Network Working Group
Internet-Draft
Intended status: Experimental
Expires: July 21, 2014

L. Jakab
Cisco Systems
A. Cabellos-Aparicio
F. Coras
J. Domingo-Pascual
Technical University of Catalonia
D. Lewis
Cisco Systems

January 17, 2014

LISP Network Element Deployment Considerations
draft-ietf-lisp-deployment-12.txt

Abstract

This document is a snapshot of different Locator/Identifier Separation Protocol (LISP) deployment scenarios. It discusses the placement of new network elements introduced by the protocol, representing the thinking of the LISP working group as of Summer 2013. LISP deployment scenarios may have evolved since. This memo represents one stable point in that evolution of understanding.

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 July 21, 2014.

Table of Contents

1. Introduction

The Locator/Identifier Separation Protocol (LISP) is designed to address the scaling issues of the global Internet routing system identified in [RFC4984] by separating the current addressing scheme into Endpoint IDentifiers (EIDs) and Routing LOCators (RLOCs). The main protocol specification [RFC6830] describes how the separation is achieved, which new network elements are introduced, and details the packet formats for the data and control planes.

LISP assumes that such separation is between the edge and core and uses mapping and encapsulation for forwarding. While the boundary between both is not strictly defined, one widely accepted definition places it at the border routers of stub autonomous systems, which may carry a partial or complete default-free zone (DFZ) routing table. The initial design of LISP took this location as a baseline for protocol development. However, the applications of LISP go beyond just decreasing the size of the DFZ routing table, and include improved multihoming and ingress traffic engineering (TE) support for edge networks, and even individual hosts. Throughout the document we will use the term LISP site to refer to these networks/hosts behind a LISP Tunnel Router. We formally define the following two terms:

Network element: Facility or equipment used in the provision of a communications service over the Internet [TELCO96].

LISP site: A single host or a set of network elements in an edge network under the administrative control of a single organization, delimited from other networks by LISP Tunnel Router(s).

Since LISP is a protocol which can be used for different purposes, it is important to identify possible deployment scenarios and the additional requirements they may impose on the protocol specification and other protocols. Additionally, this document is intended as a guide for the operational community for LISP deployments in their networks. It is expected to evolve as LISP deployment progresses, and the described scenarios are better understood or new scenarios are discovered.

Each subsection considers an element type, discussing the impact of deployment scenarios on the protocol specification. For definition of terms, please refer to the appropriate documents (as cited in the respective sections).

This experimental document describing deployment considerations and the LISP specifications have areas that require additional experience and measurement. LISP is not recommended for deployment beyond experimental situations. Results of experimentation may lead to

modifications and enhancements of the LISP protocol mechanisms. Additionally, at the time of this writing there is no standardized security to implement. Beware that there are no counter measures for any of the threads identified in [I-D.ietf-lisp-threats]. See Section 15 [of RFC 6830] for specific, known issues that are in need of further work during development, implementation, and experimentation, and [I-D.ietf-lisp-threats] for recommendations to ameliorate the above-mentioned security threats.

2. Tunnel Routers

The device that is the gateway between the edge and the core is called a Tunnel Router (xTR), performing one or both of two separate functions:

1. Encapsulating packets originating from an end host to be transported over intermediary (transit) networks towards the other end-point of the communication

2. Decapsulating packets entering from intermediary (transit) networks, originated at a remote end host.

The first function is performed by an Ingress Tunnel Router (ITR), the second by an Egress Tunnel Router (ETR).

Section 8 of the main LISP specification [RFC6830] has a short discussion of where Tunnel Routers can be deployed and some of the associated advantages and disadvantages. This section adds more detail to the scenarios presented there, and provides additional scenarios as well. Furthermore this section discusses functional models, that is, network functions that can be achieved by deploying Tunnel Routers in specific ways.

2.1. Deployment Scenarios

2.1.1. Customer Edge

The first scenario we discuss is customer edge, when xTR functionality is placed on the router(s) that connect the LISP site to its upstream(s), but are under its control. As such, this is the most common expected scenario for xTRs, and this document considers it the reference location, comparing the other scenarios to this one.

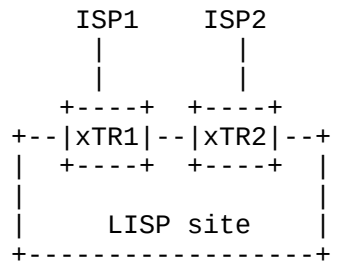


Figure 1: xTRs at the customer edge

From the LISP site perspective the main advantage of this type of deployment (compared to the one described in the next section) is having direct control over its ingress traffic engineering. This makes it easy to set up and maintain active/active, active/backup, or more complex TE policies, adding ISPs and additional xTRs at will, without involving third parties.

Being under the same administrative control, reachability information of all ETRs is easier to synchronize, because the necessary control traffic can be allowed between the locators of the ETRs. A correct synchronous global view of the reachability status is thus available, and the Locator Status Bits (Loc-Status-Bits, defined in [RFC6830]) can be set correctly in the LISP data header of outgoing packets.

By placing the tunnel router at the edge of the site, existing internal network configuration does not need to be modified. Firewall rules, router configurations and address assignments inside the LISP site remain unchanged. This helps with incremental deployment and allows a quick upgrade path to LISP. For larger sites with many external connections, distributed in geographically diverse points of presence (PoPs), and complex internal topology, it may however make more sense to both encapsulate and decapsulate as soon as possible, to benefit from the information in the IGP to choose the best path (see Section 2.2.1 for a discussion of this scenario).

Another thing to consider when placing tunnel routers is MTU issues. Encapsulation increases the amount of overhead associated with each packet. This added overhead decreases the effective end-to-end path MTU (unless fragmentation and reassembly is used). Some transit networks are known to provide larger MTU than the typical value of 1500 bytes of popular access technologies used at end hosts (e.g., IEEE 802.3 and 802.11). However, placing the LISP router connecting to such a network at the customer edge could possibly bring up MTU issues, depending on the link type to the provider as opposed to the following scenario. See [RFC4459] for MTU considerations of tunneling protocols on how to mitigate potential issues. Still, even

with these mitigations, path MTU issues are still possible.

2.1.2. Provider Edge

The other location at the core-edge boundary for deploying LISP routers is at the Internet service provider edge. The main incentive for this case is that the customer does not have to upgrade the CE router(s), or change the configuration of any equipment. Encapsulation/decapsulation happens in the provider's network, which may be able to serve several customers with a single device. For large ISPs with many residential/business customers asking for LISP this can lead to important savings, since there is no need to upgrade the software (or hardware, if it's the case) at each client's location. Instead, they can upgrade the software (or hardware) on a few PE routers serving the customers. This scenario is depicted in Figure 2.

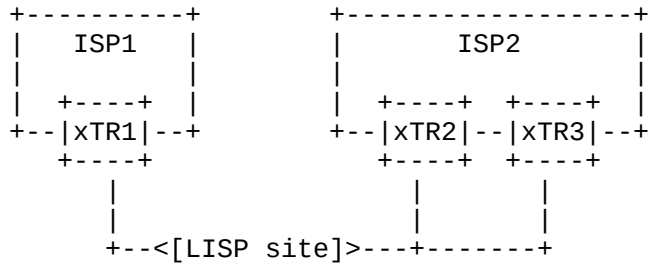


Figure 2: xTR at the PE

While this approach can make transition easy for customers and may be cheaper for providers, the LISP site loses one of the main benefits of LISP: ingress traffic engineering. Since the provider controls the ETRs, additional complexity would be needed to allow customers to modify their mapping entries.

The problem is aggravated when the LISP site is multihomed. Consider the scenario in Figure 2: whenever a change to TE policies is required, the customer contacts both ISP1 and ISP2 to make the necessary changes on the routers (if they provide this possibility). It is however unlikely, that both ISPs will apply changes simultaneously, which may lead to inconsistent state for the mappings of the LISP site. Since the different upstream ISPs are usually competing business entities, the ETRs may even be configured to compete, either to attract all the traffic or to get no traffic. The former will happen if the customer pays per volume, the latter if the connectivity has a fixed price. A solution could be to configure the Map Server(s) to do proxy-replying and have the Mapping Service Provider (MSP) apply policies.

Additionally, since xTR1, xTR2, and xTR3 are in different administrative domains, locator reachability information is unlikely to be exchanged among them, making it difficult to set Loc-Status-Bits (LSB) correctly on encapsulated packets. Because of this, and due to the security concerns about LSB described in [I-D.ietf-lisp-threats] their use is discouraged (set the L-bit to 0). Mapping versioning is another alternative [RFC6834].

Compared to the customer edge scenario, deploying LISP at the provider edge might have the advantage of diminishing potential MTU issues, because the tunnel router is closer to the core, where links typically have higher MTUs than edge network links.

2.1.3. Tunnel Routers Behind NAT

NAT in this section refers to IPv4 network address and port translation.

2.1.3.1. ITR

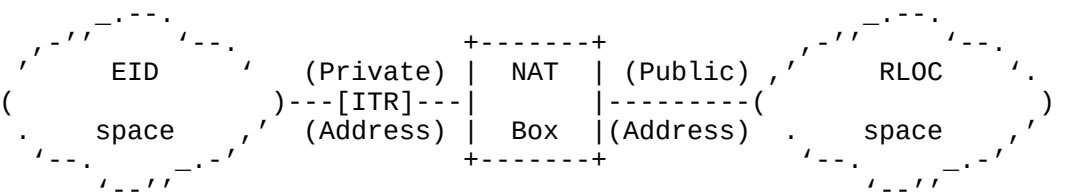


Figure 3: ITR behind NAT

Packets encapsulated by an ITR are just UDP packets from a NAT device's point of view, and they are handled like any UDP packet, there are no additional requirements for LISP data packets.

Map-Requests sent by an ITR, which create the state in the NAT table, have a different 5-tuple in the IP header than the Map-Reply generated by the authoritative ETR. Since the source address of this packet is different from the destination address of the request packet, no state will be matched in the NAT table and the packet will be dropped. To avoid this, the NAT device has to do the following:

- Send all UDP packets with source port 4342, regardless of the destination port, to the RLOC of the ITR. The most simple way to achieve this is configuring 1:1 NAT mode from the external RLOC of the NAT device to the ITR's RLOC (Called "DMZ" mode in consumer broadband routers).

o Rewrite the ITR-AFI and "Originating ITR RLOC Address" fields in the payload.

This setup supports only a single ITR behind the NAT device.

2.1.3.2. ETR

An ETR placed behind NAT is reachable from the outside by the Internet-facing locator of the NAT device. It needs to know this locator (and configure a loopback interface with it), so that it can use it in Map-Reply and Map-Register messages. Thus support for dynamic locators for the mapping database is needed in LISP equipment.

Again, only one ETR behind the NAT device is supported.

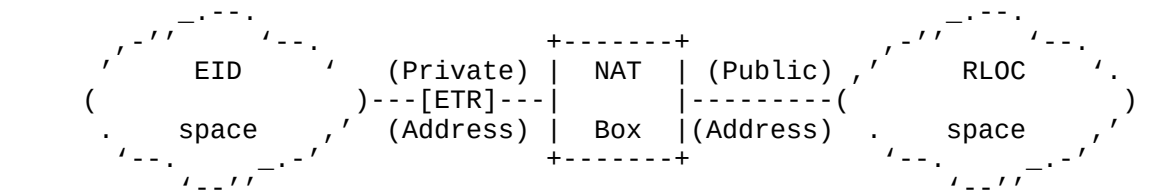


Figure 4: ETR behind NAT

2.1.3.3. Additional Notes

An implication of the issues described above is that LISP sites with xTRs can not be behind carrier based NATs, since two different sites would collide on the port forwarding. An alternative to static hole-punching to explore is the use of the Port Control Protocol (PCP) [RFC6887].

We only include this scenario due to completeness, to show that a LISP site can be deployed behind NAT, should it become necessary. However, LISP deployments behind NAT should be avoided, if possible.

2.2. Functional Models with Tunnel Routers

This section describes how certain LISP deployments can provide network functions.

2.2.1. Split ITR/ETR

In a simple LISP deployment, xTRs are located at the border of the LISP site (see Section 2.1.1). In this scenario packets are routed inside the domain according to the EID. However, more complex

networks may want to route packets according to the destination RLOC. This would enable them to choose the best egress point.

The LISP specification separates the ITR and ETR functionality and allows both entities to be deployed in separated network equipment. ITRs can be deployed closer to the host (i.e., access routers). This way packets are encapsulated as soon as possible, and egress point selection is driven by operational policy. In turn, ETRs can be deployed at the border routers of the network, and packets are decapsulated as soon as possible. Once decapsulated, packets are routed based on destination EID, according to internal routing policy.

In the following figure we can see an example. The Source (S) transmits packets using its EID and in this particular case packets are encapsulated at ITR_1. The encapsulated packets are routed inside the domain according to the destination RLOC, and can egress the network through the best point (i.e., closer to the RLOC's AS). On the other hand, inbound packets are received by ETR_1 which decapsulates them. Then packets are routed towards S according to the EID, again following the best path.

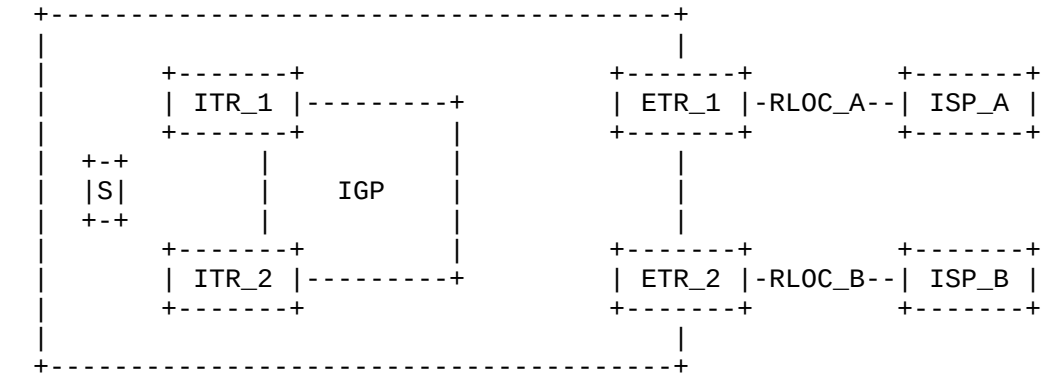


Figure 5: Split ITR/ETR Scenario

This scenario has a set of implications:

- The site must carry more specific routes in order to choose the best egress point, and typically BGP is used for this, increasing the complexity of the network. However, this is usually already the case for LISP sites that would benefit from this scenario.

- If the site is multihomed to different ISPs and any of the upstream ISPs are doing uRPF filtering, this scenario may become impractical. ITRs need to determine the exit ETR, for setting the

correct source RLOC in the encapsulation header. This adds complexity and reliability concerns.

- In LISP, ITRs set the reachability bits when encapsulating data packets. Hence, ITRs need a mechanism to be aware of the liveness of all ETRs serving their site.

- MTU within the site network must be large enough to accommodate encapsulated packets.

- In this scenario, each ITR is serving fewer hosts than in the case when it is deployed at the border of the network. It has been shown that cache hit ratio grows logarithmically with the amount of users [CACHE]. Taking this into account, when ITRs are deployed closer to the host the effectiveness of the mapping cache may be lower (i.e., the miss ratio is higher). Another consequence of this is that the site may transmit a higher amount of Map-Requests, increasing the load on the distributed mapping database.

- By placing the ITRs inside the site, they will still need global RLOCs, and this may add complexity to intra-site routing configuration, and further intra-site issues when there is a change of providers.

2.2.2. Inter-Service Provider Traffic Engineering

At the time of this writing, if two ISPs want to control their ingress TE policies for transit traffic between them, they need to rely on existing BGP mechanisms. This typically means deaggregating prefixes to choose on which upstream link packets should enter. This is either not feasible (if fine-grained per-customer control is required, the very specific prefixes may not be propagated) or increases DFZ table size.

Typically, LISP is seen applicable only to stub networks, however the LISP protocol can be also applied in a recursive manner, providing service provider ingress/egress TE capabilities without impacting the DFZ table size.

In order to implement this functionality with LISP consider the scenario depicted in Figure 6. The two ISPs willing to achieve ingress/egress TE are labeled as ISP_A and ISP_B, they are servicing Stub1 and Stub2 respectively, both are required to be LISP sites with their own xTRs. In this scenario we assume that Stub1 and Stub2 are communicating with each other and thus, ISP_A and ISP_B offer transit for such communications. ISP_A has RLOC_A1 and RLOC_A2 as upstream IP addresses while ISP_B has RLOC_B1 and RLOC_B2. The shared goal

among ISP_A and ISP_B is to control the transit traffic flow between RLOC_A1/A2 and RLOC_B1/B2.

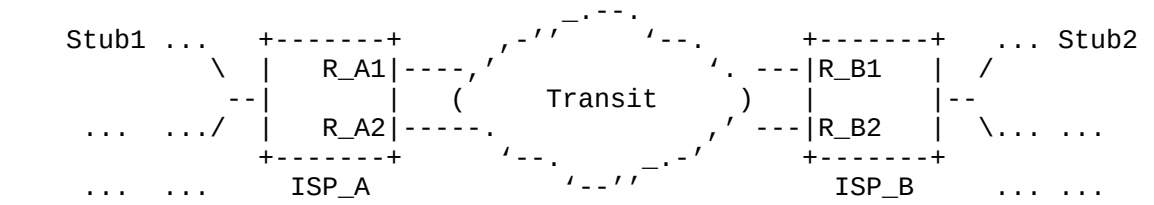


Figure 6: Inter-Service provider TE scenario

Both ISPs deploy xTRs on on RLOC_A1/A2 and RLOC_B1/B2 respectively and reach a bilateral agreement to deploy their own private mapping system. This mapping system contains bindings between the RLOCs of Stub1 and Stub2 (owned by ISP_A and ISP_B respectively) and RLOC_A1/A2 and RLOC_B1/B2. Such bindings are in fact the TE policies between both ISPs and the convergence time is expected to be fast, since ISPs only have to update/query a mapping to/from the database.

The packet flow is as follows. First, a packet originated at Stub1 towards Stub2 is LISP encapsulated by Stub1's xTR. The xTR of ISP_A recursively encapsulates it and, according to the TE policies stored in the private mapping system, the ISP_A xTR chooses RLOC_B1 or RLOC_B2 as the outer encapsulation destination. Note that the packet transits between ISP_A and ISP_B double-encapsulated. Upon reception at the xTR of ISP_B the packet is decapsulated and sent towards Stub2 which performs the last decapsulation.

This deployment scenario, which uses recursive LISP, includes three important caveats. First, it is intended to be deployed between only two ISPs. If more than two ISPs use this approach, then the xTRs deployed at the participating ISPs must either query multiple mapping systems, or the ISPs must agree on a common shared mapping system. Furthermore, keeping this deployment scenario restricted to only two ISPs maintains the solution scalable, given that only two entities need to agree on using recursive LISP, and only one private mapping system is involved.

Second, the scenario is only recommended for ISPs providing connectivity to LISP sites, such that source RLOCs of packets to be recursively encapsulated belong to said ISP. Otherwise the participating ISPs must register prefixes they do not own in the above mentioned private mapping system. This results in either requiring complex authentication mechanisms or enabling simple traffic redirection attacks. Failure to follow these recommendations may lead to operational security issues when deploying this scenario.

And third, recursive encapsulation models are typically complex to troubleshoot and debug.

Besides these recommendations, the main disadvantages of this deployment case are:

- Extra LISP header is needed. This increases the packet size and requires that the MTU between both ISPs accommodates double-encapsulated packets.
- The ISP ITR must encapsulate packets and therefore must know the RLOC-to-RLOC binding. These bindings are stored in a mapping database and may be cached in the ITR's mapping cache. Cache misses lead to an additional lookup latency, unless a push based mapping system is used for the private mapping system.
- The operational overhead of maintaining the shared mapping database.

2.3. Summary and Feature Matrix

When looking at the deployment scenarios and functional models above, there are several things to consider when choosing the approprate one, depending on the type of the organization doing the deployment.

For home users and small site who wish to multihome and have control over their ISP options, the "CE" scenario offers the most advantages: it's simple to deploy, in some cases it only requires a software upgrade of the CPE, getting mapping serice, and configuring the router. It ratains control of TE and choosing upstreams by the user. It doesn't provide too many advantages to ISPs, due to the lessened dependence on their services in case of multihomed clients. It is also unlikely that ISP wiching to offer LISP to their customers will choose the "CE" placement: they need to send a technician to each customer, and potentially a new CPE. Even if they have remote control over the router, and a software upgrade could add LISP support, the operation is too risky.

For a network operator a good option to deploy is the "PE" scenario, unless a hardware upgrade is required for its edge routers to support LISP (in which case upgrading CPEs may be simpler). It retains control of TE, choice of PETR, and MS/MR. It also lowers potential MTU issues, as dicussed above. Network operators should also explore the "Inter-SP TE" (recursive) functional model for their TE needs.

Large organizations can benefit the most from the "Split ITR/ETR" functional model, to optimize their traffic flow.

The following table gives a quick overview of the features supported by each of the deployment scenarios discussed above (marked with an "x") in the appropriate column: "CE" for customer edge, "PE" for provider edge, "Split" for split ITR/ETR, and "Recursive" for inter-service provider traffic engineering. The discussed features include:

Control of ingress TE: The scenario allows the LISP site to easily control LISP ingress traffic engineering policies.

No modifcations to existing int. network infrastruncture: The scenario doesn't require the LISP site to modify internal network configurations.

Loc-Status-Bits sync: The scenario allows easy synchronization of the Locator Status Bits.

MTU/PMTUD issues minimized: The scenario minimizes potential MTU and Path MTU Discovery issues.

Feature	CE	PE	Split	Recursive	NAT
Control of ingress TE	x	-	x	x	x
No modifications to existing int. network infrastructure	x	x	-	-	x
Loc-Status-Bits sync	x	-	x	x	-
MTU/PMTUD issues minimized	-	x	-	-	-

3. Map Resolvers and Map Servers

Map Resolvers and Map Servers make up the LISP mapping system and provide a means to find authoritative EID-to-RLOC mapping information, conforming to [RFC6833]. They are meant to be deployed in RLOC space, and their operation behind NAT is not supported.

3.1. Map Servers

The Map Server learns EID-to-RLOC mapping entries from an authoritative source and publishes them in the distributed mapping database. These entries are learned through authenticated Map-Register messages sent by authoritative ETRs. Also, upon reception of a Map-Request, the Map Server verifies that the destination EID matches an EID-prefix for which it is authoritative for, and then re-encapsulates and forwards it to a matching ETR. Map Server functionality is described in detail in [RFC6833].

The Map Server is provided by a Mapping Service Provider (MSP). The MSP participates in the global distributed mapping database infrastructure, by setting up connections to other participants, according to the specific mapping system that is employed (e.g., ALT [RFC6836], DDT [I-D.ietf-lisp-ddt]). Participation in the mapping database, and the storing of EID-to-RLOC mapping data is subject to the policies of the "root" operators, who should check ownership rights for the EID prefixes stored in the database by participants. These policies are out of the scope of this document.

The LISP DDT protocol is used by LISP Mapping Service providers to provide reachability between those providers' Map-Resolvers and Map-Servers. The DDT Root is currently operated by a collection of organizations on an open basis. See [DDT-ROOT] for more details. Similarly to the DNS root, it has several different server instances using names of the letters of the Greek alphabet (alpha, delta, etc.), operated by independent organizations. When this document was published, there were 5 such instances, one of them being anycasted. The Root provides the list of server instances on their web site and configuration files for several map server implementations. The DDT Root, and LISP Mapping Providers both rely on and abide by existing allocation policies by Regional Internet Registries to determine prefix ownership for use as EIDs.

It is expected that the DDT root organizations will continue to evolve in response to experimentation with LISP deployments for Internet edge multi-homing and VPN use cases.

In all cases, the MSP configures its Map Server(s) to publish the prefixes of its clients in the distributed mapping database and start encapsulating and forwarding Map-Requests to the ETRs of the AS. These ETRs register their prefix(es) with the Map Server(s) through periodic authenticated Map-Register messages. In this context, for some LISP sites, there is a need for mechanisms to:

- Automatically distribute EID prefix(es) shared keys between the ETRs and the EID-registrar Map Server.

- Dynamically obtain the address of the Map Server in the ETR of the AS.

The Map Server plays a key role in the reachability of the EID-prefixes it is serving. On the one hand it is publishing these prefixes into the distributed mapping database and on the other hand it is encapsulating and forwarding Map-Requests to the authoritative ETRs of these prefixes. ITRs encapsulating towards EIDs under the responsibility of a failed Map Server will be unable to look up any of their covering prefixes. The only exception are the ITRs that

already contain the mappings in their local cache. In this case ITRs can reach ETRs until the entry expires (typically 24 hours). For this reason, redundant Map Server deployments are desirable. A set of Map Servers providing high-availability service to the same set of prefixes is called a redundancy group. ETRs are configured to send Map-Register messages to all Map Servers in the redundancy group. The configuration for fail-over (or load-balancing, if desired) among the members of the group depends on the technology behind the mapping system being deployed. Since ALT is based on BGP and DDT was inspired from the Domain Name System (DNS), deployments can leverage current industry best practices for redundancy in BGP and DNS. These best practices are out of the scope of this document.

Additionally, if a Map Server has no reachability for any ETR serving a given EID block, it should not originate that block into the mapping system.

3.2. Map Resolvers

A Map Resolver is a network infrastructure component which accepts LISP encapsulated Map-Requests, typically from an ITR, and finds the appropriate EID-to-RLOC mapping by consulting the distributed mapping database. Map Resolver functionality is described in detail in [RFC6833].

Anyone with access to the distributed mapping database can set up a Map Resolver and provide EID-to-RLOC mapping lookup service. Database access setup is mapping system specific.

For performance reasons, it is recommended that LISP sites use Map Resolvers that are topologically close to their ITRs. ISPs supporting LISP will provide this service to their customers, possibly restricting access to their user base. LISP sites not in this position can use open access Map Resolvers, if available. However, regardless of the availability of open access resolvers, the MSP providing the Map Server(s) for a LISP site should also make available Map Resolver(s) for the use of that site.

In medium to large-size ASes, ITRs must be configured with the RLOC of a Map Resolver, operation which can be done manually. However, in Small Office Home Office (SOHO) scenarios a mechanism for autoconfiguration should be provided.

One solution to avoid manual configuration in LISP sites of any size is the use of anycast RLOCs [RFC4786] for Map Resolvers similar to the DNS root server infrastructure. Since LISP uses UDP encapsulation, the use of anycast would not affect reliability. LISP routers are then shipped with a preconfigured list of well know Map

Resolver RLOCs, which can be edited by the network administrator, if needed.

The use of anycast also helps improve mapping lookup performance. Large MSPs can increase the number and geographical diversity of their Map Resolver infrastructure, using a single anycasted RLOC. Once LISP deployment is advanced enough, very large content providers may also be interested running this kind of setup, to ensure minimal connection setup latency for those connecting to their network from LISP sites.

While Map Servers and Map Resolvers implement different functionalities within the LISP mapping system, they can coexist on the same device. For example, MSPs offering both services, can deploy a single Map Resolver/Map Server in each PoP where they have a presence.

4. Proxy Tunnel Routers

4.1. P-ITR

Proxy Ingress Tunnel Routers (P-ITRs) are part of the non-LISP/LISP transition mechanism, allowing non-LISP sites to reach LISP sites. They announce via BGP certain EID prefixes (aggregated, whenever possible) to attract traffic from non-LISP sites towards EIDs in the covered range. They do the mapping system lookup, and encapsulate received packets towards the appropriate ETR. Note that for the reverse path LISP sites can reach non-LISP sites simply by not encapsulating traffic. See [RFC6832] for a detailed description of P-ITR functionality.

The success of new protocols depends greatly on their ability to maintain backwards compatibility and inter-operate with the protocol(s) they intend to enhance or replace, and on the incentives to deploy the necessary new software or equipment. A LISP site needs an interworking mechanism to be reachable from non-LISP sites. A P-ITR can fulfill this role, enabling early adopters to see the benefits of LISP, similar to tunnel brokers helping the transition from IPv4 to IPv6. A site benefits from new LISP functionality (proportionally with existing global LISP deployment) when going LISP, so it has the incentives to deploy the necessary tunnel routers. In order to be reachable from non-LISP sites it has two options: keep announcing its prefix(es) with BGP, or have a P-ITR announce prefix(es) covering them.

If the goal of reducing the DFZ routing table size is to be reached, the second option is preferred. Moreover, the second option allows

LISP-based ingress traffic engineering from all sites. However, the placement of P-ITRs significantly influences performance and deployment incentives. Section 5 is dedicated to the migration to a LISP-enabled Internet, and includes deployment scenarios for P-ITRs.

4.2. P-ETR

In contrast to P-ITRs, P-ETRs are not required for the correct functioning of all LISP sites. There are two cases, where they can be of great help:

- o LISP sites with unicast reverse path forwarding (uRPF) restrictions, and
- o Communication between sites using different address family RLOCs.

In the first case, uRPF filtering is applied at their upstream PE router. When forwarding traffic to non-LISP sites, an ITR does not encapsulate packets, leaving the original IP headers intact. As a result, packets will have EIDs in their source address. Since we are discussing the transition period, we can assume that a prefix covering the EIDs belonging to the LISP site is advertised to the global routing tables by a P-ITR, and the PE router has a route towards it. However, the next hop will not be on the interface towards the CE router, so non-encapsulated packets will fail uRPF checks.

To avoid this filtering, the affected ITR encapsulates packets towards the locator of the P-ETR for non-LISP destinations. Now the source address of the packets, as seen by the PE router is the ITR's locator, which will not fail the uRPF check. The P-ETR then decapsulates and forwards the packets.

The second use case is IPv4-to-IPv6 transition. Service providers using older access network hardware, which only supports IPv4 can still offer IPv6 to their clients, by providing a CPE device running LISP, and P-ETR(s) for accessing IPv6-only non-LISP sites and LISP sites, with IPv6-only locators. Packets originating from the client LISP site for these destinations would be encapsulated towards the P-ETR's IPv4 locator. The P-ETR is in a native IPv6 network, decapsulating and forwarding packets. For non-LISP destination, the packet travels natively from the P-ETR. For LISP destinations with IPv6-only locators, the packet will go through a P-ITR, in order to reach its destination.

For more details on P-ETRs see [RFC6832].

P-ETRs can be deployed by ISPs wishing to offer value-added services

to their customers. As is the case with P-ITRs, P-ETRs too may introduce path stretch (the ratio between the cost of the selected path and that of the optimal path). Because of this the ISP needs to consider the tradeoff of using several devices, close to the customers, to minimize it, or few devices, farther away from the customers, minimizing cost instead.

Since the deployment incentives for P-ITRs and P-ETRs are different, it is likely they will be deployed in separate devices, except for the CDN case, which may deploy both in a single device.

In all cases, the existence of a P-ETR involves another step in the configuration of a LISP router. CPE routers, which are typically configured by DHCP, stand to benefit most from P-ETRs. Autoconfiguration of the P-ETR locator could be achieved by a DHCP option, or adding a P-ETR field to either Map-Notifys or Map-Replies.

5. Migration to LISP

This section discusses a deployment architecture to support the migration to a LISP-enabled Internet. The loosely defined terms of "early transition phase", "late transition phase", and "LISP Internet phase" refer to time periods when LISP sites are a minority, a majority, or represent all edge networks respectively.

5.1. LISP+BGP

For sites wishing to go LISP with their PI prefix the least disruptive way is to upgrade their border routers to support LISP, register the prefix into the LISP mapping system, but keep announcing it with BGP as well. This way LISP sites will reach them over LISP, while legacy sites will be unaffected by the change. The main disadvantage of this approach is that no decrease in the DFZ routing table size is achieved. Still, just increasing the number of LISP sites is an important gain, as an increasing LISP/non-LISP site ratio may decrease the need for BGP-based traffic engineering that leads to prefix deaggregation. That, in turn, may lead to a decrease in the DFZ size and churn in the late transition phase.

This scenario is not limited to sites that already have their prefixes announced with BGP. Newly allocated EID blocks could follow this strategy as well during the early LISP deployment phase, depending on the cost/benefit analysis of the individual networks. Since this leads to an increase in the DFZ size, the following architecture should be preferred for new allocations.

5.2. Mapping Service Provider (MSP) P-ITR Service

In addition to publishing their clients' registered prefixes in the mapping system, MSPs with enough transit capacity can offer them P-ITR service as a separate service. This service is especially useful for new PI allocations, to sites without existing BGP infrastructure, that wish to avoid BGP altogether. The MSP announces the prefix into the DFZ, and the client benefits from ingress traffic engineering without prefix deaggregation. The downside of this scenario is adding path stretch.

Routing all non-LISP ingress traffic through a third party which is not one of its ISPs is only feasible for sites with modest amounts of traffic (like those using the IPv6 tunnel broker services today), especially in the first stage of the transition to LISP, with a significant number of legacy sites. This is because the handling of said traffic is likely to result in additional costs, which would be passed down to the client. When the LISP/non-LISP site ratio becomes high enough, this approach can prove increasingly attractive.

Compared to LISP+BGP, this approach avoids DFZ bloat caused by prefix deaggregation for traffic engineering purposes, resulting in slower routing table increase in the case of new allocations and potential decrease for existing ones. Moreover, MSPs serving different clients with adjacent aggregatable prefixes may lead to additional decrease, but quantifying this decrease is subject to future research study.

5.3. Proxy-ITR Route Distribution (PITR-RD)

Instead of a LISP site, or the MSP, announcing their EIDs with BGP to the DFZ, this function can be outsourced to a third party, a P-ITR Service Provider (PSP). This will result in a decrease of the operational complexity both at the site and at the MSP.

The PSP manages a set of distributed P-ITR(s) that will advertise the corresponding EID prefixes through BGP to the DFZ. These P-ITR(s) will then encapsulate the traffic they receive for those EIDs towards the RLOCs of the LISP site, ensuring their reachability from non-LISP sites.

While it is possible for a PSP to manually configure each client's EID routes to be announced, this approach offers little flexibility and is not scalable. This section presents a scalable architecture that offers automatic distribution of EID routes to LISP sites and service providers.

The architecture requires no modification to existing LISP network elements, but it introduces a new (conceptual) network element, the

EID Route Server, defined as a router that either propagates routes learned from other EID Route Servers, or it originates EID Routes. The EID-Routes that it originates are those that it is authoritative for. It propagates these routes to Proxy-ITRs within the AS of the EID Route Server. It is worth to note that a BGP capable router can be also considered as an EID Route Server.

Further, an EID-Route is defined as a prefix originated via the Route Server of the mapping service provider, which should be aggregated if the MSP has multiple customers inside a single large continuous prefix. This prefix is propagated to other P-ITRs both within the MSP and to other P-ITR operators it peers with. EID Route Servers are operated either by the LISP site, MSPs or PSPs, and they may be collocated with a Map Server or P-ITR, but are a functionally discrete entity. They distribute EID-Routes, using BGP, to other domains, according to policies set by participants.

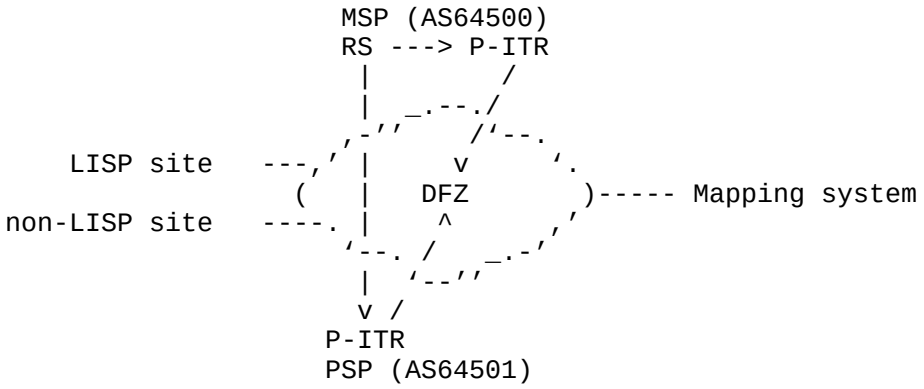


Figure 7: The P-ITR Route Distribution architecture

The architecture described above decouples EID origination from route propagation, with the following benefits:

- Can accurately represent business relationships between P-ITR operators
- More mapping system agnostic
- Minor changes to P-ITR implementation, no changes to other components

In the example in the figure we have a MSP providing services to the LISP site. The LISP site does not run BGP, and gets an EID allocation directly from a RIR, or from the MSP, who may be a LIR. Existing PI allocations can be migrated as well. The MSP ensures the

presence of the prefix in the mapping system, and runs an EID Route Server to distribute it to P-ITR service providers. Since the LISP site does not run BGP, the prefix will be originated with the AS number of the MSP.

In the simple case depicted in Figure 7 the EID-Route of LISP site will be originated by the Route Server, and announced to the DFZ by the PSP's P-ITRs with AS path 64501 64500. From that point on, the usual BGP dynamics apply. This way, routes announced by P-ITR are still originated by the authoritative Route Server. Note that the peering relationships between MSP/PSPs and those in the underlying forwarding plane may not be congruent, making the AS path to a P-ITR shorter than it is in reality.

The non-LISP site will select the best path towards the EID-prefix, according to its local BGP policies. Since AS-path length is usually an important metric for selecting paths, a careful placement of P-ITR could significantly reduce path-stretch between LISP and non-LISP sites.

The architecture allows for flexible policies between MSP/PSPs. Consider the EID Route Server networks as control plane overlays, facilitating the implementation of policies necessary to reflect the business relationships between participants. The results are then injected to the common underlying forwarding plane. For example, some MSP/PSPs may agree to exchange EID-Prefixes and only announce them to each of their forwarding plane customers. Global reachability of an EID-prefix depends on the MSP the LISP site buys service from, and is also subject to agreement between the mentioned parties.

In terms of impact on the DFZ, this architecture results in a slower routing table increase for new allocations, since traffic engineering will be done at the LISP level. For existing allocations migrating to LISP, the DFZ may decrease since MSPs may be able to aggregate the prefixes announced.

Compared to LISP+BGP, this approach avoids DFZ bloat caused by prefix deaggregation for traffic engineering purposes, resulting in slower routing table increase in the case of new allocations and potential decrease for existing ones. Moreover, MSPs serving different clients with adjacent aggregatable prefixes may lead to additional decrease, but quantifying this decrease is subject to future research study.

The flexibility and scalability of this architecture does not come without a cost however: A PSP operator has to establish either transit or peering relationships to improve their connectivity.

5.4. Migration Summary

Registering a domain name typically entails an annual fee that should cover the operating expenses for publishing the domain in the global DNS. The situation is similar with several other registration services. A LISP mapping service provider (MSR) client publishing an EID prefix in the LISP mapping system has the option of signing up for PITR services as well, for an extra fee. These services may be offered by the MSP itself, but it is expected that specialized P-ITR service providers (PSPs) will do it. Clients not signing up become responsible for getting non-LISP traffic to their EIDs (using the LISP+BGP scenario).

Additionally, Tier 1 ISPs have incentives to offer P-ITR services to non-subscribers in strategic places just to attract more traffic from competitors, thus more revenue.

The following table presents the expected effects of the different transition scenarios during a certain phase on the DFZ routing table size:

Phase	LISP+BGP	MSP P-ITR	PITR-RD
Early transition	no change	slower increase	slower increase
Late transition	may decrease	slower increase	slower increase
LISP Internet	considerable decrease		

It is expected that PITR-RD will co-exist with LISP+BGP during the migration, with the latter being more popular in the early transition phase. As the transition progresses and the MSP P-ITR and PITR-RD ecosystem gets more ubiquitous, LISP+BGP should become less attractive, slowing down the increase of the number of routes in the DFZ.

Note that throughout Section 5 we focused on the effects of LISP deployment on the DFZ route table size. Other metrics may be impacted as well, but to the best of our knowlegde have not been measured as of yet.

6. Security Considerations

All security implications of LISP deployments are to be discussed in separate documents. [I-D.ietf-lisp-threats] gives an overview of LISP threat models, including ETR operators attracting traffic by overclaiming an EID-prefix (Section 4.4.3). Securing mapping lookups is discussed in [I-D.ietf-lisp-sec].

7. IANA Considerations

This memo includes no request to IANA.

8. Acknowledgements

Many thanks to Margaret Wasserman for her contribution to the IETF76 presentation that kickstarted this work. The authors would also like to thank Damien Saucez, Luigi Iannone, Joel Halpern, Vince Fuller, Dino Farinacci, Terry Manderson, Noel Chiappa, Hannu Flinck, Paul Vinciguerra, Fred Templin, Brian Haberman, and everyone else who provided input.

9. References

9.1. Normative References

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.

9.2. Informative References

[CACHE] Jung, J., Sit, E., Balakrishnan, H., and R. Morris, "DNS performance and the effectiveness of caching", 2002.

[DDT-ROOT] "DDT Root", <http://ddt-root.org/>.

[I-D.ietf-lisp-ddt] Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-01 (work in progress), March 2013.

[I-D.ietf-lisp-sec] Maino, F., Ermagan, V., Cabellos-Aparicio, A., Saucez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-05 (work in progress), October 2013.

[I-D.ietf-lisp-threats]
Saucez, D., Iannone, L., and O. Bonaventure, "LISP Threats Analysis", draft-ietf-lisp-threats-08 (work in progress), October 2013.

[RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, April 2006.

[RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast Services", BCP 126, RFC 4786, December 2006.

[RFC4984] Meyer, D., Zhang, L., and K. Fall, "Report from the IAB Workshop on Routing and Addressing", RFC 4984, September 2007.

[RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, January 2013.

[RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, January 2013.

[RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

[TELCO96] "Telecommunications Act of 1996", 1996.

Appendix A. Step-by-Step Example BGP to LISP Migration Procedure

To help the operational community deploy LISP, this informative section offers a step-by-step guide for migrating a BGP based Internet presence to a LISP site. It includes a pre-install/ pre-turn-up checklist, and customer and provider activation procedures.

A.1. Customer Pre-Install and Pre-Turn-up Checklist

1. Determine how many current physical service provider connections the customer has and their existing bandwidth and traffic engineering requirements.

   This information will determine the number of routing locators, and the priorities and weights that should be configured on the xTRs.

2. Make sure customer router has LISP capabilities.

   * Check OS version of the CE router. If LISP is an add-on, check if it is installed.

     This information can be used to determine if the platform is appropriate to support LISP, in order to determine if a software and/or hardware upgrade is required.

   * Have customer upgrade (if necessary, software and/or hardware) to be LISP capable.

3. Obtain current running configuration of CE router. A suggested LISP router configuration example can be customized to the customer's existing environment.

4. Verify MTU Handling

   * Request increase in MTU to 1556 or more on service provider connections. Prior to MTU change verify that 1500 byte packet from P-xTR to RLOC with do not fragment (DF-bit) bit set.

   * Ensure they are not filtering ICMP unreachable or time-exceeded on their firewall or router.

   LISP, like any tunneling protocol, will increase the size of packets when the LISP header is appended. If increasing the MTU of the access links is not possible, care must be taken that ICMP is not being filtered in order to allow for Path MTU Discovery to take place.

5. Validate member prefix allocation.

   This step is to check if the prefix used by the customer is a direct (Provider Independent), or if it is a prefix assigned by a physical service provider (Provider Aggregatable). If the prefixes are assigned by other service providers then a Letter of Agreement is required to announce prefixes through the Proxy Service Provider.

6. Verify the member RLOCs and their reachability.

   This step ensures that the RLOCs configured on the CE router are in fact reachable and working.

7. Prepare for cut-over.

* If possible, have a host outside of all security and filtering policies connected to the console port of the edge router or switch.

* Make sure customer has access to the router in order to configure it.

A.2. Customer Activating LISP Service

1. Customer configures LISP on CE router(s) from service provider recommended configuration.

   The LISP configuration consists of the EID prefix, the locators, and the weights and priorities of the mapping between the two values. In addition, the xTR must be configured with Map Resolver(s), Map Server(s) and the shared key for registering to Map Server(s). If required, Proxy-ETR(s) may be configured as well.

   In addition to the LISP configuration, the following:

   * Ensure default route(s) to next-hop external neighbors are included and RLOCs are present in configuration.

   * If two or more routers are used, ensure all RLOCs are included in the LISP configuration on all routers.

   * It will be necessary to redistribute default route via IGP between the external routers.

2. When transition is ready perform a soft shutdown on existing eBGP peer session(s)

   * From CE router, use LIG to ensure registration is successful.

   * To verify LISP connectivity, find and ping LISP connected sites. If possible, find ping destinations that are not covered by a prefix in the global BGP routing system, because PITRs may deliver the packets even if LISP connectivity is not working. Traceroutes may help discover if this is the case.

   * To verify connectivity to non-LISP sites, try accessing a landmark (e.g., a major Internet site) via a web browser.

A.3. Cut-Over Provider Preparation and Changes

1. Verify site configuration and then active registration on Map Server(s)
    * Authentication key
    * EID prefix
2. Add EID space to map-cache on proxies
3. Add networks to BGP advertisement on proxies
    * Modify route-maps/policies on P-xTRs
    * Modify route policies on core routers (if non-connected member)
    * Modify ingress policers on core routers
    * Ensure route announcement in looking glass servers, RouteViews
4. Perform traffic verification test
    * Ensure MTU handling is as expected (PMTUD working)
    * Ensure proxy-ITR map-cache population
    * Ensure access from traceroute/ping servers around Internet
    * Use a looking glass, to check for external visibility of registration via several Map Resolvers

Authors' Addresses

Lorand Jakab
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Email: lojakab@cisco.com

Albert Cabellos-Aparicio
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: acabello@ac.upc.edu

Florin Coras
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: fcoras@ac.upc.edu

Jordi Domingo-Pascual
Technical University of Catalonia
C/Jordi Girona, s/n
BARCELONA 08034
Spain

Email: jordi.domingo@ac.upc.edu

Darrel Lewis
Cisco Systems
170 Tasman Drive
San Jose, CA 95134
USA

Email: darlewis@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 29, 2016

L. Iannone
Telecom ParisTech
D. Lewis
Cisco Systems, Inc.
D. Meyer
Brocade
V. Fuller

February 26, 2016

LISP EID Block
draft-ietf-lisp-eid-block-13.txt

Abstract

This is a direction to IANA to allocate a /32 IPv6 prefix for use with the Locator/ID Separation Protocol (LISP). The prefix will be used for local intra-domain routing and global endpoint identification, by sites deploying LISP as EID (Endpoint IDentifier) addressing space.

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 August 29, 2016.

Copyright Notice

Table of Contents

1. Introduction

This document directs the IANA to allocate a /32 IPv6 prefix for use with the Locator/ID Separation Protocol (LISP - [RFC6830]), LISP Map Server ([RFC6833]), LISP Alternative Topology (LISP+ALT - [RFC6836]) (or other) mapping systems, and LISP Interworking ([RFC6832]).

This block will be used as global Endpoint IDentifier (EID) space.

2. Definition of Terms

The present document does not introduce any new term with respect to the set of LISP Specifications ([RFC6830], [RFC6831], [RFC6832], [RFC6833], [RFC6834], [RFC6835], [RFC6836], [RFC6837]), but assumes that the reader is familiar with the LISP terminology. [I-D.ietf-lisp-introduction] provides an introduction to the LISP technology, including its terminology.

3. Rationale and Intent

Discussion within the LISP Working Group led to identify several scenarios in which the existence of a LISP specific address block brings technical benefits. Hereafter the most relevant scenarios are described:

Early LISP destination detection: With the current specifications, there is no direct way to detect whether or not a certain destination is in a LISP domain or not without performing a LISP mapping lookup. For instance, if an ITR is sending to all types of destinations (i.e., non-LISP destinations, LISP destinations not in the IPv6 EID block, and LISP destinations in the IPv6 EID block) the only way to understand whether or not to encapsulate the traffic is to perform a cache lookup and, in case of a LISP Cache miss, send a Map-Request to the mapping system. In the meanwhile (waiting the Map-Reply), packets may be dropped in order to avoid excessive buffering.

Avoid penalizing non-LISP traffic: In certain circumstances it might be desirable to configure a router using LISP features to natively forward all packets that have not a destination address in the block, hence, no lookup whatsoever is performed and packets destined to non-LISP sites are not penalized in any manner.

Traffic Engineering: In some deployment scenarios it might be desirable to apply different traffic engineering policies for LISP and non-LISP traffic. A LISP specific EID block would allow improved traffic engineering capabilities with respect to LISP vs. non-LISP traffic. In particular, LISP traffic might be identified without having to use DPI techniques in order to parse the encapsulated packet, performing instead a simple inspection of the outer header is sufficient.

Transition Mechanism: The existence of a LISP specific EID block may prove useful in transition scenarios. A non-LISP domain would ask for an allocation in the LISP EID block and use it to deploy LISP in its network. Such allocation will not be announced in the BGP routing infrastructure (cf., Section 4). This approach will allow non-LISP domains to avoid fragmenting their already allocated non-LISP addressing space, which may lead to BGP routing table inflation since it may (rightfully) be announced in the BGP routing infrastructure.

Limit the impact on BGP routing infrastructure: As described in the previous scenario, LISP adopters will avoid fragmenting their addressing space, since fragmentation would negatively impact the BGP routing infrastructure. Adopters will use addressing space from the EID block, which might be announced in large aggregates and in a tightly controlled manner only by proxy xTRs.

Is worth mentioning that new use cases can arise in the future, due to new and unforeseen scenarios.

Furthermore, the use of a dedicated address block will give a tighter control, especially filtering, over the traffic in the initial experimental phase, while facilitating its large-scale deployment.

[RFC3692] considers assigning experimental and testing numbers useful, and the request of a reserved IPv6 prefix is a perfect match of such practice. The present document follows the guidelines provided in [RFC3692], with one exception. [RFC3692] suggests the use of values similar to those called "Private Use" in [RFC5226], which by definition are not unique. One of the purposes of the present request to IANA is to guarantee uniqueness to the EID block. The lack thereof would result in a lack of real utility of a reserved IPv6 prefix.

4. Expected use

Sites planning to deploy LISP may request a prefix in the IPv6 EID

block. Such prefixes will be used for routing and endpoint identification inside the site requesting it. Mappings related to such prefix, or part of it, will be made available through the mapping system in use and registered to one or more Map Server(s).

The EID block must be used for LISP experimentation and must not be advertised in the form of more specific route advertisements in the non-LISP inter-domain routing environment. Interworking between the EID block sub-prefixes and the non-LISP Internet is done according to [RFC6832] and [RFC7215].

As the LISP adoption progresses, the EID block may potentially have a reduced impact on the BGP routing infrastructure, compared to the case of having the same number of adopters using global unicast space allocated by RIRs ([MobiArch2007]). From a short-term perspective, the EID block offers potentially large aggregation capabilities since it is announced by PxTRs possibly concentrating several contiguous prefixes. This trend should continue with even lower impact from a long-term perspective, since more aggressive aggregation can be used, potentially leading at using few PxTRs announcing the whole EID block ([FIABook2010]).

The EID block will be used only at configuration level, it is recommended not to hard-code in any way the IPv6 EID block in the router hardware. This allows avoiding locking out sites that may want to switch to LISP while keeping their own IPv6 prefix, which is not in the IPv6 EID block. Furthermore, in the case of a future permanent allocation, the allocated prefix may differ from the experimental temporary prefix allocated during the experimentation phase.

With the exception of PITR case (described in Section 8) prefixes out of the EID block must not be announced in the BGP routing infrastructure.

5. Block Dimension

The working group reached consensus on an initial allocation of a /32 prefix. The reason of such consensus is manifold:

- The working group agreed that /32 prefix is sufficiently large to cover initial allocation and requests for prefixes in the EID space in the next few years for very large-scale experimentation and deployment.

- As a comparison, it is worth mentioning that the current LISP Beta Network ([BETA]) is using a /32 prefix, with more than 250 sites

using a /48 sub prefix. Hence, a /32 prefix appears sufficiently large to allow the current deployment to scale up and be open for interoperation with independent deployments using EIDs in the new /32 prefix.

- o A /32 prefix is sufficiently large to allow deployment of independent (commercial) LISP enabled networks by third parties, but may as well boost LISP experimentation and deployment.

- o The use of a /32 prefix is in line with previous similar prefix allocation for tunneling protocols ([RFC3056]).

6. 3+3 Allocation Plan

This document requests IANA to initially assign a /32 prefix out of the IPv6 addressing space for use as EID in LISP (Locator/ID Separation Protocol).

IANA allocates the requested address space by MMMM/YYYY0 for a duration of 3 (three) initial years (through MMMM/YYYY3), with an option to extend this period by 3 (three) more years (until MMMM/YYYY6). By the end of the first period, the IETF will provide a decision on whether to transform the prefix in a permanent assignment or to put it back in the free pool (see Section 7 for more information).

[RFC Editor: please replace MMMM and all its occurrences in the document with the month of publication as RFC.]

[RFC Editor: please replace YYYY0 and all its occurrences in the document with the year of publication as RFC.]

[RFC Editor: please replace YYYY3 and all its occurrences in the document with the year of publication as RFC plus 3 years, e.g., if published in 2016 then put 2019.]

[RFC Editor: please replace YYYY6 and all its occurrences in the document with the year of publication as RFC plus 6 years, e.g., if published in 2016 then put 2022.]

In the first case, i.e., if the IETF decides to transform the block in a permanent allocation, the EID block allocation period will be extended for three years (until MMMM/YYYY6) so to give time to the IETF to define the final size of the EID block and create a transition plan. The transition of the EID block into a permanent allocation has the potential to pose policy issues (as recognized in [RFC2860], section 4.3) and hence discussion with the IANA, the RIR

communities, and the IETF community will be necessary to determine appropriate policy for permanent EID block allocation and management. Note as well that the final permanent allocation may differ from the initial experimental assignment, hence, it is recommended not to hard-code in any way the experimental EID block on LISP-capable devices.

In the latter case, i.e., if the IETF decides to stop the EID block experimental use, by MMMM/YYYY3 all temporary prefix allocations in such address range must expire and be released, so that the entire /32 is returned to the free pool.

The allocation and management of the EID block for the initial 3 years period (and the optional 3 more years) is detailed in [I-D.ietf-lisp-eid-block-mgmnt].

7. Allocation Lifetime

If no explicit action is carried out by the end of the experiment (by MMMM/YYYY3) it is automatically considered that there was no sufficient interest in having a permanent allocation and the address block will be returned to the free pool.

Otherwise, if the LISP Working Group recognizes that there is value in having a permanent allocation then explicit action is needed.

In order to trigger the process for a permanent allocation a document is required. Such document has to articulate the rationale why a permanent allocation would be beneficial. More specifically, the document has to detail the experience gained during experimentation and all of the technical benefits provided by the use of a LISP specific prefix. Such technical benefits are expected to lay in the scenarios described in Section 3, however, new unforeseen benefits may appear during experimentation. The description should be sufficiently articulate so to allow to provide an estimation of what should be the size of the permanent allocation. Note however that, as explained in Section 6, it is up to IANA to decide which address block will be used as permanent allocation and that such block may be different from the temporary experimental allocation.

8. Routing Considerations

In order to provide connectivity between the Legacy Internet and LISP sites, PITRs announcing large aggregates (ideally one single large aggregate) of the IPv6 EID block could be deployed. By doing so, PITRs will attract traffic destined to LISP sites in order to

encapsulate and forward it toward the specific destination LISP site. Routers in the Legacy Internet must treat announcements of prefixes from the IPv6 EID block as normal announcements, applying best current practice for traffic engineering and security.

Even in a LISP site, not all routers need to run LISP elements. In particular, routers that are not at the border of the local domain, used only for intra-domain routing, do not need to provide any specific LISP functionality but must be able to route traffic using addresses in the IPv6 EID block.

For the above-mentioned reasons, routers that do not run any LISP element, must not include any special handling code or hardware for addresses in the IPv6 EID block. In particular, it is recommended that the default router configuration does not handle such addresses in any special way. Doing differently could prevent communication between the Legacy Internet and LISP sites or even break local intra-domain connectivity.

9. Security Considerations

This document does not introduce new security threats in the LISP architecture nor in the legacy Internet architecture.

10. IANA Considerations

This document instructs the IANA to assign a /32 IPv6 prefix for use as the global LISP EID space using a hierarchical allocation as outlined in [RFC5226] and summarized in Table 1.

This document does not specify any specific value for the requested address block but suggests that should come from the 2000::/3 Global Unicast Space. IANA is not requested to issue an AS0 ROA (Route Origin Attestation [RFC6491]), since the Global EID Space will be used for routing purposes.

Attribute	Value
Address Block	2001:5::/32
Name	EID Space for LISP
RFC	[This Document]
Allocation Date	2015
Termination Date	MMMM/YYYY3 [1]
Source	True [2]
Destination	True
Forwardable	True
Global	True
Reserved-by-protocol	True [3]

[1] According to the 3+3 Plan outlined in this document termination date can be postponed to MMMM/YYYY6. [2] Can be used as a multicast source as well. [3] To be used as EID space by LISP [RFC6830] enabled routers.

Table 1: Global EID Space

[IANA: Please update the Termination Date and footnote [1] in the Special-Purpose Address Registry when the I-D is published as RFC.]

The reserved address space is requested for a period of time of three initial years starting in MMMM/YYYY0 (until MMMM/YYYY3), with an option to extend it by three years (until MMMM/YYYY6) up on decision of the IETF (see Section 6 and Section 7). Following the policies outlined in [RFC5226], upon IETF Review, by MMMM/YYYY3 decision should be made on whether to have a permanent EID block assignment. If no explicit action is taken or if the IETF review outcome will be that is not worth to have a reserved prefix as global EID space, the whole /32 will be taken out from the IPv6 Special Purpose Address Registry and put back in the free pool managed by IANA.

Allocation and management of the Global EID Space is detailed in a different document. Nevertheless, all prefix allocations out of this space must be temporary and no allocation must go beyond MMMM/YYYY3 unless the IETF Review decides for a permanent Global EID Space assignment.

11. Acknowledgments

Special thanks to Roque Gagliano for his suggestions and pointers. Thanks to Alvaro Retana, Deborah Brungard, Ron Bonica, Damien Saucez, David Conrad, Scott Bradner, John Curran, Paul Wilson, Geoff Huston,

Wes George, Arturo Servin, Sander Steffann, Brian Carpenter, Roger Jorgensen, Terry Manderson, Brian Haberman, Adrian Farrel, Job Snijders, Marla Azinger, Chris Morrow, and Peter Schoenmaker, for their insightful comments. Thanks as well to all participants to the fruitful discussions on the IETF mailing list.

The work of Luigi Iannone has been partially supported by the ANR-13-INFR-0009 LISP-Lab Project (www.lisp-lab.org) and the EIT KIC ICT-Labs SOFNETS Project.

12. References

12.1. Normative References

[I-D.ietf-lisp-eid-block-mgmnt]
Iannone, L., Jorgensen, R., Conrad, D., and G. Huston, "LISP EID Block Management Guidelines", draft-ietf-lisp-eid-block-mgmnt-06 (work in progress), August 2015.

[RFC2860] Carpenter, B., Baker, F., and M. Roberts, "Memorandum of Understanding Concerning the Technical Work of the Internet Assigned Numbers Authority", RFC 2860, DOI 10.17487/RFC2860, June 2000, <http://www.rfc-editor.org/info/rfc2860>.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <http://www.rfc-editor.org/info/rfc3692>.

[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>.

[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>.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller,

"Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <http://www.rfc-editor.org/info/rfc6832>.

[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>.

[RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, DOI 10.17487/RFC6834, January 2013, <http://www.rfc-editor.org/info/rfc6834>.

[RFC6835] Farinacci, D. and D. Meyer, "The Locator/ID Separation Protocol Internet Groper (LIG)", RFC 6835, DOI 10.17487/RFC6835, January 2013, <http://www.rfc-editor.org/info/rfc6835>.

[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>.

[RFC6837] Lear, E., "NERD: A Not-so-novel Endpoint ID (EID) to Routing Locator (RLOC) Database", RFC 6837, DOI 10.17487/RFC6837, January 2013, <http://www.rfc-editor.org/info/rfc6837>.

12.2. Informative References

[BETA] LISP Beta Network, "http://www.lisp4.net".

[FIABook2010] L. Iannone, T. Leva, "Modeling the economics of Loc/ID Separation for the Future Internet.", Towards the Future Internet - Emerging Trends from the European Research, Pages 11-20, ISBN: 9781607505389, IOS Press , May 2010.

[I-D.ietf-lisp-introduction] Cabellos-Aparicio, A. and D. Saucez, "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-introduction-13 (work in progress), April 2015.

[MobiArch2007] B. Quoitin, L. Iannone, C. de Launois, O. Bonaventure,

"Evaluating the Benefits of the Locator/Identifier Separation", The 2nd ACM-SIGCOMM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch'07) , August 2007.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <http://www.rfc-editor.org/info/rfc3056>.

[RFC6491] Manderson, T., Vegoda, L., and S. Kent, "Resource Public Key Infrastructure (RPKI) Objects Issued by IANA", RFC 6491, DOI 10.17487/RFC6491, February 2012, <http://www.rfc-editor.org/info/rfc6491>.

[RFC7215] Jakab, L., Cabellos-Aparicio, A., Coras, F., Domingo-Pascual, J., and D. Lewis, "Locator/Identifier Separation Protocol (LISP) Network Element Deployment Considerations", RFC 7215, DOI 10.17487/RFC7215, April 2014, <http://www.rfc-editor.org/info/rfc7215>.

Appendix A. Document Change Log

[RFC Editor: Please remove this section on publication as RFC]

Version 13 Posted MMMM 2016.

- Changed I-D type from "Informational" to "Experimental" as requested by A. Retana during IESG review.
- Dropped the appendix "LISP Terminology"; replaced by pointer to the LISP Introduction document.
- Added Section 7 to clarify the process after the 3 years experimental allocation.
- Modified the dates, introducing variables, so to allow RFC Editor to easily update dates by publication as RFC.

Version 12 Posted May 2015.

- Fixed typos and references as suggested by the Gen-ART and OPS-DIR review.

Version 11 Posted April 2015.

- In Section 4, deleted contradictory text on EID prefix advertisement in non-LISP inter-domain routing environments.

- In Section 3 deleted the "Avoid excessive strech" bullet, because confusing.
- Deleted last bullet of the list in Section 3 because retundant w.r.t. global content of the document.

Version 10 Posted January 2015.

- Keep alive version

Version 09 Posted July 2014.

- Few Editorial modifications as requested by D. Saucez, as shepherd, during the write up of the document.
- Allocation date postponed to beginning 2015, as suggested by D. Saucez.

Version 08 Posted January 2014.

- Modified Section 4 as suggested by G. Houston.

Version 07 Posted November 2013.

- Modified the document so to request a /32 allocation, as for the consensus reached during IETF 88th.

Version 06 Posted October 2013.

- Clarified the rationale and intent of the EID block request with respect to [RFC3692], as suggested by S. Bradner and J. Curran.
- Extended Section 3 by adding the transion scenario (as suggested by J. Curran) and the TE scenario. The other scenarios have been also edited.
- Section 6 has been re-written to introduce the 3+3 allocation plan as suggested by B. Haberman and discussed during 86th IETF.
- Section 10 has also been updated to the 3+3 years allocation plan.
- Moved Section 11 at the end of the document.
- Changed the original Definition of terms to an appendix.

Version 05 Posted September 2013.

- No changes.

Version 04 Posted February 2013.

- Added Table 1 as requested by IANA.

- Transformed the prefix request in a temporary request as suggested by various comments during IETF Last Call.

- Added discussion about short/long term impact on BGP in Section 4 as requested by B. Carpenter.

Version 03 Posted November 2012.

- General review of Section 5 as requested by T. Manderson and B. Haberman.

- Dropped RFC 2119 Notation, as requested by A. Farrel and B. Haberman.

- Changed "IETF Consensus" to "IETF Review" as pointed out by Roque Gagliano.

- Changed every occurrence of "Map-Server" and "Map-Resolver" with "Map Server" and "Map Resolver" to make the document consistent with [RFC6833]. Thanks to Job Snijders for pointing out the issue.

Version 02 Posted April 2012.

- Fixed typos, nits, references.

- Deleted reference to IANA allocation policies.

Version 01 Posted October 2011.

- Added Section 5.

Version 00 Posted July 2011.

- Updated section "IANA Considerations"

- Added section "Rationale and Intent" explaining why the EID block allocation is useful.

- Added section "Expected Use" explaining how sites can request and use a prefix in the IPv6 EID Block.

- Added section "Action Plan" suggesting IANA to avoid allocating address space adjacent the allocated EID block in order to accommodate future EID space requests.
- Added section "Routing Consideration" describing how routers not running LISP deal with the requested address block.
- Added the present section to keep track of changes.
- Rename of draft-meyer-lisp-eid-block-02.txt.

Authors' Addresses

Luigi Iannone
Telecom ParisTech

Email: ggx@gigix.net

Darrel Lewis
Cisco Systems, Inc.

Email: darlewis@cisco.com

David Meyer
Brocade

Email: dmm@1-4-5.net

Vince Fuller

Email: vaf@vaf.net

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: March 17, 2014

G. Schudel
cisco Systems
A. Jain
Juniper Networks
V. Moreno
cisco Systems

September 13, 2013

LISP MIB
draft-ietf-lisp-mib-13

Abstract

This document defines the MIB module that contains managed objects to support the monitoring devices that support the Locator/ID Separation Protocol (LISP). These objects provide information useful for monitoring LISP devices, including determining basic LISP configuration information, LISP functional status, and operational counters and other statistics.

Status of This Memo

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

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

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

This Internet-Draft will expire on March 17, 2014.

Copyright Notice

Table of Contents

1. Introduction

This document describes the Management Information Base (MIB) module for use with network management protocols in the Internet community. Specifically, the MIB for managing devices that support the Locator/ID Separation Protocol (LISP) is described.

LISP [RFC6830] specifies a network-based architecture and mechanisms that implement a new semantic for IP addressing using two separate name spaces: Endpoint Identifiers (EIDs), used within sites, and Routing Locators (RLOCs), used on the transit networks that make up the Internet infrastructure. To achieve this separation, LISP defines protocol mechanisms for mapping from EIDs to RLOCs.

From a data plane perspective, LISP traffic is handled exclusively at the network layer by devices performing Ingress Tunnel Router (ITR) and Egress Tunnel Router (ETR) LISP functions. Data plane operations performed by these devices are described in [RFC6830]. Additionally, data plane interworking between legacy (Internet) and LISP sites is implemented by devices performing Proxy ITR (PITR) and Proxy ETR (PETR) functions. The data plane operations of these devices is described in [RFC6832].

From a control plane perspective, LISP employs mechanisms related to creating, maintaining, and resolving mappings from EIDs to RLOCs. LISP ITRs, ETRs, PITRs, and PETRs perform specific control plane functions, and these control plane operations are described in [RFC6830]. Additionally, LISP infrastructure devices supporting LISP control plane functionality include Map-Servers and Map-Resolvers, and the control plane operations of these devices are described in [RFC6833].

2. Requirements Notation

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

3. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP).

Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

4. Definition of Terms

This document does not define any new terms. All terms used in this document are listed here for completeness; the authoritative definition of each term can be found in the definition section of the respective, specified reference.

Endpoint ID (EID): [RFC6830]

Routing Locator (RLOC): [RFC6830]

EID-to-RLOC Cache: [RFC6830]

EID-to-RLOC Database: [RFC6830]

Ingress Tunnel Router (ITR): [RFC6830]

Egress Tunnel Router (ETR): [RFC6830]

xTR: [RFC6830]

Proxy ITR (PITR): [RFC6832]

Proxy ETR (PETR): [RFC6832]

LISP Site: [RFC6830]

Map-Server: [RFC6833]

Map-Resolver: [RFC6833]

Map-Request: [RFC6833]

Map-Reply: [RFC6833]

Negative Map-Reply: [RFC6833]

5. LISP MIB Objectives

The objectives for this LISP MIB module are to provide a read-only mechanism to support the following functions:

- Provide a means for obtaining (read-only) a current status of LISP features enabled on a device, and (read-only) a current status of configuration attributes related to those features. As one example, this MIB could determine the ON/OFF status of LISP features such as ITR, ETR, PITR, PETR, MS or MR support, specifically as realated to both IPv4 or IPv6 address families. Other examples could include: obtaining the (read-only) status of whether rloc-probing is enabled, whether the use of a PETR is configured, and obtaining the (read-only) values of other related attributes such as the map-cache limit value, or a mapping time-to-live value.

- Provide a means for obtaining (read-only) the current attributes of various LISP tables, such as the EID-to-RLOC policy data contained in the Map-Cache, or the local EID-to-RLOC policy data contained in the Mapping-Database.

- Provide a means for obtaining (read-only) the current operational statistics of various LISP functions, such as the number of packets encapsulated and decapsulated by the device. Other counters of operational interest, depending on LISP function, include things like the current number of map-cache entries, and the total number and rate of map-requests received and sent by the device.

6. Structure of LISP MIB Module

6.1. Overview of Defined Notifications

No LISP MIB notifications are defined.

6.2. Overview of Defined Tables

The LISP MIB module is composed of the following tables of objects:

lispFeatures - This table provides information representing the various lisp features that can be enabled on LISP devices.

lispIidToVrf - This table provides information representing the mapping of a LISP instance ID to a VRF (Virtual Routing/ Forwarding).

lispGlobalStats - This table provides global statistics for a given Instance ID per address-family on a LISP device.

lispMappingDatabase - This table represents the EID-to-RLOC database that contains the EID-prefix to RLOC mappings configured on an ETR. In general, this table would be representative of all such mappings for a given site that this device belongs to.

lispMappingDatabaseLocator - This table represents the set of routing locators contained in the EID-to-RLOC database configured on an ETR.

lispMapCache - This table represents the short-lived, on-demand table maintained on an ITR that stores, tracks, and times-out EID-to-RLOC mappings.

lispMapCacheLocator - This table represents the set of locators per EID prefix contained in the map-cache table of an ITR.

lispConfiguredLocator - This table represents the set of routing locators configured on a LISP device.

lispEidRegistration - This table provides the properties of each EID prefix that is registered with this device when configured to be a Map-Server.

lispEidRegistrationEtr - This table provides the properties of the different ETRs that send registers, for a given EID prefix, to this device when configured to be a Map-Server.

lispEidRegistrationLocator - This table provides the properties of the different locators per EID prefix that is registered with this device when configured to be a Map-Server.

lispUseMapServer - This table provides the properties of all Map-Servers that this device is configured to use.

lispUseMapResolver - This table provides the properties of all Map-Resolvers that this device is configured to use.

lispUseProxyEtr - This table provides the properties of all Proxy ETRs that this device is configured to use.

7. LISP MIB Definitions

```
LISP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    mib-2, Unsigned32, Counter64,
```

```
    Integer32, TimeTicks            FROM SNMPv2-SMI        -- [RFC2578]
    TruthValue, TEXTUAL-CONVENTION,
    TimeStamp                       FROM SNMPv2-TC         -- [RFC2579]
    MODULE-COMPLIANCE, OBJECT-GROUP FROM SNMPv2-CONF       -- [RFC2580]
    MplsL3VpnName
                 FROM MPLS-L3VPN-STD-MIB                   -- [RFC4382]
    AddressFamilyNumbers
                 FROM IANA-ADDRESS-FAMILY-NUMBERS-MIB;     --
         http://www.iana.org/assignments/ianaaddressfamilynumbers-mib

lispMIB MODULE-IDENTITY
    LAST-UPDATED "201309130000Z" -- 13 September 2013
    ORGANIZATION
            "IETF Locator/ID Separation Protocol (LISP) Working Group"
    CONTACT-INFO
            "Email: lisp@ietf.org
            WG charter:
            http://www.ietf.org/html.charters/lisp-charter.html"
    DESCRIPTION
            "This MIB module contains managed objects to support
             monitoring devices that support the Locator/ID Separation
             Protocol (LISP).

            Copyright (C) The IETF Trust (2013)."
    REVISION     "201309130000Z" -- 13 September 2013
    DESCRIPTION  "Initial version of the IETF LISP-MIB module. Published
                  as RFC xxxx."
-- RFC Ed.: RFC-editor pls fill in xxxx
     ::= { mib-2 XXX }
-- RFC Ed.: assigned by IANA, see section 10 for details

--
-- Textual Conventions
--

LispAddressType ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "39a"
    STATUS current
    DESCRIPTION
        "LISP architecture can be applied to a wide variety of
         address-families. This textual-convention is a generalization
         for representing addresses belonging to those address-families.
         For convenience, this document refers to any such address as a
         LISP address. LispAddressType textual-convention consists of
         the following four-tuple:
          1. IANA Address Family Number: A field of length 2-octets,
             whose value is of the form following the assigned
             AddressFamilyNumbers textual-convention described in
```

IANA-ADDRESS-FAMILY-NUMBERS-MIB DEFINITIONS [IANA]
http://www.iana.org/assignments/ianaaddressfamilynumbers-mib.
The enumerations are also listed in [IANA]. Note that this list of address family numbers is maintained by IANA.

2. Length of LISP address: A field of length 1-octet, whose value indicates the octet-length of the next (third) field of this LispAddressType four-tuple.
3. LISP address: A field of variable length as indicated in the previous (second) field, whose value is an address of the IANA Address Family indicated in the first field of this LispAddressType four-tuple. Note that any of the IANA Address Families can be represented. Particularly when the address family is LISP Canonical Address Format (LCAF) [LCAF] http://tools.ietf.org/id/draft-ietf-lisp-lcaf-02.txt with IANA assigned Address Family Number 16387, then the first octet of this field indicates the LCAF type, and the rest of this field is same as the encoding format of the LISP Canonical Address after the length field, as defined in [LCAF].
4. Mask-length of address: A field of length 1-octet, whose value is the mask-length to be applied to the LISP address specified in the previous (third) field.

To illustrate the use of this object, consider the LISP MIB Object below entitled lispMapCacheEntry. This object begins with the following entities:

```
lispMapCacheEntry ::= SEQUENCE {
   lispMapCacheEidLength          INTEGER,
   lispMapCacheEid                LispAddressType,
    ... [skip] ...
```

Example 1: Suppose that the IPv4 EID prefix stored is 192.0.2.0/24. In this case, the values within lispMapCacheEntry would be:

```
   lispMapCacheEidLength  = 8
   lispMapCacheEid = 1, 4, 192.0.2.0, 24
   ... [skip] ...
```

where 8 is the total length in octets of the next object (lispMapCacheEID of type LispAddressType). Then, the value 1 indicates the IPv4 AF (per [IANA]), the value 4 indicates that the AF is 4-octets in length, 192.0.2.0 is the IPv4 address, and the value 24 is the mask-length in bits. Note that the lispMapCacheEidLength value of 8 is used to compute the length of the fourth

(last) field in lispMapCacheEid to be 1 octet - as computed by 8 - (2 + 1 + 4) = 1.

Example 2: Suppose that the IPv6 EID prefix stored is 2001:db8:a::/48. In this case, the values within lispMapCacheEntry would be:

```
lispMapCacheEidLength  = 20
lispMapCacheEid = 2, 16, 2001:db8:a::, 48
... [skip] ...
```

where 20 is the total length in octets of the next object (lispMapCacheEID of type LispAddressType). Then, the value 2 indicates the IPv4 AF (per [IANA]), the value 16 indicates that the AF is 16-octets in length, 2001:db8:a:: is the IPv6 address, and the value 48 is the mask-length in bits. Note that the lispMapCacheEidLength value of 20 is used to compute the length of the fourth (last) field in lispMapCacheEid to be 1 octet - as computed by 20 - (2 + 1 + 16) = 1.

Example 3: As an example where LCAF is used, suppose that the IPv4 EID prefix stored is 192.0.2.0/24 and it is part of LISP Instance ID 101. In this case, the values within lispMapCacheEntry would be:

```
lispMapCacheEidLength  = 11
lispMapCacheEid = 16387, 7, 2, 101, 1, 192.0.2.0, 24
... [skip] ...
```

where 11 is the total length in octets of the next object (lispMapCacheEID of type LispAddressType). Then, the value 16387 indicates the LCAF AF (see [IANA]), the value 7 indicates that the LCAF AF is 7-octets in length in this case, 2 indicates that LCAF Type 2 encoding is used (see [LCAF]), 101 gives the Instance ID, 1 gives the AFI (per [IANA]) for an IPv4 address, 192.0.2.0 is the IPv4 address, and 24 is the mask-length in bits. Note that the lispMapCacheEidLength value of 11 octets is used to compute the length of the last field in lispMapCacheEid to be 1 octet, as computed by 11 - (2 + 1 + 1 + 1 + 1 + 4) = 1.

Note: all LISP header formats and locations of specific flags, bits, and fields are as given in the base LISP references of RFC6830, RFC6832, and RFC6833."

REFERENCE
"RFC6830, Section 14.2, draft-ietf-lisp-lcaf-02.txt."

```
    SYNTAX OCTET STRING (SIZE (5..39))


   --
   -- Top level components of this MIB.
   --
   lispObjects       OBJECT IDENTIFIER ::= { lispMIB 1 }
   lispConformance   OBJECT IDENTIFIER ::= { lispMIB 2 }



    lispFeaturesTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispFeaturesEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the ON/OFF status of the
            various LISP features that can be enabled on LISP devices."
        REFERENCE
            "RFC6830, Section 4.0., Section 5.5., Section 6.3."
        ::= { lispObjects 1 }

    lispFeaturesEntry OBJECT-TYPE
        SYNTAX     LispFeaturesEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the lispFeaturesTable."
        INDEX      { lispFeaturesInstanceID,
                     lispFeaturesAddressFamily }
        ::= { lispFeaturesTable 1 }

    LispFeaturesEntry ::= SEQUENCE {
        lispFeaturesInstanceID                    Unsigned32,
        lispFeaturesAddressFamily                 AddressFamilyNumbers,
        lispFeaturesItrEnabled                    TruthValue,
        lispFeaturesEtrEnabled                    TruthValue,
        lispFeaturesProxyItrEnabled               TruthValue,
        lispFeaturesProxyEtrEnabled               TruthValue,
        lispFeaturesMapServerEnabled              TruthValue,
        lispFeaturesMapResolverEnabled            TruthValue,
        lispFeaturesMapCacheSize                  Unsigned32,
        lispFeaturesMapCacheLimit                 Unsigned32,
        lispFeaturesEtrMapCacheTtl                Unsigned32,
        lispFeaturesRlocProbeEnabled              TruthValue,
        lispFeaturesEtrAcceptMapDataEnabled       TruthValue,
        lispFeaturesEtrAcceptMapDataVerifyEnabled  TruthValue,
        lispFeaturesRouterTimeStamp               TimeStamp
```

```
    }

    lispFeaturesInstanceID OBJECT-TYPE
        SYNTAX     Unsigned32 (0..16777215)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This represents the Instance ID of the LISP header.
            An Instance ID in the LISP address encoding helps
            uniquely identify the AFI-based address space to which
            a given EID belongs. It's default value is 0."
         DEFVAL { 0 }
         ::= { lispFeaturesEntry 1 }

    lispFeaturesAddressFamily OBJECT-TYPE
        SYNTAX     AddressFamilyNumbers
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The IANA address family number of destination address
            of packets that this LISP device is enabled to process."
         ::= { lispFeaturesEntry 2 }

    lispFeaturesItrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of ITR role on this device. If
            this object is true, then ITR feature is enabled."
        ::= { lispFeaturesEntry 3 }

    lispFeaturesEtrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of ETR role on this device. If
            this object is true, then ETR feature is enabled."
        ::= { lispFeaturesEntry 4 }

    lispFeaturesProxyItrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Proxy-ITR role on this device.
            If this object is true, then Proxy-ITR feature is enabled."
```

```
        ::= { lispFeaturesEntry 5 }

    lispFeaturesProxyEtrEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Proxy-ETR role on this device.
            If this object is true, then Proxy-ETR feature is enabled."
        ::= { lispFeaturesEntry 6 }

    lispFeaturesMapServerEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Map Server role on this device.
            If this object is true, then Map Server feature is
            enabled."
        ::= { lispFeaturesEntry 7 }

    lispFeaturesMapResolverEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of Map Resolver role on this device.
            If this object is true, then Map Resolver feature is
            enabled."
        ::= { lispFeaturesEntry 8 }

    lispFeaturesMapCacheSize OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Size of EID-to-RLOC map cache on this device."
        ::= { lispFeaturesEntry 9 }

    lispFeaturesMapCacheLimit OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Maximum permissible entries in EID-to-RLOC map cache on
            this device."
        ::= { lispFeaturesEntry 10 }
```

```
    lispFeaturesEtrMapCacheTtl OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The stored Record TTL of the EID-to-RLOC map record in
            the map cache."
        ::= { lispFeaturesEntry 11 }

    lispFeaturesRlocProbeEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of rloc-probing feature on this
            device.  If this object is true, then this feature is
            enabled."
        ::= { lispFeaturesEntry 12 }

    lispFeaturesEtrAcceptMapDataEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of accepting piggybacked mapping
            data received in a map-request on this device. If this
            object is true, then this device accepts piggybacked
            mapping data."
        ::= { lispFeaturesEntry 13 }

    lispFeaturesEtrAcceptMapDataVerifyEnabled OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the status of verifying accepted piggybacked
            mapping data received in a map-request on this device.
            If this object is true, then this device verifies
            accepted piggybacked mapping data."
        ::= { lispFeaturesEntry 14 }

    lispFeaturesRouterTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which LISP feature was
            enabled on this device.
```

```
            If this information was present at the most recent
            re-initialization of the local management subsystem,
            then this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispFeaturesEntry 15 }



      lispIidToVrfTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF LispIidToVrfEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "This table represents the mapping of LISP Instance ID
              to a VRF."
          REFERENCE
              "RFC6830, Section 5.5. and RFC4382, Section 7."
          ::= { lispObjects 2 }

      lispIidToVrfEntry OBJECT-TYPE
          SYNTAX     LispIidToVrfEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "An entry (conceptual row) in the lispIidToVrfTable."
          INDEX      { lispFeaturesInstanceID }
          ::= { lispIidToVrfTable 1 }

      LispIidToVrfEntry ::= SEQUENCE {
          lispIidToVrfName                    MplsL3VpnName
      }

      lispIidToVrfName OBJECT-TYPE
          SYNTAX     MplsL3VpnName
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
              "The identifier for each VPN that is mapped to the
              given LISP Instance ID."
              ::= { lispIidToVrfEntry 1 }


    lispGlobalStatsTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispGlobalStatsEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides global statistics for a given
```

```
            Instance ID per address-family on a LISP device."
        REFERENCE
            "RFC6830, Section 6.1."
        ::= { lispObjects 3 }

    lispGlobalStatsEntry OBJECT-TYPE
        SYNTAX     LispGlobalStatsEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispGlobalStatsTable."
        INDEX      { lispFeaturesInstanceID,
                     lispFeaturesAddressFamily }
        ::= { lispGlobalStatsTable 1 }

    LispGlobalStatsEntry ::= SEQUENCE {
        lispGlobalStatsMapRequestsIn        Counter64,
        lispGlobalStatsMapRequestsOut       Counter64,
        lispGlobalStatsMapRepliesIn         Counter64,
        lispGlobalStatsMapRepliesOut        Counter64,
        lispGlobalStatsMapRegistersIn       Counter64,
        lispGlobalStatsMapRegistersOut      Counter64
    }

    lispGlobalStatsMapRequestsIn OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map requests received by this device for
            any EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 1 }

    lispGlobalStatsMapRequestsOut OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map requests sent by this device for any
            EID prefix of the given address family and Instance ID.
```

```
            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 2 }

    lispGlobalStatsMapRepliesIn OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map replies received by this device for any
            EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 3 }

    lispGlobalStatsMapRepliesOut OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map replies sent by this device for any EID
            prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 4 }

    lispGlobalStatsMapRegistersIn OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map registers received by this device for
            any EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
```

```
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 5 }

    lispGlobalStatsMapRegistersOut OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Total number of map registers sent by this device for any
            EID prefix of the given address family and Instance ID.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispFeaturesRouterTimeStamp."
        ::= { lispGlobalStatsEntry 6 }


    lispMappingDatabaseTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispMappingDatabaseEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the EID-to-RLOC mapping database
             that contains the EID-prefix to RLOC mappings configured
             on an ETR.

            This table represents all such mappings for the given LISP
            site to which this device belongs."
        REFERENCE
            "RFC6830, Section 6.0."
        ::= { lispObjects 4 }

    lispMappingDatabaseEntry OBJECT-TYPE
        SYNTAX     LispMappingDatabaseEntry

        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in lispMappingDatabaseTable."
        INDEX   { lispMappingDatabaseEidLength,
                  lispMappingDatabaseEid }
        ::= { lispMappingDatabaseTable 1 }

    LispMappingDatabaseEntry ::= SEQUENCE {
        lispMappingDatabaseEidLength        Integer32,
```

```
        lispMappingDatabaseEid              LispAddressType,
        lispMappingDatabaseLsb              Unsigned32,
        lispMappingDatabaseEidPartitioned   TruthValue,
        lispMappingDatabaseTimeStamp        TimeStamp,
        lispMappingDatabaseDecapOctets      Counter64,
        lispMappingDatabaseDecapPackets     Counter64,
        lispMappingDatabaseEncapOctets      Counter64,
        lispMappingDatabaseEncapPackets     Counter64
    }

    lispMappingDatabaseEidLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object gives the octet-length of
            lispMappingDatabaseEid."
        ::= { lispMappingDatabaseEntry 1 }

    lispMappingDatabaseEid OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The EID prefix of the mapping database."
        ::= { lispMappingDatabaseEntry 2 }

    lispMappingDatabaseLsb OBJECT-TYPE
        SYNTAX     Unsigned32 (0..4294967295)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The locator status bits for this EID prefix."
        ::= { lispMappingDatabaseEntry 3 }

    lispMappingDatabaseEidPartitioned OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only

        STATUS     current
        DESCRIPTION
            "Indicates if this device is partitioned from the site that
            contains this EID prefix. If this object is true, then it
            means this device is partitioned from the site."
        ::= { lispMappingDatabaseEntry 4 }

    lispMappingDatabaseTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
```

```
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the EID Prefix information
            represented by this mapping database entry was configured
            on this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMappingDatabaseEntry 5 }

    lispMappingDatabaseDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets, after decapsulation, of LISP packets
            that were decapsulated by this device addressed to a host
            within this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 6 }

    lispMappingDatabaseDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were decapsulated by this
            device addressed to a host within this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 7 }

    lispMappingDatabaseEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
```

```
        DESCRIPTION
            "The number of octets, before encapsulation, of LISP packets
            that were encapsulated by this device, whose inner header
            source address matched this EID prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 8 }

    lispMappingDatabaseEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device whose inner header source address matched this EID
            prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of LISP features
            being removed, which can be detected by observing the value
            of lispMappingDatabaseTimeStamp."
        ::= { lispMappingDatabaseEntry 9 }


   lispMappingDatabaseLocatorTable OBJECT-TYPE
       SYNTAX     SEQUENCE OF LispMappingDatabaseLocatorEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "This table represents the set of routing locators per EID
           prefix contained in the EID-to-RLOC database configured on
           this ETR."
       REFERENCE
           "RFC6830, Section 6.2."
       ::= { lispObjects 5 }

   lispMappingDatabaseLocatorEntry OBJECT-TYPE
       SYNTAX     LispMappingDatabaseLocatorEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "An entry (conceptual row) in the
           lispMappingDatabaseLocatorTable."
```

```
        INDEX  { lispMappingDatabaseEidLength,
                 lispMappingDatabaseEid,
                 lispMappingDatabaseLocatorRlocLength,
                 lispMappingDatabaseLocatorRloc }
        ::= { lispMappingDatabaseLocatorTable 1 }

    LispMappingDatabaseLocatorEntry ::= SEQUENCE {
        lispMappingDatabaseLocatorRlocLength        Integer32,
        lispMappingDatabaseLocatorRloc              LispAddressType,
        lispMappingDatabaseLocatorRlocPriority      Integer32,
        lispMappingDatabaseLocatorRlocWeight        Integer32,
        lispMappingDatabaseLocatorRlocMPriority     Integer32,
        lispMappingDatabaseLocatorRlocMWeight       Integer32,
        lispMappingDatabaseLocatorRlocState         INTEGER,
        lispMappingDatabaseLocatorRlocLocal         INTEGER,
        lispMappingDatabaseLocatorRlocTimeStamp     TimeStamp,
        lispMappingDatabaseLocatorRlocDecapOctets   Counter64,
        lispMappingDatabaseLocatorRlocDecapPackets  Counter64,
        lispMappingDatabaseLocatorRlocEncapOctets   Counter64,
        lispMappingDatabaseLocatorRlocEncapPackets  Counter64
    }

    lispMappingDatabaseLocatorRlocLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispMappingDatabaseLocatorRloc."
        ::= { lispMappingDatabaseLocatorEntry 1 }

    lispMappingDatabaseLocatorRloc OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is a locator for the given EID prefix in
            the mapping database."
        ::= { lispMappingDatabaseLocatorEntry 2 }

    lispMappingDatabaseLocatorRlocPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the RLOC."
        ::= { lispMappingDatabaseLocatorEntry 3 }
```

```
   lispMappingDatabaseLocatorRlocWeight OBJECT-TYPE
       SYNTAX     Integer32 (0..100)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The unicast weight of the RLOC."
       ::= { lispMappingDatabaseLocatorEntry 4 }

   lispMappingDatabaseLocatorRlocMPriority OBJECT-TYPE
       SYNTAX     Integer32 (0..255)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast priority of the RLOC."
       ::= { lispMappingDatabaseLocatorEntry 5 }

   lispMappingDatabaseLocatorRlocMWeight OBJECT-TYPE
       SYNTAX     Integer32 (0..100)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast weight of the RLOC."
       ::= { lispMappingDatabaseLocatorEntry 6 }

   lispMappingDatabaseLocatorRlocState OBJECT-TYPE
       SYNTAX     INTEGER {
                     up (1),
                     down (2),
                     unreachable (3)
                  }
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The state of this RLOC as per this device.
           (1 = RLOC is up; 2 = RLOC is down; 3 = RLOC is unreachable)."
       ::= { lispMappingDatabaseLocatorEntry 7 }

   lispMappingDatabaseLocatorRlocLocal OBJECT-TYPE
       SYNTAX     INTEGER {
                     siteself (1),
                     sitelocal (2)
                  }
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "Indicates whether the RLOC is local to this device
           (or remote, meaning local to another device in the same LISP
           site). (1 = RLOC is an address on this device; 2 = RLOC is
```

```
            an address on another device)."
        ::= { lispMappingDatabaseLocatorEntry 8 }

    lispMappingDatabaseLocatorRlocTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the RLOC of the EID Prefix
            represented by this mapping database entry was configured
            on this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMappingDatabaseLocatorEntry 9 }

    lispMappingDatabaseLocatorRlocDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were
            addressed to this RLOC of the EID-prefix and
            were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 10 }

    lispMappingDatabaseLocatorRlocDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were addressed to this RLOC
            of the EID-prefix and were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
```

```
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 11 }

    lispMappingDatabaseLocatorRlocEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were encapsulated
            by this device using this RLOC address as the source, and
            that were sourced by an address of this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 12 }

    lispMappingDatabaseLocatorRlocEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device using this RLOC address as the source, and that were
            sourced by an address of this EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of database
            mappings getting re-configured or RLOC status changes, which
            can be detected by observing the value of
            lispMappingDatabaseLocatorRlocTimeStamp."
        ::= { lispMappingDatabaseLocatorEntry 13 }


     lispMapCacheTable OBJECT-TYPE
         SYNTAX     SEQUENCE OF LispMapCacheEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
             "This table represents the short-lived, on-demand table on
             an ITR that stores, tracks, and is responsible for
             timing-out and otherwise validating EID-to-RLOC mappings."
         REFERENCE
             "RFC6830, Section 6.0., Section 12.0."
```

```
        ::= { lispObjects 6 }

    lispMapCacheEntry OBJECT-TYPE
        SYNTAX     LispMapCacheEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispMapCacheTable."
        INDEX      { lispMapCacheEidLength,
                     lispMapCacheEid }
        ::= { lispMapCacheTable 1 }

    LispMapCacheEntry ::= SEQUENCE {
        lispMapCacheEidLength           Integer32,
        lispMapCacheEid                 LispAddressType,
        lispMapCacheEidTimeStamp        TimeStamp,
        lispMapCacheEidExpiryTime       TimeTicks,
        lispMapCacheEidState            TruthValue,
        lispMapCacheEidAuthoritative    TruthValue,
        lispMapCacheEidDecapOctets      Counter64,
        lispMapCacheEidDecapPackets     Counter64,
        lispMapCacheEidEncapOctets      Counter64,
        lispMapCacheEidEncapPackets     Counter64
    }

    lispMapCacheEidLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispMapCacheEid."
        ::= { lispMapCacheEntry 1 }

    lispMapCacheEid OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The EID prefix in the mapping cache."
        ::= { lispMapCacheEntry 2 }

    lispMapCacheEidTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
```

```
            "The value of sysUpTime at which the EID Prefix information
            represented by this entry was learned by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMapCacheEntry 3 }

    lispMapCacheEidExpiryTime OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The time remaining before the ITR times-out this
            EID prefix."
        ::= { lispMapCacheEntry 4 }

    lispMapCacheEidState OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "This object is used to indicate the activity of this EID
            prefix. If this object is true, then it means this EID
            prefix is seeing activity."
        ::= { lispMapCacheEntry 5 }

    lispMapCacheEidAuthoritative OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "This object is used to indicate whether the EID prefix was
            installed by an authoritative map-reply. If this object is
            true, then it means this EID prefix was installed by an
            authoritative map-reply."
        ::= { lispMapCacheEntry 6 }

    lispMapCacheEidDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were decapsulated
            by this device and were sourced from a remote host within
            this EID-prefix.
```

```
            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 7 }

    lispMapCacheEidDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were decapsulated by this
            device and were sourced from a remote host within this
            EID-prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 8 }

    lispMapCacheEidEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were encapsulated
            by this device using the given EID-prefix in the map cache.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 9 }

    lispMapCacheEidEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device using the given EID-prefix in the map cache.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
```

```
            Discontinuities can also occur as a result of cache being
            removed and replaced, which can be detected by observing the
            value of lispMapCacheEidTimeStamp."
        ::= { lispMapCacheEntry 10 }


    lispMapCacheLocatorTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispMapCacheLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the set of locators per EID prefix
            contained in the map-cache table of an ITR."
        REFERENCE
            "RFC6830, Section 6.3."
        ::= { lispObjects 7 }

    lispMapCacheLocatorEntry OBJECT-TYPE
        SYNTAX     LispMapCacheLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispMapCacheLocatorTable."
        INDEX      { lispMapCacheEidLength,
                     lispMapCacheEid,
                     lispMapCacheLocatorRlocLength,
                     lispMapCacheLocatorRloc }
        ::= { lispMapCacheLocatorTable 1 }

    LispMapCacheLocatorEntry ::= SEQUENCE {
        lispMapCacheLocatorRlocLength              Integer32,
        lispMapCacheLocatorRloc                    LispAddressType,
        lispMapCacheLocatorRlocPriority            Integer32,
        lispMapCacheLocatorRlocWeight              Integer32,
        lispMapCacheLocatorRlocMPriority           Integer32,
        lispMapCacheLocatorRlocMWeight             Integer32,
        lispMapCacheLocatorRlocState               INTEGER,
        lispMapCacheLocatorRlocTimeStamp           TimeStamp,
        lispMapCacheLocatorRlocLastPriorityChange  TimeTicks,
        lispMapCacheLocatorRlocLastWeightChange    TimeTicks,
        lispMapCacheLocatorRlocLastMPriorityChange TimeTicks,
        lispMapCacheLocatorRlocLastMWeightChange   TimeTicks,
        lispMapCacheLocatorRlocLastStateChange     TimeTicks,
        lispMapCacheLocatorRlocRtt                 TimeTicks,
        lispMapCacheLocatorRlocDecapOctets         Counter64,
        lispMapCacheLocatorRlocDecapPackets        Counter64,
        lispMapCacheLocatorRlocEncapOctets         Counter64,
```

```
        lispMapCacheLocatorRlocEncapPackets        Counter64
    }

    lispMapCacheLocatorRlocLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispMapCacheLocatorRloc."
        ::= { lispMapCacheLocatorEntry 1 }

    lispMapCacheLocatorRloc OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The locator for the EID prefix in the mapping cache."
        ::= { lispMapCacheLocatorEntry 2 }

    lispMapCacheLocatorRlocPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the RLOC for this EID prefix
            (0-255); lower more preferred. "
        ::= { lispMapCacheLocatorEntry 3 }

    lispMapCacheLocatorRlocWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast weight of the RLOC for this EID prefix
            (0 - 100) percentage. "
        ::= { lispMapCacheLocatorEntry 4 }

    lispMapCacheLocatorRlocMPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast priority of the RLOC for this EID prefix
            (0-255); lower more preferred."
        ::= { lispMapCacheLocatorEntry 5 }

    lispMapCacheLocatorRlocMWeight OBJECT-TYPE
```

```
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast weight of the RLOC for this EID prefix
            (0 - 100) percentage."
        ::= { lispMapCacheLocatorEntry 6 }

    lispMapCacheLocatorRlocState OBJECT-TYPE
        SYNTAX     INTEGER {
                     up (1),
                     down (2),
                     unreachable (3)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The state of this RLOC as per this device
            (1 = RLOC is up; 2 = RLOC is down; 3 = RLOC is unreachable)."
        ::= { lispMapCacheLocatorEntry 7 }

    lispMapCacheLocatorRlocTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the RLOC of EID prefix
            information represented by this entry was learned by
            this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem,
            then this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispMapCacheLocatorEntry 8 }

    lispMapCacheLocatorRlocLastPriorityChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time elapsed since the last change of the unicast priority
            of the RLOC for this EID prefix. Note that this is
            independent of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 9 }

    lispMapCacheLocatorRlocLastWeightChange OBJECT-TYPE
        SYNTAX     TimeTicks
```

```
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time elapsed since the last change of the unicast weight
            of the RLOC for this EID prefix.  Note that this is
            independent of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 10 }

    lispMapCacheLocatorRlocLastMPriorityChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time since the last change of the multicast priority of the
            RLOC for this EID prefix."
        ::= { lispMapCacheLocatorEntry 11 }

    lispMapCacheLocatorRlocLastMWeightChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time since the last change of the multicast weight of the
            RLOC for this EID prefix."
        ::= { lispMapCacheLocatorEntry 12 }

    lispMapCacheLocatorRlocLastStateChange OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Time since the last change of the up/down state of the
            RLOC for this EID prefix."
        ::= { lispMapCacheLocatorEntry 13 }

    lispMapCacheLocatorRlocRtt OBJECT-TYPE
        SYNTAX     TimeTicks
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Round trip time of RLOC probe and map-reply for this RLOC
            address for this prefix."
        ::= { lispMapCacheLocatorEntry 14 }

    lispMapCacheLocatorRlocDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
```

```
        DESCRIPTION
            "The number of octets of LISP packets that were decapsulated
            by this device and were sourced from a remote host within
            this EID-prefix and were encapsulated for this RLOC.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 15 }

    lispMapCacheLocatorRlocDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were decapsulated by this
            device and were sourced from a remote host within this
            EID-prefix and were encapsulated for this RLOC.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 16 }

    lispMapCacheLocatorRlocEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that matched this
            EID prefix and were encapsulated using this RLOC address.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 17 }

    lispMapCacheLocatorRlocEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
```

```
            "The number of LISP packets that matched this EID prefix
            and were encapsulated using this RLOC address.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of RLOC of cache
            being removed and replaced, which can be detected by
            observing the value of lispMapCacheLocatorRlocTimeStamp."
        ::= { lispMapCacheLocatorEntry 18 }


    lispConfiguredLocatorTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispConfiguredLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table represents the set of routing locators
            configured on this device. Note that the Proxy-ITR
            configured addresses are treated as routing locators
            and therefore can be part of this table."
        REFERENCE
            "RFC6830, Section 6.3."
        ::= { lispObjects 8 }

    lispConfiguredLocatorEntry OBJECT-TYPE
        SYNTAX     LispConfiguredLocatorEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispConfiguredLocatorTable."
        INDEX  { lispConfiguredLocatorRlocLength,
                 lispConfiguredLocatorRloc }
        ::= { lispConfiguredLocatorTable 1 }

    LispConfiguredLocatorEntry ::= SEQUENCE {
        lispConfiguredLocatorRlocLength        Integer32,
        lispConfiguredLocatorRloc              LispAddressType,
        lispConfiguredLocatorRlocState         INTEGER,
        lispConfiguredLocatorRlocLocal         INTEGER,
        lispConfiguredLocatorRlocTimeStamp     TimeStamp,
        lispConfiguredLocatorRlocDecapOctets   Counter64,
        lispConfiguredLocatorRlocDecapPackets  Counter64,
        lispConfiguredLocatorRlocEncapOctets   Counter64,
        lispConfiguredLocatorRlocEncapPackets  Counter64
    }

    lispConfiguredLocatorRlocLength OBJECT-TYPE
```

```
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispConfiguredLocatorRloc."
        ::= { lispConfiguredLocatorEntry 1 }

    lispConfiguredLocatorRloc OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is a RLOC address configured on this device.
            It can be an RLOC that is local to this device or can be an
            RLOC which belongs to another ETR within the same site.
            Proxy-ITR address is treated as an RLOC."
        ::= { lispConfiguredLocatorEntry 2 }

    lispConfiguredLocatorRlocState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2),
                      unreachable (3)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The state of this RLOC as per this device. (1 = RLOC is up;
            2 = RLOC is down; 3 = RLOC is unreachable)."
        ::= { lispConfiguredLocatorEntry 3 }

    lispConfiguredLocatorRlocLocal OBJECT-TYPE
        SYNTAX     INTEGER {
                      siteself (1),
                      sitelocal (2)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates whether the RLOC is local to this device (or
            remote, meaning local to another device in the same LISP
            site). (1 = RLOC is an address on this device; 2 = RLOC is
            an address on another device)."
        ::= { lispConfiguredLocatorEntry 4 }

    lispConfiguredLocatorRlocTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
```

```
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the RLOC was configured on
            this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispConfiguredLocatorEntry 5 }

    lispConfiguredLocatorRlocDecapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were addressed to
            this RLOC and were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 6 }

    lispConfiguredLocatorRlocDecapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were addressed to this RLOC
            and were decapsulated.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 7 }

    lispConfiguredLocatorRlocEncapOctets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of octets of LISP packets that were encapsulated
```

```
            by this device using this RLOC address as the source.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 8 }

    lispConfiguredLocatorRlocEncapPackets OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The number of LISP packets that were encapsulated by this
            device using this RLOC address as the source.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of configured
            RLOC being removed and replaced, which can be detected by
            observing the value of lispConfiguredLocatorRlocTimeStamp."
        ::= { lispConfiguredLocatorEntry 9 }



    lispEidRegistrationTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispEidRegistrationEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of each LISP EID prefix
            that is registered with this device when configured to be
            a Map-Server."
        REFERENCE
            "RFC6833, Section 4.0."
        ::= { lispObjects 9 }

    lispEidRegistrationEntry OBJECT-TYPE
        SYNTAX     LispEidRegistrationEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the lispEidRegistrationTable."
        INDEX      { lispEidRegistrationEidLength,
                     lispEidRegistrationEid }
        ::= { lispEidRegistrationTable 1 }
```

```
    LispEidRegistrationEntry ::= SEQUENCE {
        lispEidRegistrationEidLength                Integer32,
        lispEidRegistrationEid                      LispAddressType,
        lispEidRegistrationSiteName                 OCTET STRING,
        lispEidRegistrationSiteDescription          OCTET STRING,
        lispEidRegistrationIsRegistered             TruthValue,
        lispEidRegistrationFirstTimeStamp           TimeStamp,
        lispEidRegistrationLastTimeStamp            TimeStamp,
        lispEidRegistrationLastRegisterSenderLength Integer32,
        lispEidRegistrationLastRegisterSender       LispAddressType,
        lispEidRegistrationAuthenticationErrors     Counter64,
        lispEidRegistrationRlocsMismatch            Counter64
    }

    lispEidRegistrationEidLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispEidRegistrationEid."
        ::= { lispEidRegistrationEntry 1 }

    lispEidRegistrationEid OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The EID prefix that is being registered."
         ::= { lispEidRegistrationEntry 2 }

    lispEidRegistrationSiteName OBJECT-TYPE
        SYNTAX     OCTET STRING (SIZE(0..63))
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Site name used by a Map-Server to distinguish different
            LISP sites that are registering with it."
        ::= { lispEidRegistrationEntry 3 }

    lispEidRegistrationSiteDescription OBJECT-TYPE
        SYNTAX     OCTET STRING (SIZE(0..255))
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Description for a site name used by a Map-Server. The EID
            prefix that is being registered belongs to this site."
        ::= { lispEidRegistrationEntry 4 }
```

```
    lispEidRegistrationIsRegistered OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates the registration status of the given EID prefix.
            If this object is true, then it means the EID prefix is
            registered.

            The value false implies the EID prefix is not registered
            with the Map Server. There are multiple scenarios when this
            could happen like authentication failures, routing problems,
            misconfigs to name a few."
        ::= { lispEidRegistrationEntry 5 }

    lispEidRegistrationFirstTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the first valid register
            message for the EID Prefix information represented by this
            entry was received by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispEidRegistrationEntry 6 }

    lispEidRegistrationLastTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the last valid register
            message for the EID Prefix information represented by this
            entry was received by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem, then
            this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispEidRegistrationEntry 7 }

    lispEidRegistrationLastRegisterSenderLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS read-only
```

```
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispEidRegistrationLastRegisterSender, the next
            object."
        ::= { lispEidRegistrationEntry 8 }

    lispEidRegistrationLastRegisterSender OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Source address of the last valid register message for the
            given EID prefix that was received by this device."
        ::= { lispEidRegistrationEntry 9 }

    lispEidRegistrationAuthenticationErrors OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Count of total authentication errors of map-registers
            received for the given EID prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of site config
            changes, which can be detected by observing the value of
            lispEidRegistrationFirstTimeStamp."
        ::= { lispEidRegistrationEntry 10 }

    lispEidRegistrationRlocsMismatch OBJECT-TYPE
        SYNTAX     Counter64
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Count of total map-registers received that had at least one
            RLOC that was not in the allowed list of RLOCs for the given
            EID prefix.

            Discontinuities in this monotonically increasing value occur
            at re-initialization of the management system.
            Discontinuities can also occur as a result of site config
            changes, which can be detected by observing the value of
            lispEidRegistrationFirstTimeStamp."
        ::= { lispEidRegistrationEntry 11 }
```

```
    lispEidRegistrationEtrTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispEidRegistrationEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of ETRs that register
            the given EID prefix with this device when configured to
            be a Map-Server."
        REFERENCE
            "RFC6830, Section 6.1."
        ::= { lispObjects 10 }

    lispEidRegistrationEtrEntry OBJECT-TYPE
        SYNTAX     LispEidRegistrationEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispEidRegistrationEtrTable."
        INDEX      { lispEidRegistrationEidLength,
                     lispEidRegistrationEid,
                     lispEidRegistrationEtrSenderLength,
                     lispEidRegistrationEtrSender }
        ::= { lispEidRegistrationEtrTable 1 }

    LispEidRegistrationEtrEntry ::= SEQUENCE {
        lispEidRegistrationEtrSenderLength          Integer32,
        lispEidRegistrationEtrSender                LispAddressType,
        lispEidRegistrationEtrLastTimeStamp         TimeStamp,
        lispEidRegistrationEtrTtl                   Unsigned32,
        lispEidRegistrationEtrProxyReply            TruthValue,
        lispEidRegistrationEtrWantsMapNotify        TruthValue
    }

    lispEidRegistrationEtrSenderLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispEidRegistrationEtrSender."
        ::= { lispEidRegistrationEtrEntry 1 }

    lispEidRegistrationEtrSender OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
```

```
            "Source address of the ETR that is sending valid register
            messages for this EID prefix to this device."
        ::= { lispEidRegistrationEtrEntry 2 }

    lispEidRegistrationEtrLastTimeStamp OBJECT-TYPE
        SYNTAX     TimeStamp
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The value of sysUpTime at which the last valid register
            message from this ETR for the EID Prefix information
            represented by this entry was received by this device.

            If this information was present at the most recent
            re-initialization of the local management subsystem,
            then this object contains a zero value."
        DEFVAL { 0 }
        ::= { lispEidRegistrationEtrEntry 3 }

    lispEidRegistrationEtrTtl OBJECT-TYPE
        SYNTAX     Unsigned32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The Record TTL of the registering ETR device for this
            EID prefix."
        ::= { lispEidRegistrationEtrEntry 4 }

    lispEidRegistrationEtrProxyReply OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates proxy-replying status of the registering ETR for
            this EID prefix. If this object is true, then it means the
            Map-Server can proxy-reply."
        ::= { lispEidRegistrationEtrEntry 5 }

    lispEidRegistrationEtrWantsMapNotify OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates whether the EID prefix wants Map-Notifications.
            If this object is true, then it means the EID prefix wants
            Map-Notifications."
        ::= { lispEidRegistrationEtrEntry 6 }
```

```
   lispEidRegistrationLocatorTable OBJECT-TYPE
       SYNTAX     SEQUENCE OF LispEidRegistrationLocatorEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "This table provides the properties of all locators per
           LISP site that are served by this device when configured
           to be a Map-Server."
       REFERENCE
           "RFC6830, Section 6.1."
       ::= { lispObjects 11 }

   lispEidRegistrationLocatorEntry OBJECT-TYPE
       SYNTAX     LispEidRegistrationLocatorEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "An entry (conceptual row) in the
           lispEidRegistrationLocatorTable."
       INDEX      { lispEidRegistrationEidLength,
                    lispEidRegistrationEid,
                    lispEidRegistrationEtrSenderLength,
                    lispEidRegistrationEtrSender,
                    lispEidRegistrationLocatorRlocLength,
                    lispEidRegistrationLocatorRloc }
       ::= { lispEidRegistrationLocatorTable 1 }

   LispEidRegistrationLocatorEntry ::= SEQUENCE {
       lispEidRegistrationLocatorRlocLength         Integer32,
       lispEidRegistrationLocatorRloc               LispAddressType,
       lispEidRegistrationLocatorRlocState          INTEGER,
       lispEidRegistrationLocatorIsLocal            TruthValue,
       lispEidRegistrationLocatorPriority           Integer32,
       lispEidRegistrationLocatorWeight             Integer32,
       lispEidRegistrationLocatorMPriority          Integer32,
       lispEidRegistrationLocatorMWeight            Integer32
   }

   lispEidRegistrationLocatorRlocLength OBJECT-TYPE
       SYNTAX     Integer32 (5..39)
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
           "This object is used to get the octet-length of
           lispEidRegistrationLocatorRloc."
       ::= { lispEidRegistrationLocatorEntry 1 }

   lispEidRegistrationLocatorRloc OBJECT-TYPE
```

```
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "The locator of the given EID prefix being registered by the
            given ETR with this device."
        ::= { lispEidRegistrationLocatorEntry 2 }

    lispEidRegistrationLocatorRlocState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The cached state of this RLOC received in map-register from
            the ETR by the device, in the capacity of a Map-Server.
            Value 1 refers to up, value 2 refers to down."
        ::= { lispEidRegistrationLocatorEntry 3 }

    lispEidRegistrationLocatorIsLocal OBJECT-TYPE
        SYNTAX     TruthValue
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "Indicates if the given locator is local to the registering
            ETR. If this object is true,  it means the locator is local."
        ::= { lispEidRegistrationLocatorEntry 4 }

    lispEidRegistrationLocatorPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the RLOC for this EID prefix in the
            register message sent by the given ETR."
        ::= { lispEidRegistrationLocatorEntry 5 }

    lispEidRegistrationLocatorWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast weight of the RLOC for this EID prefix in the
            register message sent by the given ETR."
        ::= { lispEidRegistrationLocatorEntry 6 }
```

```
   lispEidRegistrationLocatorMPriority OBJECT-TYPE
       SYNTAX     Integer32 (0..255)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast priority of the RLOC for this EID prefix in
           the register message sent by the given ETR."
       ::= { lispEidRegistrationLocatorEntry 7 }

   lispEidRegistrationLocatorMWeight OBJECT-TYPE
       SYNTAX     Integer32 (0..100)
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
           "The multicast weight of the RLOC for this EID prefix in the
           register message sent by the given ETR."
       ::= { lispEidRegistrationLocatorEntry 8 }


      lispUseMapServerTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF LispUseMapServerEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "This table provides the properties of the map-server(s)
              with which this device is configured to register."
          REFERENCE
              "RFC6833, Section 4.3."
          ::= { lispObjects 12 }

      lispUseMapServerEntry OBJECT-TYPE
          SYNTAX     LispUseMapServerEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
              "An entry (conceptual row) in the lispUseMapServerTable."
          INDEX      { lispUseMapServerAddressLength,
                       lispUseMapServerAddress }
          ::= { lispUseMapServerTable 1 }

      LispUseMapServerEntry ::= SEQUENCE {
          lispUseMapServerAddressLength Integer32,
          lispUseMapServerAddress       LispAddressType,
          lispUseMapServerState         INTEGER
      }

      lispUseMapServerAddressLength OBJECT-TYPE
          SYNTAX     Integer32 (5..39)
```

```
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispUseMapServerAddress."
        ::= { lispUseMapServerEntry 1 }

    lispUseMapServerAddress OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "Address of Map-Server configured on this device."
        ::= { lispUseMapServerEntry 2 }

    lispUseMapServerState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2),
                      unreachable (3)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "State of this Map-Server configured on this device
            (1 = Map-Server is up; 2 = Map-Server is down)."
        ::= { lispUseMapServerEntry 3 }



    lispUseMapResolverTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispUseMapResolverEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of the map-resolver(s)
            this device is configured to use."
        REFERENCE
            "RFC6833, Section 4.4."
        ::= { lispObjects 13 }

    lispUseMapResolverEntry OBJECT-TYPE
        SYNTAX     LispUseMapResolverEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispUseMapResolverTable."
```

```
        INDEX      { lispUseMapResolverAddressLength,
                     lispUseMapResolverAddress }
        ::= { lispUseMapResolverTable 1 }

    LispUseMapResolverEntry ::= SEQUENCE {
        lispUseMapResolverAddressLength   Integer32,
        lispUseMapResolverAddress         LispAddressType,
        lispUseMapResolverState           INTEGER
    }

    lispUseMapResolverAddressLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispUseMapResolverAddress."
        ::= { lispUseMapResolverEntry 1 }

    lispUseMapResolverAddress OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "Address of map-resolver configured on this device."
        ::= { lispUseMapResolverEntry 2 }

    lispUseMapResolverState OBJECT-TYPE
        SYNTAX     INTEGER {
                      up (1),
                      down (2)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "State of this Map-Resolver configured on this device
            (1 = Map-Resolver is up; 2 = Map-Resolver is down)."
        ::= { lispUseMapResolverEntry 3 }



    lispUseProxyEtrTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF LispUseProxyEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This table provides the properties of all Proxy ETRs that
            this device is configured to use."
```

```
        REFERENCE
            "RFC6830, Section 6.0."
        ::= { lispObjects 14 }

    lispUseProxyEtrEntry OBJECT-TYPE
        SYNTAX     LispUseProxyEtrEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "An entry (conceptual row) in the
            lispUseProxyEtrTable."
        INDEX      { lispUseProxyEtrAddressLength,
                     lispUseProxyEtrAddress }
        ::= { lispUseProxyEtrTable 1 }

    LispUseProxyEtrEntry ::= SEQUENCE {
        lispUseProxyEtrAddressLength        Integer32,
        lispUseProxyEtrAddress              LispAddressType,
        lispUseProxyEtrPriority             Integer32,
        lispUseProxyEtrWeight               Integer32,
        lispUseProxyEtrMPriority            Integer32,
        lispUseProxyEtrMWeight              Integer32,
        lispUseProxyEtrState                INTEGER
    }

    lispUseProxyEtrAddressLength OBJECT-TYPE
        SYNTAX     Integer32 (5..39)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "This object is used to get the octet-length of
            lispUseProxyEtrAddress."
        ::= { lispUseProxyEtrEntry 1 }

    lispUseProxyEtrAddress OBJECT-TYPE
        SYNTAX     LispAddressType
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
            "Address of Proxy ETR configured on this device."
        ::= { lispUseProxyEtrEntry 2 }

    lispUseProxyEtrPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast priority of the PETR locator."
```

```
        ::= { lispUseProxyEtrEntry 3 }

    lispUseProxyEtrWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The unicast weight of the PETR locator."
        ::= { lispUseProxyEtrEntry 4 }

    lispUseProxyEtrMPriority OBJECT-TYPE
        SYNTAX     Integer32 (0..255)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast priority of the PETR locator."
        ::= { lispUseProxyEtrEntry 5 }

    lispUseProxyEtrMWeight OBJECT-TYPE
        SYNTAX     Integer32 (0..100)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "The multicast weight of the PETR locator."
        ::= { lispUseProxyEtrEntry 6 }

    lispUseProxyEtrState OBJECT-TYPE
        SYNTAX     INTEGER {
                      down (0),
                      up (1)
                   }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
            "State of this Proxy ETR configured on this device
            (0 = Proxy ETR is down; 1 = Proxy ETR is up)."
        ::= { lispUseProxyEtrEntry 7 }
```

```
      --
      -- Conformance Information
      --

      lispCompliances OBJECT IDENTIFIER ::= { lispConformance 1 }
      lispGroups      OBJECT IDENTIFIER ::= { lispConformance 2 }


   --
   -- Compliance Statements
   --

   lispMIBComplianceEtr MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "The compliance statement for LISP ETRs. It conveys
               information if device supports ETR feature, and relevant
               state associated with that feature."
       MODULE  -- this module
       MANDATORY-GROUPS { lispMIBEtrGroup }

         GROUP   lispMIBItrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBPetrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBPitrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBMapServerGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBMapResolverGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBEtrExtendedGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBItrExtendedGroup
         DESCRIPTION
             "This group is optional."
```

```
          GROUP   lispMIBMapServerExtendedGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBTuningParametersGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBEncapStatisticsGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBDecapStatisticsGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBDiagnosticsGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBVrfGroup
          DESCRIPTION
              "This group is optional."

        ::= { lispCompliances 1 }

    lispMIBComplianceItr MODULE-COMPLIANCE
        STATUS  current
        DESCRIPTION
                "The compliance statement for LISP ITRs. It conveys
                information if device supports ITR feature, and any
                state associated with that feature."
        MODULE  -- this module
        MANDATORY-GROUPS { lispMIBItrGroup }

          GROUP   lispMIBEtrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBPetrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBPitrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBMapServerGroup
```

```
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapResolverGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEtrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 2 }

   lispMIBCompliancePetr MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "The compliance statement for LISP Proxy-ETRs. It conveys
               information if given device supports Proxy-ETR feature,
               and relevant state associated with that feature."
       MODULE  -- this module
```

```
       MANDATORY-GROUPS { lispMIBPetrGroup }

         GROUP   lispMIBEtrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBItrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBPitrGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBMapServerGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBMapResolverGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBEtrExtendedGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBItrExtendedGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBMapServerExtendedGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBTuningParametersGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBEncapStatisticsGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBDecapStatisticsGroup
         DESCRIPTION
             "This group is optional."

         GROUP   lispMIBDiagnosticsGroup
         DESCRIPTION
```

```
            "This group is optional."

          GROUP   lispMIBVrfGroup
          DESCRIPTION
              "This group is optional."

        ::= { lispCompliances 3 }

    lispMIBCompliancePitr MODULE-COMPLIANCE
        STATUS  current
        DESCRIPTION
                "The compliance statement for LISP Proxy-ITRs. It conveys
                information if device supports Proxy-ITR feature, and
                relevant state associated with that feature."
        MODULE  -- this module
        MANDATORY-GROUPS { lispMIBPitrGroup }

          GROUP   lispMIBEtrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBItrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBPetrGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBMapServerGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBMapResolverGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBEtrExtendedGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBItrExtendedGroup
          DESCRIPTION
              "This group is optional."

          GROUP   lispMIBMapServerExtendedGroup
          DESCRIPTION
              "This group is optional."
```

```
        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 4 }

   lispMIBComplianceMapServer MODULE-COMPLIANCE
       STATUS  current
       DESCRIPTION
               "The compliance statement for LISP Map Servers. It
               conveys information if device supports Map Server
               feature, and relevant state associated with that
               feature."
       MODULE  -- this module
       MANDATORY-GROUPS { lispMIBMapServerGroup }

        GROUP   lispMIBEtrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPetrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPitrGroup
        DESCRIPTION
            "This group is optional."
```

```
        GROUP   lispMIBMapResolverGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEtrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 5 }

   lispMIBComplianceMapResolver MODULE-COMPLIANCE
      STATUS  current
      DESCRIPTION
              "The compliance statement for LISP Map Resolvers. It
              conveys information if device supports Map Server
              feature, and relevant state associated with that
              feature."
      MODULE  -- this module
      MANDATORY-GROUPS { lispMIBMapResolverGroup }
```

```
        GROUP   lispMIBEtrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPetrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBPitrGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEtrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBItrExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBMapServerExtendedGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBTuningParametersGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBEncapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDecapStatisticsGroup
        DESCRIPTION
            "This group is optional."

        GROUP   lispMIBDiagnosticsGroup
        DESCRIPTION
            "This group is optional."
```

```
        GROUP   lispMIBVrfGroup
        DESCRIPTION
            "This group is optional."

      ::= { lispCompliances 6 }


   --
   -- Units of Conformance
   --

   lispMIBEtrGroup OBJECT-GROUP
       OBJECTS { lispFeaturesEtrEnabled,
                 lispMappingDatabaseLsb,
                 lispMappingDatabaseLocatorRlocPriority,
                 lispMappingDatabaseLocatorRlocWeight,
                 lispMappingDatabaseLocatorRlocMPriority,
                 lispMappingDatabaseLocatorRlocMWeight,
                 lispMappingDatabaseLocatorRlocState,
                 lispMappingDatabaseLocatorRlocLocal,
                 lispConfiguredLocatorRlocState,
                 lispConfiguredLocatorRlocLocal,
                 lispUseMapServerState
               }
       STATUS  current
       DESCRIPTION
               "A collection of objects to support reporting of basic
                LISP ETR parameters."
       ::= { lispGroups 1 }

   lispMIBItrGroup OBJECT-GROUP
       OBJECTS { lispFeaturesItrEnabled,
                 lispFeaturesMapCacheSize,
                 lispMappingDatabaseLsb,
                 lispMapCacheLocatorRlocPriority,
                 lispMapCacheLocatorRlocWeight,
                 lispMapCacheLocatorRlocMPriority,
                 lispMapCacheLocatorRlocMWeight,
                 lispMapCacheLocatorRlocState,
                 lispMapCacheEidTimeStamp,
                 lispMapCacheEidExpiryTime,
                 lispUseMapResolverState,
                 lispUseProxyEtrPriority,
                 lispUseProxyEtrWeight,
                 lispUseProxyEtrMPriority,
                 lispUseProxyEtrMWeight,
                 lispUseProxyEtrState
               }
```

```
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP ITR parameters."
        ::= { lispGroups 2 }

    lispMIBPetrGroup OBJECT-GROUP
        OBJECTS { lispFeaturesProxyEtrEnabled
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Proxy-ETR parameters."
        ::= { lispGroups 3 }

    lispMIBPitrGroup OBJECT-GROUP
        OBJECTS { lispFeaturesProxyItrEnabled,
                  lispConfiguredLocatorRlocState,
                  lispConfiguredLocatorRlocLocal
                }

        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Proxy-ITR parameters."
        ::= { lispGroups 4 }

    lispMIBMapServerGroup OBJECT-GROUP
        OBJECTS { lispFeaturesMapServerEnabled,
                  lispEidRegistrationIsRegistered,
                  lispEidRegistrationLocatorRlocState
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Map Server parameters."
        ::= { lispGroups 5 }

    lispMIBMapResolverGroup OBJECT-GROUP
        OBJECTS { lispFeaturesMapResolverEnabled
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of basic
                 LISP Map Resolver parameters."
        ::= { lispGroups 6 }

    lispMIBEtrExtendedGroup OBJECT-GROUP
```

```
        OBJECTS { lispFeaturesRlocProbeEnabled,
                  lispFeaturesEtrAcceptMapDataEnabled,
                  lispFeaturesEtrAcceptMapDataVerifyEnabled,
                  lispMappingDatabaseEidPartitioned
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of
                 LISP features and properties on ETRs."
        ::= { lispGroups 7 }

    lispMIBItrExtendedGroup OBJECT-GROUP
        OBJECTS { lispFeaturesRlocProbeEnabled,
                  lispMapCacheEidState,
                  lispMapCacheEidAuthoritative,
                  lispMapCacheLocatorRlocTimeStamp,
                  lispMapCacheLocatorRlocLastPriorityChange,
                  lispMapCacheLocatorRlocLastWeightChange,
                  lispMapCacheLocatorRlocLastMPriorityChange,
                  lispMapCacheLocatorRlocLastMWeightChange,
                  lispMapCacheLocatorRlocLastStateChange,
                  lispMapCacheLocatorRlocRtt
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects to support reporting of
                 LISP features and properties on ITRs."
        ::= { lispGroups 8 }

    lispMIBMapServerExtendedGroup OBJECT-GROUP
        OBJECTS { lispEidRegistrationSiteName,
                  lispEidRegistrationSiteDescription,
                  lispEidRegistrationIsRegistered,
                  lispEidRegistrationFirstTimeStamp,
                  lispEidRegistrationLastTimeStamp,
                  lispEidRegistrationLastRegisterSenderLength,
                  lispEidRegistrationLastRegisterSender,
                  lispEidRegistrationEtrLastTimeStamp,
                  lispEidRegistrationEtrTtl,
                  lispEidRegistrationEtrProxyReply,
                  lispEidRegistrationEtrWantsMapNotify,
                  lispEidRegistrationLocatorIsLocal,
                  lispEidRegistrationLocatorPriority,
                  lispEidRegistrationLocatorWeight,
                  lispEidRegistrationLocatorMPriority,
                  lispEidRegistrationLocatorMWeight
                }
        STATUS  current
```

```
        DESCRIPTION
                "A collection of objects to support reporting of
                 LISP features and properties on Map Servers
                 related to EID registrations."
        ::= { lispGroups 9 }

    lispMIBTuningParametersGroup OBJECT-GROUP
        OBJECTS { lispFeaturesMapCacheLimit,
                  lispFeaturesEtrMapCacheTtl
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects used to support reporting of
                 parameters used to control LISP behavior and to tune
                 performance."
        ::= { lispGroups 10 }

    lispMIBEncapStatisticsGroup OBJECT-GROUP
        OBJECTS { lispMappingDatabaseTimeStamp,
                  lispMappingDatabaseEncapOctets,
                  lispMappingDatabaseEncapPackets,
                  lispMappingDatabaseLocatorRlocTimeStamp,
                  lispMappingDatabaseLocatorRlocEncapOctets,
                  lispMappingDatabaseLocatorRlocEncapPackets,
                  lispMapCacheEidTimeStamp,
                  lispMapCacheEidEncapOctets,
                  lispMapCacheEidEncapPackets,
                  lispMapCacheLocatorRlocTimeStamp,
                  lispMapCacheLocatorRlocEncapOctets,
                  lispMapCacheLocatorRlocEncapPackets,
                  lispConfiguredLocatorRlocTimeStamp,
                  lispConfiguredLocatorRlocEncapOctets,
                  lispConfiguredLocatorRlocEncapPackets
                }
        STATUS  current
        DESCRIPTION
                "A collection of objects used to support reporting of
                 LISP encapsulation statistics for the device."
        ::= { lispGroups 11 }

    lispMIBDecapStatisticsGroup OBJECT-GROUP
        OBJECTS { lispMappingDatabaseTimeStamp,
                  lispMappingDatabaseDecapOctets,
                  lispMappingDatabaseDecapPackets,
                  lispMappingDatabaseLocatorRlocTimeStamp,
                  lispMappingDatabaseLocatorRlocDecapOctets,
                  lispMappingDatabaseLocatorRlocDecapPackets,
                  lispMapCacheEidTimeStamp,
```

```
                lispMapCacheEidDecapOctets,
                lispMapCacheEidDecapPackets,
                lispMapCacheLocatorRlocTimeStamp,
                lispMapCacheLocatorRlocDecapOctets,
                lispMapCacheLocatorRlocDecapPackets,
                lispConfiguredLocatorRlocTimeStamp,
                lispConfiguredLocatorRlocDecapOctets,
                lispConfiguredLocatorRlocDecapPackets
              }
    STATUS  current
    DESCRIPTION
            "A collection of objects used to support reporting of
             LISP decapsulation statistics for the device."
    ::= { lispGroups 12 }

lispMIBDiagnosticsGroup OBJECT-GROUP
    OBJECTS { lispFeaturesRouterTimeStamp,
              lispGlobalStatsMapRequestsIn,
              lispGlobalStatsMapRequestsOut,
              lispGlobalStatsMapRepliesIn,
              lispGlobalStatsMapRepliesOut,
              lispGlobalStatsMapRegistersIn,
              lispGlobalStatsMapRegistersOut,
              lispEidRegistrationAuthenticationErrors,
              lispEidRegistrationRlocsMismatch
            }
    STATUS  current
    DESCRIPTION
            "A collection of objects used to support reporting of
             additional diagnostics related to the LISP control plane
             state of a LISP device."
    ::= { lispGroups 13 }

lispMIBVrfGroup OBJECT-GROUP
    OBJECTS { lispIidToVrfName
            }
    STATUS  current
    DESCRIPTION
            "A collection of objects used to support reporting of
             VRF-related information on a LISP device."
    ::= { lispGroups 14 }


END
```

8. Relationship to Other MIB Modules

8.1. MIB modules required for IMPORTS

The LISP MIB imports the TEXTUAL-CONVENTION AddressFamilyNumbers from the IANA-ADDRESS-FAMILY-NUMBERS-MIB DEFINITIONS [IANA] http://www.iana.org/assignments/ianaaddressfamilynumbers-mib

The LISP MIB imports mib-2, Unsigned32, Counter64, Integer32, and TimeTicks from SNMPv2-SMI -- [RFC2578].

The LISP MIB imports TruthValue, TEXTUAL-CONVENTION, TimeStamp, and TimeTicks from SNMPv2-TC -- [RFC2579].

The LISP MIB imports MODULE-COMPLIANCE from SNMPv2-TC -- [RFC2580].

The LISP MIB imports MplsL3VpnName from MPLS-L3VPN-STD-MIB -- [RFC4382].

9. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

There are no readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) that are considered sensitive.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to

enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

10. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
lispMIB	{ mib-2 XXX }

This document instructs IANA to allocate a new value in the "SMI Network Management MGMT Codes Internet-standard MIB" subregistry of the "Network Management Parameters" registry, according to the following registration data: Decimal: [TBD by IANA] Name: lispMIB Description: Locator/ID Separation Protocol (LISP) References: [RFC XXXX (this RFC)]

11. References

11.1. Normative References

[IANA] "IANA-ADDRESS-FAMILY-NUMBERS-MIB DEFINITIONS", <http://www.iana.org/assignments/ianaaddressfamilynumbers-mib>.

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

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management

Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.

[RFC4382] Nadeau, T. and H. van der Linde, "MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base", RFC 4382, February 2006.

[RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", RFC 5591, June 2009.

[RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, June 2009.

[RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", RFC 6353, July 2011.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.

[RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.

11.2. Informative References

[LCAF] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format", draft-ietf-lisp-lcaf-02.txt (work in progress), March 2013.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

Appendix A. Acknowledgments

A thank you is owed to Dino Farinacci for his inputs and review comments on the initial versions of this draft. In addition, the

authors would like to gratefully acknowledge several others who have reviewed and commented on this draft. They include: Darrel Lewis, Isidor Kouvelas, Jesper Skriver, Selina Heimlich, Parna Agrawal, Dan Romascanu, and Luigi Iannone. Special thanks are owed to Brian Haberman, the Internet Area AD, for his very detailed review, Miguel Garcia for reviewing this document as part of the General Area Review Team, and Harrie Hazewinkel for the detailed MIB review comments.

Authors' Addresses

Gregg Schudel
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: gschudel@cisco.com

Amit Jain
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA

EMail: atjain@juniper.net

Victor Moreno
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

EMail: vimoreno@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 2, 2012

F. Maino
V. Ermagan
Cisco Systems
A. Cabellos
Technical University of Catalonia
D. Saucez
O. Bonaventure
Universite catholique de Louvain

July 1, 2011

LISP-Security (LISP-SEC)
draft-ietf-lisp-sec-00.txt

Abstract

This memo specifies LISP-SEC, 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.

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 January 2, 2012.

Copyright Notice

Table of Contents

1. Introduction

The Locator/ID Separation Protocol [I-D.ietf-lisp] defines a set of functions for routers to exchange information used to map from non-routable Endpoint Identifiers (EIDs) to routable Routing Locators (RLOCs). If these EID-to-RLOC mappings, carried through Map-Reply messages, are transmitted without integrity protection, an adversary can manipulate them and hijack the communication, impersonate the requested EID or mount Denial of Service or Distributed Denial of Service attacks. Also, if the Map-Reply message is transported unauthenticated, an adversarial LISP entity can overclaim an EID-prefix and maliciously redirect traffic directed to a large number of hosts. A detailed description of "overclaiming" attack is provided in [I-D.saucez-lisp-security].

This memo specifies LISP-SEC, 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, ensuring that the sender of a Map-Reply that provides the location for a given EID-prefix is entitled to do so according to the EID prefix registered in the associated Map Server. Map-Register security, including the right for a LISP entity to register an EID-prefix or to claim presence at an RLOC, is out of the scope of LISP-SEC. Additional security considerations are described in Section 6.

2. Definition of Terms

One-Time Key (OTK): An ephemeral randomly generated key that must be used for a single Map-Request/Map-Reply exchange.

ITR-OTK: The One-Time Key generated at the ITR.

MS-OTK: The One-Time Key generated at the Map-Server.

Encapsulated Control Message (ECM): A LISP control message that is prepended with an additional LISP header. ECM is used by ITRs to send LISP control messages to a Map-Resolver, by Map-Resolvers to forward LISP control messages to a Map-Server, and by Map-Resolvers to forward LISP control messages to an ETR.

Authentication Data (AD): Metadata that is included either in a LISP ECM header or in a Map-Reply message to support confidentiality, integrity protection, and verification of EID-

prefix authorization.

OTK-AD: The portion of ECM Authentication Data that contains a One-Time Key.

EID-AD: The portion of ECM and Map-Reply Authentication Data used for verification of EID-prefix authorization.

PKT-AD: The portion of Map-Reply Authentication Data used to protect the integrity of the Map-Reply message.

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 [I-D.ietf-lisp].

3. LISP-SEC Threat Model

LISP-SEC addresses the control plane threats, described in [I-D.saucez-lisp-security], that target EID-to-RLOC mappings, including manipulations of Map-Request and Map-Reply messages, and malicious xTR EID overclaiming. However LISP-SEC makes two main assumptions that are not part of [I-D.saucez-lisp-security]. First, the LISP Mapping System is expected to deliver Map-Request messages to their intended destinations as identified by the EID. Second, no man-in-the-middle attack can be mounted within the LISP Mapping System. Furthermore, while LISP-SEC enables detection of EID prefix over claiming attacks, it assumes that Map Servers can verify the EID prefix authorization at time of registration.

Accordingly to the threat model described in [I-D.saucez-lisp-security] LISP-SEC assumes that any kind of attack, including MITM attacks, can be mounted in the access network, outside of the boundaries of the LISP mapping system. An on-path attacker, outside of the LISP mapping service system can, for instance, hijack mapping requests and replies, spoofing the identity of a LISP node. Another example of on-path attack, called over claiming attack, can be mounted by a malicious Egress Tunnel Router (ETR), by over claiming the EID-prefixes for which it is authoritative. In this way the ETR can maliciously redirect traffic directed to a large number of hosts.

4. Protocol Operations

The goal of the security mechanisms defined in [I-D.ietf-lisp] is to prevent unauthorized insertion of mapping data, by providing origin authentication and integrity protection for the Map-Registration, and by using the nonce to detect unsolicited Map-Reply sent by off-path attackers.

LISP-SEC builds on top of the security mechanisms defined in [I-D.ietf-lisp] to address the threats described in Section 3 by leveraging the trust relationships existing among the LISP entities participating to the exchange of the Map-Request/Map-Reply messages. Those trust relationships are used to securely distribute a One-Time Key (OTK) that provides origin authentication, integrity and anti-replay protection to mapping data conveyed via the mapping lookup process, and that effectively prevent over claiming attacks. The processing of security parameters during the Map-Request/Map-Reply exchange is as follows:

- The ITR-OTK is generated and stored at the ITR, and securely transported to the Map-Server.

- The Map-Server uses the ITR-OTK to compute an HMAC that protects the integrity of the mapping data provided by the Map-Server to prevent overclaiming attacks. The Map-Server also derives a new OTK (MS-OTK) that is passed to the ETR, by applying a Key Derivation Function (KDF) to the ITR-OTK.

- The ETR uses the MS-OTK to compute an HMAC that protects the integrity of the Map-Reply sent to the ITR.

- Finally, the ITR uses the stored ITR-OTK to verify the integrity of the mapping data provided by both the Map-Server and the ETR, and to verify that no overclaiming attacks were mounted along the path between the Map-Server and the ITR.

Section 5 provides the detailed description of the LISP-SEC control messages and their processing, while the rest of this section describes the flow of protocol operations at each entity involved in the Map-Request/Map-Reply exchange:

- The ITR, upon transmitting a Map-Request message, generates and stores an OTK (ITR-OTK). This key is included into the Encapsulated Control Message (ECM) that contains the Map-Request sent to the Map-Resolver. To provide confidentiality to the ITR-OTK over the path between the ITR and its Map-Resolver, the ITR-OTK SHOULD be encrypted using a preconfigured key shared between the ITR and the Map-Resolver, similar to the key shared between

the ETR and the Map-Server in order to secure ETR registration [I-D.ietf-lisp-ms].

- The Map-Resolver decapsulates the ECM message, decrypts the ITR-OTK, if needed, and forwards through the Mapping System the received Map-Request and the ITR-OTK, as part of a new ECM message. As described in Section 5.5, the LISP Mapping System delivers the ECM to the appropriate Map-Server, as identified by the EID destination address of the Map-Request.

- The Map-Server is configured with the location mappings and policy information for the ETR responsible for the destination EID address. Using this preconfigured information the Map-Server, after the decapsulation of the ECM message, finds the longest match EID-prefix that covers the requested EID in the received Map-Request. The Map-Server adds this EID-prefix, together with an HMAC computed using the ITR-OTK, to a new Encapsulated Control Message that contains the received Map-Request.

- The Map-Server derives a new OTK (MS-OTK) by applying a Key Derivation Function (KDF) to the ITR-OTK. MS-OTK is included in the Encapsulated Control Message sent to the ETR. To provide MS-OTK confidentiality over the path between the Map-Server and the ETR, the MS-OTK should be encrypted using the key shared between the ETR and the Map-Server in order to secure ETR registration [I-D.ietf-lisp-ms].

- If the Map-Server is acting in proxy mode, as specified in [I-D.ietf-lisp], the ETR is not involved in the generation of the Map-Reply. In this case the Map-Server generates the Map-Reply on behalf of the ETR as described below.

- The ETR, upon receiving the Encapsulated Map-Request from the Map-Server, decrypts the MS-OTK, if needed, and originates a Map-Reply that contains the EID-to-RLOC mapping information as specified in [I-D.ietf-lisp].

- The ETR computes an HMAC over the original LISP Map-Reply, keyed with MS-OTK to protect the integrity of the whole Map-Reply. The ETR also copies the EID-prefix authorization data that the Map-Server included in the Encapsulated Map-Request into the Map-Reply message.

- The ITR, upon receiving the Map-Reply, uses the locally stored ITR-OTK to verify the integrity of the EID-prefix authorization data included in the Map-Reply by the Map-Server. The ITR computes the MS-OTK by applying the same KDF used by the Map-Server, and verifies the integrity of the Map-Reply. If the

integrity checks fail, the Map-Reply MUST be discarded. Also, if the EID-prefixes claimed by the ETR in the Map-Reply are not equal or less specific than the EID-prefix authorization data inserted by the Map-Server, the ITR MUST discard the Map-Reply.

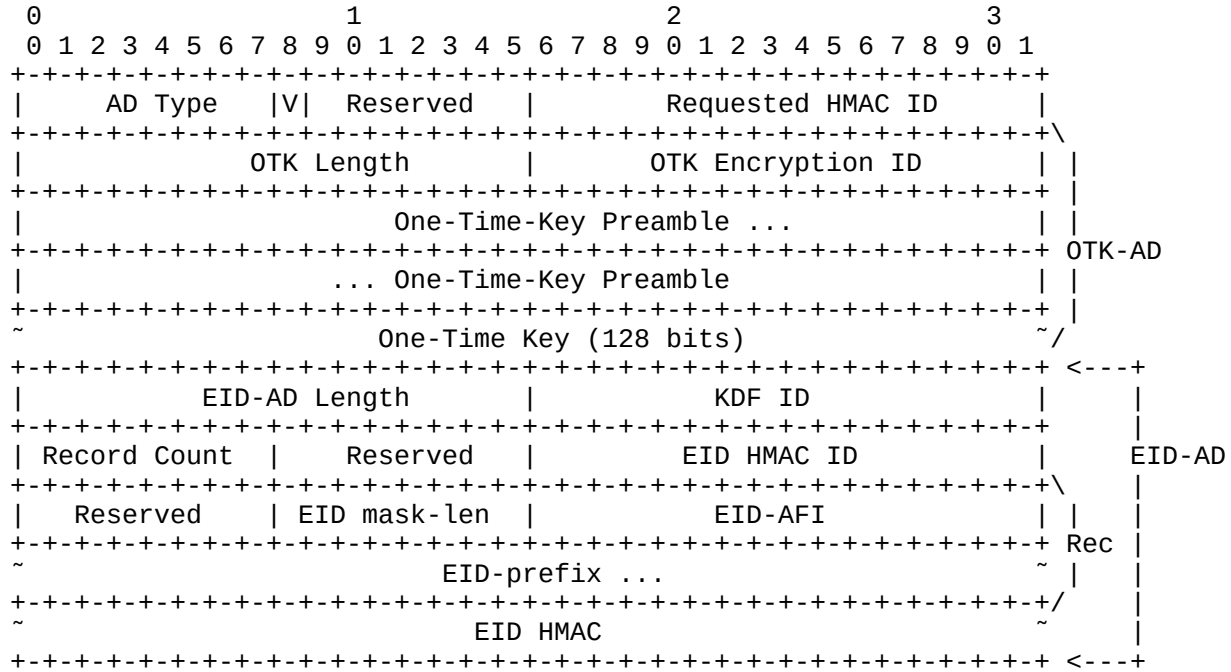
5. LISP-SEC Control Messages Details

LISP-SEC metadata associated with a Map-Request is transported within the Encapsulated Control Message that contains the Map-Request.

LISP-SEC metadata associated with the Map-Reply is transported within the Map-Reply itself.

5.1. Encapsulated Control Message LISP-SEC Extensions

LISP-SEC uses the ECM (Encapsulated Control Message) defined in [I-D.ietf-lisp] with Type set to 8, and S bit set to 1 to indicate that the LISP header includes Authentication Data (AD). The format of the LISP-SEC ECM Authentication Data is defined in the following figure. OTK-AD stands for One-Time Key Authentication Data and EID-AD stands for EID Authentication Data.



LISP-SEC ECM Authentication Data

AD Type: 1 (LISP-SEC Authentication Data)

V: Key Version bit. This bit is toggled when the sender switches to a new OTK wrapping key

Reserved: Set to 0 on transmission and ignored on receipt.

Requested HMAC ID: The HMAC algorithm requested by the ITR. See Section 5.3 for details.

OTK Length: The length (in bytes) of the OTK Authentication Data (OTK-AD), that contains the OTK Preamble and the OTK.

OTK Encryption ID: The identifier of the key wrapping algorithm used to encrypt the One-Time-Key. When a 128-bit OTK is sent unencrypted by the Map-Resolver, the OTK Encryption ID is set to NULL_KEY_WRAP_128. See Section 5.4 for more details.

One-Time-Key Preamble: set to 0 if the OTK is not encrypted. When the OTK is encrypted, this field may carry additional metadata resulting from the key wrapping operation. When a 128-bit OTK is sent unencrypted by Map-Resolver, the OTK Preamble is set to 0x0000000000000000 (64 bits). See Section 5.4 for details.

One-Time-Key: the OTK encrypted (or not) as specified by OTK Encryption ID. See Section 5.4 for details.

EID-AD Length: length (in bytes) of the EID Authentication Data (EID-AD). The ITR MUST set EID-AD Length to 4 bytes, as it only fills the KDF ID field, and all the remaining fields part of the EID-AD are not present. An EID-AD MAY contain multiple EID-records. Each EID-record is 4-byte long plus the length of the AFI-encoded EID-prefix.

KDF ID: Identifier of the Key Derivation Function used to derive the MS-OTK. The ITR SHOULD use this field to indicate the recommended KDF algorithm, according to local policy. The Map-Server can overwrite the KDF ID if it does not support the KDF ID recommended by the ITR. See Section 5.4 for more details.

Record Count: The number of records in this Map-Request message. A record is comprised of the portion of the packet that is labeled ’Rec’ above and occurs the number of times equal to Record Count.

Reserved: Set to 0 on transmission and ignored on receipt.

EID HMAC ID: Identifier of the HMAC algorithm used to protect the integrity of the EID-AD. This field is filled by Map-Server that

computed the EID-prefix HMAC. See Section 5.4 for more details.

EID mask-len: Mask length for EID-prefix.

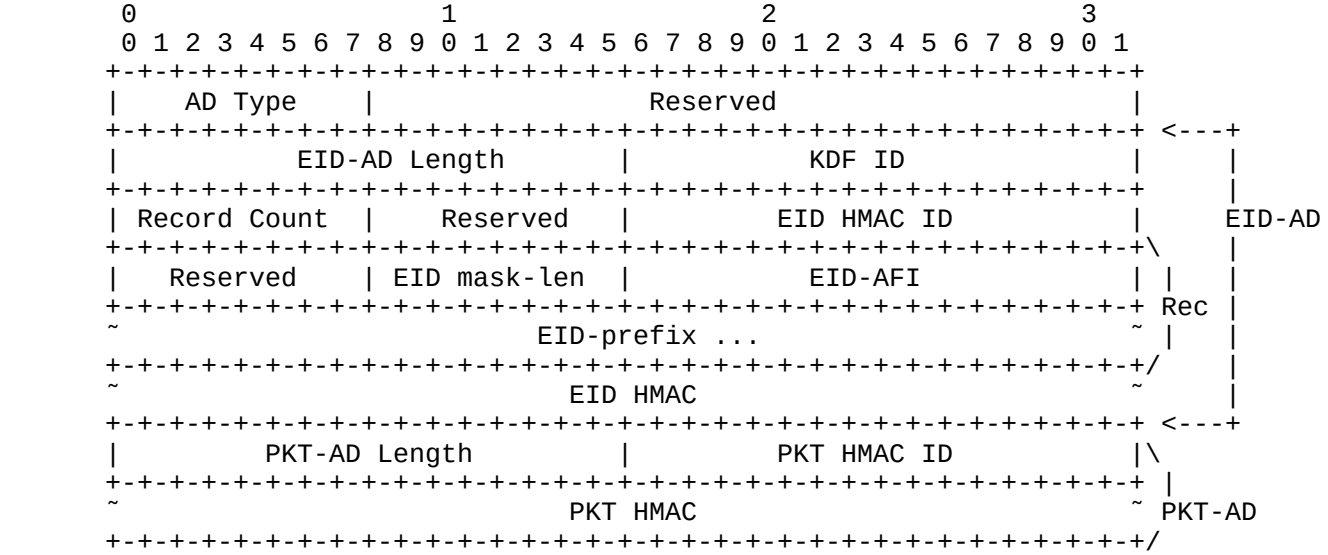
EID-AFI: Address family of EID-prefix according to [RFC5226]

EID-prefix: The Map-Server uses this field to specify the EID-prefix that the destination ETR is authoritative for, and is the longest match for the requested EID.

EID HMAC: HMAC of the EID-AD computed and inserted by Map-Server. Before computing the HMAC operation the EID HMAC field MUST be set to 0. The HMAC covers the entire EID-AD.

5.2. Map-Reply LISP-SEC Extensions

LISP-SEC uses the Map-Reply defined in [I-D.ietf-lisp], with Type set to 2, and S bit set to 1 to indicate that the Map-Reply message includes Authentication Data (AD). The format of the LISP-SEC Map-Reply Authentication Data is defined in the following figure. PKT-AD is the Packet Authentication Data that covers the Map-Reply payload.



LISP-SEC Map-Reply Authentication Data

AD Type: 1 (LISP-SEC Authentication Data)

EID-AD Length: length (in bytes) of the EID-AD. An EID-AD MAY contain multiple EID-records. Each EID-record is 4-byte long plus the length of the AFI-encoded EID-prefix.

KDF ID: Identifier of the Key Derivation Function used to derive MS-OTK. See Section 5.6 for more details.

Record Count: The number of records in this Map-Reply message. A record is comprised of the portion of the packet that is labeled 'Rec' above and occurs the number of times equal to Record Count.

Reserved: Set to 0 on transmission and ignored on receipt.

EID HMAC ID: Identifier of the HMAC algorithm used to protect the integrity of the EID-AD. See Section 5.6 for more details.

EID mask-len: Mask length for EID-prefix.

EID-AFI: Address family of EID-prefix according to [RFC5226].

EID-prefix: This field contains an EID-prefix that the destination ETR is authoritative for, and is the longest match for the requested EID.

EID HMAC: HMAC of the EID-AD, as computed by the Map-Server. Before computing the HMAC operation the EID HMAC field MUST be set to 0. The HMAC covers the entire EID-AD.

PKT-AD Length: length (in bytes) of the Packet Authentication Data (PKT-AD).

PKT HMAC ID: Identifier of the HMAC algorithm used to protect the integrity of the Map-reply Location Data.

PKT HMAC: HMAC of the whole Map-Reply packet, including the LISP-SEC Authentication Data. The scope of the authentication goes from the Map-Reply Type field to the PKT HMAC field included. Before computing the HMAC operation the PKT HMAC field MUST be set to 0. See Section 5.7 for more details.

5.3. ITR Processing

Upon creating a Map-Request, the ITR generates a random ITR-OTK that is stored locally, together with the nonce generated as specified in [I-D.ietf-lisp].

The Map-Request MUST be encapsulated in an ECM, with the S-bit set to 1, to indicate the presence of Authentication Data. If the ITR and the Map-Resolver are configured with a shared key, the ITR-OTK confidentiality SHOULD be protected by wrapping the ITR-OTK with the algorithm specified by the OTK Encryption ID field. See Section 5.4 for further details on OTK encryption.

The Requested HMAC ID field contains the suggested HMAC algorithm to be used by the Map-Server and the ETR to protect the integrity of the ECM Authentication data and of the Map-Reply.

The KDF ID field, specifies the suggested key derivation function to be used by the Map-Server to derive the MS-OTK.

The EID-AD length is set to 4 bytes, since the Authentication Data does not contain EID-prefix Authentication Data, and the EID-AD contains only the KDF ID field.

In response to an encapsulated Map-Request that has the S-bit set, an ITR MUST receive a Map-Reply with the S-bit set, that includes an EID-AD and a PKT-AD. If the Map-Reply does not include both ADs, the ITR MUST discard it. In response to an encapsulated Map-Request with S-bit set to 0, the ITR expects a Map-Reply with S-bit set to 0, and the ITR SHOULD discard the Map-Reply if the S-bit is set.

Upon receiving a Map-Reply, the ITR must verify the integrity of both the EID-AD and the PKT-AD, and MUST discard the Map-Reply if one of the integrity checks fails.

The integrity of the EID-AD is verified using the locally stored ITR-OTK to re-compute the HMAC of the EID-AD using the algorithm specified in the EID HMAC ID field. If the EID HMAC ID field does not match the Requested HMAC ID the ITR SHOULD discard the Map-Reply and send, at the first opportunity it needs to, a new Map-Request with a different Requested HMAC ID field, according to ITR’s local policy. The ITR MUST set the EID HMAC ID field to 0 before computing the HMAC.

To verify the integrity of the PKT-AD, first the MS-OTK is derived from the locally stored ITR-OTK using the algorithm specified in the KDF ID field. This is because the PKT-AD is generated by the ETR using the MS-OTK. If the KDF ID in the Map-Reply does not match the KDF ID requested in the Map-Request, the ITR SHOULD discard the Map-Reply and send, at the first opportunity it needs to, a new Map-Request with a different KDF ID, according to ITR’s local policy. The derived MS-OTK is then used to re-compute the HMAC of the PKT-AD using the Algorithm specified in the PKT HMAC ID field. If the PKT HMAC ID field does not match the Requested HMAC ID the ITR SHOULD discard the Map-Reply and send, at the first opportunity it needs to, a new Map-Request with a different Requested HMAC ID according to ITR’s local policy.

Each individual Map-Reply EID-record is considered valid only if: (1) both EID-AD and PKT-AD are valid, and (2) the intersection of the EID-prefix in the Map-Reply EID-record with one of the EID-prefixes

contained in the EID-AD is not empty. After identifying the Map-Reply record as valid, the ITR sets the EID-prefix in the Map-Reply record to the value of the intersection set computed before, and adds the Map-Reply EID-record to its EID-to-RLOC cache, as described in [I-D.ietf-lisp]. An example of Map-Reply record validation is provided in Section 5.3.1.

The ITR SHOULD send SMR triggered Map Requests over the mapping system in order to receive a secure Map-Reply. If an ITR accepts piggybacked Map-Replies, it SHOULD also send a Map-Request over the mapping system in order to securely verify the piggybacked Map-Reply.

5.3.1. Map-Reply Record Validation

The payload of a Map-Reply may contain multiple EID-records. The whole Map-Reply is signed by the ETR, with the PKT HMAC, to provide integrity protection and origin authentication to the EID-prefix records claimed by the ETR. The Authentication Data field of a Map-Reply may contain multiple EID-records in the EID-AD. The EID-AD is signed by the Map-Server, with the EID HMAC, to provide integrity protection and origin authentication to the EID-prefix records inserted by the Map-Server.

Upon receiving a Map-Reply with the S-bit set, the ITR first checks the validity of both the EID HMAC and of the PKT-AD HMAC. If either one of the HMACs is not valid, a log message is issued and the Map-Reply is not processed any further. If both HMACs are valid, the ITR proceeds with validating each individual EID-record claimed by the ETR by computing the intersection of each one of the EID-prefix contained in the payload of the Map-Reply with each one of the EID-prefixes contained in the EID-AD. An EID-record is valid only if at least one of the intersections is not the empty set.

For instance, the Map-Reply payload contains 3 mapping record EID-prefixes:

1.1.1.0/24

1.1.2.0/24

1.2.0.0/16

The EID-AD contains two EID-prefixes:

1.1.2.0/24

1.2.3.0/24

The EID-record with EID-prefix 1.1.1.0/24 is not processed since it is not included in any of the EID-ADs signed by the Map-Server. A log message is issued.

The EID-record with EID-prefix 1.1.2.0/24 is stored in the map-cache because it matches the second EID-prefix contained in the EID-AD.

The EID-record with EID-prefix 1.2.0.0/16 is not processed since it is not included in any of the EID-ADs signed by the Map-Server. A log message is issued. In this last example the ETR is trying to over claim the EID-prefix 1.2.0.0/16, but the Map-Server authorized only 1.2.3.0/24, hence the EID-record is discarded.

5.3.2. PITR Processing

The processing performed by a PITR is equivalent to the processing of an ITR. However, if the PITR is directly connected to the ALT, the PITR performs the functions of both the ITR and the Map-Resolver forwarding the Map-Request encapsulated in an ECM header that includes the Authentication Data fields as described in Section 5.5.

5.4. Encrypting and Decrypting an OTK

MS-OTK confidentiality is required in the path between the Map-Server and the ETR, the MS-OTK SHOULD be encrypted using the preconfigured key shared between the Map-Server and the ETR for the purpose of securing ETR registration [I-D.ietf-lisp-ms]. Similarly, if ITR-OTK confidentiality is required in the path between the ITR and the Map-Resolver, the ITR-OTK SHOULD be encrypted with a key shared between the ITR and the Map-Resolver.

The OTK is encrypted using the algorithm specified in the OTK Encryption ID field. When the AES Key Wrap algorithm is used to encrypt a 128-bit OTK, according to [RFC3339], the AES Key Wrap Initialization Value MUST be set to 0xA6A6A6A6A6A6A6A6 (64 bits). The output of the AES Key Wrap operation is 192-bit long. The most significant 64-bit are copied in the One-Time Key Preamble field, while the 128 less significant bits are copied in the One-Time Key field of the LISP-SEC Authentication Data.

When decrypting an encrypted OTK the receiver MUST verify that the Initialization Value resulting from the AES Key Wrap decryption operation is equal to 0xA6A6A6A6A6A6A6A6. If this verification fails the receiver MUST discard the entire message.

When a 128-bit OTK is sent unencrypted the OTK Encryption ID is set to NULL_KEY_WRAP_128, and the OTK Preamble is set to 0x0000000000000000 (64 bits).

5.5. Map-Resolver Processing

Upon receiving an encapsulated Map-Request with the S-bit set, the Map-Resolver decapsulates the ECM message. The ITR-OTK, if encrypted, is decrypted as specified in Section 5.4.

The Map-Resolver, as specified in [I-D.ietf-lisp-ms], originates a new ECM header with the S-bit set, that contains the unencrypted ITR-OTK, as specified in Section 5.4, and the other data derived from the ECM Authentication Data of the received encapsulated Map-Request.

The Map-Resolver then forwards the received Map-Request, encapsulated in the new ECM header that includes the newly computed Authentication Data fields.

5.6. Map-Server Processing

Upon receiving an ECM encapsulated Map-Request with the S-bit set, the Map-Server process the Map-Request according to the value of the S-bit contained in the Map-Register sent by the ETR during registration.

If the S-bit contained in the Map-Register was clear the Map-Server decapsulates the ECM and generates a new ECM encapsulated Map-Request that does not contain an ECM Authentication Data, as specified in [I-D.ietf-lisp]. The Map-Server does not perform any further LISP-SEC processing.

If the S-bit contained in the Map-Register was set the Map-Server decapsulates the ECM and generates a new ECM Authentication Data. The Authentication Data includes the OTK-AD and the EID-AD, that contains EID-prefix authorization information, that are ultimately sent to the requesting ITR.

The Map-Server updates the OTK-AD by deriving a new OTK (MS-OTK) from the ITR-OTK received with the Map-Request. MS-OTK is derived applying the key derivation function specified in the KDF ID field. If the algorithm specified in the KDF ID field is not supported, the Map-Server uses a different algorithm to derive the key and updates the KDF ID field accordingly.

The Map-Server and the ETR MUST be configured with a shared key for mapping registration according to [I-D.ietf-lisp-ms]. If MS-OTK confidentiality is required, then the MS-OTK SHOULD be encrypted, by wrapping the MS-OTK with the algorithm specified by the OTK Encryption ID field as specified in Section 5.4.

The Map-Server includes in the EID-AD the longest match registered

EID-prefix for the destination EID, and an HMAC of this EID-prefix. The HMAC is keyed with the ITR-OTK contained in the received ECM Authentication Data, and the HMAC algorithm is chosen according to the Requested HMAC ID field. If The Map-Server does not support this algorithm, the Map-Server uses a different algorithm and specifies it in the EID HMAC ID field. The scope of the HMAC operation covers the entire EID-AD, from the EID-AD Length field to the EID HMAC field, which must be set to 0 before the computation.

The Map-Server then forwards the updated ECM encapsulated Map-Request, that contains the OTK-AD, the EID-AD, and the received Map-Request to an authoritative ETR as specified in [I-D.ietf-lisp].

5.6.1. Map-Server Processing in Proxy mode

If the Map-Server is in proxy mode, it generates a Map-Reply, as specified in [I-D.ietf-lisp], with the S-bit set to 1. The Map-Reply includes the Authentication Data that contains the EID-AD, computed as specified in Section 5.6, as well as the PKT-AD computed as specified in Section 5.7.

5.7. ETR Processing

Upon receiving an encapsulated Map-Request with the S-bit set, the ETR decapsulates the ECM message. The OTK field, if encrypted, is decrypted as specified in Section 5.4 to obtain the unencrypted MS-OTK.

The ETR then generates a Map-Reply as specified in [I-D.ietf-lisp] and includes an Authentication Data that contains the EID-AD, as received in the encapsulated Map-Request, as well as the PKT-AD.

The EID-AD is copied from the Authentication Data of the received encapsulated Map-Request.

The PKT-AD contains the HMAC of the whole Map-Reply packet, keyed with the MS-OTK and computed using the HMAC algorithm specified in the Requested HMAC ID field of the received encapsulated Map-Request. If the ETR does not support the Requested HMAC ID, it uses a different algorithm and updates the PKT HMAC ID field accordingly. The scope of the HMAC operation covers the entire PKT-AD, from the Map-Reply Type field to the PKT HMAC field, which must be set to 0 before the computation.

Finally the ETR sends the Map-Reply to the requesting ITR as specified in [I-D.ietf-lisp].

6. Security Considerations

6.1. Mapping System Security

The LISP-SEC threat model described in Section 3, assumes that the LISP Mapping System is working properly and eventually delivers Map-Request messages to a Map-Server that is authoritative for the requested EID.

Security is not yet embedded in LISP+ALT but BGP route filtering SHOULD be deployed in the ALT infrastructure to enforce proper routing in the mapping system. The SIDR working group is currently addressing prefix and route advertisement authorization and authentication for BGP. While following SIDR recommendations in the global Internet will take time, applying these recommendations to the ALT, which relies on BGP, should be less complex, as ALT is currently small and with a limited number of operators. Ultimately, deploying the SIDR recommendations in ALT further ensures that the fore mentioned assumption is true.

It is also assumed that no man-in-the-middle attack can be carried out against the ALT router to ALT router tunnels, and that the information included into the Map-Requests, in particular the OTK, cannot be read by third-party entities. It should be noted that the integrity of the Map-Request in the ALT is protected by BGP authentication, and that in order to provide OTK confidentiality in the ALT mapping system the ALT router to ALT router tunnels MAY be deployed using IPsec (ESP).

Map-Register security, including the right for a LISP entity to register an EID-prefix or to claim presence at an RLOC, is out of the scope of LISP-SEC.

6.2. Random Number Generation

The ITR-OTK MUST be generated by a properly seeded pseudo-random (or strong random) source. See [RFC4086] for advice on generating security-sensitive random data

6.3. Map-Server and ETR Colocation

If the Map-Server and the ETR are colocated, LISP-SEC does not provide protection from overclaiming attacks mounted by the ETR. However, in this particular case, since the ETR is within the trust boundaries of the Map-Server, ETR’s overclaiming attacks are not included in the threat model.

7. IANA Considerations

7.1. HMAC functions

The following HMAC ID values are defined by this memo for use as Requested HMAC ID, EID HMAC ID, and PKT HMAC ID in the LISP-SEC Authentication Data:

Name	Number	Defined In
NONE	0	
AUTH-HMAC-SHA-1-160	1	[RFC2104]
AUTH-HMAC-SHA-256-128	2	[RFC4634]

values 2-65535 are reserved to IANA.

HMAC Functions

AUTH-HMAC-SHA-1-160 MUST be supported, AUTH-HMAC-SHA-256-128 should be supported.

7.2. Key Wrap Functions

The following OTK Encryption ID values are defined by this memo for use as OTK key wrap algorithms ID in the LISP-SEC Authentication Data:

Name	Number	Defined In
NULL-KEY-WRAP-128	1	
AES-KEY-WRAP-128	2	[RFC3394]

values 0 and 3-65535 are reserved to IANA.

Key Wrap Functions

NULL-KEY-WRAP-128, and AES-KEY-WRAP-128 MUST be supported.

NULL-KEY-WRAP-128 is used to carry an unencrypted 128-bit OTK, with a 64-bit preamble set to 0x0000000000000000 (64 bits).

7.3. Key Derivation Functions

The following KDF ID values are defined by this memo for use as KDF ID in the LISP-SEC Authentication Data:

Name	Number	Defined In
NONE	0	
HKDF-SHA1-128	1	[RFC5869]

values 2-65535 are reserved to IANA.

Key Derivation Functions

HKDF-SHA1-128 MUST be supported

8. Acknowledgements

The authors would like to acknowledge Pere Monclus, Dave Meyer, Dino Farinacci, Brian Weis, David McGrew, Darrel Lewis and Landon Curt Noll for their valuable suggestions provided during the preparation of this document.

9. Normative References

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-14 (work in progress), June 2011.

[I-D.ietf-lisp-interworking]
Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-01 (work in progress), August 2010.

[I-D.ietf-lisp-ms]
Fuller, V. and D. Farinacci, "LISP Map Server", draft-ietf-lisp-ms-09 (work in progress), June 2011.

[I-D.saucez-lisp-security]
Saucez, D., Iannone, L., and O. Bonaventure, "LISP Security Threats", draft-saucez-lisp-security-03 (work in progress), March 2011.

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

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

[RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, September 2002.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, May 2010.

Authors' Addresses

Fabio Maino
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: fmaino@cisco.com

Vina Ermagan
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: vermagan@cisco.com

Albert Cabellos
Technical University of Catalonia
c/ Jordi Girona s/n
Barcelona, 08034
Spain

Email: acabello@ac.upc.edu

Damien Saucez
Universite catholique de Louvain
Place St. Barbe 2
Louvain-la-Neuve,
Belgium

Email: damien.saucez@uclouvain.be

Olivier Bonaventure
Universite catholique de Louvain
Place St. Barbe 2
Louvain-la-Neuve,
Belgium

Email: olivier.bonaventure@uclouvain.be

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 1, 2016

D. Saucez
INRIA
L. Iannone
Telecom ParisTech
O. Bonaventure
Universite catholique de Louvain
January 29, 2016

LISP Threats Analysis
draft-ietf-lisp-threats-15.txt

Abstract

This document provides a threat analysis of the Locator/Identifier Separation Protocol (LISP).

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 August 1, 2016.

Copyright Notice

Table of Contents

1. Introduction

The Locator/ID Separation Protocol (LISP) is specified in [RFC6830]. This document provides an assessment of the potential security threats for the current LISP specifications if LISP is deployed in the Internet (i.e., a public non-trustable environment).

The document is composed of three main parts: the first defines a general threat model that attackers use to mount attacks. The second part, using this threat model, describes the techniques based on the LISP protocol and LISP architecture that attackers may use to construct attacks. The third part discusses mitigation techniques and general solutions to protect the LISP protocol and architecture from attacks.

This document does not consider all the possible uses of LISP as discussed in [RFC6830] and [RFC7215] and does not cover threats due to specific implementations. The document focuses on LISP unicast, including as well LISP Interworking [RFC6832], LISP Map-Server [RFC6833]), and LISP Map-Versioning [RFC6834]. Additional threats may be discovered in the future while deployment continues. The reader is assumed to be familiar with these documents for understanding the present document.

This document assumes a generic IP service and does not discuss the difference, from a security viewpoint, between using IPv4 or IPv6.

2. Threat model

This document assumes that attackers can be located anywhere in the Internet (either in LISP sites or outside LISP sites) and that attacks can be mounted either by a single attacker or by the collusion of several attackers.

An attacker is a malicious entity that performs the action of attacking a target in a network where LISP is (partially) deployed by leveraging the LISP protocol and/or architecture.

An attack is the action of performing an illegitimate action on a target in a network where LISP is (partially) deployed.

The target of an attack is the entity (i.e., a device connected to the network or a network) that is aimed to undergo the consequences of an attack. Other entities can potentially undergo side effects of an attack, even though they are not directly targeted by the attack. The target of an attack can be selected specifically, i.e., a particular entity, or arbitrarily, i.e., any entity. Finally, an

attacker can aim at attacking one or several targets with a single attack.

Section 2.1 specifies the different modes of operation that attackers can follow to mount attacks and Section 2.2 specifies the different categories of attacks that attackers can build.

2.1. Attacker's Operation Modes

In this document attackers are classified according to their modes of operation, i.e., the temporal and spacial diversity of the attacker. These modes are not mutually exclusive, they can be used by attackers in any combination, and other modes may be discovered in the future. Further, attackers are not at all bound by our classification scheme, so implementers and those deploying will always need to do additional risk analysis for themselves.

2.1.1. On-path vs. Off-path Attackers

On-path attackers, also known as Men-in-the-Middle, are able to intercept and modify packets between legitimate communicating entities. On-path attackers are located either directly on the normal communication path (either by gaining access to a node on the path or by placing themselves directly on the path) or outside the location path but manage to deviate (or gain a copy of) packets sent between the communication entities. On-path attackers hence mount their attacks by modifying packets initially sent legitimately between communication entities.

An attacker is called off-path attacker if it does not have access to packets exchanged during the communication or if there is no communication. In order for their attacks to succeed, off-path attackers must hence generate packets and inject them in the network.

2.1.2. Internal vs. External Attackers

An internal attacker launches its attack from a node located within a legitimate LISP site. Such an attacker is either a legitimate node of the site or it exploits a vulnerability to gain access to a legitimate node in the site. Because of their location, internal attackers are trusted by the site they are in.

On the contrary, an external attacker launches its attacks from the outside of a legitimate LISP site.

2.1.3. Live vs. Time-shifted attackers

A live attacker mounts attacks for which it must remain connected as long as the attack is mounted. In other words, the attacker must remain active for the whole duration of the attack. Consequently, the attack ends as soon as the attacker (or the used attack vector) is neutralized.

On the contrary, a time-shifted attacker mounts attacks that remain active after it disconnects from the Internet.

2.1.4. Control-plane vs. Data-plane attackers

A control-plane attacker mounts its attack by using control-plane functionalities, typically the mapping system.

A data-plane attacker mounts its attack by using data-plane functionalities.

As there is no complete isolation between the control-plane and the data-plane, an attacker can operate in the control-plane (or data-plane) to mount attacks targeting the data-plane (or control-plane) or keep the attacked and targeted planes at the same layer (i.e., from control-plane to control-plane or from data-plane to data-plane).

2.1.5. Cross mode attackers

The attacker modes of operation are not mutually exclusive and hence attackers can combine them to mount attacks.

For example, an attacker can launch an attack using the control-plane directly from within a LISP site to which it is able to get temporary access (i.e., internal + control-plane attacker) to create a vulnerability on its target and later on (i.e., time-shifted + external attacker) mount an attack on the data plane (i.e., data-plane attacker) that leverages the vulnerability.

2.2. Threat categories

Attacks can be classified according to the nine following categories. These categories are not mutually exclusive and can be used by attackers in any combination.

2.2.1. Replay attack

A replay attack happens when an attacker retransmits at a later time, and without modifying it, a packet (or a sequence of packets) that

has already been transmitted.

2.2.2. Packet manipulation

A packet manipulation attack happens when an attacker receives a packet, modifies the packet (i.e., changes some information contained in the packet) and finally transmits the packet to its final destination that can be the initial destination of the packet or a different one.

2.2.3. Packet interception and suppression

In a packet interception and suppression attack, the attacker captures the packet and drops it before it can reach its final destination.

2.2.4. Spoofing

With a spoofing attack, the attacker injects packets in the network pretending to be another node. Spoofing attacks are made by forging source addresses in packets.

It should be noted that with LISP, packet spoofing is similar to spoofing with any other existing tunneling technology currently deployed in the Internet. Generally the term "spoofed packet" indicates a packet containing a source IP address that is not the actual originator of the packet. Hence, since LISP uses encapsulation, the spoofed address could be in the outer header as well as in the inner header, this translates to two types of spoofing.

Inner address spoofing: the attacker uses encapsulation and uses a spoofed source address in the inner packet. In case of data-plane LISP encapsulation, that corresponds to spoofing the source EID (End-point IDentifier) address of the encapsulated packet.

Outer address spoofing: the attacker does not use encapsulation and spoofs the source address of the packet. In case of data-plane LISP encapsulation, that corresponds to spoofing the source RLOC (Routing LOCator) address of the encapsulated packet.

Note that the two types of spoofing are not mutually exclusive, rather all combinations are possible and could be used to perform different kinds of attacks. For example, an attacker outside a LISP site can generate a packet with a forged source IP address (i.e., outer address spoofing) and forward it to a LISP destination. The packet is then eventually encapsulated by a PITR (Proxy Ingress

Tunnel Router) so that once encapsulated the attack corresponds to a inner address spoofing. One can also imagine an attacker forging a packet with encapsulation where both inner and outer source addresses are spoofed.

It is important to note that the combination of inner and outer spoofing makes the identification of the attacker complex as the packet may not contain information that allows to detect the origin of the attack.

2.2.5. Rogue attack

In a rogue attack the attacker manages to appear as a legitimate source of information, without faking its identity (as opposed to a spoofing attacker).

2.2.6. Denial of Service (DoS) attack

A Denial of Service (DoS) attack aims at disrupting a specific targeted service to make it unable to operate properly.

2.2.7. Performance attack

A performance attacks aims at exploiting computational resources (e.g., memory, processor) of a targeted node so as to make it unable to operate properly.

2.2.8. Intrusion attack

In an intrusion attack, the attacker gains remote access to a resource (e.g., a host, a router, or a network) or information that it legitimately should not have access. Intrusion attacks can lead to privacy leakages.

2.2.9. Amplification attack

In an amplification attack, the traffic generated by the target of the attack in response to the attack is larger than the traffic that the attacker must generate.

In some cases, the data-plane can be several orders of magnitude faster than the control-plane at processing packets. This difference can be exploited to overload the control-plane via the data-plane without overloading the data-plane.

2.2.10. Passive Monitoring Attacks

An attacker can use pervasive monitoring, which is a technical attack [RFC7258], targeting information about LISP traffic that may or not be used to mount other type of attacks.

2.2.11. Multi-category attacks

Attacks categories are not mutually exclusive and any combination can be used to perform specific attacks.

For example, one can mount a rogue attack to perform a performance attack starving the memory of an ITR (Ingress Tunnel Router) resulting in a DoS (Denial-of-Service) on the ITR.

3. Attack vectors

This section presents attack techniques that may be used by attackers when leveraging the LISP protocol and/or architecture.

3.1. Gleaning

To reduce the time required to obtain a mapping, the optional gleaning mechanism defined for LISP allows an xTR (Ingress and/or Egress Tunnel Router) to directly learn a mapping from the LISP data encapsulated packets and the Map-Request packets that it receives. LISP encapsulated data packets contain a source RLOC, destination RLOC, source EID and destination EID. When an xTR receives an encapsulated data packet coming from a source EID for which it does not already know a mapping, it may insert the mapping between the source RLOC and the source EID in its EID-to-RLOC Cache. The same technique can be used when an xTR receives a Map-Request as the Map-Request also contains a source EID address and a source RLOC. Once a gleaned entry has been added to the EID-to-RLOC cache, the xTR sends a Map-Request to retrieve the actual mapping for the gleaned EID from the mapping system.

If a packet injected by an off-path attacker and with a spoofed inner address is gleaned by an xTR then the attacker may divert the traffic meant to be delivered to the spoofed EID as long as the gleaned entry is used by the xTR. This attack can be used as part of replay, packet manipulation, packet interception and suppression, or DoS attacks as the packets are sent to the attacker.

If the packet sent by the attacker contains a spoofed outer address instead of a spoofed inner address then it can achieve a DoS or a performance attack as the traffic normally destined to the attacker

will be redirected to the spoofed source RLOC. Such traffic may overload the owner of the spoofed source RLOC, preventing it from operating properly.

If the packet injected uses both inner and outer spoofing, the attacker can achieve a spoofing, a performance, or an amplification attack as traffic normally destined to the spoofed EID address will be sent to the spoofed RLOC address. If the attacked LISP site also generates traffic to the spoofed EID address, such traffic may have a positive amplification factor.

A gleaning attack does not only impact the data-plane but can also have repercussions on the control-plane as a Map-Request is sent after the creation of a gleaned entry. The attacker can then achieve DoS and performance attacks on the control-plane. For example, if an attacker sends a packet for each address of a prefix not yet cached in the EID-to-RLOC cache of an xTR, the xTR will potentially send a Map-Request for each such packet until the mapping is installed which leads to an over-utilisation of the control-plane as each packet generates a control-plane event. In order for this attack to succeed, the attacker may not need to use spoofing. This issue can occur even if gleaning is turned off since whether or not gleaning is used as the ITR may need to send a Map-Request in response to incoming packets whose EID is not currently in the cache.

Gleaning attacks are fundamentally involving a time-shifted mode of operation as the attack may last as long as the gleaned entry is kept by the targeted xTR. RFC 6830 [RFC6830] recommends to store the gleaned entries for only a few seconds which limits the duration of the attack.

Gleaning attacks always involve external data-plane attackers but results in attacks on either the control-plane or data-plane.

Note, the outer spoofed address does not need to be the RLOC of a LISP site, it may be any address.

3.2. Locator Status Bits

When the L bit in the LISP header is set to 1, it indicates that the second 32-bits longword of the LISP header contains the Locator Status Bits. In this field, each bit position reflects the status of one of the RLOCs mapped to the source EID found in the encapsulated packet. The reaction of a LISP xTR that receives such a packet is left as operational choice in [RFC6830].

When an attacker sends a LISP encapsulated packet with an illegitimately crafted LSB to an xTR, it can influence the xTR’s

choice of the locators for the prefix associated to the source EID. In case of an off-path attacker, the attacker must inject a forged packet in the network with a spoofed inner address. An on-path attacker can manipulate the LSB of legitimate packets passing through it and hence does not need to use spoofing. Instead of manipulating the LSB field, an on-path attacker can also obtain the same result of injecting packets with invalid LSB values by replaying packets.

The LSB field can be leveraged to mount a DoS attack by either declaring all RLOCs as unreachable (all LSB set to 0), or by concentrating all the traffic to one RLOC (e.g., all but one LSB set to 0) and hence overloading the RLOC concentrating all the traffic from the xTR, or by forcing packets to be sent to RLOCs that are actually not reachable (e.g., invert LSB values).

The LSB field can also be used to mount a replay, a packet manipulation, or a packet interception and suppression attack. Indeed, if the attacker manages to be on the path between the xTR and one of the RLOCs specified in the mapping, forcing packets to go via that RLOC implies that the attacker will gain access to the packets.

Attacks using the LSB are fundamentally involving a time-shifted mode of operation as the attack may last as long as the reachability information gathered from the LSB is used by the xTR to decide the RLOCs to be used.

3.3. Map-Version

When the Map-Version bit of the LISP header is set to 1, it indicates that the low-order 24 bits of the first 32 bits longword of the LISP header contain a Source and Destination Map-Version. When a LISP xTR receives a LISP encapsulated packet with the Map-Version bit set to 1, the following actions are taken:

- It compares the Destination Map-Version found in the header with the current version of its own configured EID-to-RLOC mapping, for the destination EID found in the encapsulated packet. If the received Destination Map-Version is smaller (i.e., older) than the current version, the ETR should apply the SMR (Solicit-Map-Request) procedure described in [RFC6830] and send a Map-Request with the SMR bit set.

- If a mapping exists in the EID-to-RLOC Cache for the source EID, then it compares the Map-Version of that entry with the Source Map-Version found in the header of the packet. If the stored mapping is older (i.e., the Map-Version is smaller) than the source version of the LISP encapsulated packet, the xTR should send a Map-Request for the source EID.

A cross-mode attacker can use the Map-Version bit to mount a DoS attack, an amplification attack, or a spoofing attack. For instance if the mapping cached at the xTR is outdated, the xTR will send a Map-Request to retrieve the new mapping which can yield to a DoS attack (by excess of signalling traffic) or an amplification attack if the data-plane packet sent by the attacker is smaller, or otherwise uses fewer resources, than the control-plane packets sent in response to the attacker's packet. With a spoofing attack, and if the xTR considers that the spoofed ITR has an outdated mapping, it will send an SMR to the spoofed ITR which can result in performance, amplification, or DoS attack as well.

Map-Version attackers are inherently cross mode as the Map-Version is a method to put control information in the data-plane. Moreover, this vector involves live attackers. Nevertheless, on-path attackers do not have specific advantage over off-path attackers.

3.4. Routing Locator Reachability

The Nonce-Present and Echo-Nonce bits in the LISP header are used to verify the reachability of an xTR. A testing xTR sets the Echo-Nonce and the Nonce-Present bits in LISP data encapsulated packets and include a random nonce in the LISP header of packets. Upon reception of these packets, the tested xTR stores the nonce and echoes it whenever it returns a LISP encapsulated data packets to the testing xTR. The reception of the echoed nonce confirms that the tested xTR is reachable.

An attacker can interfere with the reachability test by sending two different types of packets:

1. LISP data encapsulated packets with the Nonce-Present bit set and a random nonce. Such packets are normally used in response to a reachability test.

2. LISP data encapsulated packets with the Nonce-Present and the Echo-Nonce bits both set. These packets will force the receiving ETR to store the received nonce and echo it in the LISP encapsulated packets that it sends. These packets are normally used as a trigger for a reachability test.

The first type of packets are used to make xTRs think that an other xTR is reachable while it is not. It is hence a way to mount a DoS attack (i.e., the ITR will send its packet to a non-reachable ETR when it should use another one).

The second type of packets could be exploited to attack the nonce-based reachability test. If the attacker sends a continuous flow of

packets that each have a different random nonce, the ETR that receives such packets will continuously change the nonce that it returns to the remote ITR, which can yield to a performance attack. If the remote ITR tries a nonce-reachability test, this test may fail because the ETR may echo an invalid nonce. This hence yields to a DoS attack.

In the case of an on-path attacker, a packet manipulation attack is necessary to mount the attack. To mount such an attack, an off-path attacker must mount an outer address spoofing attack.

If an xTR chooses to periodically check with active probes the liveness of entries in its EID-to-RLOC cache (as described in section 6.3 of [RFC6830]), then this may amplify the attack that caused the insertion of entries being checked.

3.5. Instance ID

LISP allows to carry in its header a 24-bits value called Instance ID and used on the ITR to indicate which local Instance ID has been used for encapsulation, while on the ETR the instance ID decides the forwarding table to use to forward the decapsulated packet in the LISP site.

An attacker (either a control-plane or data-plane attacker) can use the instance ID functionality to mount an intrusion attack.

3.6. Interworking

[RFC6832] defines Proxy-ITR and Proxy-ETR network elements to allow LISP and non-LISP sites to communicate. The Proxy-ITR has functionality similar to the ITR, however, its main purpose is to encapsulate packets arriving from the DFZ (Default-Free Zone) in order to reach LISP sites. A PETR (Proxy Egress Tunnel Router) has functionality similar to the ETR, however, its main purpose is to inject de-encapsulated packets in the DFZ in order to reach non-LISP sites from LISP sites. As a PITR (or PETR) is a particular case of ITR (or ETR), it is subject to similar attacks as ITRs (or ETRs).

As any other system relying on proxies, LISP interworking can be used by attackers to hide their exact origin in the network.

3.7. Map-Request messages

A control-plane off-path attacker can exploit Map-Request messages to mount DoS, performance, or amplification attacks. By sending Map-Request messages at high rate, the attacker can overload nodes involved in the mapping system. For instance sending Map-Requests at

high rate can considerably increase the state maintained in a Map-Resolver or consume CPU cycles on ETRs that have to process the Map-Request packets they receive in their slow path (i.e., performance or DoS attack). When the Map-Reply packet is larger than the Map-Request sent by the attacker, that yields to an amplification attack. The attacker can combine the attack with a spoofing attack to overload the node to which the spoofed address is actually attached.

Note, if the attacker sets the P bit (Probe Bit) in the Map-Request, it will cause legitimately sending the Map-Request directly to the ETR instead of passing through the mapping system.

The SMR bit can be used to mount a variant of these attacks.

For efficiency reasons, Map-Records can be appended to Map-Request messages. When an xTR receives a Map-Request with appended Map-Records, it does the same operations as for the other Map-Request messages and so is subject to the same attacks. However, it also installs in its EID-to-RLOC cache the Map-Records contained in the Map-Request. An attacker can then use this vector to force the installation of mappings in its target xTR. Consequently, the EID-to-RLOC cache of the xTR is polluted by potentially forged mappings allowing the attacker to mount any of the attacks categorized in Section 2.2 (see Section 3.8 for more details). Note, the attacker does not need to forge the mappings present in the Map-Request to achieve a performance or DoS attack. Indeed, if the attacker owns a large enough EID prefix it can de-aggregate it in many small prefixes, each corresponding to another mapping and it installs them in the xTR cache by mean of the Map-Request.

Moreover, attackers can use Map Resolver and/or Map Server network elements to relay its attacks and hide the origin of the attack. Indeed, on the one hand, a Map Resolver is used to dispatch Map-Request to the mapping system and, on the other hand, a Map Server is used to dispatch Map-Requests coming from the mapping system to ETRs that are authoritative for the EID in the Map-Request.

3.8. Map-Reply messages

Most of the security risks associated with Map-Reply messages will depend on the 64 bits nonce that is included in a Map-Request and returned in the Map-Reply. Given the size of the nonce (64 bits), if best current practice is used [RFC4086] and if an ETR does not accept Map-Reply messages with an invalid nonce, the risk of an off-path attack is limited. Nevertheless, the nonce only confirms that the Map-Reply received was sent in response to a Map-Request sent, it does not validate the contents of that Map-Reply.

If an attacker manages to send a valid (i.e., in response to a Map-Request and with the correct nonce) Map-Reply to an ITR, then it can perform any of the attacks categorised in Section 2.2 as it can inject forged mappings directly in the ITR EID-to-RLOC cache. For instance, if the mapping injected to the ITR points to the address of a node controlled by the attacker, it can mount replay, packet manipulation, packet interception and suppression, or DoS attacks, as it will receive every packet destined to a destination lying in the EID prefix of the injected mapping. In addition, the attacker can inject a plethora of mappings in the ITR to mount a performance attack by filling up the EID-to-RLOC cache of the ITR. The attacker can also mount an amplification attack if the ITR at that time is sending a large number of packets to the EIDs matching the injected mapping. In this case, the RLOC address associated to the mapping is the address of the real target of the attacker and so all the traffic of the ITR will be sent to the target which means that with one single packet the attacker may generate very high traffic towards its final target.

If the attacker is a valid ETR in the system, it can mount a rogue attack if it uses prefixes over-claiming. In such a scenario, the attacker ETR replies to a legitimate Map-Request message which it received with a Map-Reply message that contains an EID-Prefix that is larger than the prefix owned by the attacker. For example if the owned prefix is 192.0.2.0/25 but the Map-Reply contains a mapping for 192.0.2.0/24, then the mapping will influence packets destined to other EIDs than the one attacker has authority on. With such technique, the attacker can mount the attacks presented above as it can (partially) control the mappings installed on its target ITR. To force its target ITR to send a Map-Request, nothing prevents the attacker to initiate some communication with the ITR. This method can be used by internal attackers that want to control the mappings installed in their site. To that aim, they simply have to collude with an external attacker ready to over-claim prefixes on behalf of the internal attacker.

Note, when the Map-Reply is in response to a Map-Request sent via the mapping system (i.e., not send directly from the ITR to an ETR), the attacker does not need to use a spoofing attack to achieve its attack as by design the source IP address of a Map-Reply is not known in advance by the ITR.

Map-Request and Map-Reply messages are exposed to any type of attackers, on-path or off-path but also external or internal attackers. Also, even though they are control message, they can be leveraged by data-plane attackers. As the decision of removing mappings is based on the TTL indicated in the mapping, time-shifted attackers can take advantage of injecting forged mappings as well.

3.9. Map-Register messages

Map-Register messages are sent by ETRs to Map Servers to indicate to the mapping system the EID prefixes associated to them. The Map-Register message provides an EID prefix and the list of ETRs that are able to provide Map-Replies for the EID covered by the EID prefix.

As Map-Register messages are protected by an authentication mechanism, only a compromised ETR can register itself to its allocated Map Server.

A compromised ETR can over-claim the prefix it owns in order to influence the route followed by Map-Requests for EIDs outside the scope of its legitimate EID prefix (see Section 3.8 for the list of over-claiming attacks).

A compromised ETR can also de-aggregate its EID prefix in order to register more EID prefixes than necessary to its Map Servers (see Section 3.7 for the impact of de-aggregation of prefixes by an attacker).

Similarly, a compromised Map Server can accept an invalid registration or advertise an invalid EID prefix to the mapping system.

3.10. Map-Notify messages

Map-Notify messages are sent by a Map Server to an ETR to acknowledge the reception and processing of a Map-Register message.

Similarly to the pair Map-Request/Map-Reply, the pair Map-Register/ Map-Notify is protected by a nonce making it difficult for an attacker to inject a falsified notification to an ETR to make this ETR believe that the registration succeeded when it has not.

4. Note on Privacy

As reviewed in [RFC6973], universal privacy considerations are difficult to establish as the privacy definitions may vary for different scenarios. As a consequence, this document does not aim at identifying privacy issues related to the LISP protocol but the security threats identified in this document could play a role in privacy threats as defined in section 5 of [RFC6973].

Similar to public deployments of any other control plane protocols, in an Internet deployment, LISP mappings are public and hence provide information about the infrastructure and reachability of LISP sites

(i.e., the addresses of the edge routers). Depending upon deployment details, LISP map replies might or might not provide finer grained and more detailed information than is available with currently deployed routing and control protocols.

5. Threats Mitigation

Most of the above threats can be mitigated with careful deployment and configuration (e.g., filter) and also by applying the general rules of security, e.g. only activating features that are necessary for the deployment and verifying the validity of the information obtained from third parties.

The control-plane is the most critical part of LISP from a security viewpoint and it is worth to notice that the LISP specifications already offer an authentication mechanism for mappings registration ([RFC6833]). This mechanism, combined with LISP-SEC [I-D.ietf-lisp-sec], strongly mitigates threats in non-trustable environments such as the Internet. Moreover, an authentication data field for Map-Request messages and Encapsulated Control messages was allocated [RFC6830]. This field provides a general authentication mechanism technique for the LISP control-plane which future specifications may use while staying backward compatible. The exact technique still has to be designed and defined. To maximally mitigate the threats on the mapping system, authentication must be used, whenever possible, for both Map-Request and Map-Reply messages and for messages exchanged internally among elements of the mapping system, such as specified in [I-D.ietf-lisp-sec] and [I-D.ietf-lisp-ddt].

Systematically applying filters and rate-limitation, as proposed in [RFC6830], will mitigate most of the threats presented in this document. In order to minimise the risk of overloading the control-plane with actions triggered from data-plane events, such actions should be rate limited.

Moreover, all information opportunistically learned (e.g., with LSB or gleaning) should be used with care until they are verified. For example, a reachability change learned with LSB should not be used directly to decide the destination RLOC, but instead should trigger a rate-limited reachability test. Similarly, a gleaned entry should be used only for the flow that triggered the gleaning procedure until the gleaned entry has been verified [Trilogy].

6. Security Considerations

This document provides a threat analysis and proposes mitigation techniques for the Locator/Identifier Separation Protocol.

7. IANA Considerations

This document makes no request to IANA.

8. Acknowledgments

This document builds upon the document of Marcelo Bagnulo ([I-D.bagnulo-lisp-threat]), where the flooding attack and the reference environment was first described.

The authors would like to thank Deborah Brungard, Ronald Bonica, Albert Cabellos, Ross Callon, Noel Chiappa, Florin Coras, Vina Ermagan, Dino Farinacci, Stephen Farrell, Joel Halpern, Emily Hiltzik, Darrel Lewis, Edward Lopez, Fabio Maino, Terry Manderson, and Jeff Wheeler for their comments.

This work has been partially supported by the INFSO-ICT-216372 TRILOGY Project (www.trilogy-project.org).

The work of Luigi Iannone has been partially supported by the ANR-13-INFR-0009 LISP-Lab Project (www.lisp-lab.org) and the EIT KIC ICT-Labs SOFNETS Project.

9. References

9.1. Normative References

[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>.

[RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, DOI 10.17487/RFC6832, January 2013, <http://www.rfc-editor.org/info/rfc6832>.

[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>.

[RFC6834] Iannone, L., Saucez, D., and O. Bonaventure, "Locator/ID Separation Protocol (LISP) Map-Versioning", RFC 6834, DOI 10.17487/RFC6834, January 2013, <http://www.rfc-editor.org/info/rfc6834>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <http://www.rfc-editor.org/info/rfc6973>.

9.2. Informative References

[I-D.bagnulo-lisp-threat]
Bagnulo, M., "Preliminary LISP Threat Analysis", draft-bagnulo-lisp-threat-01 (work in progress), July 2007.

[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-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.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <http://www.rfc-editor.org/info/rfc4086>.

[RFC7215] Jakab, L., Cabellos-Aparicio, A., Coras, F., Domingo-Pascual, J., and D. Lewis, "Locator/Identifier Separation Protocol (LISP) Network Element Deployment Considerations", RFC 7215, DOI 10.17487/RFC7215, April 2014, <http://www.rfc-editor.org/info/rfc7215>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <http://www.rfc-editor.org/info/rfc7258>.

[Trilogy] Saucez, D. and L. Iannone, "How to mitigate the effect of scans on mapping systems", Trilogy Future Internet Summer

School., 2009.

Appendix A. Document Change Log (to be removed on publication)

- Version 15 Posted January 2016.
  - Few changes to address Stephen Farrel comments as part of the IESG Review.
- Version 14 Posted December 2015.
  - Editorial changes according to Deborah Brungard’s (Routing AD) review.
- Version 13 Posted August 2015.
  - Keepalive version.
- Version 12 Posted March 2015.
  - Addressed comments by Ross Callon on the mailing list (http://www.ietf.org/mail-archive/web/lisp/current/msg05829.html).
  - Addition of a section discussing mitigation techniques for deployments in non-trustable environments.
- Version 11 Posted December 2014.
  - Editorial polishing. Clarifications added in few points.
- Version 10 Posted July 2014.
  - Document completely remodelled according to the discussions on the mailing list in the thread http://www.ietf.org/mail-archive/web/lisp/current/msg05206.html and to address comments from Ronald Bonica and Ross Callon.
- Version 09 Posted March 2014.
  - Updated document according to the review of A. Cabellos.
- Version 08 Posted October 2013.
  - Addition of a privacy consideration note.
  - Editorial changes

- Version 07 Posted October 2013.
  - This version is updated according to the thorough review made during October 2013 LISP WG interim meeting.
  - Brief recommendations put in the security consideration section.
  - Editorial changes
- Version 06 Posted October 2013.
  - Complete restructuration, temporary version to be used at October 2013 interim meeting.
- Version 05 Posted August 2013.
  - Removal of severity levels to become a short recommendation to reduce the risk of the discussed threat.
- Version 04 Posted February 2013.
  - Clear statement that the document compares threats of public LISP deployments with threats in the current Internet architecture.
  - Addition of a severity level discussion at the end of each section.
  - Addressed comments from V. Ermagan and D. Lewis' reviews.
  - Updated References.
  - Further editorial polishing.
- Version 03 Posted October 2012.
  - Dropped Reference to RFC 2119 notation because it is not actually used in the document.
  - Deleted future plans section.
  - Updated References
  - Deleted/Modified sentences referring to the early status of the LISP WG and documents at the time of writing early versions of the document.

    * Further editorial polishing.

    * Fixed all ID nits.

  o Version 02 Posted September 2012.

    * Added a new attack that combines over-claiming and de-aggregation (see Section 3.8).

    * Editorial polishing.

  o Version 01 Posted February 2012.

    * Added discussion on LISP-DDT.

  o Version 00 Posted July 2011.

    * Added discussion on LISP-MS>.

    * Added discussion on Instance ID.

    * Editorial polishing of the whole document.

    * Added "Change Log" appendix to keep track of main changes.

    * Renamed "draft-saucez-lisp-security-03.txt.

Authors' Addresses

Damien Saucez
INRIA
2004 route des Lucioles BP 93
06902 Sophia Antipolis Cedex
France

Email: damien.saucez@inria.fr

Luigi Iannone
Telecom ParisTech
23, Avenue d'Italie, CS 51327
75214 PARIS Cedex 13
France

Email: ggx@gigix.net

Olivier Bonaventure
Universite catholique de Louvain
Place St. Barbe 2
Louvain la Neuve
Belgium

Email: olivier.bonaventure@uclouvain.be