Adding Data-Plane Security to the LISP Protocol

For SAAG
Vancouver IETF
November 2013

Dino Farinacci
Roger Jorgensen
Ed Lopez
Joel Halpern

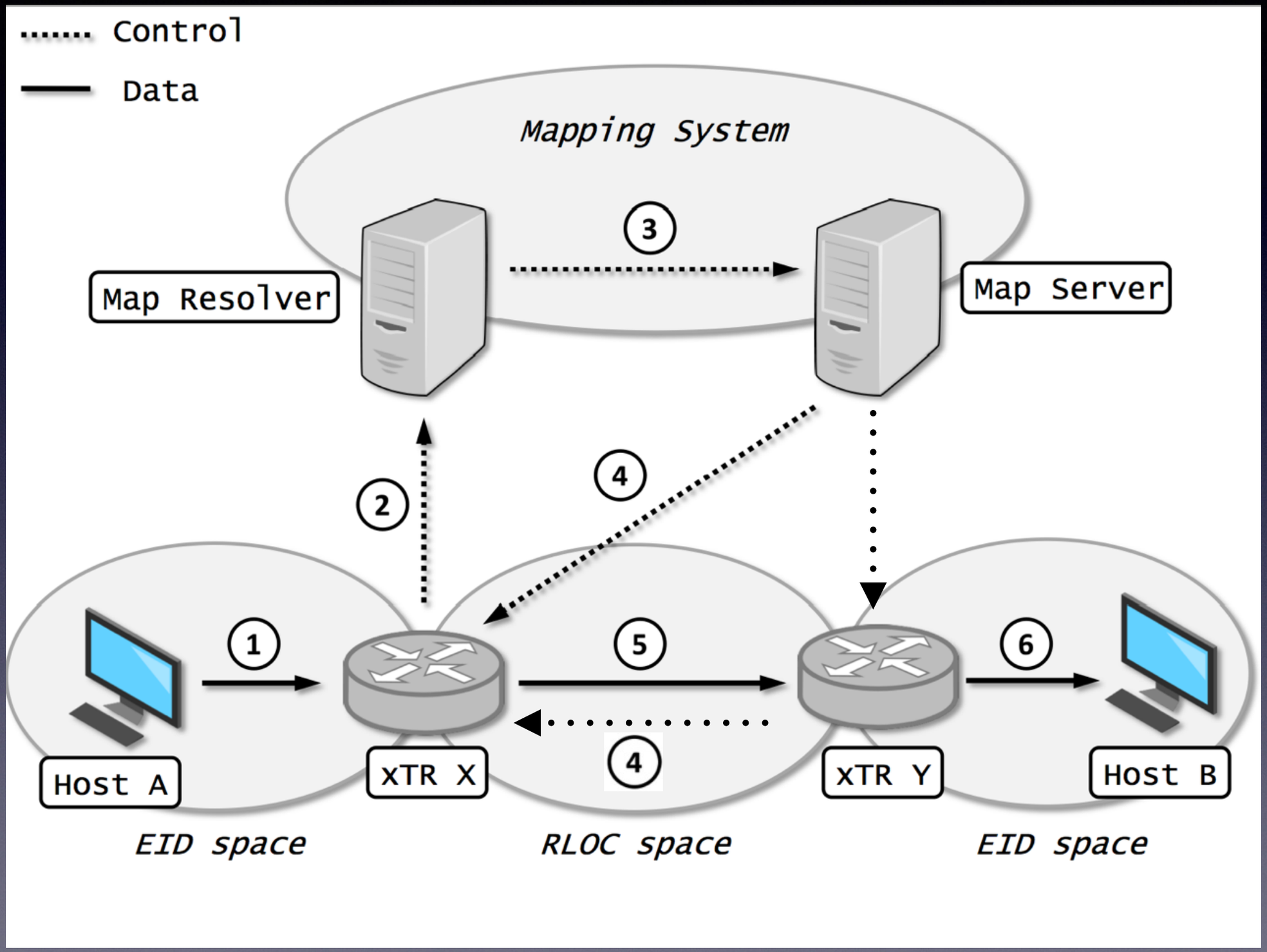
Preface

- I will save you all time and effort
  - I'm a Security Idiot
  - I'm a Routing Guy
- The LISP WG is coming to you right now for help
- These ideas are early and are just initial ideas
  - There is no Internet Draft written yet
- We are trying to be proactive (and not reactive)

Problem Statement

- Wouldn't it be good to protect the LISP data-plane?
  - Requirement is Confidentiality
  - If we get Integrity Checks for free we'll take it
- Wouldn't it be simpler to not require a PKI infrastructure?
  - LISP has a Mapping Database that could be used as a lightweight PKI
- Wouldn't it be good to do key exchange with one request/reply transaction?

LISP At-a-Glance



Obvious Solution
(from a Routing Perspective)

- Put key material in the LISP mapping database system
- Exchange keys with a LISP Map-Request/Map-Reply transaction

How?

- We have a Security Type LCAF that encodes key-type, cipher-type, and key material
- The RLOC-record in Map-Reply contains a 2-tuple of:
  - RLOC address
  - Security key
- ITR caches 2-tuple and then **encrypts-and-encaps**
- ETR **decaps-and-decrypts**

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

Please Advise Us

Backup Slides

(the rest of this slide-set were presented at the LISP WG)

Key Management - *asymdb*

- Use asymmetric keys
  - ETR register it's public key to mapping system
  - ITR uses public key to encrypt
  - ETR uses private key to decrypt
- **Pro**: keys can be exchanged in clear (with a 2-packet exchange)
- **Con**: asymmetric ciphers more compute intensive

Key Management - *symdb*

- Use symmetric keys - but must be transmitted securely
- Could use 2 step approach
    - Use public/private key in mapping database to secure the symmetric key
        - Then create shared secret symmetric key to use in data-plane
- ITR uses symmetric key for encryption, ETR uses same symmetric key for decryption
- **Pro**: faster ciphers
- **Con**: more keys to manage and more than a 2-packet exchange required

Key Management - *symmr*

- Use symmetric keys - alternative 2 step approach
  - Do not put keys in mapping database
  - Symmetric key returned in Map-Reply securely
  - Map-Reply is encrypted with map-server OTK
    - Map-Server OTK derived from ITR's OTK via LISP-SEC design
    - ITR can decrypt Map-Reply and cache shared secret symmetric key
- ITR uses symmetric key for encryption, ETR uses same symmetric key for decryption
- **Pro**: faster ciphers and one transaction to exchange shared secret
- **Con**: more keys

Key Management

- I'm sure there are other approaches with more combinations of key usage
- Let's try not to over-engineer this