Informative References

[I-D.boucadair-lisp-bulk] Boucadair, M. and C. Jacquenet, "LISP Mapping Bulk Retrieval", draft-boucadair-lisp-bulk-01 (work in progress), March 2016.

[I-D.boucadair-lisp-ms-assisted-forwarding]
Boucadair, M. and C. Jacquenet, "Mapping System-Assisted Forwarding for Inter-Domain LISP Deployments", draft-boucadair-lisp-ms-assisted-forwarding-00 (work in progress), September 2015.

[I-D.boucadair-lisp-subscribe]
Boucadair, M. and C. Jacquenet, "Improving Mapping Services in LISP Networks", draft-boucadair-lisp-subscribe-02 (work in progress), March 2016.

[I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-03 (work in progress), April 2015.

[I-D.zhao-lisp-mn-extension]
Wang, J., Meng, Y., and N. Zhao, "LISP Mobile Node extension", draft-zhao-lisp-mn-extension-02 (work in progress), October 2011.

[RFC6994] Touch, J., "Shared Use of Experimental TCP Options", RFC 6994, DOI 10.17487/RFC6994, August 2013, <http://www.rfc-editor.org/info/rfc6994>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet
Orange
Rennes 35000
France

EMail: christian.jacquenet@orange.com

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: June 6, 2016

D. Farinacci
lispers.net
B. Weis
cisco Systems

December 4, 2015

LISP Data-Plane Confidentiality
draft-ietf-lisp-crypto-03

Abstract

This document describes a mechanism for encrypting LISP encapsulated traffic. The design describes how key exchange is achieved using existing LISP control-plane mechanisms as well as how to secure the LISP data-plane from third-party surveillance attacks.

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 June 6, 2016.

Copyright Notice

Table of Contents

1. Introduction

The Locator/ID Separation Protocol [RFC6830] defines a set of functions for routers to exchange information used to map from non-routable Endpoint Identifiers (EIDs) to routable Routing Locators (RLOCs). LISP ITRs and PITRs encapsulate packets to ETRs and RTRs. Packets that arrive at the ITR or PITR are typically not modified. Which means no protection or privacy of the data is added. If the source host encrypts the data stream then the encapsulated packets can be encrypted but would be redundant. However, when plaintext packets are sent by hosts, this design can encrypt the user payload to maintain privacy on the path between the encapsulator (the ITR or PITR) to a decapsulator (ETR or RTR). The encrypted payload is unidirectional. However, return traffic uses the same procedures but with different key values by the same xTRs or potentially different xTRs when the paths between LISP sites are asymmetric.

This draft has the following requirements for the solution space:

- Do not require a separate Public Key Infrastructure (PKI) that is out of scope of the LISP control-plane architecture.
- The budget for key exchange MUST be one round-trip time. That is, only a two packet exchange can occur.
- Use symmetric keying so faster cryptography can be performed in the LISP data plane.
- Avoid a third-party trust anchor if possible.
- Provide for rekeying when secret keys are compromised.
- Support Authenticated Encryption with packet integrity checks.
- Support multiple cipher suites so new crypto algorithms can be easily introduced.

2. Overview

The approach proposed in this draft is to NOT rely on the LISP mapping system (or any other key infrastructure system) to store security keys. This will provide for a simpler and more secure mechanism. Secret shared keys will be negotiated between the ITR and the ETR in Map-Request and Map-Reply messages. Therefore, when an ITR needs to obtain the RLOC of an ETR, it will get security material to compute a shared secret with the ETR.

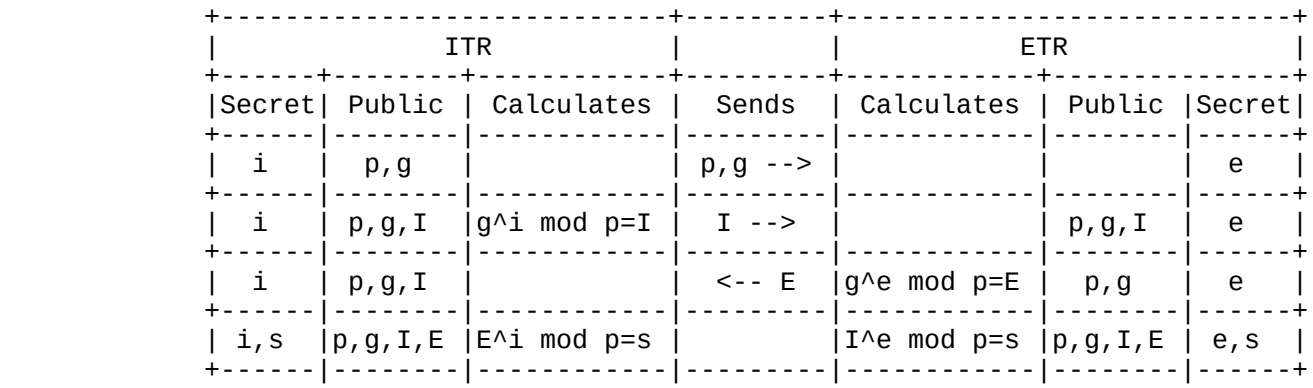
The ITR can compute 3 shared-secrets per ETR the ITR is encapsulating to. And when the ITR encrypts a packet before encapsulation, it will identify the key it used for the crypto calculation so the ETR knows which key to use for decrypting the packet after decapsulation. By using key-ids in the LISP header, we can also get real-time rekeying functionality.

When an ETR (when it is also an ITR) encapsulates packets to this ITR (when it is also an ETR), a separate key exchange and shared-secret computation is performed. The key management described in this documemnt is unidirectional from the ITR (the encapsulator) to the ETR (the decapsultor).

3. Diffie-Hellman Key Exchange

LISP will use a Diffie-Hellman [RFC2631] key exchange sequence and computation for computing a shared secret. The Diffie-Hellman parameters will be passed via Cipher Suite code-points in Map-Request and Map-Reply messages.

Here is a brief description how Diff-Hellman works:

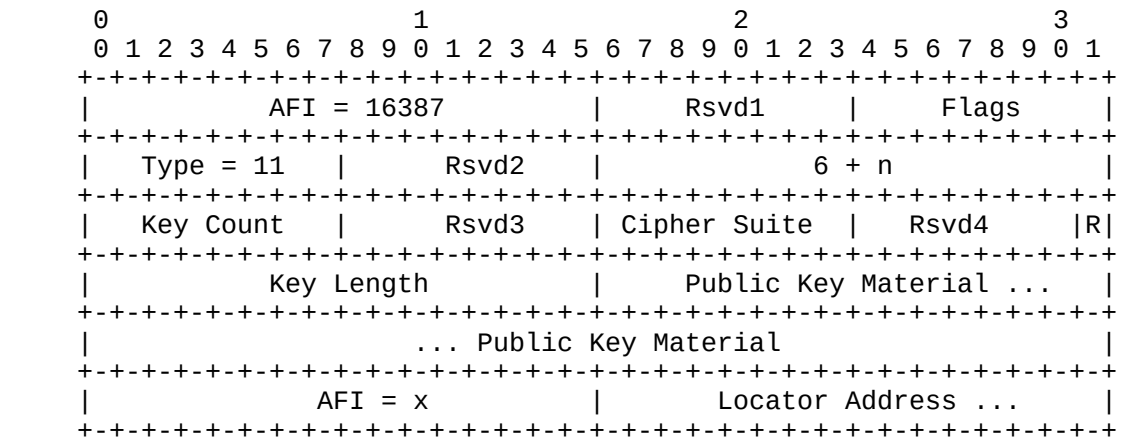


Public-key exchange for computing a shared private key [DH]

Diffie-Hellman parameters ’p’ and ’g’ must be the same values used by the ITR and ETR. The ITR computes public-key ’I’ and transmits ’I’ in a Map-Request packet. When the ETR receives the Map-Request, it uses parameters ’p’ and ’g’ to compute the ETR’s public key ’E’. The ETR transmits ’E’ in a Map-Reply message. At this point, the ETR has enough information to compute ’s’, the shared secret, by using ’I’ as the base and the ETR’s private key ’e’ as the exponent. When the ITR receives the Map-Reply, it uses the ETR’s public-key ’E’ with the ITR’s private key ’i’ to compute the same ’s’ shared secret the ETR computed. The value ’p’ is used as a modulus to create the width of the shared secret ’s’.

4. Encoding and Transmitting Key Material

The Diffie-Hellman key material is transmitted in Map-Request and Map-Reply messages. Diffie-Hellman parameters are encoded in the LISP Security Type LCAF [LCAF].



Cipher Suite field contains DH Key Exchange and Cipher/Hash Functions

The 'Key Count' field encodes the number of {'Key-Length', 'Key-Material'} fields included in the encoded LCAF. The maximum number of keys that can be encoded are 3, each identified by key-id 1, followed by key-id 2, an finally key-id 3.

The 'R' bit is not used for this use-case of the Security Type LCAF but is reserved for [LISP-DDT] security.

```
  Cipher Suite 0:
    Reserved

  Cipher Suite 1:
    Diffie-Hellman Group: 2048-bit MODP [RFC3526]
    Encryption:           AES with 128-bit keys in CBC mode [AES-CBC]
    Integrity:            Integrated with [AES-CBC] AEAD [RFC5116] encryption
    IV length:            16 bytes

  Cipher Suite 2:
    Diffie-Hellman Group: 256-bit Elliptic-Curve 25519 [CURVE25519]
    Encryption:           AES with 128-bit keys in CBC mode [AES-CBC]
    Integrity:            HMAC-SHA1-96 [RFC2404]
    IV length:            16 bytes

  Cipher Suite 3:
    Diffie-Hellman Group: 2048-bit MODP [RFC3526]
    Encryption:           AES with 128-bit keys in GCM mode [AES-GCM]
    Integrity:            Integrated with [AES-GCM] AEAD [RFC5116] encryption
    IV length:            12 bytes

 Cipher Suite 4:
    Diffie-Hellman Group: 3072-bit MODP [RFC3526]
    Encryption:           AES with 128-bit keys in GCM mode [AES-GCM]
    Integrity:            Integrated with [AES-GCM] AEAD [RFC5116] encryption
    IV length:            12 bytes

 Cipher Suite 5:
    Diffie-Hellman Group: 256-bit Elliptic-Curve 25519 [CURVE25519]
    Encryption:           AES with 128-bit keys in GCM mode [AES-GCM]
    Integrity:            Integrated with [AES-GCM] AEAD [RFC5116] encryption
    IV length:            12 bytes

 Cipher Suite 6:
    Diffie-Hellman Group: 256-bit Elliptic-Curve 25519 [CURVE25519]
    Encryption/Integrity: Chacha20-Poly1305 [CHACHA-POLY] [RFC7539]
    Integrity:            Integrated with Chacha20-Poly1305 AEAD [RFC1116] encry
ption
    IV length:            8 bytes
```

The "Public Key Material" field contains the public key generated by one of the Cipher Suites defined above. The length of the key in octets is encoded in the "Key Length" field.

When an ITR or PITR send a Map-Request, they will encode their own RLOC in the Security Type LCAF format within the ITR-RLOCs field. When a ETR or RTR sends a Map-Reply, they will encode their RLOCs in

Security Type LCAF format within the RLOC-record field of each EID-record supplied.

If an ITR or PITR sends a Map-Request with the Security Type LCAF included and the ETR or RTR does not want to have encapsulated traffic encrypted, they will return a Map-Reply with no RLOC records encoded with the Security Type LCAF. This signals to the ITR or PITR that it should not encrypt traffic (it cannot encrypt traffic anyways since no ETR public-key was returned).

Likewise, if an ITR or PITR wish to include multiple key-ids in the Map-Request but the ETR or RTR wish to use some but not all of the key-ids, they return a Map-Reply only for those key-ids they wish to use.

5. Shared Keys used for the Data-Plane

When an ITR or PITR receives a Map-Reply accepting the Cipher Suite sent in the Map-Request, it is ready to create data plane keys. The same process is followed by the ETR or RTR returning the Map-Reply.

The first step is to create a shared secret, using the peer’s shared Diffie-Hellman Public Key Material combined with device’s own private keying material as described in Section 3. The Diffie-Hellman group used is defined in the cipher suite sent in the Map-Request and copied into the Map-Reply.

The resulting shared secret is used to compute an AEAD-key for the algorithms specified in the cipher suite. A Key Derivation Function (KDF) in counter mode as specified by [NIST-SP800-108] is used to generate the data-plane keys. The amount of keying material that is derived depends on the algorithms in the cipher suite.

The inputs to the KDF are as follows:

- KDF function. This is HMAC-SHA-256.
- A key for the KDF function. This is the computed Diffie-Hellman shared secret.
- Context that binds the use of the data-plane keys to this session. The context is made up of the following fields, which are concatenated and provided as the data to be acted upon by the KDF function.

Context:

- A counter, represented as a two-octet value in network-byte order.

- The null-terminated string "lisp-crypto".
- The ITR's nonce from the the Map-Request the cipher suite was included in.
- The number of bits of keying material required (L), represented as a two-octet value in network byte order.

The counter value in the context is first set to 1. When the amount of keying material exceeds the number of bits returned by the KDF function, then the KDF function is called again with the same inputs except that the counter increments for each call. When enough keying material is returned, it is concatenated and used to create keys.

For example, AES with 128-bit keys requires 16 octets (128 bits) of keying material, and HMAC-SHA1-96 requires another 16 octets (128 bits) of keying material in order to maintain a consistent 128-bits of security. Since 32 octets (256 bits) of keying material are required, and the KDF function HMAC-SHA-256 outputs 256 bits, only one call is required. The inputs are as follows:

key-material = HMAC-SHA-256(dh-shared-secret, context)

where: context = 0x0001 || "lisp-crypto" || <itr-nonce> || 0x0100

In contrast, a cipher suite specifying AES with 256-bit keys requires 32 octets (256 bits) of keying material, and HMAC-SHA256-128 requires another 32 octets (256 bits) of keying material in order to maintain a consistent 256-bits of security. Since 64 octets (512 bits) of keying material are required, and the KDF function HMAC-SHA-256 outputs 256 bits, two calls are required.

key-material-1 = HMAC-SHA-256(dh-shared-secret, context)

where: context = 0x0001 || "lisp-crypto" || <itr-nonce> || 0x0200

key-material-2 = HMAC-SHA-256(dh-shared-secret, context)

where: context = 0x0002 || "lisp-crypto" || <itr-nonce> || 0x0200

key-material = key-material-1 || key-material-2

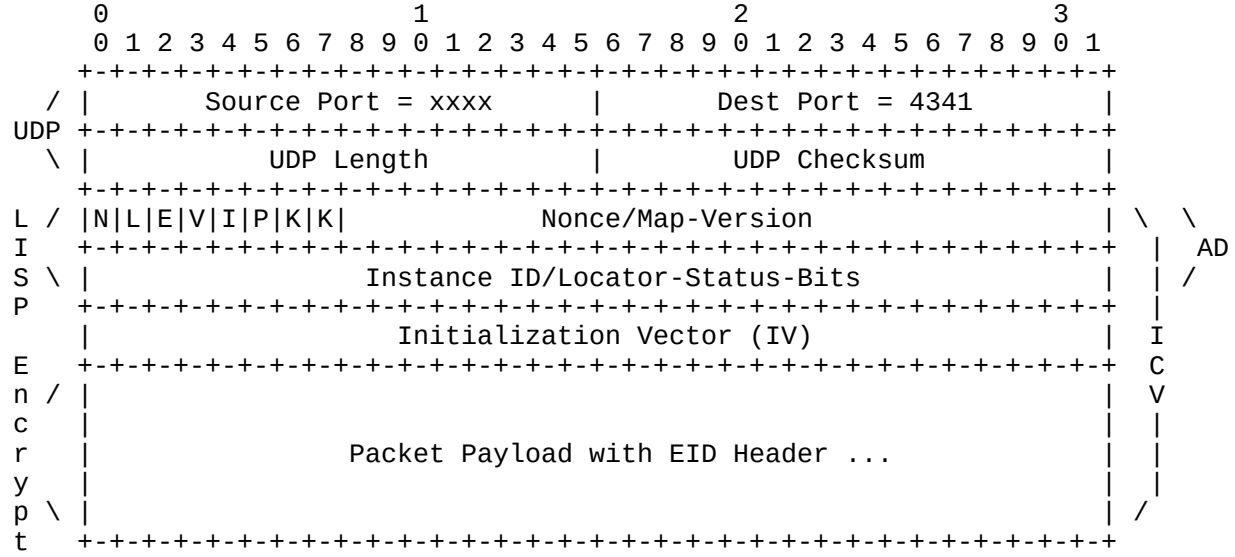
If the key-material is longer than the required number of bits (L), then only the most significant L bits are used.

From the derived key-material, the most significant 256 bits are used for the AEAD-key by AEAD ciphers. The 256-bit AEAD-key is divided

into a 128-bit encryption key and a 128-bit integrity-check key internal to the cipher used by the ITR.

6. Data-Plane Operation

The LISP encapsulation header [RFC6830] requires changes to encode the key-id for the key being used for encryption.



K-bits indicate when packet is encrypted and which key used

When the KK bits are 00, the encapsulated packet is not encrypted. When the value of the KK bits are 1, 2, or 3, it encodes the key-id of the secret keys computed during the Diffie-Hellman Map-Request/ Map-Reply exchange. When the KK bits are not 0, the payload is prepended with an Initialization Vector (IV). The length of the IV field is based on the cipher suite used. Since all cipher suites defined in this document do Authenticated Encryption (AEAD), an ICV field does not need to be present in the packet since it is included in the ciphertext. The Additional Data (AD) used for the ICV is shown above and includes the LISP header, the IV field and the packet payload.

When an ITR or PITR receives a packet to be encapsulated, they will first decide what key to use, encode the key-id into the LISP header, and use that key to encrypt all packet data that follows the LISP header. Therefore, the outer header, UDP header, and LISP header travel as plaintext.

There is an open working group item to discuss if the data encapsulation header needs change for encryption or any new applications. This draft proposes changes to the existing header so experimentation can continue without making large changes to the data-plane at this time.

7. Procedures for Encryption and Decryption

When an ITR, PITR, or RTR encapsulate a packet and have already computed an AEAD-key (detailed in section Section 5) that is associated with a destination RLOC, the following encryption and encapsulation procedures are performed:

1. The encapsulator creates an IV and prepends the IV value to the packet being encapsulated. For GCM and Chacha cipher suites, the IV is incremented for every packet (beginning with a value of 1 in the first packet) and sent to the destination RLOC. For CBC cipher suites, the IV is a new random number for every packet sent to the destination RLOC. For the Chacha cipher suite, the IV is an 8-byte random value that is appended to a 4-byte counter that is incremented for every packet (beginning with a value of 1 in the first packet).

2. Next encrypt with cipher function AES or Chacha20 using the AEAD-key over the packet payload following the AEAD specification referenced in the cipher suite definition. This does not include the IV. The IV must be transmitted as plaintext so the decrypter can use it as input to the decryption cipher. The payload should be padded to an integral number of bytes a block cipher may require. The result of the AEAD operation may contain an ICV, the size of which is defined by the referenced AEAD specification. Note that the AD (i.e. the LISP header exactly as will be prepended in the next step and the IV) must be given to the AEAD encryption function as the "associated data" argument.

3. Prepend the LISP header. The key-id field of the LISP header is set to the key-id value that corresponds to key-pair used for the encryption cipher.

4. Lastly, prepend the UDP header and outer IP header onto the encrypted packet and send packet to destination RLOC.

When an ETR, PETR, or RTR receive an encapsulated packet, the following decapsulation and decryption procedures are performed:

1. The outer IP header, UDP header, LISP header, and IV field are stripped from the start of the packet. The LISP header and IV

are retained and given to the AEAD decryption operation as the "associated data" argument.

2. The packet is decrypted using the AEAD-key and the IV from the packet. The AEAD-key is obtained from a local-cache associated with the key-id value from the LISP header. The result of the decryption function is a plaintext packet payload if the cipher returned a verified ICV. Otherwise, the packet has been tampered with, is dropped, and an optional log message may be issued. If the AEAD specification included an ICV, the AEAD decryption function will locate the ICV in the ciphertext and compare it to a version of the ICV that the AEAD decryption function computes. If the computed ICV is different than the ICV located in the ciphertext, then it will be considered tampered.

3. If the packet was not tampered with, the decrypted packet is forwarded to the destination EID.

8. Dynamic Rekeying

Since multiple keys can be encoded in both control and data messages, an ITR can encapsulate and encrypt with a specific key while it is negotiating other keys with the same ETR. Soon as an ETR or RTR returns a Map-Reply, it should be prepared to decapsulate and decrypt using the new keys computed with the new Diffie-Hellman parameters received in the Map-Request and returned in the Map-Reply.

RLOC-probing can be used to change keys or cipher suites by the ITR at any time. And when an initial Map-Request is sent to populate the ITR’s map-cache, the Map-Request flows across the mapping system where a single ETR from the Map-Reply RLOC-set will respond. If the ITR decides to use the other RLOCs in the RLOC-set, it MUST send a Map-Request directly to negotiate security parameters with the ETR. This process may be used to test reachability from an ITR to an ETR initially when a map-cache entry is added for the first time, so an ITR can get both reachability status and keys negotiated with one Map-Request/Map-Reply exchange.

A rekeying event is defined to be when an ITR or PITR changes the cipher suite or public-key in the Map-Request. The ETR or RTR compares the cipher suite and public-key it last received from the ITR for the key-id, and if any value has changed, it computes a new public-key and cipher suite requested by the ITR from the Map-Request and returns it in the Map-Reply. Now a new shared secret is computed and can be used for the key-id for encryption by the ITR and decryption by the ETR. When the ITR or PITR starts this process of negotiating a new key, it must not use the corresponding key-id in encapsulated packets until it receives a Map-Reply from the ETR with

the same cipher suite value it expects (the values it sent in a Map-Request).

Note when RLOC-probing continues to maintain RLOC reachability and rekeying is not desirable, the ITR or RTR can either not include the Security Type LCAF in the Map-Request or supply the same key material as it received from the last Map-Reply from the ETR or RTR. This approach signals to the ETR or RTR that no rekeying event is requested.

9. Future Work

For performance considerations, newer Elliptic-Curve Diffie-Hellman (ECDH) groups can be used as specified in [RFC4492] and [RFC6090] to reduce CPU cycles required to compute shared secret keys.

For better security considerations as well as to be able to build faster software implementations, newer approaches to ciphers and authentication methods will be researched and tested. Some examples are Chacha20 and Poly1305 [CHACHA-POLY] [RFC7539].

10. Security Considerations

10.1. SAAG Support

The LISP working group has and will continue to seek help from the SAAG working group for security advice. The SAAG has been involved early in the design process so they have early input and review.

10.2. LISP-Crypto Security Threats

Since ITRs and ETRs participate in key exchange over a public non-secure network, a man-in-the-middle (MITM) could circumvent the key exchange and compromise data-plane confidentiality. This can happen when the MITM is acting as a Map-Replier, provides its own public key so the ITR and the MITM generate a shared secret key among each other. If the MITM is in the data path between the ITR and ETR, it can use the shared secret key to decrypt traffic from the ITR.

Since LISP can secure Map-Replies by the authentication process specified in [LISP-SEC], the ITR can detect when a MITM has signed a Map-Reply for an EID-prefix it is not authoritative for. When an ITR determines the signature verification fails, it discards and does not reuse the key exchange parameters, avoids using the ETR for encapsulation, and issues a severe log message to the network administrator. Optionally, the ITR can send RLOC-probes to the compromised RLOC to determine if can reach the authoritative ETR.

And when the ITR validates the signature of a Map-Reply, it can begin encrypting and encapsulating packets to the RLOC of ETR.

11. IANA Considerations

This draft may require the use of the registry that selects Security parameters. Rather than convey the key exchange parameters and crypto functions directly in LISP control packets, the cipher suite values can be assigned and defined in a registry. For example, Diffie-Hellman group-id values can be used from [RFC2409] and [RFC3526].

This draft specifies how the 7-bit cipher suite values from the Security Type LCAF are partitioned. The partitions are:

```
0:      Reserved
1-96:   Allocated by registry, but first 3 values defined in this document
97-127: Private use
```

12. References

12.1. Normative References

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, DOI 10.17487/RFC2409, November 1998, <http://www.rfc-editor.org/info/rfc2409>.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, DOI 10.17487/RFC2631, June 1999, <http://www.rfc-editor.org/info/rfc2631>.

[RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, DOI 10.17487/RFC3526, May 2003, <http://www.rfc-editor.org/info/rfc3526>.

[RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", RFC 4106, DOI 10.17487/RFC4106, June 2005, <http://www.rfc-editor.org/info/rfc4106>.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, DOI 10.17487/RFC4492, May 2006, <http://www.rfc-editor.org/info/rfc4492>.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, DOI 10.17487/RFC5116, January 2008, <http://www.rfc-editor.org/info/rfc5116>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI 10.17487/RFC6090, February 2011, <http://www.rfc-editor.org/info/rfc6090>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC7539] Nir, Y. and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols", RFC 7539, DOI 10.17487/RFC7539, May 2015, <http://www.rfc-editor.org/info/rfc7539>.

12.2. Informative References

[AES-CBC] McGrew, D., Foley, J., and K. Paterson, "Authenticated Encryption with AES-CBC and HMAC-SHA", draft-mcgrew-aead-aes-cbc-hmac-sha2-05.txt (work in progress).

[CHACHA-POLY] Langley, A., "ChaCha20 and Poly1305 based Cipher Suites for TLS", draft-agl-tls-chacha20poly1305-00 (work in progress).

[CURVE25519] Bernstein, D., "Curve25519: new Diffie-Hellman speed records", Publication http://www.iacr.org/cryptodb/archive/2006/PKC/3351/3351.pdf.

[DH] "Diffie-Hellman key exchange", Wikipedia http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange.

[LCAF] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format", draft-ietf-lisp-lcaf-04.txt (work in progress).

[LISP-DDT] Fuller, V., Lewis, D., Ermaagan, V., and A. Jain, "LISP Delegated Database Tree", draft-fuller-lisp-ddt-03 (work in progress).

[LISP-SEC]
Maino, F., Ermagan, V., Cabellos, A., and D. Saucez, "LISP-Secuirty (LISP-SEC)", draft-ietf-lisp-sec-06 (work in progress).

[NIST-SP800-108]
"National Institute of Standards and Technology, "Recommendation for Key Derivation Using Pseudorandom Functions NIST SP800-108"", NIST SP 800-108, October 2009.

Appendix A. Acknowledgments

The authors would like to thank Dan Harkins, Joel Halpern, Fabio Maino, Ed Lopez, Roger Jorgensen, and Watson Ladd for their interest, suggestions, and discussions about LISP data-plane security.

The authors would like to give a special thank you to Ilari Liusvaara for his extensive commentary and discussion. He has contributed his security expertise to make lisp-crypto as secure as the state of the art in cryptography.

In addition, the support and suggestions from the SAAG working group were helpful and appreciative.

Appendix B. Document Change Log

B.1. Changes to draft-ietf-lisp-crypto-03.txt

- Posted December 2015.
- Changed cipher suite allocations. We now have 2 AES-CBC cipher suites for compatibility, 3 AES-GCM cipher suites that are faster ciphers that include AE and a Chacha20-Poly1305 cipher suite which is the fastest but not totally proven/accepted..
- Remove 1024-bit DH keys for key exchange.
- Make clear that AES and chacha20 ciphers use AEAD so part of encrytion/decryption does authentication.
- Make it more clear that separate key pairs are used in each direction between xTRs.
- Indicate that the IV length is different per cipher suite.
- Use a counter based IV for every packet for AEAD ciphers. Previously text said to use a random number. But CBC ciphers, use a random number.

- Indicate that key material is sent in network byte order (big endian).

- Remove A-bit from Security Type LCAF. No need to do authentication only with the introduction of AEAD ciphers. These ciphers can do authentication. So you get ciphertext for free.

- Remove language that refers to "encryption-key" and "integrity-key". Used term "AEAD-key" that is used by the AEAD cipher suites that do encryption and authenticaiton internal to the cipher.

B.2. Changes to draft-ietf-lisp-crypto-02.txt

- Posted September 2015.

- Add cipher suite for Elliptic Curve 25519 DH exchange.

- Add cipher suite for Chacha20/Poly1305 ciphers.

B.3. Changes to draft-ietf-lisp-crypto-01.txt

- Posted May 2015.

- Create cipher suites and encode them in the Security LCAF.

- Add IV to beginning of packet header and ICV to end of packet.

- AEAD procedures are now part of encrpytion process.

B.4. Changes to draft-ietf-lisp-crypto-00.txt

- Posted January 2015.

- Changing draft-farinacci-lisp-crypto-01 to draft-ietf-lisp-crypto-00. This draft has become a working group document

- Add text to indicate the working group may work on a new data encapsulation header format for data-plane encryption.

B.5. Changes to draft-farinacci-lisp-crypto-01.txt

- Posted July 2014.

- Add Group-ID to the encoding format of Key Material in a Security Type LCAF and modify the IANA Considerations so this draft can use key exchange parameters from the IANA registry.

- Indicate that the R-bit in the Security Type LCAF is not used by lisp-crypto.
- Add text to indicate that ETRs/RTRs can negotiate less number of keys from which the ITR/PITR sent in a Map-Request.
- Add text explaining how LISP-SEC solves the problem when a man-in-the-middle becomes part of the Map-Request/Map-Reply key exchange process.
- Add text indicating that when RLOC-probing is used for RLOC reachability purposes and rekeying is not desired, that the same key exchange parameters should be used so a reallocation of a pubic key does not happen at the ETR.
- Add text to indicate that ECDH can be used to reduce CPU requirements for computing shared secret-keys.

B.6. Changes to draft-farinacci-lisp-crypto-00.txt

- Initial draft posted February 2014.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, California 95120
USA

Phone: 408-718-2001
Email: farinacci@gmail.com

Brian Weis
cisco Systems
170 West Tasman Drive
San Jose, California 95124-1706
USA

Phone: 408-526-4796
Email: bew@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 15, 2016

D. Farinacci
lispers.net
D. Meyer
Brocade
J. Snijders
NTT Communications
March 14, 2016

LISP Canonical Address Format (LCAF)
draft-ietf-lisp-lcaf-12

Abstract

This draft defines a canonical address format encoding used in LISP control messages and in the encoding of lookup keys for the LISP Mapping Database System.

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 September 15, 2016.

Copyright Notice

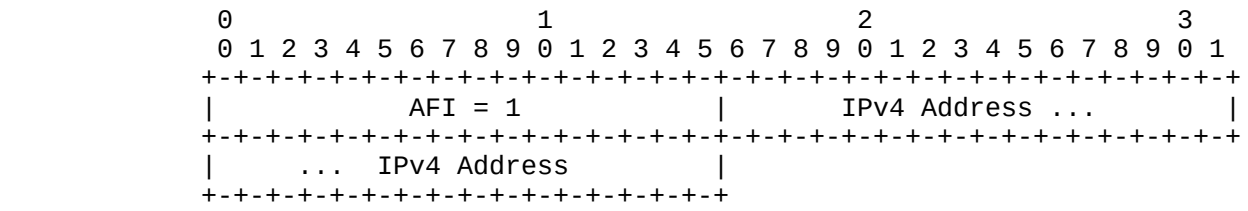
Table of Contents

1. Introduction

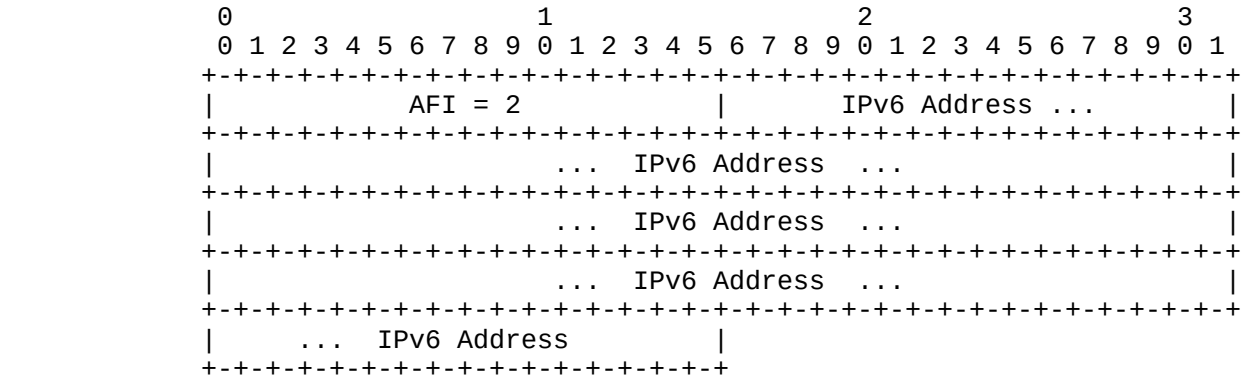
The LISP architecture and protocols [RFC6830] introduces two new numbering spaces, Endpoint Identifiers (EIDs) and Routing Locators (RLOCs) which are intended to replace most use of IP addresses on the Internet. To provide flexibility for current and future applications, these values can be encoded in LISP control messages using a general syntax that includes Address Family Identifier (AFI), length, and value fields.

Currently defined AFIs include IPv4 and IPv6 addresses, which are formatted according to code-points assigned in [AFI] as follows:

IPv4 Encoded Address:



IPv6 Encoded Address:

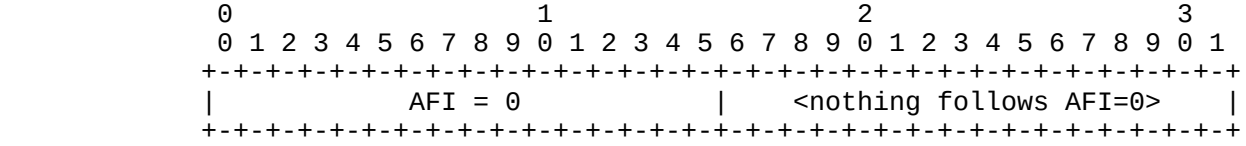


This document describes the currently-defined AFIs the LISP protocol uses along with their encodings and introduces the LISP Canonical Address Format (LCAF) that can be used to define the LISP-specific encodings for arbitrary AFI values.

2. Definition of Terms

Address Family Identifier (AFI): a term used to describe an address encoding in a packet. An address family currently defined for IPv4 or IPv6 addresses. See [AFI] and [RFC1700] for details. The reserved AFI value of 0 is used in this specification to indicate an unspecified encoded address where the the length of the address is 0 bytes following the 16-bit AFI value of 0.

Unspecified Address Format:



Endpoint ID (EID): a 32-bit (for IPv4) or 128-bit (for IPv6) value used in the source and destination address fields of the first (most inner) LISP header of a packet. The host obtains a destination EID the same way it obtains a destination address today, for example through a DNS lookup or SIP exchange. The source EID is obtained via existing mechanisms used to set a host's "local" IP address. An EID is allocated to a host from an EID-prefix block associated with the site where the host is located. An EID can be used by a host to refer to other hosts.

Routing Locator (RLOC): the IPv4 or IPv6 address of an egress tunnel router (ETR). It is the output of a EID-to-RLOC mapping lookup. An EID maps to one or more RLOCs. Typically, RLOCs are numbered from topologically aggregatable blocks that are assigned to a site at each point to which it attaches to the global Internet; where the topology is defined by the connectivity of provider networks, RLOCs can be thought of as PA addresses. Multiple RLOCs can be assigned to the same ETR device or to multiple ETR devices at a site.

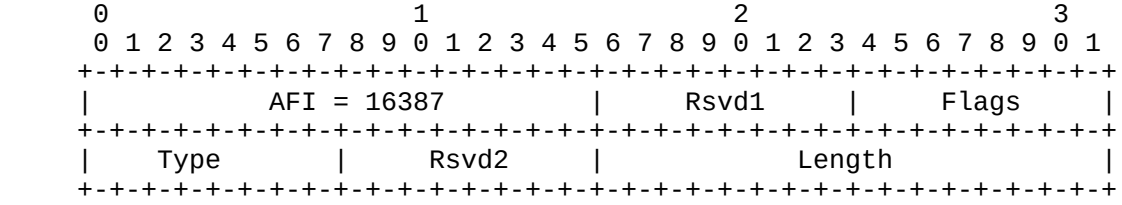
3. LISP Canonical Address Format Encodings

IANA has assigned AFI value 16387 (0x4003) to the LISP architecture and protocols. This specification defines the encoding format of the LISP Canonical Address (LCA). This section defines both experimental types as well as types that reside in the registry that have corresponding working group drafts. See IANA Considerations section for a list of types that will reside in the LISP-LCAF Registry.

The Address Family AFI definitions from [AFI] only allocate code-points for the AFI value itself. The length of the address or entity

that follows is not defined and is implied based on conventional experience. Where the LISP protocol uses LISP Canonical Addresses specifically, the address length definitions will be in this specification and take precedent over any other specification.

The first 6 bytes of an LISP Canonical Address are followed by a variable length of fields:



Rsvd1: this 8-bit field is reserved for future use and MUST be transmitted as 0 and ignored on receipt.

Flags: this 8-bit field is for future definition and use. For now, set to zero on transmission and ignored on receipt.

Type: this 8-bit field is specific to the LISP Canonical Address formatted encodings, values are:

Type 0: Null Body Type

Type 1: AFI List Type

Type 2: Instance ID Type

Type 3: AS Number Type

Type 4: Application Data Type

Type 5: Geo Coordinates Type

Type 6: Opaque Key Type

Type 7: NAT-Traversal Type

Type 8: Nonce Locator Type

Type 9: Multicast Info Type

Type 10: Explicit Locator Path Type

Type 11: Security Key Type

Type 12: Source/Dest Key Type

Type 13: Replication List Entry Type

Type 14: JSON Data Model Type

Type 15: Key/Value Address Pair Type

Type 16: Encapsulation Format Type

Rsvd2: this 8-bit field is reserved for future use and MUST be transmitted as 0 and ignored on receipt.

Length: this 16-bit field is in units of bytes and covers all of the LISP Canonical Address payload, starting and including the byte after the Length field. So any LCAF encoded address will have a minimum length of 8 bytes when the Length field is 0. The 8 bytes include the AFI, Flags, Type, Reserved, and Length fields. When the AFI is not next to encoded address in a control message, then the encoded address will have a minimum length of 6 bytes when the Length field is 0. The 6 bytes include the Flags, Type, Reserved, and Length fields.

[RFC6830] states RLOC records are sorted when encoded in control messages so the locator-set has consistent order across all xTRs for a given EID. The sort order is based on sort-key {afi, RLOC-address}. When an RLOC is LCAF encoded, the sort-key is {afi, LCAF-Type, payload}. Therefore, when a locator-set has a mix of AFI records and LCAF records, all LCAF records will appear after all the AFI records.

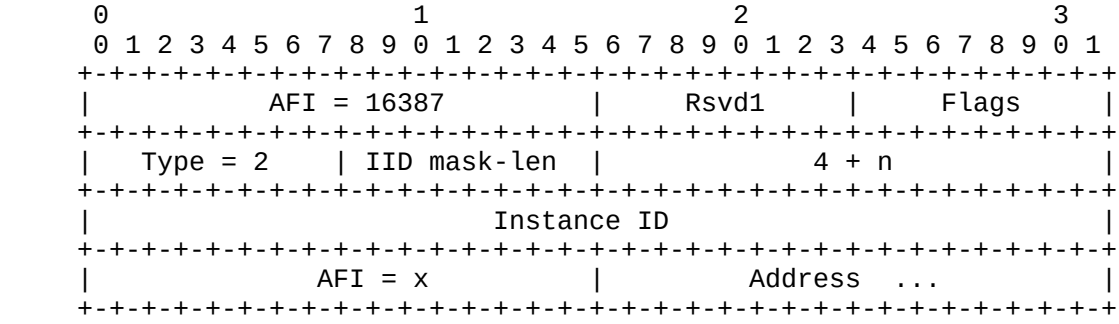
4. LISP Canonical Address Applications

4.1. Segmentation using LISP

When multiple organizations inside of a LISP site are using private addresses [RFC1918] as EID-prefixes, their address spaces must remain segregated due to possible address duplication. An Instance ID in the address encoding can aid in making the entire AFI based address unique.

Another use for the Instance ID LISP Canonical Address Format is when creating multiple segmented VPNs inside of a LISP site where keeping EID-prefix based subnets is desirable.

Instance ID LISP Canonical Address Format:



IID mask-len: if the AFI is set to 0, then this format is not encoding an extended EID-prefix but rather an instance-ID range where the 'IID mask-len' indicates the number of high-order bits used in the Instance ID field for the range.

Length value n: length in bytes of the AFI address that follows the Instance ID field including the AFI field itself.

Instance ID: the low-order 24-bits that can go into a LISP data header when the I-bit is set. See [RFC6830] for details.

AFI = x: x can be any AFI value from [AFI].

This LISP Canonical Address Type can be used to encode either EID or RLOC addresses.

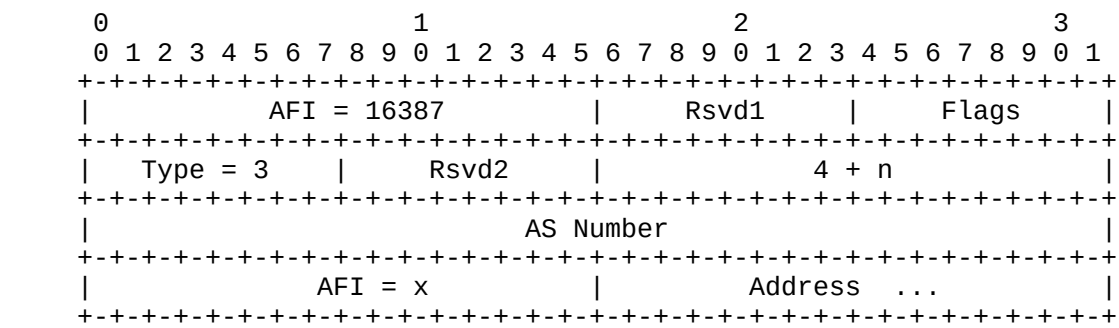
Usage: When used as a lookup key, the EID is regarded as a extended-EID in the mapping system. And this encoding is used in EID records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages.

When LISP-DDT [LISP-DDT] is used as the mapping system mechanism, extended EIDs are used in Map-Referral messages.

4.2. Carrying AS Numbers in the Mapping Database

When an AS number is stored in the LISP Mapping Database System for either policy or documentation reasons, it can be encoded in a LISP Canonical Address.

AS Number LISP Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the AS Number field including the AFI field itself.

AS Number: the 32-bit AS number of the autonomous system that has been assigned either the EID or RLOC that follows.

AFI = x: x can be any AFI value from [AFI].

The AS Number Canonical Address Type can be used to encode either EID or RLOC addresses. The former is used to describe the LISP-ALT AS number the EID-prefix for the site is being carried for. The latter is used to describe the AS that is carrying RLOC based prefixes in the underlying routing system.

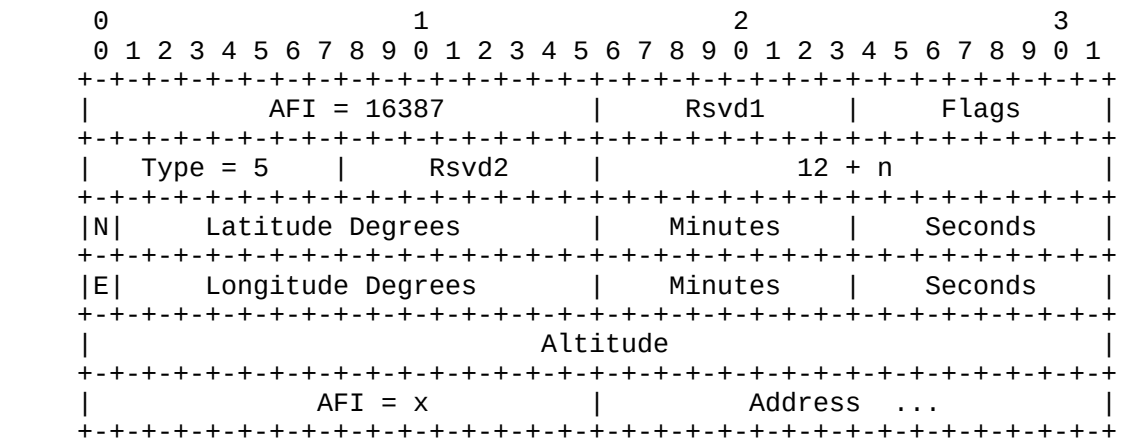
Usage: This encoding can be used in EID or RLOC records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. When LISP-DDT [LISP-DDT] is used as the mapping system mechanism, extended EIDs are used in Map-Referral messages.

4.3. Assigning Geo Coordinates to Locator Addresses

If an ETR desires to send a Map-Reply describing the Geo Coordinates for each locator in its locator-set, it can use the Geo Coordinate Type to convey physical location information.

Coordinates are specified using the WGS-84 (World Geodetic System) reference coordinate system [WGS-84].

Geo Coordinate LISP Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the 8-byte Longitude and Latitude fields including the AFI field itself.

N: When set to 1 means North, otherwise South.

Latitude Degrees: Valid values range from 0 to 90 degrees above or below the equator (northern or southern hemisphere, respectively).

Latitude Minutes: Valid values range from 0 to 59.

Latitude Seconds: Valid values range from 0 to 59.

E: When set to 1 means East, otherwise West.

Longitude Degrees: Value values are from 0 to 180 degrees right or left of the Prime Meridian.

Longitude Minutes: Valid values range from 0 to 59.

Longitude Seconds: Valid values range from 0 to 59.

Altitude: Height relative to sea level in meters. This is a signed integer meaning that the altitude could be below sea level. A value of 0x7fffffff indicates no Altitude value is encoded.

AFI = x: x can be any AFI value from [AFI].

The Geo Coordinates Canonical Address Type can be used to encode either EID or RLOC addresses. When used for EID encodings, you can

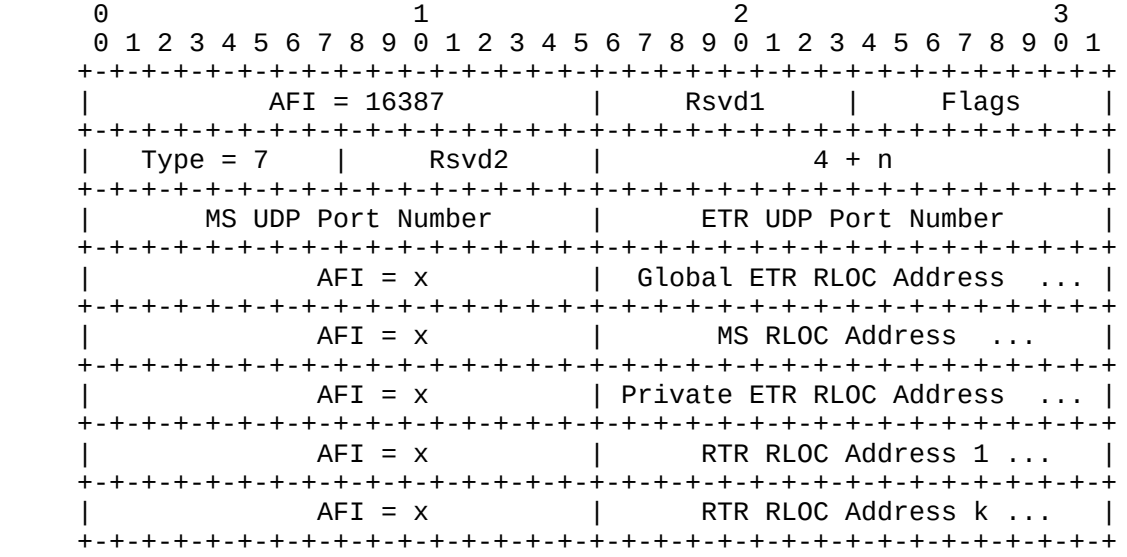
determine the physical location of an EID along with the topological location by observing the locator-set.

Usage: This encoding can be used in EID or RLOC records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. When LISP-DDT [LISP-DDT] is used as the mapping system mechanism, extended EIDs are used in Map-Referral messages.

4.4. NAT Traversal Scenarios

When a LISP system is conveying global address and mapped port information when traversing through a NAT device, the NAT-Traversal LCAF Type is used. See [LISP-NATT] for details.

NAT-Traversal Canonical Address Format:



Length value n: length in bytes of the AFI addresses that follows the UDP Port Number field including the AFI fields themselves.

MS UDP Port Number: this is the UDP port number of the Map-Server and is set to 4342.

ETR UDP Port Number: this is the port number returned to a LISP system which was copied from the source port from a packet that has flowed through a NAT device.

AFI = x: x can be any AFI value from [AFI].

Global ETR RLOC Address: this is an address known to be globally unique built by NAT-traversal functionality in a LISP router.

MS RLOC Address: this is the address of the Map-Server used in the destination RLOC of a packet that has flowed through a NAT device.

Private ETR RLOC Address: this is an address known to be a private address inserted in this LCAF format by a LISP router that resides on the private side of a NAT device.

RTR RLOC Address: this is an encapsulation address used by an ITR or PITR which resides behind a NAT device. This address is known to have state in a NAT device so packets can flow from it to the LISP ETR behind the NAT. There can be one or more NTR addresses supplied in these set of fields. The number of NTRs encoded is determined by the LCAF length field. When there are no NTRs supplied, the NTR fields can be omitted and reflected by the LCAF length field or an AFI of 0 can be used to indicate zero NTRs encoded.

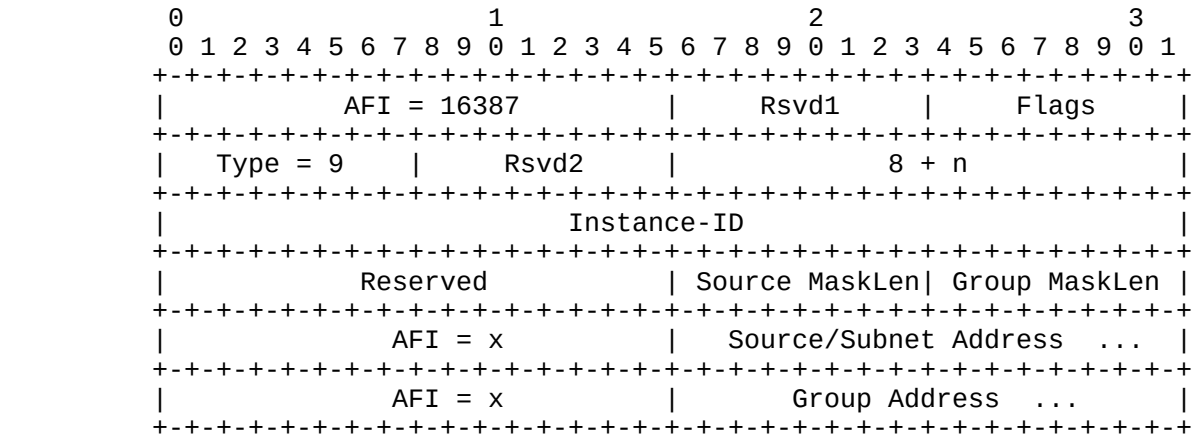
Usage: This encoding can be used in Info-Request and Info-Reply messages. The mapping system does not store this information. The information is used by an xTR and Map-Server to convey private and public address information when traversing NAT and firewall devices.

4.5. Multicast Group Membership Information

Multicast group information can be published in the mapping database so a lookup on an EID based group address can return a replication list of group addresses or a unicast addresses for single replication or multiple head-end replications. The intent of this type of unicast replication is to deliver packets to multiple ETRs at receiver LISP multicast sites. The locator-set encoding for this EID record type can be a list of ETRs when they each register with "Merge Semantics". The encoding can be a typical AFI encoded locator address. When an RTR list is being registered (with multiple levels according to [LISP-RE]), the Replication List Entry LCAF type is used for locator encoding.

This LCAF encoding can be used to send broadcast packets to all members of a subnet when each EIDs are away from their home subnet location.

Multicast Info Canonical Address Format:



Length value n: length in bytes of fields that follow.

Reserved: must be set to zero and ignore on receipt.

Instance ID: the low-order 24-bits that can go into a LISP data header when the I-bit is set. See [RFC6830] for details. The use of the Instance-ID in this LCAF type is to associate a multicast forwarding entry for a given VPN. The instance-ID describes the VPN and is registered to the mapping database system as a 3-tuple of (Instance-ID, S-prefix, G-prefix).

Source MaskLen: the mask length of the source prefix that follows.

Group MaskLen: the mask length of the group prefix that follows.

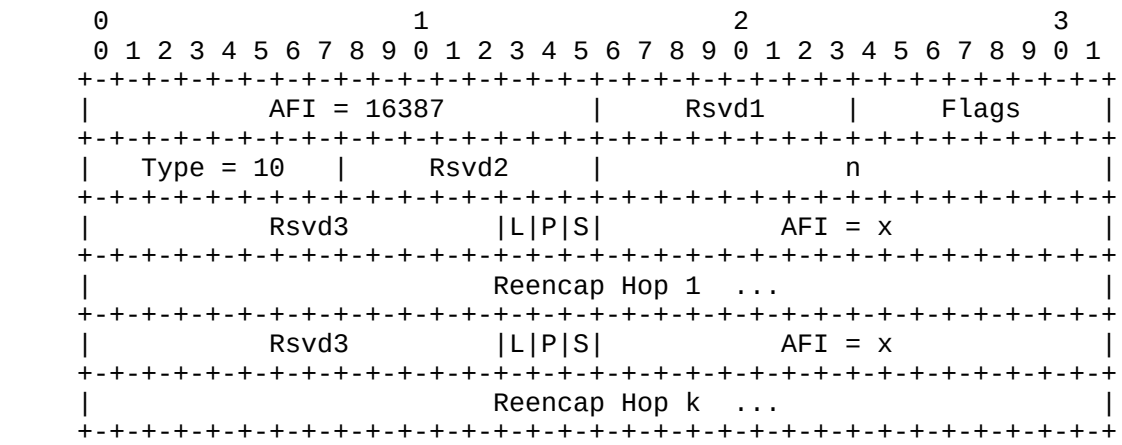
AFI = x: x can be any AFI value from [AFI]. When a specific AFI has its own encoding of a multicast address, this field must be either a group address or a broadcast address.

Usage: This encoding can be used in EID records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. When LISP-DDT [LISP-DDT] is used as the mapping system mechanism, extended EIDs are used in Map-Referral messages.

4.6. Traffic Engineering using Re-encapsulating Tunnels

For a given EID lookup into the mapping database, this LCAF format can be returned to provide a list of locators in an explicit re-encapsulation path. See [LISP-TE] for details.

Explicit Locator Path (ELP) Canonical Address Format:



Length value n: length in bytes of fields that follow.

Lookup bit (L): this is the Lookup bit used to indicate to the user of the ELP to not use this address for encapsulation but to look it up in the mapping database system to obtain an encapsulating RLOC address.

RLOC-Probe bit (P): this is the RLOC-probe bit which means the Reencap Hop allows RLOC-probe messages to be sent to it. When the R-bit is set to 0, RLOC-probes must not be sent. When a Reencap Hop is an anycast address then multiple physical Reencap Hops are using the same RLOC address. In this case, RLOC-probes are not needed because when the closest RLOC address is not reachable another RLOC address can reachable.

Strict bit (S): this the strict bit which means the associated Rencap Hop is required to be used. If this bit is 0, the reencapsulator can skip this Reencap Hop and go to the next one in the list.

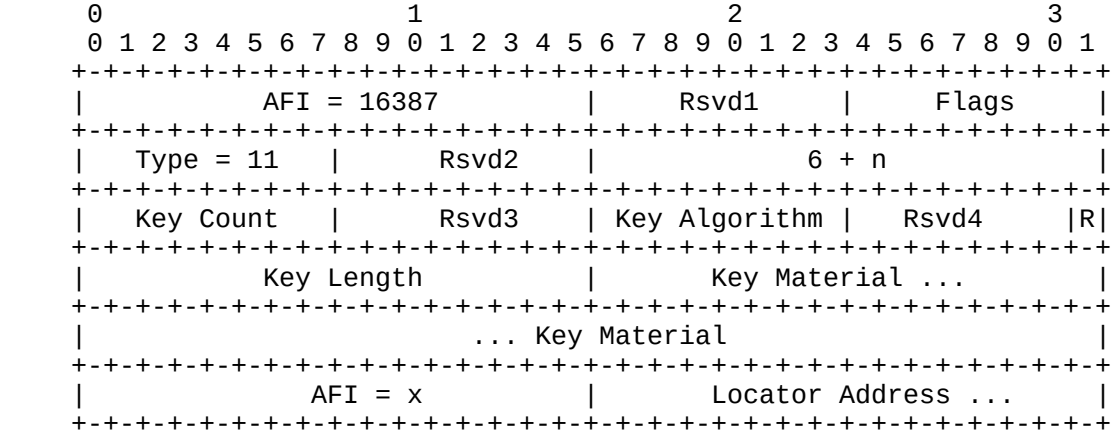
AFI = x: x can be any AFI value from [AFI]. When a specific AFI has its own encoding of a multicast address, this field must be either a group address or a broadcast address.

Usage: This encoding can be used in RLOC records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. This encoding not need to be understood by the mapping system for mapping database lookups since this LCAF type is not a lookup key.

4.7. Storing Security Data in the Mapping Database

When a locator in a locator-set has a security key associated with it, this LCAF format will be used to encode key material. See [LISP-DDT] for details.

Security Key Canonical Address Format:



Length value n: length in bytes of fields that start with the Key Material field.

Key Count: the Key Count field declares the number of Key sections included in this LCAF.

Key Algorithm: the Algorithm field identifies the key's cryptographic algorithm and specifies the format of the Public Key field.

R bit: this is the revoke bit and, if set, it specifies that this Key is being Revoked.

Key Length: this field determines the length in bytes of the Key Material field.

Key Material: the Key Material field stores the key material. The format of the key material stored depends on the Key Algorithm field.

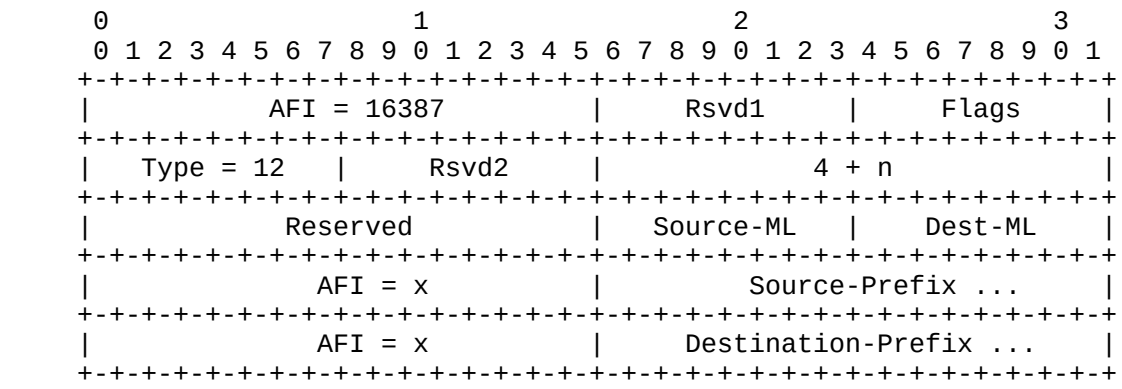
AFI = x: x can be any AFI value from [AFI].This is the locator address that owns the encoded security key.

Usage: This encoding can be used in EID or RLOC records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. When LISP-DDT [LISP-DDT] is used as the mapping system mechanism, extended EIDs are used in Map-Referral messages.

4.8. Source/Destination 2-Tuple Lookups

When both a source and destination address of a flow needs consideration for different locator-sets, this 2-tuple key is used in EID fields in LISP control messages. When the Source/Dest key is registered to the mapping database, it can be encoded as a source-prefix and destination-prefix. When the Source/Dest is used as a key for a mapping database lookup the source and destination come from a data packet.

Source/Dest Key Canonical Address Format:



Length value n: length in bytes of fields that follow.

Reserved: must be set to zero and ignore on receipt.

Source-ML: the mask length of the source prefix that follows.

Dest-ML: the mask length of the destination prefix that follows.

AFI = x: x can be any AFI value from [AFI]. When a specific AFI has its own encoding of a multicast address, this field must be either a group address or a broadcast address.

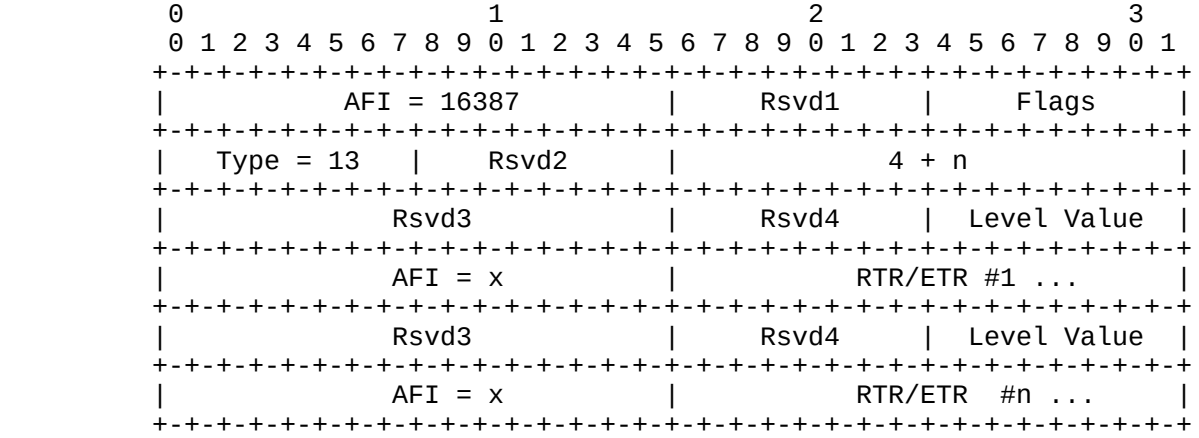
Refer to [LISP-TE] for usage details.

Usage: This encoding can be used in EID records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. When LISP-DDT [LISP-DDT] is used as the mapping system mechanism, extended EIDs are used in Map-Referral messages.

4.9. Replication List Entries for Multicast Forwarding

The Replication List Entry LCAF type is an encoding for a locator being used for unicast replication according to the specification in [LISP-RE]. This locator encoding is pointed to by a Multicast Info LCAF Type and is registered by Re-encapsulating Tunnel Routers (RTRs) that are participating in an overlay distribution tree. Each RTR will register its locator address and its configured level in the distribution tree.

Replication List Entry Address Format:



Length value n: length in bytes of fields that follow.

Rsvd{1,2,3,4}: must be set to zero and ignore on receipt.

Level Value: this value is associated with the level within the overlay distribution tree hierarchy where the RTR resides. The level numbers are ordered from lowest value being close to the ITR (meaning that ITRs replicate to level-0 RTRs) and higher levels are further downstream on the distribution tree closer to ETRs of multicast receiver sites.

AFI = x: x can be any AFI value from [AFI]. A specific AFI has its own encoding of either a unicast or multicast locator address. All RTR/ETR entries for the same level should be combined together by a Map-Server to avoid searching through the entire multi-level list of locator entries in a Map-Reply message.

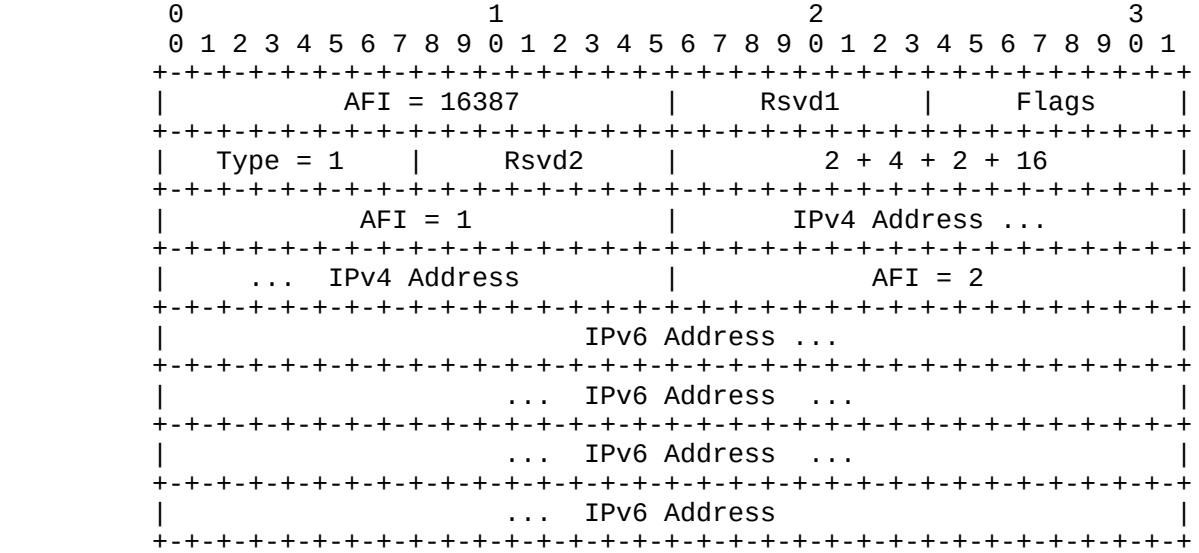
Usage: This encoding can be used in RLOC records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages.

4.10. Applications for AFI List Type

4.10.1. Binding IPv4 and IPv6 Addresses

When header translation between IPv4 and IPv6 is desirable a LISP Canonical Address can use the AFI List Type to carry multiple AFIs in one LCAF AFI.

Address Binding LISP Canonical Address Format:



Length: length in bytes is fixed at 24 when IPv4 and IPv6 AFI encoded addresses are used.

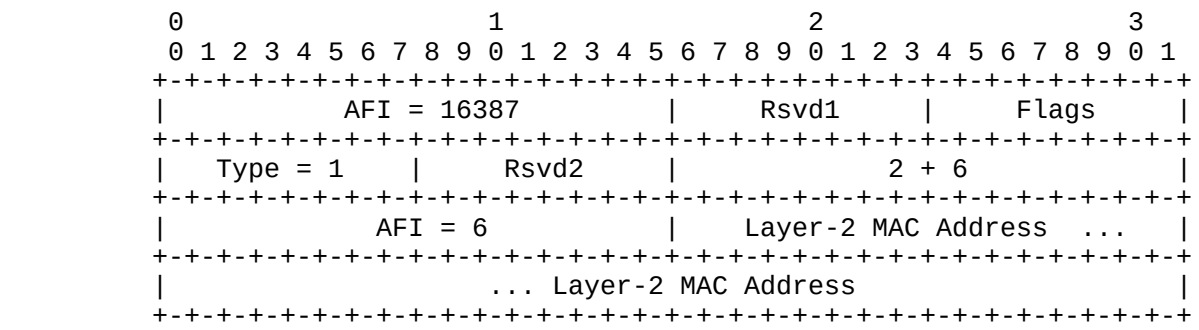
This type of address format can be included in a Map-Request when the address is being used as an EID, but the Mapping Database System lookup destination can use only the IPv4 address. This is so a Mapping Database Service Transport System, such as LISP-ALT [RFC6836], can use the Map-Request destination address to route the control message to the desired LISP site.

Usage: This encoding can be used in EID or RLOC records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. See subsections in this section for specific use cases.

4.10.2. Layer-2 VPNs

When MAC addresses are stored in the LISP Mapping Database System, the AFI List Type can be used to carry AFI 6.

MAC Address LISP Canonical Address Format:



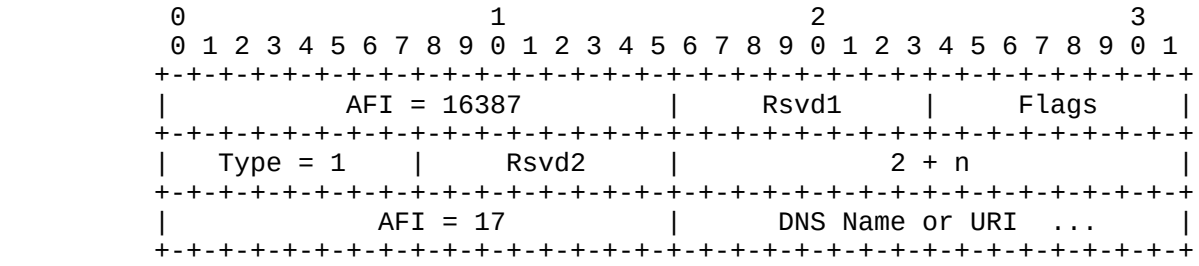
Length: length in bytes is fixed at 8 when MAC address AFI encoded addresses are used.

This address format can be used to connect layer-2 domains together using LISP over an IPv4 or IPv6 core network to create a layer-2 VPN. In this use-case, a MAC address is being used as an EID, and the locator-set that this EID maps to can be an IPv4 or IPv6 RLOCs, or even another MAC address being used as an RLOC.

4.10.3. ASCII Names in the Mapping Database

If DNS names or URIs are stored in the LISP Mapping Database System, the AFI List Type can be used to carry an ASCII string where it is delimited by length 'n' of the LCAF Length encoding.

ASCII LISP Canonical Address Format:

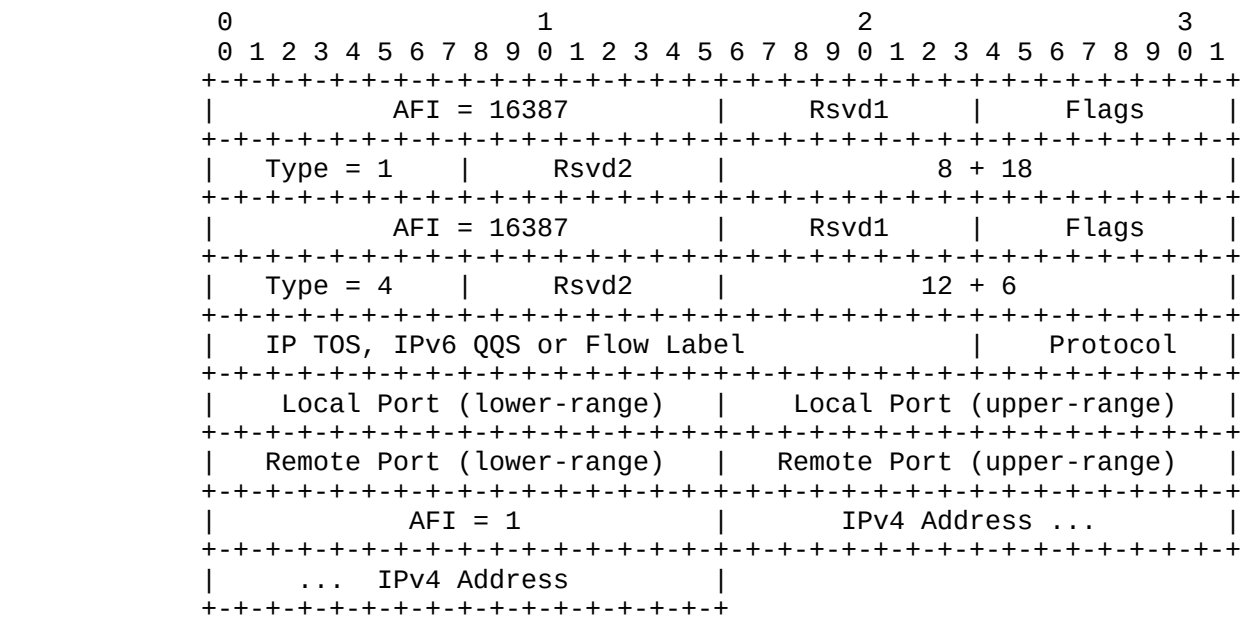


Length value n: length in bytes AFI=17 field and the null-terminated ASCII string (the last byte of 0 is included).

4.10.4. Using Recursive LISP Canonical Address Encodings

When any combination of above is desirable, the AFI List Type value can be used to carry within the LCAF AFI another LCAF AFI.

Recursive LISP Canonical Address Format:



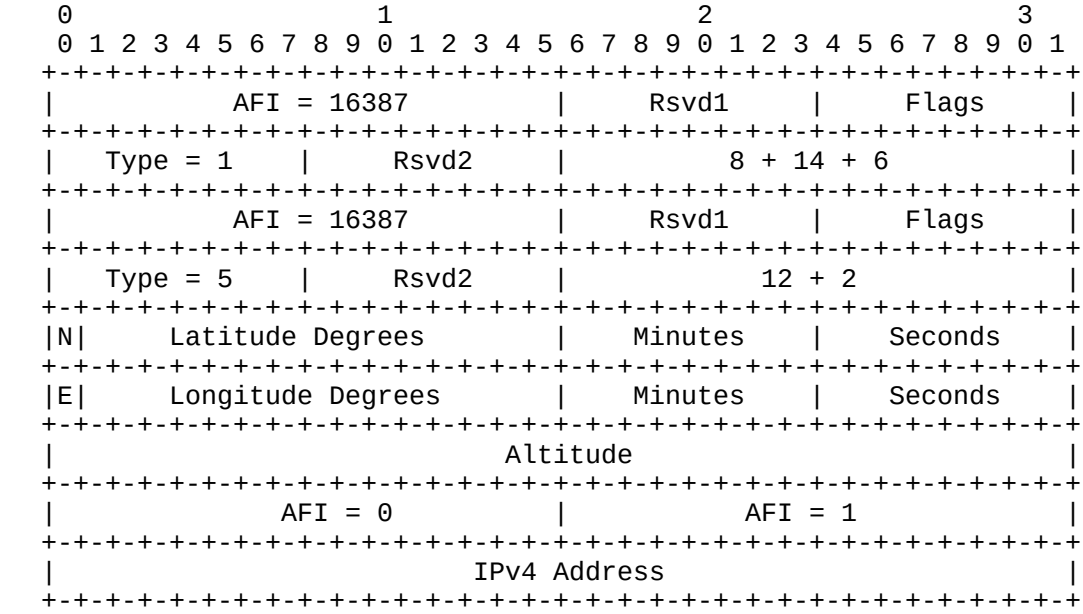
Length: length in bytes is fixed at 18 when an AFI=1 IPv4 address is included.

This format could be used by a Mapping Database Transport System, such as LISP-ALT [RFC6836], where the AFI=1 IPv4 address is used as an EID and placed in the Map-Request destination address by the sending LISP system. The ALT system can deliver the Map-Request to the LISP destination site independent of the Application Data Type AFI payload values. When this AFI is processed by the destination LISP site, it can return different locator-sets based on the type of application or level of service that is being requested.

4.10.5. Compatibility Mode Use Case

A LISP system should use the AFI List Type format when sending to LISP systems that do not support a particular LCAF Type used to encode locators. This allows the receiving system to be able to parse a locator address for encapsulation purposes. The list of AFIs in an AFI List LCAF Type has no semantic ordering and a receiver should parse each AFI element no matter what the ordering.

Compatibility Mode Address Format:



If a system does not recognized the Geo Coordinate LCAF Type that is accompanying a locator address, an encoder can include the Geo Coordinate LCAF Type embedded in a AFI List LCAF Type where the AFI in the Geo Coordinate LCAF is set to 0 and the AFI encoded next in the list is encoded with a valid AFI value to identify the locator address.

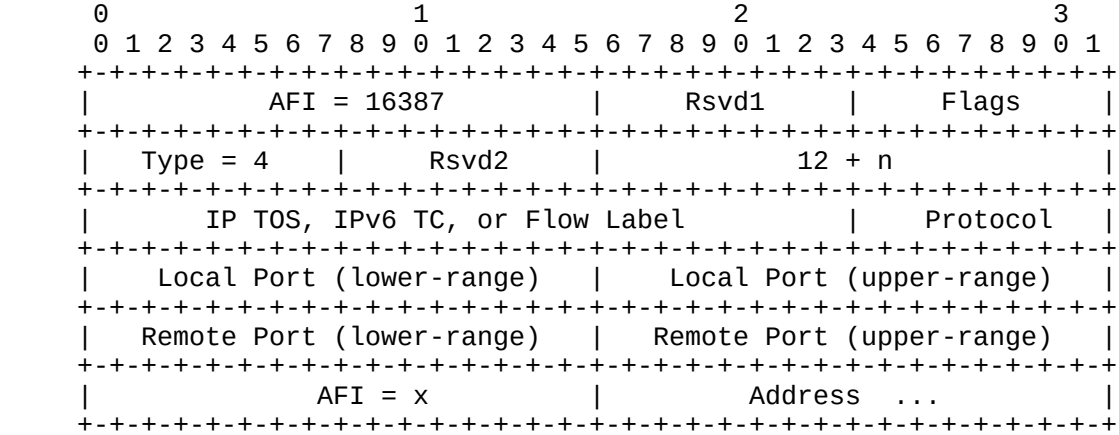
A LISP system is required to support the AFI List LCAF Type to use this procedure. It would skip over 10 bytes of the Geo Coordinate LCAF Type to get to the locator address encoding (an IPv4 locator address). A LISP system that does support the Geo Coordinate LCAF Type can support parsing the locator address within the Geo Coordinate LCAF encoding or in the locator encoding that follows in the AFI List LCAF.

5. Experimental LISP Canonical Address Applications

5.1. Convey Application Specific Data

When a locator-set needs to be conveyed based on the type of application or the Per-Hop Behavior (PHB) of a packet, the Application Data Type can be used.

Application Data LISP Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the 8-byte Application Data fields including the AFI field itself.

IP TOS, IPv6 TC, or Flow Label: this field stores the 8-bit IPv4 TOS field used in an IPv4 header, the 8-bit IPv6 Traffic Class or Flow Label used in an IPv6 header.

Local Port/Remote Port Ranges: these fields are from the TCP, UDP, or SCTP transport header. A range can be specified by using a lower value and an upper value. When a single port is encoded, the lower and upper value fields are the same.

AFI = x: x can be any AFI value from [AFI].

The Application Data Canonical Address Type is used for an EID encoding when an ITR wants a locator-set for a specific application. When used for an RLOC encoding, the ETR is supplying a locator-set for each specific application is has been configured to advertise.

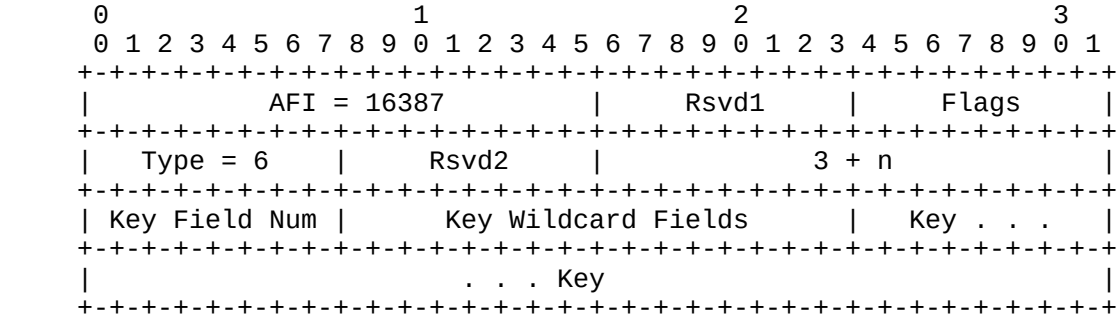
Usage: This encoding can be used in EID records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages. When LISP-DDT [LISP-DDT] is used as the mapping system mechanism, extended EIDs are

used in Map-Referral messages. This LCAF type is used as a lookup key to the mapping system that can return a longest-match or exact-match entry.

5.2. Generic Database Mapping Lookups

When the LISP Mapping Database system holds information accessed by a generic formatted key (where the key is not the usual IPv4 or IPv6 address), an opaque key may be desirable.

Opaque Key LISP Canonical Address Format:



Length value n: length in bytes of the type's payload. The value n is the number of bytes that follow this Length field.

Key Field Num: the number of fields (minus 1) the key can be broken up into. The width of the fields are fixed length. So for a key size of 8 bytes, with a Key Field Num of 4 allows 4 fields of 2 bytes in length. Valid values for this field range from 0 to 15 supporting a maximum of 16 field separations.

Key Wildcard Fields: describes which fields in the key are not used as part of the key lookup. This wildcard encoding is a bitfield. Each bit is a don't-care bit for a corresponding field in the key. Bit 0 (the low-order bit) in this bitfield corresponds the first field, right-justified in the key, bit 1 the second field, and so on. When a bit is set in the bitfield it is a don't-care bit and should not be considered as part of the database lookup. When the entire 16-bits is set to 0, then all bits of the key are used for the database lookup.

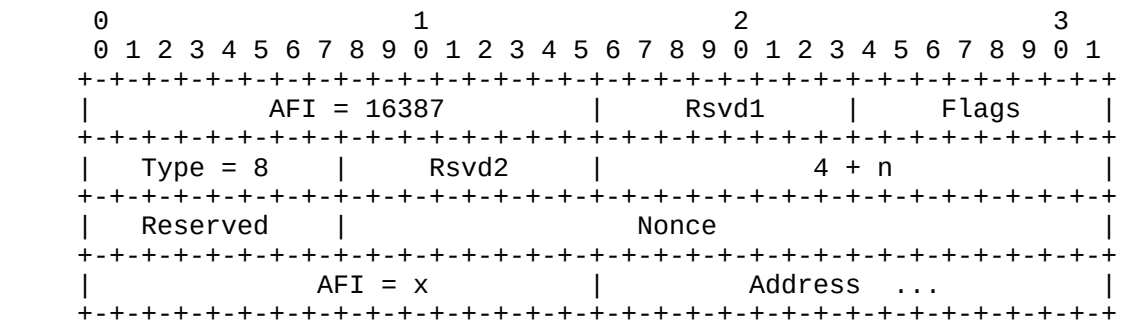
Key: the variable length key used to do a LISP Database Mapping lookup. The length of the key is the value n (shown above) minus 3.

Usage: This is an experimental type where the usage has not been defined yet.

5.3. PETR Admission Control Functionality

When a public PETR device wants to verify who is encapsulating to it, it can check for a specific nonce value in the LISP encapsulated packet. To convey the nonce to admitted ITRs or PITRs, this LCAF format is used in a Map-Register or Map-Reply locator-record.

Nonce Locator Canonical Address Format:



Length value n: length in bytes of the AFI address that follows the Nonce field including the AFI field itself.

Reserved: must be set to zero and ignore on receipt.

Nonce: this is a nonce value returned by an ETR in a Map-Reply locator-record to be used by an ITR or PITR when encapsulating to the locator address encoded in the AFI field of this LCAF type.

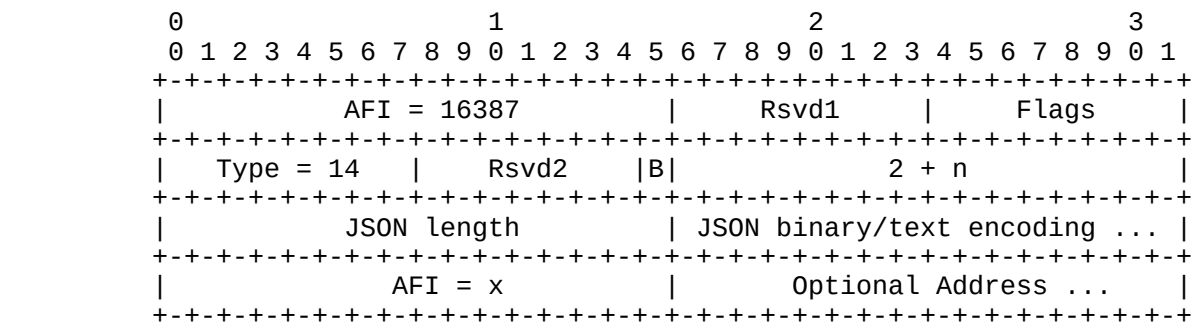
AFI = x: x can be any AFI value from [AFI].

Usage: This is an experimental type where the usage has not been defined yet.

5.4. Data Model Encoding

This type allows a JSON data model to be encoded either as an EID or RLOC.

JSON Data Model Type Address Format:



Length value n: length in bytes of fields that follow.

Rsvd{1,2}: must be set to zero and ignore on receipt.

B bit: indicates that the JSON field is binary encoded according to [JSON-BINARY] when the bit is set to 1. Otherwise the encoding is based on text encoding according to [RFC4627].

JSON length: length in octets of the following 'JSON binary/text encoding' field.

JSON binary/text encoding field: a variable length field that contains either binary or text encodings.

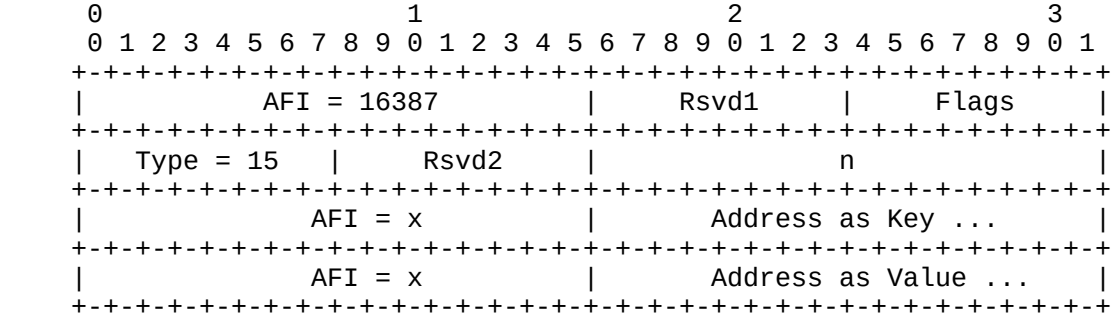
AFI = x: x can be any AFI value from [AFI]. A specific AFI has its own encoding of either a unicast or multicast locator address. All RTR/ETR entries for the same level should be combined together by a Map-Server to avoid searching through the entire multi-level list of locator entries in a Map-Reply message.

Usage: This is an experimental type where the usage has not been defined yet.

5.5. Encoding Key/Value Address Pairs

The Key/Value pair is for example useful for attaching attributes to other elements of LISP packets, such as EIDs or RLOCs. When attaching attributes to EIDs or RLOCs, it's necessary to distinguish between the element that should be used as EID or RLOC, and hence as key for lookups, and additional attributes. This is especially the case when the difference cannot be determined from the types of the elements, such as when two IP addresses are being used.

Key/Value Pair Address Format:



Length value n: length in bytes of fields that follow.

Rsvd{1,2}: must be set to zero and ignore on receipt.

AFI = x: x can be any AFI value from [AFI]. A specific AFI has its own encoding of either a unicast or multicast locator address. All RTR/ETR entries for the same level should be combined together by a Map-Server to avoid searching through the entire multi-level list of locator entries in a Map-Reply message.

Address as Key: this AFI encoded address will be attached with the attributes encoded in "Address as Value" which follows this field.

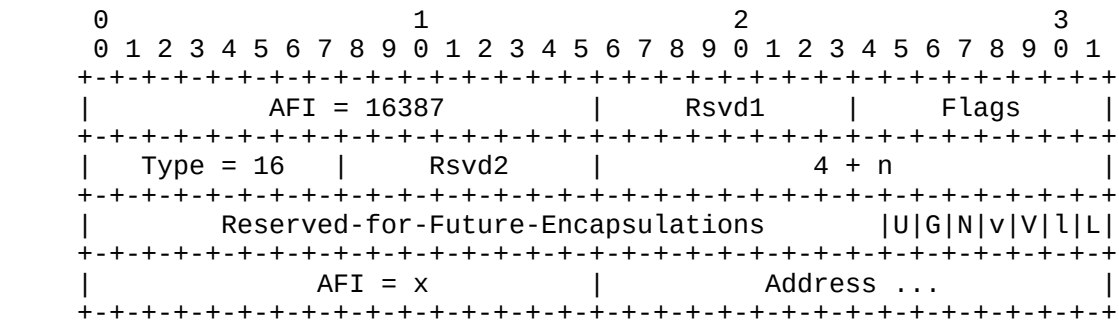
Address as Value: this AFI encoded address will be the attribute address that goes along with "Address as Key" which precedes this field.

Usage: This is an experimental type where the usage has not been defined yet.

5.6. Multiple Data-Planes

Overlays are becoming popular in many parts of the network which have created an explosion of data-plane encapsulation headers. Since the LISP mapping system can hold many types of address formats, it can represent the encapsulation format supported by an RLOC as well. When an encapsulator receives a Map-Reply with an Encapsulation Format LCAF Type encoded in an RLOC-record, it can select an encapsulation format, that it can support, from any of the encapsulation protocols which have the bit set to 1 in this LCAF type.

Encapsulation Format Address Format:



Rsvd1/Rsvd2: must be set to zero and ignored on receipt.

Length value n: length in bytes of the AFI address that follows the next 32-bits including the AFI field itself.

Reserved-for-Future-Encapsulations: must be set to zero and ignored on receipt. This field will get bits allocated to future encapsulations, as they are created.

L: The RLOCs listed in the AFI encoded addresses in the next longword can accept layer3 LISP encapsulation using destination UDP port 4341 [RFC6830].

l: The RLOCs listed in the AFI encoded addresses in the next longword can accept layer2 LISP encapsulation using destination UDP port 8472 [L2-LISP].

V: The RLOCs listed in the AFI encoded addresses in the next longword can accept VXLAN encapsulation using destination UDP port 4789 [RFC7348].

v: The RLOCs listed in the AFI encoded addresses in the next longword can accept VXLAN-GPE encapsulation using destination UDP port 4790 [GPE].

N: The RLOCs listed in the AFI encoded addresses in the next longword can accept NV-GRE encapsulation using IPv4/ IPv6 protocol number 47 [NVGRE].

G: The RLOCs listed in the AFI encoded addresses in the next longword can accept GENEVE encapsulation using destination UDP port 6081 [GENEVE].

U: The RLOCs listed in the AFI encoded addresses in the next longword can accept GUE encapsulation using destination UDP port TBD [GUE].

Usage: This encoding can be used in RLOC records in Map-Requests, Map-Replies, Map-Registers, and Map-Notify messages.

6. Security Considerations

There are no security considerations for this specification. The security considerations are documented for the protocols that use LISP Canonical Addressing. Refer to the those relevant specifications.

The use of the Geo-Coordinates LCAF Type may raise physical privacy issues. It can be up to the mapping system, based on policy parameters, when this LCAF type is returned to a Map-Requester.

7. IANA Considerations

This document defines a canonical address format encoding used in LISP control messages and in the encoding of lookup keys for the LISP Mapping Database System. Such address format is based on a fixed AFI (16387) and a LISP LCAF Type field.

The LISP LCAF Type field is an 8-bit field specific to the LISP Canonical Address formatted encodings, for which IANA is to create and maintain a new registry (as outlined in [RFC5226]) entitled "LISP LCAF Type". Initial values for the LISP LCAF Type registry are given below. Future assignments are to be made through expert review with a specification required publication. Assignments consist of a LISP LCAF Type name and its associated value:

Value	LISP LCAF Type Name	Definition
0	Null Body Type	Section 3
1	AFI List Type	Section 3
2	Instance ID Type	Section 3
3	AS Number Type	Section 3
5	Geo Coordinates Type	Section 3
7	NAT-Traversal Type	Section 3
9	Multicast Info Type	Section 3
10	Explicit Locator Path Type	Section 3
11	Security Key Type	Section 3
12	Source/Dest Key Type	Section 3
13	Replication List Entry Type	Section 3

Table 1: LISP LCAF Type Initial Values

8. References

8.1. Normative References

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, DOI 10.17487/RFC1700, October 1994, <http://www.rfc-editor.org/info/rfc1700>.

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <http://www.rfc-editor.org/info/rfc1918>.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, DOI 10.17487/RFC4627, July 2006, <http://www.rfc-editor.org/info/rfc4627>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, DOI 10.17487/RFC6836, January 2013, <http://www.rfc-editor.org/info/rfc6836>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <http://www.rfc-editor.org/info/rfc7348>.

8.2. Informative References

[AFI] IANA, , "Address Family Identifier (AFIs)", ADDRESS FAMILY NUMBERS http://www.iana.org/numbers.html, Febuary 2007.

[GENEVE] Gross, J., Sridhar, T., Garg, P., Wright, C., Ganga, I., Agarwal, P., Duda, K., Dutt, D., and J. Hudson, "Geneve: Generic Network Virtualization Encapsulation", draft-gross-geneve-02 (work in progress).

[GPE] Quinn, P., Agarwal, P., Fernando, R., Kreeger, L., Kreeger, L., Lewis, D., Maino, F., Smith, M., Yadav, N., Yong, L., Xu, X., Elzur, U., and P. Garg, "Generic Protocol Extension for VXLAN", draft-quinn-vxlan-gpe-03.txt (work in progress).

[GUE] Herbert, T. and L. Yong, "Generic UDP Encapsulation", draft-herbert-gue-02.txt (work in progress).

[JSON-BINARY] "Universal Binary JSON Specification", URL http://ubjson.org.

[L2-LISP] Smith, M., Dutt, D., Farinacci, D., and F. Maino, "Layer 2 (L2) LISP Encapsulation Format", draft-smith-lisp-layer2-03.txt (work in progress).

[LISP-DDT] Fuller, V., Lewis, D., and V. Ermagan, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-01.txt (work in progress).

[LISP-NATT] Ermagan, V., Farinacci, D., Lewis, D., Skriver, J., Maino, F., and C. White, "NAT traversal for LISP", draft-ermagan-lisp-nat-traversal-10.txt (work in progress).

[LISP-RE] Coras, F., Cabellos-Aparicio, A., Domingo-Pascual, J., Maino, F., and D. Farinacci, "LISP Replication Engineering", draft-coras-lisp-re-08.txt (work in progress).

[LISP-TE] Farinacci, D., Lahiri, P., and M. Kowal, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-10.txt (work in progress).

[NVGRE] Sridharan, M., Greenberg, A., Wang, Y., Garg, P., Venkataramiah, N., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-06.txt (work in progress).

[WGS-84] Geodesy and Geophysics Department, DoD., "World Geodetic System 1984", NIMA TR8350.2, January 2000, <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>.

Appendix A. Acknowledgments

The authors would like to thank Vince Fuller, Gregg Schudel, Jesper Skriver, Luigi Iannone, Isidor Kouvelas, and Sander Steffann for their technical and editorial commentary.

The authors would like to thank Victor Moreno for discussions that lead to the definition of the Multicast Info LCAF type.

The authors would like to thank Parantap Lahiri and Michael Kowal for discussions that lead to the definition of the Explicit Locator Path (ELP) LCAF type.

The authors would like to thank Fabio Maino and Vina Ermagan for discussions that lead to the definition of the Security Key LCAF type.

The authors would like to thank Albert Cabellos-Aparicio and Florin Coras for discussions that lead to the definition of the Replication List Entry LCAF type.

Thanks goes to Michiel Blokzijl and Alberto Rodriguez-Natal for suggesting new LCAF types.

Thanks also goes to Terry Manderson for assistance obtaining a LISP AFI value from IANA.

Appendix B. Document Change Log

B.1. Changes to draft-ietf-lisp-lcaf-12.txt

- Submitted March 2016.
- Updated references and document timer.
- Removed the R, J, and L bits from the Multicast Info Type LCAF since working group decided to not go forward with draft-farinacci-lisp-mr-signaling-03.txt in favor of draft- ietf-lisp-signal-free-00.txt.

B.2. Changes to draft-ietf-lisp-lcaf-11.txt

- Submitted September 2015.
- Reflecting comments from Prague LISP working group.
- Readying document for a LISP LCAF registry, RFC publication, and for new use-cases that will be defined in the new charter.

B.3. Changes to draft-ietf-lisp-lcaf-10.txt

- Submitted June 2015.
- Fix coauthor Job’s contact information.

B.4. Changes to draft-ietf-lisp-lcaf-09.txt

- Submitted June 2015.
- Fix IANA Considerations section to request a registry to allocate and track LCAF Type values.

B.5. Changes to draft-ietf-lisp-lcaf-08.txt

- Submitted April 2015.
- Comment from Florin. The Application Data Type length field has a typo. The field should be labeled "12 + n" and not "8 + n".
- Fix length fields in the sections titled "Using Recursive LISP Canonical Address Encodings", "Generic Database Mapping Lookups", and "Data Model Encoding".

B.6. Changes to draft-ietf-lisp-lcaf-07.txt

- Submitted December 2014.
- Add a new LCAF Type called "Encapsulation Format" so decapsulating xTRs can inform encapsulating xTRs what data-plane encapsulations they support.

B.7. Changes to draft-ietf-lisp-lcaf-06.txt

- Submitted October 2014.
- Make it clear how sorted RLOC records are done when LCAFs are used as the RLOC record.

B.8. Changes to draft-ietf-lisp-lcaf-05.txt

- Submitted May 2014.
- Add a length field of the JSON payload that can be used for either binary or text encoding of JSON data.

B.9. Changes to draft-ietf-lisp-lcaf-04.txt

- Submitted January 2014.
- Agreement among ELP implementors to have the AFI 16-bit field adjacent to the address. This will make the encoding consistent with all other LCAF type address encodings.

B.10. Changes to draft-ietf-lisp-lcaf-03.txt

- Submitted September 2013.
- Updated references and author’s affilations.

- Added Instance-ID to the Multicast Info Type so there is relative ease in parsing (S,G) entries within a VPN.
- Add port range encodings to the Application Data LCAF Type.
- Add a new JSON LCAF Type.
- Add Address Key/Value LCAF Type to allow attributes to be attached to an address.

B.11. Changes to draft-ietf-lisp-lcaf-02.txt

- Submitted March 2013.
- Added new LCAF Type "Replication List Entry" to support LISP replication engineering use-cases.
- Changed references to new LISP RFCs.

B.12. Changes to draft-ietf-lisp-lcaf-01.txt

- Submitted January 2013.
- Change longitude range from 0-90 to 0-180 in section 4.4.
- Added reference to WGS-84 in section 4.4.

B.13. Changes to draft-ietf-lisp-lcaf-00.txt

- Posted first working group draft August 2012.
- This draft was renamed from draft-farinacci-lisp-lcaf-10.txt.

Authors' Addresses

Dino Farinacci
lispers.net
San Jose, CA
USA

Email: farinacci@gmail.com

Dave Meyer
Brocade
San Jose, CA
USA

Email: dmm@1-4-5.net

Job Snijders
NTT Communications
Theodorus Majofskistraat 100
Amsterdam 1065 SZ
NL

Email: job@ntt.net

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: June 23, 2016

V. Moreno
Cisco Systems
D. Farinacci
lispers.net
December 21, 2015

Signal-Free LISP Multicast
draft-ietf-lisp-signal-free-multicast-00

Abstract

When multicast sources and receivers are active at LISP sites, the core network is required to use native multicast so packets can be delivered from sources to group members. When multicast is not available to connect the multicast sites together, a signal-free mechanism can be used to allow traffic to flow between sites. The mechanism within here uses unicast replication and encapsulation over the core network for the data-plane and uses the LISP mapping database system so encapsulators at the source LISP multicast site can find de-capsulators at the receiver LISP multicast sites.

Requirements Language

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

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 June 23, 2016.

Table of Contents

1. Introduction

When multicast sources and receivers are active at LISP sites, and the core network between the sites does not provide multicast support, a signal-free mechanism can be used to create an overlay that will allow multicast traffic to flow between sites and connect the multicast trees at the different sites.

The signal-free mechanism here proposed does not extend PIM over the overlay as proposed in [RFC6831], nor does the mechanism utilize direct signaling between the Receiver-ETRs and Sender-ITRs as described in [I-D.farinacci-lisp-mr-signaling]. The signal-free mechanism proposed reduces the amount of signaling required between sites to a minimum and is centered around the registration of Receiver-sites for a particular multicast-group or multicast-channel with the LISP Mapping System.

Registrations from the different receiver-sites will be merged at the Mapping System to assemble a multicast-replication-list inclusive of all RLOCs that lead to receivers for a particular multicast-group or multicast-channel. The replication-list for each specific multicast-entry is maintained as a LISP database mapping entry in the Mapping Database.

When the ITR at the source-site receives multicast traffic from sources at its site, the ITR can query the mapping system by issuing Map-Request messages for the (S,G) source and destination addresses in the packets received. The Mapping System will return the RLOC replication-list to the ITR, which the ITR will cache as per standard LISP procedure. Since the core is assumed to not support multicast, the ITR will replicate the multicast traffic for each RLOC on the replication-list and will unicast encapsulate the traffic to each RLOC. The combined function or replicating and encapsulating the traffic to the RLOCs in the replication-list is referred to as "rep-encapsulation" in this document.

The document describes the General Procedures and information encoding that are required at the Receiver-sites and Source-sites to achieve signal-free multicast interconnectivity. The General Procedures for Mapping System Notifications to different sites are also described. A section dedicated to the specific case of SSM trees discusses the implications to the General Procedures for SSM multicast trees over different topological scenarios. At this stage ASM trees are not supported with LISP Signal-Free multicast.

2. Definition of Terms

LISP related terms, notably Map-Request, Map-Reply, Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Map-Server (MS) and Map-Resolver (MR) are defined in the LISP specification [RFC6830].

Extensions to the definitions in [RFC6830] for their application to multicast routing are documented in [RFC6831].

Terms defining interactions with the LISP Mapping System are defined in [RFC6833].

The following terms are consistent with the definitions in [RFC6830] and [RFC6831]. The terms are specific cases of the general terms and are here defined to facilitate the descriptions and discussions within this particular document.

Source: Multicast source end-point. Host originating multicast packets.

Receiver: Multicast group member end-point. Host joins multicast group as a receiver of multicast packets sent to the group.

Receiver-site: LISP site where multicast receivers are located.

Source-site: LISP site where multicast sources are located.

RP-site: LISP site where an ASM PIM Rendezvous Point is located. The RP-site and the Source-site may be the same in some situations.

Receiver-ETR: LISP xTR at the Receiver-site. This is a multicast ETR.

Source-ITR: LISP xTR at the Source-site. This is a multicast ITR.

RP-xTR: LISP xTR at the RP-site. This is typically a multicast ITR.

Replication-list: Mapping-entry containing the list of RLOCs that have registered Receivers for a particular multicast-entry.

Multicast-entry: A tuple identifying a multicast tree. Multicast-entries are in the form of (S-prefix, G-prefix).

Rep-encapsulation: The process of replicating and then encapsulating traffic to multiple RLOCs.

3. Reference Model

The reference model that will be used for the discussion of the Signal-Free multicast tree interconnection is illustrated in Figure 1.

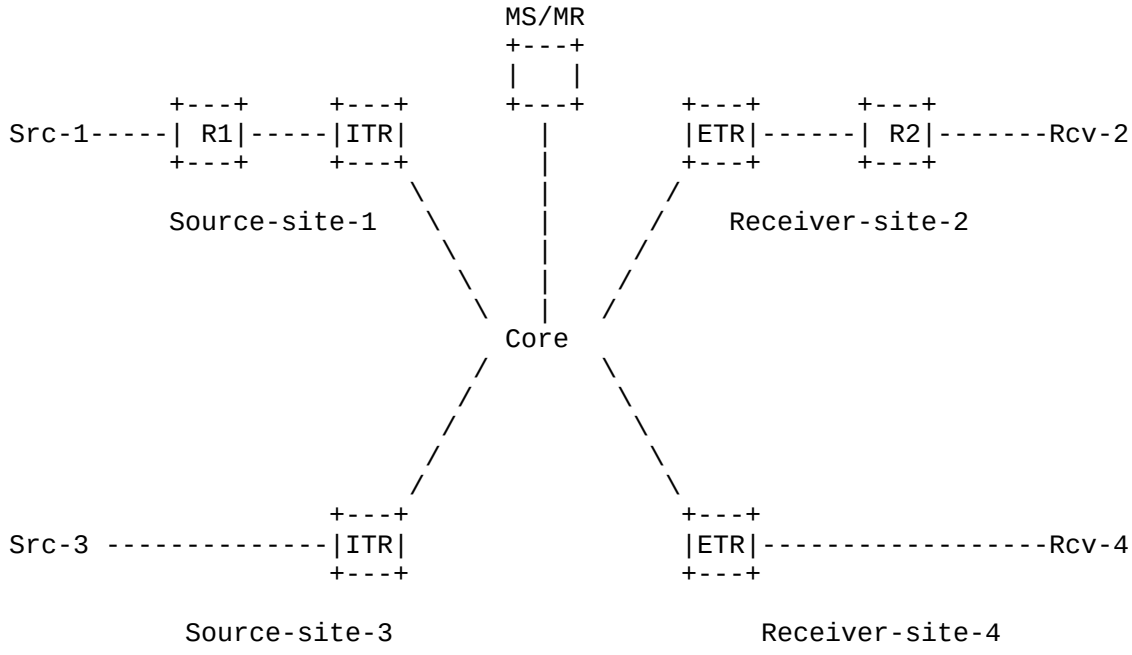


Figure 1: LISP Multicast Generic Reference Model

Sites 1 and 3 are Source-sites.

Source-site-3 presents a Source (Src-3) that is directly connected to the Source-ITR

Source-site-1 presents a Source (Src-1) that is one hop or more away from the Source-ITR

Receiver-site-2 and 4 are receiver sites with not-directly connected and directly connected Receiver end-points respectively

R1 is a router in Source-site-1.

R2 is a PIM router at the Receiver-site.

The Map-Servers and Resolvers are reachable in the RLOC space in the Core, only one is shown for illustration purposes, but these can be many or even part of a DDT tree.

The procedures for interconnecting multicast Trees over an overlay can be broken down into three functional areas:

- Receiver-site procedures
- Source-site procedures
- LISP notification procedures

The receiver site procedures will be common for most tree types and topologies.

The procedures at the source site can vary depending on the type of trees being interconnected as well as based on the topological relation between sources and source-site xTRs. For ASM trees, a special case of the Source-site is the RP-site for which a variation of the Source-site procedures may be necessary if ASM trees are to be supported in future specifications of LISP Signal-Free multicast.

The LISP notification procedures between sites are normalized for the different possible scenarios. Certain scenarios may benefit from a simplified notification mechanism or no notification requirement at all.

4. General Procedures

The interconnection of multicast trees across different LISP sites involves the following procedures to build the necessary multicast distribution trees across sites.

1. The presence of multicast Receiver end-points is detected by the Receiver-ETRs at the Receiver-sites.
2. Receiver-ETRs register their RLOCs as part of the replication-list for the multicast-entry the detected Receivers subscribe to.
3. The Mapping-system merges all receiver-ETR or delivery-group RLOCs to build a comprehensive replication-list inclusive of all Receiver-sites for each multicast-entry.
4. LISP Map-Notify messages should be sent to the Source-ITR informing of any changes in the replication-list.

5. Multicast-tree building at the Source-site is initiated when the Source-ITR receives the LISP Notification.

Once the multicast distribution trees are built, the following forwarding procedures may take place:

1. The Source sends multicast packets to the multicast group destination address.

2. Multicast traffic follows the multicast tree built at the Source-site and makes its way to the Source-ITRs.

3. The Source-ITR will issue a map-request to resolve the replication-list for the multicast-entry.

4. The Mapping System responds to the Source-ITR with a map-reply containing the replication-list for the multicast group requested.

5. The Source-ITR caches the replication-list received in the map-reply for the multicast-entry.

6. Multicast traffic is rep-encapsulated. That is, the packet is replicated for each RLOC in the replication-list and then encapsulated to each one.

4.1. General Receiver-site Procedures

4.1.1. Multicast receiver detection

When the Receiver-ETRs are directly connected to the Receivers (e.g. Receiver-site-4 in Figure 1), the Receiver-ETRs will receive IGMP Reports from the Receivers indicating which group the Receivers wish to subscribe to. Based on these IGMP Reports, the receiver-ETR is made aware of the presence of Receivers as well as which group they are interested in.

When the Receiver-ETRs are several hops away from the Receivers (e.g. Receiver-site-2 in Figure 1), the Receiver-ETRs will receive PIM join messages which will allow the Receiver-ETR to know that there are multicast Receivers at the site and also learn which multicast group the Receivers are for.

4.1.2. Receiver-site Registration

Once the Receiver-ETRs detect the presence of Receivers at the Receiver-site, the Receiver-ETRs will issue Map-Register messages to

include the Receiver-ETR RLOCs in the replication-list for the multicast-entry the Receivers joined.

The Map-Register message will use the multicast-entry (Source, Group) tuple as its EID record type with the Receiver-ETR RLOCs conforming the locator set.

The EID in the Map-Register message must be encoded using the Multicast Information LCAF type defined in [I-D.ietf-lisp-lcaf]. The R, L and J bits in the Multicast-info LCAF frame are not used and should be set to zero.

The RLOC in the Map-Register message must be encoded using the Replication List Entry (RLE) LCAF type defined in [I-D.ietf-lisp-lcaf] with the Level Value fields for all entries set to 128 (decimal).

The encoding described above must be used consistently for Map-Register messages, entries in the Mapping Database, Map-reply messages as well as the map-cache at the Source-ITRs.

The Map-Register messages [RFC6830] sent by the receiver-ETRs should have the following bits set as here specified:

1. merge-request-bit set to 1. The Map-Register messages must be sent with "Merge Semantics". The Map-Server will receive registrations from a multitude of Receiver-ETRs. The Map-Server will merge the registrations for common EIDs and maintain a consolidated replication-list for each multicast-entry.

2. want-map-notify-bit (M) set to 0. This tells the Mapping System that the receiver-ETR does not expect to receive Map-Notify messages as it does not need to be notified of all changes to the replication-list.

3. proxy-reply-bit (P) set to 1. The merged replication-list is kept in the Map-Servers. By setting the proxy-reply bit, the receiver-ETRs instruct the Mapping-system to proxy reply to map-requests issued for the multicast entries.

Map-Register messages for a particular multicast-entry should be sent for every receiver detected, even if previous receivers have been detected for the particular multicast-entry. This allows the replication-list to remain up to date.

4.1.3. Consolidation of the replication-list

The Map-Server will receive registrations from a multitude of Receiver-ETRs. The Map-Server will merge the registrations for common EIDs and consolidate a replication-list for each multicast-entry.

4.2. General Source-site Procedures

Source-ITRs must register the unicast EIDs of any Sources or Rendezvous Points that may be present on the Source-site. In other words, it is assumed that the Sources and RPs are LISP EIDs.

The registration of the unicast EIDs for the Sources or Rendezvous Points allows the map-server to know where to send Map-Notify messages to. Therefore, the Source-ITR must register the unicast S-prefix EID with the want-map-notify-bit set in order to receive Map-Notify messages whenever there is a change in the replication-list.

4.2.1. Multicast Tree Building at the Source-site

When the source site receives the Map-Notify messages from the mapping system as described in Section 4.3, it will initiate the process of building a multicast distribution tree that will allow the multicast packets from the Source to reach the Source-ITR.

The Source-ITR will issue a PIM join for the multicast-entry for which it received the Map-Notify message. The join will be issued in the direction of the source or in the direction of the RP for the SSM and ASM cases respectively.

4.2.2. Multicast Destination Resolution

On reception of multicast packets, the source-ITR must obtain the replication-list for the (S,G) addresses in the packets.

In order to obtain the replication-list, the Source-ITR must issue a Map-Request message in which the EID is the (S,G) multicast tuple which is encoded using the Multicast Info LCAF type defined in [I-D.ietf-lisp-lcaf].

The Mapping System (most likely the Map-Server) will Map-reply with the merged replication-list maintained in the Mapping System. The Map-reply message must follow the format defined in [RFC6830], its EID must be encoded using the Multicast Info LCAF type and the corresponding RLOC-records must be encoded using the RLE LCAF type. Both LCAF types defined in [I-D.ietf-lisp-lcaf].

4.3. General LISP Notification Procedures

The Map-Server will issue LISP Map-Notify messages to inform the Source-site of the presence of receivers for a particular multicast group over the overlay.

Updated Map-Notify messages should be issued every time a new registration is received from a Receiver-site. This guarantees that the source-sites are aware of any potential changes in the multicast-distribution-list membership.

The Map-Notify messages carry (S,G) multicast EIDs encoded using the Multicast Info LCAF type defined in [I-D.ietf-lisp-lcaf].

Map-Notify messages will be sent by the Map-Server to the RLOCs with which the unicast S-prefix EID was registered.

When both the Receiver-sites and the Source-sites register to the same Map-Server, the Map-Server has all the necessary information to send the Map-Notify messages to the Source-site.

When the Map-Servers are distributed in a DDT, the Receiver-sites may register to one Map-Server while the Source-site registers to a different Map-Server. In this scenario, the Map-Server for the receiver sites must resolve the unicast S-prefix EID in the DDT per standard LISP lookup procedures and obtain the necessary information to send the Map-Notify messages to the Source-site. The Map-Notify messages must be sent with an authentication length of 0 as they would not be authenticated.

When the Map-Servers are distributed in a DDT, different Receiver-sites may register to different Map-Servers. This is an unsupported scenario with the currently defined mechanisms.

5. Source Specific Multicast Trees

The interconnection of Source Specific Multicast (SSM) Trees across sites will follow the General Receiver-site Procedures described in Section 4.1 on the Receiver-sites.

The Source-site Procedures will vary depending on the topological location of the Source within the Source-site as described in Section 5.1 and Section 5.2 .

5.1. Source directly connected to Source-ITRs

When the Source is directly connected to the source-ITR, it is not necessary to trigger signaling to build a local multicast tree at the Source-site. Therefore Map-Notify messages may not be required to initiate building of the multicast tree at the Source-site.

Map-Notify messages are still required to ensure that any changes to the replication-list are communicated to the Source-site so that the map-cache at the Source-ITRs is kept updated.

5.2. Source not directly connected to Source-ITRs

The General LISP Notification Procedures described in Section 4.3 must be followed when the Source is not directly connected to the source-ITR. On reception of Map-Notify messages, local multicast signaling must be initiated at the Source-site per the General Source Site Procedures for Multicast Tree building described in Section 4.2.1.

In the SSM case, the IP address of the Source is known and it is also registered with the LISP mapping system. Thus, the mapping system may resolve the mapping for the Source address in order to send Map-Notify messages to the correct source-ITR.

6. PIM Any Source Multicast Trees

LISP signal-free multicast will not support ASM Trees at this time. A future revision of this specification may include procedures for PIM ASM support.

PIM ASM in shared-tree only mode could be supported in the scenario where the root of the shared tree (the PIM RP) is placed at the source site.

7. Signal-Free Multicast for Replication Engineering

The mechanisms in this draft can be applied to the LISP Replication-Engineering [I-D.coras-lisp-re] design. Rather than having the layered LISP-RE RTR hierarchy use signaling mechanisms, the RTRs can register their availability for multicast tree replication via the mapping database system. As stated in [I-D.coras-lisp-re], the RTR layered hierarchy is used to avoid head-end replication in replicating nodes closest to a multicast source. Rather than have multicast ITRs replicate to each ETR in an RLE entry of a (S,G) mapping database entry, it could replicate to one or more layer-0 RTRs in the LISP-RE hierarchy.

There are two formats an (S,G) mapping database entry could have. One format is a 'complete-format' and the other is a 'filtered-format'. A 'complete-format' entails an (S,G) entry having multiple RLOC records which contain both ETRs that have registered as well as the RTRs at the first level of the LISP-RE hierarchy for the ITR to replicate to. When using 'complete-format', the ITR has the ability to select if it replicates to RTRs or to the registered ETRs at the receiver sites. A 'filtered-format' (S,G) entry is one where the Map-Server returns the RLOC-records that it decides the ITR should use. So replication policy is shifted from the ITRs to the mapping system. The Map-Servers can also decide for a given ITR, if it uses a different set of replication targets per (S,G) entry for which the ITR is replicating for.

The procedure for the LISP-RE RTRs to make themselves available for replication can occur before or after any receivers join an (S,G) entry or any sources send for a particular (S,G) entry. Therefore, newly configured RTR state will be used to create new (S,G) state and inherited into existing (S,G) state. A set of RTRs can register themselves to the mapping system or a third-party can do so on their behalf. When RTR registration occurs, it is done with an (S-prefix, G-prefix) entry so it can advertise its replication services for a wide-range of source/group combinations.

When a Map-Server receives (S,G) registrations from ETRs and (S-prefix, G-prefix) registrations from RTRs, it has the option of merging the RTR RLOC-records for each (S,G) that is more-specific for the (S-prefix, G-prefix) entry or keep them separate. When merging, a Map-Server is ready to return a 'complete-format' Map-Reply. When keeping the entries separate, the Map-Server can decide what to include in a Map-Reply when a Map-Request is received. It can include a combination of RLOC-records from each entry or decide to use one or the other depending on policy configured.

Here is a specific example of (S,G) and (S-prefix, G-prefix) mapping database entries when a source S is behind an ITR and there are receiver sites joined to (S,G) via ETR1, ETR2, and ETR3. And there exists a LISP-RE hierarchy of RTR1 and RTR2 at level-0 and RTR3 and RTR4 at level-1:

```
EID-record: (S,G)
    RLOC-record: RLE: (ETR1, ETR2, ETR3), p1
EID-record: (S-prefix, G-prefix)
    RLOC-record: RLE: (RTR1(L0), RTR2(L0), RTR3(L1), RTR4(L1)), p1
```

The above entries are in the form of how they were registered and stored in a Map-Server. When a Map-Server uses 'complete-format', a Map-Reply it originates has the mapping record encoded as:

```
EID-record: (S,G)
    RLOC-record: RLE: (RTR1(L0), RTR3(L1)), p1
    RLOC-record: RLE: (ETR1, ETR2, ETR3), p1
```

The above Map-Reply allows the ITR to decide if it replicates to the ETRs or if it should replicate only to level-0 RTR1. This decision is left to the ITR since both RLOC-records have priority 1. If the Map-Server wanted to force the ITR to replicate to RTR1, it would set the ETRs RLOC-record to priority greater than 1.

When a Map_server uses "filtered-format', a Map-Reply it originates has the mapping record encoded as:

```
EID-record: (S,G)
    RLOC-record: RLE: (RTR1(L0), RTR3(L1)), p1
```

An (S,G) entry can contain alternate RTRs. So rather than replicating to multiple RTRs, one of a RTR set may be used based on the RTR reachability status. An ITR can test reachability status to any layer-0 RTR using RLOC-probing so it can choose one RTR from a set to replicate to. When this is done the RTRs are encoded in different RLOC-records versus together in one RLE RLOC-record. This moves the replication load off the ITRs at the source site to the RTRs inside the network infrastructure. This mechanism can also be used by level-n RTRs to level-n+1 RTRs.

The following mapping would be encoded in a Map-Reply sent by a Map-Server and stored in the ITR. The ITR would use RTR1 until it went unreachable and then switch to use RTR2:

```
EID-record: (S,G)
    RLOC-record: RTR1, p1
    RLOC-record: RTR2, p2
```

8. Security Considerations

[I-D.ietf-lisp-sec] defines a set of security mechanisms that provide origin authentication, integrity and anti-replay protection to LISP's EID-to-RLOC mapping data conveyed via mapping lookup process. LISP-SEC also enables verification of authorization on EID-prefix claims in Map-Reply messages.

Additional security mechanisms to protect the LISP Map-Register messages are defined in [RFC6833].

The security of the Mapping System Infrastructure depends on the particular mapping database used. The [I-D.ietf-lisp-ddt] specification, as an example, defines a public-key based mechanism

that provides origin authentication and integrity protection to the LISP DDT protocol.

Map-Replies received by the source-ITR can be signed (by the Map-Server) so the ITR knows the replication-list is from a legit source.

Data-plane encryption can be used when doing unicast rep-encapsulation as described in [I-D.ietf-lisp-crypto]. For further study we will look how to do multicast rep-encapsulation.

9. IANA Considerations

This document has no IANA implications

10. Acknowledgements

The authors want to thank Greg Shepherd, Joel Halpern and Sharon Barkai for their insightful contribution to shaping the ideas in this document. Thanks also goes to Jimmy Kyriannis, Paul Vinciguerra, and Florin Coras for testing an implementation of this draft.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <http://www.rfc-editor.org/info/rfc3618>.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <http://www.rfc-editor.org/info/rfc4601>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <http://www.rfc-editor.org/info/rfc4607>.

11.2. Informative References

[I-D.coras-lisp-re]
Coras, F., Cabellos-Aparicio, A., Domingo-Pascual, J., Maino, F., and D. Farinacci, "LISP Replication Engineering", draft-coras-lisp-re-08 (work in progress), November 2015.

[I-D.farinacci-lisp-mr-signaling]
Farinacci, D. and M. Napierala, "LISP Control-Plane Multicast Signaling", draft-farinacci-lisp-mr-signaling-06 (work in progress), February 2015.

[I-D.ietf-lisp-crypto]
Farinacci, D. and B. Weis, "LISP Data-Plane Confidentiality", draft-ietf-lisp-crypto-03 (work in progress), December 2015.

[I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-03 (work in progress), April 2015.

[I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-11 (work in progress), September 2015.

[I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., and D. Saucez, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-09 (work in progress), October 2015.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, DOI 10.17487/RFC6831, January 2013, <http://www.rfc-editor.org/info/rfc6831>.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, DOI 10.17487/RFC6833, January 2013, <http://www.rfc-editor.org/info/rfc6833>.

Appendix A. Document Change Log

A.1. Changes to draft-ietf-lisp-signal-free-multicast-00

- Posted late December 2015.
- Converted draft-farinacci-lisp-signal-free-multicast-04 into LISP working group draft.

A.2. Changes to draft-farinacci-lisp-signal-free-multicast-04

- Posted early December 2015.
- Update references and document timer.

A.3. Changes to draft-farinacci-lisp-signal-free-multicast-03

- Posted June 2015.
- Update references and document timer.

A.4. Changes to draft-farinacci-lisp-signal-free-multicast-02

- Posted December 2014.
- Added section about how LISP-RE can use the mechanisms from signal-free-multicast so we can avoid head-end replication and avoid signalling across a layered RE topology.

A.5. Changes to draft-farinacci-lisp-signal-free-multicast-01

- Posted June 2014.
- Changes based on implementation experience of this draft.

A.6. Changes to draft-farinacci-lisp-signal-free-multicast-00

- Posted initial draft February 2014.

Authors' Addresses

Victor Moreno
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: vimoreno@cisco.com

Dino Farinacci
lispers.net
San Jose, CA 95120
USA

Email: farinacci@gmail.com

LISP Working Group
Internet-Draft
Intended status: Informational
Expires: September 22, 2016

F. Maino
V. Ermagan
J. Evans
H. Miclea
Cisco Systems

March 21, 2016

GPE-VPN: Programmable LISP-based Virtual Private Networks
draft-maino-gpe-vpn-00

Abstract

GPE-VPN is an architecture for programmable SD-WAN solutions that leverages the Generic Protocol Encapsulation (GPE) overlay.

GPE-VPN uses an extended LISP-based map-assisted control plane to dynamically lookup forwarding policies on demand. A northbound programmable mapping system is used to store and retrieve mappings and forwarding policies.

The GPE-VPN data plane is secured with IPsec based encryption.

Overlay tunnels, as well as cryptographic parameters, are provisioned on demand.

Requirements Language

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

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 September 22, 2016.

Table of Contents

1. Introduction

GPE-VPN is an architecture for programmable Software Defined VPNs that leverages the Generic Protocol Encapsulation (GPE) overlay [I-D.ietf-nvo3-vxlan-gpe].

GPE is effectively merging VXLAN [RFC7348] and LISP [RFC6830] encapsulation in a single format with supports for multi-tenancy and multi-protocol payloads.

GPE-VPN uses an extended LISP-based map-assisted control plane to dynamically lookup forwarding policies on demand. A controller-based mapping system is used to store and retrieve the mapping and forwarding policies. The mapping system is programmable via northbound API.

GPE-VPN data plane is secured with IPsec based encryption.

Overlay tunnels, as well as cryptographic parameters, are provisioned on demand.

2. Definition of Terms

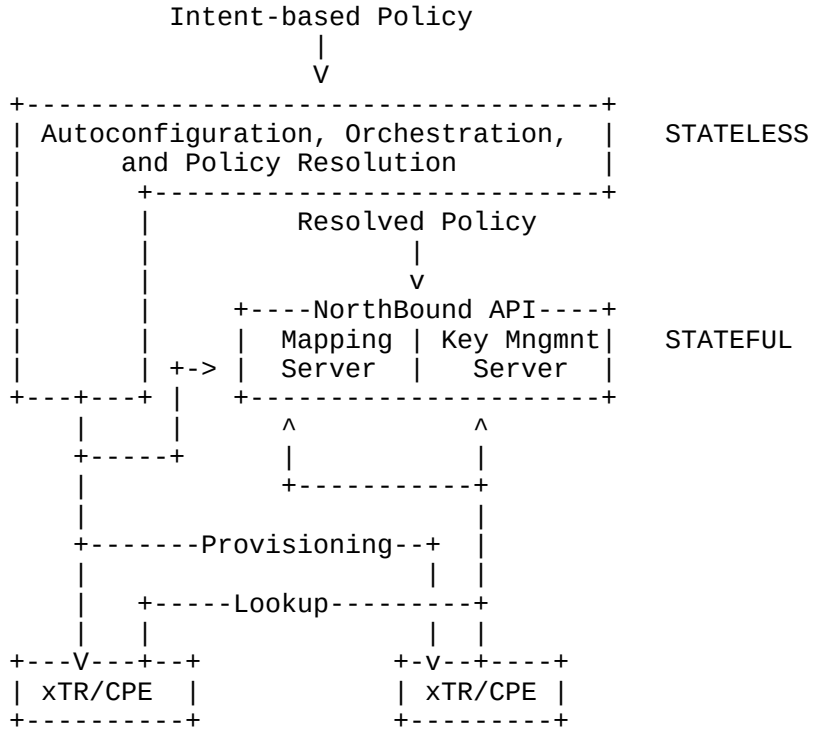
CPE: Customer-premises equipment or customer-provided equipment (CPE) is the VPN Tunnel Endpoint that enables access to the VPN. In this memo CPE and xTR are used interchangeably.

GPE: Generic Protocol Encapsulation. In this memo is used to refer to both VXLAN-GPE and LISP-GPE frame formats.

For definitions of other terms, notably Map-Request, Map-Reply, Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Map-Server (MS), and Map-Resolver (MR) please consult the LISP specification [RFC6830].

3. GPE-VPN Overall Architecture

A GPE-VPN is designed to enable VPN administrators to specify a high level intent-based policy that shall be implemented by the VPN. This includes policies such as connectivity, encryption, access control, and service chaining.

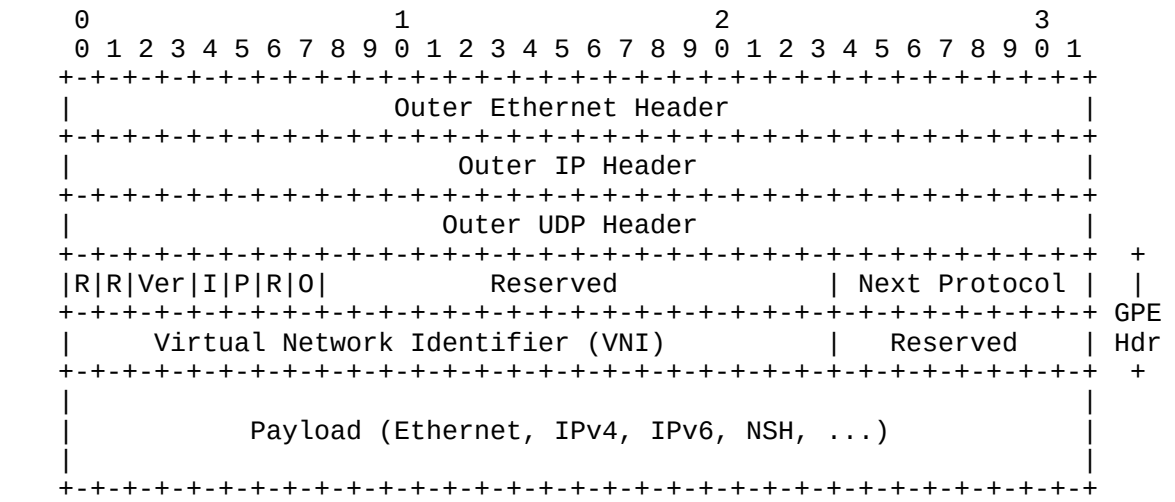


GPE-VPN Architecture

As specified by the intent-based policy, the GPE-VPN auto-configures the CPEs that implement the data plane of the VPN, and orchestrates and provisions the infrastructure needed to operate the VPN. The intent-based policy is then resolved into network forwarding policy, and is stored in the GPE-VPN mapping infrastructure along with other provisioned network state. The network state is then made available, on demand, to the CPEs using the LISP control protocol. Similarly, the cryptographic state is provisioned on demand by the Key Management Server.

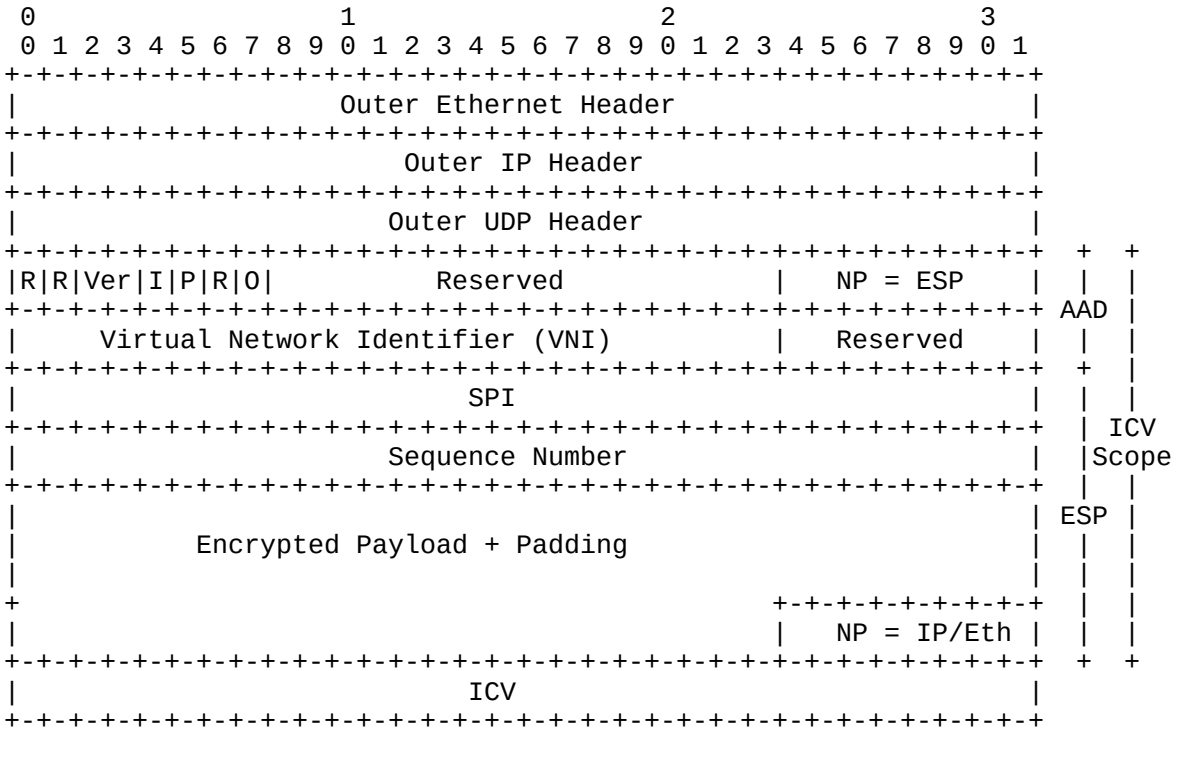
4. Data Plane Encapsulation

In a GPE-VPN frames are encapsulated accordingly to the VXLAN-GPE specification.



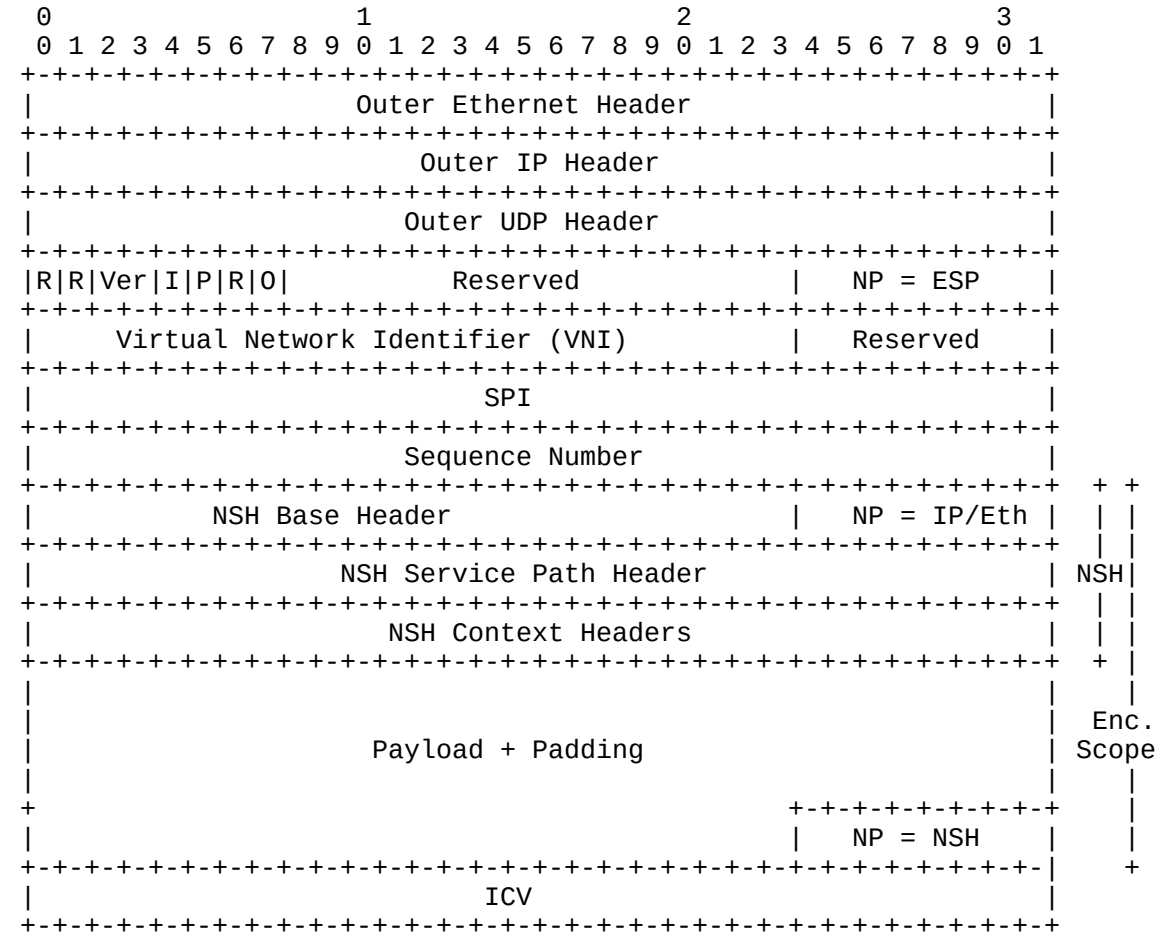
VXLAN-GPE Encapsulation

Confidentiality and integrity protection are afforded by the use of ESP (Encapsulating Security Payload) in transport mode. ESP is used with an authenticated encryption with associated data (AEAD) cipher such as GCM [RFC4106] that provides confidentiality, integrity, and anti-replay protection to the payload. The use of an AEAD cipher provides integrity and anti-replay protection to the GPE header as well as protection for the Virtual Network Identifier (VNI) and the other GPE fields from being hijacked over the network.



GPE with ESP-GCM

GPE can also be used in combination with the Network Service Header (NSH), as defined in [I-D.ietf-sfc-nsh], to provide application-level service chaining. Encryption and NSH are combined thanks to GPE multiprotocol support.



GPE+NSH with ESP-GCM

5. Data Plane Operations

in a GPE-VPN the CPE performs two main data plane functions:

1. GPE encapsulate un-encapsulated packets that are routed in the overlay network

2. De-capsulate GPE encapsulated packets that are routed in the underlay network

In order to perform the encapsulation function, the CPE uses a map-cache that maps the flow in the overlay to the location(s) (IP address in the underlay network) of the next hop or the destination

CPE depending on the mapping/forwarding policy defined in the mapping system.

The map cache is populated on demand using the LISP [RFC6830] map-request/map-reply protocol. In order to support multi-tenancy, each entry in the map-cache is associated with a VNI that identifies the overlay network of a specific tenant.

The map-cache supports multi-homing and load balancing by supporting mapping a single overlay end point to multiple locations in the underlay along with their priority and weight.

5.1. Per Destination Mapping

The simplest form of mapping supported by a map-cache is the mapping between the Endpoint Identifier (EID, the destination IP or MAC address of the endpoint in the overlay space) and the locator IP address of the destination CPE (its IP address in the underlay space).

EIDs can be either an IP address (L3 overlay) or a MAC address (L2 overlay).

This is the basic function that allows to interconnect VPN sites creating a virtual overlay network that constitutes the VPN. Any endpoint identified by a source EID in a VPN, can reach any other endpoint identified by a destination EID as long as the CPE has an entry in its map-cache for that destination EID.

5.2. FlowMapping

Some CPEs have the capability to map a generic n-tuple (typically a subset of the <source EID, destination EID, source Port, destination Port, Protocol> as defined in [I-D.rodrigueznatal-lisp-multi-tuple-eids]) onto the next hop or destination CPE location .

This allows a much finer granularity in applying the connectivity policy of the VPN. A different connectivity policy can be applied to each pair of endpoints in the overlay, or even per each protocol.

Per Destintion Mapping is, in fact, a subset of FlowMapping.

5.3. Generic Mapping

Some CPEs have the capability to map and encap based on a generic "tag" (metadata contained in the packet).

This enables GPE-VPN to offer overlay services to various protocols and applications, as a specific flow is tunneled to a given destination CPE based on the value of the metadata.

As an example a packet may include an NSH header [I-D.ietf-sfc-nsh] that contains metadata used to identify an application-level service function chain that should be applied to that packet before reaching destination. The map-cache can be programmed [I-D.ermagan-lisp-nsh] to tunnel that packet to the service node that implements the next hop in that service function chain and, eventually, to its destination.

5.4. Interworking

Interworking between the VPN and outside networks (e.g. the Internet) is provided by special gateways (LISP PxTRs) that support the encap and decap function for incoming (to the VPN) and outgoing packets.

Gateways also provide reachability to the VPN endpoints from the outside network by advertising highly aggregated EID-Prefix space on behalf of the GPE-VPN.

6. Control Plane Operations

The rendering of the connectivity policy between GPE-VPN sites is based on map-and-encap. When an un-encapsulated flow reaches a CPE, the CPE uses the LISP map-request/map-reply protocol to query the mapping system for the location of the next hop associated with this flow.

The CPE then caches the map-reply that contains the mapping associated with the given flow in its map-cache, so that subsequent packets in this flow hitting that CPE can be encapsulated right away. The map-cache entries are aged and refreshed accordingly to their utilization.

The map-request is encoded using an extensible format [I-D.ietf-lisp-lcaf] that can include a rich set of information not only about the specific flow that has generated the map-request, but also about the CPE sending the request.

As an example, the location of the source CPE can be included in the map-request, or its unique ID, providing the mapping server with information about the current location of the source endpoint that initiated that flow.

6.1. Dynamic Policy Rendering

The map-request is sent to a logically centralized map server that is programmed, via northbound API, with the rendering of the high level policies of the GPE-VPN.

The mapping is dynamically updated to reflect both high level policy changes, as well as changes to the state of the network (as measured by various metrics, including probing sent in the data plane, or reachability information registered by the CPEs). In general telemetric infrastructures can be used to provide the mapping server with a very detailed monitoring of both the underlay and overlay network. In a typical GPE-VPN implementation a telemetry server will collect telemetry data from the network via southbound API. Data is fed to an analytics engine that will update the mapping server via northbound API.

The mapping server can then leverage this amount of information to provide a map-reply that is not only the rendering of the connectivity policy, but also the rendering of a number of other policies applied to the VPN (overlar re-encapsulation, in-bound load balancing, virtual topologies, group based access control, service chaining, ...).

This is one of the most powerful characteristics of a GPE-VPN that makes the mapping server a logically centralized policy enforcement point.

To allow for a more dynamic policy change, the LISP protocol is being extended [I-D.rodrigueznatal-lisp-ms-smr] to support map-server notifications that are used to implement publish/subscribe mechanisms.

Below is a list of the most common policy renderings that may be implemented in a GPE-VPN.

6.1.1. In-Bound Load Balancing

For each overlay end point, the mapping system can specify a list of the locator IP addresses associated with the destination CPE. Each locator is associated with a priority and weight that will determine the in-bound load balancing at the destination CPE.

The sending CPE, upon receiving a map-reply, will encapsulate the outgoing traffic according to the priority and weight associated with the locators in the mapping. This is used to implement active/active or active/stand-by inbound load balancing.

6.1.2. Overlay Re-encapsulation

The high level policy of a GPE-VPN may include overlay re-encapsulation at Re-encapsulating Tunnel Routers (RTRs). The re-encapsulation network function is rendered by the mapping system by providing a different next hop to mapping requests coming from different CPEs/routers.

This section provides a few examples of the use of the re-encapsulation network function.

6.1.2.1. Virtual Topologies

While a GPE-VPN supports any-to-any connectivity, it is possible to implement virtual topologies by manipulating the mappings and using overlay re-encapsulation.

For example to implement an hub-and-spoke topology, the mapping system will be programmed in such a way that map-requests coming from a spoke will always generate map-replays containing the address of the hub as destination locator. In this way all the traffic generated at a spoke will be directed to the hub first, de-capsulated and then re-encapsulated to the destination spoke.

6.1.2.2. Hierarchical VPNs

some GPE-VPNs may have an hierarchical structure with CPEs grouped in regions that convey traffic to a core via a set od data centers positioned at the edge of the core. Depending on its geographical location, a CPE will send traffic to the RTR located at the edge data center that oversees that region.

Re-encapsulation may be required for a number of reasons, including traffic inspection.

Once a flow from a VPN site A directed to VPN site B hits CPE A, it will generate a map-request that will contain not only the destination EID, but also information about the requesting CPE (e.g. its location, or a site ID). The mapping system, in order to render the hierarchical VPN policy, will return a map-replay to CPE A specifying the location of the re-encapsulating router R as tunnel destination. Once the flow gets to the re-encapsulating router R, it will generate another map-request, for the same flow, but with attributes associated with router R. The mapping system will then render the hierarchical VPN policy by returning the locations of CPE B.

The dynamic manipulation of the mapping is what allows the mapping system to render complex re-encapsulation policies.

6.1.3. Group Based Access Control

Mapping manipulation can also be used to render Group Based Policies (GBP). The intent-based GBP will define groups of endpoints, and the ACLs that shall apply to each group. This is resolved into mapping state so that when the mapping system is resolving a map-request to connect endpoint A to endpoint B, the mapping will reflect the GBP policy, and will not provide connectivity if there's an ACL preventing communication between the groups to which A and B belongs.

6.1.4. Service Chaining

A GPE-VPN renders application-level Service Chaining Policies by using the mapping system to support the NSH protocol, as described in [I-D.ermagan-lisp-nsh]. NSH uses classification engines to classify flows that should be forced through a given service chain and tags them with an NSH header that contains a certain Service Path Identifier (SPI). Once classified and tagged, the packet needs to be routed to the associated next hop service node(s) that will implement the service chain. The mapping system in this case is programmed to support an SPI to Service Node Location lookup, so that a CPE receiving a packet with a given SPI and Service Index can lookup the address(es) of the next hop service nodes for the specified service chain.

This is an example of a generic mapping service provided by the GPE-VPN mapping system to support a specific protocol other than IP or Ethernet.

6.2. Key Management Services

Provisioning of Security Associations (SAs) for a GPE-VPN is a trade-off between the time needed to set up on demand the security association and the overall security afforded by the GPE-VPN security infrastructure.

There is a continuum of solutions that can be listed in (approximate) increasing order of security afforded:

1. Leverage the LISP map-request/map-reply protocol to provision on demand uni-directional security associations. The LISP crypto draft [I-D.ietf-lisp-crypto], is an example where the LISP map-request/map-reply protocol is augmented with an un-authenticated Diffie-Hellman exchange that provisions encryption keys to the CPEs tunnel endpoints. These mechanisms provide the most

efficient setup time of the SAs (that in [I-D.ietf-lisp-crypto] requires just an additional round-trip between the CPEs) as a trade-off with the security afforded, that relies on the security of the LISP map-request/map-replay protocol and on the relatively simple structure of the security messages exchanged between the CPEs.

2. Leverage a Group Domain of Interpretation (GDOI) crypto protocol, such as the GDOI IPsec extension [RFC6407], for key management. These protocols use group key distribution mechanisms to securely provision IPsec encryption keys (and SA parameters) to the CPEs that belong to the same GPE-VPN. This typically requires slightly longer SA setup times, compared with the previous solutions, but the key management infrastructure is independent from the LISP mapping infrastructure. This, typically, offers a higher level of security compared to the previous solutions.

3. Leverage the traditional IKEv2 [RFC5996] protocol to negotiate pairwise SAs between CPEs. While this typically offers the highest level of security, the setup of SAs is quite time consuming, and it has a significant impact on the advantages introduced by creating tunnels on demand using a map-and-encap protocol.

7. Security Considerations

Security considerations that applies to a GPE-VPN are discuss through this memo.

8. IANA Considerations

No IANA considerations.

9. Acknowledgements

10. Normative References

[I-D.ermagan-lisp-nsh]
Ermagan, V., Quinn, P., Lewis, D., Maino, F., and F. Coras, "LISP Control Plane integration with NSH", draft-ermagan-lisp-nsh-00 (work in progress), October 2015.

[I-D.ietf-lisp-crypto]
Farinacci, D. and B. Weis, "LISP Data-Plane Confidentiality", draft-ietf-lisp-crypto-03 (work in progress), December 2015.

[I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-11 (work in progress), September 2015.

[I-D.ietf-nvo3-vxlan-gpe]
Quinn, P., Manur, R., Kreeger, L., Lewis, D., Maino, F., Smith, M., Agarwal, P., Yong, L., Xu, X., Elzur, U., Garg, P., and D. Melman, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-01 (work in progress), November 2015.

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-02 (work in progress), January 2016.

[I-D.rodrigueznatal-lisp-ms-smr]
Rodriguez-Natal, A., Cabellos-Aparicio, A., Ermagan, V., Maino, F., and S. Barkai, "MS-originated SMRs", draft-rodrigueznatal-lisp-ms-smr-00 (work in progress), September 2015.

[I-D.rodrigueznatal-lisp-multi-tuple-eids]
Rodriguez-Natal, A., Cabellos-Aparicio, A., Barkai, S., Ermagan, V., Lewis, D., Maino, F., and D. Farinacci, "LISP support for Multi-Tuple EIDs", draft-rodrigueznatal-lisp-multi-tuple-eids-01 (work in progress), January 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", RFC 4106, DOI 10.17487/RFC4106, June 2005, <http://www.rfc-editor.org/info/rfc4106>.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, DOI 10.17487/RFC5996, September 2010, <http://www.rfc-editor.org/info/rfc5996>.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, DOI 10.17487/RFC6407, October 2011, <http://www.rfc-editor.org/info/rfc6407>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <http://www.rfc-editor.org/info/rfc6830>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <http://www.rfc-editor.org/info/rfc7348>.

Authors' Addresses

Fabio Maino
Cisco Systems

Email: fmaino@cisco.com

Vina Ermagan
Cisco Systems

Email: vermagan@cisco.com

John Evans
Cisco Systems

Email: joevans@cisco.com

Horia Miclea
Cisco Systems

Email: hmiclea@cisco.com