

LISP Data-Plane Confidentiality

draft-farinacci-lisp-crypto-00

LISP Working Group
London IETF
March 2014

Dino Farinacci

*Thanks to: Dan Harkins, Brian Weis, Joel Halpern, Fabio Maino,
Roger Jorgensen, Ed Lopez*

Chronology

- Presented ideas in LISP WG in Vancouver fall 2013
- Seek advice from SAAG in Vancouver fall 2013
- Present solution here in London spring 2014

Requirements

- Confidentiality of packet stream in core network
 - Between ITR and ETR
- Do not incur additional send latency
 - Do not increase mapping database lookup time
 - Do not increase time before encapsulation can begin
- Use state-of-the-art cryptography for best packet switching performance
 - Use symmetric keys for encryption
- Keep OpEx as low as possible

First Thoughts

- Don't use a separate PKI outside of LISP
 - Use mapping database to store key material
- Use asymmetric keying to reduce key message exchange
- Encrypt with public-key and Decrypt with private-key

SAAG said . . .

- Better to not store keys anywhere
- You can do a key exchange with 2 messages
 - In 1 RTT
- Use Diffie-Hellman

draft-farinacci-lisp-crypto-00

- DO NOT use mapping database to store keys
- Use Map-Request/Map-Reply exchange between ITR and ETR for key exchange
- Same shared secret is computed by ITR for encryption and used by ETR for decryption
- Encrypt the EID payload
 - EIDs are obfuscated - user payload is ciphertext
 - UDP and LISP headers sent in the clear

Diffie-Hellman Exchange

Alice				Bob		
Secret	Public	Calculates	Sends	Calculates	Public	Secret
a	p, g		$p, g \rightarrow$			b
a	p, g, A	$g^a \bmod p = A$	$A \rightarrow$		p, g	b
a	p, g, A		$\leftarrow B$	$g^b \bmod p = B$	p, g, A, B	b
a, s	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	b, s

1. Alice and Bob agree to use a prime number $p = 23$ and base $g = 5$.

Map-Request

Map-Reply

draft-farinacci-lisp-crypto-00

- Each DH exchange computes shared-key for a key-id
- We have 2 flag bits left in LISP header
 - b'00' - packet not encrypted
 - b'01' - key-id 1
 - b'10' - key-id 2
 - b'11' - key-id 3
- Can use multiple keys between ITR and ETR for:
 - Mixing encryption
 - Rekeying when threat of key compromise

Encoding

- We have a Security Type LCAF that encodes key-id, cipher-type, and key material
- ETR uses RLOC-record in Map-Reply to encode 2-tuple:
 - RLOC address
 - Security material
- ITR builds Security LCAF in ITR-RLOCs field of Map-Request with 2-tuple

What has to change

- Nothing in the core network
- Nothing at the LISP site
- Nothing in the mapping system
- xTR data-plane requires changes
- xTR control-plane needs to build and parse Security Type LCAF

Comments Received

- What if MITM intercepts the key exchange?
 - Response: Use LISP-SEC to verify signed Map-Replies
- Do not pass g/p parameters in key material
 - Response: Use a registry to assign values to popular g/p pairs
- Can we Authenticate the encapsulation stream?
 - Response: Considering Authenticated Encryption with AEAD where UDP/LISP headers are AD

Comments Received

- What if the ETR doesn't want to do crypto?
 - Response: Then it doesn't return a Security Type LCAF in the Map-Reply
- What if the ETR doesn't want to do multiple keys?
 - Response: Then it returns a public-key for the number of key-ids it desires
- Is this design using the R-bit in the Security Type LCAF?
 - Response: No, that is there for LISP-DDT-sec - the Security Type LCAF is used for multiple use cases

Working Group Work Item?

- Security is in WG charter
- There has been so much attention recently on data privacy - go ask Angela Merkel :-)