

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: January 21, 2015

D. Farinacci
lispers.net
July 20, 2014

LISP Data-Plane Confidentiality
draft-farinacci-lisp-crypto-01

Abstract

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on January 21, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Overview	3
3.	Diffie-Hellman Key Exchange	3
4.	Encoding and Transmitting Key Material	4
5.	Data-Plane Operation	6
6.	Dynamic Rekeying	6
7.	Future Work	7
8.	Security Considerations	7
8.1.	SAAG Support	7
8.2.	LISP-Crypto Security Threats	8
9.	IANA Considerations	8
10.	References	8
10.1.	Normative References	8
10.2.	Informative References	9
Appendix A.	Acknowledgments	9
Appendix B.	Document Change Log	10
B.1.	Changes to draft-farinacci-lisp-crypto-01.txt	10
B.2.	Changes to draft-farinacci-lisp-crypto-00.txt	10
	Author's Address	10

[1.](#) Introduction

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

This draft has the following requirements for the solution space:

- o Do not require a separate Public Key Infrastructure (PKI) that is out of scope of the LISP control-plane architecture.
- o The budget for key exchange MUST be one round-trip time. That is, only a two packet exchange can occur.
- o Use symmetric keying so faster cryptography can be performed in the LISP data plane.
- o Avoid a third-party trust anchor if possible.

- o Provide for rekeying when secret keys are compromised.
- o At this time, encapsulated packet authentication is not a strong requirement.

2. Overview

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

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

3. Diffie-Hellman Key Exchange

LISP will use a Diffie-Hellman [\[RFC2631\]](#) key exchange sequence and computation for computing a shared secret. The Diffie-Hellman parameters will be passed in Map-Request and Map-Reply messages.

Here is a brief description how Diff-Hellman works:

ITR			ETR		
Secret	Public	Calculates	Sends	Calculates	Public
i	p, g		p, g -->		e
i	p, g, I	$g^i \bmod p=I$	I -->		p, g, I
i	p, g, I		<-- E	$g^e \bmod p=E$	p, g
i, s	p, g, I, E	$E^i \bmod p=s$		$I^e \bmod p=s$	p, g, I, E
					e, s

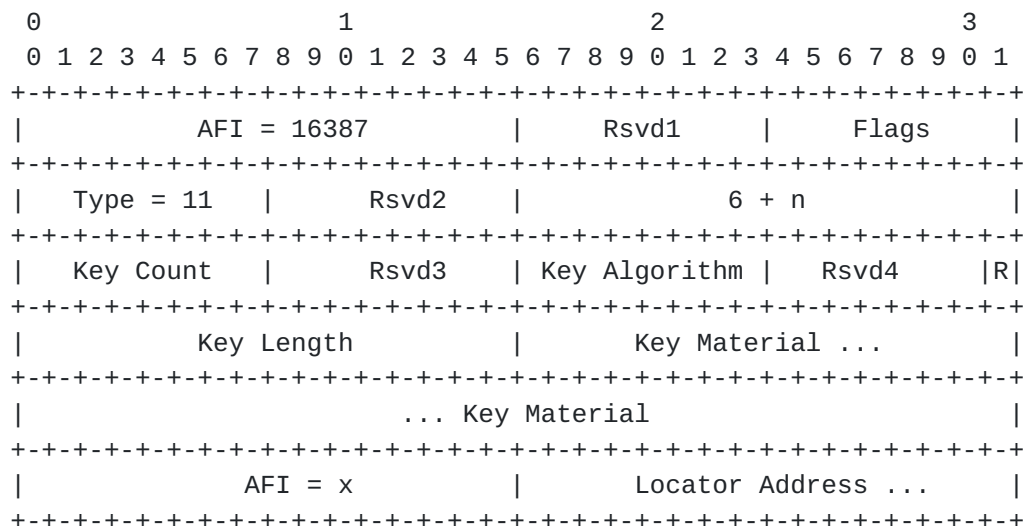
Public-key exchange for computing a shared private key [\[DH\]](#)

Diffie-Hellman parameters 'p' and 'g' must be the same values used by the ITR and ETR. The ITR computes public-key 'I' and transmits 'I'

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

4. Encoding and Transmitting Key Material

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



Diffie-Hellman parameters encoded in Key Material field

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

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

The 'Key Algorithm' encodes the cryptographic algorithm used. The following values are defined:

Null: 0
 Group-ID: 1
 AES: 2
 3DES: 3
 SHA-256: 4

When the 'Key Algorithm' value is 1 (Group-ID), the 'Key Material' field is encoded as:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                               Group ID                               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                               Public Key ...                          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  
```

Points to Key Material values from IANA Registry

The Group-ID values are defined in [\[RFC2409\]](#) and [\[RFC3526\]](#) which describe the Diffie Hellman parameters used for key exchange.

When the 'Key Algorithm' value is not 1 (Group-ID), the 'Key Material' field is encoded as:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|   g-length   |                               g-value ...           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|   p-length   |                               p-value ...           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                               Public Key ...                          |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
  
```

Key Length describes the length of the Key Material field

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

If an ITR or PITR sends a Map-Request with a Security Type LCAF included and the ETR or RTR does not want to have encapsulated traffic encrypted, they will return a Map-Reply with no RLOC records encoded with the Security Type LCAF. This signals to the ITR or PITR

that it should not encrypt traffic (it cannot encrypt traffic anyways since no ETR public-key was returned).

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

5. Data-Plane Operation

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

	0										1										2										3									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+						
/		Source Port = xxxx											Dest Port = 4341																											
UDP	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-						
\		UDP Length											UDP Checksum																											
+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+						
L		N		L		E		V		I		P		K		Nonce/Map-Version																								
I	\	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+						
S	/		Instance ID/Locator-Status-Bits																																					
P	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-	+	-						

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

When the KK bits are 00, the encapsulated packet is not encrypted. When the value of the KK bits is 1, 2, or 3, it encodes the key-id of the secret keys computed during the Diffie-Hellman Map-Request/Map-Reply exchange.

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

6. Dynamic Rekeying

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

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

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

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

7. Future Work

By using AES-GCM [[RFC5116](#)], or HMAC-CBC [[AES-CBC](#)], it has been suggested that encapsulated packet authentication (through encryption [[RFC4106](#)]) could be supported. There is current work in progress to investigate these techniques for the LISP data-plane. However, it will require encapsulation header changes to LISP.

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

8. Security Considerations

8.1. SAAG Support

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

8.2. LISP-Crypto Security Threats

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

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

9. IANA Considerations

This draft requires the use of the registry that selects Diffie Hellman parameters. Rather than convey the key exchange parameters directly in LISP control packets, a Group-ID from the registry will be used. The Group-ID values are defined in [[RFC2409](#)] and [[RFC3526](#)].

10. References

10.1. Normative References

- [RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", [RFC 2409](#), November 1998.
- [RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", [RFC 2631](#), June 1999.
- [RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", [RFC 3526](#), May 2003.
- [RFC4106] Viega, J. and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)", [RFC 4106](#), June 2005.

- [RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", [RFC 4492](#), May 2006.
- [RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", [RFC 5116](#), January 2008.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", [RFC 6830](#), January 2013.

10.2. Informative References

- [AES-CBC] McGrew, D., Foley, J., and K. Paterson, "Authenticated Encryption with AES-CBC and HMAC-SHA", [draft-mcgrew-aead-aes-cbc-hmac-sha2-03.txt](#) (work in progress), .
- [DH] "Diffie-Hellman key exchange", Wikipedia http://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange, .
- [LCAF] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format", [draft-ietf-lisp-lcaf-04.txt](#) (work in progress), .
- [LISP-DDT] Fuller, V., Lewis, D., Ermaagan, V., and A. Jain, "LISP Delegated Database Tree", [draft-fuller-lisp-ddt-03](#) (work in progress), .
- [LISP-SEC] Maino, F., Ermagan, V., Cabellos, A., and D. Saucez, "LISP-Security (LISP-SEC)", [draft-ietf-lisp-sec-06](#) (work in progress), .

Appendix A. Acknowledgments

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

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

Appendix B. Document Change Log

B.1. Changes to [draft-farinacci-lisp-crypto-01.txt](#)

- o Posted July 2014.
- o Add Group-ID to the encoding format of Key Material in a Security Type LCAF and modify the IANA Considerations so this draft can use key exchange parameters from the IANA registry.
- o Indicate that the R-bit in the Security Type LCAF is not used by lisp-crypto.
- o Add text to indicate that ETRs/RTRs can negotiate less number of keys from which the ITR/PITR sent in a Map-Request.
- o Add text explaining how LISP-SEC solves the problem when a man-in-the-middle becomes part of the Map-Request/Map-Reply key exchange process.
- o Add text indicating that when RLOC-probing is used for RLOC reachability purposes and rekeying is not desired, that the same key exchange parameters should be used so a reallocation of a public key does not happen at the ETR.
- o Add text to indicate that ECDH can be used to reduce CPU requirements for computing shared secret-keys.

B.2. Changes to [draft-farinacci-lisp-crypto-00.txt](#)

- o Initial draft posted February 2014.

Author's Address

Dino Farinacci
lispers.net
San Jose, California
USA

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

