Location and Identity Privacy for LISP-MN

Alberto Rodriguez-Natal*, Lorand Jakab†, Vina Ermagan†,
Preethi Natarajan†, Fabio Maino† and Albert Cabellos-Aparicio*
*Universitat Politecnica de Catalunya. {arnatal, acabello}@ac.upc.edu
†Cisco Systems. {lojakab, vermagan, prenatar, fmaino}@cisco.com

***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, also offers native mobility. In this context, LISP-MN is a particular case of LISP and specifies mobility. Mobility protocols have an inherent issue with privacy since some users may not want to reveal their location or their identity. In this paper, we present an overview of LISP-MN and propose solutions to enable privacy, both in terms of location and identity.**

I. Introduction

The Locator/ID Separation Protocol (LISP) [1] decouples identity from location on current IP addresses by creating two separate namespaces, 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. Among other location/identity separation schemes, LISP has a unique position: it is incrementally deployable, it does not require changes to transport/application implementations and, more importantly, it is already under active deployment [2]. Currently, there exist proprietary LISP implementations [3] as well as open-source initiatives [4], [5].

LISP design allows EIDs to remain unchanged even if a topological change, such as a handover, occurs which makes it well suited for mobility [6], [7]. Indeed, the LISP mobility protocol (LISP-MN [6]) proposes LISP-enabled endpoints, providing legacy applications with smooth mobility across access technologies and service providers. LISP introduces a Mapping System [8] as well, a distributed database that contains EID-to-RLOC bindings. The LISP-MN protocol uses the Mapping System to disseminate such bindings.

Since mobility protocols typically use addresses to locate users, they raise 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 geographical localization techniques [9]. This issue is exacerbated in LISP-MN when compared to other mobility protocols, such as Mobile IP [10], [11]. 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. In addition to location privacy, anonymity is of an increasing concern as well for a subset of today's Internet users. As a result of these concerns, the industry is developing mechanisms to improve 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 behavioral tracking. However, a LISP-MN host still discloses its unique EID even in these browsers operating mode, making EID based tracking possible. Given the fact that an assigned EID rarely changes (e.g., a mobile phone number), it can be easily associated to the user's identity and might be desirable to not disclose it in order to protect user's anonymity.

In this paper we discuss how LISP-MN can address both issues: location and identity privacy. It is important to note that we take a realistic approach when extending LISP-MN, since we aim to propose *deployable* solutions, and minimize the changes to the main LISP protocol. Further, we also analyze the level of security achieved with the proposals that appear in this paper, their required trade-offs and the feasibility of their implementation. Finally, we evaluate the burden introduced by the proposals in both the data and control planes.

II. LISP-MN Overview

The Locator/ID Separation Protocol (LISP) [1] 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 EID of the host for transport layer connections. RLOCs are IPv4 or IPv6 addresses used for routing through transit networks. In order to reach a host, identified by its EID, one must first find the current location (RLOC) of the host. LISP introduces a distributed and publicly accessible Mapping System [8] (constituted by Map Servers [12]), that is designed to serve the EID-to-RLOC mappings and policy information. It is updated by Map-Registers, queried by Map-Requests, and answers with Map-Reply messages. EID-based packets

This work has been partially supported by a Cisco grant, by the Spanish Ministry of Education under grant FPU2012/01137 and by the Catalan Government under grant 2014SGR-1427.

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

are encapsulated into RLOC-based packets to travel through the legacy Internet. LISP uses special gateway routers called Tunnel Routers (xTR) to communicate with the Mapping System, and perform these encapsulation and decapsulation operations.

LISP Mobile Node (LISP-MN) [6] uses these features to build a mobility architecture and protocol based on LISP. 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 MN itself is in charge of the xTR operations. The location of the host can change several times during an ongoing connection without breaking the transport layer connection. When the host's location (RLOC) changes, the MN will update the binding in the Mapping System and encapsulate the packets towards the new RLOC. More details and improvements of LISP-MN can be found in [13].

Figure 1 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.

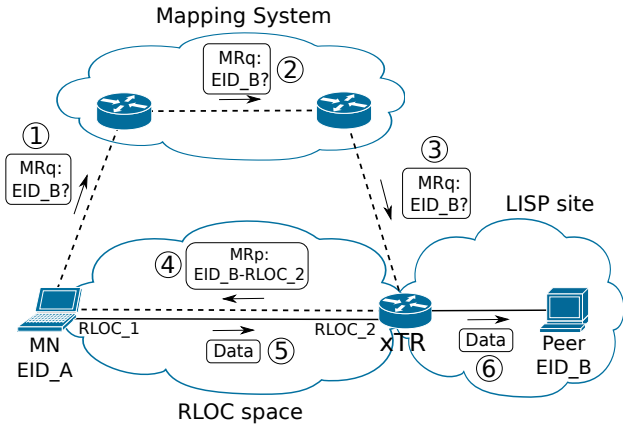

Fig. 1. **LISP-MN Overview.**

III. Privacy in LISP-MN

In this section we describe the proposed solutions to provide both location and identity privacy to the MN. Although we present different solutions that address these issues independently, both proposals can be combined to provide full privacy to LISP-MN.

A. Location privacy

Location privacy is a well-known problem in mobility and the most common solution is to use a trusted proxy. This way the proxy forwards the traffic from the MN and only the locator of the proxy is exposed. The LISP architecture offers proxies called RTRs (Re-Encapsulation Tunnel Routers) that can be used for this purpose. The RTRs receive LISP traffic, decapsulate it and rather than forward the traffic to end-hosts, they lookup in the Mapping System for an appropriate next LISP hop and re-encapsulate the traffic towards it. They serve in LISP deployments to provide Traffic Engineering possibilities [14] and NAT Traversal capabilities [15] to LISP nodes.

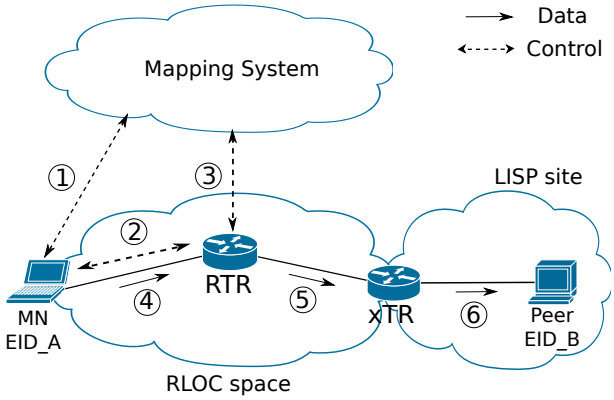

Fig. 2. **LISP-MN using an RTR proxy**

In order to achieve location privacy for LISP-MN using an RTR proxy, we can leverage on the NAT traversal mechanism. The NAT traversal procedure involves detecting the presence of a NAT, negotiate the use of an available RTR proxy, establishing a tunnel through the NAT towards the proxy and detour all traffic from, and to, the MN through the proxy. For details on the NAT traversal procedure please see the specification in [15].

Figure 2 covers the part of the NAT traversal mechanism relevant to this paper, i.e. the negotiation of the RTR and the traffic detour. The MN requests and gets a list of available RTRs from the Mapping System (1), the MN selects one of them and configures the RTR as its network attachment point (2, 3). From that point on, the MN detours all its traffic towards the RTR (4) and therefore remote nodes receive the traffic from the RTR (5,6) and not from the MN. The traffic follows the same path on its way back to the MN.

Due to the presence of the RTR on the path, the location of the MN is guaranteed to be private during the NAT traversal session. An MN willing to hide its location can trigger the NAT traversal procedure, and thus force its traffic to go through an RTR, even when it knows there is no NAT present. Moreover, for the specific purpose of location privacy, the NAT traversal procedure can be improved as follows.

We propose to add a flag to the control messages exchanged on the bootstrap of the NAT traversal mechanism to notify that the procedure is going to be used to achieve location privacy (it might or might not be used also for NAT traversal). The extra flag allows the Mapping System to know that it should send, alongside the list of available RTRs, extra privacy-related information. This could include the geo-coordinates of the

RTRs, their the current load, the probability of an attack on each on them, etc. The exact information included is up to the specific implementation, but it can be encoded in the LISP control messages using the format defined in [16]. The MN can use that extra information provided by the Mapping System to choose the most suitable RTR for its needs, e.g either choose a close RTR to reduce the latency or choose a far away one to better mislead possible attackers.

A company interested in offering location privacy to its costumers can deploy a set of RTRs in the Internet. In order to access the RTR the MN requires a pre-shared key, in a similar way it needs one to register to its Map Server [12]. This pre-shared key, that grants access to the RTR, can be used to enforce that the client is paying for the service. 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 provide more deceptive locations to offer a better service to the subscribers. With this in mind, the company that invests in more well placed RTRs will be more competitive.

B. Identity privacy

In this section we extend LISP-MN to offer identity privacy, the main purpose being to hide the EID to untrusted peers. A classic approach on legacy IP networks to deal with identity protection is to use temporary IP addresses [17], we take base on that concept and propose to use temporary identifiers rather than the real one. This section proposes two different approaches to provide such temporary identifiers on a LISP-MN deployment, a MN-driven infrastructure-less solution and a solution dependent on the deployment of a new element. It is important to note that in both cases the identity privacy can only be offered when the MN initiates the connection.

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 [17] 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, we borrow the mechanisms described in [17]. 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 on its Map Server to detect duplicated addresses. If the address is already in use by another MN the Map Server replies positively and then the MN has to generate another address and query again. If the address is not in use, the Map Server replies with a negative Map-Reply and the MN knows that it can register the tEID.

In the case that two different MNs generate the same tEID at the same time, both MNs will receive negative Map-Reply at the time of checking the presence of that tEID on the Map Server, and therefore both of them will think that the tEID is available to be used. Since the probability that two different MNs generate the same tEID and query the Map Server at the same time is extremely low [18] due to the 64 bits address space, an optimistic address collision detection mechanism can be applied, i.e. the MN register its tEID as soon as it checks that it is not already registered and starts establishing connections with it. After a time-out the MN checks again the data in the Mapping System to see if its information was correctly recorded or if something went wrong during the registration process (i.e. another MN registered the same tEID). In the rare case that a collision occurred, it will roll back, drop the established connections and reinitialize the tEID registration (with a new generated tEID).

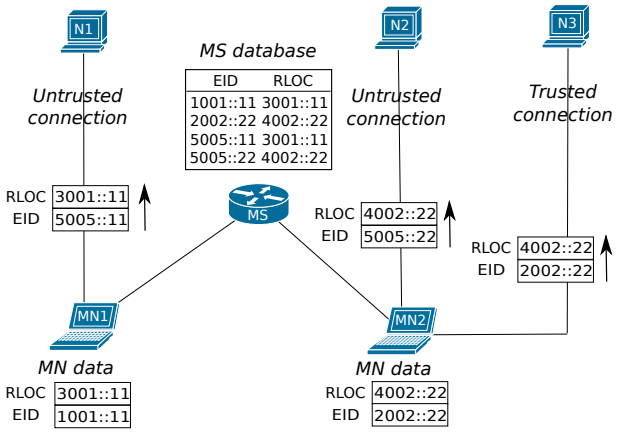

Fig. 3. **MN generated temporary EIDs.**

Figure 3 shows an example of the proposal. The MNs are sharing the prefix 5005::/64 to generate temporary EIDs. The last 64 bits of the addresses belonging to that prefix are generated by the MN. They register these generated addresses in the Map Server, and use them to establish connection to not trusted nodes.

With this architecture, a misbehaving node, with access to the shared prefix, 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 (a pre-shared key for their real EID) to authenticate each individual node. Avoiding traffic hijacking can be achieved by requiring explicit dropping of a tEID in use by the previous owner.

The infrastructure-less solution 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 own a prefix and share it between MNs of their property. If any of the MNs 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.).

2) Infrastructure dependent: This approach introduces a new element, the "Anonymity server" (AnonS). Its function is similar to that of a DHCP server [19], 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 [12]).

Figure 4 illustrates the solution. The AnonS keeps a database of its tEIDs (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 tEIDs can belong to different prefixes and Map Servers (5005::55 belongs to MS2 and 7007::77 to MS3).

Deploying an AnonS generates revenue for its operator, which controls the access to the identity privacy service. At sign-up 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.

IV. Related work

Location privacy in mobility is a well-known issue which Mobile IP has faced up in [20]. In particular they use a similar approach as the one presented in section III-A to solve it, in this case the Home Agent acts as the proxy. The authors of [21] extend this idea by proposing to deploy redundant Home Agents to enhance privacy. Finally, a different approach to location privacy has been proposed in [22] 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 [17] are generated for identity privacy. Although these "pseudo home addresses" are intended to provide location privacy, they implicitly also serve as an identity-privacy mechanism.

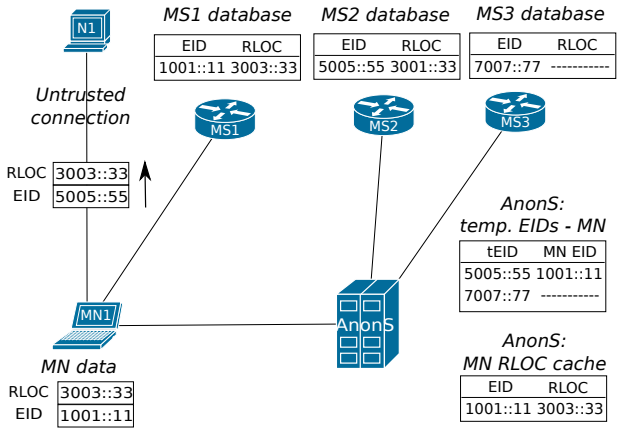

Fig. 4. **Anonymity Server.**

In general, using temporal addresses is a well-known approach to provide identity privacy in the IPv6 area, being the main proposal the one standardized in [17]. Another good example can be found in [23]. Note that [17] forces keeping one temporary identifier per connection, which can lead to runtime issues related with closing long-term connections and the maximum number of temporary addresses supported by the system. This can be observed in the current Linux kernel implementation. On the other hand, mechanisms to hand out addresses from a pool to hosts are also well-known [19]. In this paper we have taken these established approaches and adapt them to the LISP-MN architecture.

V. Analysis

This section discusses the level of security provided by the mechanisms proposed in this paper, the trade-offs they impose and the feasibility of their implementation.

A. Location privacy

The proxy-based approach proposed guarantees that the location of the MN is never exposed to remote nodes, however the use of a proxy introduces an inefficient routing path that degrades the performance of the LISP-MN communications. To alleviate this, the extensions proposed to improve vanilla proxy selection allow the MN to choose the most suitable RTR for its needs. Particularly the MN can get the geo-location of the proxy and select an RTR based on that information. If that is the case, there is a trade-off on which RTR to select

since closer RTRs would provide better performance, but also disclose more data about the potential location of the MN. We recommend select randomly from a set of mid-range located RTRs to balance among location disclosure and performance degradation. Besides, the load on the RTRs can be alleviated deploying more RTRs and providing their load information to MNs to help to select a non-overloaded one.

In terms of implementation, an MN compatible with NAT traversal can use the NAT traversal mechanism to get basic location privacy. The usage of the extensions proposed on this paper requires support for parsing and encoding/decoding the extra information on both sides, i.e. on the MN and on the Mapping System. Additionally, the Mapping Systems needs to get populated with the information regarding the RTRs, how to populate the Mapping System with that information is out of the scope of this paper.

B. Identity privacy

In this paper we propose a simple, yet practical, design of an auto-managed identity privacy by means of auto-generated temporary EIDs that does not require of any new infrastructure deployment. The trade-offs of this approach are that it only serves for IPv6 addresses, that it imposes extra computation on the MNs and that the MNs are still traceable at prefix level. We extend it by proposing the Anonymity Severs, which enable the nodes to use identities from different prefixes, at the cost of requiring new infrastructure elements. Moreover, the use of different prefixes, gives to the users of an AnonS a higher level of anonymity than the use of traditional IPv6 privacy mechanisms. In those, the MN still can be tracked at IP prefix level, whereas with the AnonS solution the MN's EID prefix can be regularly changed among prefixes that can belong to different domains.

There is what makes the AnonS specially attractive as a mechanism to provide identity privacy and distinguishes it from the previous presented solution. 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.

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 tEID 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 amount of tEIDs required to provide a unique one to each connection can be potentially huge. Having just one tEID changing over time keeps the complexity of the implementation at a reasonable level and is enough to fulfill the requirements of the common use case.

Another issue 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, we propose using a new socket option [24]. 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.

VI. Evaluation

This section evaluates the proposed solutions to asses the extent of their impact in both the data and control planes.

A. Data-plane

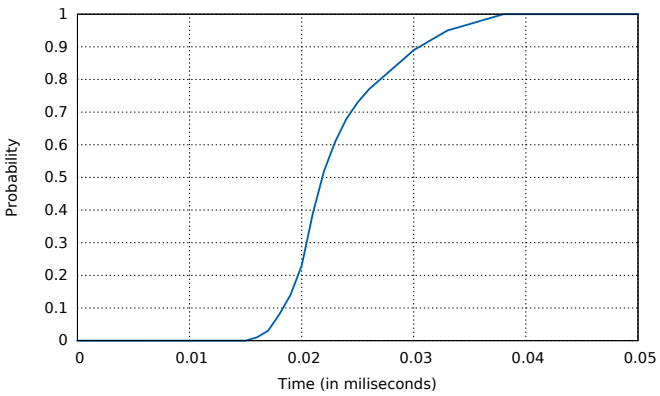

Fig. 5. **Constant delay introduced by the RTR**

In terms of evaluating the data-plane burden imposed by the proposed solutions, it is worth to note that both approaches for identity privacy do not modify the data-plane operation and therefore do not impose any burden. On the other hand, the solution for location privacy does impose a penalty in the data-plane since, like in every proxy solution, the use of another element in the path increases end-to-end latency due to the longer path and the extra processing time. The latency increment caused by the path stretch varies depending on the location of the RTR chosen, however the processing time at the RTR is constant. To evaluate how much constant delay an RTR introduces, we have built a prototype of the proposal in V-A using the LISPmob open-source implementation [4] following the topology depicted in figure 2.

TABLE I
EXTRA CONTROL MESSAGES REQUIRED BETWEEN ENTITIES

	MN MS	MN AnonS	AnonS MS
Location (with NAT capable nodes)	0	0	0
Identity: Infrastructure-less (no check for duplicates/collisions)	0	0	0
Identity: Infrastructure-less (checking duplicates)	2	0	0
Identity: Infrastructure-less (checking duplicates & collisions)	4	0	0
Identity Infrastructure-dependant	0	2	2

Figure 5 shows the Cumulative Distribution Function of the processing time of packets (1000000 packets at 1000 packets per second) at an RTR running LISPmob on a desktop Linux machine (2GHz dual-core with 1 GB of RAM). The figure shows that most packets suffer a delay of less than 30 microseconds, which is negligible for most scenarios. Furthermore, it is expected that hardware-based LISP solutions [3] can provide even lower processing times.

B. Control-plane

We evaluate the control-plane modifications in terms of the number of extra control messages that are required to exchange between entities in order to support the proposed solutions. For location privacy, there is no need for any extra messages, since the signaling is the same that is used to perform NAT traversal. The identity-privacy solutions require however additional control messages.

The infrastructure-less approach requires no extra control messages if there is no duplicate address detection or collision check. If the MN looks for duplicated addresses, then one request to the Mapping System and its reply are needed. Twice this number if collision check is also performed. The infrastructure-dependent solution doubles the number of signaling messages of vanilla LISP registration due that first it is the MN who registers to the AnonS, and then is the AnonS who registers to the Mapping System. Note that in this case the MN does not register the tEID to the Mapping System since this is done by the AnonS. Table I summarizes the extra signaling messages required between entities.

VII. CONCLUSIONS

In this paper we have 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 we have presented a proxy based solution that takes advantage of the LISP NAT-traversal mechanism and extend it to better serve the location privacy purpose.

We have also proposed two different approaches to solve the identity privacy issue. Based on the idea of using temporary identifiers to hide the real identity of the MNs we have defined 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 temporary auto-generated identifiers. The second one requires the deployment of a new element called Anonymity Server. It serves as a kind of DHCP server to provide and manage heterogeneous and distributed temporary identifiers.

We have addressed both privacy issues taking a realistic approach aiming for deployment. In particular we have briefly discussed the trade-offs of the proposed solutions alongside with the feasibility of their implementation. The evaluation shows that the burden that the solutions impose in the data and control plane operations is reasonable.

REFERENCES

[1] D. Farinacci et al., "Locator/ID Separation Protocol (LISP)", IETF RFC 6830, Feb. 2013.
[2] http://www.lisp4.net/
[3] http://lisp.cisco.net/
[4] http://lispmob.org/
[5] D. C. Phung et al., "The OpenLISP control-plane architecture", IEEE Network Magazine, Vol. 38, No. 2, pp: 34-40, March-April 2014.
[6] A. Rodriguez-Natal et al., "LISP-MN: mobile networking through LISP", Wireless personal communications, 70(1), 253-266, May 2013.
[7] A. Galvani et al., "LISP-ROAM: network-based host mobility with LISP", Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture, ACM, 2014.
[8] V. Fuller et al., "LISP Delegated Database Tree", draft-ietf-lisp-ddt-02, Internet Engineering Task Force, Oct. 2014, work in progress.
[9] B. State, I. Weber and E. Zagheni, "Studying inter-national mobility through IP geolocation", In Proceedings of the sixth ACM international conference on Web search and data mining (WSDM '13), 2013.
[10] C. Perkins, "IP Mobility Support" RFC 3344, August 2002.
[11] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
[12] V. Fuller and D. Farinacci, "LISP Map-Server Interface", IETF RFC 6833, Feb. 2013.
[13] M. Menth, D. Klein, and M. Hartmann, "Improvements to LISP Mobile Node", in Proceedings of the 22nd International Teletraffic Congress (ITC), Amsterdam, Netherlands, September 2010.
[14] D. Farinacci, M. Kowal and P. Lahiri, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-07, Sep. 2014, work in progress.
[15] V. Ermagan et al., "NAT traversal for LISP", draft-ermagan-lisp-nat-traversal-07, Internet Engineering Task Force, Feb 2015, work in progress.
[16] D. Farinacci, D. Meyer, and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-05, May 2014, work in progress.
[17] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
[18] N. Moore, "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, April 2006.
[19] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, 1997.
[20] R. Koodli, "IP Address Location Privacy and Mobile IPv6: Problem Statement", RFC 4882, March 2007.
[21] H. Takahashi and T. Minohara, "Enhancing location privacy in Mobile IPv6 by using redundant home agents", IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), March 2012
[22] Y. Qiu, F. Zhao, Ed., R. Koodli, "Mobile IPv6 Location Privacy Solutions", RFC 5726, February 2010
[23] S. Hanet al., "Expressive privacy control with pseudonyms", In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM '13), ACM, New York, NY, USA, 291-302, 2013.
[24] E. Nordmark, S. Chakrabarti, and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, September 2007.