UPC - BarcelonaTech

Master CANS Thesis

Privacy extensions for LISP-MN

Student:
Alberto
Rodríguez Natal

Advisor:
Albert
Cabellos Aparicio
Co-Advisor:
Loránd Jakab

June, 2012

Abstract

The current Internet architecture was not designed to easily accommodate mobility because IP addresses are used both to identify and locate hosts. The Locator/Identifier Separation Protocol (LISP) decouples them by considering two types of addresses: Endpoint IDentifiers (EIDs) to identify hosts, and Routing LOCators (RLOCs) that identify network attachment points. LISP, with such separation in place, offers native mobility. In this context, LISP-MN (LISP Mobile Node) is a particular case of LISP. Mobility protocols have an inherent issue with privacy since some users may not want to reveal their location or their identity. In this work, an overview of LISP-MN is presented as well as a proposal to extend it to enable privacy, both in terms of location and identity.

Contents

Chapter 1

Introduction

Today's Internet has become a network which dimension is far from the original idea. Designed as an intercommunication network between a few nodes relatively close, today it serves millions of users all over the world.

However, its design principles have remained unchanged despite the tremendous growth of the Internet. Most of the original design and its consequent limitations has survived until the present day. This is why exists a huge research towards the design of the future Internet. These research proposals can be split into two main approaches. The clean-slate ones stand for a complete redesign of the current Internet architecture, while the evolutionary ones uphold the current Internet design and aim to build the improvements over the already established infrastructure [1].

One of these evolutionary approaches is location/identity separation. The idea was originally proposed in [2] and later, several schemes following this idea can be found in [3], [4] and [6]. Among the various location/identity separation schemes, the Locator/Identifier Separation Protocol (LISP) ([14],[15]) has a unique position: LISP is incrementally deployable, it does not require changes to transport/application implementations and, more importantly, it is already under active deployment.

At the time of this writing, the main LISP IETF draft of the protocol specification has been approved to become a RFC, and so do some of the most important drafts related. There is an active community behind LISP and important research efforts are being devoted into its development and deployment. LISP is currently being deployed at a Beta-Network [11], this pilot network is a multi-company multi-vendor effort to research real-world behaviour of the LISP Protocol and includes a total of 156 LISP-enabled networks spread in 26 different countries. There are LISP implementations for Cisco routers operating systems, for FreeBSD (OpenLISP [12]), a private version for Android and an open-source version for GNU/Linux (LISPmob

[13]). LISP proposes two different types of addresses: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). EIDs identify hosts, and are assigned independently of the network topology while RLOCs identify network attachment points, and are used for routing. This allows EIDs to remain unchanged even if a topological change, such as a handover, occurs. Thus, LISP innate support for location/identity separation makes it well suited for mobility. Indeed, the LISP mobility protocol (LISP-MN) [18] proposes LISP-enabled endpoints, providing legacy applications with smooth mobility across access technologies and service providers. LISP introduces a Mapping System [16] as well, a distributed database that contains EID-to-RLOC bindings. The LISP-MN protocol leverages the Mapping System to disseminate such bindings.

LISP-MN, as any mobility protocol is subject to several security concerns. Among them, one of the most severe issues for a mobile device is privacy. Due to its condition of mobile, and in many cases personal device (i.e. mobile phone), it tends to be more relevant the location of this device and who is using it. Regarding that, privacy issue can be split in two different problems. First, a mobile node may want to keep its location safe in order to not be tracked or located by third parties. On the other hand, it may also want to hide its identity to prevent being identified.

This work proposes extensions to LISP-MN to address both issues: location and identity privacy. It is important to note that this work takes a realistic approach when extending LISP-MN since the aim is to propose a *deployable* solution. Although the LISP protocol is currently under development, some of its elements and specifications are not trivial to redefine and hence, the changes to the main LISP protocol are minimized. Further, this work briefly discusses a business model for the proposed extensions and, how to implement them on top of LISPmob, the reference LISP-MN implementation.

The rest of this work is structured as follows. Chapter 2 defines the state of the art regarding location/identity split, mobility and privacy while chapter 3 provides an overview of LISP and LISP-MN. Next, in chapter 4 are exposed the problems this work is trying to solve. Chapter 6 explains the proposed solutions for these problems. After that, chapter 7 discusses the proposed solutions. Finally, in chapter 8 conclusions are presented. At the end of this document there is an appendix (A) compiling all the related contributions this work has generated.

Chapter 2

State of the art

2.1 Location/identity split

One of the forums that point out the current Internet architecture issues was the Internet Architecture Board's (IAB)'s Routing and Addressing Workshop [37] issued in 2006. The main targets of this workshop were to analyse the factors that are inducing the growth of the routing tables as well as the constrains in current router technology and the limitations of the nowadays addressing system. One of the concerns exposed in this workshop (and in many others) was the semantic overload of the in use addresses. Today Internet addresses are used to identify points of anchor to the network, but also to identify end hosts. This excess of meaning associated to the Internet addresses restricts the evolution of the network architecture. Renewed the interest in these topics, due to the IAB's workshop, many proposals were made to solve these problems. Several shared the same idea of splitting the location and identity use of the Internet addresses.

The location/identity split idea is based in separate these two functions into two different, disjoint, name-spaces. Endpoint identifiers associated with endpoints and routing locators which define locations in the network topology. This split provides several advantages. One of them is that it lets to scale the grow of the routing table. Locators could be highly aggregated if they are assigned accordingly to the network topology. This can be done because the locators have no longer to follow organizational hierarchies. This is now done by the endpoint identifiers. The locator/identity split lets to manage the logical and the physical network as two different entities. The approaches towards this split can be divided in two main groups. The ones that defend a map and encapsulation scheme and the ones that propose a system based in addresses rewriting.

One of the first proposals around map and encapsulation approaches can be found in [38]. The idea behind these approaches is to deploy a scheme in where packets from a name-space (i.e. identity space) are encapsulated in the other name-space (i.e. location space). For doing such a thing a map between both spaces should be performed. The encapsulation takes place in the border network elements between the two name-spaces and is performed by a adding an outer IP header to the original packet. The packet is routed through the network using the outer name-space header towards its destination within this name-space. There is decapsulated and sent to its ultimate destination in the inner header name-space. Map and encapsulation approaches have several valuable features. They preserve the original source address, do not require changes in the core of the network and work both for IPv4 and IPv6. On the other hand it introduces overhead in the network due to encapsulation overload.

The first proposal of an address rewriting approach is done in [39]. This idea proposes use the lowest 64 bits of a IPv6 address as the identifier of the host, while the highest 64 bits will represent the routing locator (called "Routing Group" here). When a packet leaves its own domain to reach another one the source 64 bits of the identifier will remain unchanged, but the source domain bits will change. Thus this address rewriting is done at the edge routers of the host domain. When the packet reaches its destination domain its source higher address bits are again rewritten by the ingress router, before forward the packet to the final destination.

This work is centred in the Locator/Identifier Separation Protocol (LISP) ([14],[15]) which is one of proposals motivated by the IAB's workshop. LISP is a map-and-encapsulation solution based in network level encapsulation. LISP intends to be simple, modular and incrementally deployable. It does not require changes in the core of the network or in the end-hots. LISP also introduces a Mapping System [16], a distributed database that maps identifiers to locators.

2.2 Mobility

Nowadays people are changing their way to connect to the Internet. Years ago the most common way to access the network was from a wired, fixed and limited connection. Now the society is evolving into something that some authors have defined as "always connected" society [7]. People use their smart-phone or their laptop to get to the Internet through a 3G connection or a Wi-Fi access. People wants to be connected all time regardless of their location. Thus, it is necessary an infrastructure that supports the mobility of the user through different access networks.

LISP with its separation between location and identity offers a perfect scenario to provide mobility. Its own specification of mobility, LISP-MN, offers mobility based on the LISP infrastructure. However, the today *de facto* mobility standard is Mobile IP [22].

Mobile IP protocol appeared as a response to the mobility issues that traditional TCP/IP protocols family had to face up to. Today it is specified for both IPv4 [22] and IPv6 [23] mobile nodes. There are several elements and terminology related to the protocol. The Home Network is the default network of the MN, and the one it is supposed to be when it is not moving, the Home Agent is a special node within this network that serves the MN as a fixed point of anchor to the network, the Home Address is an address which belongs to the Home Network address space and is used by the MN as its main address. When the MN is not in its Home Network, the network it is connected to is called Foreigner Network and the address this network assigns to the MN is named Care of Address. In order to provide connectivity regardless the Foreigner Network the MN is connected to, and to maintain a establish connection during a handover, Mobile IP works as follows. The connections established from, or to, the MN are done using the Home Address of the MN, and is the Home Agent who redirects all the traffic from, or to, the current MN's Care of Address. The MN is reachable wherever it is because its Home Address remains unchanged and the Home Agent always knows its current Care of Address.

2.3 Privacy

Privacy today is a concern for most of the users of the Internet. People are more aware each day of the risks of the use of the net and they want to hide their private information. The normal use of the Internet exposes the user in many ways, so ways to protect him/her are needed.

The greatest research advances performed on the Internet have been done at the application layer, which is the one that allows a better flexibility. Because of that, the main solutions for the privacy problem on the Internet have been developed at this level. In this context, onion routing is one of the most well-known privacy solution at higher layers. Using onion routing, packets travelling through the network have been repeatedly encrypted in their origin, and are layer by layer unencrypted by the routers they go through. This way, routers in the path only know the previous and next hop of the packets. The main idea was originally proposed in [33]. A patent free, improved, and already deployed version is Tor network [34].

Whereas this application level solutions can serve in some cases to mobile nodes, mobility, by its conception, has inherent issues with privacy. Since the most common protocol to provide mobility today on the Internet is Mobile IP, the mobility privacy solutions are developed to this protocol. Many solutions force the use of the Home Agent as a proxy, or a similar proxy mechanism, to avoid revealing the mobile node location ([30], [31] and [32]). Others propose the use of fake Home Address to no disclose the true information to malicious parts [27]. For technical details of these solutions and a comparison with the ideas proposed in this work, see section 7.3.

Chapter 3

Background

3.1 Locator/ID Separation Protocol (LISP)

3.1.1 Overview

The Locator/ID Separation Protocol (LISP) decouples host identity from its location. This separation is achieved by replacing the addresses currently used in the Internet with two separate name-spaces: Endpoint Identifiers (EIDs), and Routing Locators (RLOCs). In order to enable incremental deployment, and to avoid any changes to the application layer, EIDs are syntactically identical to IP addresses: 32 bit (for IPv4) or 128 bit (for IPv6) values that identify the device attached to the network. Host applications bind to the hosts EID for transport layer connections. RLOCs are IPv4 or IPv6 addresses used for routing through transit networks and are to be allocated according to the topology of the network. In order to reach a host, identified by its EID, one must first find the current location (RLOC) of the host. LISP introduces a new level of indirection. It changes the protocol stack to introduce a new level at network layer. More information about the new protocol stack can be found in section 3.1.2.

Within a LISP site, hosts are provided with EIDs (that can be normal IPv4 or IPv6 addresses) that identify them, and can be routed within the site, so no change is needed within the domain. Once a host on a LISP site wants to stablish a communication with another LISP host in another LISP site, it sends a packet from its EID to the destination EID. To exit the LISP domain the packet has to pass through a xTR router (Egress/Ingress Tunnel Router). This xTRs can be just ETR (Egress Tunnel Router) when they just connect from a LISP site to the Internet, or just ITR (Ingress Tunnel Router) when they connect from the Internet to a LISP site. In general, when the same device performs both operations, they are called xTRs.

The xTR routers are special routers that understand LISP and are connected both to LISP site(s) and to the legacy Internet. This xTR routers have assigned a location within the global network, an RLOC. This RLOC can be routed on the legacy Internet, so an xTR can send a packet to another xTR without problem. The EIDs assigned to the hosts within the LISP domain in charge of an xTR are associated with the RLOC of that xTR. When a packet from the LISP site arrives at a xTR, first it checks if it knows the RLOC of the associated xTR of the destination EID. If that is the case, it encapsulates the packet into another IP header and sends it through the legacy Internet to finally arrive at the proper xTR. The destination xTR dencapsulates and routes it to its internal LISP site based on its EID. See 3.1.3 for more details of the packet encapsulation.

If the xTR has no knowledge about the RLOC associated with an EID, it has to query the Mapping System, which stores all the information about RLOC-EID associations. The Mapping System is discussed in 3.1.4

When a host within a LISP site wants to communicate with a normal host on the legacy Internet, or the other way around, a legacy host trying to contact a host in a LISP site, special proxy xTR routers are needed. See section 3.1.5.

In case that a LISP site is behind a NAT, special protocol operations and equipment are needed to correct traverse the NAT in order to contact and be contacted from the outside. Section 3.1.6 extends this.

The figure 3.1 shows an example of the protocol deployed. There are two disjoint LISP domains (*LISP sites*), a Mapping System space (*Mapping System*), and the legacy Internet (*RLOC space*). One of the hosts (*Host 1*) on one of the LISP domains wills to communicate with another host (*Host 2*) in the other LISP domain. It knows the destination EID (*EID_B*), so it sends a packet (1) with this destination and with its own EID (*EID_A*) as the source EID. When this packet arrives at the xTR of the LISP site, the xTR checks this cache to see if it knows which RLOC corresponds to *EID_B*, since it has no entry for this EID, it has to query the Mapping System. It sends a Map-Request (2) to its configured Map Server to ask for this EID. The Map Server routes (3) this query through the Mapping System. The request finally arrive (4) at the xTR in charge of this *EID_B*. This xTR then sends a Map-Reply (5) to the original xTR staying that the *EID_B* can be found on the *RLOC_B*. From now on, the first xTR will know that the *EID_B* can be reached at the *RLOC_B*. The original data packet is now encapsulated using this information. The xTR adds an extra IP header in which the source address is its own RLOC (*RLOC_A*) and the destination address is *RLOC_B*. This encapsulated data packet (6) can be routed through the Internet. When the data packet arrives at the destination xTR it will be dencapsulated by

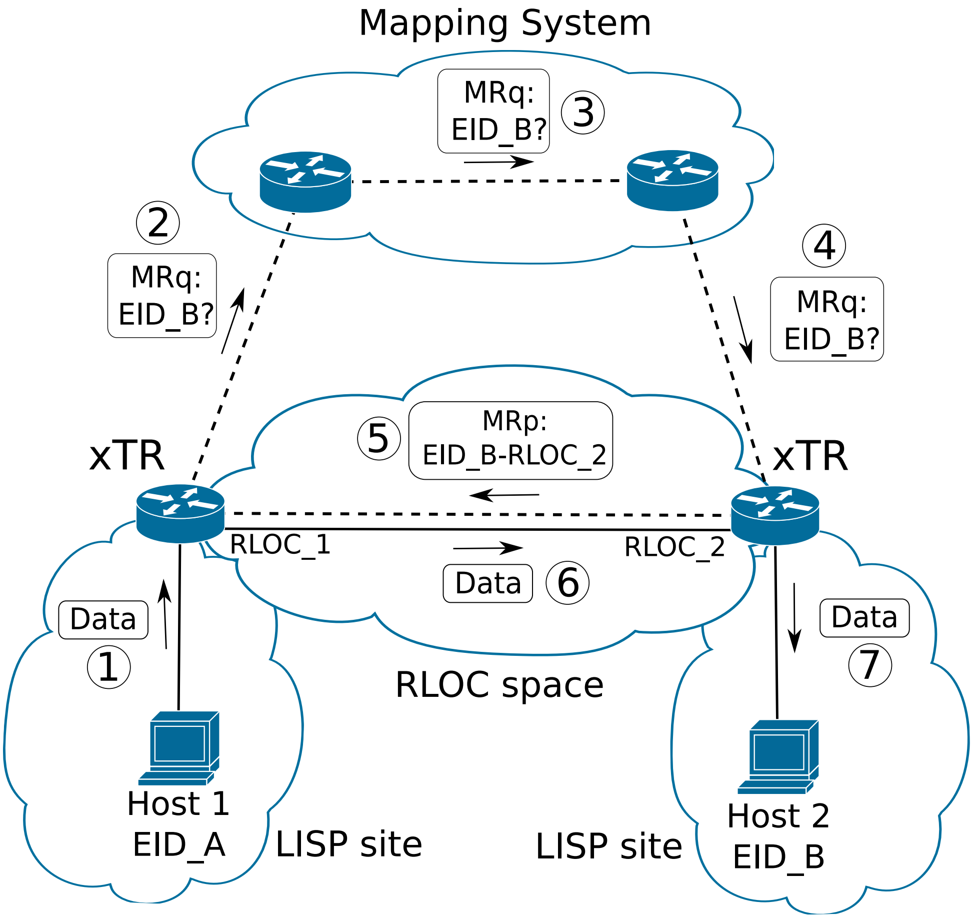


Figure 3.1: **LISP overview**

it and routed (7) to the LISP site without the outer IP header. Within the LISP site the *EID_B* can be routed, so the packet will reach *Host 2*.

3.1.2 Protocol stack

Figure 3.2 shows the classical TCP/IP protocol stack. In contrast, figure 3.3 shows the new stack once we introduce LISP in the model. As it can been seen in the figures, there is a new IP level introduced by LISP. The inner IP level (the one nearer to the application layer) is used by end hosts to communicate between themselves, the IP of this level is the EID. The outer IP address is the RLOC, it servers to route the packet through the global network, once it exits the site domain and access the general Internet. This two IP addresses are related by means of a mapping that is stored in the Mapping System. The encapsulation (i.e. the addition of the external

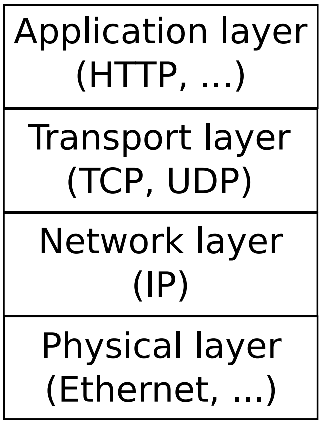


Figure 3.2: **Classical TCP/IP protocol stack**

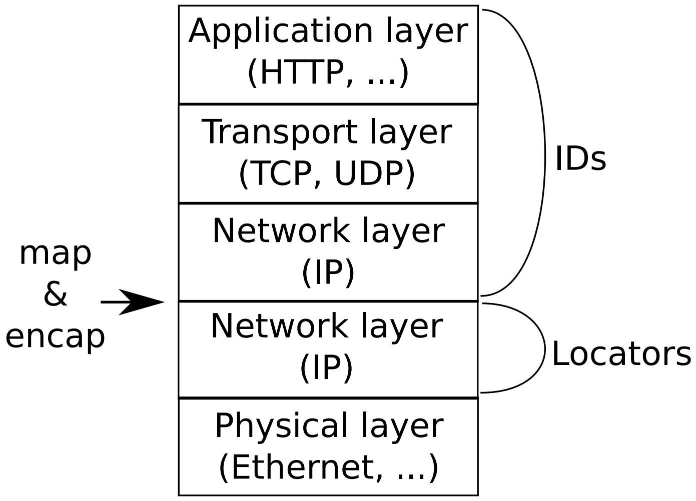


Figure 3.3: **LISP protocol stack**

IP header containing the RLOC) and the decapsulation (i.e. the extraction of this outer header) is performed by the xTRs when the packet leaves or arrives at the LISP site respectively.

3.1.3 Packet encapsulation

Figure 3.4 shows the packet format when an IPv4 packet is lisp-encapsulated in a another IPv4 packet. Just above the original IP header (Inner header, IH) a LISP header is placed. This LISP header contains a minimal amount of information. On top of that an UDP header is added. This UDP header uses well-known ports, assigned to the LISP protocol. The UDP header is

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  / |Version|  IHL  |Type of Service|          Total Length         |
 /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |         Identification        |Flags|      Fragment Offset    |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
OH  |  Time to Live | Protocol = 17 |         Header Checksum       |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                    Source Routing Locator                     |
 \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  \ |                 Destination Routing Locator                   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  / |       Source Port = xxxx      |       Dest Port = 4341        |
UDP +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  \ |           UDP Length          |        UDP Checksum           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
L   |N|L|E|V|I|flags|            Nonce/Map-Version                  |
I \ +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
S / |                 Instance ID/Locator Status Bits               |
P   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  / |Version|  IHL  |Type of Service|          Total Length         |
 /  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |         Identification        |Flags|      Fragment Offset    |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
IH  |  Time to Live |    Protocol   |         Header Checksum       |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                           Source EID                          |
 \  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  \ |                         Destination EID                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.4: **LISP IPv4-in-IPv4 Header Format**

consistent with the TCP/IP scheme and serves to let the packet arrive to the correct LISP-processing applications running on the xTRs and the rest of the LISP equipment deployed in the network. Finally there is another IP header (Outer Header, OH) that contains the RLOC address. The whole packet will be seen in the network as a common TCP/IP packet, so it will pass through the network from a LISP site to another LISP site without problem.

3.1.4 Mapping System

LISP introduces a distributed and publicly accessible Mapping System [16] that is designed to serve the EID-to-RLOC mappings and policy information. The current LISP Mapping System structure is inspired in a previous work published in [17]. When the RLOC associated to an EID changes, an update in the associated Mapping System is needed to maintain reachability at its new location. This update is done through a signalling packet named Map-Register. The signalling packet that queries the Mapping System for the EID-to-RLOC mapping is called Map-Request. The reply packet to this, which contains the current mapping, is known as Map-Reply. The nodes that give access to this Mapping System are named Map Servers.

3.1.5 Proxy Tunnel Routers (PxTR)

LISP uses proxy tunnel routers (PxTR) to interoperate with legacy Internet. These proxies encapsulate or decapsulate, as needed, LISP packets to allow LISP hosts to communicate with non-LISP hosts. When a LISP node discovers that an IP address is not an EID but a traditional IP address, it send the encapsulated packet, with this address in the inner header, to its assigned Proxy-ETR. The Proxy-ETR router serves as a point of entry to the general Internet where the non-LISP host are located. It decapsulates the packet and forwards a normal IP packet to the Internet. When a non-LISP host tries to communicate with a LISP host their packets travel through a Proxy-ITR. This Proxy-ITR have a set of EIDs assigned, and announces them highly aggregated over the Internet, so packets going to that destinations will be absorbed by it. PITRs serve as point of entry to LISP sites. Further details of these PxTRs can be found in [21].

3.1.6 NAT traversal

The RTR (Re-encapsulating Tunnel Router) is a recently developed LISP element that provides NAT [24] traversal capabilities to LISP. Basically it works as follows: A LISP xTR (or a MN) requests a list of available RTRs from its Map Server. If the xTR is behind a NAT-box, it registers to a RTR, configures a tunnel towards it, and then the RTR acts basically as a proxy. The RTR registers the EID of the xTR on its behalf to attract all the traffic sent/received by the xTR and forward it through the above-mentioned tunnel. A pre-configured shared key is required in order to authenticate all the process. The RTR performs all the operations of port translation required. More details about LISP NAT traversal can be found in [20].

Figure 3.5 shows how a RTR works, in that case serving the mobile node name *MN 1* (it could be a generic xTR), see section 3.2 for more details of LISP-MN. Once RTR has been associated with the MN, any Map-Request from a third party requesting the RLOC of the MN will be redirected through the Map Server to the RTR. It replies to the originator of the Map-Request with its own RLOC. With that, all data packets generated at the peer will travel to the RTR which will redirect them to the MN behind the NAT.

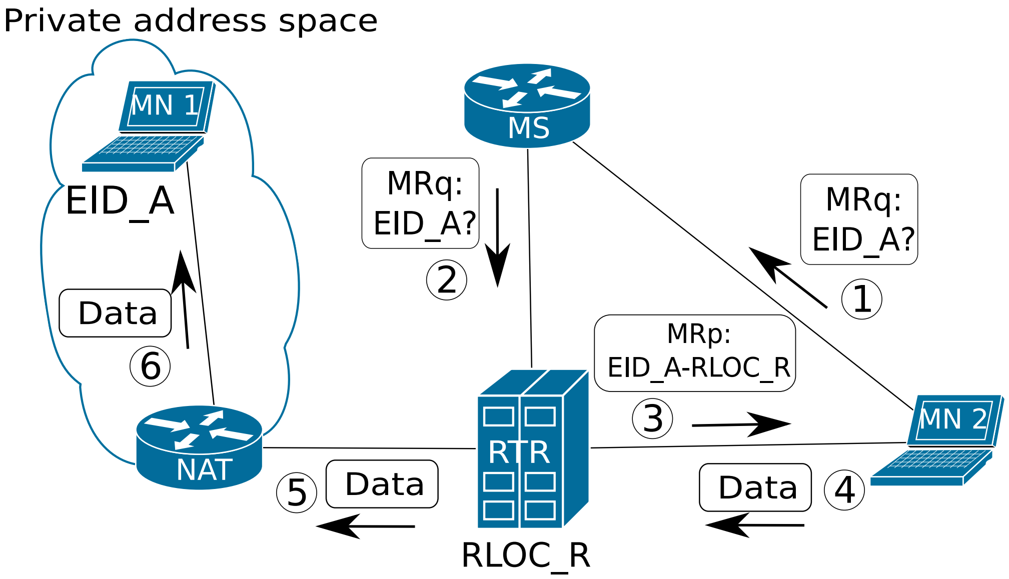


Figure 3.5: **Re-encapsulating Tunnel Router**

3.2 LISP Mobile Node (LISP-MN)

LISP Mobile Node (LISP-MN) leverages LISP features to build a mobility architecture and protocol based on it. In LISP-MN a Mobile Node (MN) is typically statically provisioned with an EID that is used for all its connections. This allows the applications to bind to a static address. The current point of attachment to the network defines the current RLOC for the MN. The location of the host can change several times during an ongoing connection without breaking the transport layer connection. When the hosts location (RLOC) changes, the LISP-MN will encapsulate the packets towards the new RLOC. This is done thanks to the LISP Mapping System which always has the latest RLOC for the MN's EID.

LISP Mapping System acts as a location management system. But unlike in traditional mobility protocols, such as Mobile IP [22], LISP's Mapping System is distributed and federated. Mobile IP location management system (the Home Agent) is deployed at the mobile clients service provider. Interestingly, LISP Mapping System avoids mobile service provider lock-in.

Once the set of RLOCs associated to an EID prefix is discovered, packets with network layer headers from the EID namespace are encapsulated in a second header from the RLOC space, and are routed towards the destination. Upon reception at the destination site, the LISP header is removed before delivering the packet to the end-host. LISP introduces special gateway routers (xTRs) that perform the LISP encapsulation and decapsulation at each sites ingress and egress points. This is done in LISP-MN by a lightweight LISP xTR implemented in the MN. Packets - except for management protocols such as DHCP [28] - are LISP encapsulated by the lightweight LISP tunnel at the mobile node, and routed based on the RLOCs to the destination site. Mobile node tunnel routers remove also the LISP header from incoming packets before sending them to upper layers to ultimately reaching the destination application.

Figure 3.6 shows the basic operation of LISP-MN. The MN wants to communicate with its peer, from which only knows its EID. It sends (1) a Map-Request (MRq) to obtain the RLOC of the peer. This MRq is routed (2) through the Mapping System to finally reach (3) the Tunnel Router (xTR) of the LISP site where the peer is. The xTR replies (4) to the MN with its RLOC in a Map-Reply message (MRp). Finally the MN sends (5) the data to the xTR which forwards (6) it to the peer.

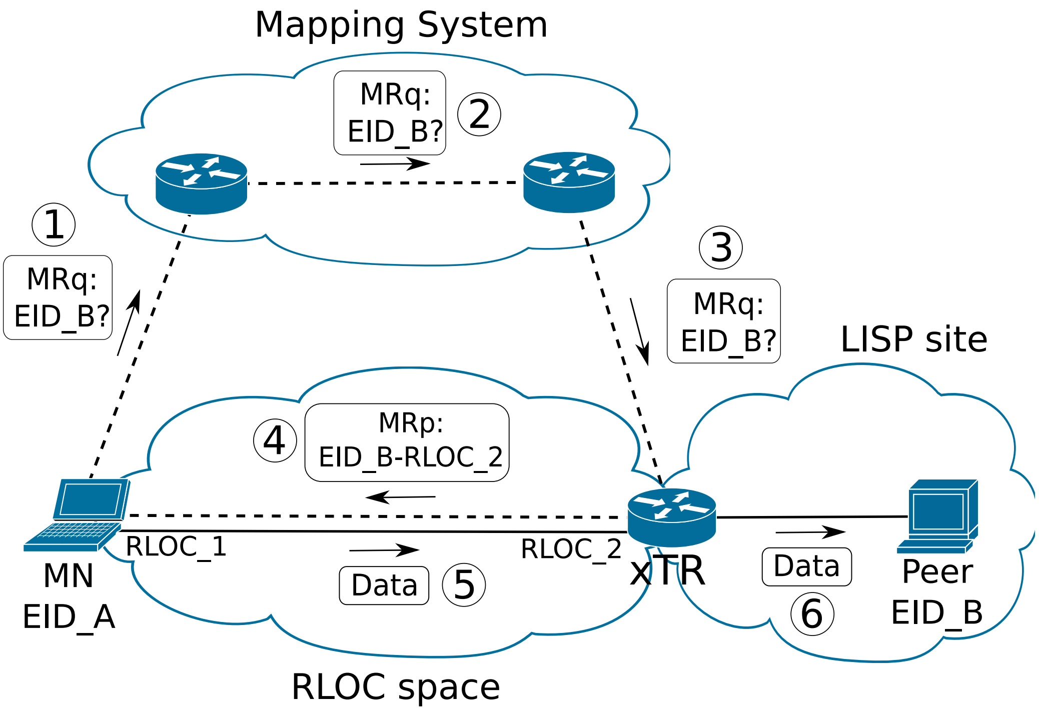


Figure 3.6: **LISP-MN Overview.**

Chapter 4

Problem statement

At its very fundamentals semantics, privacy is defined as *a state in which one is not observed or disturbed by others* [40]. Extrapolating this to privacy on networking, the goal is to prevent others from tracking the users' personal information both in terms of from where they connect and to where they connect.

4.1 Scope of the problem

It is worth to note that privacy on the Internet is not just related to network level. It is an issue that impacts all the layers. The purpose of this work is to provide privacy for LISP-MN, which is a network level protocol, so it is focused on network level privacy.

This work assumes that the appropriate mechanisms to protect the user's privacy are both deployed at lower and upper layers (i.e. using encryption to protect the data sent, to avoid onlookers over the path) and focus only on network-layer privacy

4.2 Location privacy

Since mobility protocols typically use addresses to locate users, they rise privacy concerns, and in this context LISP-MN is not an exception. An attacker could learn the (approximate) physical location of a user by monitoring its locator address, for instance by using IP geo-localization techniques ([8] and [9]).

This issue is exacerbated in LISP-MN when compared to other mobility protocols, such as Mobile IP [22]. In Mobile IP an attacker has to establish a connection with the mobile node to learn its location, this way a mobile node can reject inbound connections from untrusted peers. However, in LISP-MN an attacker has just to query the (publicly accessible) LISP Mapping System to learn the location (RLOC) of a user, which is beyond its control.

It is needed to provide the MN a way to ensure that, if it wants to, it can operate in a way that prevents an attacker from knowing its current location. A complete location privacy mechanism for the MN means that this attacker can not know the MN location queering the Mapping System, neither it can by means of establish a communication to the MN.

4.3 Identity privacy

In addition to location privacy, anonymity is an increasing concern as well for a subset of today's Internet users. A good example is the recent discovery of the CarrierIQ application, which comes pre-installed in many smartphone devices and monitors several aspects of the user's behaviour [10].

As a result of these concerns, users are demanding mechanism that can guarantee their online anonymity. For instance some popular web browsers include a *private browsing mode*, where tracking cookies have the lifetime of a single browsing session, and a "Do Not Track" option to opt-out from advertising network behavioural tracking.

However, a LISP-MN host still discloses its unique EID even in this browser operating mode, making EID based tracking possible. Given the fact that an EID can be easily associated to the user's identity (e.g., his/her mobile phone number), some users may not want to reveal it in order to protect their anonymity. The MN should be capable of establish connections to whoever it wants while preventing these third parties to know its identity.

Chapter 5

Preliminary solutions

This chapter aims to describe all the preliminary solutions that were designed prior to the design of the ultimate ones. The solutions exposed in this chapter fulfil the requirements to solve some of the privacy issues exposed in chapter 4. If they are not proposed as final solutions is because they suffer of excessive complexity or they mean an important performance drop.

Even so, they are described here since they serve as alternative approaches to deal with the problem exposed and can suit in some concrete scenarios. Moreover the proposal exposed in section 5.2 provides not only a privacy solution, it also gives further advantages.

5.1 PxTR proxies

Traditionally, to provide location privacy, a proxy-like solution is the most usual approach. In the LISP environment, already exist network elements that perform proxy operation, the Proxy xTRs. The PxTRs serve LISP hosts to reach hosts outside the LISP domain, in the traditional Internet, or the other way around. This proxies can be used by a mobile node to hide its location. A malicious node can know the location of the proxy, but no the location of the MN itself.

To force the use of these PxTRs, the MN should perform some *tricks* to make the system believe that its EID is a not a LISP EID, that it is a classical IP address from the legacy Internet. The idea is that packets addressed to the MN will exit the LISP domain through a PETR proxy and travel within the legacy Internet to a PITR that let them enter again in a LISP site and finally reach the MN.

To do so, the MN has to tell the Mapping System that it does not want to be directly reachable. It also needs to register with a PITR that allows it to receive incoming traffic. In other words, it tells the Mapping System that the IP that it uses as an EID is not a LISP EID, so the Mapping System will reply all the queries against this EID with a negative reply. The querier will understand that this address can not be reached within a LISP domain and hence, it has to send the packets to the legacy Internet through a PETR. Once in the Internet, the PITR will attract these packets due that it announces the MN EID as an IP address that can be reached through it. It is important to note that this PITR will know the real location of the MN, so the PITR chosen by the MN to announce its EID over the Internet should be one that it trusts.

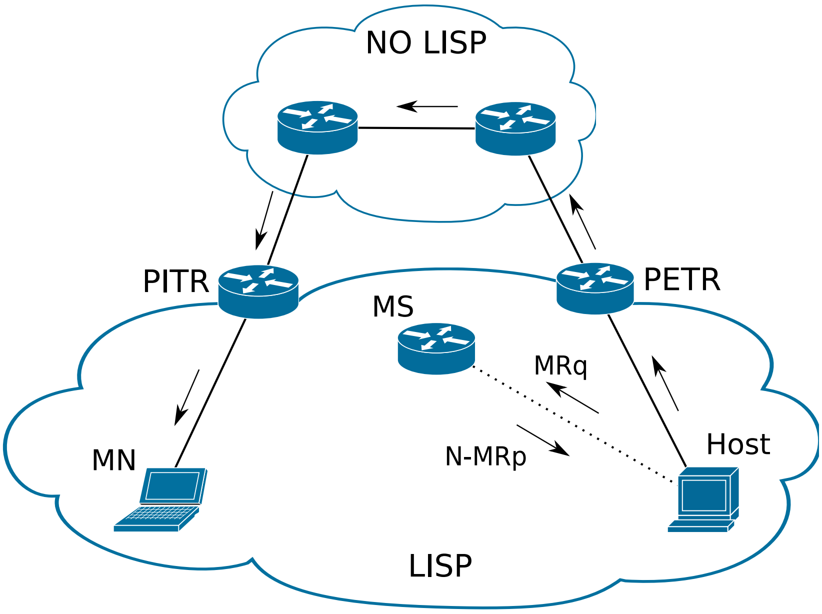


Figure 5.1: **PxTR solution**

Figure 5.1 shows this *trick* in place. The MN has arranged everything to force third nodes use PxTRs to contact with it. A host wants to communicate to the MN so it queries a Map Server (MS) to know the locator associated with the EID of the MN. The MS replies with a Negative Map-Reply that means that this address is not a LISP EID. Knowing that, the host sends its packets to its PETR which forwards them to the non-LISP space, the legacy Internet. These packets travel across the non-LISP domain until they arrive to the PITR in charge of the MN's EID. This PITR takes the packets and forwards them to the locator of the MN which finally receives the packets.

The case when it is the MN which is sending packets is analogue to this one. Since the MN already wants to send the packets through these PxTRs, it directly send the packets to its PETR, and they will arrive to the other node in the same way.

The problems that this solution confronts are that with it the MNs are fundamentally loosing all the LISP advantages since they are forcing a non-LISP way of operation. Not only they are loosing all the LISP features, but they are also suffering from a high increment in latency since now the packets have to pass through two different proxies.

5.2 EID Sharing

This solution comes from the idea of having multiple MNs, each one with its own RLOC, but all of them sharing the same EID. It is a similar concept of today's NAT. With NAT, several identities share a single location, with EID Sharing several locations share a single identity. To achieve that, a similar NAT-like proxy is needed to allow MNs to share identity.

This provide both location and identity privacy, since the location exposed is the one of the proxy, and an attacker can not know the identity of the MN since several ones are sharing the same. The concrete MN behind a connection can not been told apart from the other ones. The mechanism proposed in this section not only provide privacy protection. Thanks to that reuse of identifiers, it also serve to face other problems of the today Internet. The most clear and urgent one is the IPv4 address space depletion. Using a single IPv4 identifier with many IPv6 locators could serve to mitigate the transition to the IPv6 Internet. An ISP willing to deploy IPv4 native connectivity, can use a shared IPv4 EID while using IPv6 for the RLOCs.

The details of how this system works are as follows. It is needed to distinguish between the nodes sharing the same EID address, so this shared address should be *extended* to make each node unique and distinguishable. Transport layer ports could be used for this task they same way they are used in traditional NAT. It is also needed a device that stores this association between ports and nodes. For this purpose the RTR LISP element can be extended and used.

	Many Identifiers	Single Identifier
Many Locators	EID+RLOC (Normal)	EID[port] + RLOC (EID Sharing)
Single Locator	EID + RLOC[port] (NAT)	EID[port] + RLOC[port] (NAT + EID Sharing)

Table 5.1: **Locator-Identifiers relation**

Table 5.1 shows the relation of single/many locators with single/many identifiers, with the name that the relationship takes in each case, and where ports should be used in order make distinctions.

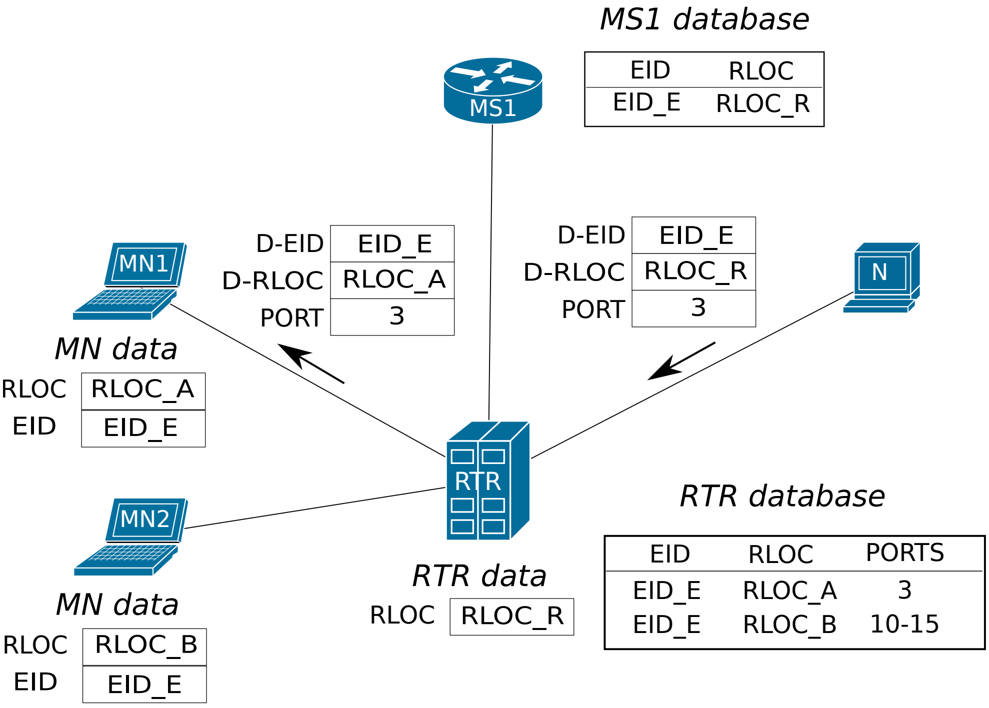
With EID Sharing, the first thing a MN should do is to ask its Map Server if there is need for use EID Sharing. If that is the case, the Map Server will reply telling the MN the location of its assigned RTR. From now on, each packet the MN sends will be addressed to its RTR. The MN will use the RLOC of the RTR as the destination RLOC in the outer header. The RTR will store the RLOC and port the MN is using. With that, the RTR will form a packet with its own RLOC as source locator. The shared EID remains as the source EID. The destination EID will still be the destination EID chosen by the MN, but the destination RLOC is now the appropriate one for this EID. The RTR may have to query the Mapping System to obtain this RLOC. Finally the RTR forwards the packet to its destination.

At some point the RTR has to have been registered in the Mapping System the shared EID as its own. So the RLOC of the RTR is now the one associated publicly with the shared EID. The reply packets will be addressed to the RTR which will re-encapsulate them changing the RLOC to the proper one before sending them to the MN.

This is the normal operation of the system, but it is only valid in the case the MN is only a client and does not receive incoming connection. If that is not the case, and the MN is acting like a server, then prior configuration in the RTR is needed. The MN should send a configuration packet to the RTR specifying the range of ports it want to reserve to itself. Those are the ones where incoming connections are expected to appear. With that, the RTR will know to which one of the RLOCs it has associated with a certain EID, should send the incoming packets addressed to that ports.

Figure 5.2 illustrates a scenario with this mechanism in place. Two MNs (*MN1* and *MN2*) share the same EID (*EID_E*) but have different RLOCs (*RLOC_A* and *RLOC_B* respectively). In the stage shown in the figure both MNs has already store ports in the RTR, either for have explicitly reserved them in a previous moment, of for have established any connection using them. The RTR has also registered the EID of these MNs as its own on the Mapping System. The figure shows how a third node sends a packet to the *MN1*. This packet arrives at the RTR since the EID of *MN1* is registered with the RLOC of the RTR. The RTR receives these packets and checks its database for the port it is using. Once knowing to which MN the packet is addressed, it modifies the destination RLOC to match the one of the proper MN, and forwards the packet.

It is worth to note that with this mechanism, IP packets, at EID level, are unmodified. In some ways it could be seen as a ”clean” NAT. A NAT in which



MS1 database

EID	RLOC
EID_E	RLOC_R

D-EID	EID_E
D-RLOC	RLOC_A
PORT	3

D-EID	EID_E
D-RLOC	RLOC_R
PORT	3

MN data

RLOC	RLOC_A
EID	EID_E

RTR database

EID	RLOC	PORTS
EID_E	RLOC_A	3
EID_E	RLOC_B	10-15

RTR data

RLOC	RLOC_R

MN data

RLOC	RLOC_B
EID	EID_E

Figure 5.2: **EID Sharing solution**

protocols such IPsec still work. Despite this solution contribute with several desirable features, it is too complex for just provide privacy protection, and moreover it suffer of several disadvantages. One of them is that it is necessary to reserve ports before using them to accept incoming connections. Another constriction is that all the MNs sharing the same EID are tied to the same RTR. Finally the limitation of not being possible for two different MNs to use the same port at the same time is a severe drawback. However, it would be possible to extend this idea to use NAT-like port translation to avoid this problem.

Chapter 6

Privacy on LISP-MN

This section describes the proposed extensions to provide both location and identity privacy to the MN. Although two different extensions that address these issues independently are presented, both solutions can be combined to provide full privacy to LISP-MN.

6.1 Location privacy

Location privacy is a well-known problem in mobility and the most common solution is to use a proxy. This way the proxy forwards the traffic to the MN while only its locator is exposed. In this context the MN trusts the proxy, since it is the only one which knows the real location of the node.

In order to offer location privacy for LISP-MN this work takes advantage of the RTR proxy. Since this proxy attracts all the traffic from/to the LISP-MN, even for trusted peers, it introduces an inefficient routing path that degrades the performance of the LISP-MN communications. To solve this issue a minor extension is needed to the RTR. With this extension, the RTR forwards all the Map-Requests towards the LISP-MN, then the LISP-MN can reply with its own RLOC if the source of the Map-request is trusted, or with the RLOC of the RTR if the peer is not trusted. How this can be implemented is discussed in section 7.4.

Here the business model is simple. A company interested in offering location privacy to its costumers can deploy a set of RTRs on the Internet. In order to access the RTR the LISP-MN requires a pre-shared key, in a similar way it needs one to register to its Map Server [19]. This pre-shared key that grants access to the RTR can be used to enforce that the client is paying for the service. The RTRs of the company should be also configured with the EID of the subscribers for authentication. The company has incentives to deploy more RTRs, and more importantly, with a good global coverage. This will reduce the routing inefficiencies of private communications and thus, the subscribers will receive a better service. With this in mind, the company that invests in more well placed RTRs will be more competitive. Finally it is worth to note that Mapping Service providers and such companies are orthogonal and do not compete.

6.2 Identity privacy

This section extends LISP-MN to offer identity privacy, the main purpose being to hide the EID to untrusted peers. The proposed solution here is to use temporary identifiers rather than the real one. It is important to note that identity privacy can only be offered when the MN initiates the connection.

Specifically this section proposes two different approaches to provide such temporary identifiers. The first one is a MN-driven infrastructure-less solution intended for organizations that are willing to manage their own identity privacy. The second one requires deploying a new entity and it is intended for companies willing to sell this service to third-party organizations/LISP-MNs. Again, section 7.4 discusses how to implement both solutions along with how to use such temporary identifiers.

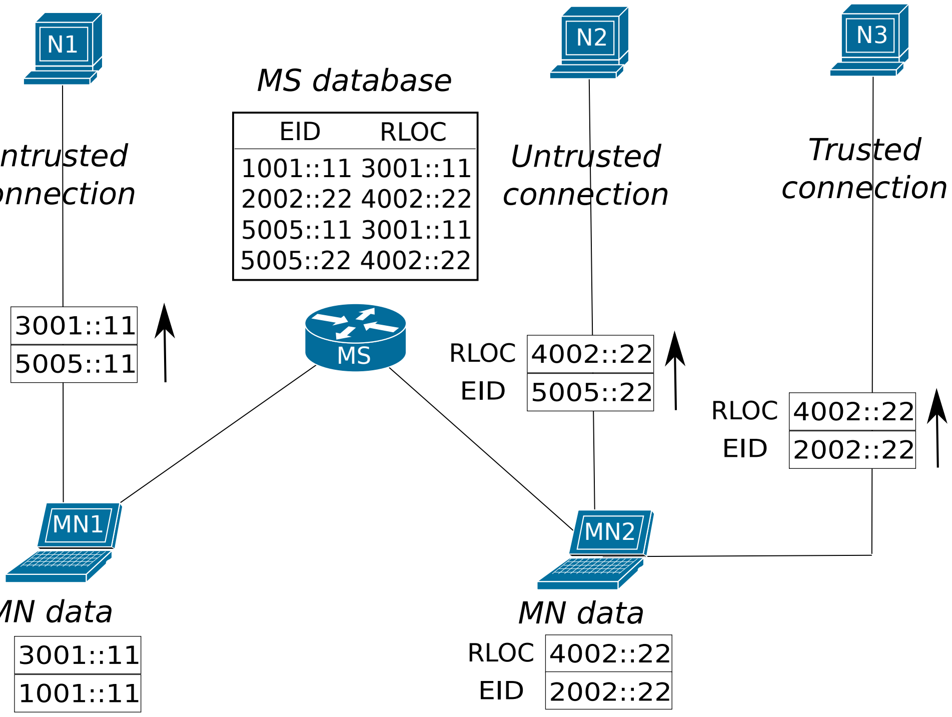
6.2.1 Infrastructure-less proposal

This section describes a solution to provide MN-generated temporary EIDs (tEIDs). This solution takes advantage of the IPv6 address format and its least significant 64 bits which can be auto-configured. This idea has been (similarly) applied to plain IPv6 before (see [27] for further details). It is worth to note that this solution cannot be applied to IPv4 due to its limited address space.

The main idea behind this proposal is that a set of MNs that are sharing the same IPv6 prefix and hence, are being served by the same Map Server, can auto-generate different temporary addresses to use as EIDs. Each of these tEIDs will be under the same prefix. This way, even if an attacker can track this prefix, it cannot track individual nodes. The mechanism is more efficient as the number of MNs sharing the same prefix increases.

In order to generate the above-mentioned tEIDs, this work borrows the mechanisms described in [27]. By means of a hash algorithm, the MN generates a random set of bits to fill the least significant 64 bits of a given prefix. Then the MN queries its auto-generated temporary address from its

Map Server. This is done to avoid collisions: If the address is already in use by another MN the Map Server replies positively and then the MN will generate another address and query again. If the address is not in use the Map Server replies with a negative Map-Reply and finally the MN registers (Map-Register) the EID.



EID	RLOC
1001::11	3001::11
2002::22	4002::22
5005::11	3001::11
5005::22	4002::22

Figure 6.1: **MN generated temporary EIDs.**

Figure 6.1 shows an example of the proposal. The MNs are sharing the prefix 5005::/64 to generate temporal EIDs. The last 64 bits of the addresses belonging to that prefix are generated by the MN. They register these generated addresses in the Mapping System, and use them to establish connection to not trusted nodes.

With this architecture, a misbehaved node could attempt to deplete the available pool of tEID addresses by registering as many as possible. Alternatively, it could also take over a tEID (and hijack its traffic) that is in use by another node, by simply registering that tEID. To avoid this, the Map Server stores a list of authorized users for each tEID prefix, while still using the existing security association (the pre-shared key for their real EID) to authenticate each individual node. To avoid the performance penalty at the Map Server caused by searching for the pre-shared key in the Map-Register message, the MN should include its real EID as well in that Map-Register message. Avoiding traffic hijacking can be achieved by requiring explicit dropping of a tEID in use by the previous owner. Only after that can the tEID be registered by a different node.

6.2.2 Infrastructure dependent proposal

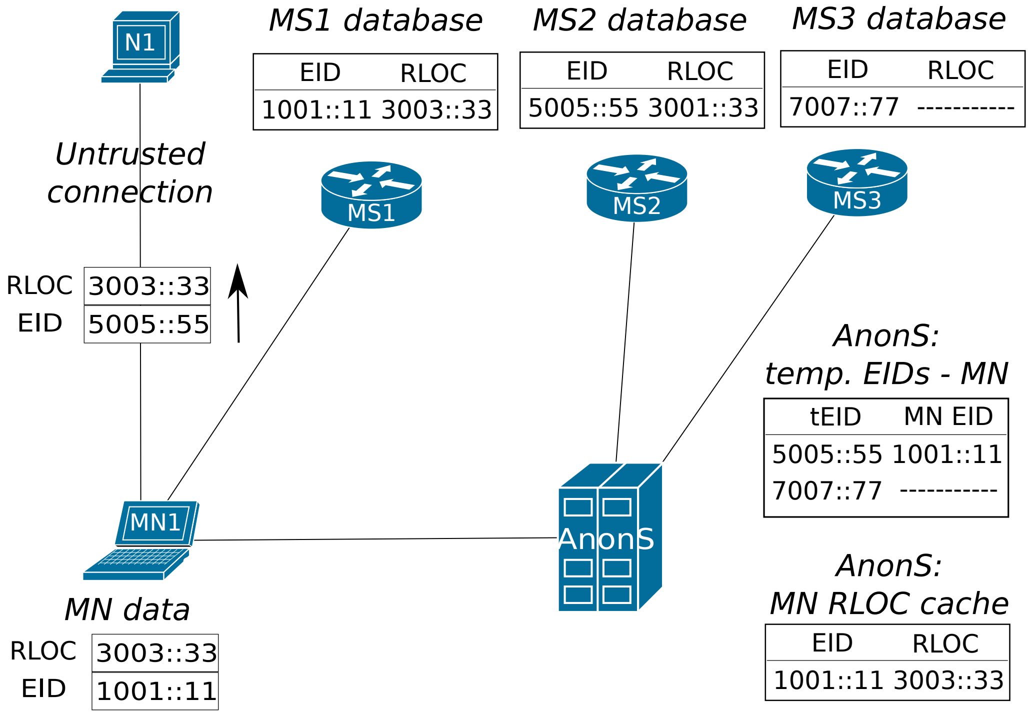
This approach introduces a new element in the LISP infrastructure, the "Anonymity Server" (AnonS). Its function is similar to that of a DCHP server [28], handing out tEIDs on demand to the MNs which request them.

This AnonS can register tEIDs (update the EID-to-RLOC binding) to one or several Map Servers. The key point is that this AnonS does not register its own RLOC for the tEID, rather it registers the MN's RLOC, and hands out a lease on the use of the registered tEID to the MN. The AnonS is responsible for updating the EID-RLOC association for the tEID when necessary. The complete mechanism works as follows.

A MN wants a tEID, so it sends a request to the AnonS telling it its real EID and its current location. The AnonS stores this information and assigns a tEID from the available pool to the MN. Then the AnonS registers this tEID to the Map Server responsible for the covering prefix, with the RLOC data of the MN. When this process is completed, the AnonS notifies the MN that it can start using the tEID. When the MN wants a new address, it only has to ask the AnonS for a new one. When the MN roams, it notifies both the Map Server responsible for its real EID, and the AnonS, if a tEID is in use. Finally, the approach is secured similar to the usual Map Server registration: authentication data is associated to each tEID request. This data is based on pre-shared keys stored both at the MN and the AnonS, and is generated as in the Map Server case (see [19]).

In figure 6.2 an illustration of the solution is shown. The AnonS keeps a database of its EIDs (5005::55 and 7007::77) and to whom they have been assigned (5005::55 assigned to the MN1 with EID 1001:11). It also keeps record of the last known position of all the MNs using its EIDs (MN1 last RLOC is 3003::33). The AnonS EIDs can belong to different prefixes and Map Servers (5005::55 belongs to Map Server 2 and 7007::77 belongs to Map Server 3).

New messages This new infrastructure requires new messages to support the new control plane operations. These messages are no more than just extended Map-Register and Map-Notify. The original Map-Register and Map-Notify structure and fields can be checked in sections 6.1.6 and 6.1.7 respectively of the main LISP draft [14]. The new messages add an extra



MS1 database

EID	RLOC
1001::11	3003::33

MS2 database

EID	RLOC
5005::55	3001::33

MS3 database

EID	RLOC
7007::77	-----------

RLOC	3003::33
EID	5005::55

AnonS: temp. EIDs - MN

tEID	MN EID
5005::55	1001::11
7007::77	-----------

MN data

RLOC	3003::33
EID	1001::11

AnonS: MN RLOC cache

EID	RLOC
1001::11	3003::33

Figure 6.2: **Anonymity Server**.

field at the end of the original messages and some control flags at the very begging of the header.

The Map-Register with the extra field and flags is called tEID Request message (See figure 6.3), whereas the extended Map-Notify is called tEID Reply (Figure 6.4). The new LISP control message type numbers assigned to them are 9 for tEID request and 10 for tEID Reply. Please note that this assigned numbers are just provisional and may be subject to change in the future. Both messages share the tEID field at the end, which corresponds with the last 96 bits. This tEID field is just required when performing some operations, and may no appear in all cases. Both of them also use a new bit just after the message type field. This new E bit is used to indicate if there is a tEID field at the end or not. In the case of tEID Request, the P bit is removed from the original Map-Register message structure. The tEAct field stands for tEID Action, and is only present in the tEID Request message. This field tells the AnonS the exact meaning of this tEID Request packet, i.e. what it is supposed to do in response to it.

The actions performed by the tEID Request message are:

- Request tEID: This action is performed when the MN sends the tEID

Request message as an explicit request, asking for a temporal EID. In this case, the E bit is set to 0, since there is no tEID field attached.

- Drop tEID: Once the MN is done with the tEID, it has to tell the AnonS that it will no longer use that tEID. This is done by means of a tEID Request message that specifies that the tEID attached can be used by another MN. To do so the E bit is set to 1, and a tEID field containing the intended-to-drop tEID appears at the end of the message.

- Renew tEID: Before the time-out for a specific tEID expires, if the MN wants to still use it, it has to inform the AnonS. To do so, it has to send a tEID Request message indicating the tEID it is using, and it wills to continue use. In order to do that, the E bit is 1, since there is a tEID containing the to-be-renewed tEID.

- Update RLOCs: Every time a tEID Request is sent, regardless its meaning or operation, the associated MN can use it to update its current RLOC information in the AnonS. As in a normal Map-Register, each tEID Request message contains the fields to send the RLOC information, if desired.

The tEID Action field codes associated with each of the actions that the tEID Request message can perform are the following:

- 000: Request

- 001: Renew

- 010: Drop

There is no action code associated with the update RLOCs operation, since this operation can be performed at any time a tEID Request message is sent. It does not matter which action it is supposed to serve to, the RLOC data can be sent anyway.

Regarding the tEID Reply messages, the only operation that they are supposed to do is the following:

- Send tEID: When an AnonS receives a tEID Request message with the tEID Action code 000 it knows that the sender MN is asking for a tEID. This tEID is sent to the MN in a tEID Reply message, which only servers this purpose. There is no further action associated with the tEID Reply message.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Type= 9|E|tEAct|      Reserved               |M| Record Count  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Nonce . . .                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         . . . Nonce                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Key ID             |  Authentication Data Length   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                     Authentication Data                       ~
+-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                          Record  TTL                          |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
R   | Locator Count | EID mask-len  | ACT |A|      Reserved         |
e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
c   | Rsvd  |  Map-Version Number   |        EID-prefix-AFI         |
o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
r   |                          EID-prefix                           |
d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
| L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| o |        Unused Flags     |L|p|R|           Loc-AFI             |
| c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  \|                             Locator                           |
+-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           tEID  TTL                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Req-AFI             |           tEID-AFI            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                             tEID                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6.3: **tEID Request message**

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Type=10|E|            Reserved                 | Record Count  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         Nonce . . .                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         . . . Nonce                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |            Key ID             |  Authentication Data Length   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    ~                     Authentication Data                       ~
+-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   |                          Record  TTL                          |
|   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
R   | Locator Count | EID mask-len  | ACT |A|      Reserved         |
e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
c   | Rsvd  |  Map-Version Number   |        EID-prefix-AFI         |
o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
r   |                          EID-prefix                           |
d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
| L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| o |        Unused Flags     |L|p|R|           Loc-AFI             |
| c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  \|                             Locator                           |
+-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           tEID  TTL                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |          Unused Flags         |           tEID-AFI            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                             tEID                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6.4: **tEID Reply message**

Chapter 7

Discussion

7.1 Evaluation of proposed solutions

In terms of evaluation, in the proposed location privacy solution, like in every proxy solution, the use of another element in the path increases end-to-end latency. Specially when the RTR is located far from the optimal end-to-end path. On the other hand, additional signalling messages are needed to use the RTR. Those are the same signalling messages that are needed to perform NAT traversal [20].

The identity privacy solutions also requires additional signalling messages. The infrastructure-less approach only requires of an additional Map-Request (before sending the Map-Register) than the normal MN operation. The infrastructure-dependant doubles the number of signalling messages due that first it is the MN who registers to the AnonS, and then is the AnonS who registers to a Map Server.

The infrastructure-less solution for identity privacy can be used without additional cost in a trusted network. The nodes simply share an EID prefix for temporary address usage, and achieve identity privacy this way. This can be used by companies which owns a prefix and shared it between MN of their property. If any of the MN sharing a prefix does not belong to a domain under the company control or trustiest, then presence of misbehaved nodes should be assumed. When that is the case, there is an opportunity to sell an authentication service to the entities operating the mobile nodes. Registration is only allowed to paying customers, and a tiered service can be offered based on an anonymity quality metric defined by the provider(e.g., nodes allowed per prefix, prefix size, etc.).

There is a key issue that makes the AnonS specially attractive as an identity privacy mechanism and distinguishes it from the solution presented in the previous section. The MN can use as many addresses, even from disjoint prefixes, as it wants. As a result, an attacker tracking tEIDs will have difficulties to correlate them to a single MN. An anonymity server can work with IPv4, IPv6 or both address families. In contrast to the infrastructure-less approach, using an AnonS is a viable solution for IPv4 temporary EIDs, because it optimizes address usage, in the face of the IPv4 address shortage.

Deploying an AnonS generates revenue for its operator, which controls the access to the identity privacy service. At signup the client MN is configured in the AnonS, and a pre-shared key is stored in both entities. Pricing can be made dependent on several factors, such as the number of distinct tEIDs requested over a period, their lease time, etc. Additionally, increasing Map Server diversity by acquiring several (t)EID prefixes registered to different servers is another price differentiator, or a means to rise above competition.

7.2 Temporal addresses collision probability

The strength of the idea proposed in 6.2.1 is due to the high space that the IPv6 suffix offers, where the probability of a collision of two random generated temporal suffixes is low. It is worth to calculate how low this probability is. It seems logical that, as the pigeon-hole principle stands [36], if more than 2^{64} suffixes are generated, at least two are going to collide for sure. If only one suffix is generated the probability of collision is zero. To determine the collision probability for and arbitrary number of suffixes generated, this work uses a birthday paradox-like approach [35]. Calculate the probability of having a collision between two or more suffixes is the same that calculating the opposite probability of that such event does not occur (i.e. with n generated suffixes there is no collision, all are different). Equation 7.1 shows this model.

$$p(n) = 1 - \bar{p}(n) = 1 - \frac{n! \cdot \binom{N}{n}}{N^n} \tag{7.1}$$

Since this is not easily computable for a 2^{64} space, in order to obtain a result an approximation is used. This approximation gives a good result, and for the purposes of this work is accurate enough. The approximation followed is expressed in equation 7.2.

$$p(n) = 1 - \bar{p}(n) \approx 1 - e^{\frac{-n(n-1)}{(2 \times N)}}. \tag{7.2}$$

With this approximation, it can be computed the probability for the full space of possible number of suffixes and see up to which point the collision probability is lower enough to let the system work properly.

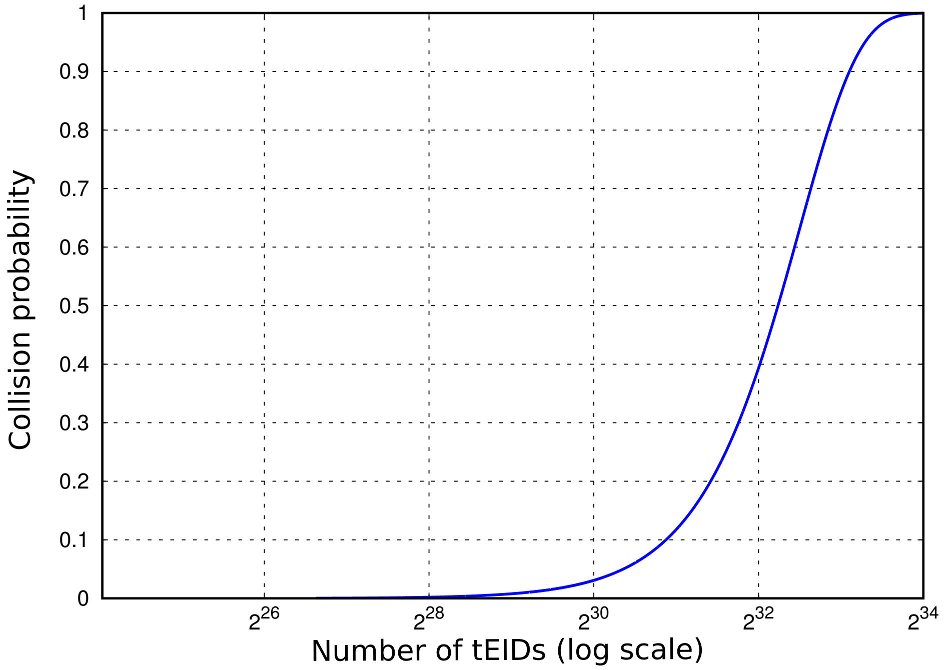


Figure 7.1: **Collision probability**

Figure 7.1 shows the plot defined by the equation 7.2. Please note that probability is always greater than 0 and never reaches 1 (except for just one tEID or for 2^{64} tEIDs respectively), despite what can be inferred from the detail level of the image. The plot shows how up to 2^{28} the collision probability is almost zero. That shows that the address space of the IPv6 suffixes is large enough to allow the system to work.

7.3 Comparison with existing approaches

Location privacy in mobility is a well-known issue which Mobile IP has faced up in [25]. In particular they use a similar approach as the one presented in 6.1 to solve it, in this case the Home Agent acts as the proxy. Similar approaches have been followed by the authors of [30], [31] and [32]. For instance in [32], the authors propose to deploy a new entity (Tunneling Agents) which represent on-demand proxies that are chosen by the Home Agent to offer location privacy to the nodes. Finally, a different approach to location identity has been proposed in [26] where the authors propose to extend Mobile IP to use IPv6 ”pseudo home addresses”. These addresses are generated in a similar way the ones proposed in [27] are generated for identity privacy.

Although these ”pseudo home addresses” are intended to provide location privacy, they implicitly serve as an identity privacy mechanism.

With respect to identity privacy, the authors of [27] propose a similar solution to the one presented in this work, particularly in section 6.2. This proposal forces to keep one temporal identifier per connection, leading to runtime issues related with closing long-term connections and the maximum number of temporal addresses supported by the system. This can be observed in their current Linux kernel implementation. This work proposes a simple, yet practical, implementation for this, and extends it by proposing the Anonymity Severs, which enable the nodes to use identities from different prefixes and hence, companies (Mapping Service providers).

7.4 Implementation

This section discusses implementation considerations for the solutions exposed in 6. An RTR-compatible MN requires minor implementation changes in order to use the proposed location privacy solution. A decision process, to decide which nodes are trustworthy or not, is the only additional code to be implemented. The proposed implementation for this is using a configuration file in the MN, which stores a list of EIDs that the MN will trust. When receiving a query from one of those EIDs, the MN will reply with its real RLOC.

Before delving into the details of the identity privacy implementation, its common use case should be discussed. Typical users do not want (or even be aware of) privacy in their normal communications. They want to be private just when connecting to untrusted sites. Those kinds of connections are not frequent and are distributed in time. The “private mode” on modern web browsers could serve as an example of this usage pattern.

With this in mind, the solution that seems more balanced between complexity and efficiency is using a single tEID rather than one per connection. This EID is shared by all the connections that require privacy and it is refreshed after a pre-defined period. If there are active connections, then the tEID will not change until the system does not have any active (private) connections.

The main reason for this is that using a unique tEID for each connection in the system is infeasible and impractical. The amount of tEIDs required to provide a unique one to each connection can be potentially huge. Moreover, restricting the temporary EIDs to just a few applications or connections still requires the implementation to deal with a variable number of simultaneous tEIDs. Having just one tEID changing over time keeps the complexity of the implementation at a reasonable level and is enough to fulfil the requirements of the common use case.

Another issue to discuss is how the system decides which connections require identity privacy. Leaving this to the network-layer is not trivial, since it does not usually have enough information. The proposed approach is to delegate this decision to the upper layers. Each application decides which connections use the tEID (for instance as the private browsing mode). In order to implement this, this work proposes using a new socket option [29]. This provides the programmers the flexibility to choose when privacy extensions should be applied. For backwards compatibility with existing applications not using this socket option, an alternative is proposed by means of a connection-manager application. The connection manager can be used to enable or disable identity privacy globally, for all applications, by switching between the real and temporary EIDs.

Chapter 8

Conclusions

This work has presented a set of solutions to provide location and identity privacy to LISP Mobile Nodes. Location privacy is a well-known problem usually solved by proxies. Here it has been presented a proxy based solution that takes advantage of LISP NAT-traversal capabilities. Since communications through proxies have a higher latency, this work has proposed a simple extension to allow mobile nodes to choose whether or not they want privacy. This prevents performance degradation for non-private connections.

This work has also proposed two different approaches to solve the identity privacy issue. Based on the idea of using temporal identifiers to hide the real identity of the MNs it has defined two different solutions adapted to different scenarios. The first approach does not require (or requires just a few) modifications to the LISP infrastructure, it is based on temporal auto-generated identifiers. The second one requires the deployment of a new element called Anonymity Server. It serves as a kind of DCHP server to provide and manage heterogeneous and distributed temporary identifiers.

Both issues have been addressed taking a realistic approach aiming for deployment. In particular this work has briefly discussed the business model for the proposed extensions along with its implementation.

Appendix A

Contributions

The research exposed in this work has conducted to several contributions, the most significant ones are gathered in this appendix.

A.1 Papers

The following three papers have been written or are currently being written based partially or fully in the work that appears in this document. At the time of this writing, one of the documents has been approved to be published, another has been submitted and is pending of approval and the last one is currently to be submitted.

- Alberto Rodríguez Natal, Loránd Jakab, Marc Portolés, Vina Ermagan, Preethi Natarajan, Fabio Maino, David Meyer, Albert Cabellos Aparicio.
  LISP-MN: Mobile Networking through LISP.
  in Proc. of Springer Wireless Personal Communications Journal, 2012
  Impact factor (2010): 0.507

- Alberto Rodríguez Natal, Loránd Jakab, Vina Ermagan, Preethi Natarajan, Fabio Maino, Albert Cabellos Aparicio.
  Privacy extensions for LISP-MN.
  IEEE Globecom. 2012.
  (Submitted).

- Alberto Rodríguez Natal, Florin Coras, Loránd Jakab, Marc Portolés, Vina Ermagan, Preethi Natarajan, Fabio Maino, David Meyer, Albert Cabellos Aparicio.
  LISP-MN: What a mobility protocol can provide to the Future Internet?
  IEEE Communications Magazine.
  (To be submitted)

A.2 IETF draft collaboration

To provide the privacy mechanism exposed in 6.1 a RTR and NAT traversal aware MN and Map Server are needed. At the beginning of this research, none of these were available. The current efforts at that time were towards the publication of the IETF draft.

To help the LISP community and to support the authors writing the draft, an active implication within the development of the IETF draft was adopted. Several reviews of the draft were done during the stages prior to its publication on the IETF site. Meanwhile, helpful comments, suggestions and feedback were provided to the authors.

This help was acknowledged by the draft authors, and can be checked on the "Acknowledgements" section of the IETF document [20].

A.3 Code

The first step to prove the correctness of the solutions provided in 6.1 was to test the NAT traversal draft specification with a RTR, a NAT traversal aware MN and a NAT traversal capable Map Server. There was no available code because the draft was on a very early stage at the time this research started. In order to perform NAT traversal tests and hence basic location privacy tests, a LISP NAT traversal implementation was developed on top of LISPmob. The code for an experimental branch of a LISPmob MN implementation with NAT traversal support can be found in [41]. Since there was no NAT traversal aware Map Server, neither RTR, available, a dummy RTR was developed. Just to test the NAT traversal capabilities of the MN. The Map Server NAT traversal functionalities were moved to this dummy RTR. This proof-of-concept, highly experimental and unstable RTR code can be found in [42].

Bibliography

[1] J. Rexford, C. Dovrolis, "Point/Counterpoint: Future Internet architecture: clean-slate versus evolutionary research",Communications of the ACM, Volume 53, Number 9 (2010), Pages 36-40.

[2] J. F. Shoch, "Inter-Network Naming, Addressing, and Routing", in IEEE Proc. COMPCON Fall 1978, pp. 72-79. Also in Thurber, K. (ed.), Tutorial: Distributed Processor Communication Architecture, IEEE Publ. #EHO 152-9, 1979, pp. 280-287.

[3] E. Nordmark, M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.

[4] R. Moskowitz et al "Host Identity Protocol", RFC 5201, April 2008.

[5] R. Atkinson, S. Bhatti, An Introduction to the Identifier-Locator Network Protocol (ILNP), IEEE London Communications Symposium (LCS), September 2006.

[6] J. Pan, R. Jain, S. Paul, C. So-In, "MILSA: A New Evolutionary Architecture for Scalability, Mobility, and Multihoming in the Future Internet", in IEEE Journal on Selected Areas in Communications (JSAC), Special issue on Routing Scalability, 2010

[7] O. Peters, S. ben Allouch, "Always connected: a longitudinal field study of mobile communication", Telematics and Informatics, Volume 22, Issue 3, August 2005

[8] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of internet hosts", in Proceedings of the 4th ACM SIGCOMM conference on Internet measurement (IMC '04), 2004.

[9] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards IP geolocation using delay and topology measurements", in Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06), 2006.

[10] http://androidsecuritytest.com/features/logs-and-services/loggers/carrieriq/

[11] http://www.lisp4.net/

[12] http://www.openlisp.org/

[13] http://lispmob.org/

[14] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol (LISP)", draft-ietf-lisp-23, Internet Engineering Task Force, May 2012, work in progress.

[15] D. Meyer, "The Locator Identifier Separation Protocol (LISP)", The Internet Protocol Journal, Volume 11, No. 1, March 2008.

[16] V. Fuller, D. Lewis, V. Ermagan, "LISP Delegated Database Tree", draft-fuller-lisp-ddt-01, Internet Engineering Task Force, March 2012, work in progress.

[17] L. Jakab, A. Cabellos-Aparicio, F. Coras, D. Saucez, and O. Bonaventure, "LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System", in IEEE Journal on Selected Areas in Communications, March 2010.

[18] D. Meyer, D. Lewis, D. Farinacci, C. White, "LISP Mobile Node", draft-meyer-lisp-mn-07, Internet Engineering Task Force, April 2012, work in progress.

[19] V. Fuller, D. Farinacci "LISP Map Server Interface", draft-ietf-lisp-ms-16, Internet Engineering Task Force, March 2012, work in progress.

[20] V. Ermagan, D. Farinacci, D. Lewis, J. Skriver, F. Maino, C. White, "NAT traversal for LISP", draft-ermagan-lisp-nat-traversal-01, Internet Engineering Task Force, March 2012, work in progress.

[21] D. Lewis, D. Meyer, D. Farinacci, V. Fuller, "Interworking LISP with IPv4 and IPv6", draft-ietf-lisp-interworking-06, Internet Engineering Task Force, March 2012, work in progress.

[22] C. Perkins, "IP Mobility Support" RFC 3344, August 2002.

[23] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[24] P. Srisuresh, K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[25] R. Koodli, "IP Address Location Privacy and Mobile IPv6: Problem Statement", RFC 4882, March 2007.

[26] Y. Qiu, F. Zhao, Ed., R. Koodli, "Mobile IPv6 Location Privacy Solutions", RFC 5726, February 2010

[27] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[28] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[29] E. Nordmark, S. Chakrabarti, and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, September 2007.

[30] F. Bao, R. Deng, J. Kempf, Y. Qiu, and J. Zhou, "Protocol for Protecting Movement of Mobile Nodes in Mobile IPv6", draft-qiu-mip6-mnprivacy-00, March 2005.

[31] C. Castelluccia , F. Dupont, and G. Montenegro, "A Simple Privacy Extension for Mobile IPv6", draft-dupont-mip6-privacyext-04, July 2006.

[32] K. Weniger and T. Aramaki, "Route Optimization and Location Privacy using Tunneling Agents (ROTA)", draft-weniger-rota-01, October 2005.

[33] D. Goldschlag, M. Reed, and P. Syverson "Hiding Routing Information", in the Proceedings of Information Hiding: First International Workshop, May 1996, pages 137-150.

[34] R. Dingledine, N. Mathewson, and P. Syverson "Tor: The Second-Generation Onion Router", in the Proceedings of the 13th USENIX Security Symposium, August 2004.

[35] W. Feller, "An Introduction to Probability Theory and Its Applications", Vol. 1, 3rd edition, John Wiley & Sons, New York, 1970.

[36] R. Grimaldi, "Discrete and Combinatorial Mathematics: An Applied Introduction", 4th edition, 1998.

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

[38] R. Hinden, "New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG", RFC 1955, June 1996.

[39] M. O'Dell, "GSE - An Alternate Addressing Architecture for IPv6", draft-ietf-ipngwg-gseaddr-00, February 1997

[40] Concise Oxford English Dictionary, Oxford University Press, 2008.

[41] https://github.com/LISPmob/lispmob/tree/natt

[42] https://github.com/arnatal/lispmob/tree/natrtr