Securing the LISP Map Registration Process

D. Montero, M. S. Siddiqui, R. Serral-Gracià, X. Masip-Bruin, and M. Yannuzzi
Advanced Network Architectures Lab (CRAAX), Technical University of Catalonia (UPC)
Vilanova i la Geltrú, Spain

***Abstract*—The motivation behind the Locator/Identifier Separation Protocol (LISP) has shifted over time from routing scalability issues in the core Internet to a set of use cases for which LISP stands as a technology enabler. Among these are the mobility of physical and virtual appliances without breaking their TCP connections, seamless migration and fast deployments of IPv6, multihoming, and data-center applications. However, LISP was born without security, and therefore is susceptible to attacks in its control-plane. The IETF's LISP working group has recently started to work in this direction, but the protocol still lacks end-to-end mechanisms for securing the overall registration process on the mapping system. In this paper, we address this issue and propose a solution that counters the attacks. We have deployed LISP in a real testbed, and compared the performance of our proposal with current LISP implementations, in terms of both messaging and packet size overhead. Our preliminary results prove that our solution offers much higher security with minimum overhead.**

***Index Terms*—LISP, security, Loc/ID split, routing, Internet.**

I. Introduction

The Locator/Identifier Separation Protocol (LISP) [1] was initially devised to tackle routing scalability issues in the core Internet. However, due to its intrinsic address splitting and its simple architecture, LISP was promptly spotted as a technology with a remarkable potential in other areas in networking. As a consequence, the focus on LISP has shifted over time and is now becoming a key technology in areas related to virtualization, mobility, and cloud applications. The basic idea in LISP is the decoupling of the Route LOCator (RLOC) and the End-point IDentifier (EID) spaces in the addressing scheme. LISP supports provider independent and globally unique Identifier addresses, and employs a Map-and-Encap scheme, along with an Identifier-to-Locator Mapping System to bind the two address spaces. Another important feature is that LISP is address family agnostic, so the Map-and-Encap and Decap processes can handle mixes of IPv4 and IPv6 indistinctively. These features have made it highly flexible, and therefore, it is considered an enabler for a variety of applications.

In order to be able to use LISP, an edge network implementing LISP, i.e., a LISP-Site, registers the EID prefixes on a Map Server (MS) in the Mapping System. The registration could be done against a single or a set of RLOC addresses, thus enabling global reachability. As currently defined in [1], this map registration process is a static procedure based on manual configurations that need to be set in advance. These configurations need to be done both on the border routers in the LISP-Site, called Egress Tunnel Routers (ETRs) and on the Map Server. Once the manual configurations are in place, each ETR will attempt to register its mappings with the Map Server. The latter can verify the requests against the predefined configuration using pre-shared keys. The pre-shared keys allow to assess the validity of the map registration, since each ETR has its own key which is shared only with the Map Server.

It is important to notice that this existing *pre-shared key* security mechanism between the ETR and the MS falls short of countering a number of relatively simple attacks, such as RLOC address spoofing. Indeed, LISP lacks a procedure for ensuring whether a certain ETR is allowed to use a particular RLOC address for registering an EID prefix. In addition, current LISP specifications exclude the EID prefix owner's role (i.e., the EID-Holder) in the map registration process, since the set of valid EID prefixes are manually preconfigured within the ETR. With this approach, the registration process undermines the provider independence and mobility features of the EID address space, which are in fact main drivers for LISP. These manual and static practices are due to the fact that LISP lacks mechanisms for global EID prefix authorization, which, as we shall show later on, are essential for the practical feasibility of mobility and roaming scenarios in LISP. In a nutshell, global EID prefix authorization refers to the development of security mechanisms through which a Map Server can determine whether a particular ETR belonging to a particular LISP-Site is authorized to register an EID prefix on its behalf—these aspects shall be described in more detail later in Section II.

Overall, securing the map registration process is a vital requirement, not only to avoid attacks on the LISP control-plane but also to maintain a correct operational state of the Mapping System. The IETF's working group on LISP has recently started to work in this direction, and their first initiative is LISPSEC [2]. For the moment, LISPSEC has only focused on securing the mapping queries, i.e., the *Map-Request* and *Map-Reply* messages with the Mapping System, so the vulnerabilities mentioned above remain unsolved.

In this paper, we propose a secure map registration process for LISP that not only works end-to-end (i.e., it now involves the EID-Holder) but also enables dynamic map registrations. Our solution is based on very efficient cryptographic mechanisms, which combined with actual LISP messaging, are able to enhance the overall security of the map registration process. During our work, we observed that both the Secure Inter-Domain Routing Working Group (SIDR WG) of IETF [3] and the LISP Working Group (LISP WG) [4] are targeting security aspects that involve the Internet's global routing system, but

This work was supported in part by an RFP granted by Cisco Systems, Inc., by the Spanish Ministry of Science and Innovation under contract TEC2012-34682, and by the Catalan Government under contract 2009 SGR1508.

978-1-4799-1353-4/13/$31.00 ©2013 IEEE

their efforts so far have been entirely disjoint. To bridge this gap, our work also explores how LISP can benefit from the security infrastructure already designed by the SIDR WG, including the Resource Public Key Infrastructure (RPKI) [5] and Route Origin Authorizations (ROAs) [6]. This paper proposes extensions for their adoption within the LISP ecosystem.

Moreover, our proposal has been deployed in a real testbed, on which we performed a performance comparison between legacy LISP and our solution in terms of both messaging and packet size overheads. Our preliminary results prove that our solution can offer much higher security with minimum overhead. It is worth highlighting that, the critical part of our solution can be implemented entirely in hardware, which mitigates any potential computation overhead caused by the encrypted messages.

The rest of the paper is organized as follows. Section II outlines some of the main issues and vulnerabilities in LISP. Section III introduces the new registration process and describes our approach for achieving end-to-end security. Then, we lay out our testbed in Section IV, and provide an analysis assessing the impact of our solution compared to the existing LISP implementation. Finally, in Section V, we conclude the paper with a discussion and an overview of future work.

II. LISP Control-Plane Vulnerabilities

The security vulnerabilities of LISP not only jeopardize its normal operations but also hamper its broader reach, since they could compromise the applications for which it is positioned as a technology enabler. In this section, we overview two of the most important security issues in LISP, namely, RLOC spoofing and lack of global EID authorization. These vulnerabilities enable the registration of mapping entries in the Mapping System that could result in the redirection of data plane traffic elsewhere, with consequences that might range from blackholing up to traffic sniffing.

A. RLOC Spoofing

The mapping entries on the Map Server (MS) consist of EID-to-RLOC bindings. However, a Map Register request can include an incorrect RLOC and marginalize the integrity of the mapping entry. In order to avoid that, the MS needs to ensure that a certain ETR is authorized to use a particular RLOC address for registering an EID prefix. Lack of such assurance can lead to different attacks by a malicious ETR, such as DoS attacks by traffic flooding.

Figure 1 illustrates an RLOC spoofing scenario. The malicious ETR from SP_1 performs a registration targeting SP_2 by specifying its locator $RLOC_2$ in the registration request. A number of such false RLOC registrations can be done to increase the impact of the flooding which could result in a DoS at the victim. Likewise, when an RLOC is spoofed, the mapping entries are compromised and further queries for the EID_A prefix's locator will retrieve the wrong $RLOC_2$. To verify this in practice, we have tried out the RLOC spoofing attack in our testbed with the current LISP implementation, and we have succeeded.

In summary, LISP does not define a mechanism to verify the authorization of RLOCs to ETRs. Any ETR can claim any RLOC during the registration process, which poses a serious concern on the dependability of the LISP control-plane. As we shall see in Section III, by introducing a slightly adapted version of the existing Route Origin Authorizations (ROAs) [6], we can provide dynamic assessment of the RLOC ownership and effectively avoid such attacks.

B. No Global EID Authorization

As mentioned earlier, the current map registration process completely alienates the EID host's role, hence making it dependent on the LISP-Site's ETRs. With this approach, there is no way that a MS can verify if an ETR is authorized by an EID host to perform map registrations on its behalf—this is because it is not even involved in the process. The current shared key security mechanism for the map registration process is a static stop-gap solution for handling the ETR registration. It requires manual preconfigurations of the EID prefixes, both on the ETR and on the MS, and a shared key between them. Furthermore, due to lack of recognition of the EID host as a separate entity, it not only falls short of providing global EID authorization but also fails to ensure end-to-end security. A solution to address these issues was proposed in [7], which leverages the RPKI/ROA infrastructure and defines a signed object, called *Identifier Origin Authorization (IOA)*. The *IOA* object can act as an authorization from an EID prefix holder towards a particular set of RLOCs to populate the mapping database. However, this approach burdens the EID prefix holder device with intensive cryptographic chores (i.e., signing, verifying and handling certificates).

Another noticeable observation regarding current LISP specification is its impact on the mobility of the EID host, as it recommends to explore Mobile IP technology in the case of mobility when an EID host moves relatively fast and requires to change its RLOC attachment point while maintaining

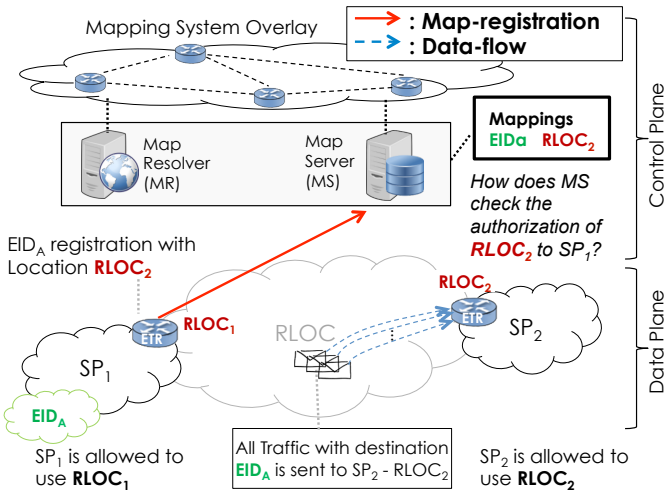


Fig. 1. RLOC Spoofing in LISP.

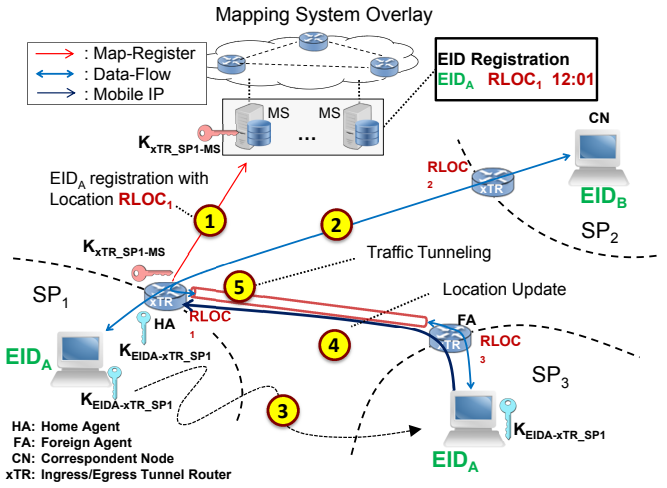


Fig. 2. Mobility scenario with current registration process.

session continuity. Figure 2 illustrates this scenario, where an EID host, EID_A, in SP_1 is registered on the mapping system by its ETR, namely xTR_{SP1} (Step 1). At certain moment, EID_A starts communicating with an EID host, EID_B in SP_2 (Step 2). Later on, EID_A moves to another LISP-Site SP_3 (Step 3). In order to keep an uninterrupted communication with EID_B, EID_A will use the shared key it has with the xTR_{SP1} to authenticate its location update through xTR_{SP3} (Step 4). Once authenticated and updated, xTR_{SP1} starts tunneling the traffic coming from EID_B towards EID_A at its new location (Step 5).

The extra burden for making this possible and securely can be summarized as follows: a) it requires handling another shared key between the xTR and the EID host; b) the EID host needs to authenticate the new location with the ETR; and c) the latter needs to sub-optimally forward traffic by tunneling so as to avoid losing the session. All this can be avoided by involving the EID host in the map registration process directly. This would enable the EID host to directly update its mapping entry in the mapping system when it is on the move. In our proposal, we show that involving the EID host in the map registration process, not only enables us to avoid using cross technologies in case of mobility (LISP and Mobile IP), but also paves the way for EID authorization.

III. Map Registration Proposal

In this section, we present our approach toward providing end-to-end security for the map registration process in LISP.

A. Preliminaries

Before getting into the details, we proceed to define some terms that will help explaining of our solution.

- **RLOC Verification Process:** The mechanism by which a Map Server in the Mapping System is able to securely establish the fact that a particular ETR belonging to a certain Service Provider is authorized to use an RLOC or a set of RLOCs.
- **EID Authorization Process:** The mechanism by which a Map Server in the Mapping System is able to securely establish the fact that a particular ETR is authorized to register an EID prefix on its behalf.

With RLOC verification in place, the RLOC spoofing attacks can be completely mitigated. In turn, EID authorization process will not only enable dynamic registrations on the move, but would also avoid the burden of relaying on third-party technologies, such as Mobile IP.

1) EID Ownership: In addition to the traditional actors in a LISP ecosystem, namely, the Ingress Tunnel Routers (ITRs) and the Egress Tunnel Routers (ETRs) in the LISP-Site, and the Map Resolvers (MRs) and the Map Server (MSs) in the Mapping System, we introduce a new role, represented by the "user" or "host in the LISP-Site bearing the EID", called the "EID-Holder". In our proposal, the EID-Holder is considered independent of the service provider, which is in fact one of the main hooks of LISP. The term EID-Holder refers to the fact that the user or host is the owner of the EID prefix. The EID prefix can be acquired through a service provider, a broker, or directly from the respective regional authority (e.g., RIRs), but its ownership stays with the EID-Holder. The EID-Holder identification allows it to initiate the map registration process itself, by sending a Service Request to the ETR of the service provider from which it plans to get the Internet service. The ETR of the service provider forwards the request to the MS in the Mapping System. Figure 3 depicts the updates proposed to the LISP architecture.

With the introduction of the EID-Holder—and emphasizing its separation from the service provider—there are now three actors involved in the map registration process: (1) the EID-Holder; (2) the ETR; and (3) the MS. An end-to-end secure registration process refers to the phenomenon that the EID-Holder is able to securely register its EID along with the RLOC of its current Service Provider on the MS, with MS making sure that: i) the ETR requesting to register the EID is authorized to do so; and ii) the ETR is authorized to use the RLOC given in the map registration request.

2) RLOC Authorization (RA): Next, we present an extension to the ROA concept as developed by the SIDR WG, which exploits the similarities between Route Origination in an inter-domain network with an RLOC used by an EID in a LISP-network. To this end, we propose an extension to legacy ROA that can be used for RLOC Authorizations. The ROA, as described in [6], is based on cryptographically signed information that binds the IP prefix with its legitimate owner's Autonomous System Number (ASN), and it is accompanied with the corresponding certificate. It assists the relying party to verify whether a particular ASN is the legitimate owner of a certain IP prefix or not. For the purpose of RLOC Authorization, we reuse the ROA design and structure for RLOC addresses, and thus:

> *"We define an RLOC Authorization (RA) as cryptographically signed information binding the xTR_{ID}, the ASN, and the set of RLOC addresses that are*

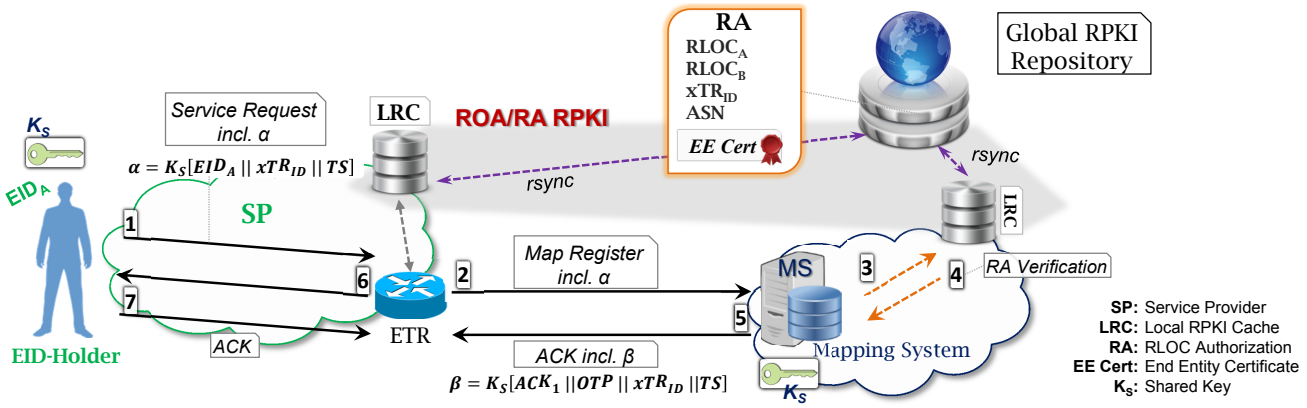


Fig. 3. Step-by-step overview of the new *Map Registration Process*. Each step in solid color determines exchanged messages related with the Map Registration. Gradient steps relate to messages exchanged with the RPKI, not directly related with the actual registration.

authorized to be used along with the respective certificate."

In the above definition, the xTR_{ID} uniquely identifies a LISP border router within an AS. In order to ensure global and timely dissemination of RAs, we will reuse the RPKI developed by the SIDR WG. It is worth highlighting that, RPKI has already been implemented and deployed by ARIN [8] and RIPE [9], and it is now under testing phase. The utilization of RPKI, however, requires some changes in the LISP architecture. Firstly, LISP Service Providers and Mapping System operators require the deployment of an RPKI-Cache to synchronize with the global RPKI. Secondly, the xTR in the LISP Service Provider and the MS in the Mapping System have to implement a protocol, similar to the RTR-RPKI protocol [10]. This is used for the communication with the Local RPKI Cache (LRC), in order to complete an RA verification query in a timely manner. As for the ROAs, a LISP Service Provider has to publish its RAs in the RPKI before conducting a map registration involving an RLOC, so that an MS can verify the legitimate use of the RLOC address.

3) Trust Scenarios: Depending on the relation among the different actors, i.e., EID-Holder, ETR, and MS, we identify three different trust environments for dynamic and secure end-to-end map registrations: i) completely trusted; ii) partially trusted; and iii) completely untrusted scenarios.

The first scenario assumes complete trust between the EID-Holder, ETR and the MS. This scenario is possible in the case that a user requests an EID through the same Service Provider from which it plans to request service as well. Furthermore, the Service Provider runs its own MS. In this scenario, security may be regarded as an optional requirement.

The second scenario assumes trust between the EID-Holder and the ETR only. This means that the Service Provider does not run a MS, and thus is using the mapping service offered by a third-party. This scenario has strong security requirements between the ETR and the MS.

The third scenario assumes no trust at all among the EID-Holder, the ETR, and the MS. This is typically the case of roaming scenarios, and requires strong security involving the three actors. In this paper, we focus on this case, since it is precisely the worst possible scenario.

The rest of the section is devoted to the presentation of our dynamic and secure end-to-end map registration proposal.

B. Secure LISP Map Registration

Our secure end-to-end map registration proposal is divided into three stages. In the first stage, the EID-Holder initiates the Service Request towards the ETR of the Service Provider. With this request, the Service Provider can register the new EID for the service in its xTRs. We assume that the EID-Holder is aware of the correct xTR_{ID} of the Service Provider through which it plans to get the service. The EID-Holder can learn about the xTR_{ID} through different means, e.g., in advance through certified templates advertised by the providers, online through DHCP, by manual entry, etc.[1] The second stage is when the Service Provider sends a Map-Register request to the MS for Map Registration. And the last stage is when the MS verifies and processes the registration request.

Once the registration is validated, the Mapping System may or may not send back an acknowledgement to the ETR or to the EID-Holder. The acknowledgement requirement can be tuned according to the trust environment scenario. As mentioned earlier, in this paper, we focus on untrusted scenarios, so any party can be an attacker. In order to achieve end-to-end security and EID authorization, we propose to use a shared key between the EID-Holder and the Map Server, leaving for future research the use of Public Key Cryptography in this part. The shared key is used as a way to validate the Map Register request at the MS and achieve EID authorization. Although, this technique is simple and not far from what is currently defined in LISP (i.e., a shared key between the ETR and the MS for ETR validation), we will show that our proposal cap-

[1]Due to space limitations, we cannot discuss here issues such as DHCP spoofing. In any case, there are palliatives to counter these actions, and clearly the technology used for learning the xTR_{ID} is auxiliary to the LISP map registration process, which is the target of this paper. As a side note, the simplest DHCP spoofing attacks inside the Service Provider could derive in DoS for the EID. More elaborated ones, such as coordinated attacks with an external provider could end up in traffic diversion and sniffing.

tures the whole problem and now allows dynamic registrations while offering end-to-end security.

The overall process to secure the map registration is shown in Fig. 3. In the first stage and prior to the Service Request, the EID-Holder must be aware of the Service Provider Identity (xTR_{ID}) from which it plans to use the service. Then, the EID-Holder computes α (cf. (1)) by first concatenating its EID, xTR_{ID}, and a timestamp TS, and then encrypting this information with the shared key $\mathcal{K}_S$ it has with the MS.

$$\alpha = \mathcal{K}_S\left(EID_a \| xTR_{ID} \| TS\right) \tag{1}$$

Hence, α is meant to be only visible to the corresponding MS in charge of the EID prefix, and will be used for the EID authorization process. The EID-Holder sends α in the *Service-Request* message to the ETR of the Service Provider, and it also adds in plain text the RLOC of the target Map Server, $RLOC_{MS}$, and its prefix EID_a (cf. step 1 in Fig. 3). Note that a potential attacker within the Service Provider—or the Service Provider itself—will not be able to change any information in α due to encryption and lack of $\mathcal{K}_S$. Moreover, a replay attack is not feasible as the timestamp may be used as a key to the registration, denying registrations with invalid timestamps. Furthermore, the Service Provider cannot overclaim EID prefixes due to the inability to produce a corresponding α.

Assuming that the Service Provider has already published (e.g., time ago) the respective RAs on the RPKI for the RLOC that it plans to use during the registration, the ETR can send then a signed *Map-Register* message to the corresponding MS. The signature in the message includes α (received from the EID-Holder), its xTR_{ID}, its RLOCs, and the EID prefix it wants to register, EID_a (cf. step 2 in Fig. 3).

In the third stage (cf. steps 3 and 4 in Fig. 3), the MS verifies the following:

- It verifies the signature of the *Map-Register* message. If valid then proceed, otherwise discard the request.
- It verifies the α and its contents using the respective shared key. If valid then proceed, otherwise discard the request.
- It verifies if the xTR_{ID} inside the α is the same as sent in the *Map-Register* message. If valid then proceed, otherwise discard the request.
- It also verifies if the requesting ETR is authorized to register against the RLOCs present in the Map Register request using the RA and the RPKI. The MS verifies the xTR_{ID} inside the α with the one present in the RA to complete the RLOC verification process.

If the EID authorization and RLOC verification processes are successful, then the MS adds this mapping entry into its records and sends back a signed acknowledgement to the ETR. In order to avoid any Man-in-the-Middle and coordinated attack on the acknowledgement, the MS includes in the signature of the reply message: an ACK, a One Time Password (OTP), EID_a (for which it conducted the map registration), and β. As detailed in (2), β is obtained by encrypting: the ACK, the locally generated OTP, xTR_{ID} (against which it registered the EID in the mapping entry), and the timestamp with the respective shared key $\mathcal{K}_S$ (cf. step 5 in Fig. 3).

$$\beta = \mathcal{K}_S\left(ACK \| OTP \| xTR_{ID} \| TS\right) \tag{2}$$

Then, the ETR verifies the signature of the ACK, and if successful, it forwards only β to the EID-Holder who initiated the Service Request (cf. step 6 in Fig. 3). The EID-Holder verifies β using the shared key and validates its contents. Note that while α was meant for the "eyes" of the MS only, β is meant for the "eyes" of the EID-Holder only.

If successful, the EID-Holder sends back an ACK to the ETR encrypting it with the OTP. Finally, the ETR verifies the encrypted ACK from the EID-Holder, and completes the secure triangle that involves the three actors required for providing end-to-end security in the LISP map registration process. Observe that part of the steps described above can be avoided in the other two trust scenarios, since they are less demanding in terms of security.

In summary, by including: (a) A shared key between the EID-Holder and the MS; and (b) the RAs, our solution can achieve both EID authorization and RLOC verification, thus enabling dynamic and end-to-end secure map registrations.

IV. Performance Analysis

In this section, we evaluate the overhead that our solution imposes on the current LISP implementation. We will first introduce the experimental testbed that we used for carrying out the experiments. Then, we will examine the impact on the number of messages required to achieve secure map registrations in an end-to-end fashion. And finally, we will analyze the overheads caused by different types of signatures and encryption algorithms.

A. Testbed

The objectives of the experiments in the testbed are twofold. First, to evaluate the feasibility and reach of RLOC spoofings with the different LISP implementations. And second, to analyze the information exchanged among the different actors during the map registration process.

The testbed used for assessing and validating our map registration solution is shown in Fig. 4, where Fig. 4(a) describes the real network topology while Fig. 4(b) depicts its logical view. The testbed is built using GNS3 [11], and runs two different implementations of LISP, one with a Cisco IOS image for the xTRs in the LISP-Sites `EID-site1` and `EID-site2`, and another running OpenLISP [12] on the LISP-Site `EID-site3`. To complete the testbed we used the LISP-ALT mapping system, and for the purpose of this paper, we configured just one Map-Server (MS). In order to enable the LISP control-plane functionality in xTR_3, we configured Open Control-Plane [13] on top of OpenLISP. Moreover, the MS and the xTRs were configured with their respective EID-prefixes, including their shared keys.

In this setting, we were able to confirm that multiple RLOC spoofing attacks are feasible, and we were also able to asses

the performance and overhead of our solution to avoid such attacks.

B. Overhead in the Number of Messages

As currently defined in LISP, the map registration process consists of two messages. The first one is the *Map-Register* message from the ETR toward the MS. This message includes a claimed EID prefix, a set of RLOCs (each with its attributes according to the Traffic Engineering policy), and a block of Authentication Data (AD). The second message is an acknowledgement from the MS to the ETR, and it is actually optional. The AD in the first message provides a minimum level of security by validating the entire *Map-Register* message payload.

Although current LISP specification deems sufficient to send only two messages for the Map Registration, this approach provides only poor security guarantees over the whole process. In particular, the fact that the EID-Holder is not involved in the process, makes it susceptible to a number of serious attacks, which can undermine the whole LISP functionality. In our proposal, we require a higher amount of messages, though offering significantly improved and adaptable end-to-end security. As shown in Figure 3, in the worst case our scheme requires seven messages. More precisely, messages 5–7 are required to counter Man-In-the-Middle and coordinated attacks between the EID-Holder and the ETR, so they only apply for the untrusted scenario defined in Section III. The first five messages are sufficient in partially trusted scenarios, i.e., in trusted environments except between the ETR and the Map Server. Note that these include the final acknowledgement from the MS, which is optional in LISP. Indeed, in a completely trusted scenario, only the first four messages are needed to provide end-to-end security to the map registration process.

In the first stage of our proposal, the registration is initiated by the EID-Holder, which sends a *Service-Request* message towards the Service Provider (cf. Step 1 in Fig. 3). This is a new LISP control-plane message consisting of the following information: [EID prefix $\|$ $RLOC_{MS}$ $\|$ α]. Figure 5 shows the proposed *Service-Request* LISP message format—recall that α contains encrypted data. In the second stage, we keep the same message format as already defined in the specification of LISP *Map-Register* message. However, the AD field is replaced by α and the signature data of the message payload. Likewise, for the third stage, the acknowledgement message, namely, the *Map-Notify*, can keep its format as in the current specifications, but we insert the encrypted β and the payload's signature data on the AD field. Furthermore, for messages 6 and 7, we can reuse the *Service-Request* message format shown in figure 5.

C. Overhead caused by the Security Enhancements

The proposed solution produces some overhead on LISP's control-plane messages, increasing their size due to the extra information required to improve the security. We first analyze the new *Service-Request* message. This message includes the encrypted α information, whose size depends directly on the selected encryption algorithm. To compare the results in terms

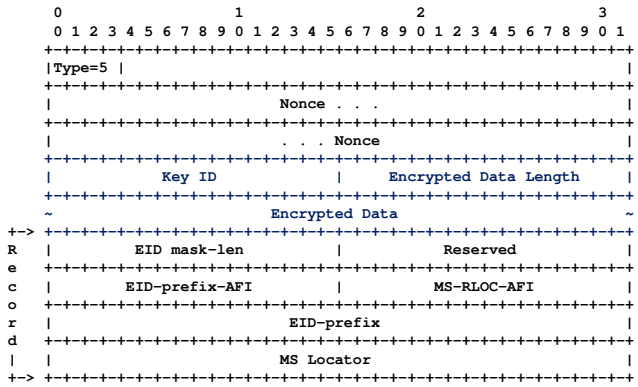
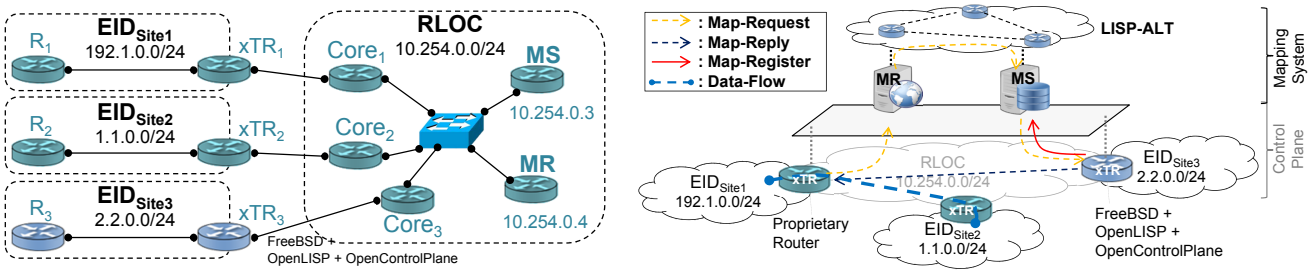


Fig. 5. Service-Request message format.

of the size necessary to encrypt the data, we encrypted α using different alternatives of up-to-date versions of the AES encryption algorithm. We selected AES since it is a broadly supported and an efficient algorithm; it can be implemented in hardware, and most importantly, it is considered secure. The results obtained are summarized in Table I, which shows the overhead incurred for each message, considering the encryption type, the signature, and their sum for computing the overall overhead.

In this evaluation, we considered an IPv6 EID-prefix (128 bits), 128 bits for xTR_{ID} and a timestamp of 64 bits. This adds up to a total of 320 bits (40 bytes) for α. Once encrypted, α grows to a size between 48 and 64 bytes, plus the Initialization Vector (IV) amounting to a total between 64 and 96 bytes depending on the selected AES key depth. On top of that, the new message has also to include the EID-prefix, and the RLOC of its Map Server ($RLOC_{MS}$). Again, assuming that we are using only IPv6 addresses, the estimated size of the new message is 56 bytes + α (cf. Fig. 5 for the whole message format). For the second message, i.e., the *Map-Register*, the overhead imposed by our solution includes α plus the signature of the message payload. All this information replaces the AD present in the legacy version. Thus, the impact on the overhead can be addressed by analyzing the total amount of bits that α plus the signature incur. Table I shows the size of the *Map-Register* message's signature data for different algorithms, as well as the total overhead including the size of α. For this analysis, we considered that AES-256 provides good enough strength security to encrypt α. Therefore, the total overhead oscillates between 144 to 168 bytes depending on the selected signature algorithm.

As for the third change in the control-plane messages required by our proposal, i.e., those that need to be applied on the *Map-Notify* message, it is sufficient to include an encrypted acknowledgement β destined for the EID-Holder, and the signature data of the message itself (cf. message 5 in Fig. 3). The size of β is *328* bits (41 bytes) including: 128 bits for the *OTP*, 128 bits for the xTR_{ID}, 64 bits for the timestamp and 8 bits for the acknowledgement. Thus, the size of encrypted β, analogously to the case of α, will depend on the selected AES algorithm. Furthermore, the size of the signature is the



(a) Topological View.

(b) Logical View.

Fig. 4. Testbed.

same as the one presented for the *Map-Register* message in Table I.

Followed by the *Map-Notify* message, the ETR forwards β towards the EID-Holder in message number 6 (cf. Fig. 3). The estimated size of this message, keeping in mind the format shown in Fig. 5, amounts to 56 bytes + β. In the last message, the EID-Holder confirms back the acknowledgement for the registration to the ETR. This message includes encrypted data consisting of the EID-prefix, the timestamp and the acknowledgement bit. The security overhead of this message is similar to the *Service-Request* message shown in Table I.

In summary, the total security overhead of our registration scheme fluctuates approximately between 952 and 1160 bytes, as compared to the 176 to 200 bytes for the current registration process in LISP. Moreover, a basic implementation of our solution including encryption/decryption as well as signatures and verification was developed and tested. The initial results reveal the average time to complete the proposed end-to-end secure map registration process to be 0.259 seconds, compared with the average time of 0.11 seconds for the current registration process—this result excludes the time required for RA verification. Clearly, enhancing the security has an associated cost, but the benefits obtained allow for a broader technological reach, especially in areas requiring mobility, where the users roam to foreign networks while keeping their original identifiers and sessions alive. With our lightweight solution, this can be achieved without the complexities and extra burden of tunneling across protocols and mobile technologies.

V. Ongoing/Future Work and Discussion

In this paper, we have presented a novel and adaptable approach for secure map registrations in LISP. Our proposal works end-to-end and covers both EID and RLOC authorizations, thus providing a framework to counter a variety of attacks against the control-plane, including RLOC spoofing. As we have shown, even in a completely untrusted environment, our security scheme requires only a few messages and produces low overhead. Furthermore, our approach leverages on the design and infrastructure already developed by the IETF's Secure InterDomain Routing (SIDR) WG for resolving the RLOC Authorization part, while presenting a potential adoption blueprint. We plan to make available our solution to the open source community in LISP, and we will examine its performance in larger topologies, including the impact on the Mapping System. We also plan to implement our scheme as a Software Defined Network (SDN) application, using Cisco's onePK [14] environment.

	Encryption		Signature			
Message	Algorithm	α/β (B)	Algorithm	Key (b)	Sign. (B)	Total (B)
Service Request	AES-128[a]	48				64[b]
	AES-192[a]	48				72[b]
	AES-256[a]	64				96[b]
Map Register	AES-256[a]	96	DSA-SHA-1	1024	48	144
			DSA-SHA-1	2048	72	168
			DSA-SHA-256	1024	48	144
			DSA-SHA-256	2048	72	168
			ECDSA-SHA-1-P256	256	72	168

[a] AES-CBC encryption Mode.
[b] Counting the size of the Initialization Vector (IV).

TABLE I
New Map Registration Process Security Overhead

Acknowledgements

The authors would like to acknowledge the support received and the valuable discussions had with Fabio Maino, Vina Ermagan, and Rodolfo Milito from Cisco Systems.

References

[1] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)," IETF, RFC 6830, Jan. 2013.
[2] F. Maino, V. Ermagan, A. Cabellos, D. Saucez, and O. Bonaventure, "LISP-Security (LISP-SEC)," IETF, Internet-Draft, Oct. 2012. [Online]. Available: http://www.ietf.org/id/draft-ietf-lisp-sec-04.txt
[3] "IETF Secure Inter-Domain Routing (sidr) Working Group." [Online]. Available: http://datatracker.ietf.org/wg/sidr/
[4] "IETF Locator/ID Separation Protocol (LISP) Working Group." [Online]. Available: http://tools.ietf.org/wg/lisp/
[5] M. Lepinski and S. Kent, "An Infrastructure to Support Secure Internet Routing." IETF, RFC 6480, Feb. 2012.
[6] M. Lepinski, S. Kent, and D. Kong, "A Profile for Route Origin Authorizations (ROAs)," IETF, RFC 6482, Feb. 2012.
[7] R. Gagliano, "A profile for endpoint identifier origin authorizations (IOA)," IETF, Internet-Draft, Mar. 2009. [Online]. Available: https://tools.ietf.org/html/draft-rgaglian-lisp-iao-00
[8] "American Registry for Internet Numbers (ARIN)." [Online]. Available: http://www.arin.net/
[9] "RIPE Network Coordination Center." [Online]. Available: http://www.ripe.net/
[10] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol," IETF, RFC 6810, Jan. 2013.
[11] "GNS3: Graphical Network Simulator." [Online]. Available: http://www.gns3.net/
[12] "The OpenLISP Project." [Online]. Available: http://www.openlisp.org/
[13] D. P. Chi, S. Secci, G. Pujolle, P. Raad, and P. Gallard, "An Open Control-Plane Implementation for LISP Networks," *IEEE INT. Conference on Network Infrastructure and Digital Content*, 2012.
[14] "Cisco Open Network Environment." [Online]. Available: http://www.cisco.com/go/one