Novel Architectures and Strategies for Security Offloading

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Arquitectura de Computadors

Diego Montero

Advisors: Dr. René Serral-Gracià
Dr. Marcelo Yannuzzi
Department of Computer Architecture
Technical University of Catalunya

A thesis submitted in fulfillment of the requirements for the degree of
Doctor in Computer Science

September 2017

to Isabel, Mireya, Josué and Pepe
. . . la familia feliz. . .

Acknowledgements

I would like to extend my sincere gratitude to the people and organizations who have supported me throughout this endeavor. The uncountable challenges, experiences and lessons I have learned, alongside the extraordinary people I have met are the most valuable memories that will endure in my life.

This thesis realization has been possible thanks to the unconditional support and guidance of my advisors Dr. René Serral-Gracià and Dr. Marcelo Yannuzzi. I really appreciate your encouragement and wise advices to pursue my goals and beyond.

I would also like to thank all the former and current fellow doctoral students at the ANA research group for their feedback, cooperation and of course friendship. My gratitude also goes to my fellow labmates for the stimulating discussions, for the sleepless working nights before the deadlines, and for all the fun we have had in the last four years.

Some organizations have played a key role supporting the development of this thesis. I would like to especially thank the *Secretaria de Educación Superior, Ciencia, Tegnología e Innovación del Ecuador*. I am also indebted to the Engineering Faculty of the University of Cuenca, Ecuador.

Words can not express how grateful I am to my parents and to my sister and brother. Their unconditional support, encouraging thoughts and empathy have deeply helped me throughout this journey.

Diego Montero, September 2017

Abstract

Internet has become an indispensable and powerful tool in our modern society. Its ubiquitousness, pervasiveness and applicability have fostered paradigm changes around many aspects of our lives. This phenomena has positioned the network and its services as fundamental assets over which we rely and trust. However, Internet is far from being perfect. It has considerable security issues and vulnerabilities that jeopardize its main core functionalities with negative impact over its players. Furthermore, these vulnerabilities' complexities have been amplified along with the evolution of Internet user mobility and its limited support.

In general, Internet security includes both security for the correct network operation and security for the network users and endpoint devices. The former involves the challenges around the Internet core infrastructure's control and management vulnerabilities, while the latter encompasses security vulnerabilities over end users and endpoint devices. Similarly, Internet mobility poses major security challenges ranging from routing complications, connectivity disruptions and lack of global authentication and authorization. These issues have motivated this thesis, which is structured in two main parts with a common traversal objective derived from user and device mobility. In the first part, we address some security vulnerabilities of the Internet routing system, while in the second part we focus more on the security protection of end users and devices.

The purpose of this thesis is to present the design of novel architectures and strategies for improving Internet security in a non-disruptive manner. Our novel security proposals follow a protection offloading approach. The motives behind this paradigm target the further enhancement of the security protection while minimizing the intrusiveness and disturbance over the Internet routing protocols, its players and users. To accomplish such level of transparency, the envisioned solutions leverage on well-known technologies, namely, Software Defined Networks, Network Function Virtualization and Fog Computing.

From the Internet core building blocks, we focus on the vulnerabilities of two key routing protocols that play a fundamental role in the present and the future of the Internet, i.e., the Border Gateway Protocol (BGP) and the Locator-Identifier Split Protocol (LISP). To this purpose, we first investigate current BGP vulnerabilities and countermeasures

with emphasis in an unresolved security issue defined as *Route Leaks*. Therein, we discuss the reasons why different BGP security proposals have failed to be adopted, and the necessity to propose innovative solutions that minimize the impact over the already deployed routing solution. To this end, we propose pragmatic security methodologies to offload the protection with the following advantages: no changes to the BGP protocol, neither dependency on third party information nor on third party security infrastructure, and self-beneficial. Similarly, we research the current LISP vulnerabilities with emphasis on its control plane and mobility support. We leverage its by-design separation of control and data planes to propose an enhanced location-identifier registration process of end point identifiers, while securely validating their respective authorizations. This proposal improves the mobility of end users with regards on securing a dynamic traffic steering over the Internet.

On the other hand, from the end user and devices perspective we research new paradigms and architectures with the aim of enhancing their protection in a more controllable and consolidated manner. To this end, we propose a new paradigm which shifts the device-centric protection paradigm toward a user-centric protection. Our proposal focus on the decoupling or extending of the security protection from the end devices toward the network edge. It seeks the homogenization of the enforced protection per user independently of the device utilized. We further investigate this paradigm in a mobility user scenario. Similarly, we extend this proposed paradigm to the IoT realm and its intrinsic security challenges. Therein, we propose an alternative to protect both the things, and the services that leverage from them by consolidating the security at the network edge. We validate our proposal by providing experimental results from prof-of-concepts implementations.

Resumen

Internet se ha convertido en una poderosa e indispensable herramienta para nuestra sociedad moderna. Su omnipresencia y aplicabilidad han promovido grandes cambios en diferentes aspectos de nuestras vidas. Este fenómeno ha posicionado a la red y sus servicios como activos fundamentales sobre los que contamos y confiamos. Sin embargo, Internet está lejos de ser perfecto. Tiene considerables problemas de seguridad y vulnerabilidades que ponen en peligro sus principales funcionalidades. Además, las complejidades de estas vulnerabilidades se han ampliado junto con la evolución de la movilidad de usuarios de Internet y su limitado soporte.

La seguridad de Internet incluye tanto la seguridad para el correcto funcionamiento de la red como la seguridad para los usuarios y sus dispositivos. El primero implica los desafíos relacionados con las vulnerabilidades de control y gestión de la infraestructura central de Internet, mientras que el segundo abarca las vulnerabilidades de seguridad sobre los usuarios finales y sus dispositivos. Del mismo modo, la movilidad en Internet plantea importantes desafíos de seguridad que van desde las complicaciones de enrutamiento, interrupciones de la conectividad y falta de autenticación y autorización globales.

El propósito de esta tesis es presentar el diseño de nuevas arquitecturas y estrategias para mejorar la seguridad de Internet de una manera no perturbadora. Nuestras propuestas de seguridad siguen un enfoque de desacople de la protección. Los motivos detrás de este paradigma apuntan a la mejora adicional de la seguridad mientras que minimizan la intrusividad y la perturbación sobre los protocolos de enrutamiento de Internet, sus actores y usuarios. Para lograr este nivel de transparencia, las soluciones previstas aprovechan nuevas tecnologías, como redes definidas por software (SDN), virtualización de funciones de red (VNF) y computación en niebla.

Desde la perspectiva central de Internet, nos centramos en las vulnerabilidades de dos protocolos de enrutamiento clave que desempeñan un papel fundamental en el presente y el futuro de Internet, el Protocolo de Puerta de Enlace Fronterizo (BGP) y el Protocolo de Separación Identificador/Localizador (LISP). Para ello, primero investigamos las vulnerabilidades y medidas para contrarrestar un problema no resuelto en BGP definido como Route Leaks. Proponemos metodologías pragmáticas de seguridad para desacoplar la protección con las siguientes ventajas: no cambios en el protocolo

BGP, cero dependencia en la información de terceros, ni de infraestructura de seguridad de terceros, y de beneficio propio.

Del mismo modo, investigamos las vulnerabilidades actuales sobre LISP con énfasis en su plano de control y soporte de movilidad. Aprovechamos la separacçón de sus planos de control y de datos para proponer un proceso mejorado de registro de identificadores de ubicación y punto final, validando de forma segura sus respectivas autorizaciones. Esta propuesta mejora la movilidad de los usuarios finales con respecto a asegurar un enrutamiento dinámico del tráfico a través de Internet.

En paralelo, desde el punto de vista de usuarios finales y dispositivos investigamos nuevos paradigmas y arquitecturas con el objetivo de mejorar su protección de forma controlable y consolidada. Con este fin, proponemos un nuevo paradigma hacia una protección centrada en el usuario. Nuestra propuesta se centra en el desacoplamiento o ampliación de la protección de seguridad de los dispositivos finales hacia el borde de la red. La misma busca la homogeneización de la protección del usuario independientemente del dispositivo utilizado. Además, investigamos este paradigma en un escenario con movilidad. Validamos nuestra propuesta proporcionando resultados experimentales obtenidos de diferentes experimentos y pruebas de concepto implementados.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AS	Autonomus System
BGP	Border Gateway Protocol
DDoS	Distributed Denial of Service
DNS	Domain Name Service
DNSSEC	Domain Name System Secure
DoS	Denial of Service
EID	Endpoint Identifier
ETR	Egress Tunnel Router
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
ICT	Information and communications technology
INR	Internet Number Resources
IoT	Internet of Things
IP	Internet Protocol
IPSec	Internet Protocol Secure
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6

ISP	Internet Service Provider
ITR	Ingress Tunnel Router
IXP	Internet Exchange Point
LISP	Locator-Identifier Split Protocol
NFV	Network Function Virtualization
OS	Operating System
PGP	Pretty Good Privacy
RIB	Routing Information Base
RLOC	Routing Locator
ROA	Resource Origin Authorization
RPKI	Resource Public Key Infrastructure
SDN	Software Define Networks
SSH	Secure Shell
TCAM	Ternary Content-addressable memory
VPN	Virtual Private Network
xTR	Ingress/Egress Tunnel Router

Part I: Introduction

Chapter 1

Summary and Road Map

Internet has become an indispensable and powerful tool in our modern society. Its ubiquitousness, pervasiveness and applicability have fostered paradigm changes around many aspects of our lives, impacting not only the way we interact and communicate but also entertain and work. Its evolution and worldwide adoption have reached levels far from initially envisioned. This widespread positions the Internet as a critical asset or service, over which we rely upon. However, Internet is far from being perfect. It has considerable issues that might disrupt not only its core functionality but jeopardize all its players. The increasing security issues have affected not only end users, but also crucial Internet core systems—e.g., exploitation over the Internet resources ownership, and its routing system.

The purpose of this thesis is to present the design of novel architectures and strategies for improving the security of Internet in a "non-disruptive" manner. Our novel security proposals follow an "offloading" approach. In general, Internet security includes both security for the correct network operation and security for the network users. The former involves the challenges around the Internet core infrastructure and its vulnerabilities. Meanwhile, the latter encompasses the needs to secure the information on the users devices and in transit across the Internet, as well as protect the devices of unauthorized access. The idea behind the offloading paradigm is to propose novel architectures and frameworks to further enhance the security protection while minimizing the intrusiveness and disturbance over the Internet and its players. To accomplish such level of transparency, our proposed solution leverages on well-known technologies, namely, Software Defined Networks (SDN) [45], and Network Function Virtualization [22]. In a nutshell, the former enables the decoupling and offloading of the control and management part from a protocol, while the latter embeds core functions to the network through the definition

and deployment of "network functions" over the user traffic, which are offloaded from the end-device.

Similarly, architectures such as Edge Computing [75] and Fog Computing [11] are leveraged to support the offloading of security protections out from the end user's device. These architectures propose the spread or enhancement of network edge nodes with computing and storage capabilities. In this context, the edge of the network refers to network nodes (e.g., points of presence) geographically distributed closer to the end users and devices. These architectures in concordance with SDN and NFV represent a powerful fabric for hosting and enforcing the end users' and devices' security applications.

This introductory chapter delineates the motivations and objectives that influenced this work. The main contributions are described next, followed by an overview of the manuscript structure.

1.1 Motivations

The proliferation and increasing dependence over Internet's services and applications have impacted and shaped our Society. The "network of networks" evolution has positioned itself as a vital asset to our daily lives. Its development has promoted new services in a broad variety of fields, reaching from the ways we communicate, do business, and get entertained. On top of this, it has fostered the creation of a vast market of opportunities, yielding a competitive environment for innovation. Its decentralized, openness, and modular design principles have influenced its great acceptance and expansion worldwide. Alongside, more dependency over the Internet means the requirement of higher security protection. However, from its beginnings Internet security has been a second-class consideration, and in some cases not considered at all. This stems from the original design which assumed a totally trusted multi-party ecosystem. From there, securing the network has followed an "ad hoc" feature approach.

Internet security encompasses the protection of computer and network systems from both the damage or manipulation of information, and the disruption or forgery of its services and infrastructure. Security is underpinned on the principles of "Availability", "Integrity" and "Confidentiality" (the "IAC" triad). While confidentiality consists of restricting the access to those granted, integrity assures that the information accessed has not been altered by an unauthorized party. Therein, availability ensures that information and resources are available to those who need them in an appropriate time-frame.

In general, security in Internet depends on the compound security measures enforced in each subsystem, e.g. at the Application level, or network stack layer. On top of this,

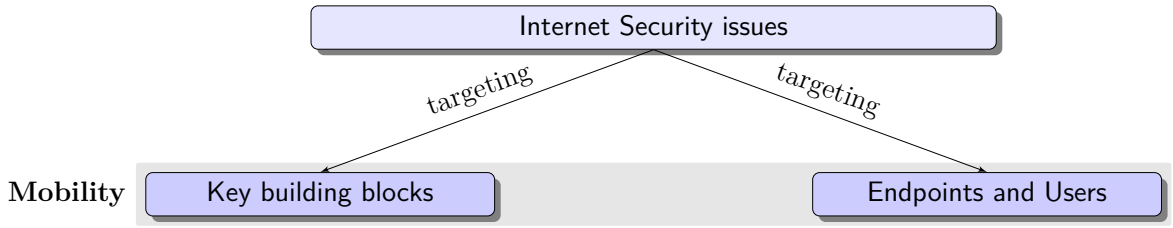


Fig. 1.1 Problem definition and motivations

any security solution proposed at any level will be far more effective and appropriate only when all the participants involved apply them, in a collaborative approach seeking to better secure the network as a whole.

The Internet security issues that motivated this work can be better contextualized from the abstract overview shown in figure 1.1. These motivations come from two perspectives, each targeting problems at different levels. On one hand, there are key security issues over core Internet building blocks, which affect its correct functionality in terms of network infrastructure reliability and safety. These problems have aroused as the consequences of some Internet design principles, including the self-describing datagram packet, the end-to-end arguments, diversity in technology, global addressing and a trusted ecosystem [14]. The reality of today's Internet has evolved to an interconnected world of different stakeholders, each with different interests—possibly adverse to each other—and with the general tendency of each party to favor theirs.

On the other hand, let us consider the perspective of an Internet user (the second perspective as depicted in figure 1.1). For her/him/it, Internet enables the communication with any addressable end-point within the network realm. This interconnection in the public "wild" Internet, by default, is assumed to be insecure. Any third-party infiltrated in the network can listen and eavesdrop the traffic information. The end points, or a third element in between (e.g. a gateway), are in charge of securing the user connections and the information exchanged. Thus, security at the end-points continuous to be an evolving Internet challenge.

Following the Internet architecture, the security at the end points also consists of the compound security mechanisms at different levels of the network stack. Starting from the top, Internet applications implement either customized or standard security mechanisms to protect their users communications and exchanged information. For example, secure applications such as SSH, HTTPS, and DNSSEC lie under the Internet application layer, which assume that Internet services are untrusted. One level down, the transport layer security targets the security of every socket connection opened by an application (e.g., TLS). Meanwhile, the security at the network level protects all application traffic over an

IP network (e.g., IPSec). Finally, the security at the link layer is specific of the employed link technology—for example WPA security in WiFi. This per-layer security enables the end-points and users to leverage and enforce a wide spectrum of security options and levels, accordingly to their specific needs. An important remark, security comes at a price, be it either extra resources' consumption (e.g., processing, or power), or time impact in the communications, due to its intrinsic overhead. Furthermore, thanks to this layered approach, the security of some layers can be implemented outside of the end points, transparent to the end users. For example, IPSec can provide security between a pair of gateways interconnecting two private networks.

The plethora of Internet security options at the end-points foster a flexible, adaptable framework to better secure the end users. However, it also increases the burden on the proper selection of what type of security is required and at which level. Every security countermeasure comes with many advantages, but also alongside with trade-offs and impacts. All this complexity is handled at the user's applications and operating system, which ultimately impact over the user experience.

Now, due to the fast proliferation of massive, affordable devices (e.g., a single user surrounded by different Internet-enabled devices, such as smart-phone, tablet, and laptop), a new complexity vector around security has been introduced. The ultimate goal around end user security targets the protection of the user information properly adapted to the device used. A change of paradigm from a device-oriented security toward a user-centric security is envisioned.

Similarly, the rapid growing of resource constrained devices with Internet capabilities represents a complex battlefield for security threats. The Internet of Things extends the network reach to the physical world by enabling their interaction and actuation, even to the point of removing the human factor (e.g. initiatives like Device-to-Device communications). These devices are designed and streamlined for a set of specific purposes, which may entail either the generation of information (e.g. a temperature sensor or a video camera), or the actuation over a physical element (e.g. a light-bulb switch).

Whilst the fast proliferation of these devices foster new markets driven by the huge amount of generated data and enhanced control and actuation over their surrounding world, there is a major trade-off security wise. These devises may require enforcing certain level of security protection, depending on the criticality of their function. However, given their resource constrained nature, their security may be customized to reduce its overhead, or require the action and protection of an external entity, e.g. this functionality is usually undertaken by a gateway device. Thus, securing this wide universe of devices

impose a huge challenge in terms of how to protect both the devices and the network from a wide threat surface. These arguments support the idea to research new paradigms of security offloading from these devices, alongside with new networking and computing architectures capable to scale with them at the network edge.

Below, we shall further elaborate each perspective and the reasons behind that motivated this research. Figure 1.2 depicts in a nutshell our research road-map, addressing both perspectives detailed above, and further highlights the problems addressed at each one, respectively.

Motivations targeting key Internet building blocks

The lack of by-design Internet security measures in some of its fundamental building blocks, e.g., the domain-name translation service, and the inter-domain routing protocol, have revealed mayor network service vulnerabilities with unprecedented consequences. Take into consideration the cases of Internet outages like the Youtube service breakdown [73], alleged Chinese Telecom traffic hijacking [47], massive routing leaks in Malaysia [8], or the India Internet traffic hijack [9], to mention a few. These attacks have shown the exposure of one foundational sub-system, crucial for the inter-domain network traffic routing, i.e., the Border Gateway Protocol (BGP)[71].

Some of these large-scale Internet disruptions might had been the result of either a misconfiguration, or a direct attack targeting the hijacking of Internet traffic. For example, in an attempt to block local access to YouTube (by local we mean nation-wide), *Pakistan Telecommunication Co. Ltd.* mis-configured the BGP prefixes and routes advertised toward its providers, leading to a massive YouTube service disruption. The Internet routing system dynamically learned that the Pakistani ISP now owned the routes and network resources to reach YouTube. This routing information was further spread by its providers, as they lacked the mechanisms to filter out this "incorrect" routing update. Thus, the Internet traffic toward YouTube was wrongly steered, overloading the deceiving ISP's network capacity. Ultimately, YouTube resources became unreachable for a period of time.

The inter-domain routing protocol was logically designed as a set of trusted parties or domains sharing routing information with each other in a totally distributed system—an Internet player is referred as Autonomous Systems (AS). This assumption makes really difficult for any AS to verify the shared information by its peers, enabling the spread of both deceit route injections (by any transit AS), and the origination of unauthorized network routes (by any AS claiming illegitimate prefix ownership). The by-product of spreading this rogue routing information over the Internet ranges from inadvertently attracting network traffic toward a dead end, or enabling a man-in-the middle attack. To

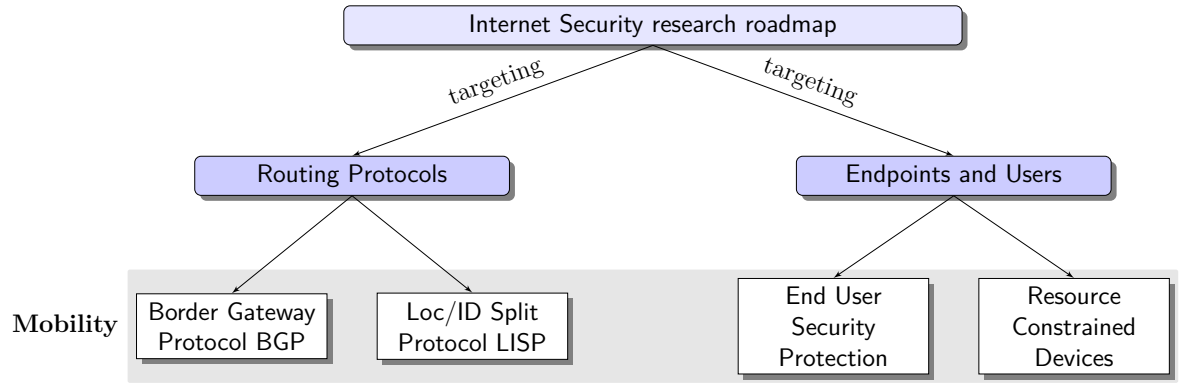


Fig. 1.2 Overview of the Targeted Security Problems

countermeasure these problems, several security proposals based on either fully replacing BGP, or extending it through security patches have been introduced. However, none of them have promoted a wide adoption due to their complexity or heavyweight disruptions over the current BGP deployment, as well as their poor incentives toward initial adopting players. These arguments are the basis behind our proposal to enhance BGP security causing the minimum disruption by offloading it to an external entity.

Solutions proposed by the SIDR IETF working group like the Resource Public Key Infrastructure (RPKI) [71], Route Origin Authorizations (ROAs) [50], and BGP Security (BGPSEC) [51] target to secure the BGP protocol by enabling mechanisms to validate the Internet resource allocation and authorization verification, the correct route origination verification, as well as the AS-path route verification. RPKI and ROA follow the offloaded paradigm as a new external infrastructure entity is devised, which main purpose is to cryptographically support the validation of resources authorizations. In contrast, the BGPSEC proposes major changes that affect not only the inner protocol but also imposes new requirements such as in-line chain verification of cryptographic signatures, alongside with hardware upgrades. As a result, BGPSEC has faced the most resistance from the community.

Within this context, our research interests are aligned with a not so well-defined, policy related security issue in BGP. This security flaw is effective even when all the previous BGP security measures are in place. The reason behind is that this vulnerability exploits an AS's internal routing policy by violating the exported routes announced to a neighbor AS. This seemingly trivial problem has proven to be hard to solve, as each AS maintains its export policies rules secret. We shall refer to this problem as "Route Leak" and formally define it in section 4.2.

Internet is also facing major issues regarding its routing scalability. The design principle around Internet addressing considered a global addressing scheme and namespace, i.e.,

the Internet Protocol (IP). This principle works well in a highly hierarchical, aggregative routing abstraction, which targets the optimization of the routing table size. However, due to new technical factors such as multi-homing, traffic engineering, mobility, as well as the lack of security within BGP, the Internet routing table size has experimented an exponential growth [68]. Around these factors, the mobility factor contributes to this increase as new, more specific routing prefix advertisements are required when the prefix owner is allowed to maintain its prefixes, independently of the Internet network addressing hierarchy (e.g., the case when an end user owns a public IP address and switches Internet providers). Similarly, in an attempt to prevent prefix hijacking within BGP, some Internet autonomous systems are advertising larger, more specific prefixes (e.g., a larger set of /24 prefixes instead of the aggregated ones).

Another problem, related to the above, stem from the semantics overload in the Internet addressing space. Currently, an IP address represents not only the *Identification* of an end-point, but also its topological *Location* within the network hierarchy; the former identifies the "who", while the latter defines the "where" (location in the addressing network topology) in an end-to-end IP communication. This semantical-split principle has fostered new routing proposals such as the Locator/Identifier Separation Protocol (LISP) [23] and the Identifier/Locator Network Protocol (ILNP) [4]. These new routing proposals introduce a clear distinction regarding this semantic overload, while keeping a dynamic control and routing system to properly steer the traffic through the network. Their advantages include a reduced public routing table, multi-homing, and better support for mobility. On this work, we shall focus our research efforts over the LISP protocol as a mobility enabler, and the security issues around its control plane.

The LISP protocol was initially designed to address the Internet routing table explosion problem. This proposal is based on the semantic-split idea which renders two different addressing spaces, each serving specific purposes. These two addressing spaces, referred as routing locator (RLOC) and endpoint identifier (EID) namespaces, map the core-edge network architecture, respectively. This inserts a level of indirection, requiring a global biding register to tie both spaces. Whilst having two different addressing spaces allows the stabilization of the core Internet routing table size and foster different and scalable addressing schemes at the network edge, it comes with the challenge of maintaining a global mapping binding register alongside with its security concerns.

Apart from its initial designed purpose, LISP has proven to be an enabler for other factors such as multi-homing, traffic engineering and mobility within the IP networking arena. However, this proposal has similar issues as BGP in terms of both network resources allocation and authorization, as well as the dynamic update of locations-

identifiers bindings. For example, the mobility use case arises different challenges from two perspectives: a) how to verify the correct allocation and authorization of each addressing space, and b) secure update of the addressing mappings in the global biding register. Currently, LISP provides a simple security control plane (LISPSEC [55]) which focuses on its functional integrity. However, the proposal lacks of security methods to assure resource authorization, e.g., prefix ownership and dynamic authorization of location-identifier bindings. We shall research these challenges and their similarities with the BGP ones, previously introduced.

Motivations targeting end points and users

The proliferation of Internet and its ubiquitous access fostered by both, the advances of Internet-capable devices and the evolution of network access technologies towards a converged network paradigm, have positively impacted over the end users and the services provided in general. However, new devices connected to the Internet mean more endpoints to secure and, conversely, more targets exposed for attacks. The burden and complexity of properly securing them and keeping the network resilient to bad players seems to be a daunting task to overcome.

Any Internet enabled end point device may represent either a target for an attack, or the source of an attack. The best way to protect a device, network and system from the Internet is by unplugging it from the open network. Whereas this rather extremist case is applied to certain critical systems, devices and networks, it is not the case for a normal Internet user, his services and networks. Therefore, the security around the edges of the network, which includes both the end points and inherently its users, is key to preserve a secure Internet environment.

Internet enables the end-to-end communication paradigm, which implies that any two end points can establish a communication channel and exchange information. Prior to reach this level, any device first is required to join a network at the physical and link level, either by a wired or wireless technology. This represents the first security barrier to pass (e.g a wireless password means the network has enforced a link protection level). Now that the device is attached to a local network, it will be able to communicate with any other local devices attached to the same network, and to the Internet through a default gateway. The security concerns regarding this local network access is out of the scope of this work. This work focuses on new paradigms to provide security protection to end points that already have Internet connectivity, or at least are attached to a network and reach a gateway. The idea is to propose novel paradigms and architectures for enforcing the protection for the end point at the network edge, the closest to it. The edge of

the network might be defined as the point where a user or group of users (a network) concentrate and share a common gateway.

Any security protection measure enforced at the end-point comes not only with additional complexity over the end user, but also an extra burden in terms of resources requirements (e.g., processing, storage, or data transmission)—and the consequent impact over performance and energy efficiency, critical in the case of battery-enabled devices. This problem not only affects end user devices such as smart-phones, tablets and laptops, but also to a plethora of resource-constrained Internet-enabled devices in the realm of Internet of Things (IoT).

From the perspective of end-users, there is a critical problem around having a transversal, simple to define and manage security protection. The complexity increases along with the number of devices normally a user employs daily. Each device poses particular requirements regarding security protection, exposing users to the complex hurdle of setting up device-tailored protection countermeasures. This problem is further exacerbated when a device is shared between multiple users (e.g., a family tablet). The setup of distinct security profiles, policies and protection rules for the different users of a terminal is far from trivial.

End users are exposed to devices with different architectures (e.g., Intel or ARM) as well as different capabilities and operating systems (e.g., Android, Windows, or Linux). The appropriate protection tools may not be available for all platforms, or lack the support for a specific one. As a result, the most common practice is to install different security applications on the various devices—or simply rely on the default protection means provided by the operating systems. Let us assume for a moment that users would like to have the same security policy and exactly the same protection level enforced on all of their devices. To achieve this goal, the user would need to understand the configuration details of each device, which typically involves the setup of different security applications on different platforms. For non-technically savvy people, this turns out to be an impossible hurdle to overcome. As a result, most Internet users suffer from wide variations in their protection levels, and this problem is exacerbated as the number of devices per user grows.

The problem scenario described above stem from a device-centric security paradigm. The main drawbacks around this paradigm, from an end user perspective, include the need for dissimilar installations of security applications in different devices due to their different platforms, and the problem of non-uniform protection due to the difficulties in the configurations needed. These, problems have motivated the research of new security

paradigms that focus more on the user protection with a homogeneous, device agnostic and policy like oriented security definition.

A similar problem but at a different growth scale is happening with the Internet of Things (IoT). The explosion in the number of Internet-enabled devices (things) like thermostats, home appliances, light-bulbs, fitness trackers, door locks and medical devices have enabled the creation of new, added value services. However, each of these devices represent a new target for attacks. Consider also that, in an attempt to be competitive, IoT device manufactures seem to sacrifice security in favor of price and time-to-market. This trade-off gets amplified as these devices lack of user-friendly security updates and mechanisms to apply patches, as well as they are meant to last for years—they are not replaced as often as our laptops or phones, thus users ignore their security flaws and consequences as long as they work as expected.

The wide diversity of IoT devices in terms of their resources and capabilities influence different surface attacks. This diversity may hinder the enforcement of security protections at the level of device. However, we shall keep in mind that many of these devices have specialized functions and predictable behaviors. Monitoring them in an aggregation point would help not only to detect anomalies which hint potential attacks or compromised devices, but also address the IoT security in a scalable approach. However, the requirement of a gateway per each device or set of devices induces to vertical solutions, which hinders both the security complexity and the interaction between them. The commodititation of networking and processing virtualization at the network edge provides some advantages and opens the possibility to better approach the verticals problem. Fog Computing and Edge Computing propose a different network and processing architecture, which extends them along the continuum datacenter-edge network, with the common aim to locate their services closer to the user. We shall research these new architectures and their advantages aligned with our purpose of providing an offloading security to IoT devices.

The above described problems reveal a clear necessity for new approaches, architectures and solutions with the objective to better approach the Internet security continuum. We propose a different paradigm which seeks to address inter-related vulnerabilities described above, which ultimately seeks to better protect both the network and its players.

1.2 Objectives

The general objective of this thesis is to study the design of novel architectures and strategies for securing different segments of the Internet under the “offloading” paradigm. We approach this goal from two, related, perspectives: a) Internet core building blocks

security, and b) endpoints and end users security protection. The more specific objectives in this thesis include:

1. To analyze the BGP vulnerabilities and the up-to-date proposed solutions, while understanding the reasons behind the lack of security in this fundamental routing system.
2. To design and prototype a novel architecture to supports the offloading of the BGP security with minimum changes in both the protocol and its adopters.
3. To study and propose a solution for the BGP "Route Leak" vulnerability following the offload security protection.
4. To study the LISP protocol and its vulnerabilities in the context of End User mobility.
5. To design and prototype an architecture that partially leverages the BGP security to improve the LISP map registration.
6. To design and prototype an architecture that provides a user protection by offloading the execution of common security applications from user devices into the network edge.
7. To assess the devised architecture solution in a user case scenarios with and without considering end user mobility.
8. To design and prototype an architecture that provides enhanced offloaded security to resource-constrained devices at the network edge.

1.3 Adopted strategies and Contributions

1.3.1 Key Internet routing vulnerabilities

Internet vulnerabilities at its core routing system not only hinders the security of all its players, but also jeopardizes its stability, integrity and availability as a data communications infrastructure. BGP is the routing protocol that sustains the Internet together, and rules how the packets are steered across the public network. It is the standard that allows the exchange of routing information between Autonomous Systems, enabling all the parties to create a route map for data communications. More precisely, BGP allows an AS to share and discover routes to IP destination prefixes to/from its neighbors, which in turn learn the information from their neighbors. This gathered information assists the AS's routers with the routing decision process of data traffic. The issue here is that bogus routing information can spread easily and almost instantly

across Internet, as BGP lacks of the mechanisms to properly validate it. This problem stems from the BGP design assumption of trusted parties—BGP runs on the honor system. Thus, securing it has been constant battle as its vulnerabilities enable a broad vector of attacks, including (but not limited to) Denial of Service, traffic hijacking, or eavesdropping.

The Internet community cognizant of BGP's vulnerabilities have proposed a number of solutions focused on securing the route advertisements information shared among ASes. Past proposals such as S-BGP [44], soBGP [90], psBGP [65], and Internet Route Verification (IRV) [34] as well as the more recent ones devised by SIDR WG—RPKI [12], ROA [50] and BGPSEC [51]—share a similar design principal, i.e., all of them advocate for new extensive architectures which empower ASes to validate shared routing information, as well as the honesty of the advertisers.

However, an important non-technical consideration regarding the business interplay between Autonomous Systems seems to be absent from these proposed BGP security solutions, which unveil a new, apparently trivial vulnerability. The relationships between ASes are driven by economic motivations, i.e., the cost of forwarding traffic depends on the type of relationship between the neighboring ASes. Two relevant relationships here are *customer-provider*, and *settlement-free peer-peer* relationships. The former represents the formal agreement that all routes learned from the provider will be further advertised to all provider's neighbors (it includes its providers, customers and settlement-free peers). In contrasts, the latter agreement dictates that all the client routes of a peer are advertised to the other peer, and vice versa.

The violation of these policies entail a security issue, imperceptible even with the latest security proposals in place. This thesis shows how effective this kind of attack can be and proposes a solution to heuristically detect and prevent it by an AS using only local information. Also, it follows the offloading security paradigm with the objective to be the least intrusive over both the BGP protocol and its implantation.

Alongside with the vulnerabilities identified in the core routing system of Internet, we seek to also address other problems in the BGP routing system derived from factors such as multi-homing, and mobility. The problems introduced by them affect not only BGP, but also IP in general. The LISP protocol has proven to be an enabler for these factors, as it inherently enables the separation of routing concerns into two independent addressing schemes. One of them is deemed as the core addressing scheme, which may represent today's core Internet routing with BGP, while the other represent the different edge networks. These two addressing schemes are bounded together by a key LISP building block referred as the location-identification register (i.e. the Mapping system). It allows

not only to verify the resource allocation authorization of edge network resources, but also to validate the new location address on the core addressing space prior accepting the new bindings. Our strategy is to introduce a novel security LISP address allocation and authorization for both schemes, leveraging from some BGP's solutions, e.g. RPKI and ROA.

The registration of an edge network (either one device or a set of them) in the mapping register requires the specification of the current core address (location). This process involves two parties, which might require to verify the authenticity and correct claim of proper resource ownership, respectively. From the perspective of the mapping register, it not only authenticates and validates the edge network registrations request, but also might require to validate the authorization of the provided location address.

This registration process in LISP currently is secured by a shared key between the border router and the mapping system. This key allows the validation of the claimed resources by the originator. However, there are no mechanism to validate the authorization of the location addresses included in the registration. Furthermore, we propose a third party in this process, which represents a mobile end user. This new actor shall interact with the other two in the process of registering his network resources, while also validating the correct location at the network border representing him.

1.3.2 Endpoint devices and End User security protection

The security protection of end devices represent the first line of defense against attacks toward other end points. The correct enforcement of security measures at them reduces the risk of an insecure network. However, the current evolution of Internet-enabled end points have marked a notorious tendency with respect to the hurdle the security protection becomes. On one side, we have the group of end-user security, which considers the current phenomena regarding the number of devices employed by a user to access the Internet. The security protection measures available at them follow a device-centric approach, where each device due to their specificities in terms of hardware, operating system and others impose stringent requirements to the protection solutions. We propose a novel change of paradigm toward a user-centric security protection, which focuses on the enforcement of security protections in a device agnostic approach. We shall further introduce and discuss the new network and infrastructure architectures leveraged to achieve this approach in a security offloading fashion.

A similar problem at a different scale in terms of end points represents the security protection of IoT devices. In this scenario, due to the wide range of dissimilar Internet-enabled devices characteristics, their protection at the per device granularity seems a

next to impossible hurdle. However, it is important to remark that these devices are usually function-specific, with a predictable behavior, which can be monitored by an external entity (e.g. a gateway). Also, these external entities can protect a group of nearby-devices, which ultimately helps with the security protection at the scale of IoT.

1.4 Manuscript structure

This thesis is organized in four parts. The below sequel briefly describes the contents of each part, while introducing the topics addressed in its corresponding chapters.

PART I

This section is comprised of two introductory chapters, which help position the reader within the context of this thesis.

Chapter 1 provides a brief summary of the motivations, objectives and adopted strategies.

Chapter 2 describes in short the necessary background to understand the issues addressed and the reach of the solutions proposed along this thesis. A comprehensive list of references is provided in this part for the reader who wants to get deeper into the details regarding the Internet vulnerabilities.

PART II

This part consists of two subparts, each dealing with BGP and LISP vulnerabilities. First, the Internet routing vulnerabilities at present and the defense efforts proposed by the community are addressed in Chapter 3, while Chapter 4 presents the analysis, design and test of a novel security architecture with emphasis on a BGP vulnerability named "Route Leak". On the other hand, Chapter 5 describes the current LISP control plane vulnerabilities and the proposed security measures, while Chapter 6 delineates our analysis, design and test of a novel end-to-end map registration process.

Chapter 3 reviews the main BGP vulnerabilities from a threat model perspective. The chapter deepens in the analysis of some of the key threats that have caused major Internet outages. Therein, the defense efforts for securing these BGP vulnerabilities are described, while emphasizing on an open problem regarding the export routes policy

violations.

Chapter 4 proposes a novel solution to counter the Inter-domain "Route Leak" vulnerability. Therein, the advantages of our proposed solution are described, while the techniques to offload this security protection are delineated. This chapter also reports the results obtained from the experiments over our detection technique.

Chapter 5 reviews the main LISP control plane vulnerabilities, with emphasis on the dynamism for end user mobility. Therein, the proposed solutions for EID-to-RLOC registration and EID-to-RLOC lookup query are described.

Chapter 6 proposes a novel LISP Map Registration approach with emphasis in end user mobility. Therein, the advantages and the different scenarios our solution consider are delineated, while the technique to provide and end-to-end secure map registration with EID and RLOC verification is described. Finally, the evaluation results from different experiments are reported.

PART III

This part addresses the Internet security segment from the perspective of end users and end devices. First, the device-centric protection problem is described along with our proposed change of paradigm, which focuses more on a user-centric security protection. Therein, the architecture design of our user-centric offloaded security protection is delineated along with its main components (Chapter 7). Our proposal has been devised to support user mobility, while maintaining the offloaded security closer to the user location. Chapter 8 describes this use case and presents some insight from the experiments results obtained from a real proof-of-concepts implementation. Finally, Chapter 9 describes our offload security approach in the realm of IoT.

Chapter 7 reviews the end user vulnerabilities derived from the current device-centric security protection. Therein, this chapter outlines our proposal for a user-centric protection approach while highlighting the main components of its architecture. It also reviews the open challenges around the IoT security with special focus on the device diversity, and the trade-offs imposed by the security and the consumed resources.

Chapter 8 addresses the mobile use case within the user-centric security approach. This chapter presents the mobility considerations and the Trusted Virtual Domain

(TVD) state migration strategies to support the user mobility. The experimental testbed description, the results and insight obtained are fully reported.

PART IV

This final part summarizes the main achievements, insights and conclusions drawn from this thesis and analyzes future lines of work.

Chapter 9 highlights the main conclusions and achievements drawn from this thesis.

Chapter 10 proposes different research lines and perspectives for extending the reach of this thesis.

Chapter 2

Background

This chapter objective is to introduce the general context around this thesis and our approach to the complex "Security" concept. First, we describe the security continuum in the Internet, and position our strategy toward the different challenges we target. Next, the basics of two Internet network routing protocols are described, specifically the BGP and LISP protocols, and briefly highlighting their shortcomings. Meanwhile, from the perspective of end-points and end user Internet security, a high-level description of the current protection models applied to them and its challenges are drawn.

2.1 Internet and the Security continuum

This section introduces the concept of Security around both the Internet and this thesis context. A brief overview of what Security means in the context of this thesis, and which are the threats that the Internet core as well as end-points and end-users face.

Internet security in general is a complex and broad research topic which covers a wide security spectrum of systems, services, networks, devices and end users. Therein, Internet security encompasses the protection of computer and network systems from both the damage or manipulation of information, and the disruption or forgery of its services and infrastructure.

A computer network, as the Internet, enables the interaction and exchange of information between their members. In an ideal world, all the members in this network are trusted and play fair, i.e., there is no greedy incentive for anyone to attack or disrupt neither the network nor deceive their counterparts. This utopia is far from real as, by the moment a machine is plugged in, there are an overwhelming number of ways its use might deviate from its intended purpose. This deviation is a malfunction, which could stem from an internal, unexpected error or accident. However, when this deviation is caused

by an adversary through any means, this malfunction becomes a "security" problem. The motivations behind can be wide and diverse, whether to cause harm, profit, or raise people awareness, to mention a few (e.g., stealing relevant information, impersonation, service disruption, protests, spread of viruses or malware, etc).

A common misconception around "security" is the understanding of being isolated from danger, e.g., a computer system or network defined as secure wrongly suggest they are free of dangers. On the contrary, "security" is a concept associated with the presence of an adversary. In the Internet and computing world, there is no perfect system devised without mistakes, vulnerabilities or errors. Thus, any detected problem only becomes a security issue or threat if an adversary exploits it, seeking to gain something. Similarly, this is an iterative game where the defender and the attacker improve their techniques to protect and exploit, which might results in a never-ending process as long as the perpetrator deems valuable the effort.

To elaborate, consider the Internet disruption over YouTube services in 2008. In an attempt to block local access to this popular service, *"Pakistan Telecommunication Co. Ltd."* unilaterally advertised ownership of the routes destined to YouTube toward its customers, seeking to block their access [73]. However, this (mis)configuration reached its Internet provider, Hong Kong-based *"PCCW Telecom"*, and further propagated. These new routes were believed by routers around the globe, diverting the YouTube traffic toward the Pakistan ISP, and consequently taking YouTube off-line for a considerable period of time. This incident was the result of a misleading configuration, which was not properly filtered by neither the originator nor its providers. Whilst this misconfiguration was not intentionally targeting YouTube, it can easily become a security issue if we take the same situation and change the "misconfiguration" intention with a "hack" or "attack" name.

Similarly, more recent incidents of country-wide Internet censorship include cases in Egypt, Libya, Syria, and Iraq. All these countries have leveraged their control over state-run Internet service providers (ISPs) to shut down Internet connectivity. These events have ignited technical discussions regarding the fragility of current networks. In fact, a number of different techniques have been used to trigger these self-induced attacks, including the withdrawal of Border Gateway Protocol (BGP) routes, Internet Protocol (IP) filtering, IP prefix hijacking or domain name system (DNS) hijacking [17, 74].

Internet security principles stem from the *Information Security* ones, which include *confidentiality*, *integrity* and *availability*. Confidentiality consists of keeping data private and accessible only to legitimate users. This private information not only includes personal data, but also all transactional data. The information relevance relies on its

value to the people, thus protecting its confidentiality continues to be a paramount goal. For example, cryptographic encryption has been the main mechanism employed to achieve this end.

Integrity assures that the system and the data in it have not been compromised, tampered or improperly altered by an unauthorized party. It encompasses the consistency, accuracy and trustworthiness of data over its life cycle. Examples of integrity means include cryptographic checksums, which ultimate end is to help a receiver to verify whether the information was altered while in transit.

The availability principle consists on being able to use the system as anticipated. It includes the proper maintenance of hardware and software upgrades, seeking to maximize the system uptime. This principle becomes a security issue when an attacker exploits the lack of availability by some mean. For example, consider the case of Distributed Denial of Service (DDoS) occurred against the *Dyn company* [92]. Part of this attack impacted on the availability over its DNS service, which ultimately affected some major websites. Similarly, malware attacks known as "ransomware" [81] target to completely block a victim system and data, threating with completely wiping them out in exchange of a ransom payment.

The security aspects around the Internet are not only technical issues, but they extend also from organizational, legal, economic, political and social as well. Most importantly, any security mean has inherent limits, trade-offs, and incur on non negligible costs—not only economically but also costs in time, convenience, capabilities, liabilities, liberties and more. Furthermore, the responses against security threats vary according to the type of vulnerability, which can affect the integrity, confidentiality, or availability. This is the reason why security is so complex, costly and in constant evolution. All in all, the naive idea of "absolute security" in Internet is a utopia.

2.1.1 Threats, Vulnerabilities and Attacks

Here, we present the definitions of some terms that shall be used throughout this manuscript.

- **Vulnerability**: a vulnerability is an inherent weakness in the design, configuration, implementation, or management of a network or system that renders it susceptible to a threat. Vulnerabilities are what make networks susceptible to information loss and downtime. The Internet is a complex system probe to undiscovered vulnerabilities, as it is every network and system.

- **Threat**: a threat is anything that can disrupt the operation, functioning, integrity, or availability of a network or system. This can take any form and can be malevolent, accidental, or simply an act of nature.
- **Attack**: an attack is an instantiation of a threat which is caused by a specific attacker with a specific goal in mind and a strategy for reaching that goal.

The threats against the Internet, and any computing system, are better understood from a threat model approach. A threat model identifies possible actors and their motivations against a system, helping to better understand the possible vulnerabilities as well as the defensive actions to thwart possible attacks. Therein, an attack is an instantiation of a threat scenario which is caused by an attacker motivated by specific goals, and prepared with strategies for attaining them. An attacker might seek to steal data, obtain and misuse credentials, hijack resources, or disrupt services. By doing any of them, a skilled adversary could cause a lot of damage, as the stolen data might represent highly classified and confidential information, or by using the credentials, an escalation of rights within a system might allow an attacker to modify code and data, access and control of critical infrastructure. Similarly, the hijacking of resources might result in service disruption, interception of information and spreading of rogue information (e.g., the Pakistan telecom case previously described.).

The vulnerabilities, threats and attacks ideas described above shall serve as the basis concepts for the security study of this thesis, around the network protocols and the end users and end points. Our approach to each shall include first the description of the current vulnerabilities and threats (by a threat model), and next our proposal to tackle some of them.

2.2 Internet Routing

The Internet is based on the idea of inter-connecting multiple, independent networks of rather arbitrary design. It began with the ARPANET as the pioneering packet switching network, but soon included packet satellite networks, ground-based packet radio networks and other kind of networks. The key principle behind this diversity is rooted in an open networking architecture, where the choice of any individual network technology is not dictated by a particular, global network architecture.

The Internet is a packet switching network. At its core, it represents a global carrier of information in form of packets, providing transport services between any two endpoints. These packets bounce through different networks until they reach their destination.

This principle along with its decentralized and layered architecture have fostered a worldwide adoption, harmonizing the interconnection of a plethora of communication technology-specific networks under a global name-space scheme. This scheme targets two key challenges regarding how to identify and address endpoints, as well as how to route the packets. The Internet Protocol IP was devised as the solution targeting both addressing and routing. It defines a single name-space scheme and assigns to each end-point a unique identifier (an IP address). This identifier is addressable and reachable from any other Internet end-point.

A key building block that sustains the Internet is its global addressing and routing scheme. The whole universe of addresses are split into subsets, each representing a network (referred as a network prefix). These interconnected networks exchange their network prefix information with their neighbors as well as the learned information from other neighbors. This information allows each network to build a map of the available and reachable networks, resulting in the construction of a routing table. This table helps in the decision process for relaying packets across network boundaries.

2.2.1 Network Control and Packet Forwarding

A packet switching network consists of different interconnected elements (packet forwarders), which main purpose is to relay packets following a set of rules. There is a key subtle split of concerns regarding this process at each element. On one hand, the forwarding process imposes the requirement to minimize the overhead introduced to the packets passing through each element. This overhead stem from the time required to compute the incoming packets, take a forwarding decision and relay to the next hop. On the other hand, the process to learn and populate the forwarding rules serves as a control mechanism over the element behavior. The former is known as the "data plane", while the latter as "control plane". In this work, we shall focus on the control plane of the Internet and the protocols that govern it.

The data plane has been designed to process packet at line-rate speeds. The forwarding decision process reads the incoming packet address information and utilizes it to take a decision. Therein, the decision result indicates whether the packet is forwarded or not, and toward which output link. Fast matching tables in memory and specialized hardware (e.g., TCAMs) have been devised to support this process at high packets rate. The configuration and population of these decision tables are a key control element of the network, as who has access to them can affect the network behavior at will.

The control plane of a network element consists on the process that coordinates with its neighbors the information regarding who and how to reach other end points

and networks. It is in charge of managing and populating the table that supports the forwarding decision process. Thus, this is a critical component within the network as it can directly affect how the traffic is steered.

It is important to highlight that a given network element may take decisions based on different criteria according to its capabilities. For example, an Ethernet level-2 switch by definition only understands and processes Ethernet frames. Thus, it uses the Ethernet packet header to learn and forward packets. The control plane maintains an "Ethernet address: physical port" biding table to support the forwarding decision process. At this level, the forwarding process is within a specific network technology domain.

Similarly, a layer 3 device is a network element capable of reading and understanding IP headers. This capability allows it to forward packets based on its IP destination address. The control plane is in charge of configuring and maintaining an IP routing table. This table supports the forwarding decision process, enabling the routing of packets between IP networks. In contrast, with the previous Ethernet example, a network element IP-capable is referred as an "IP Router", and it can interconnect different network with specific layer two technology (e.g., Ethernet, LoRa or WiFi). In this work, we focused our research over some of the control protocols that govern the Internet at the IP level, and some of their security issues.

The forwarding decision logic, its coordination and cooperation between the different IP network elements has been designed and devised to be totally distributed. Each network element poses its internal logic to control and govern its forwarding table. To this end, different distributed protocols have been devised, such as BGP, OSPF and LISP. On the other hand, there is another approach for handling and managing the forwarding decision logic, which focuses on the idea of decoupling the control logic from the data forwarding elements. The control and configuration of forwarding rules is concentrated over a central controller. This controller is in a position to handle different elements of a network at a privileged position, which enables the creation of different, isolated virtual networks. This approach is known as Software Defined Network (SDN). Our proposal for offloading the security out from BGP is aligned with the latter paradigm for network virtualization, as it provides a powerful tool for dynamic traffic steering and isolation.

2.2.2 Network addressing scheme and routing

Routing within the Internet was designed under the principle of a global addressing scheme. To this end, the Internet Protocol (IP) was devised (there are two versions: IPv4 [42] and IPv6 [18]). In a nutshell, IPv4 defines a 32-bits long addressing space, where each address is commonly represented as a quartet of decimal numbers separated

by a dot, each bounded between 0 and 255, e.g., "147.83.42.130"—each number is a byte in decimal. An IP address identifies unequivocally an end-point, and also belongs to a network prefix, i.e., semantically it represents the end-point identity (the "who"), and which network prefix is part of (the "where" in the IP network hierarchy and routing terms).

The set of continuous IP addresses constitute an IP network identified by a prefix. Conversely, an IP prefix such as 147.83.0.0/16, defines the set of IP addresses { 147.83.0.0 , 147.83.0.1 , . . . , 147.83.255.255 }, where all share the first 16 bits, i.e., 147.83 (the /16 is commonly referred as "the prefix"). IP prefixes have variable lengths, which allow specifying networks of different sizes. For example the prefix 147.83.0.0/16 contains $2^8 * 2^8$ end-point addresses, while the prefix 147.83.0.0/30 only 2^2. Hereafter, we shall use the term "IP-network" as the set of IP addresses with a common prefix. The characteristics of prefix aggregation and specificity enable the definition of a hierarchical arrangement of IP-prefixes which facilitate the network route learning process.

This addressing scheme and network segmentation pose the challenge of how to steer network traffic exchange between IP-networks. An "IP Router" is the element in charge of steering packets between IP-networks. This element acts as a gateway for all its connected networks, enabling each to reach its counterpart. For example, two IP-networks seeking to exchange traffic shall share a common "IP Router" element, which understands both network prefixes. Aligned with this example, a full interconnection between a set of different IP-networks can be achieved if all share only one "IP Router" element. This creates a centralized topology with the shared routing element at the center. However, this goes against key Internet principles: decentralization and distribution.

To foster Internet decentralization and distribution, an IP-Network can act as a transit network between networks. To illustrate, let us consider the case of three different IP-networks with two routing elements, where each "IP-Router" can be shared only between two networks. This configuration arranges one middle network with two routing elements. The edge networks can reach only the middle one, as the routers do not know the existence of other, not directly connected networks. To achieve a total connectivity between all networks, a coordination between routers is required. This coordination process acts as a control system, which learns and decides how to steer the packets in the network. Furthermore, this control plane seeks to create a logical routing map table with the relevant information regarding how to steer the traffic based on the destination address of packets. Thus, this table consists of a set of "IP-prefix: route to reach it" mappings. Each route entry indicates which is the next hop to forward the traffic.

From this addressing and routing perspective, Internet can be regarded as a set of millions of interconnected IP-networks. Their interconnection is totally distributed, which poses the challenge on the routing control process and its global routing table. Having a manual and static table to all possible IP addresses is far from practical. Different solutions have been devised to dynamically coordinate and learn how to steer the traffic. These solutions are considered part of the control plane, as their objective is to influence the forwarding decision process of each network element by populating and updating their respective routing tables.

In this work, we have focused our attention into two control plane protocols, namely the Border Gateway Protocol (BGP), and the Locator/Identifier Split Protocol (LISP). The former is a key building block that sustains Internet today, while the latter is a new proposal for routing control plane which seeks to tackle some core Internet routing issues related with BGP and with IP itself.

2.2.3 Inter-domain routing

The Internet is comprised by a set of interconnected domains, where domains represent large networks operated by different organizations, such as the government, a cable TV provider, a University or a company. Similarly, a domain is a network or group of networks autonomously operated by a single authority under a common network policy. A network domain is also known as an Autonomous System (AS). From this perspective, the Internet can be described as a graph, where nodes represent network domains (ASes), and edges the links interconnecting them. This domain segmentation promotes a 1:n relationship type among ASes and IP-networks, i.e., one AS can represent one or more IP-networks. To illustrate these concepts, figure 2.1 depicts a set of interconnected ASes represented by circles, each assigned an AS number (ASN). Therein, AS 32 holds the IP-network 147.83.142.0/24, which is advertised to the AS 20. On the Internet today there are over 40K ASes, which represent millions of IP-networks, and their interconnections are changing and shifting over time.

This logical segmentation of the Internet in autonomous systems follows a twofold objective. First, to define a different domain addressing space (i.e., AS numbers) for Internet governability, business relationships and management. The second objective targets to foster a path vector routing protocol, based on the mappings between IP-network and the path of ASes to reach them (referred as AS-path). More specifically, an AS-path consist of the sequence of ASes required to traverse in order to reach a destination. These AS-paths represent the routes toward an IP-network, which are dynamically selected by a policy-based routing decision process. Thus, an AS can

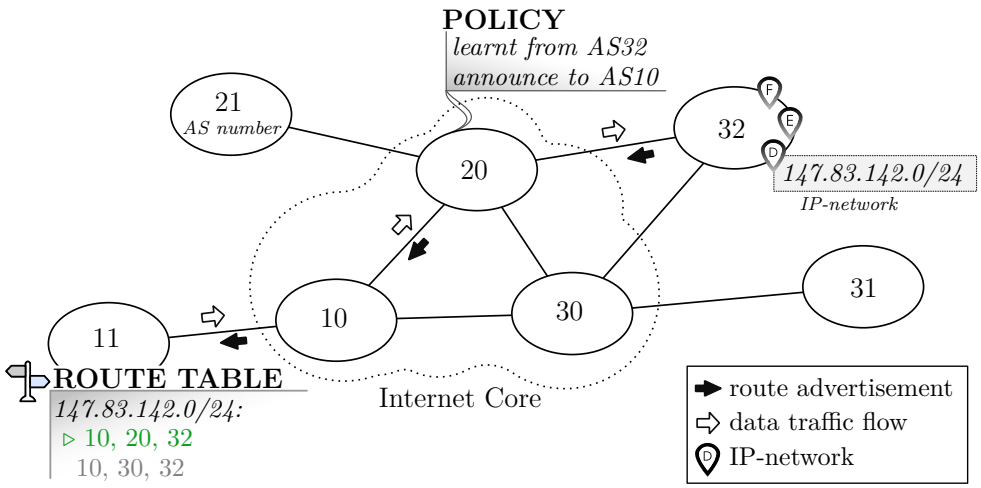


Fig. 2.1 Inter-domain routing example.

be responsible for a set of different IP-networks, and decide to announce them to its neighbors.

These principles define the Border Gateway Protocol (BGP). BGP is a control plane protocol which specifies how the network elements exchange routing information, as well as the criteria to take decisions that control the data plane. A BGP router can learn the different routes from its neighbors, and selectively discriminate its further spread based on local policies.

To illustrate, figure 2.1 depicts a toy example of inter-domain networking. Therein, AS 32 is responsible for the prefix 147.83.142.0/24, and advertises it to its neighbor AS 20. This advertisement contains not only the prefix, but also the AS-path {AS32}. This AS-path contains the route toward the IP-network. Next, AS 20 decides to further advertise the prefix to its neighbor AS 10 by sending the AS-path {32, 20}. Every AS that decides to advertise an IP-network, either its own or a learned one, must append its AS-number to the AS-path being advertised.

The collection of these announcements at each AS allows the construction of a BGP routing table. Note that for a specific IP-network, one or more AS-paths can exist. For example, in figure 2.1 AS 11's route table shows two AS-path routes toward one IP-network. The selection of the preferred AS-path route depends on an internal decision process based on different criteria, which include the longest IP-prefix match, AS-path length, and network policies and preferences defined by an AS network administrator. Furthermore, the decision to propagate an IP-prefix learned from a neighbor to other neighbors shall depend on whether the AS provides traffic transit to its peers, and be aligned with their business relationship agreements.

Overall, only the preferred AS-path route per IP-network are advertised by an AS. This restriction protects an AS by disclosing only its preferred "IP-network:AS-path" route to its neighbors out of all its possible available routes. On top of this, the set of best routes to be propagated are selectively chosen accordingly with the targeted neighbor, discriminated by their business relationships. The following section further elaborates this concept.

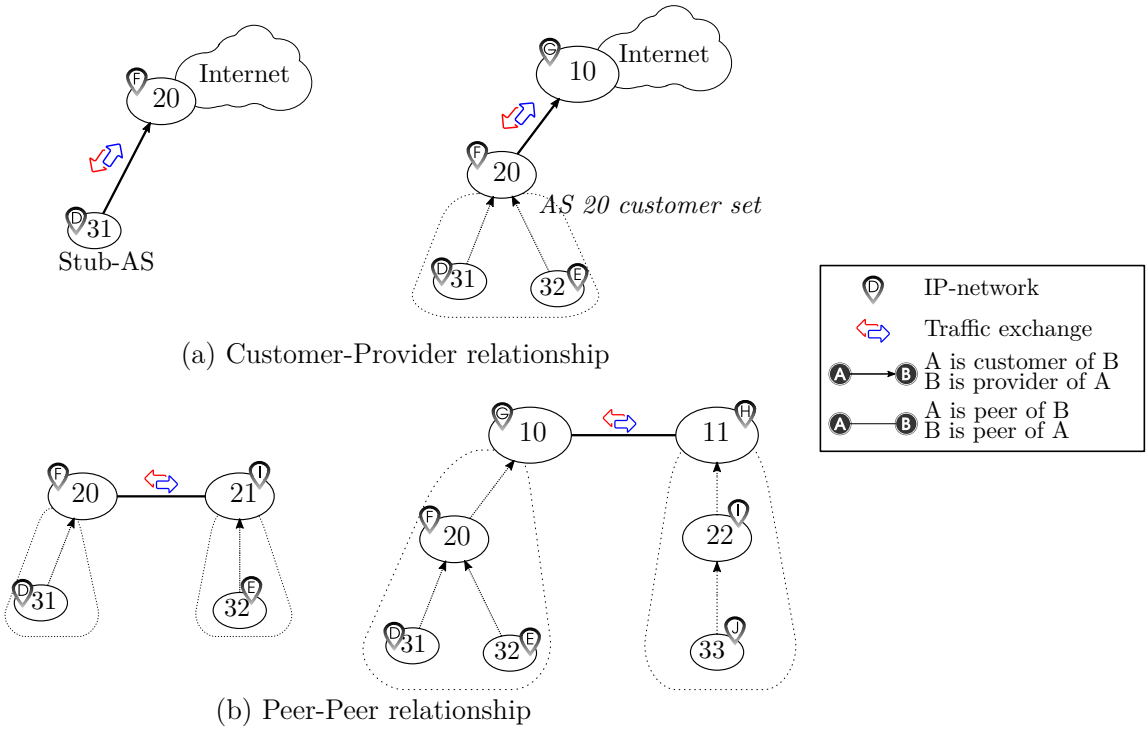
2.2.4 Inter-domain business relationships

The interconnection between any two ASes represents an agreement to exchange traffic under specific conditions. Based on the type of agreement and business interests, these relationships can be classified in two broad categories: *Customer-Provider* (or *Provider-Customer*), and *Peer-Peer*. Each category represents a contractual agreement between the parties, and dictates the policies to enact by the routing control system in terms of routes propagation.

BGP is the dynamic routing protocol controlling over an AS routing table and its inherent decision process for routes selection. An AS collects routing information from all its neighbors and processes it according to its internal control logic, which is influenced by internal preference policies. This process results in the construction of a routing table with all the learned paths and their respective IP-prefix. Out of this set of routes, only the preferred ones form the selected set of routes available to propagate. Then, this set is further discriminated based upon the neighbor relationship. Thus, the type of relationship between two domains dictates the policies regarding both the preference of routes in the learning process and the selection of routes to be exported per neighbor.

The ***Customer-Provider*** relationship consists of a Customer domain requesting for traffic transit service to a Provider domain. Conversely, a Provider AS charges its Customer for the service provided, i.e. for forwarding its traffic to the Internet in both directions. This implies the traffic destined to the Customers' networks, as well as all the traffic coming from them towards other domains and the Internet. The Provider acts as a transit domain, and it is in its business interests to offer the best routes to its clients seeking to maximize the traffic exchange.

On the other hand, the ***Peer-Peer*** category establishes a reciprocal, profit-neutral relationship among domains. Both AS parties agree to mutually transit the traffic between their networks and their clients' networks. This neutral settlement is commonly agreed upon a certain traffic radio, and established between ASes of relatively equal size (in terms of the number of clients or IP-networks).



(a) Customer-Provider relationship

(b) Peer-Peer relationship

Fig. 2.2 Inter-domain relationships.

To illustrate these types of relationships we refer to figure 2.2. The left part of figure 2.2(a) shows a basic Customer-Provider relationship, where AS 20 provides transit connectivity to AS 31, i.e., all traffic destined to AS 31's networks (depicted as network D) is reachable from Internet through AS 20. Similarly, all AS 31's networks can reach Internet through its provider AS 20. Following this idea, the right part of figure 2.2(a) shows another example of this relationship, highlighting the customer set concept. Therein, AS 10 provides connectivity to both AS 20 and its customers networks (ASes 31 and 32).

On the other hand, the part (b) of figure 2.2 depicts a Peer-Peer relationship between ASes 20 and 21 (left side), and ASes 10 and 11 (right side). The former depicts the case of traffic exchanging only considering direct clients, while the latter includes also the customers' customers networks.

These inter-domain types of relationships have enabled the creation of a market around network connectivity and traffic exchange. Figure 2.3 depicts from an inter-domain hierarchical perspective a domain classification based on the type of relationships. At the core, there are tier-1 ASes which do not require to pay any other for transit services. They peer with all other tier-1 ASes (there are around fourteen worldwide), covering all available networks in the Internet. Conversely, on the customer edge there

are domains which do not provide transit services and have only Customer-Provider links. These are referred as Stub-ASes.

Another AS category considers the number of Customer-Provider links a domain has. A single-homed AS means it only has one Customer-Provider, whilst a multi-homed AS has more than one. The latter allows to have multiple routes both from and toward its networks and customers' with respect of the Internet. For example, in figure 2.3 ASes 25 and 31 are multi-homed ASes, whereas AS 30 is single-homed.

Similarly, there is an AS category which has only Peer-Peer connections with other ASes. Consider the case when a big content provider company, or government agency seeks to reach a broad region of clients while having not to pay a provider for transit services—the reasons behind might be geopolitical, economical, strategical or other. To this end, this domain negotiates a Peer-Peer relationship with the domain(s) covering the aimed region. For example, in figure 2.3, AS 20 maintain only Peer-Peer relationships, which allow it to reach and be reachable only from its peers' clients.

Inter-domain learning routes preferences

The BGP decision process considers all the learned routes from its neighbors and computes the preferred ones based on local criteria. Each AS is in a position to unilaterally select the best routes according to its own interests. This selection is influenced by the type of relationship with the neighbor who advertised the route. Thus, these are the general rules regarding the preference over learned routes:

- Routes learned from Customers are preferred over routes learned from Peers and Providers.
- Routes learned from Peers are preferred over routes learned from Providers.

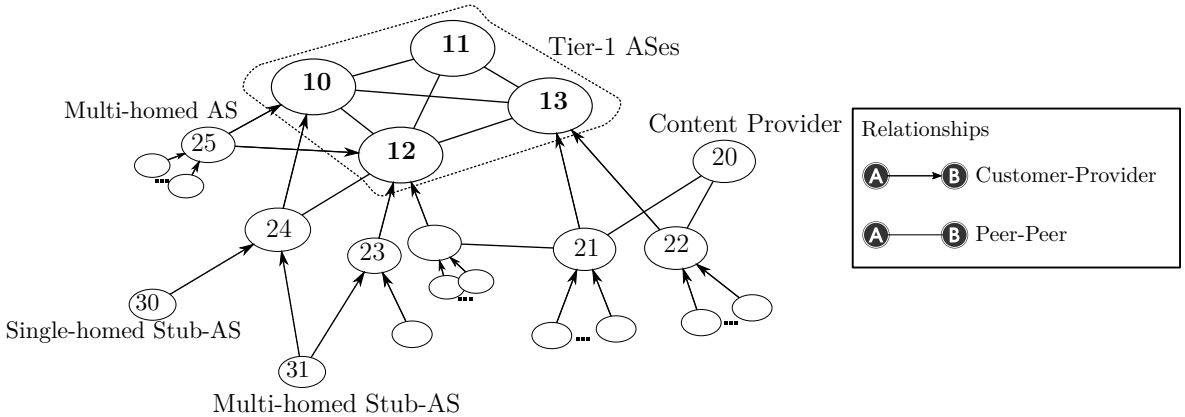


Fig. 2.3 AS classification according to their inter-domain relationships.

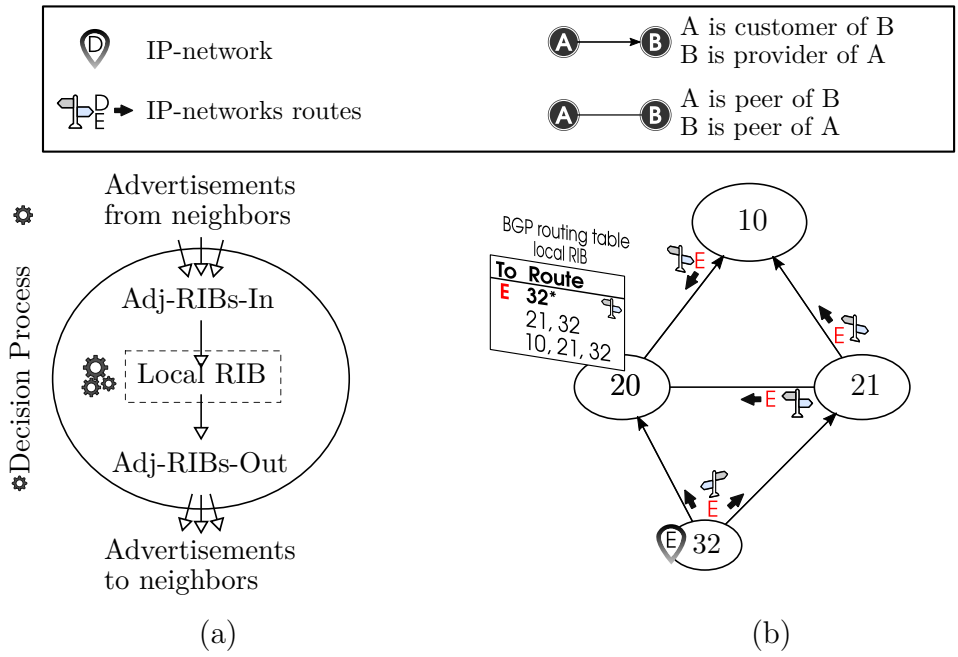


Fig. 2.4 Inter-domain incoming routes selection preference from the perspective of AS 20. The IP-Network example E is originated and advertised by AS 32. This route gets propagated by the domain ASes 21, and 10.

These rules follow a business logic. For example, let us consider the case of a Provider AS with different Customers, Peers and Providers. Out of all learned routes, consider the subset of routes with IP-prefixes reachable through more than one AS-Path. The BGP decision process is in charge of selecting locally the preferred AS-path route per each IP-network. This selection is influenced by the relationship with the next AS hop who advertised the IP-network. More specifically, a Provider shall prefer the routes learned from a Customer over other routes coming from a Peer or a Provider for a specific IP-Network. It is in the interest of the Provider to attract as much as possible traffic destined to its customers from all the other neighbors. Thus, by selecting its customer routes as preferred, these routes are guaranteed to be considered in the export routing process, increasing its possibilities to be further propagated to other domains. The more neighbors accept the route as their preferred, more traffic will be attracted and transited the Provider toward its Client, ultimately increasing its revenue.

To illustrate, figure 2.4(a) shows in a nutshell the control plane components behind BGP [71]. There are three different Routing Information Based (RIB) entities, each serving a specific purpose. The *Adjacent RIB Incoming* (Adj-RIB-in) contains all the incoming announced routes (one adjacent table per domain). Similarly, the Adj-RIB-out entities (one per domain) holds the reconciled export routes. Meanwhile, the Local RIB

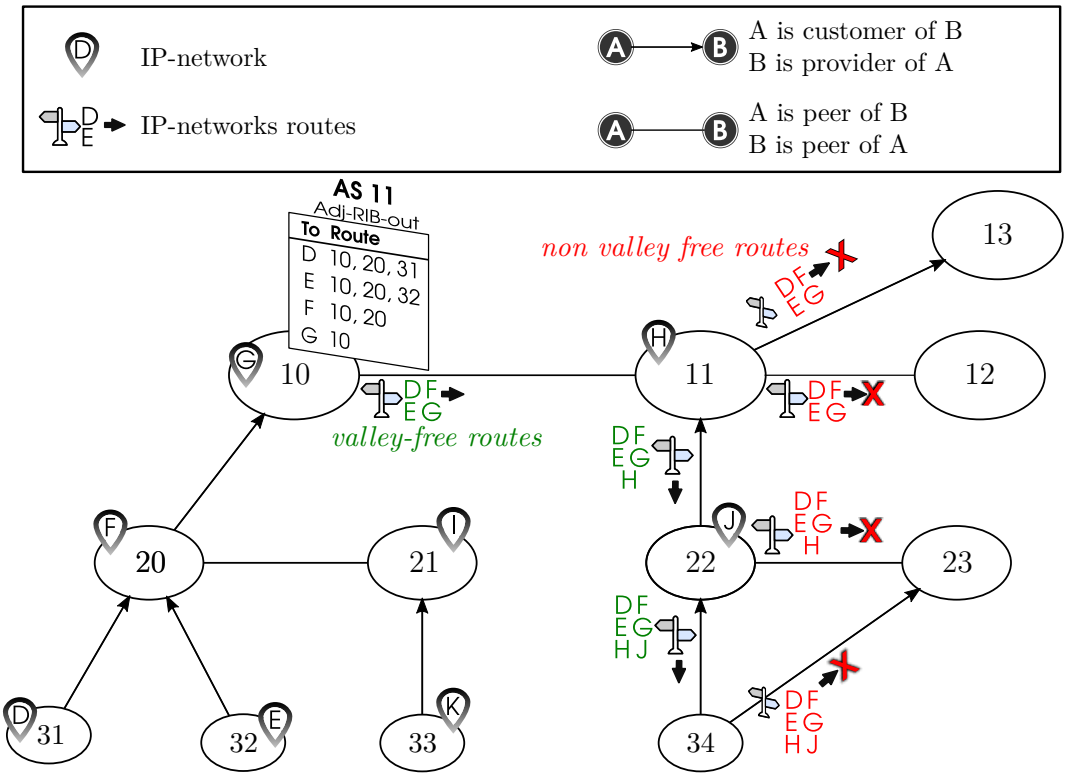


Fig. 2.5 Inter-domain "valley-free" export routes.

(Loc-RIB) contains the superset of local learned routes along with the preferred ones. The BGP decision process is in charge of compute the both the local and the different output RIBs based on the local policies and relationships.

An example of how the Adj-RIB-in/out tables are populated is depicted in figure 2.4(b). Therein, the routing advertisement illustration starts from AS 32. Consider the case of AS 20, who receives different route advertisements from all its neighbors regarding the IP-network E. The criteria followed by these domains to further propagate the route is discussed in the next section. Then, based on the type of relationships between AS 20 and its neighbors ASes 32, 21, and 10, the decision process selects the route with AS-Path $\{10\}$ as the preferred for the IP-Network E.

Once the Decision Process has computed the set of preferred routes for all learned IP-Networks, the next step is to select the export routes to be announce to each neighbor in order to announce its neighbors about its IP visibility. The next section introduces the criteria to compute these sets of export routes.

Inter-domain exporting routes preferences

BGP is a policy based routing protocol which enables the control of advertised (exported) routes based on the neighbor domain and their type of relationship. Prior this process, a network element first has to compute its local routing table and select its preferred routes (as previously described). With this routing information, an AS can decide which routes to advertise to which neighbor. The simplest rule would be to advertised all the local preferred routes to all neighbors. However, from a business perspective this approach may not only hinder the domain interests but also open the door for possible security issues.

In an ideal Internet where all BGP domains behave properly, every one of they should follow the "valley-free" guidelines for exporting routes proposed by [69]. From the perspective of any AS deciding the exporting routes set per neighbor, the rules are:

- Routes **learned from a Customer** can be further advertised to other **Customers, Peers and Providers**.
- Routes **learned from a Peer** can be further advertised to **Customers only**.
- Routes **learned from a Provider** can be further advertised to **Customers only**.

These guidelines are promoted from a business stand, as the decision of exporting a specific route by a domain immediately positions it as a tentative transit toward the route IP-Network. In the case of a Customer route, it is in the interest of the AS to propagate it to all of its neighbors as it charges the customer for the traffic it transits. To illustrate, consider the case of domain $AS10$ in figure 2.5. To maximize it revenue, this domain advertises aSll the routes learned from its customer $AS20$ in an attempt to attract the most traffic from its peer $AS11$ for the traffic destined to IP-Networks [D, E, and F].

In the same figure, consider the case for $AS11$ and the export routes toward $AS13$. These routes were learned from its Peer $AS10$, and it attempts to propagate them toward its Provider $AS13$. This goes against the "valley-free" guidelines, and most importantly, it goes against its own interests. In the case this route is propagated, $AS11$ could become a transit domain for traffic it is not getting any revenue. By the contrary, this traffic impacts its costs, as it involves using the network resources from its provider $AS13$. The figure also depicts other cases where the export of route policies is against the "valley-free" guidelines.

The "valley-free" rules induce to some general restrictions to be considered regarding the decision process applied over the exporting routes. These guidelines include:

- Routes **learned from a Peer** should not be further advertised **neither to other Peers, nor Providers**.
- Routes **learned from a Provider** should not be further advertised **neither to other Providers, nor Peers**.

Our research over the BGP security focuses on the problem of "valley-free" export rules violation, and their security implications over the BGP routing control. In section 3.1, we further elaborate the targeted security issues and proposed approach to address them.

2.2.5 Locator-Identifier Addressing scheme and Routing

The Locator/Identifier Separation Protocol (LISP) [23] is a routing protocol initially designed and devised to tackle routing scalability issues in the core Internet. However, due to its intrinsic address semantic splitting and simple architecture, LISP was promptly spotted as a technology with a remarkable potential in other areas in networking. As a consequence, the focus of LISP has shifted over time and is now becoming a key technology in areas related to network virtualization, mobility, and cloud applications.

The design principle around Internet Addressing considers a global addressing scheme and namespace. This namespace was devised in a way that it allows both the identification

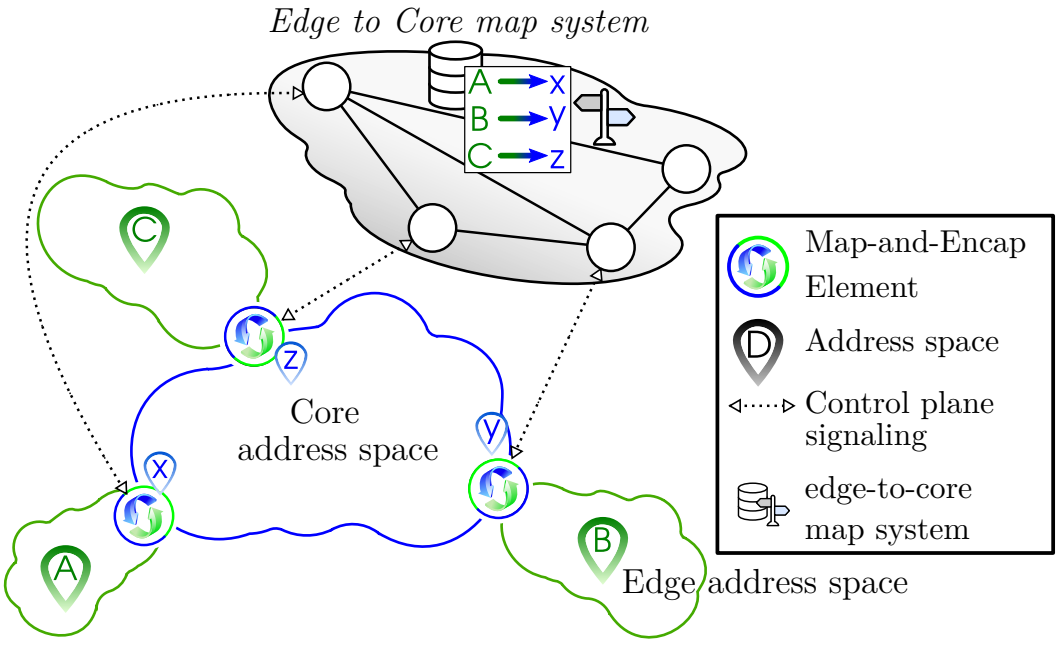


Fig. 2.6 Locator-Identifier Addressing and Routing scheme.

of end points, as well as the segmentation and aggregation of these addresses by groups. Thus, this solution was engineered to form a hierarchy of addresses by aggregating more specific addresses into more general networks. Therein, each end point address serves a twofold purpose. On one hand, it unequivocally identifies an end point within the global network, while in parallel it also signals the routing system the location of the end point with respect to the addresses hierarchy. The aggregation feature enables to optimize the routing tables, as less number of routes toward end points are required. In contrast, this generates a rather static addressing hierarchy hindering new requirements in terms of multi-homing and mobility.

Different factors which were not initially foreseen when the Internet addressing scheme was designed have accelerated the growing of the core Internet routing table. These factors include muti-homing, mobility, traffic engineering and others related with BGP security. They require the creation and spread of more specific routes, which goes against the address aggregation hierarchy by definition. For example, a mobile end user may require to be able to use her address independently of the network service provider. This implies that the user owns a part of the Internet addressing resources, and now the routing control system should react to better steer the traffic toward her. Similar use cases happen when we consider multi-homing or traffic engineering.

There is an inherent problem regarding the semantic meaning of an Internet IP address. It represents not only the identification of an end-point within the network, but also its location according to the network address hierarchy. In other words, an IP address no only signify the "who" in an end-to-end communication (the end-point identity) but also "where" the endpoint is located. This problem has motivated the design of new addressing and routing solutions–one of them is the LISP protocol.

LISP tackles the semantic overload problem by decoupling the unique IP address space into two different address spaces. One of them defines the address space of the core network which interconnects different edge networks. This address spaces is referred as Route LOCator (RLOC) namespace. The second namespace defines the addressing scheme used by the edge networks and end points. This is referred as End point IDentifier (EID) namespace.

The two address spaces enable a level of indirection where different edge addressing schemes and routing systems can coexist over a common core addressing scheme. Encapsulation is a key concept that underpins this technology. On top of it, LISP defines a Map-and-Encap routing mechanism, which supports the dynamic map resolution of end point identifiers (EID addresses), prior the encapsulation and traffic forwarding.

To illustrate, figure 2.6 depicts the concept separation of concerns in two independent addressing namespaces. This splitting allows the definition of any addressing and routing technique at each one (e.g., the core address space could be IPv4, while the edge namespace IPv6). The RLOC namespace is defined as the core address space, which might resemble the Internet core, as this address space could be the current IP network. Meanwhile, the Edge address space represents the EID identifiers. A communication from A to B requires the intervention of two Map-and-Encap elements, as they are co-located at their corresponding edge networks. These elements are in the middle of the communication, as they dynamically and transparently retrieve the EID-to-Locator resolution, prior triggering the encapsulation. Once the packets are encapsulated, they can traverse the core network until they reach the counterpart which is in charge of decapsulate and steer them toward the final destination. Thus, LISP novelty stems from its capacity to create and establish on the fly encapsulation tunnels between the intermediate elements in order to handle the traffic generated between the edge networks and their endpoints.

This dynamic map-and-encap implies that there is a control plane in charge of two key tasks. On one hand, the process of edge-to-core address lookup, while on the other hand a global mapping register in charge of keeping the up-to-date EID-to-Location information. Next, we shall briefly elaborate them, and describe the issues we attempt to address in this work.

Edge-to-Core lookup process

LISP defines the elements in charge of the encapsulation as Ingress and Egress Tunnel Routers (ITR and ETR respectively). These elements are in charge of both handling the end-points traffic by properly tunneling it between them, while simultaneously interacting with the control plane for the edge-to-core translation.

The translation process involves three parties, the ITR, the Map Resolver along with a Mapping System, and the ETR as depicted in figure 2.7. To illustrate therein, consider the case of a communication between two endpoints at namespaces EID_A and EID_B, respectively. The xTR at EID_A requires to learn the location of the destination namespace EID_B, thus requests to an external element the resolution EID-to-RLOC, i.e., the Map Resolver. This element interacts with the mapping system in an attempt to find the current RLOC, and forwards the request to the proper Map Server. The latter is in charge of maintaining an updated register of the current bindings between namespaces. Finally, it forwards the request to the ETR. Once the ITR acquires this information, it starts encapsulating the traffic and passing it toward the ETR. At the destination, the

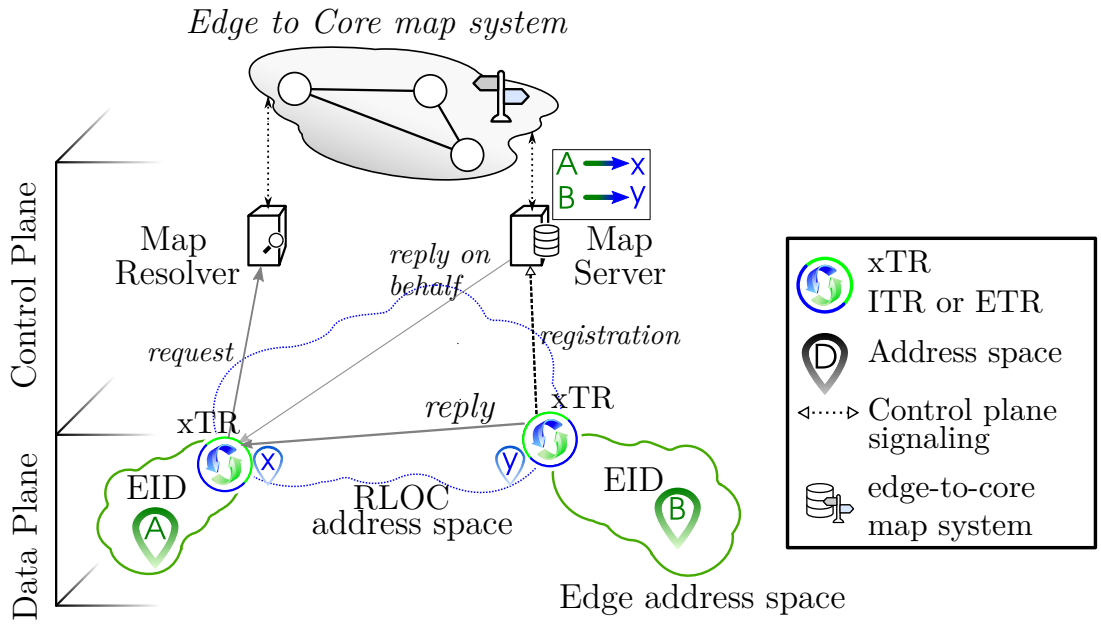


Fig. 2.7 LISP control and data plane split of concerns.

ETR on behalf of EID_B receives its traffic and removes the encapsulation, then forwards it to the endpoint.

The protocol design has enabled a clear separation between data plane and control plane, fostering their individual evolution. For example, there have been different proposals for a global Mapping Systems. Initially, the protocol was devised with a BGP-based mapping system called LISP+ALT [25], but it has been replaced by a DNS-like indexing system called DDT [26].

The whole EID-to-RLOC lookup occurs when a local xTR lacks the information about the *RLOC* destination of a given *EID*. The protocol has designed and defined this whole procedure and its intrinsic security concerns. This is important as a rogue translation reply may derive in the forwarding of encapsulated traffic toward an attacker. The IETF's working group on LISP has defined LISPSEC [55] in this regard. It provides a set of security mechanisms that targets security concerns regarding the origin authentication, integrity and replay protection to the control messages exchanged in the lookup process.

The messages involved in the lookup process include a Map Request and a Map Replay messages. The former is originated to request the Map Resolver an EID-to-RLOC lookup, while the latter contains the mapping data sent by the ETR (or the Map server in the case of proxy mode). These security mechanism target a threat model excluding the Mapping system, and they target to validate the lookup triggered process by means

of a dynamic ephemeral one time key. This key enables all parties to accept only valid lookup requests and replies.

EID Map registration

Any EID prefix (a whole network or an endpoint address) is required to register and update its RLOC into the Mapping system. To this end, LISP defines the Map Server as the entity in charge of handling the registration requests and location updates. It interacts with both the ETR and ITR for the registration and lookup process. The latter interaction only takes place when the registration includes the proxy mode, i.e. the ETR requests the Map Server to answer the Map Requests messages on its behalf. Figure 2.7 illustrates this interaction, as the ETR at site EID_B registers this EID with its RLOC y. The Map Server on handling EID_B is also in charge of forwarding the Map Requests toward the specific ETR, or directly answering them in the case of proxy mode.

On one hand, the interaction between the Map Server and the ETR corresponds to the exchange of credentials and the updated location. Thus, LISP defines the mechanisms to authenticate and validate this registration process via a pair-wise shared key. Furthermore, 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, the ETR, and on the Map Server [24]. 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.

Similarly, the interaction between the Map Server and the Map Resolver consists in the exchange of location information as the result of a Map Request lookup. This information signals to the ITR not only the availability of an EID but also the set of RLOCs through which reach it, and their preferences. This exchange of information is secured by a means of an ephemeral One Time Key. These keys are dynamically generated when the lookup process is triggered by an ITR, and serves to the purpose of validating the request [55].

These two elements abstract the Mapping system, and provides a front end interface for both the Ingress and Egress Tunnel Routers. This removes any dependency between them and the mapping system, fostering the proposal of new mapping systems.

It is important to notice that this existing pre-shared key security mechanism between the ETR and the Map Server 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. We shall focus our attention to these shortcomings and further elaborate them in Chapter 6.

2.3 End Users and End-points security protection

The security protection model over users and their devices follow a twofold approach. On one hand, there is a device-centric protection paradigm which consists on the protection of the device through the composed security from applications and the operating system security. This approach consists on provisioning each endpoint with the application tools and system updates to protect the end user from external threats. On the other hand, there is a network-oriented security protection which focuses more on protecting the user's connections and shared information through the network interaction, i.e., a control over the users connections and transferred data. This section describes an overview regarding these approaches in the context of end user devices, and resource-constrained devices.

Properly securing a user's devices requires of many steps and efforts to accomplish. A multi-layered computer security approach or defense in depth is recommended [16], where additional layers of defense provide better protection in the event of a security layer is breached. Defense in depth security is extremely important considering that the end user is often the first line of defense, but also often referred to as the weakest link in information security.

Table 2.1 presents a broad classification of different security protections from a layered perspective. It considers the cases when the security is enforced either at the device or

		Non-constrained devices	Resource constrained devices
Device security protection	**Application + Layer 4**	Anti-virus, Anti-phishing, Anti-spam, Parental Control, End-to-End Encryption, Web Filtering, IDS[a] IPS[b] L4-7 Firewall, HTTPS[c] TLS[d]	HTTPS, TLS, DTLS[e]
	Operating System + Network stack	Firewall, VPN, IDS, IPS, Wired and Wireless Security	Wireless Security (Wifi, LoRa, Zigbee, BluetoothLE)
Network security protection	**Layer 4 and above**	Firewall, IDS, IPS	
	IP layer	VPN, Ipsec, tunnels	
	Access layer	Wired security, Wireless Secrity (e.g., WPA), network segmentation (VLAN)	

[a] IDS: Intrusion Detection System.
[b] IPS: Intrusion Protection System.
[c] HTTPS: HTTP Secure.
[d] TLS: Transport Layer Security.
[e] DTLS: Datagram TLS.

Table 2.1 End users and endpoints security protection from the twofold perspective: i) end device protection, and ii) network protection.

in its network connection. Therein, we further classify between non-constrained devices and resource constrained devices.

2.3.1 Device security protection

The security protection at the users' devices and the things can be broadly classified in two layers, i.e., security applications, and Operating System plus Network stack security. This classification is shown in the top part of table 2.1. Giving the nature of the device, and its available resources we further classify into two subgroups: a) non-constrained devices and b) resource-constrained devices. Therein, end user devices with non-constrained resources can select different security options from a broad catalog of security applications, ranging from anti-virus, anti-phishing, parental control, end-to-end encryption like PGP, and more. Furthermore, these devices support standard libraries and implementations to foster the use of secure Layer 4 transport protocols such as HTTPS and TLS. Similarly, each device Operating System (OS) in conjunction with its network stack support security

protections such as Layer 3 Firewall, Virtual Private Networks (VPN) like IPSec, and access link security such as WiFi WPA.

On the other hand, resource-constrained devices usually come enabled with optimized versions of the security options as compared with the non-constrained devices. At the application layer, these devices in general support HTTPS, TLS and DTLS. Lower in the stack, the security at the OS and network access link is dependent with the security implemented by the access technology (e.g., WPA in the case of WiFi).

2.3.2 Network security protection

The second component to enforce security to end users and devices is derived from the protection provided by the network connection. This network-centric protection generally focuses on the aggregated protection provided to each connected user (as shown in the bottom part of table 2.1). In this case, the norm is to enforce security policies based on coarse, generic user-profiles. Therein, the network can enforce security at different layers. For example, a Layer 3 to 7 Firewall, an Intrusion Detection System (IDS), or an Intrusion Protection System (IPS). Similarly, VPN and IPSec tunnels at layer 3, while MAC filtering, WiFi WPA and VLAN at the lower layers.

The network-centric security is decoupled from the end device, thus it can be applied to all the devices that are connected and communicating through the network box in charge. For example, in the case of IoT, a gateway is an external element that provides certain services to the devices, as well as enforce the required security protection to access and share the information gathered. This kind of security follows the offloading paradigm. However, legacy and cumbersome equipment with support of specific network and access technologies were required.

2.4 Summary

This chapter has reviewed the key concepts and ideas that support this thesis. First, we introduce the problem of the Internet and the security continuum. It is key to understand the differences between threats, vulnerabilities and attacks. These concepts shall be used throughout the rest of this thesis.

Similarly, we describe the current state of Internet routing and address the topics around the inter-domain routing, the different business relationships among domains, and other relevant routing protocols such as LISP. This section provides an idea about these key Internet components, their design principles and basic functionality.

The final part of the chapter describes the end users and End-points security protection. Therein, we describe the compounded protection over endpoints, which results of the aggregated security applications provisioned at the devices, their operating system security, and the protection at the network connection. At the end devices, we further split and describe the distinction between non-constrained and constrained devices, each group with their own protection applications adapted accordingly to their settings.

Part II: Internet routing vulnerabilities

Chapter 3

Why is the Inter-domain routing system insecure?

The Inter-domain routing system was originally designed under certain principles and considerations which contemplated a fair-play environment among its participants. However, the widespread adoption of Internet has influenced major changes over its players' behavior, which exposed the vulnerabilities derived from these assumptions. This has happened mainly due to the changes in the interaction dynamics between domains and their priorities driven by, for example, their business models. These vulnerabilities have proven to be effective, as discussed in the introduction with the examples of Pakistan and China incidents, and can cause serious Internet outages.

BGP "insecurity" mainly stems from the design principle of mutual trusted parties. This implies that each party trusts its counterparts and accepts as valid all the routing information advertised to them. Although this model proved sufficient in the early Internet stages, it has become highly vulnerable to abuses and attacks. For example, any AS can announce any route path containing any IP-network to any subset of its neighbors. This vulnerability has enabled a wide range of threats, which a misbehaving domain can exploit to affect the routing control system with or without a specific purpose. For example, these vulnerabilities can result in an Internet outage, service disruption, traffic hijacking or eavesdropping.

This section describes some BGP vulnerabilities within the context of a threat model. Therein, we shall highlight the current work that address some of them while remarking the ones that we target in this work.

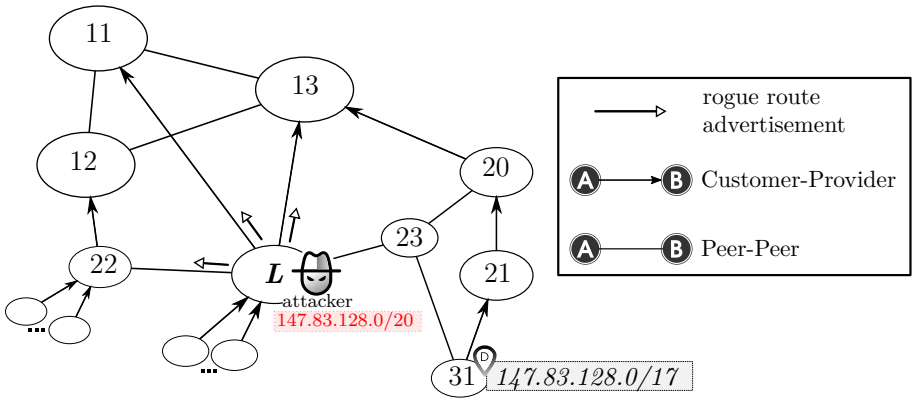


Fig. 3.1 AS sub-graph example. The attacker AS is represented by "L".

3.1 BGP threat model

Our threat model considers a single AS attacker (depicted as L in figure 3.1) seeking to exploit BGP vulnerabilities with the objective of influencing over the routing decision process of its direct neighbors. The considered vulnerabilities allow the spread of rogue routes containing invalid AS-paths, non-authorized IP-networks ownership or in violation of the *valley-free* export policies. These vulnerabilities affect not only to the attacker's neighbors, as they may change their preference routes in favor of the attacker, but also may impact the owner of the IP-network. The implications regarding the spread of this information beyond the attacker's neighbors is not considered in this work.

From the attacker's neighbor perspective (the victim AS), these vulnerabilities realize only when the rogue route advertisements are accepted as valid and influence the domain routing preferences, i.e. the victim selects the new route as the preferred. This selection is crucial in the inter-domain routing system, as it determines how the traffic is steered as well as how the routing knowledge is further advertised by a domain.

The selection of the set of best routes is subjected to private, per-domain policies. These policies encompass the AS preferences over learned routes, as well as policies for exporting routes. They delineate the criteria that rules the complex process of validating, and accepting new routes into the domain routing knowledge. Based on the set of learned routes per neighbor, each domain computes the set of best routes, one route per IP-network. Finally, the domain has to decide on a per-neighbor basis which routes can be advertised to each of its neighbors without violating any export policy. The whole process takes into consideration the type of relationship with the neighbor, and should follow the "valley free" rules. The following section describes the model for routing policies and export policies considered.

On the other hand, the spread of rogue routes claiming unauthorized IP-network ownership affect also to the legitimate owners. In this case, the attacker attempts to deceive its neighbor by announcing himself as the owner of the targeted IP-network (e.g. the AS-path contains a set only with his ASN). If successful (quite likely as the AS-path may be shorter than the valid route), the victims traffic transiting through the tainted attacker's neighbor is diverted to him. The final objective of the attacker might be to disrupt the victim's service by dropping its traffic, or eavesdrop it and then properly forward it (if possible) without being noticed. This vulnerability is also subjected to acceptance of the rogue route. Next, we describe the model for routing policies and export policies followed in this work.

3.1.1 Routing and Export policies model

Overall, there are public guidelines that defines this BGP decision process. However, these guidelines are overridden by the local, private policies. This fact makes it very difficult to predict the possible outcome of a rogue advertisement. Thus, we use the routing model described by Gao and Rexford [30] which captures the policies of a considerable number of ASes [33]. The model defines the general set of rules and their priorities when selecting a single route from the set of routes to a given IP-Network (this set of routes include all the routes learned from all the neighbors):

- ***Local Preference***: Aligned with the type of AS relationship, prefer routes from customers over peers, and over providers (in that strict order). This policy follows the valley-free rationale and is aligned with the business incentives, as described in section 2.2.4.
- ***AS Paths***: Prefer routes with shorter AS-paths over longer routes.
- ***Tie-Break Rule***: use other criteria, such as the next-hop router-id or its geographic location to break ties among routes.

The result of this policy-based routing decision process is a list of the selected routes (one per IP-network) preferred by the domain. Each one of them are then available to be announced to a subset of its neighbors according to the following rule:

- ***Export Policy***: A customer route (i.e. a route learned from a customer) can be exported to all neighbors. Peer and Providers routes can be exported only to customers. Similarly, as with the Local Preference directive, this rule is aligned with the domain business incentives in terms of attracting and transiting the traffic between two of its neighbors when at least one of them is a paying customer.

3.2 BGP routing threats

The considered subset of threats around BGP have been classified into three subgroups. The first one addresses the vulnerabilities regarding IP-prefixes ownership, while the second targets the threats over AS-paths. Finally, the third group focuses on the threat posed by the violation of "valley-free" export policies.

3.2.1 Prefix and sub-prefix ownership

Any AS in its attempt to attract traffic destined to a victim's IP-network can originate a route declaring itself as the owner of the prefix (or sub-prefix). For example, the attacker AS L in figure 3.1 can advertise to its providers and peers the rogue route:

$$147.83.128.0/20\text{: } [L]$$

This vulnerability will affect not only to the victim AS 31, but also to the attacker's neighbors, as their routing state will be deceived. To elaborate, let us assume that the IP-network prefix 147.73.128.0/17 has been legitimately allocated to the AS 31. In a normal and fair BGP behavior (following the routing policies and export policies rules described in the previous section), AS 11 shall receive the AS-path route $[12, 20, 21, 31]$ to reach this network. The attacker L arbitrarily can declare as the owner of the network and announce it to his neighbors. They quite likely shall accept this new route based on the shortest path rule. Similarly, a more specific and effective hijack attack can consist on the announcement of a sub-prefix such as 147.83.128.0/20 with any AS-path. The reach of these kinds of attacks depends on how further the rogue advertisement is spread over ASes.

The difference between prefix and sub-prefix hijacking derive from the *longest-prefix match* routing preference in BGP. This rule indicates that every router identifies the longest IP prefix that covers the destination IP address of a packet, and steers it along the route to that IP prefix. This preference for the longest-prefix match may influence a broader impact of the attack, regardless of the announced AS-path. For example, when the attacker announces the sub-prefix 147.83.128.0/20 with AS-path $[L, 21, 31]$, its peers and providers shall prefer this route over the route 147.83.128.0/17: $[L]$ even though it has a shorter path.

3.2.2 False AS-paths

The attacker L in figure 3.1 can attempt to deceive its neighbors not by overtaking the IP-prefix of a victim AS, but by modifying the AS-path with rogue or inexistent inter-

domain links. To illustrate, L can announce the bogus AS-path $[L, 31]$ to its neighbors. There is no direct link between ASes L and 31, but BGP lacks a mechanism to validate neither the AS topology nor the agreement of each domain for this route. This threat allows the attacker to play with the "AS-path" length advertised seeking to influence the route preference over its neighbors routing state.

3.2.3 Export policy violation

This vulnerability appears when any AS violates the export policy over the set of learned routes. More specifically, this threat raises when an AS does not follow the "valley-free" guidelines regarding the selection and further advertisement of routes, which is characterized by the type of neighbor agreements and business incentives.

For example, in figure 3.1 the attacker L can deceive the AS 11 to prefer the AS-path $[31, 21, 20, L]$ over the valid learned path $[31, 21, 20, 13]$ toward the IP-network 147.83.128.0/17. The former path violates the export policy guideline regarding learned routes from a peer. From the perspective of AS 11, this new route is preferable as it comes from a customer.

Similarly, AS 22 may be the victim of an attack if it receives the route $[31, 21, 20, L]$ from L. In this case, L is further advertising a route to a peer learned from another peer, which violates the "valley-free" export policies. From the perspective of the victim AS 22, this new route is more attractive as it comes from its peer, as compared with the legitimate one coming from its provider.

This particular threat has attracted our attention as it is partially addressed by the solutions proposed in the Internet community. We shall further investigate this threat in Chapter 4.

3.3 Defense efforts for securing BGP vulnerabilities

This section briefly describes some of the current efforts for improving the BGP security. These efforts focus primarily on providing mechanisms to address the vulnerabilities and threats described in the previous section.

3.3.1 Route origin authorization and validation

The prefix and sub-prefix hijack threats arise in BGP due to its lack of proper mechanisms to validate both the allocation of Internet addressing resources—AS numbers and IP-prefixes—as well as route origin authorization. The former consists on providing each

domain with the capabilities to verify a resource allocation claim made by any other domain. Meanwhile, the latter focuses on the capability to authoritatively prove what AS(es) may announce and represent an IP-prefix. The origin validation defense provides a trusted distributed database binding ASes to the IP-prefixes allocated to them. Any route advertisement which violates this binding may be discarded according to the local policy.

Notable contributions proposed to secure BGP in more than a decade of study include Secure-BGP (S-BGP) [44], Secure Origin BGP (soBGP) [90], Pretty Secure BGP (psBGP) [65], and IRV [34]. These proposals outline extensive support architectures for enhancing BGP security in an effort to strengthen both the operation of the protocol, and the validation of exchanged BGP messages. Whilst the study of each of these proposals is out of the scope of this work, it is important to highlight their common factor: all of them required a capability to sustain basic assertions relating to both the validation of resources ownership and the origination of a route into the inter-domain routing system.

The current proposal by SIDR WG understood this required capability and approached it through the use of "Resource Certificates". These certificates contain an extension field that lists a collection of resources, and their purpose is to attest that the certificate issuer has granted to the certificate subject a unique "right-of-use" for the associated set of resources.

This proposal is referred as Resource Public Key Infrastructure (RPKI)[49], and consists of three main parts: i) a resource allocation hierarchy, ii) a set of cryptographically protected objects, and iii) a distributed repository framework to publish these objects. By using cryptographically verifiable statements, RPKI helps to ensure that Internet address resource holders are certifiably linked to those resources. Also, this capability enhances BGP entities to gather reliable route origin information and further decide to discard advertisements which violate both the resource ownership and the proper route origination authorizations.

Overall, RPKI mirrors the currently practiced administrative allocation hierarchy of Internet Number Resources (INRs) for both IP addresses and AS numbers. The RPKI hierarchy is based on the administrative resource allocation hierarchy as depicted in figure 3.2, where resources are distributed starting from the Internet Assigned Numbers Authority (IANA), Regional Internet Registry (RIR), National Internet Registry, Internet Service Provider and End Users. At each level in the hierarchy, the resources allocated are matched with their respective certificates.

In the presence of X.509 certificates, each resource allocation action becomes cryptographically verifiable, as the certificate attests to the allocation of a particular resource,

either an IP-prefix or AS number. A Certification Authority (CA) corresponds to an entity that can further sub-allocate resources and delegate authorities using resource certificates. Figure 3.2 depicts an overview of the RPKI hierarchy, e.g., the chain of trust starts from the IANA root, further sub-delegating the resources to RIRs. These CA certificates enable to form a chain of cryptographically verifiable trust from IANA to a particular AS or ISP. End Entity (EE) certificates are another type of resource certificates which are used for delegating authorities, e.g., every Route Origin Authorization (ROA) includes an EE certificate which enables its cryptographic verification, as shown in figure 3.2. These certificates and authorization objects are published in the respective RPKI repository publication point of each CA. Every CA in the RPKI regularly issues Certificate Revocation Lists (CRLs) to revoke invalid certificates. The collection of all such distributed repositories from all the CAs constitute the global RPKI, which is available to Relying Parties (RP) that would want to validate an attestation or authority.

Given such security skeleton, ASes can obtain certificates for the resources they own from the concerned resource allocation authorities. The second and third SIDR proposals, i.e., ROA and BGPSEC, utilize these certificates to offer security to the exchanged information, such that the receiving party could cryptographically verify the presented credentials. Thus, both ROA and the BGPSEC extensively rely on RPKI to achieve their goals, i.e., verifying route origin advertisements, and securing route propagation updates, respectively. Each AS can have its own RPKI cache synchronized and regularly updated against the global RPKI. Observe that the RPKI is a new addition in the inter-domain routing infrastructure, and therefore, it requires extra investment for new hardware and software components. The SIDR WG has published a number of proposed standards

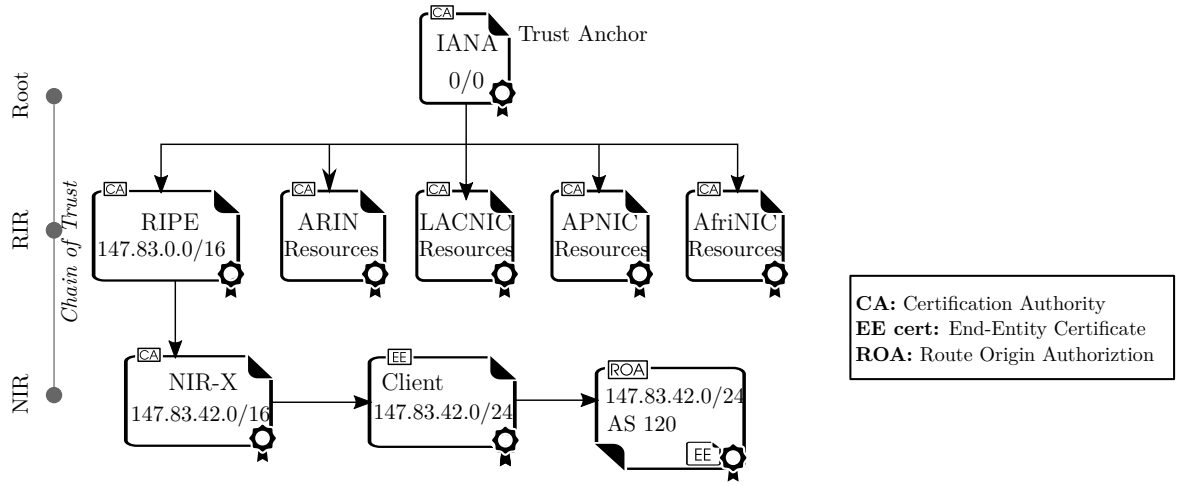


Fig. 3.2 Internet Administrative resource allocation hierarchy.

as well as best practices RFCs related to RPKI. The RFC 6480 [49] provides detailed description of an infrastructure to support secure Internet routing, while RFC 6481 [39] describes a standard profile for a resource certificate repository structure. A complete list of RFCs can be found in [41].

Resource Origin Authorization (ROA)

The Route Origin Authorization (ROA) proposal of the SIDR WG targets the traffic hijacking problem due to false route origin advertisements. The ROA proposal makes use of RPKI to assure integrity in the route origin announcements. This is achieved by the use of a particular signed authority, called Route Origination Authorization. The RPKI enables the legitimate owner of an IP-prefix to produce a ROA and publish it in the RPKI repository. This signed authority is formatted according to the Cryptographic Message Syntax (CMS) [37], and it binds the IP-prefix resource with its owner's AS number by including the corresponding EE certificate inside it (see the bottom right of figure 3.2).

Now, when an AS announces a particular IP prefix as its owner, the Relying Party (RP) can verify if this route origination announcement is legitimate or not with the help of RPKI. The RP queries the RPKI to confirm whether there exists an ROA for the announced IP resource with the advertising AS as its legitimate owner. The response of the query from the RPKI can be used to influence the BGP decision process according to the internal policy of the AS. In practice, instead of querying the global RPKI repository for every route origin announcement, RPs create validation filters. The validation filters are created using the IP prefix (including its length), and the originating AS contained in the published ROAs, which are all available through a locally cached collection of valid ROAs.

For example, consider the case of figure 3.3, where an attacker "L" attempts to hijack the IP-prefix 147.83.128.0/17. To this end, "L" announces to its neighbors ASes 20 and 13 the IP-prefix reclaiming its ownership. However, as the ROA object has been properly published, the AS victims can verify the route announcement sent by the attacker and detect its non validity—there is no ROA stating that the IP-prefix can be announced by AS "L". A new protocol called Router-RPKI (Rtr-RPKI) [12], allows routers to reliably interact with RPKI to retrieve IP prefix origin data from a trusted RPKI cache. Clearly, the RPKI caches need to be synchronized with the global RPKI repository, and for the moment, this is done through *rsync* [53]. Finally, it is important to mention that without additional means, ROA requires minor changes to the BGP protocol itself for performing IP-prefix origin validation. More specifically, as we shall discuss later in Chapter 4, the

advent of Software Defined Networking (SDN) [45] could avoid the introduction of such changes in BGP, since the origin validation can be outsourced and run as a separate process not embedded in BGP. For further details on the procedure for validating an ROA using RPKI, the reader is referred to [40].

3.3.2 Internet AS topology validation

The route origin authorization and validation falls short for the case when an AS modifies the AS-path without changing the end AS. This path might be inexistent or rogue, however is valid as long as the originating AS is not changed. To illustrate, ROA does not stop AS L in figure 3.1 from announcing either the route $147.83.128.0/17 : [L, 21, 31]$, or the shorter $[L, 31]$ to its neighbors. This originator of the route is the legitimately owner of the IP-network, and every neighbor of L will be in a position to deem as valid the route.

ROA along with RPKI fall short on validating the AS-path announced in a route advertisement. This verification consists on checking the existence of the path within the AS-graph. Solutions like ps-BGP [65], so-BGP [90] and IRV [34] have proposed the

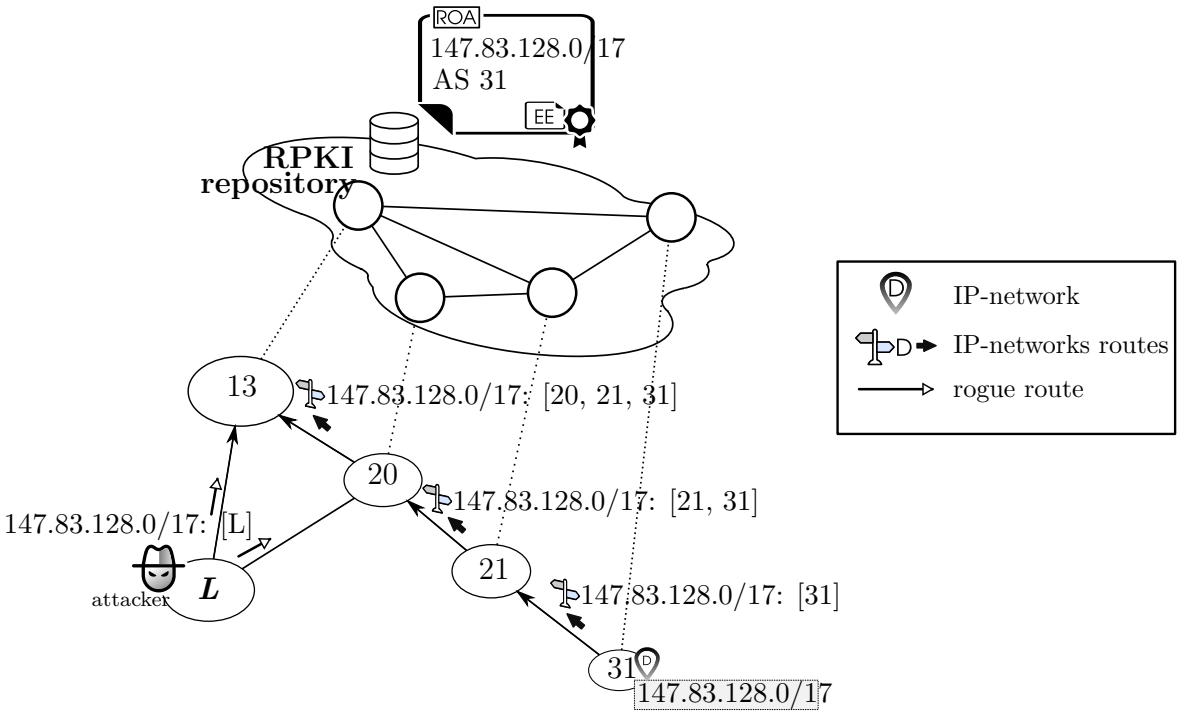


Fig. 3.3 Mitigation example of false IP-prefix origin advertisement using ROA.

validation of the AS-topology. This validation ensures the existence of a path starting at the advertising AS toward the originating AS.

Nevertheless, the verification of path existence in the AS topology does not preclude the advertisements of existent paths by an AS, even when some of the intermediate domains have not agreed to export the route. For example, consider the case when the AS L announce to its neighbors the route $147.83.128.0/17 : [L, 23, 31]$ in figure 3.1. This route exists but AS 23 may have not agreed to propagate it toward L, as it may go against to its export route policy.

3.3.3 Path verification

The false AS-path vulnerability stems from the lack of not only the verification of the path existence in the AS-graph, but also that each of the ASes in the path actually participated in the route announcement (i.e. how to verify the correct route propagation). In effect, the AS-Path contains a sequential list of all the ASes that a specific route passed through. Thus, securing route propagation refers to securing the AS-Path attribute of a particular route. The BGPSEC proposal provides such mechanism based on public key cryptography and forward signing to secure.

BGPSEC allows a domain to verify the authenticity of a route announcement all the way back to the originator domain. This implies that each domain must include the information for the route validation received from the announcer along with its announcement. This feature allows both the path verification plus the proper route propagation. However, the introduction of changes in the structural and operational aspects of BGP have made the BGPSEC proposal prone to rejection. The structural changes are due to the need of new optional attributes, while the operational ones take into consideration ROA validations and the signatures verification.

3.3.4 Prefix filters

All the solutions reviewed so far provide mechanism for validating the correctness of BGP announcements. However, neither restricts the export policies an AS can use. All the defenses described above remain vulnerable to the issue of validating routes that goes against to the “valley free” exporting routes guidelines described in section 2.2.4.

A common practice to control the allowed IP-prefixes announced by a domain is to enforce prefix filters. This first line of protection enables the definition of BGP announcements whitelists. Goldberg et al. [32] have showed the effectiveness of this technique to counter this problem. However, it falls short in the case the attacker is

not a Stub AS. Furthermore, other main problems with filters in large domains are the administrative overhead and the cost of maintaining them updated. The timely and accurate maintenance of route filters becomes challenging as the number of allowed prefixes increase up to thousands, due to the administrative burden. As a result, the ASes prefer to rely on trust and do not maintain up- to-date prefix filters—hence saving their high maintenance cost.

3.4 Summary and Contributions

This chapter has focused on describing the main issues and vulnerabilities around the current inter-domain routing system. Therein, we have described different threats faced by the BGP routing system, which affect both the protocol and the routing decision processes at the nodes. Specifically, we have highlighted the current vulnerabilities that are being partially solved by the Internet community. Conversely, the adoption of these solutions have faced some resistance due to the changes involved, and the consequent trade-off between security and its costs. While some solutions require the deployment of new, parallel infrastructures to support and improve the security in BGP (e.g. RPKI), others require changes to the BGP protocol itself (e.g., BGPSEC) with impact on the routers' hardware as it requires in-line cryptographic capabilities.

Our approach to describe the BGP vulnerabilities includes the definition of a BGP threat model. This model considers one bad player, i.e., the attacker, who seeks to exploit BGP in order to disrupt or disguise other players. The vulnerabilities addressed include Prefix and sub-prefix ownership, False AS-paths and Export policy violation. Subsequently, we describe the different efforts proposed by the community to counter these vulnerabilities. As a result of this research, the following paper has been published:

- M.S. Siddiqui, D. Montero, R. Serral-Gracià, X. Masip-Bruin, M. Yannuzzi, "A survey on the recent efforts of the Internet Standardization Body for securing inter-domain routing," *Computer Networks*, Volume 80, 7 April 2015, Pages 1-26, ISSN 1389-1286. DOI: 10.1016/j.comnet.2015.01.017.
  Contribution: I focused my efforts into researching the current BGP vulnerabilities and open issues. We surveyed the vulnerabilities that remain unsolved even with the SIDR solutions in place. Furthermore, we focused our efforts into understanding the reasons behind the slow adoption of RPKI and ROA. To this end, we studied the impacts and obstacles faced by these proposals along with their scalability issues and required changes over the protocol. We approached these

problems by studying 1) the overhead incurred by the large RPKI repository in terms of both number of objects, and synchronization time; and 2) the BGPSEC overhead impact derived from the on-line forward signature of BGP messages and their backward signature verification.

Chapter 4

BGP security offloading: the "Route Leak" threat

The Inter-domain BGP protocol vulnerabilities persists, as described in Chapter 3, due to the non-effective or unadopted solutions already proposed by the community. The reasons behind might be influenced by how disruptive these solutions are, or the lack of incentives to adopt them. Thus, we believe a different approach for securing the BGP protocol with low impact over the adopting AS domains is feasible. Our proposed approach focuses on minimal changes to the BGP control plane protocol, seeking to decouple the security protection from the protocol. To better elaborate our idea, we shall study a specific BGP security threat, defined as "Route Leaks", and how we attempt to address it in a non-invasive integration with BGP.

The offloading paradigm within BGP can be designed and devised under two different conditions, based on the location where the BGP decision process is being executed. Legacy BGP-enabled networking devices execute a distributed BGP control plane, where each element takes local decisions. On the other hand, the new SDN paradigm proposes the decoupling of the control plane from the data plane networking devices, which results in the offloading of the whole BGP routing and decision process to a controller element.

Our security proposal seamlessly intercept the BGP control information received by a domain and process it prior it reaches the BGP decision process. The collected information is used for executing real-time analytics verifying possible route leaks. This offloaded process can result in accepting the control information or denying it. Meanwhile, the BGP routing process remains unaware of this security process. The decoupling of the security to tackle route leaks is aligned with the new IETF SIDR proposals for BGP security, including RPKI and ROA.

4.1 Outline

Section 4.2 describes the "Route Leak" BGP vulnerability and our proposed technique to detect it. This technique has been designed and devised to be self-contained, which relies only on real-time analytics gathered by the border routers of an AS (e.g., the Routing Information Base RIB). Next, we present the offloading architecture design for decoupling the route leak detection from the protocol. Finally, we validate our proposal through first exhaustive simulations and a proof-of-concept implementations.

4.2 Inter-domain "Route Leak" vulnerability

A "Route Leak" is a threat within the inter-domain routing system where an attacker domain violates its "normal" export policy by announcing a legitimate route to too many of its neighbors. These violations by default should not occur as they hinder the domain business interests and the agreements with its neighbors. The guidelines recommended for route export follow these rules (the valley-free rules [69]):

> **Rule 4.1.** Routes learned from Customers can be further advertised to other Customers, Peers and Providers.
>
> **Rule 4.2.** Routes learned from Peers can be further advertised to Customers only.
>
> **Rule 4.3.** Routes learned from Providers can be further advertised to Customers only.

Any exported route that violates any of these rules represent an anomaly for the routing system, which impact could reach its directs neighbors and further propagate.

4.2.1 "Route Leaks" definition

> **Definition 4.1.** An AS's route announcement which is in violation of either valley-free rules $R.4.2$ or $R.4.3$ constitutes a Route Leak.

In other words, any route advertisement by an AS which infringes the valley-free rules $R.4.2$ and $R.4.3$ is a Route Leak. Note that rule $R.4.1$ cannot be infringed, since an AS can always export customer routes independently of the business relationship with the addressee neighbor. The above definition reveals two possible types of route leaks, viewed from the perspective of the addressee neighbor (i.e., the victim domain receiving

announcements). It is in its interest to be able to detect whether a specific learned route is in concordance with the valley-free rules as well as their business relationships.

- **Customer route leak**: this group encompasses the case when the attacker (leaker) is the victim's customer.
- **Peer route leak**: conversely, in this group the attacker is a peer of the victim.

In both cases, the attacker attempts to deceive its neighbor by announcing non-compliant routes learned from other providers or peers.

4.2.2 Route Leak identification

The identification of route leaks is the first step toward tackling this vulnerability. We shall systematically analyze the different environments vulnerable to the attack and propose a mechanism for their identification using the definition of valley free rules stated in the previous section.

Our proposal for the identification considers the following:

- The route leak identification analysis only uses readily available data, e.g., information obtained directly from the BGP routing system.
- A domain AS knows with complete certainty the business relationship hold with its direct neighbors. This has direct implications when trying to infer a route leak.
- This analysis is decoupled from the BGP protocol.
- The detection of a leak implies the proper reaction over the domain's BGP routing system.

We particularly exclude from our study data obtained from external sources, such as route information imported from vantage points. In this sense, our identification analysis focuses on what can actually be inferred in a domain under realistic routing conditions by solely examining the learned routes from its neighbors.

We start our study by defining two facts that we shall use later on while formalizing the identification of route leaks.

> **Fact 4.1.** An attacker can produce a route leak targeting to either its providers or peers.

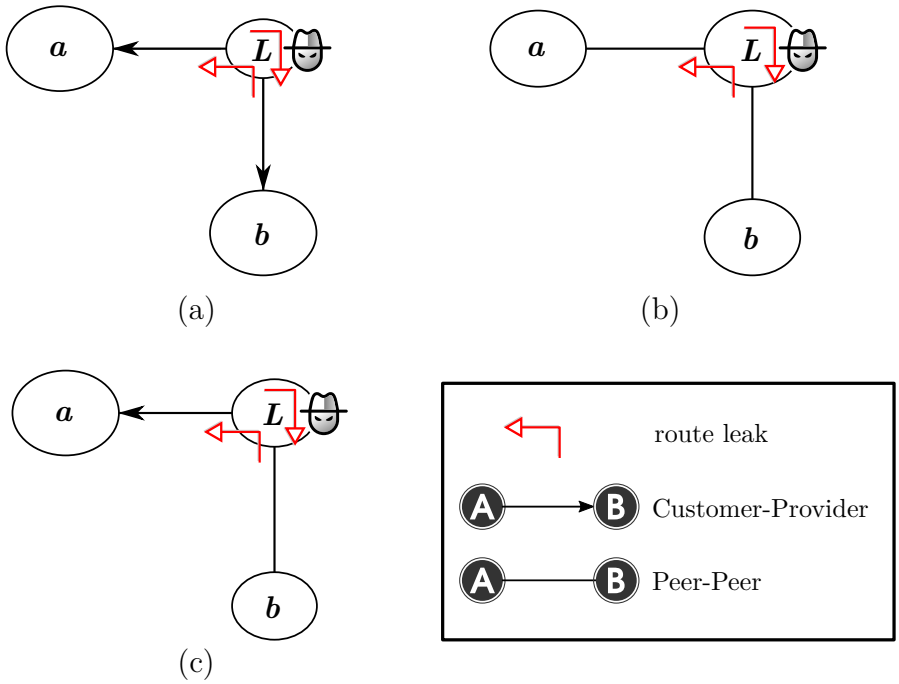


Fig. 4.1 Possible cases for an Attacker "L" to leak routes; the leaks can happen in both directions which implies "a" and "b" can be the victims.

According to the business nature of a Provider AS, it cannot leak a route toward its customers since it inherently has the role of providing transit to them. Thus, it can advertise "all" its own and learned routes to them. Directly derived from fact *F*.4.1 and definition *D*.4.1, we obtain the cases where a route leak is possible.

> **Fact 4.2.** A route leak is considered a leak only when a legitimate route learned by the attacker from its peers or providers is announced to its victim (i.e., the attacker announces legit routes to too many neighbors).

Fact *F*.4.2 indicates that a route leak consists of an actual, valid route with respect the prefix ownership as well as the AS-path advertised. This excludes other types of vulnerabilities with respect of false prefix ownership and false AS-paths announcements, as discussed in section 3.2.

To illustrate, let us assume a reference AS L as the leaker, as shown inf figure 4.1. The attacker can leak legit routes among its neighbors, depicted as a and b, as long as they are either providers or peers. Thus, figure 4.1(a) presents the case of route leak between two providers of L. Likewise, the figures 4.1(b) and 4.1(c) represent the cases of leaks between a Provider and a Peer, or between Peer neighbors, respectively.

It is worth mentioning that an AS domain victim of a route leak is only aware of the AS relationships with its direct neighbors, and has no information about the relationships

that its neighbors have with their respective neighbors. A BGP route contain the identity of all the domains required to traverse in order to reach a network, however, remains unaware of their type of relation. This is because an AS domain has limited knowledge of the network, since the relationships and policies among ASes are kept confidential. The challenge for a victim AS is thus to independently detect route leaks despite the lack of information.

Let us consider a network topology scenario for generalizing the local identification of a route leak. Figure 4.3 depicts the case where our reference AS a is the victim AS receiving new route advertisements from its neighbors. The goal is to examine under which conditions AS a can locally validate these advertisements prior to accept them into its BGP routing tables RIB and FIB. Domain b represents a neighbor that is directly connected to AS a under a customer-provider or peer-peer relationship, and it is the suspect of leaking routes (the attacker). Furthermore, AS c is a direct neighbor of b, which advertises valid routes to AS b of the form $[c, ...]$ (the $\ldots$ refers to zero or mores ASes in the AS-path route). These routes can be potentially announced by b to a , e.g., routeds of the form $[b, c, ...]$. These announcements can be identified as leaks by the victim whether they are against the valley-free rules. However, from AS a's perspective, the announcements cannot be validated due to the lack of information about the type of relationship between the suspect b and its direct neighbor c.

The leak scenarios considered so far state that a route leak occurrence contemplate three actors: the victim, the leaker and the leaker's neighbor (either a provider or a peer domain) who owns the route. However, for the sake of generality, we consider the case when the suspect b also leaks the routes learned from its neighbor's neighbor, i.e., routes learned by AS c of the form $[c, ..., d]$. Considering that the Internet is a connected graph, it is sound to assume that before the leak occurrence, the victim has a valid route to d, of the form $[..., d]$. When the suspect AS b leaks the route to reach AS d through itself (i.e., AS b announces to a a route of the form $[b, c, ..., d]$), the victim will be in a position to observe a new route advertisement for the same destination network. This reference topology and the general assumptions that we will make next shall be used in the remainder of this section, while formalizing the identification of route leaks in Theorems 1 and 2.

> **Hypothesis 4.1.** The state of the routing database of the victim AS is valley-free valid before the route leak occurs.

Remark. The purpose of our theoretical study is to capture what the victim AS can infer upon a route leak. Therefore, our analysis is focused on the transition from a valley-free valid routing state to the routing state right after the leak. In summary, $H.4.1$ indicates

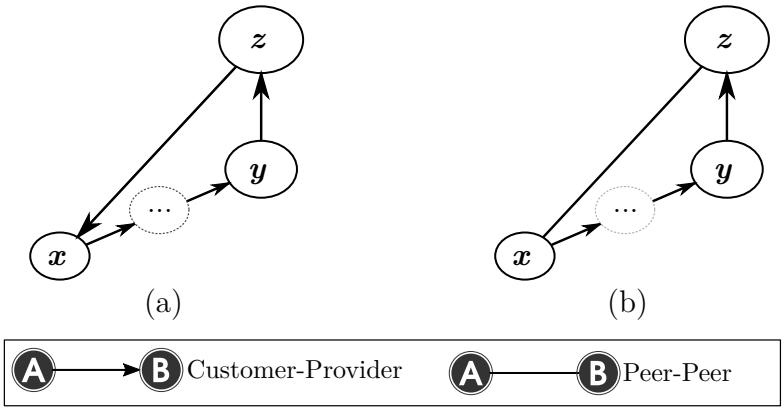


Fig. 4.2 Unlikely AS relationships: (a) hypothesis $H.4.2$, and (b) hypothesis $H.4.3$

.

that any route contained in the initial state of the RIBs at AS a is compliant with $R.4.1$, $R.4.2$ and $R.4.3$.

Hypothesis 4.2. A cyclic chain of provider relationships among ASes is non-existent.

Remark. This hypothesis means that we assume an Internet that is loop-free in terms of provider-customer relationships. Figure 4.2(a) illustrates this case, where it is implausible that AS x is the provider of the provider of its providers. It is a common assumption in the literature that Internet topologies can be modeled as Directed Acyclic Graphs (DAGs) [38].

Hypothesis 4.3. An AS does not have a peer relationship with the providers of its provider.

Remark. This hypothesis is based on the assumption that a provider AS is much larger than the customer AS in terms of infrastructure. As shown in figure 4.2(b), it is very unlikely that AS x has a peer relationship with a provider of its providers, since a very large provider z will have no economical incentives for peering with a domain x at lower tiers of the AS hierarchy. On the contrary, the incentive will be to charge AS x for the transit traffic.

Now, given the valley-free rules, and the hypotheses defined above, we proceed to formalize the conditions for detecting leaks from a customer (refer to figure 4.3(a)).

Theorem 4.1. Customer Route Leak (CRL). Let the initial state of the routing databases of an AS a contain the following:

- A direct route to a customer AS b, i.e., $[b]$.

- An alternative route to the customer AS b via AS b's direct neighbor AS c, i.e., a route of the form $[\ldots, c, b]$.

Under the hypotheses $H.4.1$, $H.4.2$, and $H.4.3$, if AS a receives a route from its customer AS b with AS-path $[b, c, \ldots]$, then AS a can identify it is a route leak.

Proof. According to rules $R.4.1$–$R4.3$, AS b could only advertise a route with AS-path $[b, c, \ldots]$ to a, iff, c is a customer of b. This is because if c is a peer or provider of b, then b is not allowed to advertise routes learned from c to its provider AS a. Let us suppose then that c is a customer of b. We know that the initial state of the routing databases at AS a contain a route to b with AS-path $[\ldots, c, b]$. Now, a could only receive the route to b with AS-path $[\ldots, c, b]$, iff, a belongs to the customer cone of c. This is because according to $R.4.3$, c would advertise its provider routes only to its customers. But if a belongs to the customer cone of c, then this contradicts $H.4.2$, since there is a cyclic chain of provider relationships among a, b, and c, that is, a is a provider of b, which is a provider of c, which in turn is provider of a. We conclude that AS c cannot be a customer of AS b. This implies that c is either a peer or a provider of b. Hence, the route announced by AS b toward AS a with AS-path $[b, c, \ldots]$ is a route leak. □

Theorem 4.2. Peer Route Leak (PRL). Let the initial state of the routing databases of an AS a contain the following:

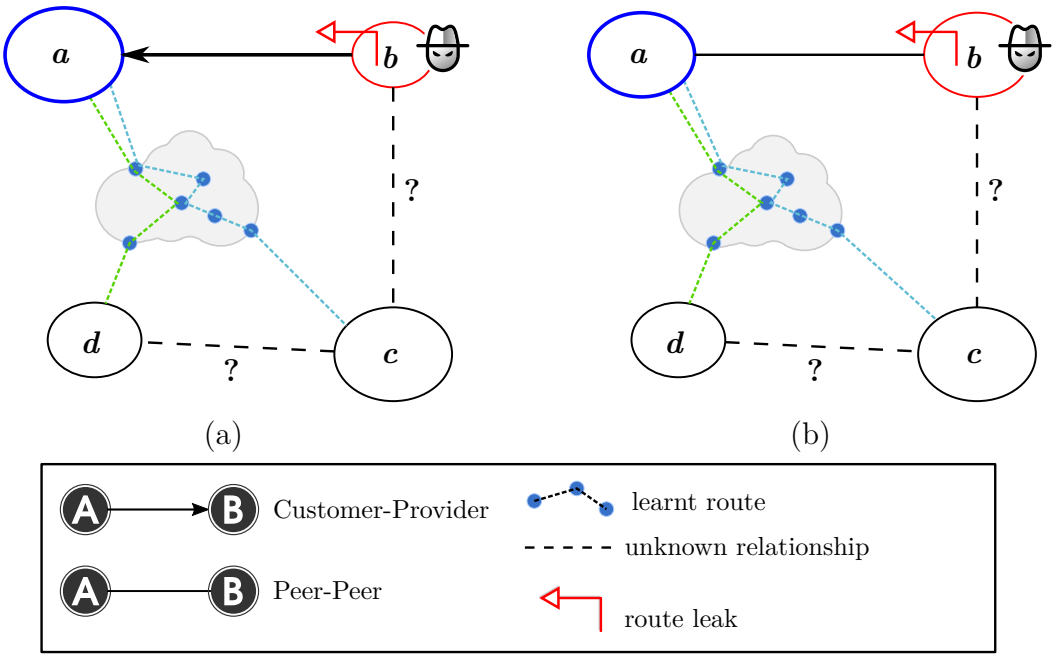


Fig. 4.3 Generalized topologies for route leak detection: (a) Customer Route Leak (CRL), (b) Peer Route Leak (PRL).

- A direct route to a peer AS b, i.e., $[b]$.
- An alternative route to the peer AS b via AS b's direct neighbor AS c, i.e., a route of the form $[\ldots, c, b]$.

Under the hypotheses $H.4.1$, $H.4.2$, and $H.4.3$, if AS a receives a route from its peer AS b with AS-path $[b, c, \ldots]$, then AS a can identify it is a route leak.

Proof. Similarly as in Theorem 1, AS b could only advertise a route with AS-path $[b, c, \ldots]$ to AS a, iff, AS c is a customer of AS b. This is because if AS c is a peer or provider of AS b, then AS b is not allowed to advertise routes learned from AS c to its peer AS a. Let us suppose then that AS c is a customer of AS b. We know that the initial state of the routing databases at AS a contain a route to b with AS-path $[\ldots, c, b]$. Now, a could only receive the route to b with AS-path $[\ldots, c, b]$, iff, AS a belongs to the customer cone of AS c. This is because according to $R.4.3$, c would advertise its provider routes through b only to its customers. But if a belongs to the customer cone of c, then this contradicts the hypothesis $H.4.3$, that is, a has a peer relation with the provider of its provider. Therefore, we conclude that AS c cannot be a customer of AS b. This implies that c is either a peer or a provider of b, and therefore, the route advertised by AS b toward AS a with AS-path $[b, c, \ldots]$ is a route leak. □

4.2.3 Cross-Path (CP) route lead identification technique

The Cross-Path (CP) technique is based on the theoretical route leak countering framework described in the previous section. Algorithm 1 summarizes the Cross-Path logic for identifying route leaks. The CP utilizes information available in the router RIBs as well as the information about the business relationships with neighbor ASes. Observe that, at the beginning of the detection process, the assumption is that the RIB tables are initially correct (i.e., they are free from entries derived by neighbor route leaks). A common solution to ensure the valley-free property of the routes is to momentarily set up route filters for all incoming BGP updates. This is only required for an initial bootstrapping period to ensure that the BGP routers only hold valley-free routes. Once the CP route leak detection technique has started, the route filters can be removed—or they can be kept though with the advantage that they neither need to be maintained nor updated.

For every incoming route advertisement from a neighbor customer or peer AS, the algorithm looks for an existing cross-path in the router RIBs considering the hypothesis and conditions outlined in the previous section. In order to make the cross-path checking

more rigorous, we can generalize the cross-path check in the form $[\ldots, o, \ldots, l, \ldots]$ in the valley-free valid RIBs. In this case, a received route from a customer or a peer AS l of the form $[l, o, \ldots]$ can be declared as a route leak if the route $[\ldots, o, \ldots, l, \ldots]$ exists in the valley-free valid RIBs. If a cross-path is found, then the received route advertisement is considered a route leak and discarded, otherwise, it is included in the valley-free RIB.

Another particularity of our algorithm is that it uses the set of public Tier-1 ASes as input for detecting route leaks. Specifically, we consider the route advertisement received from a peer or customer AS a route leak if it contains a Tier-1 AS in the AS-Path. This logic is different from [61], where the author considers a route advertisement as a route leak only if it contains more Tier-1 ASes than a predefined threshold. Based on our route leak identification framework, we contend that it is highly unlikely that an AS learns a route to a Tier-1 AS or a route to any destination via Tier-1 through a neighbor customer

Algorithm 1 CP identifies whether a new route advertisement $\mathcal{R}$ received by AS v is a leak.

Input: Valley-free $RIBs$ - Routing Information Bases at AS v
$\mathcal{N}_c$: Set of customer neighbors of v
$\mathcal{N}_{pe}$: Set of peer neighbors of v
$\mathcal{N}_{pr}$: Set of provider neighbors of v
$\mathcal{T}$: List of Tier-1 ASes
A new route advertisement $\mathcal{R}$ of the form $[l, o, \ldots]$.

Output: **true** if the new route received is a leak
false otherwise.

if AS $l \in \mathcal{N}_{pe} \cup \mathcal{N}_c$ **then**
  for all $a_i \in \mathcal{R}$, where $0 < i \leq \mathcal{R}.length$ **do**
    if $a_i \in \mathcal{T}$ **then**
      $\mathcal{R} \leftarrow \emptyset$
      return true
    end if
  end for
  $\mathcal{R}' \leftarrow [\ldots, o, \ldots, l, \ldots]$
  if $\mathcal{R}' \in RIBs$ **then**
    $\mathcal{R} \leftarrow \emptyset$
    return true
  end if
end if
$RIBs \leftarrow RIBs \cup \mathcal{R}$
return false

or peer AS. In this regard, our approach is more comprehensive and encompasses the logic used in [61].

4.3 Route Leak detection (RLD) offloading

Outsourcing the BGP security targets to decouple the security from the BGP protocol with the aim of minimizing the impact over an AS routers' installed based as well as on the protocol itself. In this section, first we discuss the reasons behind our propose to decouple and offload the Route Leak Detection, and more generally to offload the security. Next, we present two approaches which focus on transparently improving the BGP security, with their implications to both the routers software and the protocol specification. The key difference between them derive from where the BGP control plane is executed, as well as how the BGP control information is intercepted and properly diverted to an external entity.

4.3.1 Rationale behind our offloading approach

Throughout our research we have identified resistance from the community to different BGP security proposals that require either major changes to the protocol, or whether their implementation would reveal confidential business information among ASes. We believe that proposals such s-BGP, so-BGP, ps-BGP and IRV have failed to be adopted mainly due to these reasons.

Similarly, the SIDR proposals have faced acceptance as well as resistance within the research and industrial communities. RPKI and ROA are already standardized, implemented and are in an adoption phase. They have been partially successful thanks that they do not require direct changes to the protocol, and can be implemented as an independent, parallel infrastructure. ROA supports the route origination authorization validation, which can influence the acceptance of a route announcement. This verification can be performed by any domain independently, without affecting other domains who lack ROA support. More specifically, RPKI and ROA are backward compatible and do not require a wide adoption to start enhancing the BGP security.

However, BGPSEC is still being discussed and is facing resistance, especially from industry. The main reasons behind the strong opposition against it are related to the induced overhead and the disclosure of sensitive information. The security data added to each update generated by an AS's BGP router, and consequently their on-line

cryptographic processing overhead are strong arguments against the solution. Similarly, the BGPSEC forward-signing approach inadvertently reveal or leak sensitive information regarding the peering relationships between domains, at the level of BGP-speaker routers. These issues reveal that BGPSEC seems to create problems that are worse than what it was designed to solve.

These arguments suggest that introducing big changes to the BGP protocol, even to enhance its security, is a daunting and challenging task. We argue that any BGP security solution requiring disruptive changes to the BGP protocol, or that discloses peering relationship information will face fateful resistance for its adoption. In this regard, it is crucial to explore different ways by which a proposed security mechanism could be integrated into the existing inter-domain routing system and foster adoption while avoiding collateral burden.

We approach to this problem from a perspective of decoupling and offloading the security chores from BGP. Therein, the security chores are decoupled from either the protocol or the whole data processing elements toward and external element. This new element in charge of the security protection is provisioned with better capabilities, and can control a set of network elements. This idea is aligned with the concepts introduced by Software Define Networks (SDNs). The following subsections further describe two approaches to achieve the BGP offloaded security.

4.3.2 BGP messages intercept approach

The first approach consists on intercepting the BGP packets at the routers and diverting them toward a decoupled, external security application by means of a remote procedure call (e.g., through the use of an API). This decoupled application is on charge of control and verify the validity of the BGP routing information and take a decision. If the results is correct, the application re-inject the BGP information toward the BGP control plane at the network element. Meanwhile, on the other case the application can drop the BGP information, without modifying its routing and control state.

To illustrate, figure 4.4 represents the basic offloading concept by showing two neighboring domains and the interactions between the different components. The vertical interaction between the BGP routers and RLD application for route leak detection seeks to enable non-disruptive control over the BGP process running on the router. This control is achieved by intercepting the BGP information for inspection. The BGP control process remains distributed, each controlling the BGP routing information locally, and unaware of the security enforced.

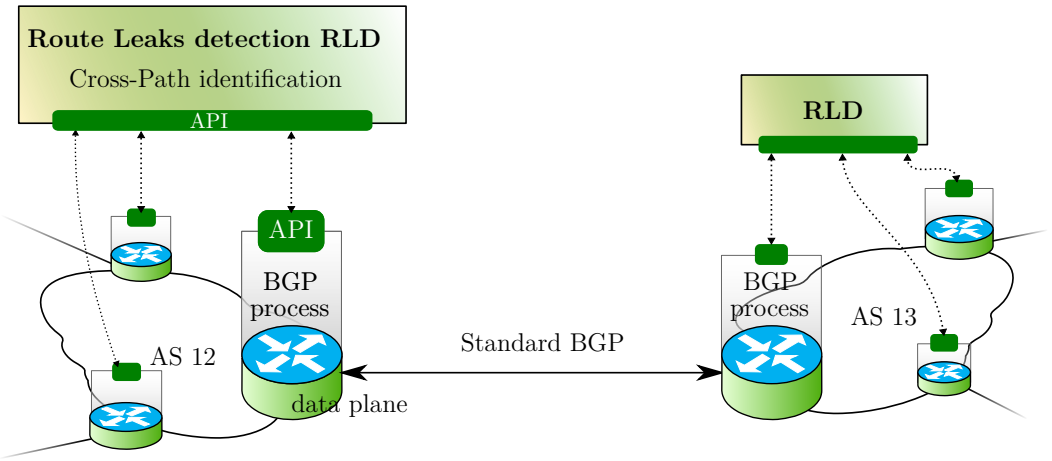


Fig. 4.4 Route Leak detection decoupling with distributed BGP control plane.

Zooming in into a BGP router, figure 4.5 depicts the proposed architecture and components of our proposal. The interactions and coupling between these components aim at a seamless and non-disruptive integration with the existing routing system. Two "entities" can be clearly identified: 1) the BGP routers and 2) the RLD layer, which is the one on charge of executing the Cross-Path route leak detection. At the BGP router, we split our architecture into two subcomponents, i) *BGP Messages Manager*, and ii) *Messages Broker*. The former represents a technology-agnostic component which handles the new BGP announcements independent of the BGP process technology used, and interacts with the RLD layer, while the latter is the modifications within the BGP Process to support the offloaded security (e.g., the modifications and extensions to the Quagga BGP process).

Inside the Router Operating System

This section specifies the interactions between the RLD layer and the BGP processes internally in the router. We describe the general architecture of our proposal, which have been implemented by leveraging the Quagga's router implementation [70]. Our prototype leverages its event driven system with internal queues to optimize the BGP processing and to schedule and prioritize different tasks.

Figure 4.6 depicts in a nutshell our approach to decouple and intercept the BGP update messages. Next, we describe the tree main components:

- *Messages Broker*: This is the technology-dependent part which plugs in in into the BGP Messages Process to intercept and divert the update messages toward the *BGP Messages Manager*. Furthermore, this component also enables the re-injection

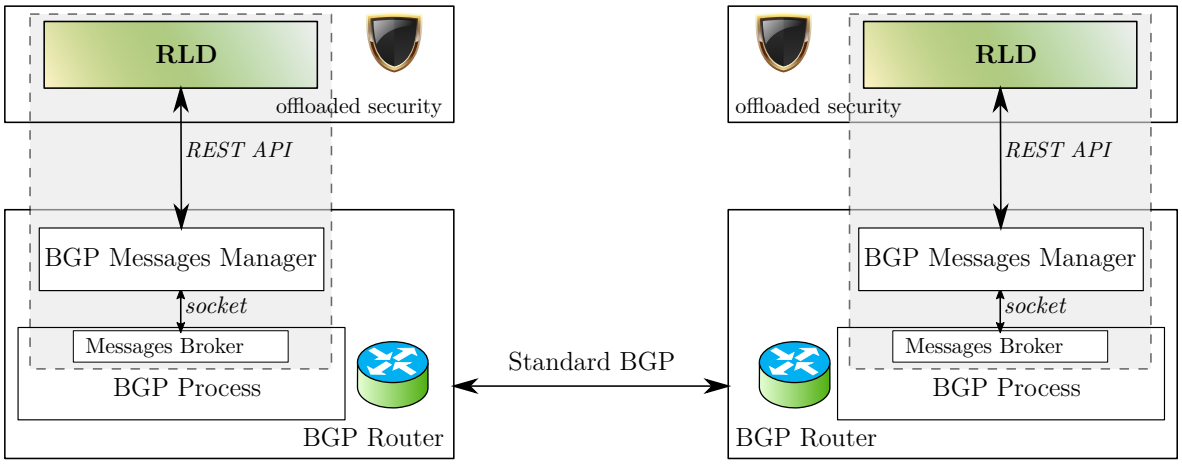


Fig. 4.5 General view of the architecture and the interactions between the layers. The bottom layer is provided by the BGP protocol, whereas the top layer is provided by the RLD.

of the update messages back to the BGP process to be further processes. Note that, the BGP decision process remain intact, and when an announcement is considered valid by RLD, it is BGP itself which process the update.

- *BGP Messages Manager*: this component provides a layer of abstraction to interact with the RLD layer. It provides the necessary APIs to the *Messages Broker* to be able to process a new update announcement, queue it and further transport it to

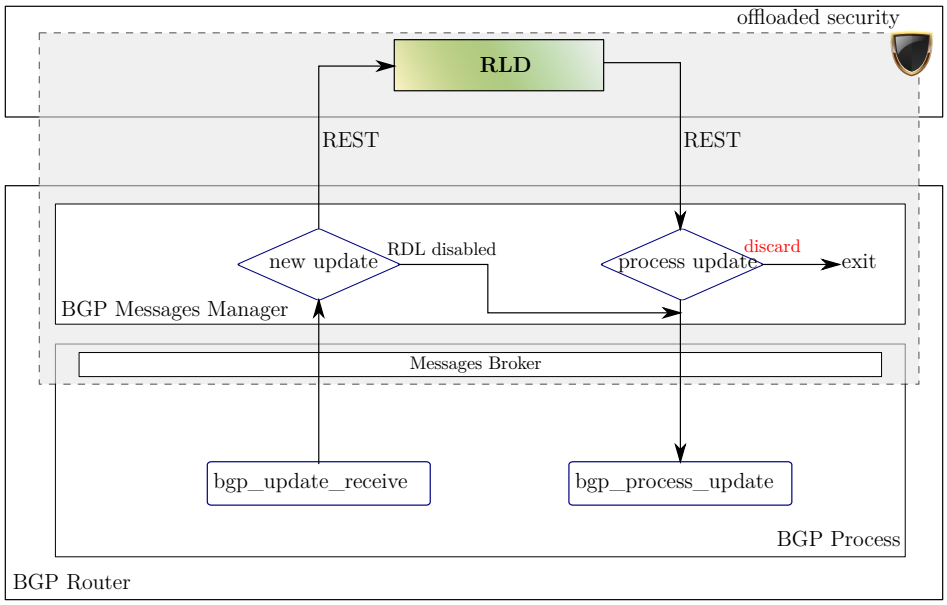


Fig. 4.6 Offloading operation within Quagga and interaction with the external RLD.

the RLD layer. Similarly, it processes and further re-injects the validated BGP update messages received from the RLD layer. This component runs in a separated process, along with the BGP process. As a security measure, if the RLD layer is unavailable, the update messages are forwarded back to the BGP process.

- *RLD layer*: this component implements the route leak detection technique. This layer interacts with the BGP Messages Manager of different routers within an AS domain. All the recollected routing information is processed and maintained by our RLD detection process. This knowledge shall help to process new announcements seeking to detect whether they are route leaks. The result back to the routing elements per each new update message is either Discard or Accept.

It is important highlighting that only minor modifications have been made to BGP for supporting the communication and coordination between the bottom BGP layer and the external RDL security layer. All the functions of the BGP speaking system remain unmodified, except for the way in which BGP advertisements are controlled and processed. Events that involve the processing of updates received from a BGP neighbor are inspected by the RLD layer, and depending on the RLD validation result, the updates might follow the normal BGP flow or be discarded.

4.3.3 SDN like approach

Software Defined Networks (SDN) proposed the the decoupling of the control plane from the networking elements, and positioning it into a central controller. The controller becomes the central coordinator of different network elements, and decides how the traffic is steered through them. In this environment, the BGP process is an application running in the at controller, which understands the BGP protocol. This results in having only

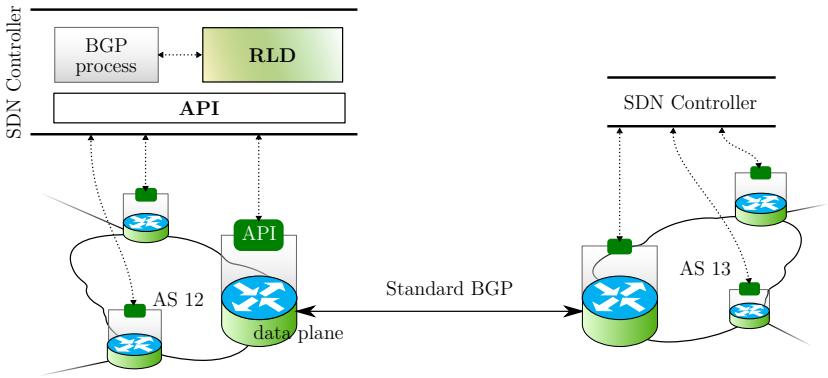


Fig. 4.7 SDN-like Route Leak detection.

one BGP process per set of edge router. Similarly, our RLD application can be deployed in the controller, interacting directly with the BGP process. This improves the network control and visibility, while fostering the development of new network applications like our RLD for example.

A realization of this approach is the "BGP LS PCEP" [66] project under the umbrella of OpenDaylight (ODL) [57]. To illustrate, figure 4.7 depicts the SDN architecture, which remarks the decoupling of the control plane from the network elements, while providing a southbound API for the element–controller interaction. ODL integrates different southbound plugins to support a range of protocols including Openflow, LISP, BGP and Netconf. On top of the controller, the BGP control process resides alongside with our RLD application. Furthermore, the controller also provides a northbound APIs to interact with external applications (e.g. by exposing a standardized REST/NETCONF API.).

4.4 Evaluation

This section describes the experimental and simulation setup used to evaluate both the CP route leak detection technique along with the offloaded detection idea. First, we shall address the simulation setup and the obtained results. To contrast these results, next we delineate the experimental proof of concept along with its results for the RLD offloading idea. In this work, we have evaluated the first offload approach described in the previous section, i.e. the offloading of the RLD application only, and the interception of the BGP packets. As future work, we plan to continue our efforts aligned with the Opendaylight BGP LS PCEP project.

4.4.1 RLD simulations setup and results analysis

In order to validate the CP route leaks detection RLD technique, we have tested it through exhaustive simulations using a scaled-down version of an actual Internet topology. It is worth mentioning that the cost for running event-driven simulations on a complete Internet graph is prohibitively expensive in terms of time and resources. Thus, it is a common research practice to utilize small realistic graphs, extracted from the actual Internet graph, to avoid excessive computational cost [19].

In our testing framework, an important consideration during topology reduction was to preserve some of the essential properties of the complete Internet graph, e.g., average node degree and degree distribution. This particular serves as the base for rationally

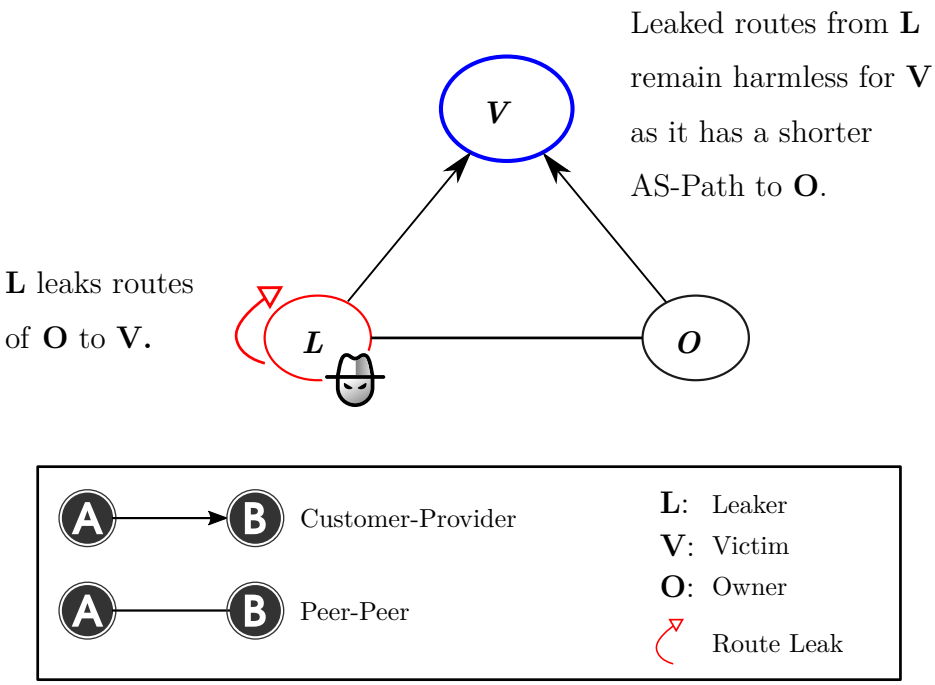


Fig. 4.8 General representation of a harmless route leak scenario.

extrapolate the obtained results to larger, more realistic Internet topologies. Furthermore, we assured that the scaled-down topology contains all the AS relationships considered in our route leak detection framework with the potential to produce all different types of route leaks. For this purpose, we obtained a subgraph of ARK's Internet graph (2013) [89] containing 1650 ASes and 3744 inter-domain links using graph reduction technique presented in [46]. Observe that our approach of using a subgraph of ARK's Internet graph means that the topology we used in our simulations is actually part of the Internet. In the rest of this section, we refer to the obtained scaled-down topology as *"Topology-1650"*. The simulations were setup and run using the Network Simulator NS2 [64], along with BGP++ [7] to complement NS2's BGP support. We considered a single router per AS and each AS's BGP router was configured according to its policies and relationships with its neighbors according to the extracted topology. As a result, we were able to simulate and test the RLD techniques with BGP in an Internet-like topology for different route leak scenarios and evaluate their impact.

For *Topology-1650*, we identified a total of 20,747 different possible route leak scenarios, out of which 17,151 were harmful route leaks, i.e., the route leak poisoned the RIB of the victim AS successfully. That is, although the remainder route leaks occurred, the leaked routes were not chosen as they were not the best path to the corresponding destination, thus failing to poison the BGP forwarding table of the victim AS V. One example of

Leak scenarios	Cross Path (Simulation)			
	# Leaks	# Harmful Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	4879	4773	4492	94.11
CRL (Pe)[b]	5714	3974	943	23.73
PRL (Pr)[a]	5724	5406	5044	93.30
PRL (Pe)[b]	4430	2998	177	5.90

[a] CRL/PRL cases where O is provider of L.
[b] CRL/PRL cases where O is peer of L.

Table 4.1 Simple Cross-Path (CP) Detection: Simulation results for different route leak scenarios.

such a route leak scenario is depicted in figure 4.8. Even if L leaks routes of O to V, these leaked routes will not affect the forwarding table of V. This is because on receiving routes toward O from L and directly from O itself, the victim V, following the shorter AS-Path criteria, prefers the direct shorter AS-Path route. It is important to note that the reason V decides the best route based on shorter AS-Path criteria is because V has same provider relation with both L and O. Table 4.1 shows the simulation results of CP route leak detection technique for the harmful route leaks.

We have classified the route leaks further on the basis of the AS relationship between the leaker AS L and its neighbor O, in order to analyze the results in depth. That is, we divide Customer Route Leaks CRL (the leaker is a customer of the victim) cases into two subcategories, one where O has a provider relation with L, denoted by *CRL(Pr)*, and other where O has a peer relation with L, denoted by *CRL (Pe)*. Similarly, for Peer Route Leaks PRL (the leaker is a peer of the victim), we classify them into *PRL (Pr)* and *PRL (Pe)* cases, where O has a provider and a peer relation with AS L, respectively. We observe that CP detects 94.11% and 93.30% of all the *CRL (Pr)* and *PRL (Pr)* route leak cases, respectively. Whereas, for the *CRL (Pe)* and *PRL (Pe)*, the CP detection rate is low, with results of 23.73% for *CRL (Pe)* and 5.90% for *PRL (Pe)*. The reason behind better performance of CP in route leak cases where O is provider of L is that O being the provider of L advertises L's route to all its providers, peers and other customers, thus increasing the chances for the possibility of cross-path observance at AS V. In the route leak cases where O is a peer of L, the chances of observing a cross-path involving the two consecutive peers are very low in practice, since a peer does not advertise routes of another peer any further except to its customer cone, hence the poor performance of the CP technique for those cases.

Leak scenarios	Cross Path (Experiment)			
	# Leaks	# Harmful Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	725	713	701	98.31
CRL (Pe)[b]	226	97	21	21.64
PRL (Pr)[a]	825	811	792	97.65
PRL (Pe)[b]	154	70	4	5.71

[a] CRL/PRL cases where O is provider of L.
[b] CRL/PRL cases where O is peer of L.

Table 4.2 Simple Cross-Path (CP) Detection: Experimental results for different route leak scenarios.

4.4.2 Offloaded RLD experimental setup and results analysis

Besides the simulations, we also tested the CP technique using real-time experiments. The main purpose of experimental evaluation is to verify the robustness of the technique in close to real environment. In order to build an experimental setup which is with in the computational and memory limits of our testbed infrastructure, we extracted a subgraph of Topology-1650 containing 990 ASes and 2146 inter-domain links. We used similar criterion for obtaining the 990 ASes topology (referred to as *"Topology-990"*) as we did for *Topology-1650*. Our testbed consisted of a single virtual Linux Container (Docker [20, 58]) for each AS in Topology-990 equipped with a customized BGP implementation from the Quagga routing suite. This BGP implementation was developed by our team in the framework of the projects Path-State Protocol (PSP) and OPENER [2, 94]. This framework enabled the implementation and execution of the RLD application offloaded from the Quagga BGP process. This prototype required the modification of 88 lines of the BGP-4 code to support the interception and re-injection of BGP messages, and thereby enabling the offloading of security.

Initially, as determined by our hypotheses, all the nodes were connected and configured in line with the valley-free rules. This enabled the capture and learning of routes without any route leaks. Then, for each experiment once BGP converged, a route leak was generated, i.e., an AS (L) leaked routes of one of its neighbors (O) to another neighbor (V). Then, AS V enabled with the CP RLD technique performed the detection process using the new route announcement and its prior BGP routing information. In this topology, we were able to anticipate 951 Customer Route Leaks (CRL) and 979 Peer Route Leaks (PRL) possible scenarios. Thus, we ran a total of 1930 different experiments, each with one route leak occurrence. Out of the 1930 different route leak scenarios, we were able to rule out 239 leaks that were harmless. It is noteworthy that the set of

route leak scenarios anticipated for *Topology-990* is a subset of route leak scenarios for *Topology-1650*. This is because *Topology-990* is a subgraph of *Topology-1650*. In the remaining 1691 harmful route leaks, there are 810 CRL and 881 PRL route leaks.

Table 4.2 shows the results obtained with the CP technique for the harmful route leak experiments. From the perspective of the extended classification of the route leaks, we observe a similar performance trend in the experiments as well. Therein, the CP route leak detection performance is more than 97% in both CRL (Pr) and PRL (Pr), whereas for CRL (Pe) and PRL (Pe), it detects 21.64% and 5.71% of the route leaks, respectively. The justification of these results is similar to the one given for the simulation results of the CP technique. Furthermore, it is worth mentioning that the respective route leaks, for different cases, that are detected in our experimental study were also correctly detected in our simulation evaluation. This assures the behavior stability of our technique in both modes of evaluation. The difference in success rate percentages, for different route leak types, is due to the difference in the number of leak scenarios ran in each mode of evaluation.

4.5 Open issues

Whilst the Route Leak Detection technique proposed can be applied in many practical situations (e.g., the Telekom Malaysia route leak of about 179,000 prefixes incident [8] could have been avoided), there are still some others that might neither satisfy the hypotheses of Theorems 4.1 and 4.2, nor comply with the two types of ASes relationships considered in the valley-free model. In the remainder of this Section we discuss the limitations of the proposed RLD technique.

4.5.1 Siblings and Hybrid relationships

The valley-free rules for exporting routes serve as a reasonable stepping stone toward theoretically modeling the route leak problem. However, the valley-free export rules are not necessarily satisfied under certain complex relationships between ASes, such as *siblings* and *hybrid* ASes relationships. A sibling↔sibling relation exists between two ASes which belong to, and are under the administration of a single organization. These ASes typically offer customized transit to each other, which implies a different type of AS polices among them. The proposed Cross-Path technique may fall prey to false positive detections due to lack of prior sibling relation information. This information is required

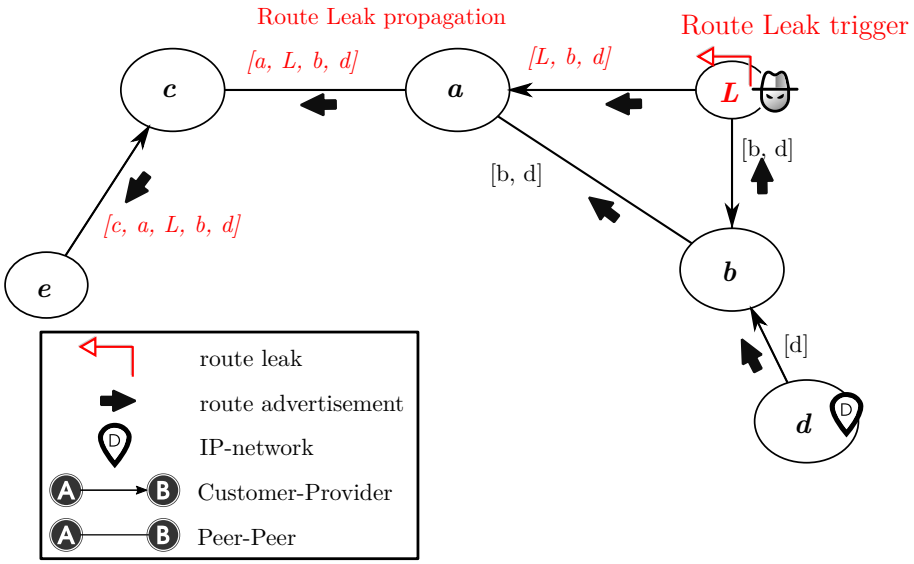


Fig. 4.9 Example of a Route Leak attack and further propagation. The victim AS a unable to detect the leak route $[L,b,d]$ to IP-network D, further announces it to its neighbor c, which subsequently announces it to e. These two ASes have not violated the valley-free rules, however the route leak has affected them.

as the proposed technique assumes that the two ASes in a sibling relation are separate entities.

On the other hand, a *hybrid* relation type refers to cases where two large ASes have different relationships between them at geographically different points of presence. For example, two ASes may have a customer→provider relation in one region and a peer–peer relation in another region. We contend that the analysis presented may even stay valid in various hybrid scenarios, since the routing information that is relevant for the detection is the one contained in the routers in proximity with the occurrence of the route leak, independently of the divergence on the routing views at geographically separated areas. However, more in-depth analysis and evaluations of these specific cases is required.

4.5.2 Route leak propagation

The proposed technique for Route Leak detection has been devised for detecting the cases when a route leak is initiated, i.e., the victim AS is a direct neighbor of the leaker. Furthermore, we only consider the cases when an attacker AS acts independently, excluding the cases of coordinated leaks attacks. These considerations leave out the detection of route leaks propagation.

A route leak propagation refers to the scenario where the victim AS receives a rogue route and forwards it further to its neighbors, thus increasing the impact over other domains. The victim AS unaware of the attack may forward the route leak to its neighbors according to the relationship it has with them, and complying with the valley-free rules. This makes more difficult for any AS receiving the propagated rogue route to detect it as a leak.

To elaborate, figure 4.9 depicts an example of a route leak attack and propagation. In this case, AS a is the victim which is unable to identify the route $[L, b, d]$ toward IP-prefix D as a leak. Thus, AS a in comply with the valley-free rules announces the new route learned from its customer L toward its peer c. Subsequently, AS c decides to announce the route to its customer e. Observe that ASes c and d are affected by a propagated leak, which makes more difficult to detect the leak. The further a AS is from the leaker, the more complex it becomes to identify it. This highlights the importance of detecting the leak the closest to the attacker.

4.5.3 Initial valley-free state

From an engineering perspective, the hypothesis $H.4.1$ is reasonably achievable by many transit domains, since route filters can be set to that end for a short period. This will ensure that the routes imported up to that stage are valley-free. Once the learned routes are guaranteed, the route filters need not further maintenance and could be removed. Observe that the reluctance of providers for using filters does not lie on their initial configuration, but rather on keeping them updated.

This dynamic method of applying and removing route filters is challenging from the perspective of very large providers, and without SIDR's solutions in place (mainly RPKI and ROA, described in 3.3), this could only be achieved through a chain of trust during filter configuration. For instance, when a client requires to advertise a new route, this requirement will have to be notified and accepted not only to his providers, but also to the providers' providers where filters are being enforced. Further research is needed on how to ensure that the initial state at the potential victims is valley-free.

4.6 Related work on Route Leaks

There has been an arising awareness regarding the research of the route leak problem, its definition and solutions' proposals. Apart from the research studies, there are a few

conventional methods, e.g., route filters, that can be used as a possible solution for the route leak problem. In this section, we discuss the research studies and the conventional mitigation methods that particularly target to resolve the route leak problem.

4.6.1 Research studies

There has been some discrepancies regarding a formal definition of "Route Leaks". To shed some light and common ground, the authors in [86] have proposed a *working Route Leak definition* as: "the propagation of routing announcement(s) beyond their intended scope...which is in violation of the intended policies of the receiver, the sender, and/or one of the ASes along the preceding AS path." The intended policies are defined in terms of the pair-wise peering business relationship between ASes. This definition as well as the four initial types of route leaks defined therein are in concordance with our definition and classification of route leaks scenarios.

The primary difficulty in solving the route leak problem lies in the secrecy of the AS relationships in the Internet. There are several AS relationship inference schemes proposed in the literature, including contributions such as [29, 19, 87]. The existing solutions typically infer the relationships between any two ASes by analyzing the BGP data collected at different points in the network, called *vantage points*. One fundamental critique on such inference schemes is that their knowledge base for inferring the AS relationships is partial, i.e., their view of the Internet is restricted to the data collection points. Ager et al. [1] highlight the limited nature of such AS relationship inference schemes, by detecting far higher number of peer-to-peer links within only one large Internet Exchange Point (IXP), as compared to the number of peer-to-peer links in the entire Internet discovered by well-known inference schemes.

Sriram *et al.*[85] examines which of the route-leak types defined in [86] are detected and mitigated by the existing origin validation (OV). The authors propose two recommendations to be followed by local AS domains seeking to prevent the creation of route leaks. Another solution involves carrying a "1-bit" per-hop route-leak protection (RLP) field in BGP updates. The RLP field is proposed to be carried in a new optional transitive attribute, which targets to help with the detection and mitigation of route leaks at ASes downstream from the leaking AS. This proposal differs from ours as it includes both the extension of the BGP protocol with a new non-transient option, and a global adoption and proper enforcement of their defined routing selection and processing rules. On top of this, the RLP bit may reveal information regarding the type of relationships among ASes in the announced AS-path. This information is not public, as each AS keeps their business relations confidential.

Sundaresan *et al.*[88] also investigate the export policy violation attacks in inter-domain routing, calling them traffic attraction attacks. In essence, to detect export policy violations they exploit the valley-free path feature that a particular BGP update once traversed through a provider customer link or a peer–peer link should not go over a customer→provider link or another peer−peer link, respectively. They propose to set a flag in the BGP advertisement when it is sent to a peer AS or a customer AS. The flag is contained in a new ATTEST attribute which is appended by all the ASes in the AS-Path. Furthermore, they proposed to include the ATTEST attribute in the signed part of the Secure BGP (S-BGP) [44] message to maintain the integrity of the flags set by each AS in the AS-Path. In this way, any AS can determine if an update received from a customer AS or a peer AS has violated the export policy rules by verifying the flags in the chain of ATTEST attribute. However, according to their results, their solution becomes effective when more than 60% of ASes deploy the scheme. The main shortcomings of this scheme is that it requires changes in the BGP protocol to accommodate the new *ATTEST* attribute. The scheme also depends on the Route Attestations (RA) and Address Attestations (AA) mechanisms of S-BGP which incur software and hardware burden of third party security infrastructure. Furthermore, this scheme requires high deployment percentage in order to be effective for a certain type of route leaks. And more importantly, the setting and signing of the flag in the *ATTEST* attribute discloses AS policies more than what are already revealed by the BGP protocol at present.

Li *et al.*[52] have studied the routing loops effects caused by route leaks. They propose a novel mechanism that identifies route leaks by monitoring routing loops. The loop is detected by any AS who receives a route announcement from a peer or a customer, whose AS number is already included in the announced AS-Path. Their solution passively monitors BGP routes to detect route leaks. In comparison, our proposal differentiates as our technique seeks for a cross-path in the announced routes' AS-paths by using only local available routing information which includes both routes with and without loops, while their proposal rely on the observance of an AS-path loop, and uses external information to complement their detection process.

4.6.2 Conventional methods

The conventional methods to mitigate route leaks include route filters, Internet Route Registries (IRRs), and BGP monitoring tools. The utilization of route filters on the BGP routers between two ASes aims at filtering out routes that are in violation or out of the scope of the agreed policies. However, the timely and accurate maintenance of route filters are challenging as the number of allowed prefixes increase up to thousands

for ASes with large customer cones. The administrative and technical burdens to keep the filters updated hinders their effectiveness. As a result, ASes tend to rely on trust and do not maintain up-to-date prefix filters.

IRRs provide an on-line structured database of route objects that can be used to automate the maintenance of the route filters. However, IRRs also suffer from high maintenance cost because the route objects in the IRRs have to be defined first and then kept up-to-date, so the route filters can be automatically maintained. Besides, IRR records are not maintained by all ASes, and existence of duplicate, false, and incomplete records have raised questions on the sanity of the information contained in IRRs.

Similarly, BGP monitoring tools, such as Nemecis [82], Prefix Hijack Alert System [48], and Pretty Good BGP (PGBGP) [43] analyze BGP data collected at different vantage points to detect irregularities. These monitoring tools have to be trained on up-to-date policies to detect any irregularity, thus causing similar administrative burdens as route filters and IRRs. Such monitoring tools are good as long as the irregularities are observed at the vantage points, so strategic attacks avoiding the vantage points can still succeed without detection. Both, BGP monitoring tools and AS relationship inference schemes depend on BGP data collected at different vantage points. However, the former utilize the data to detect irregularities against predefined policies, whereas the latter use the data to infer the business relationships and type of peering among ASes.

An interesting route leak detection solution defined in [61] proposes to count the number of predefined "Big Network" ASes in an AS-Path of a route under consideration. The set of "Big Network" ASes is composed of mostly Tier-1 ASes. This simple technique is based on the fact that an AS-Path should not contain more than two Tier-1 ASes in it. Thus, if an AS-Path contains more than the fixed threshold number (default threshold is 2) of allowed "Big Network" ASes, then it is flagged as a route leak. This solution concept is similar to the one defined in our RLD technique, that is to utilize BGP knowledge to detect the route leaks.

4.7 Summary and Contributions

This chapter describes our proposal for addressing the BGP "Route Leak" vulnerability following the offloading paradigm. First, we identify and define what a "Rout Leak" vulnerability is, its possible causes and probable consequences. We have seen real cases of "Route Leaks" reported, which denounced sudden changes on the public BGP routes with serious consequences over the Internet. Some cases have impacted large parts of Internet or major service providers.

Next, we propose a novel "Route Leak" detection solution with the advantages of relying only on the domain's available information regarding the BGP routing announcements and business relationships with its neighbors. Our security solution follows an offloading approach which does not disrupt the BGP protocol. To validate this idea, we have performed extensive evaluations in a simulation framework. Furthermore, we have also evaluated our Route Leak Detection technique on an experimental setup, which provided similar results as in the simulations. The outcomes of our work have been published in:

- M. S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracià and X. Masip-Bruin, "Route leak identification: A step toward making inter-domain routing more reliable" *2014 10th International Conference on the Design of Reliable Communication Networks (DRCN)*, Ghent, 2014, pp. 1-8. DOI: 10.1109/DRCN.2014.6816139
  Contribution: I collaborated with the research and study several real cases of Internet outages derived from abnormal exporting routing events. We approached to the "Route Leak" problem from a theoretical perspective, and proposed our definition of it. Therein, we defined a theoretical framework to identify a "Route Leak" based on the observance of a "Cross-Path" singularity. This framework is based on a set of realistic hypothesis. I also contributed with the insight obtained from replaying and verifying the leak cases in controlled environments through BGP simulations and testbed emulations.

- M. S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracià, X. Masip-Bruin and W. Ramirez, "Route leak detection using real-time analytics on local BGP information," *2014 IEEE Global Communications Conference*, Austin, TX, 2014. DOI: 10.1109/GLOCOM.2014.7037092
  Contribution: Design, implementation and evaluation of the Cross-Path Route Leak identification technique. I also contributed with the setting up, configuration and execution of experiments with large scale, Internet-like topologies over a simulator and a virtual emulator. This also includes the collection and processing of the BGP routing information for all the Route Leak scenarios. The results' analysis and interpretation, along with the deeper research and understanding of unforeseen cases form part of my contribution too.

- M. S. Siddiqui, D. Montero, R. Serral-Gracià, M. Yannuzzi, "Self-reliant detection of route leaks in inter-domain routing," *Computer Networks*, Volume 82, 8 May 2015, Pages 135-155, ISSN 1389-1286. DOI: 10.1016/j.comnet.2015.02.029

Contribution: experiments execution and evaluations of the Route Leak identification techniques. We used two different Internet-like topologies to test over each, the simulator and an emulator. I contributed with the analysis, interpretation and verification of both the simulation and the experimental results obtained from applying our detection technique.

Chapter 5

Locator-Identifier Split Protocol vulnerabilities

The Locator/Identifier Separation Protocol (LISP) [23] 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 network addressing scheme (as previously described in section 2.2.5). 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 for example. 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 [23], 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 EID(s)-to-RLOC(s) 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.

5.1 LISP threat model

Our threat model considers a single LISP site attacker seeking to exploit a mapping system vulnerability with the objective of provoking an impact over other LISP sites. The considered vulnerability exploits the lack of mechanism to check the authorizations over network resources within the locator addressing space. This problem is similar to the BGP resource authorizations, thus we plan to leverage on the proposed solutions (e.g. RPKI and ROA) to enhance the LISP control plane security.

The decoupling of control functionalities from data forwarding elements and its in-line routing learning dynamic in LISP pose a great advantage for network dynamism, however it also opens new security vulnerabilities. An extensive LISP threat analysis has been laid out in [76], which considers a threat model with emphasis on the vulnerabilities related with the interactions between the different LISP components. Therein, some of the considered vulnerabilities have been partially addressed by security solutions over the control plane. For instance, LISPSEC [55] focuses on securing the mapping discovery of unknown EID-prefix's locations. These queries include the *Map-Request* and *Map-Reply* messages, which involves three specific LISP components, i.e., the Map-Resolver, Map-Server and the ETR.

The location update of an EID prefix is determined by a *Map-Registration* process, which involves both the ETR and the Map-Server on charge of the prefix. These elements are in charge of signaling the current location of the prefix to the Mapping-System in terms of at which set of locator addresses the prefix is reachable. Thus, the security implications of this process concerns to the ETR, the Mapping System and the EID-prefix owner. The first two are considered in the current registration process, while providing a shared-key mechanism for authentication. It contemplates that the EID-prefix and the ETR are under the same administrative domain. However, the registration process enables only the authentication from the Map-Server side whether a "Map-Register" request is valid and blindly accepts the proposed set of RLOC addresses. The Map-Server falls short in the verification of whether the proposed RLOCs are valid and authorized to be used by the ETR.

Similarly, the verification of EID-prefixes authorizations is through this shared key. This method works well in the case of a fixed, non-mobile end point scenario, which holds and utilize the EID addresses for establishing end-to-end communications (i.e., the network address the applications use). However, when we consider the mobility case of an "EID-holder", which requires to change its current location with a new one while maintaining its EID address, the current registration procedure falls short.

The implications of this mobility scenario include not only the challenges on the locator registration procedure at different ETRs (which might imply the sharing of keys between ETRs at different administrative domains), but also on both the EID and locator authorizations. We study the inclusion of a third player in the EID-location registration, i.e., the EID-holder, and the impacts on the whole registration procedure. The EID-holder represents an end user and its ability to roam through different edge networks while maintaining its EID identifier. Other major challenge under this scenario is the verification granted by the EID-holder to be represented by a specific ETR and its locator address, which directly depends on the trust between the end user and the service provider offering to intermediate and route the traffic in its behalf. We shall further address this case aligned with the mobility use case, and investigate how this impacts over the LISP control-plane.

5.2 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.

5.2.1 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 5.1 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 $RLOC_2$.

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

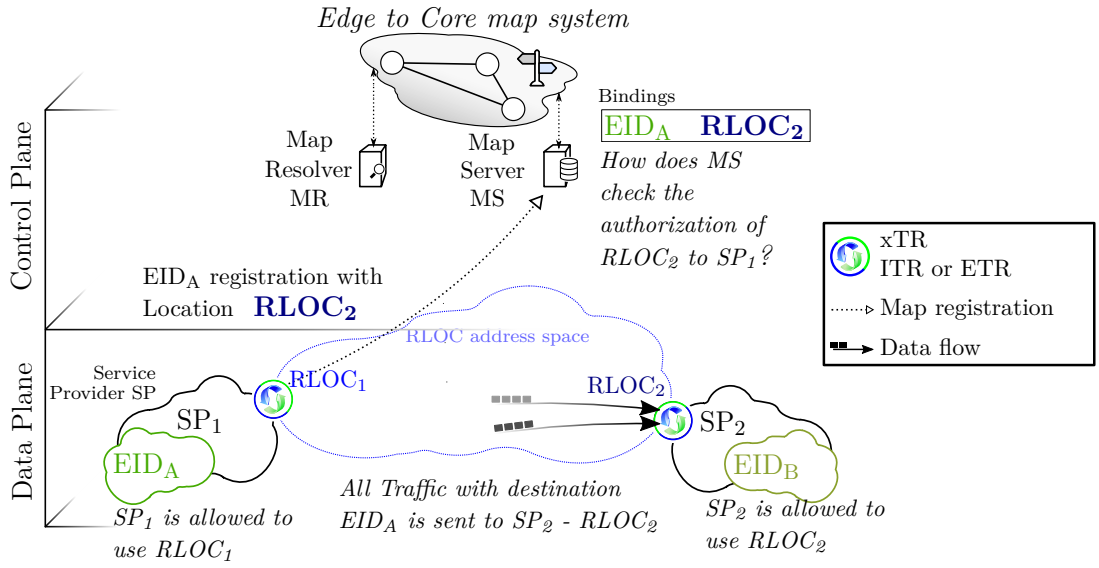


Fig. 5.1 RLOC Spoofing in LISP.

6.2, by introducing a slightly adapted version of the existing Route Origin Authorizations (ROAs) [50], we can provide dynamic assessment of the RLOC ownership and effectively avoid such attacks.

5.2.2 No Global EID Authorization

As mentioned earlier, the current map registration process completely alienates the EID-holder role, hence making it dependent on the LISP-Site's ETRs. With this approach, there is no way that a Map-Server 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 an end-to-end EID-locator map registration security. A solution to address these issues was proposed in [27], 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

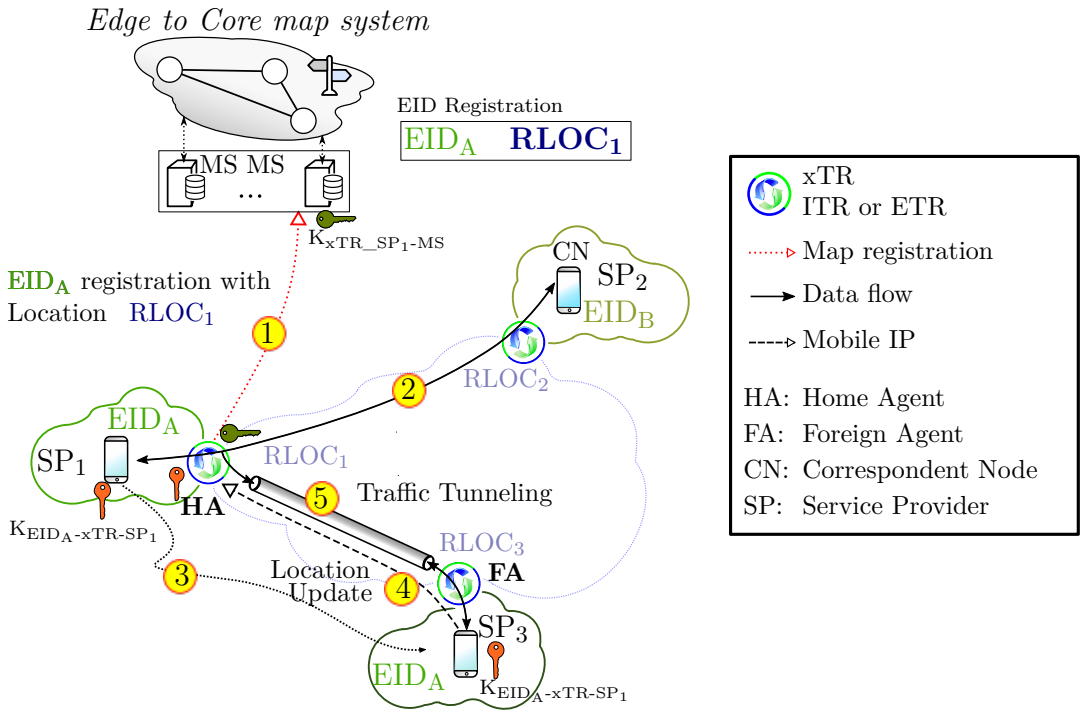


Fig. 5.2 Mobility scenario with current registration process.

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 exploring 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 session continuity. Figure 5.2 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.

5.3 Defense efforts for securing the LISP Control Plane

The on-demand discovery of EID locations poses the challenge on the ETRs regarding how to verify whether the reply from the Map-Server is valid, thus utilize the locator addresses provided. This process opens a variety of threats over the LISP control plane, which are being discussed and addressed by different proposals. Specifically, LISPSEC targets the security of the locator query between the Map-Resolver, the Map-Server and the ETR.

The registration and location update of an EID prefix is also crucial for the correct traffic encapsulation and routing within LISP. The initial shared-key approach currently defined enables the Map-Server to verify the proper authorization of an ETR to register

any provided locator address. This mechanism assures that the proper ETR is authorized to route and encapsulate the traffic of an EID-prefix site.

In this section, we shall further elaborate the previous efforts to secure the LISP control plane, and describe their shortcomings.

5.3.1 EID-to-RLOC Authoritative registration

The LISP Mapping System defines two types of LISP-speaking devices: the Map-Resolver, which accepts Map-Requests from an Ingress Tunnel Router (ITR) and resolves the EID-to-RLOC mapping by queering a global mapping database. Meanwhile, the Map-Server element [24] learns authoritative EID-to-RLOC mappings from an Egress Tunnel Router (ETR) and publishes them in the bindings database.

When a Map Server receives a Map-Request, from either an ITR or the Mapping System, it consults its local mapping database to find an ETR that can answer with the set of RLOCs for the requested EID-prefixes. To publish its EID-Prefixes, an ETR periodically register them along with the set of RLOCs that can be used to reach the ETR. This information is conveyed in a Map-Register message sent to the Map-Server.

To secure the mapping registration of authoritative EID-to-RLOC bindings, LISP has defined an authentication mechanism which includes the information of both EIDs, and RLOCs involved in the registration. This security data is derived from a hash over the Map-Reply message using a pair-wise shared key. The LISP specification recommends the use of HMAC-SHA-1-96 and as well as the HMAC-SHA-256-128 (SHA-256 truncated to 128 bits). This rather basic authentication mechanism has been defined as a first approach to secure the mapping registration process by involving the ETR and the Map-Server. However, it alienates the end user which may hinder the user mobility use case. Aligned with this issue, we have studied alternatives which allow the inclusion of the EID-holder in the map register process to enable EID authorization as well as to support end-point mobility without disrupting ongoing connections.

5.3.2 EID-to-RLOC Mapping Lookup security

Data-plane triggered mappings events occur when an ITR does not know the current locator of an EID-prefix. Thus, it initiates an EID-to-RLOC lookup process. The security and integrity of this process outcome highly affect the entire LISP routing security. As a first level of security, LISP [23] defines a mechanism in the ITRs to accept only solicited Map-Replies. To this end, the Map-Request message includes a nonce field

which objective is to authenticate the Map-Reply answer sent from either the respective ETR (or the allocated Map-Server in case of proxy configuration).

The goal of these security mechanisms has been to prevent unauthorized insertion of mapping data in a Map-Reply message by providing origin authentication and integrity protection for the Map-Registration messages exchange, and by using the nonce to detect unsolicited Map-Reply sent by off-path attackers. However, they fall short on securing the EID-to-locator query process.

LISP-SEC [55] provides a secure mechanism to validate the EID-to-locator query process. This DNS-like query allows the dynamic, on-demand discovery of any EID's locators through the assistance of the Map-Resolver, Map-Server and the ETR on charge. The first two elements act as a broker, enabling a complete decoupling between the control plane and the Mapping system. This has enabled the development of different mapping system in parallel with the LISP protocol evolution. The security of the mapping system is out of the scope of LISP-SEC and of this thesis.

LISP-SEC focuses on providing the required security over the Map-Request and Map-Reply procedure. It provides origin authentication, integrity and anti-replay protection to LISP's EID-to-locator mapping data conveyed via the mapping lookup process. LISP-SEC provides origin authentication, integrity and anti-replay protection to mapping data conveyed via the mapping lookup process by means of an Onet-Time Key (OTK). This security information is included in the Map-Request and Map-Reply messages. The final goal is to verify whether the corresponding ETR is on charge of the EID-prefix as well as to prevent the overclaiming of EID-prefixes.

5.4 Summary and Contributions

This chapter reviews some of the issues and vulnerabilities around the LISP control plane. The threats studied hinder the dynamic and secure registration of Endpoint Identifiers (EIDs) over the LISP Mapping Server. On one hand, the EID-to-RLOC registration process lacks of the means to verify the proper authorization to the claimed RLOC by an ETR. This vulnerability enables a rogue ETR to deceive its corresponding Map Server by providing spoofed RLOCs as its current location. As a result, this attack might end up with a traffic diversion toward the legitimate owner of the targeted RLOC, flooding the interface and provoking a DoS. On the other hand, the current EID registration process completely alienates the EID host role, hence making it dependent on the LISP-Site's ETRs. Therein, the Map Server is unable to verify whether an ETR

is authorized by the EID host to perform a map registration on its behalf. The current LISP specification does not include the EID host and EID owner as a decoupled element from the ETR. As a result, the mobility of end users between LISP sites is only possible with the help of other, cumbersome solutions like Mobile-IP.

We also review the current LISP specifications in terms of control plane security. More specifically, we describe the EID-to-RLOC authoritative registration and the EID-to-RLOC mapping lookup security. The former consists of a shared key between the ETR and its corresponding Map Server. This shared key enables the Map Server to authenticate and validate the claims made by an ETR regarding the new EIDs' locations.

Overall, this work has been included in the following publications:

- D. Montero, M. S. Siddiqui, R. Serral-Gracià, X. Masip-Bruin and M. Yannuzzi, "Securing the LISP map registration process," 2013 *IEEE Global Communications Conference (GLOBECOM)*, Atlanta, GA, 2013, pp. 2145-2151. DOI: 10.1109/GLOCOM.2013.6831392

Chapter 6

Offloading the LISP Map Registration

This chapter describes our proposal for enhancing the LISP control plane security aligned with the offloading approach. The initial LISP architecture design favors this approach as it decouples the control plane from the data plane. We leverage this advantage and propose new mechanisms to improve the map registration and enhance the end user mobility support.

6.1 Outline

First, section 6.2 delineates our proposal objectives which consider two key security challenges, i.e., RLOC verification and EID authorizations. Therein, we introduce the concepts of *EID Ownership*, *RLOC Authorization* and the different trust scenarios for our proposed solution. Next, section 6.2.2 describes in detail the proposed *Secured LISP Map Registration*. Finally, section 6.3 provides initial results that validate our proposal.

6.2 Map Registration proposal

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

6.2.1 Preliminaries

Before getting into the details, we proceed to define some terms that will help to explain our proposed 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 a 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.

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 Servers (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

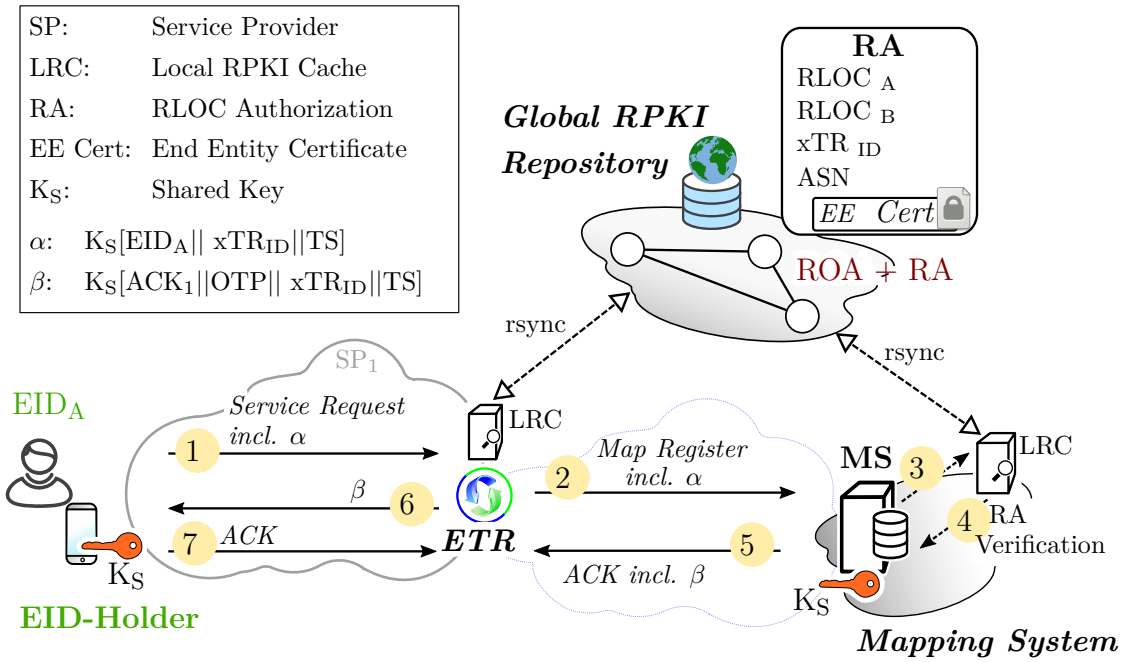


Fig. 6.1 Step-by-step overview of the new *Map Registration Process*.

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 6.1 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.

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 [50], 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 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. Notice that RPKI has already been implemented and deployed by ARIN [3] and RIPE [72], 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 [12]. 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.

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. We focus on this case since it is the worst possible scenario, which cover up the previous ones.

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

6.2.2 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. 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, we focus on untrusted scenarios, thus any party can potentially 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 captures 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 figure 6.1. 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. (6.1)) 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.

$$\alpha = \mathcal{K}_S\left(EID_a \| xTR_{ID} \| TS\right) \tag{6.1}$$

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 figure 6.1). 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 figure 6.1).

In the third stage (cf. steps 3 and 4 in figure 6.1), 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 (6.2), 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 figure 6.1).

$$\beta = \mathcal{K}_S\,(ACK \| OTP \| xTR_{ID} \| TS) \tag{6.2}$$

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 figure 6.1). 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.

6.3 Evaluation

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.

6.3.1 Testbed

The objectives of the experiments in the testbed are two-fold. 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 figure 6.2. It describes the network topology, the split between three different EID-Prefixes and a common RLOC address space. On top of this, a three nodes mapping system with one Map Server is depicted. The testbed is built using GNS3 [31], 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 [67] on the LISP-Site `EID-site3`. To complete the testbed we used the LISP-DDT Cisco IOS image [63] for the Mapping System 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.

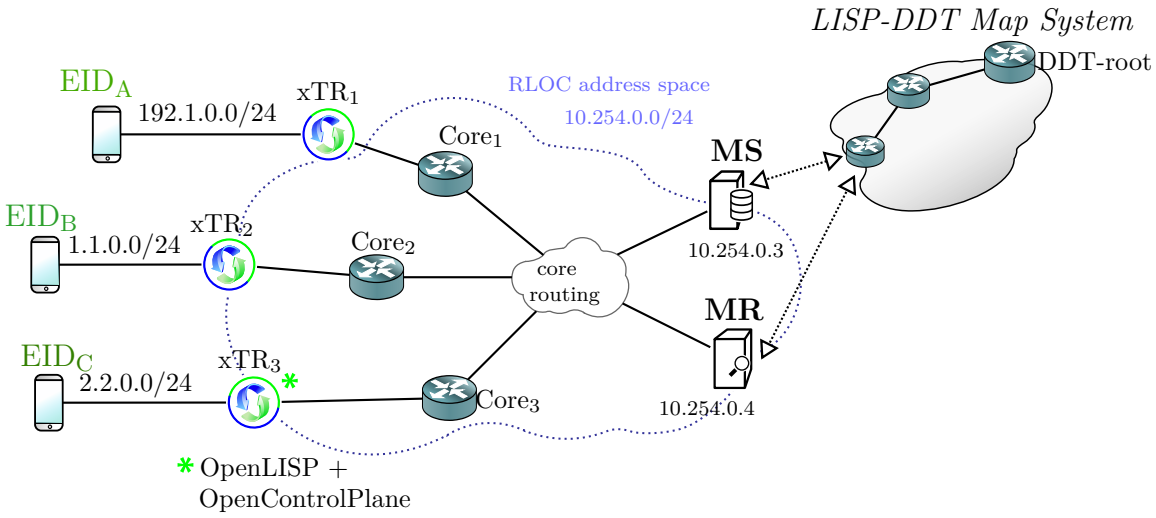


Fig. 6.2 Experimental LISP Tesbed with LISP-DDT Mapping System.

6.3.2 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 Authentication Data 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 basic 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 6.1, 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 6.2. 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.

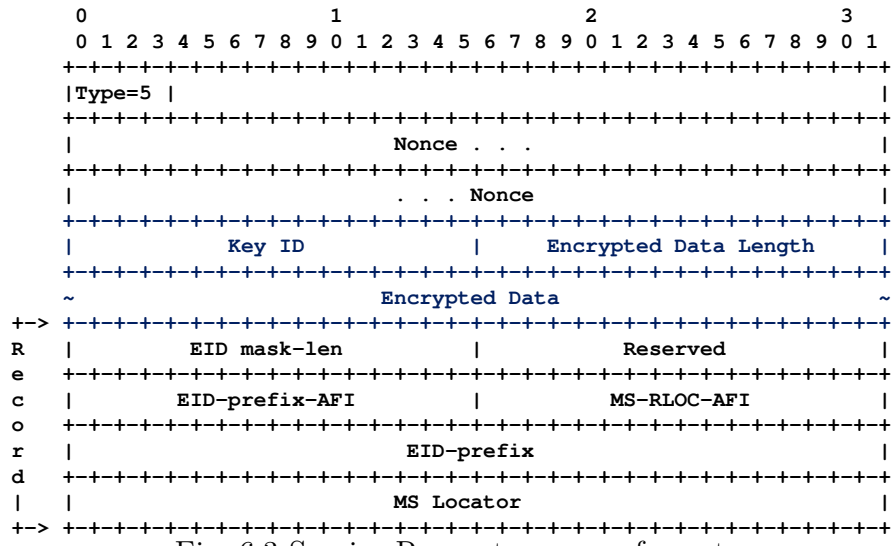


Fig. 6.3 Service-Request message format.

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. 6.1). This is a new LISP control-plane message consisting of the following information: [EID prefix $\|$ $RLOC_{MS}$ $\|$ α]. Figure 6.3 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 6.3.

6.3.3 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 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 6.1, which shows the overhead incurred for each

	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
Map Notify	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 6.1 New Map Registration Process Security Overhead

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. 6.3 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 6.1 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 figure 6.1). 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 same as the one presented for the *Map-Register* message in Table 6.1.

Followed by the *Map-Notify* message, the ETR forwards β towards the EID-Holder in message number 6 (cf. Fig. 6.1). The estimated size of this message, keeping in mind the format shown in Fig. 6.3, 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 6.1.

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.

6.3.4 Overhead over the Map Registration completion time

The time required for the Map Registration completion is another evaluation perspective that can provide insightful information regarding the caused overhead. To this end, we have implemented a proof of concept registration process which includes the whole registration procedure described in Section 6.2, and evaluated the time required to complete the registration considering different case scenarios.

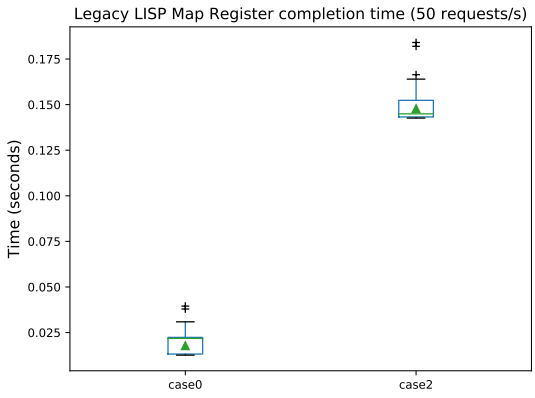
The first evaluation considers the current map registration process which entails two players, namely the xTR and the Map Server. Table 6.2 details the case scenarios. We have considered two different evaluation dimensions, (i) the number of requests per second, and (ii) the round trip time (rtt) between the xTR and the Map Server. The latter contemplates the scenarios whether the Map Server is under the same administration domain as the xTR. Furthermore, we have selected the AES-256 encryption algorithm.

	Case0	Case1	Case2	Case3
Execution Time	5 min	5 min	5 min	5 min
Avg. Requests per second	50	100	50	100
xTR to Map Server rtt	10ms	10ms	150ms	150ms

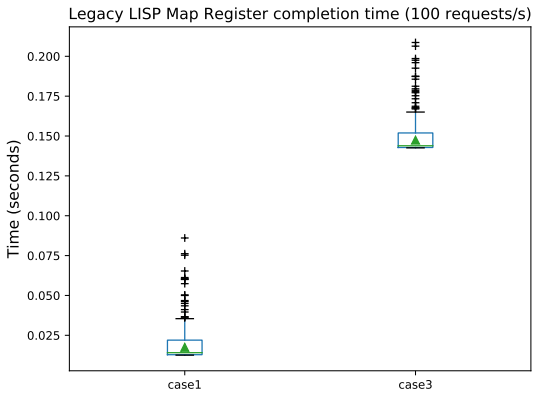
Table 6.2 Experiment cases for the current Map Registration evaluation.

These initial results reveal that the average time required to complete the registration is between $0.014s$ and $0.17s$ according the case scenario. Table 6.3 presents an extended summary of results per each case scenario described in Table 6.2 for the current registration process. Furthermore, figure 6.4 depicts a box-plot of the completion time results per each scenario. Therein, figure 6.4a shows the cases with 50 requests per second and two different round trip times among the players. Similarly, figure 6.4b depicts the cases with 100 request per second.

The second part of the evaluation considers our proposed end-to-end Map Registration process. As described in Section 6.2, our approach introduces a new player named the EID Holder, thus, the number of evaluation cases increases. Table 6.4 describes the different scenarios considered along with the dimensions for this evaluation, which include, (i) the number of requests per second, (ii) the rtt between the EID Holder and the xTR, and (iii) the rtt between the xTR and the Map Server. The different round trip times emulate the cases whether the EID Holder, xTR and Map Server are under the same



(a) *Case0*: 10ms xTR - Map Server rrt
Case2: 150ms xTR - Map Server rrt.



(b) *Case1*: 10ms xTR - Map Server rrt
Case3: 150ms xTR - Map Server rrt.

Fig. 6.4 Current LISP Map Registration completion time per case defined at Table 6.2.

	case0	case2	case1	case3
count	14481	14487	28114	28105
mean	0.017791	0.147783	0.017458	0.14742
std	0.004595	0.004592	0.004693	0.004684
min	0.01261	0.142628	0.012554	0.142582
25.00%	0.013212	0.143237	0.012834	0.142806
50.00%	0.021799	0.144952	0.014097	0.143895
75.00%	0.022353	0.152351	0.021968	0.151923
max	0.039508	0.184077	0.086041	0.208665

Table 6.3 Results summary of the LISP Map Registration completion time in seconds (Table 6.2 describes each case).

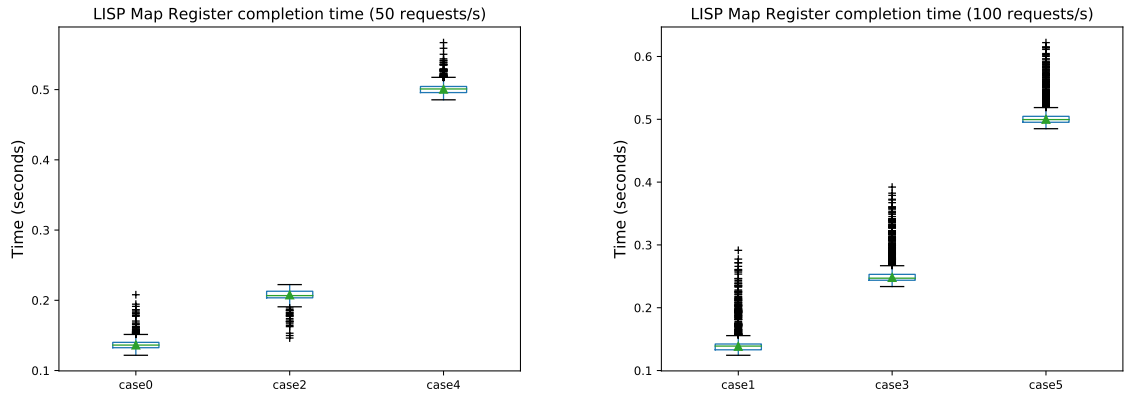
administration. We have selected the AES-256 encryption algorithm, and the RSA public cryptosystem, with a 2048 bits key private key.

The obtained results shows that in average the registration completion time varies among $0.13s$ and $0.5s$ according with the cases scenarios presented in Table 6.4. More detailed results are described in Table 6.5. Furthermore, figure 6.5 depicts two box-plots with the completion time results per each scenario. Therein, figure 6.5a shows the cases with 50 requests per second and three different round trip times among the players. Similarly, figure 6.5b depicts the cases with 100 request per second.

Overall, the initial results reveal insight regarding the overhead introduced in the whole registration completion time. First, the time for the registration is impacted by the network latency among the players (i.e. the EID-Holder, the xTR and the Map Server). This has shown that in the context of this evaluation, the overhead introduced by the security and encryption chores is lower than the network latency. For the legacy registration process, the average time is $0.017s$ and $0.147s$ for two different scenarios (i.e., network rtt of $10ms$ and $150ms$ between the xTR and the MS respectively).

	Case0	Case1	Case2	Case3	Case4	Case5
Execution Time	5 min	5 min	5 min	5 min	5 min	5 min
Avg. Request per second	50	100	50	100	50	100
EidHolder - xTR rtt	10ms	10ms	10ms	10ms	150ms	150ms
xTR - Map Server rtt	35ms	35ms	150ms	150ms	150ms	150ms

Table 6.4 Experiments cases for the New LISP Map Registration process



(a) *Case0*, *Case2* and *Case4* from Table 6.4. (b) *Case1*, *Case3* and *Case5* from Table 6.4.

Fig. 6.5 New LISP Map Registration completion time per each case defined in Table 6.4.

Similarly, the results for our proposal varies from $0.13s$ to $0.5s$ in cases where the network latency between the EID-Holder, the xTR and MS are different. The worst case considers rtt of $150ms$ between the three players. These experiments exclude the time required for RA verification.

On the other hand, the results obtained regarding the impact of concurrent registration requests over the registration time shows no major impact at the considered rates. We have tested two different rates of 50 and 100 requests per second in our tests. However, evaluations with higher rates are part of our future work.

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

	case0	**case2**	**case4**	**case1**	**case3**	**case5**
count	14540	14535	14522	28286	28322	28241
mean	0.136321	0.207677	0.50058	0.138596	0.248163	0.49998
std	0.006358	0.006592	0.006688	0.007866	0.008369	0.00827
min	0.121722	0.146254	0.485457	0.124312	0.233724	0.485022
25.00%	0.132493	0.203519	0.495773	0.133022	0.243762	0.495272
50.00%	0.136235	0.206685	0.500919	0.138953	0.247043	0.499547
75.00%	0.140125	0.212899	0.504468	0.142258	0.253155	0.504726
max	0.207922	0.222357	0.566902	0.291389	0.392101	0.622083

Table 6.5 Results summary of the new LISP Map Registration completion time (in seconds) for each case described in Table 6.4

our lightweight solution, this can be achieved without the complexities and extra burden of tunneling across protocols and mobile technologies.

6.4 Summary and Contributions

This chapter describes our approach for addressing the LISP control plane issues described in Chapter 5. We propose a novel and adaptable EID-to-RLOC registration process that works end-to-end and covers both EID and RLOC authorizations. The inclusion of the EID holder in the registration plays a key role in the enhancement of user Mobility. Our proposal enables an EID holder to dynamically register its current location with the RLOCs made available by the service provider. This results in a decoupling between the end user EID and her representing ETR. In other words, an EID holder is enabled to verify the RLOC proposed by the service with support of her Map Server (i.e., RLOC authorization), while the service provider in parallel verifies the proper EID authorization.

Our approach leverages on the design and infrastructure already developed by the IETF's Secure InterFomain Routing (SIDR) working group for resolving the RLOC authorization part, while presenting a potential adoption blueprint. The outcomes of our work have been published in:

- D. Montero, M. Yannuzzi, A. Shaw, L. Jacquin, A. Pastor, R. Serral-Gracià, A. Lioy, F. Risso, C. Basile, R. Sassu, M. Nemirovsky, F. Ciaccia, M. Georgiades, S. Charalambides, J. Kuusijarvi, F. Bosco. "Virtualized Security at the Network Edge: A User-centric Approach," in *IEEE Communications Magazine*, vol. 53, no. 4, pp. 176-186, April 2015. DOI: 10.1109/MCOM.2015.7081092

Part III: End-points and Users security vulnerabilities

Chapter 7

End Users security and the device-centric protection

The security of Internet end users and endpoints contemplates a wide range of threats starting at the moment they are connected and reachable. An end user is the victim of an attack when her device has been compromised in an attempt to steal information or gain access to it and further escalate the attack. Similarly, an attack can consists of the sharing of rogue information in deceitful form (e.g. a rogue attached file or link) which seems normal, but is actually effective once locally opened or executed. In this work, we consider the attacks that can be derived from a remote site through a network connection (e.g., the Internet), either by directly attempting to connect and remote control it or by sharing rogue information (e.g. a malware file or spam email). Attacks that require physical access to the devices, for instance a virus infection via an external USB memory, are out of the scope of this work.

This section describes a serious problem that concerns normal end-users as the number of Internet-enabled devices utilized to access the network increases, so does their security complexity. More devices result in more possible attack targets over a user. The security protection over the devices are usually specialized solutions customized accordingly with the devices characteristics, such as hardware architecture or operating system. This device-centric security paradigm focus more on securing the specific device, while ignoring the end user security over other devices. A lack of traversal security more focused on the user rather than in the specific device is raising. To elaborate, a user can have the protection software on her laptop, but no security solutions (or default protection provided by the operating system) on her smart-phone or tablet.

From a network perspective, the network security enforced by a provider represents a device-independent protection. However, this protection is usually not in a per-user basis

and differs between network providers. It focuses on the aggregated protection of security policies based on coarse, generic user-profiles. Consequently, this results in a hurdle for a user to obtain the same level of network security protection independently of the network attachment and location, as well as having customizable per-user security policies. Furthermore, user devices support multiple Internet connection technologies, which foster not only connection diversity but also mobility. The multi-connectivity of user devices adds up to the complexity of getting similar network protections, independently of the chosen networking connection and provider. The latter poses a major challenge as it considers the cases when different user's network connections are provided by either the same provider or different providers (e.g., a public WiFi access point and a different 3/4G provider.) Different network domains mean different rules and security protection policies, which might come along with the type of subscribed contract.

This problem is exacerbated more when we consider the Internet of Things IoT devices a user might own. These also require security protection when directly connected to a public network. In this case, on top of devices diversity, there are other drivers regarding their lack of security. More specifically, due to their constrained-resources nature, the toll for having local security protection may go in detriment with their final purpose. Thus, many IoT manufactures prioritize the main "thing" functionalities over security.

The rest of this section describes the problems delineated above and our approach to address them. In a nutshell, we propose a change of paradigm which consists of providing end-user security in an offloaded manner. We shall describe the goals of a user-centric security, and the vision of decoupling the security from the devices toward the network, with emphasis in the network edge. We discuss the suitability of our proposal and its alignment with current tendencies in the areas of Software Defined Network SDN, Network Function Virtualization and Fog Computing.

7.1 The Device-Centric security paradigm

The current device-centric protection model against security threats has serious limitations. This basically consists of installing a set of security applications on each device, such as anti-virus software and personal firewall. An average user nowadays has multiple terminals, including a smartphone, a smart TV, and a notebook, and in many cases also a tablet, a desktop computer and even a game console. These devices usually have different capabilities, architectures and operating systems (e.g. Android, MAC IOS, and Linux). Thus, the appropriate protection tools may not be available for all platforms. As a result, the most common practice is to install different security applications on

the various terminals, or simply rely on the default protection means provided by the operating systems.

Let us assume for a moment that users would like to have the same security policy and exactly the same protection level enforced on all of their devices. In the context of this thesis, we shall call this the "uniform security aim". To achieve this goal, the user would need to understand the configuration details of each device, which typically involves the setup of different security applications on different platforms. For non-technically savvy people, this turns out to be an impossible hurdle to overcome. As a result, most Internet users suffer from wide variations in their protection levels, and this problem is exacerbated as the number of devices per user grows.

We propose a user-centric model for security protection which is independent of the device utilized. This paradigm specifically addresses the two main drawbacks of the device-centric protection, i.e., (a) the need for dissimilar installations of security applications in different devices due to their different platforms, and (b) the problem of non-uniform protection to the user due to the difficulties in the configurations needed. This proposal is not devised as a competitor for the device-centric protection model; quite on the contrary, it is envisioned as an ally for improving the user and endpoints security. The contributions and results from this work were aligned with the *SECURED* research project [78].

To cope with the first problem, we propose a model in which the protection and security policies are now unified and remain homogeneous for each user, independent of the device used. This is achieved by means of a user-specific trusted virtual domain (TVD), which is dynamically instantiated at a secure place in the network edge. As we shall show, the TVD can be instantiated either on the user's side (e.g., on a home gateway) or on the provider's side (e.g., on a next-generation broadband access site handling the user's connections). This concept is aligned and can leverage the concepts of Network Function Virtualization (NFV), Edge Computing and Cloud Computing.

The second problem identified above represents a big challenge around of a user-defined security model with emphasis at ease of use by design. There is a main importance in how the protection policies are exposed to the average user in a high-level, alongside the necessity to enforce the configurations required transparently. This strategy detaches the definition of the protection policies from their corresponding configurations, thus allowing tailored protection even by non-technically savvy users.

Moreover, the envisioned model should support multiple actors who could simultaneously operate on the same traffic. Each of these actors may impose its potentially conflicting security policy. For instance, a user can decide the level of protection that

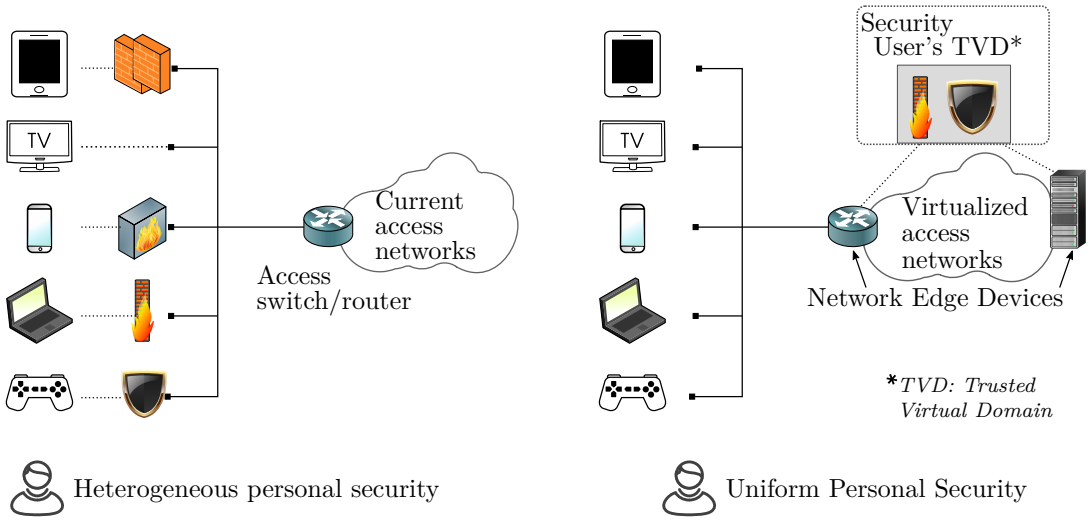


Fig. 7.1 User-centric model paradigm: (a) Current device-centric protection, (b)offloading security to the virtualized access network.

he needs, but the ISP may impose other limitations in order to guarantee the integrity of its network. In turn, the Government may impose additional restrictions. In case of conflict between the different policies in the hierarchy, our approach is to automatically resolve such anomalies, and inform the user about the issue and its outcome.

In order to resolve such conflicts, a "Reconciliation" [56] process is required. It takes the policies of the different actors that must be reconciled, and obtains a single set of policies to be enforced by the user's PSAs. The core of this process is the resolution of contradictions among rules from different policies. Priorities and hierarchies are some of the simplest forms to resolve contradictions (i.e., rules from higher priority policies/actors prevail), and they typically map well to contractual frameworks. However, custom reconciliation strategies can be defined. The user-friendly policies definitions and reconciliations are out of the scope of this work, and further information is detailed in [60] within the context of the *SECURED* project.

It is important to highlight that the offloaded virtualized security model here described can be applied both to residential and corporate scenarios. We describe its application in the form of a multi-tenant platform, considering the main stakeholders involved (i.e., service providers, infrastructure providers, security application developers, and users).

7.2 A different Protection Paradigm

Figure 7.1 depicts the basic concepts, showing the evolution from device-specific security to a common security framework for all devices hosted in the access network. In our model, security applications that are commonplace today (anti-virues, firewalls, content inspection tool, etc.) shall be called personal security applications (PSAs). Observe that under the current protection model, the heterogeneity of devices and platforms requires the installation of various PSAs with similar roles and functions; actually, four PSAs are required in the example shown in figure 7.1(a). Also, observer that some devices may remain completely unprotected, as in the case of some smart TVs.

Under our paradigm, the heterogeneous set of PSAs protecting the different devices is now moved and consolidated into a TVD. The TVD is positioned to handle and secure all the user's connections and exchanged information from and to the Internet. Each one only needs to host the minimum set of complementary PSAs required by the user (e.g., an anti-virus and a firewall in the example).

A TVD is a "logical container" that is instantiated *per user*, and is composed of the following elements:

- The execution environments hosting the user's PSAs.
- The required data, control and management plane interconnectivity in order to guarantee the isolation between different users' TVDs.

The virtual idea of a TVD enables its allocation and instantiation at either end of the access link, as shown in figure 7.1(b). Indeed, as a logical container, a TVD may run entirely within a single network edge device (NED), or in a distributed way involving several network devices. In our terminology, a NED is a device with virtualization capabilities that supports the instantiation of TVDs in a multi-tenant manner. If the TVD is placed in a user's premises, the NED could be either an enhanced home gateway or customer premises equipment (CPE). Those devices may need additional compute, storage and networking resources, and could be managed by the Internet Service Provider (ISP). If the TVD is placed in the ISP premises, as will be the case with the upcoming Network Functions Virtualization (NFV) based access networks [21], or similarly in an infrastructure with Fog Computing [93], a pool of nodes belonging to the provider infrastructure could be the NEDs devoted to host the users' TVDs. Note that this second deployment strategy leverages the virtualization and processing power of commodity hardware, and the unquestionable trend toward its ubiquite at the network edge—although it does not exclude the adoption of the first deployment strategy.

This model has a remarkable advantage over cloud-based protection [80]. Whereas in the latter case the virtualized resources supporting the users' security are rarely on the path that would naturally be followed by user traffic, in our model the TVD is always instantiated on-path. In other words, our model avoids routing detours, which would occur if the TVDs are located off the path between the user terminal and its traffic, e.g., in the cloud.

7.3 User-Centric Offloaded Security Architecture

This section introduces the envisioned architecture to support the offloading of users' security applications to their nearest compatible NED. The architecture is specifically devised to be heavily multi-tenanted and flexible enough to be used in scale-out systems. From a use case point of view, it can be expanded and deployed in a variety of ways, ranging from small set-top boxes or home gateways up to deployments on a much larger scale in a distributed environment (e.g., in localized data centers at the edge of ISP networks). Furthermore, it considers also the dynamic behavior of mobile users and our plan to support it.

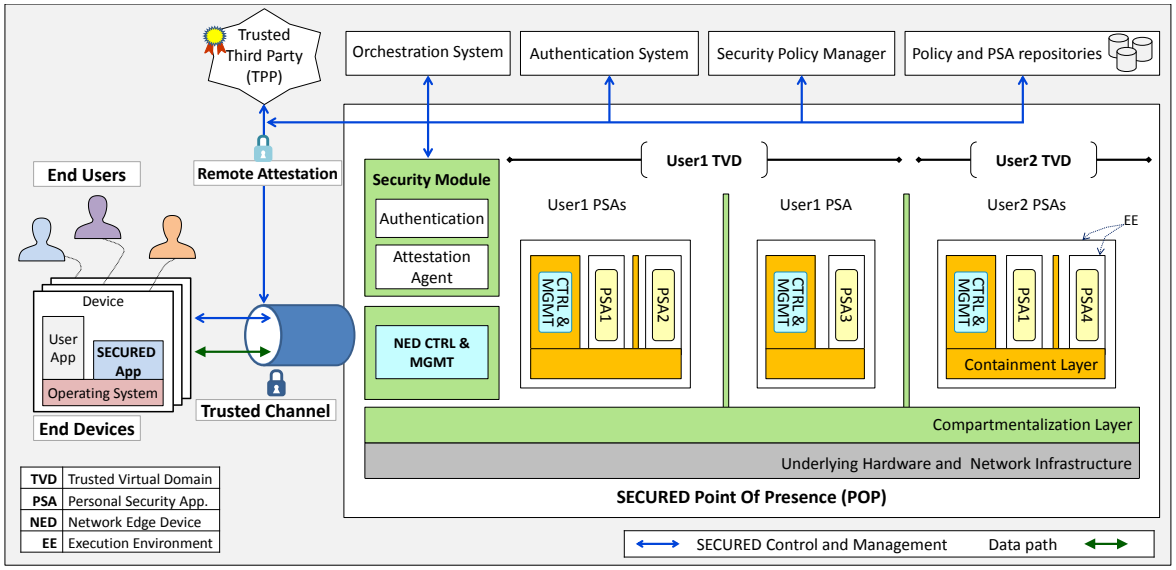


Fig. 7.2 The basic SECURED architecture showing a multi-tenant scheme on a Point of Presence (POP).

7.3.1 General Overview

The architecture must support the dynamic allocation and instantiation of users' security. The security functionality of each user can be comprised of different PSAs in a defined arrangement through service chaining, and these PSAs can be deployed within the same physical host or in a distributed manner. As a result, the architecture is required to support massive multi-tenancy, which implies isolation of users, their applications, and network traffic. A general view of the basic architecture is depicted in figure 7.2. The figure shows a generic deployment (e.g., on an NFV POP of an ISP). It is worth noting that, in simpler deployments (e.g., when the NED is a home gateway), the functionality provided by some of the systems on top of figure 7.2 could be simplified and embedded in the NED itself, or they could not be needed, such as the case of the NFV Orchestrator.

Overall, the TVD concept has been designed as an isolated environment which will hold the security applications of a user, and will in turn process the user's traffic. A TVD comprises one or more Execution Environments (EEs). An EE is a lightweight and heavily controlled environment that contains and executes one or more user PSAs—each one operating on the principle of least-privilege. Thus, in general two levels of isolation are defined: *i)* the *Compartmentalization Layer*, which is mainly responsible for the isolation between user TVDs; and *ii)* the *Containment Layer*, which handles isolation between PSAs within an EE. Thus, an EE could be either a *Compartment* or a *Containment* layer, respectively.

A derived requirement posed by multi-tenancy is network isolation. The proposed architecture must ensure the isolation of traffic amongst different users. More precisely, each tenant will be configured with a dedicated and private virtual network. This network connects the different PSAs with the end user on one side, and the Internet on the other side. Furthermore, the architecture defines a private management network that sets up, controls and manages the different TVDs. Both the Compartmentalization and Containment Layers have a *Control and Management* component, which aims to establish a separation between the technology independent part and the implementation-dependent technology. Likewise, this separation is in concordance and aligned with the proposed NFV architecture (further elaborated in section 7.3.5).

The second requirement is related to the establishment of trust between the end user's device and the corresponding TVD entrance point. This requirement is vital, since users would like to establish a certain level of trust prior to requesting the instantiation of security applications and sending their traffic. We address this requirement by leveraging two security concepts, i.e., the establishment of virtual secure tunnels and the concept of *Remote Attestation* (RA). The prior consists on the establishment of a secure channel

between the user's device and its access point to the TVD, while the lather focuses on trusted computing mechanisms to measure the system software upon component startup, where resulting measurement digests are held by a secure root of trust, e.g. a hardware device like a Trusted Platform Module (TPM) [95]. These measurements can be cryptographically signed by the device and sent to the users whenever they send an attestation request. The process of RA has proven to be a major challenge for SECURED project and is out of the scope of this work. Further information on the project's contribution in this regard can be found at [60].

7.3.2 Main Components

Security Module: This module is the front end, which is contacted during the connection establishment. It is comprised of two elements, the *Attestation Agent* and the *Authentication module*. Prior to authenticating, the end user first attempts to establish a secure connection with its nearest or designated NED. In parallel, a challenge request to perform an attestation of its software configuration is submitted. A mutually Trusted Third Party (TTP) system is involved in the attestation process. The TTP is responsible to keep a copy of known-good measurements, and provide a secure verification service to the user for verifying remote attestation responses. After a successful check, a secure channel is created and the user safely sends his credentials to the *Authentication module*.

Authentication System: The authentication of users is a key component of the proposed architecture. This can be implemented either using a local (standalone) authentication system, or relying on an existing external authentication infrastructure (e.g., an AAA+ system). The result of the authentication process is to obtain tokens allowing the interplay between the main components within a NED, and external subsystems, such as the PSA repositories. Once the user is authenticated, the instantiation of his security and the proper steering of her traffic must be enforced.

NED Control and Management: Once the user is authenticated, this module retrieves the user policies and metadata related to the composition of the required security applications. After that, the Control and Management module drives the instantiation of the user TVD, including its applications and the setup of the virtual network. More specifically, this module determines the resources required for the user TVD, and commands the instantiations required as well as the deployment and interconnection of the PSAs. This computation encompasses an analysis of the required compartments, containments and virtual networks to be allocated in order to instantiate the security applications. This analysis considers the PSA requirements along with the availability of resources, and the required configuration on the network (physical and virtual). In

addition, this module also manages the extension of the user data path to connect the user's device to the newly created TVD.

NFV Orchestration System: In an NFV POP, the NED Control and Management module is assisted by the NFV Orchestration system, However, in simpler scenarios, the former could entirely handle all the configurations required. In other words, when the NED is embodied in a home gateway of a residential user, the orchestrations needed will be handled locally without requiring and external orchestrator. In general terms, the NFV Orchestration system should deal with the instantiations and configurations in large distributed systems (e.g., an NFV POP), and preferably, in a "technology-agnostic" way. The "technology-dependent" part could be managed by the Control and Management module embedded in the NED.

Security Policy Manager: This module is in charge of handling the users' policies and the reconciliation process prior to performing the configuration of the user's PSAs. The reconciliation of policies are derived from a multi player policy definition, which might include policies defined by different entities, for example a goverment, service provider, or a father over his children. The policy reconciliation is a key novelty within the *SECURED* project and further details can be found at [60]

PSA Repositories: The applications are retrieved from these repositories with their respective manifest of capabilities. These are then further instantiated within an Execution Environment context part of a TVD.

End User App: This is the only application that needs to be installed in a user device. Its role is basically to support the secure communications with the NED, to handle the Remote Attestations and to signal and coordinate the user mobility.

Overall, the architecture introduced in this section allows the dynamic creation of trusted and virtualized execution environments throughout the access network. In this framework, several actors such as users, corporate ICT managers, infrastructure providers, security service providers, and software developers, can interplay and benefit from our user-centric protection model. An important remark about the proposed architecture is its alignment with the emerging NFV technology. It will be essential for guaranteeing its scalability. The following section details the usage of NFV as the execution framework for a distributed implementation of the proposed architecture.

The subsequent section introduces an example of a home user case scenario, which describes the interactions among the different components.

7.3.3 User-centric security initialization steps

This section describes a home user case scenario, which consists of a end user requesting the offloaded security service, and its instantiation at his provider edge network. We assume that the user previously defined his policies and required security application through a different channel. In a nutshell, the whole process consists of a user authentication, user profile fetch and PSA retrieval, and finally the instantiation and configuration of the security application.

Figure 7.3 illustrates the sequence diagram of a user requesting the offloaded security protection. This case we assume that the user's profile has already been created. The steps include:

1. **User authentication**: the NED authenticates the user using his SECURED credentials; upon successful authentication, the SM receives the session token back.
2. **Request TVD creation**: the SM requests the Orchestrator to instantiate a basic TVD for the user (and passes the session token).
3. **User profile query**: the orchestrator uses the session token to fetch the user profile (for the PSA related section).

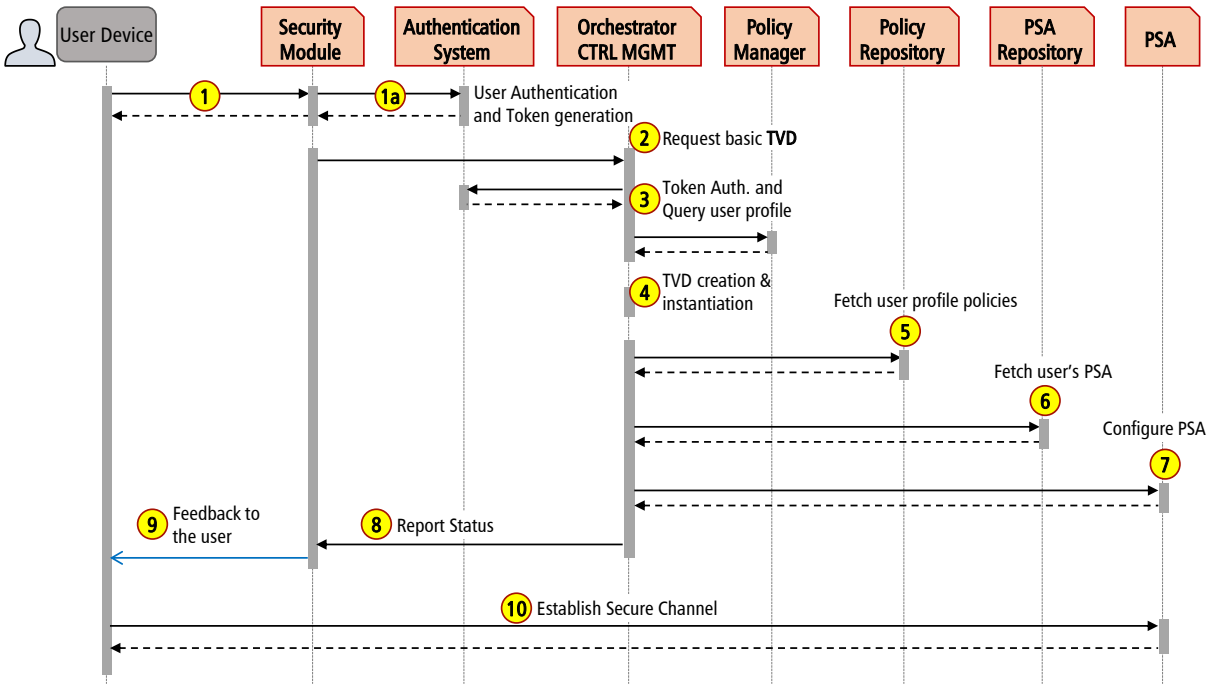


Fig. 7.3 Sequence diagram of an end user authentication and security protection instantiation.

4. **TVD creation**: the NED Management allocates and configures a new the TVD at the edge devices. This module validates the resources availability and properly assigns the PSAs across them. The simplest case considers one NED device at the user's premises.

5. **Fetch user's policies**: the Orchestrator fetches the policies from a Policy Repository. These policies are defined through a secondary channel by the users prior requesting the offloaded protection.

6. **Fetch user's PSAs**: The Orchestrator validates the user's PSAs local availability, and whether necessary fetch them from a PSA repository.

7. **PSA setup**: The Orchestrator sets up the TVD's network configuration and installs the PSAs.

8. **Secure datapath channel**: the secure channel from the user is extended to the PSAs. From this point on, all the user traffic pass through its PSAs.

The virtualization of the network and of the compartmentalization enforcement layer allows the SECURED architecture to easily scale–out to local data-centres, such as the case of NFV POPs of telecom operators in the access network. Observe that, independently of whether the user's TVD is confined to a single NED or it is distributed across the virtualized infrastructure, the functional view of SECURED for the user would remain exactly the same.

Overall, the architecture discussed in this section creates a trusted and virtualized execution environment at the access network, allowing different actors such as single users, corporate ICT managers, infrastructure providers, security service providers, and software developers to interplay, and benefit from the proposed paradigm shift in user protection.

The next section describe the reference implementations we considered in

7.3.4 Reference Implementation

This section describes in more detail core components of our proposed architecture from an implementation perspective. Therein, we include different technologies that could be leveraged to implement an embodiment of the envisioned architecture.

Trusted Virtual Domain TVD

The Trusted Virtual Domain represents a logical abstraction, agnostic of where it is instantiated. It comprises all the executing components for handling and securing a user traffic, including different execution environments and possibly multiple virtual switches. The latter are in charge of properly chaining the PSAs.

Execution Environment

An Execution Environment (EE) represents the framework where a PSA is executed. There are different options to implement an EE, which include:

- Full VM virtualization: a guest operating system running in a partition of the physical host. The hypervisor is in charge of enforcing the isolation of EEs. This option enables the execution of different PSAs that require the same OS.
- Lightweight virtualization: Container-based execution environment which shares the host OS. This type of execution environment restricts the PSAs OS options.
- Language-specific VMs: this option is a higher abstraction technology specific EE like the Java Virtual Machine (JVM). Its disadvantage stems from the software compatibility required giving the stringent language requirements.

The number and types of EEs is decided by the orchestrator according with the user's PSAs characteristics and available resources. The objective is to chain the PSAs in the least number of EEs possible by grouping multiple PSAs that share the same execution environment characteristics (e.g., all the application that run in linux). This is a non-trivial task, which is beyond the scope of this work. We refer the reader to [60] for more details.

PSA implementation categories

Our proposal envision different types of PSAs, which depends on their implementation characteristics and requirements. Among the possible options:

- Standard Executable: this PSA category encompasses any security application which is packaged as standard executable (e.g., an application running in a specific operating system). A single Execution Environment can host different PSAs.
- Full-fledged Virtual Machine: this type of PSA runs within its own execution environment without any compromise with other PSAs. Note that in this case, the

PSA is coupled with the proper management and control modules to enable the control, monitoring and configuration of the execution environment along with its PSA.

- Container-based: the PSA is packaged in a container image, while the execution environment is shared by the TVD.

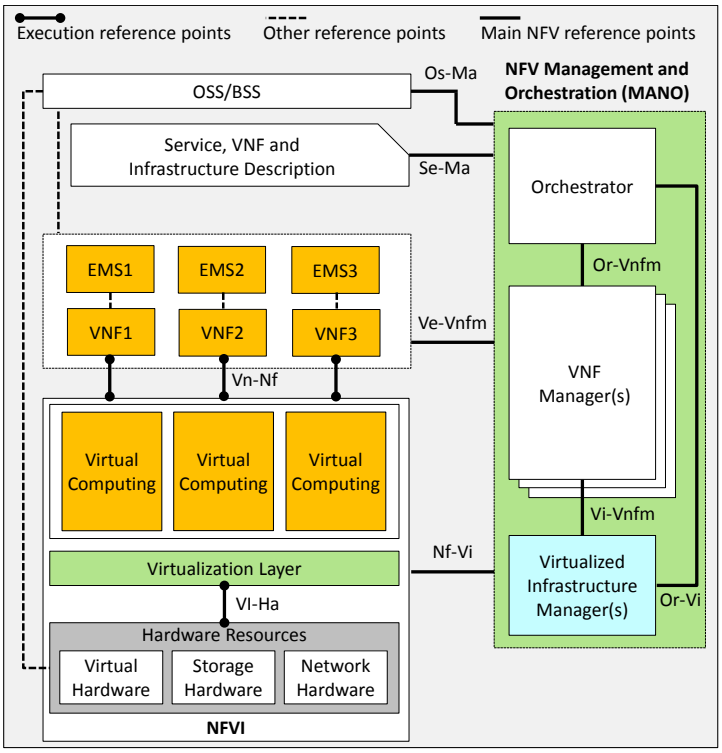
All in all, a prototype implementation of this architecture can be found in [79]. This implementation was the result of the joint work done during the SECURED project.

7.3.5 Mapping SECURED onto NFV

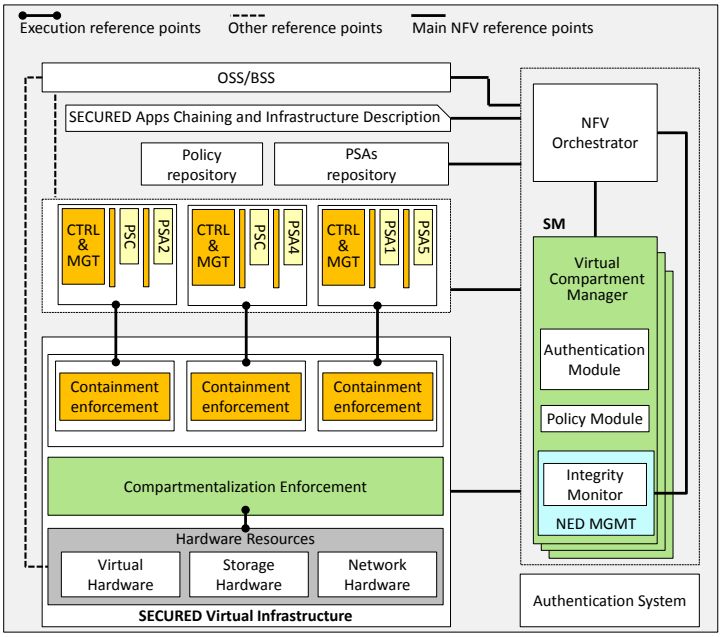
NFV is currently positioned as the main technology for building network functions by means of virtualization techniques. In NFV, network services are built by deploying and composing such functions on top of a virtualized infrastructure. Figure 7.4a depicts the general architectural framework defined by NFV. The complete description of each component can be found in [22].

The SECURED architecture, and in particular its most promising model based on the concept of a SECURED POP, is specifically tailored to be supported by NFV. In this way, network operators can leverage NFV plasticity and scalability when planning the provision of SECURED-based services. Figure 7.4b shows how the SECURED architecture maps to the NFV reference framework. Since the Operations and Business Support Systems (OSS/BSS) and the NFV Orchestrator are the general elements for configuring the global service operation and management of the network provider, they can act as general support mechanisms for SECURED-enabled services. The description repositories that the VNF architecture considers for the infrastructure and services will now contain SECURED-specific ones, as well as application repositories. The common NFV infrastructure is used through a virtualization layer that enforces compartmentalization and the containment of each VNF for a single user after appropriate attestation. These actions are managed by the interplay between the NFV Orchestrator and the NED CTRL & MGMT module.

Moreover, the PSAs are mapped into individual components of a VNF, matching the so-called VNFCs (VNF Components). These VNFCs are the deployment units considered by the NFV framework, which perfectly match the nature of PSAs. This is because PSAs are modular components of the security services offered to users, and they are dynamically instantiated according to the policies applicable to the particular user. Also, observe that the Element Management Systems (EMSs) map to the control and management functions of the Execution Environments in SECURED.



(a) The NFV Architecture.



(b) SECURED alignment with NFV.

Fig. 7.4 SECURED vision aligned with NFV.

Finally, the mapping of the Security Module to VNF Managers (VNFMs) becomes a natural consequence. The VNFMs are responsible of VNF lifecycle management and

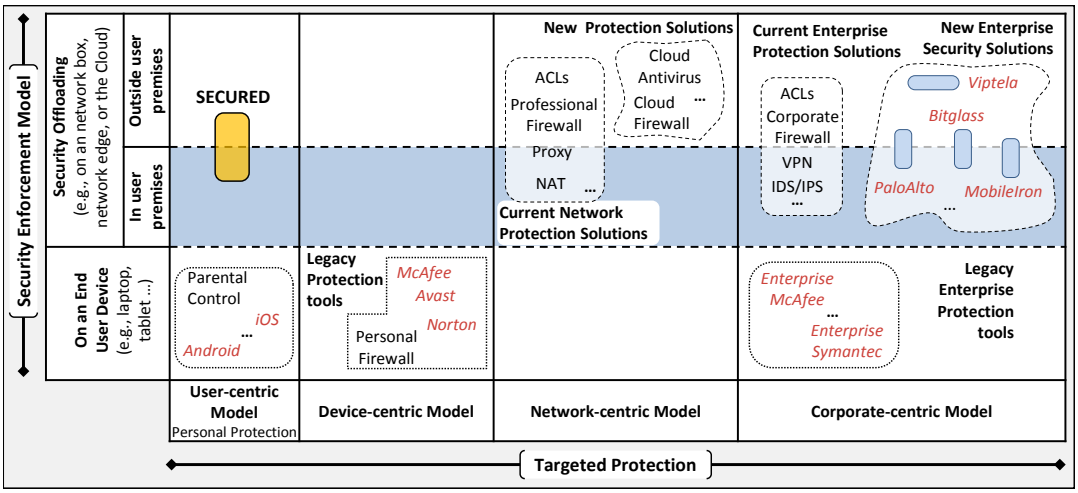


Fig. 7.5 Positioning SECURED considering some of the most common tools as well as some of the most recent and compelling solutions in the area.

can be specialized according to VNF goals and characteristics. Thus, they could be differentiated from other functions out of the SECURED scope, while still able to be integrated under the general operator orchestration mechanisms.

In summary, NFV represents a perfect deployment environment for SECURED, especially, considering that NFV will be implemented to scale up to several thousand nodes.

7.4 Positioning SECURED within the security panorama

The offloaded security model proposed in this thesis has several distinctive factors that make it unique. To show this, we position SECURED in the current spectrum of protection techniques, and highlight its main differences with state-of-the-art solutions.

The spectrum of solutions designed to counter security threats is really broad. The solutions available today can be reasonably categorized according to the table shown in figure 7.5. As it can be observed, there are solutions that are focused on protecting the end user device, while others propose different forms of security offloading. Moreover, current protection schemes can be classified based on whether they are user-centric, device-centric, network-centric, or corporate-centric. In a nutshell, figure 7.5 presents a high-level comparison of different security protection schemes according to two general criteria: i) the targeted protection model; and ii) where the security is enforced.

As far as our knowledge, SECURED is the only solution available nowadays that proposes a true user-centric model, which specifically addresses the need for device-

independent security. An important aspect to highlight is that, conversely to many of the offloading solutions available today, which are typically deployed in the Cloud, our solution admits a rich variety of deployments on either edge of an access link. Cloud-based solutions provide compelling protection schemes while avoiding several of the overheads for end-users (e.g., for corporate customers). The downside, however, is that: (a) they require routing detours; (b) they are not really user-centric (at least not yet); (c) they do not provide essential trust means, such as remote attestation; (d) they do not support advanced features such as policy reconciliation techniques; and e) total or partial support for user mobility.

7.5 Resource-constrained Devices (IoT) Security Protection

In the next few years, more and more "things" will produce and consume data in ways we are just starting to envision. Many of these "things" will be part of much larger systems, which will obviously require compute, network and storage capacity for processing their data and controlling their actuation. Similarly, these devices shall come enabled with different network connectivity technologies, which expose them to local-area or Internet-wide interactions. These interactions may or may not include a human, enabling new types of interactions like machine-to-human and machine-to-machine [35]. The security implications derived from the diversity of devices and their functionalities, the types of interactions, and the services they support remain as an open challenge.

The IoT security challenges stem from at least four dimensions: device diversity, number of nodes, resource constraints, and system exposure. Similar to the end user device-centric protection, the **IoT device diversity** poses a major challenge to provide a consistent and general way to manage the device security. This challenge is amplified as the **number of IoT devices is expected to increase exponentially**. Moreover, as in other areas, providing security and protections over such devices already **resource constrained** represents a big issue, as security is known to be a resource intensive feature. The trade-off between protection and efficiency will highly depend on how critical the device functionality is, and its available resources.

On top of all these issues, we believe that another major challenge derives from the **network connectivity exposure**, where, IP-enabled IoT devices are reachable from any other IP endpoint, extending the reach of the Internet to the physical world, alongside with its vulnerabilities. These kinds of devices support the interaction *cloud–*

thing (left part of figure 7.6). Meanwhile, other IoT devices with different network access technologies depend on a gateway or a hub (right par of figure 7.6). This external component enables a better security for the things, while also aggregating and processing the collected information prior to publish it to the cloud. At the same time, this approach promotes vertical, and closed IoT solutions.

7.5.1 IoT device protection

The nature of the elements in the Internet of Things (IoT) is expected to be quite elementary, which means that they will embed the computing, storage and network resources specifically provisioned to sustain their primal function. This design principle has opened a wide spectrum of new devices, ranging from streamlined temperature and air sensors, smart locks, to more processing powerful ones such as video cameras, smart-TVs, home-hubs, etc. As a result, an innovative environment for entrepreneurship has arisen, with emphasis on creating and providing new "smart" services based on the spreading of Internet-capable devices. However, in an attempt to be competitive IoT entrepreneurs and device manufactures seem to have sacrificed security in favor of price and time-to-market. This trade-off gets amplified as these devices usually lack of user-friendly security updates and mechanisms to apply patches, as well as they are meant to last for years; they are not replaced as often as our laptops or phones, thus users ignore their security flaws and consequences as long as they work as expected.

The common approach to secure the IoT devices seems to rely on the standard protections provided by the operating systems and network protocols supported. A more specific security protection for a specific device might hinder its functionality impacting its computing capacity or energy consumptions. The heterogeneity of devices along with their dissimilarities in hardware and software requirements contribute to the IoT security complexity.

For example, recent distributed denial-of-service attacks have demonstrated the vulnerability of Internet of Things (IoT) systems and devices at scale. These attacks have exploited the vulnerabilities of some IoT devices seeking to build large-scale botnets. A botnet is a network of infected devices or bots that has a command-and-control infrastructure, and is used for malicious activities such as distributed denial-of-service (DDoS) attacks. In September 2016, an IoT botnet built from the *Mirai* malware was responsible for a 600 Gbps attack targeting Brian Krebs's security blog (krebsonsecurity.com). The strategy behind *Mirai* is rather quite simple. It had used a list of 62 common default usernames and passwords to gain access primarily to home routers, network-enabled cameras, and digital video recorders [6].

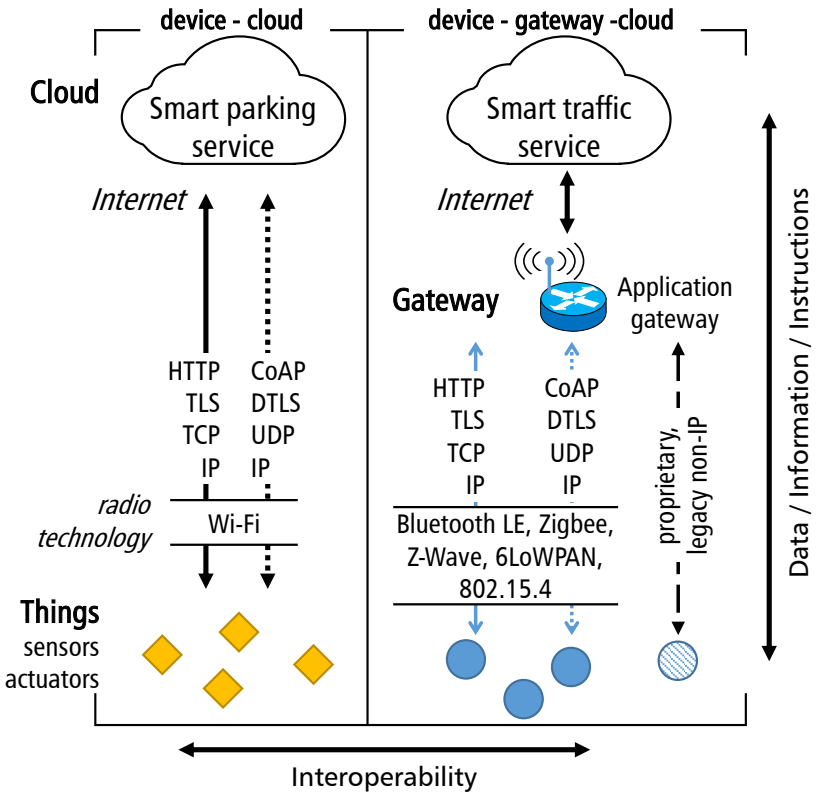


Fig. 7.6 IoT vertical solutions considering two communications patterns: (a)device-cloud, and (b) device-gateway-cloud.

The wide diversity of IoT devices in terms of their resources and capabilities influence different surface attacks. This diversity hinders the enforcement of security protections at the level of device. However, we shall keep in mind that many of these devices have specialized functions and predictable behaviors. Monitoring them in an aggregation point would help not only to detect anomalies which hint potential attacks or compromised devices, but also enhance the IoT security in general. However, the requirement of a specific gateway (and its hardware installation) per each device or set of devices induces to vertical solutions. We further delve into this issue in the next section.

IoT Vertical Solutions: The Silos Problem

IoT solutions have evolved from a closed, proprietary approach where initially the device's communication technologies and protocols were proprietary. The reasons behind were the specific devices' requirements regarding their constrained resources. Thus, the vendors' business model consisted on providing a whole vertical solution starting from the specialized "thing", its streamlined communication protocol, an in-between

application-gateway (or hub) up to a specific service at the cloud (figure 7.6) [28]. This approach fostered a silo problem, where for each IoT solution a complete new vertical is required. For example, a Smart City would end up with a really complex aggregated system of different verticals solutions, each one increasing the overall complexity.

The interoperability between IoT ecosystems remain an open problem, partially derived by their different architectures for communication. For example, the popular RESTful architecture in web and web-based applications has resulted in the IoT specific RESTful protocol stacks, i.e., CoAP. In contrast, device-to-device communication protocols and stacks like MQTT or DDS have existed before the popularization of IoT. This has resulted in IoT data models tightly coupled to the protocol stacks supported by the device. Thus, there is a lack of standard data models specifications regarding the semantic descriptions of objects and their capabilities.

In an effort to open the IoT solutions and boost standardization regarding the communications of the Things with local-area networks or the Internet, different standardization communities have worked on Internet-like standards and communication protocols. For instance, low-power wireless protocols options include Bluetooth Low Energy [10], Wi-Fi HaLow [91] (based on the IEEE 802.11ah specification [5]), Zigbee [96], Z-Wave [83] and 6LoWPAN [59] (all three based on the IEEE 802.15.4 specification [36]). These efforts' focus has been on enabling IP-based communications to the things, while providing interoperability at the network and communication levels.

On the other hand, the needs for building more complex and inter-operable systems have influenced initiatives upper in the stack (e.g., at the application or service level). For example, CoAP at the application layer, DTLS at the transport layer and CBOR data format. As a result, two communication patterns prone to create silos have emerged, as shown in figure 7.6, *device-cloud* and *device-gateway-cloud*. The former communication pattern is convenient for a widely deployed radio technology in a vendor's targeted market, such as Wi-Fi for smart home use cases. On the contrary, the latter communication pattern introduces a gateway that bridges the Internet with (a) a specific, optimized radio and network technologies, as well as (b) with legacy, non-IP based devices; an application-gateway might be necessary to intermediate, translating from one application-layer protocol to another one.

Similarly, the continuous growth of connected "things" projected for the coming years poses a big challenge regarding the increment of data to be processed and consumed. The pervasiveness of these elements implies a large scale, highly distributed set of deployed objects with limited processing, storage, power and connection capabilities. These challenges have risen important questions such as, (i) where should the computing and

storage be placed?, (ii) how will the plethora of things scale and inter-operate with their environment and the cloud?, and (iii) how and where to enforce the security protection of IoT devices. In [93], Fog computing is considered as the perfect ally along with Cloud computing to address the three initial questions. In this work, we leverage this architecture to propose an offloading security protection for the IoT devices.

7.5.2 Computing and protection toll over IoT devices

The resource-constrained nature of IoT devices pose the requirement of external means to perform resource-consuming tasks such as data processing or security protection. More specifically, the security protection enforced on the IoT devices highly depends on the trade-offs between its impact and costs compared with their main functionality and its criticality.

The demands for compute and storage resources will probably come from trillions of fixed and mobile endpoints, which will span vast geographical areas and will be arranged in various different forms, covering a plethora of different use cases and settings. Securing these devices will require scalable security solutions optimized for the IoT ecosystem. Following our proposed security offloading vision, we extend the user-centric protection paradigm into the IoT realm. To this end, we present a different approach to provide security protection to the "things" as well as to enhance the protection of the services that rely on or consume their generated information. Our proposal leverages the Fog Computing architecture with the purpose to extend and position dynamically these resource consuming tasks off the endpoints.

Similarly, the commoditization of networking and processing virtualization at the network edge provides some advantages and opens the possibility to better approach the verticals problem in the IoT realm. Fog Computing proposes a different network and processing architecture, which extends them along the cloud-thing continuum, with the common aim to locate their services closer to the endpoints. Therein, these virtualized services share a common communication bus with discoverable interfaces to exchange information.

7.6 Toward a new IoT Protection Paradigm

This section describes the service model proposal and architecture envisioned for the support of offloaded security within the IoT realm. Our objective for securing the IoT ecosystems follows a twofold approach, by extending the security protection off the

"things", while better protecting the infrastructure and its services from malicious things located at the network edge. Furthermore, the offloading paradigm enables the decoupling of not only security tasks, but also others that require or consume resources, toward close entities with better capabilities in terms of resources, scope and visibility.

Our vision leverages the Fog Computing architecture to provide a horizontal virtualization environment to support the offloaded security from the things. The security protection at the network edge enables the monitoring and detection of abnormal behaviors of things, as well as the processing of analytics over the devices' gathered information. Therein, our proposal targets to address two main IoT issues, i.e., (i) the homogeneous security protection of heterogeneous devices, and (ii) avoid the silos problem. To this end, we introduce two fundamental concepts and an architecture devised to support and implement them.

To cope with the first problem, we propose a model in which the security protection is unified and enforced in an enhanced device close to the "things". This is achieved by means of an "IoT Virtual Domain" (IoT-VD), which is dynamically configured and instantiated in the network edge. This virtual domain represents the execution environment hosting the IoT security applications, along with the proper configuration to both reach the required things, and other input services and information. Furthermore, each IoT-VD shall announce the new services and information it is able to support. The IoT-VD concept is similar to the Trusted Virtual Domains (TVD) described in section 7.2.

On the other hand, to break the silos problem and foster a horizontal intercommunication and interaction between tenants and ultimately things, we propose a model driven IOT-Service approach. This model represents the "things service intention" or the "what", and is referred as IoT-SM (IoT-Service Model). It defines which are the requirements in terms of input information and services, the new services and information it shall provide (the ouputs), policies to access and consume the services, and the IoT applications. The instantiation of an IoT-SM shall result in the creation of a IoT-VD, alongside the creation and announcement of new services and capabilities the tenant support.

In the remaining section, we shall further elaborate these ideas and the proposed architecture.

7.6.1 IoT Virtual Domain definition

An IoT-VD is a logical container that is instantiated on demand per a tenant and is composed of the following elements:

- The execution environments hosting the IoT applications
- Access to the intercommunication devices to reach the things (e.g., access to a WiFi or specialized network hardware)
- The required data, control and management plane interconnectivity in order to guarantee the isolation between different tenants.

The IoT-VD, as a logical container, is envisioned to run entirely within a single network edge device (referred as a Fog Node), or in a distributed way over a pod of Fog nodes. In our terminology, a Fog Node is a device with virtualization capabilities that supports the instantiation of IoT-VDs in a multi-tenant fashion. These devices are equipped with additional compute, storage and networking resources. The latter includes different networking access technologies to be able to communicate with the "things", such as BLE, Z-Wave, Zigbee or 6LoWPAN.

The proposed deployment strategy leverages the virtualization and processing power of commodity hardware, and the unquestionable trend toward its ubiquity at the network edge. Furthermore, it has a remarkable advantage over cloud-based IoT solutions, as the IoT-VD is instantiated closer to the devices, while also poses location and a broader scope for promptly processing and trimming the information as well as properly providing the services leveraged from the things.

7.6.2 IoT Model Driven service approach

To tackle the second issue regarding the silos, we introduce the model driven service idea. To illustrate, consider for example the case of a set of BLE noise sensors located close to a Fog Node. The IoT-VD definition allows the specification of the IoT applications to be instantiated along with the required interconnectivity (i.e., access to a BLE device). In parallel, there may be other tenants who are interested in getting the information regarding noise levels, but does not have the resources to deploy the sensors and collect and process the data. Thus, this poses the challenge to define a horizontal communication channel among IoT-VD in order to break the verticals problem. For example, one IoT-VD could publish the sensor information somewhere accessible to others IoT-VDs.

The IoT-Service Model (IoT-SM) comprises the declaration of the capabilities and services to be announced, as well as its requirements in terms of other services and input information. These models are meant to be machine-readable and can be interpreted and processed by an orchestration system. This orchestrator is in charge of translating the model to the actual instantiation at the network edge, alongside the corresponding

configuration on specific devices and exposing the services and information produced to both granted tenants and external users (e.g., to the Cloud).

Services are deployed by combining the IoT-SM, associated images for the IoT applications to be instantiated, configurations of the tenant network connectivity to both the "things" and the external world, message brokers and data flows, security policies, etc. This proposal enables the composability of services, which can leverage other IoT-VD services models to support and further improve their service, while also extending the services and information available. At this stage, we consider that the services and information are available among tenants instantiations around the same Fog Node location.

7.6.3 IoT-VD service-centric architecture

This section introduces the envisioned architecture and its position within the IoT realm. As explained above, this architecture provides a system where IoT tenants offload resource consuming applications off the things to their nearest compatible Fog Node. The architecture is specifically devised to be heavily multi-tenant and flexible enough to be used in scale-out systems. Our focus in this section is on the main architectural components.

General Overview

The architecture must support the dynamic allocation and instantiation of tenants IoT-VD services in a multi-tenant environment. Figure 7.7 depicts a generic deployment of three different IoT-VD services instantiated over a Fog Node. An IoT-VD was designed as an isolated environment that will execute the applications of a tenant. These are deployed in different Execution Environments (EE), which are connected through an isolated network. Each tenant will be configured with a dedicated and private virtual network. Furthermore, the underlying virtualization supports the allocation and assignment of specific network resources to communicate with the things.

Main Components

Node Control and Management — This module is in charge of coordinating the deployment of the IoT-VD based on the provided service model. The model defines the required applications as well as the requirements in terms of resources. Furthermore, it defines the types of services this IoT-VD instance will consume and the new ones it will support. This enables both the dynamic creation of services, APIs, and the registration

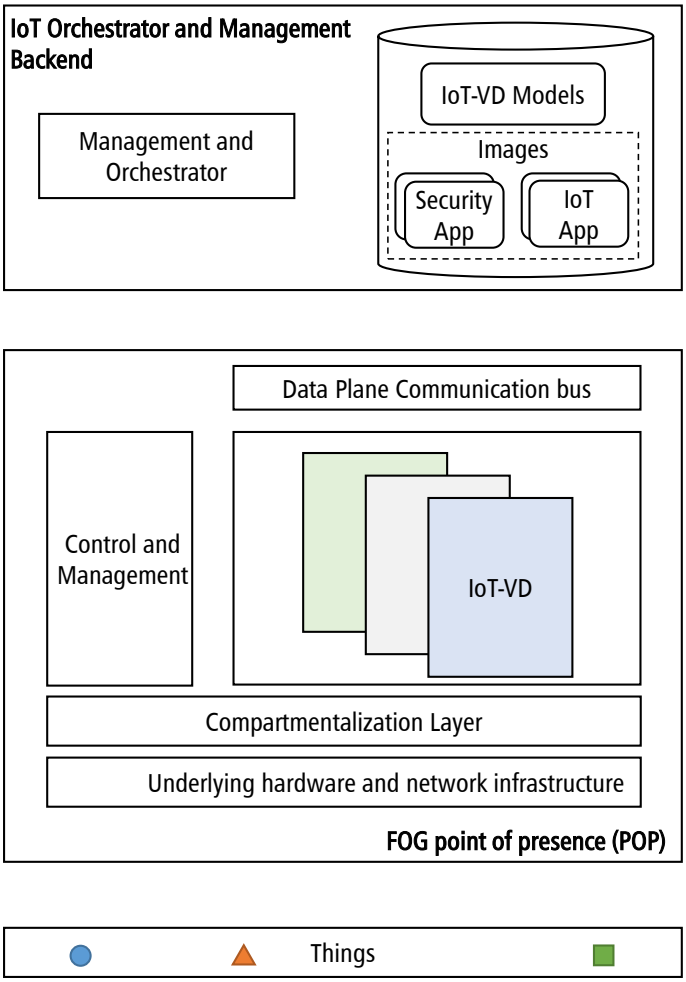


Fig. 7.7 IoT Model-driven Architecture

to the corresponding channels to publish its messages (e.g., following a publish/subscribe paradigm)

IoT-VD Service Models and Applications Repositories — The service models and the IoT applications are retrieved from these repositories, and instantiated on the specific execution environment.

Data Plane Communication Bus — This module represents the internal buses, brokers and data policy enforcement rules required to locally share and correlate data between IoT-VD of different tenants. This approach enables the interaction between tenants locally at the Fog node level.

Management and Orchestration System — This module deals with the instantiations and configurations in a "technology" agnostic way. The technology-dependent part could be managed by the control and management module embedded in the Fog Nodes.

This architecture is envisioned to provide the flexibility to offload the security protection from the "things" to a virtual gateway located at the network edge. Furthermore, we enable the interoperability among tenants by providing a shared communication channel.

7.7 Summary and Contributions

In this chapter we have described the "device-centric" security paradigm along with the issues and challenges it imposes over end users' protection. Therein, we claim that a different approach that shifts the paradigm toward a "user-centric" model seems appealing and necessary, as the number and diversity of end devices per user continues to increase. This proposed paradigm focuses on providing user protection independently of the device utilized to access the Internet. To this end, a new security architecture is proposed which leverages the power of computing and network virtualization as well as the new architectures for distributing and placing these resources in the network edge, closer to the users. More specifically, we contend that with architectures such as Edge-Computing, Fog Computing, Network Function Virtualization and Cloud Computing, the user-centric protection paradigm proposed is feasible.

A key feature of the proposed paradigm is its versatility to adapt and leverage all the underlying computing and networking architectures and services with the purpose of protecting end users in a uniform manner. Furthermore, the user-centric paradigm is not proposed as a replacement for the device-centric, but as an ally, extending and better securing the user and his devices.

We have deemed important to highlight our proposed architecture and its alignment with NFV. This stems from adaptability and scalability concerns. On the other hand, we also have positioned our proposal among current security protection solutions and remarked our novelty.

Similarly, we have discussed the security challenges around the IoT realm. IoT faces similar security issues as the end user device-centric protection, with the augmented complexity of removing the user from the interactions. The challenges stem from different dimensions which include their resource-constrain nature, device heterogeneity, fast growth and network exposure diversity. Then, we delineate our proposal to offload the protection from the IoT devices, and locate it at the network edge. This proposal follows

a similar approach as the user-centric protection paradigm. Finally, we describe our envisioned architecture along with the new concept of IoT-Service Model.

Part of the results and contributions obtained from this research are presented in the following publication:

- D. Montero, M. Yannuzzi, A. Shaw, L. Jacquin, A. Pastor, R. Serral-Gracià, A. Lioy, F. Risso, C. Basile, R. Sassu, M. Nemirovsky, F. Ciaccia, M. Georgiades, S. Charalambides, J. Kuusijarvi, F. Bosco. "Virtualized Security at the Network Edge: A User-centric Approach," in *IEEE Communications Magazine*, vol. 53, no. 4, pp. 176-186, April 2015. DOI: 10.1109/MCOM.2015.7081092

- Yannuzzi, M., van Lingen, F., Jain, A., Parellada, O. L., Flores, M. M., Carrera, D., Pérez, J. L., Montero, D., Chacin, P., Corsaro, A., and Olive, A. "A new era for cities with fog computing", in *IEEE Internet Computing*, vol. 21, no. 2, pp. 54–67, April 2017. DOI: 10.1109/MIC.2017.25
  Contribution: research, design, development and evaluation of a fog computing proof of concept. I contributed with the research process during the architecture design and definition. Therein, I collaborated with different researches in the development of the Fog Computing proof-of-concept orchestration and management system. On top of this, I contributed with the development of the use case *Event-Base video*.

Chapter 8

Offloading Personal Security Applications to the network edge: a Mobile User case scenario

This Chapter extends the user-centric security protection paradigm introduced in section 7.2, with emphasis in a mobile user case scenario. Our proposed model is based on the idea of offloading the security applications from the end user device, and placing them in a trusted network node at the network's edge. Our research perspective is particularly centered around three interrelated mobility challenges, i) the allocation of the security applications "close" to the user, i.e., on network nodes with enhanced processing capabilities, ii) seamless mobility with negligible disruption of ongoing network connections, and iii) dynamic orchestration and management with support of security applications migration. Based on our arguments, we expose the main requirements and trade-offs to be considered in the attempt to support mobility in such environment. Therein, we propose a flexible solution that leverages different key technologies such as Software Defined Networking, Network Function Virtualization and Computing at the Network Edge to offer a seamless on-path security protection to mobile users. Finally, our preliminary experiments' results considering a WiFi mobile user show that seamless security migration and mobility are feasible in a simple real scenario. Vertical mobility and more complex use cases scenarios are envisioned for future research.

8.1 Positioning a Mobile User in the Current Internet

User mobility in the Internet has become a common, not yet solved challenge. The proliferation of user devices enabled with different network access technologies have enabled the user to remain connected, even on the move. On the contrary, the Internet mobility support remains an open issue. However, this mobility trend has fostered the apparition of new applications and services that exploit the ubiquitous connectivity capacity of the devices. This has resulted in a whole new set of requirements in terms of service continuity and security protection.

In a world where a user can access the Internet via multiple devices, each equipped with one or more network access technologies, security protection has been a relevant and partially addressed issue. In light of this, end user security solutions traditionally have focused on the provision of security protection by installing software applications, e.g., firewalls and anti-viruses, on the users' devices. However, this device-centric security applications approach has arisen clear disadvantages, such as a per-device specific configuration and maintenance, or impact on the battery consumption. These issues could be alleviated by offloading the security applications to external parties, such as the cloud. However, this draws an important routing issue, i.e., the traffic must be redirected to the cloud facilities [80], which can cause an impact over the latency. To alleviate this, there are solutions that provide mechanisms to bring the required computational capabilities close to the edge of the network. Solutions such Fog Computing [93], or Cloudlet [75] provide an ecosystem where our solution may evolve on. Most importantly, they ultimately justify our approach, as it seems clear that in a short future, citizens will interact with their Smart Cities in a new communications' ecosystem that will allow the digital interaction with the city itself [77], thus both security protection and user mobility will play a key role.

Aligned with the smart cities' concept, mobile end users are always looking for the best Internet connection regardless of the network access technology used. In the context of this thesis, we define a mobile user case scenario as the case when a user's mobile device connection is switched as the result of the user's displacement. In this scenario, we approach the mobility challenge from the networking and routing perspective, i.e., mobility at the TCP/IP network layer. Thus, we consider only the cases where all the communications are IP based, regardless of the underlying connection technology (e.g., WiFi, WiMax or 4G).

From this perspective, there has been some efforts in the literature to support smooth handover between different IP networks. For example, in [54] the authors propose "LISPmob", a solution that leverages the LISP network protocol as a mobility enabler. Its main advantage is the decoupling of the Identity and Location of addresses at the mobile device level. This allows the end device to be unequivocally identified, independently of the attached network. Also, this solution is transparent to the applications. In this work, we leverage LISPMob as a mobility enabler for our solution prototype.

In contrast, to implement the *"close to the edge computation"* paradigm, a combined orchestration and management between the network and the computational nodes is required. Nowadays, this may be achieved by using two key technologies, i.e., Software Defined Networking (SDN) [45] and Network Function Virtualization (NFV) [21]. These technologies allow us to abstract the processing logic applied over a user's network traffic into generic functions, as well as dynamically steer it. In this work, we leverage them to provide a programmable and transparent network orchestration and management system.

8.2 Virtual Mobile Security Architecture

This section extends the envisioned architecture described in Section 7.2 to support offloaded security at the network edge in the environment of mobile users. First, the main building blocks and their tasks are described. Next, we dive into the essential mobility considerations for which this architecture is devised. Therein, we address the challenges imposed by a mobile user event with regard to switching connection points.

Figure 7.2 in Chapter 7 depicts the general proposed architecture for a user-centric security protection, including its main components, *i)* Trusted Virtual Domain, *ii)* NED Control and Management, *iii)* Resource Migrator, and *iv)* Orchestrator. As described before, the Trusted Virtual Domain (TVD) is the part of the system in charge of providing the framework for the security application (or chain of applications) as determined by the user or administrator. One important feature that must be present in this solution is the support for multi-tenancy. Each tenant (user) requires exclusive control and resources dedicated to its appliance. We provide support to this by leveraging the use of a virtual appliance, and then dynamically deploying as many instances as necessary. In fact, the virtual instances are created upon the end-user connection. A Network Edge Device (NED) element is the entry connection point which dynamically enforce a particular security policy configuration, in a per-user basis.

The *NED Control and Management* component is responsible for configuring the different network elements to comply with the necessary network traffic steering toward

the TVD. The basis of this component is that it instantiates a virtual private network for each user, providing a fully isolated data path from and toward the user, as described in section 7.3.

The *Resource Migrator* is a new architecture component in charge of synchronizing with the destination NED the state migration of a particular TVD upon request from the Orchestrator. As we shall describe later in this section, the Resource Migrator supports two modes of operation, full resource migration, and pure security applications' state migration.

One key component within our architecture is the *Orchestrator*. It is responsible of coordinating all the resource allocation, network configuration, and TVD migration. In this current proposal, the Orchestrator is a distributed entity present in each of the domains supporting the Virtual Mobile Security Architecture. The different roles of the Orchestrator include to provide the interfaces for dynamic configuration of the network, to monitor the state of the Mobile Device and orchestrate the handover process, and to manage the virtual resources' allocation and instantiation. The proposed architecture is specifically tailored to be supported by NFV. The aim is to leverage its scalability when planning the provision of virtual network functions.

With respect to the end user, we propose the establishment of a secure channel between the Mobile Device and the NED, as shown in figure 7.2. The purpose of this is two-fold. On one hand, it provides a more secure connectivity channel with the TVD where the user security is enforced, thus effectively avoiding potential man-in-the-middle attacks. On the other hand, by having a mechanism on the mobile device to inform the Orchestrator about the status of the surrounding network access points to coordinate an eventual handover, and consequently, the migration of the security and network connection state to the new, closest NED.

The establishment of such a channel is orchestrated by the NED itself and by an agent running on the mobile device. Particularly, the agent requests permissions from the *Orchestrator* to establish the channel, (for example an IPSec tunnel). In parallel, the Orchestration instantiates and enforces the user security applications. An advantage derived from this approach consists to securely validate the identity of the mobile device by the NED, thus providing a more robust and secure solution, yet keeping the setup simple for the end user.

8.2.1 Mobility considerations

This section introduces the challenges that the mobility of a user imposes over the envisioned offloaded security solution. These challenges are related with maintaining the

security state close to the user by migrating it along with him. Thus, the orchestration system requires to promptly react to the mobility event and handle it with the minimal impact over the user experience. This implies the reallocation, instantiation and security state migration toward to the new user Network Edge Device (NED) connection point. In parallel, there is also the requirement of reconfiguring the network as well as the handover process. The objective of the orchestration system is to maintain the user security the closest to the end user. Likewise, the second scenario considers the user mobility in a context where the user security does not require to be reallocated and migrated, it is maintained at the same point regardless of the user changing its connection point. Nevertheless, like in the previous case, a handover process as well as network reconfiguration orchestration is required.

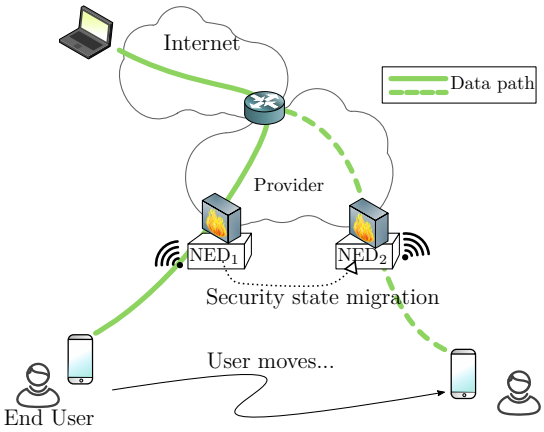
Figure 8.1 illustrates these scenarios. For both cases, we shall discuss the mobility requirements from the perspective of the orchestration system and the end user device. We shall address different trade-offs regarding how seamless the mobility could be with respect to ongoing connections as well as how the user security protection is moved or migrated along with the user in terms of maintaining the security state. In this work, we do not cover the legal and management issues that may arise in an inter-domain state migration case, where different network domains are involved, as we contend that from the technical perspective is a feasible solution. It is also important to highlight that the case scenarios consider the mobility of only one end of the communication, i.e., only the mobile user while the other end is fixed.

In the scenario depicted in figure 8.1a, prior the mobility event we assume that the user's security is already in place in the NED_1, i.e., the user previously connected to and requested the offloaded protection service. The orchestration module is in charge of managing the allocation, instantiation, configuration and deployment of the user security applications. Now, consider that the user moves away from the first connection point NED_1 toward a new connection point NED_2. This event triggers a joint coordination process between the end-device and the orchestration module. The expected result from the user perspective would be to have a smooth switch between the connection points. The smoothness of this transition imposes the following three requirements:

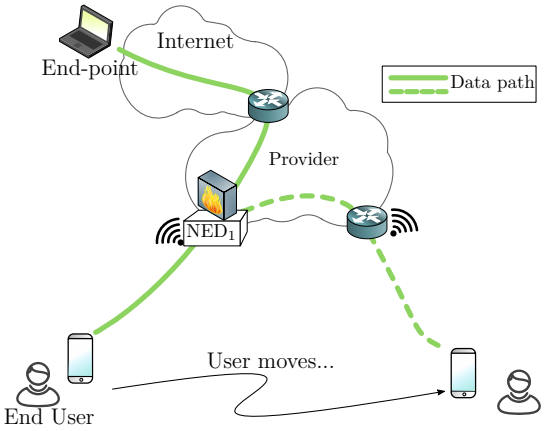
R.1 the user security state is correctly migrated,

R.2 the user is seamlessly disconnected and reconnected with negligible interruption of its ongoing connections,

R.3 the orchestration system is capable to react and allocate the security protection the closest at the destination connection point.



(a) Mobility Scenario: The user security state is migrated along with the user.



(b) Mobility Scenario: The user security state is maintained at the origin NED.

Fig. 8.1 End User Offloaded Security Mobility scenarios.

The second scenario represented depicted in figure 8.1b shows a similar case as the previous case with the difference that the user security applications are not required to be migrated. This is a simpler case due to it only considers the orchestration of the handover, proper steering of the traffic and the handling of ongoing connections, i.e., R.2 and R.3. However, the rerouting of the user's traffic through the TVD hosted at the initial connection NED is required.

It is important to remark that these requirements as well as the described mobility case are not intended to be exhaustive, but rather a starting point proposal to better understand and foster the discussion around the user mobility challenge in the context of this work. Next, we shall describe deeper the challenges that each of the requirements imposes.

8.2.2 Implications of an end user mobility

This section describes in a nutshell the implications derived from an end user mobility over the orchestration and connectivity management. Figure 8.2 depicts the sequence diagram detailing the main steps a mobility event comprises, as well as the different components and actors involved with respect to the mobility case scenario depicted in figure 8.1a. To facilitate the mobility decision process, the end user application periodically announces its available and accessible WiFi networks toward its current attached NED. This simple approach provides the NED with enough information to take the proper decisions. Whilst this approach serves to demonstrate the use case, it has drawbacks to be considered in the future, like the extra resources consumed by the user application due to the periodic query of available WiFi networks nearby.

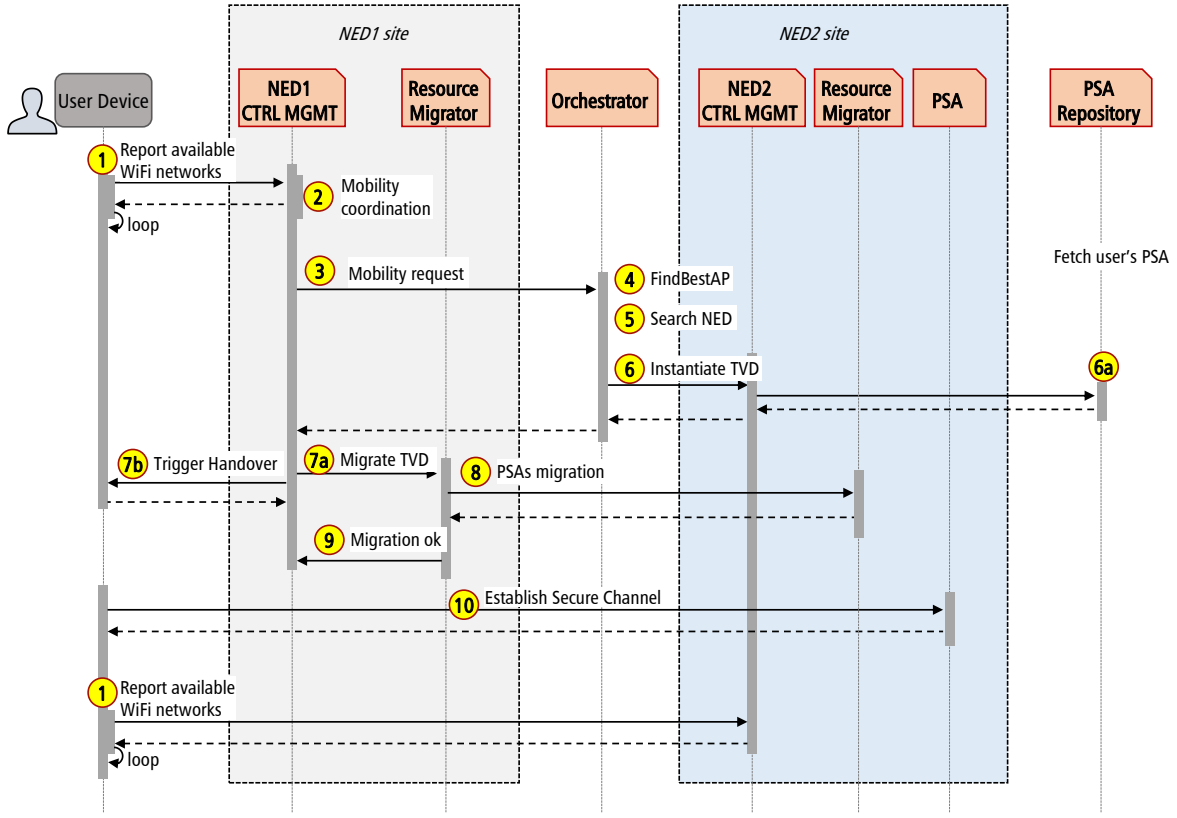


Fig. 8.2 Sequence diagram of an end user mobility migration. This diagram assumes that the end user is already authenticated at NED1, and his security protection has been properly instantiated. Also, the mobility use case involves an IP Layer 3 mobility.

The initial state considered prior the user mobility assumes that the end user has previously signed up to *NED1* and his security protection was properly instantiated. Then, the user starts moving away from *NED1* toward *NED2*. This event is captured by the former through the report sent by the endpoint, as the available WiFi signal strength changes. Once a threshold is reached, the mobility process is triggered. We proceed to elaborate these steps:

1. The end user SECURED application is periodically reporting the list of available WiFi networks, including their signal sthrength.
2. *NED1* is constantly verifying the available WiFi networks reported by the user. Once a criteria is reached, it decides to start a mobility process.
3. *NED1* triggers the Mobility request toward the Orchestration, including the end user credentials and the network list report.
4. *FindBestAP*: the orchestrator determines the best AP for mobility. This selection shall be next notified to the end user.
5. *Search NED*: based on the selected AP, the orchestrator determines the closer to the user NED. This selection takes into consideration not only the location, but also the available resources at destination; it is a multi-objective function.
6. The Orchestrator requests to the selected NED the pre-instantiation of the user TVD. This step also considers the fetching of missing PSAs into its local repository (step 4a in figure 8.2).
7. *NED1* triggers in parallel the TVD Migration and the connection handover. The latter is a crucial step as it involves a detachment and reattachment into a different NED.
8. Resource Migrator decides the best approach available to migrate the PSAs to the destination NED.
9. Once the migration is completed, the Resource Migrator notifies it to the *NED1*.
10. End user re-establishes the trusted channel with the PSA.

All in all, this simplified diagram captures the complexities and challenges behind the mobility use case. In the next section, we shall elaborate the concepts regarding TVD state migration and the end user connection handover.

8.3 TVD on the fly

The idea of removing the security protection from the end user's device leads to assume that the protection moves dynamically along with the user. Within the context of a static user, i.e. the user does not change of connection point, the instantiation and deployment of the security is handled by the orchestration system in a per-request manner. However, when the user starts moving and switching connection points while requesting the security protection service, it results in a "follow-close-to-me" protection service. The latter case can be addressed by promptly instantiating the user security at the closest new NED. The problem arises when some security applications require maintaining its state in order to function without disrupting the user experience. Thus, the system has to deal with the security state at the initial connection point and accordingly migrate it.

Within our solution, the user protection will be enforced by security applications and according to the application, there might be a state to be maintained while they are executed. These applications can be broadly classified in stateless and stateful. The stateless applications provide security independently whether they were previously being executed (e.g. a firewall). Thus, in the case of mobility there is no state required to be migrated. On the other hand, stateful applications initiate and maintain a state while they are being executed (e.g. anti-virus). As a result, in the event of a user movement, this state requires to be migrated where possible.

In order to address how to handle the latter case of applications, we foresee two approaches: 1) pure state migration, and 2) full resource migration. In the first case, all the burden of maintaining the application state is transferred toward the application developers. Thus, the application should provide the APIs to save and reload its state. Then, the orchestration coordinates the migration by properly invoking the application methods and transferring the state. For instance, an *iptables based firewall* [62] would only require the migration of the IP rules into the destination PSA.

In the second case, the migration of the user security is totally handled by the system, i.e. the security application is unaware of it. When a migration is requested, the orchestration system will have to deal with the stateful applications and dynamically migrate them. To this end, we also envision the implementation of the TVD through different techniques of virtualization, specially considering the support for resource migration, e.g. Virtual Machine or Container migration. This option introduces an extra overhead to the migration according to the size of the object to be moved. For instance, a stateful Web content filter application like Squid [84] maintains the state of active connections, which are migrated along with the whole execution environment with the

objective to avoid their disruption. Later in section 8.5, we present state-of-the-art technologies that can be leveraged to achieve this.

8.3.1 Virtual Resources Migration

The migration of the virtual resources is a broad topic, as it tightly depends on the used virtualization techniques, the type of virtual machines and the status of the network. At this stage, we consider the case of PSAs being VMs, and the migration supported by the hypervisor, e.g., KVM.

In this type of execution environment, we can find three different kinds of resource migration, *a)* Full VM Migration, *b)* Incremental VM Migration, *c)* Memory only VM Migration. Below we further elaborate each type.

Full VM Migration

When migrating a VM to a remote location, the disk representing the virtual machine needs to have an exact representation at the destination hypervisor. Full VM migration has been used for a while, e.g., in [15] the authors describe the basis and some optimizations for the migration. In our case, the necessary high-level steps to perform the migration are as follows:

- Transfer and create the VM definition at the destination NED.
- Transfer the disk (or disks) composing the VM.
- Incrementally transfer the used memory pages to the destination.
- When all the pages have been transferred, stop and destroy the origin VM and start the other at the destination.

The main issue with this alternative is that transferring the whole VM disk(s) to the destination may take a very long time, e.g., a VM with 20GB of used virtual disk and a connection of 100Mbps may take around 27 minutes, which is unpractical in most situations.

Incremental VM Migration

Opposed to the full VM Migration, another feasible alternative is to leverage on the fact that the PSA's virtual disk are the same for all the different NED. To this end, the

system can deploy the set of base PSAs that may be instantiated by the users demands. Then, the instantiated PSA derives a new virtual disk from the base PSA, incrementally writing the changes to that disk and leaving the base disk as a read-only resource to be used when the derived disk does not have the information. With this neat solution it is much easier to migrate the VM, now the steps are:

- Transfer and create the derived VM definition to the destination NED.
- Transfer the incremental disk (or disks) changes in the VM.
- Incrementally transfer the used memory pages to the destination.
- When all the pages have been transferred, stop and destroy the origin VM and start the other at the destination.

This new solution provides a little configuration overhead but gives a very large advantage as now the amount of data to transfer over the network is greatly reduced to several seconds in the most common case. The only issue of this solution is that the incremental disk changes, many times refer to logs, or cache files which may be discarded in a real scenario as they are not mandatory at the destination. Then, to further optimize this method there is also the Memory only VM Migration.

Memory only VM Migration

This option builds upon the concept of a base VM disk which is configured in read-only mode. The execution environment creates a *whole* filesystem as a read-only blob, and all the necessary write operations are performed on ram-drives. Using ram-drives for this purpose forces a slight increase—application dependent—on the VM but permits to avoid the transmission of the disks to the destination. Then the steps to perform the VM migration now are:

- Transfer and create the derived VM definition to the destination.
- Incrementally transfer the used memory pages to the destination.
- When all the pages have been transferred, stop and destroy the origin VM and start the other at the destination.

The main issue of this solution is that it imposes a great burden on the configuration of the VM, mainly because creating a read-only filesystem requires some specially crafted configurations that make it harder to deploy and create a new PSA.

8.3.2 TVD Migration

The TVD Migration process consists of dynamically migrating the user PSAs along with the TVD configuration into a new NED site. Algorithm 2 describes the migration scheduling logic, which considers both the PSAs and the TVD configuration.

Algorithm 2 PSA Migration Scheduling logic

Input: NED Status.
$\mathcal{P} = \{P_0, P_1, \ldots, P_i\}$: List of PSAs
$\mathcal{N}$: Destination NED
$\mathcal{T}$: TVD Configuration

Output: **True**: if the migration was successful.
False: Otherwise

for all $p \in \mathcal{P}$ **do**
  if $\exists p$ in $\mathcal{N}$ **then**
    $mode \leftarrow Incremental$
  else
    $mode \leftarrow Full$
  end if
  $status = migrate(p, \mathcal{N}, mode)$
  if $!status$ **then**
    return $status$
  end if
end for
return $migrateConfiguration(\mathcal{T}, \mathcal{N})$

The algorimth first checks if the PSA virtual image is locally available. In case the image is not available, a Full VM migration is required. Once all the PSAs are properly moved, then the TVD configuration is migrated.

8.4 Orchestrating the End User Mobility

A very important challenge for the orchestration system in a mobile user scenario is twofold. In parallel, it has to decide when and where it has to allocate, instantiate, configure and migrate the security protection of a user in movement, while continuously coordinating with the end user device the actual connection handover. The continuity of the service depends on the correct outcome of both tasks. In this work, we assume that

every time a connection handover is performed, the device gets IP connectivity and all the ongoing connections will be route accordingly to the new location of the user.

However, this connection switch may imply changing the access connection technology in the case of a vertical handover (e.g., a switch from WiFi to 3G), or a horizontal handover when the same access technology is used. In the latter there may be a short connection interruption when the access technology does not support the simultaneous connection to two access points (e.g., WiFi simultaneous connections to two access points). In this work, we consider the second case as a proof-of-concept, and more complex cases including the vertical handover is considered for future research.

Therein, the handover coordination challenge can be addressed either in a proactive or reactive manner. The former takes into consideration an active interaction between the orchestration system and the end user device via either polling the device from the orchestrator, or the device pushing information related with mobility. Thus, the orchestrator preemptively can allocate, instantiate and migrate the user security prior triggering the handover.

On the other hand, the reactive approach considers the scenario when the end user devices unilaterally decides to handover and the orchestration system has to react to this event. This case can also consider the exchange of mobility information between the parties, but as the end device is the one which takes the decision, the orchestration system has to predict or get informed when the user has disconnected, and where it potentially will reconnect prior triggering the reallocation, instantiation and migration of the user security.

Aligned with this, Algorithm 3 describes our approach for handling a WiFi proactive handover coordination. Therein, it evaluates the signal levels of the current access points against a threshold δ, which serves as a simple criteria for triggering the handover. The subsequent steps include:

- *FindBestAP($\mathcal{L}$)*: from the list reported by the user, the Orchestrator seeks for an available AP. In this embodiment, we have a list of predefined SSIDs. In case there is non AP available, the handover process finishes.

- *SearchNED(newAP)*: The orchestrator determines the best NED according with the new selected AP. The objective is to locate the closest NED to the end user and visible to the AP. In this embodiment, we have assigned to each AP a specific NED.

Algorithm 3 Seamless WiFi Handover triggering and orchestration process

Input: $\mathcal{L}$: List with the surrounding wireless signal levels of the mobile device

Output: [**True**, **AP**] if handover necessary
False otherwise.

1: $\mathcal{C} \leftarrow$ AssociatedAP
2: **if** Signal($\mathcal{C}$) $\leq \delta$ **then**
3: newAP $\leftarrow$ FindBestAP($\mathcal{L}$)
4: newNED $\leftarrow$ SearchNED(newAP)
5: InstantiateTVD(newNED)
6: $MigrateTVD$(newNED)
7: **return** [**true**, newAP]
8: **else**
9: **return false**
10: **end if**

- *InstantiateTVD(newNED)*: The Orchestrator instructs the newNED to allocate and pre-configure the new TVD. This also includes the fetching of PSAs to local repositories.
- *MigrateTVD(newNED)*: This function triggers the PSAs migration between NEDs. In parallel, the return value contains the newAP to connect.

It is important to note that the migration is triggered in such a way that at the same time the handover is executed. The objective is to minimize connection interruption from the end user perspective. The biggest challenge is to maintain the active TCP connections even after the mobility event.

This rather intuitive and simple procedure is our initial attempt for handling the handover from the orchestrator perspective. There is plenty room for future research in topics related with this process, including for example defining the best optimization for selecting the appropriate NED and AP toward which the user device should handover. Likewise, defining the intelligence behind the Orchestrator regarding how to select the node for the reallocation of the security. In Section 8.5, we shall describe an experimental prototype employing this proactive coordination and the initial results considering the service interruption time induced by the mobility.

8.5 Experimental Setup and Results

In order to analyze the capabilities and limitations of state-of-the-art technologies for end user mobility support within the context of offloaded security, we have run a set of experiments following the scenario illustrated in figure 8.1. In this section, first we shall describe the setup and the technologies integrated. Then, the insights from the experimental results are reported.

8.5.1 Experiment's objectives

The objectives of the experiment include:

- to evaluate the impact of the user mobility over ongoing connections while being protected by a firewall and content filter.
- to assess the time overhead incurred by the TVD migration.

The case scenario to test considers an end user watching a video while moving between WiFi APs. Also, the user is generating ICMP messages every 50ms to a remote location. The former experiment serves to evaluate the disruptiveness of our proposal over ongoing TCP connections, while the latter will help us to asses the WiFi handover delay. Therein, we shall also consider the overhead induced by the network management (i.e., the dynamic assignment of IP and the LISP registration process). Our scenario consider mobility at the IP level, thus the end user keeps its IP address while switches APs.

8.5.2 Testbed Setup

The testbed setup targets the WiFi mobile user case along with one offloaded security application per user. The application is stateful and the migration is totally handled by our solution. Our objective is to test how disruptive the mobility event may be over the user's ongoing network connections, considering the overhead introduced by the resource migration and the handover.

In a nutshell, the testbed is composed by a user device, two access points at different locations and controlled by their corresponding NEDs, and an Orchestrator. The technologies integrated in the NED, Orchestrator and the user device are detailed next.

The NED component is where the users applications will be executed. Thus, there is the need for multi-tenancy support, i.e., isolation of execution environments and network programmability, and network mobility support. **Multi-tenancy** allows the

secure coexistence of multiple tenants in the same physical infrastructure. We address this by leveraging the use of Virtualization. There are two main options to consider: Virtual Machine (VM) virtualization and Lightweight Container-based virtualization. One key requirement in our proposal is the support of resources migration. Thus, VM virtualization using Qemu-KVM was selected. This solution provides support for live VM migration, enabling the migration of the security application. Nevertheless, migration support for lightweight virtualization technologies such as Docker Containers are work in progress, and shall be considered in the future.

Along with the virtualized computation technology, there is the need of a **networking virtualization** counterpart. Network virtualization is necessary to provide extensive network reconfiguration, unparalleled flexibility, and multi-tenancy. We chose OpenvSwitch as the technology to virtualize the network in the NED. This technology provides us the flexibility to dynamically steer the user traffic when a user moves.

Similarly, mobility support in an IP network imposes the challenge to allow the end user to switch networks and maintain its ongoing connections, i.e., all the traffic for the user is correctly routed to the new access connection point. To address this challenge, we consider the Locator/Identifier Separation Protocol (LISP) [23] technology along with *LISPmob* [54]. *LISPmob* is a mobile implementation of the LISP protocol which enables the mobile device to switch IP network connections points without disrupting the ongoing network connections. Along with it, the LISP control plane supports rapid changes in the routing locators to achieve fast handovers with reduced service disruptions in terms of efficient traffic routing.

Finally, for the **Orchestration module**, we consider the OpenStack project as a key technology for managing and orchestrating the infrastructure|mainly because it has support for SDN and is an enabler for NFV. The Network Functions Virtualization (NFV) can be built on top of it to create a network that becomes a powerful and extensible API, on which applications perform actions and operations by invoking virtualized functions directly on the network. For our setup, we have simplified this component by implementing a basic Orchestrator that directly manages the VM instantiation and migrations through libvirt and KVM, and the network configuration through an OpenDaylight SDN controller, leaving the adoption of OpenStack as an important part of our future work. Furthermore, our ongoing developed orchestrator is configurable through REST interfaces that can be automatically invoked by the network upon the connection of a new user, or in the event of a moving user. This capability enables our solution to be easily integrated in larger orchestration infrastructures.

8.5.3 Experimental results

The goal of the experiments is to study the feasibility of maintaining operative the user protection upon WiFi mobility considering two cases: with and without involving migration of the user security TVD. The disadvantage of the latter is that it requires traffic detours from the new location to the NED where the user's security application remains. On the other hand, the former avoids the traffic detour at the cost of requiring the migration of the TVD.

As discussed above, the overall Mobility process consists of two related processes, i) TVD migration, and ii) connection handover. Figure 8.3 depicts the involved stages considering both. When the Orchestrator decides to execute the mobility process, it initiates first the TVD migration. This proactive approach allows to reduce the connection disruption time, as the PSAs remain active while being migrated. By the time the migration progres reaches a $\Delta\%$, the Orchestrator notifies to the end user to execute the handover. Once the end user device reconnects to the destination NED, it requests an IP address through DHCP. Then, the end user LISPmob component registers its new location into the mapping system.

Next, we evaluate the effects of technologies such as LISP (e.g., the refreshing delay for updating the routing locators in the mapping system upon handovers), in conjunction with DHCP (when routing locators are assigned dynamically), in order to understand the lower bounds achievable for seamless mobility considering the technologies previously covered in this section. To this end, we setup a testbed using the following elements: i) an OpenDaylight LISP Flow Mapping controller, ii) OpenvSwitch integrated in KVM, and iii)

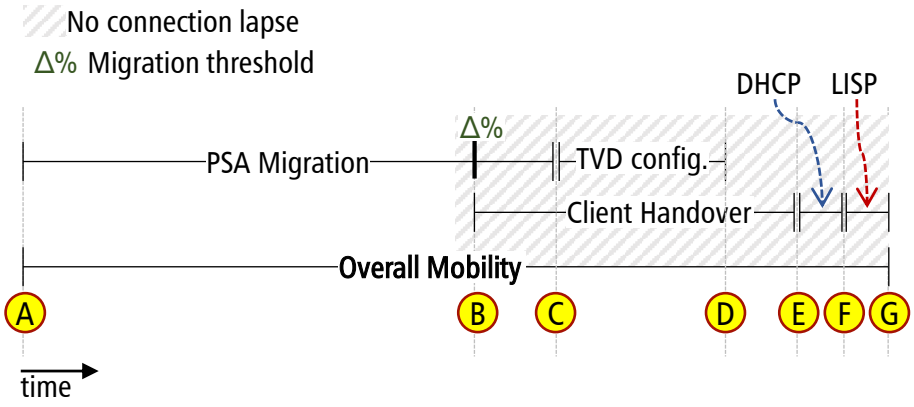


Fig. 8.3 End User mobility stages with TVD migration. Stage A–B: pre-copy VM migration with progress of $\Delta\%$; B–C: VM migration completion; C–D:TVD Configuration; B–E: WiFi disconnect and reconnet; E–F: DHCP IP retrieval; F–G: LISP registration.

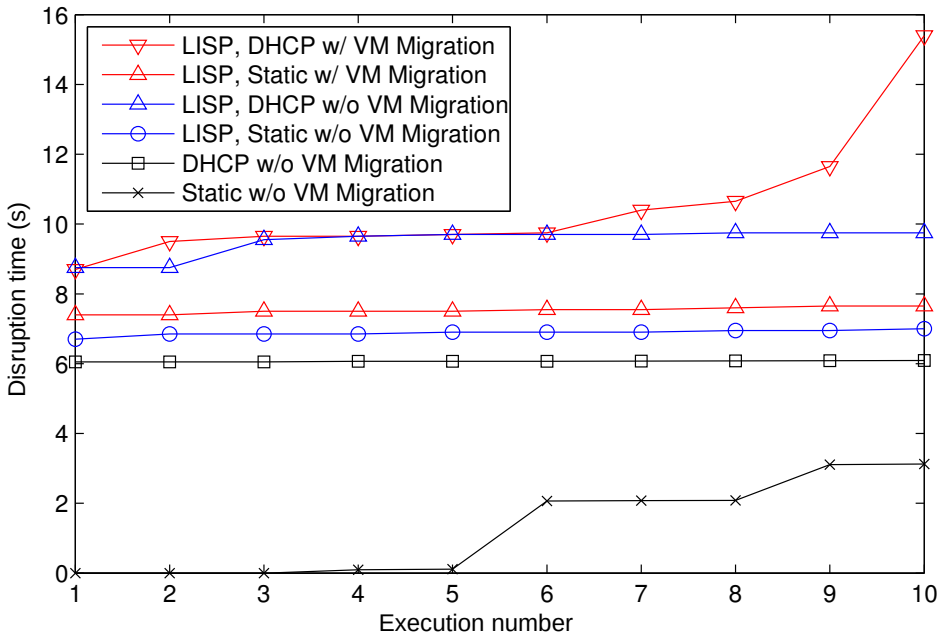


Fig. 8.4 Comparison of the delays obtained for recovering the data plane connectivity through the user's security application when there are different technologies involved in the mobility (w/ stands for with and w/o stands for without).

LISPMob for the mobile end point along with a user application for proactive reporting of available wireless networks. Furthermore, we implemented a basic Orchestrator which is in charge of coordinating the potential migration of the user security during the mobility. The Orchestrator consumes information from the OpenDaylight controller through its REST interfaces, as well as its provided services to dynamically program the virtual network and the traffic steering.

One of the main challenges is to keep the data plane connections after the handover, as WiFi does not allow the attachment to two access points simultaneously. The SDN OpenDaylight controller provide us the flexibility to configure the OpenFlow rules on the OpenvSwitches dynamically. It is worth highlighting that the data plane traffic is enabled if and only if, the user security application is reachable from the user terminal once the handover is successful.

In this setup, our Orchestrator deals directly with the VM migrations through libvirt and KVM. Then, our security applications are composed by a firewall (iptables) and Content Filtering (using the Squid Web Proxy). These applications are bundled into a 512 MB RAM VM image. At the same time, in order to assess the connection lapses during the mobility process, we generated ICMP traffic at 50ms intervals between the user terminal and a remote destination.

The overall results are shown in Fig. 8.4. This figure presents the aggregated disruption time over the user connections. The horizontal axes represent the number of trials the experiment was executed, while the vertical axes the disruption time in seconds. Observe that, since the Orchestrator triggers the Pre-copy live migration in advance and coordinates the overall process, there is no significant difference between handovers with and without VM migration. In other words, the pre-copy live migration up to the 80% ($\Delta = 80$) progress is negligible compared with the time required for the WiFi disconnection and connection.

We also observe that, even in a small set up, both LISP and DHCP introduce considerable delay, with combined effects that may produce variations between 7 seconds up to almost 10 seconds. For the simplest handovers, i.e., no DHCP, no LISP, and no VM migration, we surprisingly observed some cases with zero packet losses. This might be caused due to buffering on the sender transmission queues. Clearly, this type of handover is the one that would actually set the reference of what can be achieved without technical tweaks, which is around 3 seconds. These results have provided us an initial insight about the feasibility of our proposal.

8.6 Summary and Contributions

This chapter contributions focuses on extending our user-centric security protection for the case of mobile users. Different challenges are still to overcome prior the realization of the proposed user mobility support, as our main aim is to provide a protection to the users independently of the device used, and maintain it deployed and instantiated the closest network-wise. The advantages of leveraging the network edge capabilities allow the on-path protection of the users, while also consolidates and facilitates the proper migration of the protection when required.

Along with the advent of virtualization and computing at the edge technologies, offloading the security and protection of the user to the network edge is becoming more attractive. To this end, in this work we leveraged the use of virtualization and fog computing to propose a security framework, which combined with SDN and an NFV-like technologies is able to orchestrate a user's security applications and their migration. To validate our proposal, we deployed a proof-of-concept prototype considering a WiFi mobile user. The obtained results provide an initial insight regarding the impact introduced by the handover and migration processes over the ongoing connections. Challenges such as scalability of users, other network access technologies and vertical handover, and new

lightweight container-based migration technologies are part of our future work. As a result of this research, the following paper has been published:

- D. Montero and R. Serral-Gracià, "Offloading personal security applications to the Network Edge: A mobile user case scenario," *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Paphos, 2016, pp. 96-101. DOI: 10.1109/IWCMC.2016.75770409

- M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing," *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, 2014, pp. 325-329. DOI: 10.1109/CAMAD.2014.7033259
  Contribution: research and evaluation of current technologies to support mobility of endpoints and process migration. I configured and executed different experiments of a case scenario to demonstrate the IP mobility of endpoints while the traffic was inspected by a process located at the network edge. This experimentation also included the migration of the process along with the endpoint movement toward a closer location. Technologies like live virtual machine migrations and container/process live migration were studied and tested.

Part IV: Conclusions and Future Work

Chapter 9

Conclusions

This thesis has studied a different approach to enhance the security protection in the Internet, with special focus on the design and development of pragmatic solutions aimed at improving the security of both core Internet building components and end users and devices. In a nutshell, we addressed this problem from the perspective of an offloading security paradigm. Its main objective seek to provide an attractive and compelling solution in order to foster its adoption. Throughout our research work, there were a lot of lessons learned on regard the vulnerabilities of the Internet and all the complexities blocking already proposed security solutions by the community. Similarly, we have addressed a really sensitive topic driven by the growing awareness around cyberspace security of Internet core players, but more prominently end users. The continuous discoveries of new and more sophisticated threats and attacks over the Internet has contributed to get the attention of more people.

The first part of the thesis addressed main Internet vulnerabilities that affect key core networking components. More specifically, we focused on the vulnerabilities of the BGP and LISP Internet routing protocols. We have identified current open issues and proposed novel solutions to tackle them with the objective to improve their security and have a positive impact over the Internet security in general.

Current Inter-domain BGP vulnerabilities have proven to be exploitable at large scale with unpredictable outcomes. Cases of large Internet disruptions have been denounced and investigated. The motivations behind them range from inadvertent mis-configurations, to possible attacks seeking a benefit against adversaries or to self-induced censorship. The Internet community cognizant of the BGP shortcomings have been working in this direction proposing different solutions to enhance the BGP security. However, we have

focused over the "Route Leak" vulnerability, which remains open even in the presence of the latest solutions proposed by the IETF SIDR WG: RPKI, ROA and BGP-SEC.

Our study around Route Leaks presented a basic theoretical framework including realistic hypotheses and theorems, under which an AS is able to detect route leak initiation autonomously. A main concern that drove our proposal was the impact it would have over the BGP protocol. Thus, we devised our solution following the offloading paradigm. Therein, we discussed the different possibilities to enhance the BGP security while reducing the impact over its current deployment. The main advantages of our approach include, (a) no reliance on third party information (e.g., vantage points), and (b) offloading of the leak detections, thus, no changes required to control-plane protocols. Therein, we proposed a novel Route Leak Detection technique name Cross-Path (CP), which seeks to infer whether new advertisements are leaks based on the current BGP information. To prove the feasibility and effectiveness of our technique, we have tested it with the help of large scale simulations. The initial insights obtained from the results showed different detection success rates. These differences are highly related with the topological location of the domains, and their Internet routes visibility in term of paths diversity.

The second key Internet component we studied in this thesis was the LISP protocol vulnerabilities. Therein, we proposed a novel and adaptable approach for secure the 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. In our solution we introduced a new, separate role in the registration process named as EID-Holder. This new player enabled secure and dynamic EID authorization, while enhancing a dynamic registration to support mobility. 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 Inter-domain Routing (SIDR) WG for resolving the RLOC Authorization part.

The second part of this thesis focused over the security and protection vulnerabilities of end users and end devices. More specifically, we focused on the current issues faced by end users due the current device-centric security protection, as well as the main security issues and challenges faced by the IoT realm. Our approach and proposed solutions followed the offloading paradigm.

From the end user perspective, we have argued that for the large majority of Internet users, the current protection model against security threats is broken. Users typically have multiple devices, but achieving the same level of protection irrespective of the

device used has become “mission impossible”. We have proposed a paradigm shift in user protection through a user-centric model that also decouples security from user terminals. The protection model we envision is based on the setup of a trusted virtual domain per user, placed in the access network.

The uniform security demanded by users within an “always connected” world environment unveils a great opportunity for innovation. Different devices and network access technologies enable users to access ubiquitously and remain connected, even when a user is on the move. In this scenario, both security and mobility remain important research topics. On top of this, the current security paradigms and how mobility is addressed have been probed to fall short in fulfilling the current and future end users requirements. To this end, we have proposed a novel architecture able to overcome most of the issues present in a mobile user-centric scenario with offloaded security. In this work we leveraged the use of virtualization and fog computing to propose a security framework, which combined with SDN and an NFV-like technologies is able to orchestrate a user’s security applications and their migration at the network edge. To validate our proposal, we deployed a proof-of-concept prototype considering a WiFi mobile user. The obtained results provide an initial insight regarding the impact introduced by the handover and migration processes over the ongoing connections. Challenges such as scalability of users, other network access technologies and vertical handover, and new lightweight container-based migration technologies are part of our future work.

Finally, in the IoT realm we have discussed the current security challenges around the fast growing adoption of resource-constrained devices and the services they support. We argued the different dimensions that complicates the security of IoT, which include device diversity, geographical distribution, scalability, and resource-constrained capabilities. Therein, our contributions focus on extending the proposed user-centric security protection along the IoT realm. Different challenges remain to overcome prior the realization of the proposed IoT and service protection at scale, while instantiating the IoT-VD the closest to the things. The advantages of leveraging the fog computing architecture allow the on-path protection of the “things” while also consolidates and facilitates the proper management and control of the different devices.

Chapter 10

Future Work

This chapter describes the different research directions that remain open and represent potential lines of future work.

10.1 Offloaded security of core Internet components

In this thesis, we studied a systematic and analytical approach to leverage locally available information from the BGP routing system to enhance its security. Our proposed framework enables the decoupling from the protocol of all this information processing, offloading it to an external entity. In this thesis, the offloaded Route Leak Detection algorithm proposed is based only on the local BGP control plane information of one router. However, the framework enables for new detection proposals to be investigated, while extending the number of BGP speakers per domain consider. We plan to continue our research around further extending and improving the detection algorithm, while scouting in other fields such as Kalman filters, Bayesian models and machine learning. Aligned with this, we plan to continue our efforts to deploy our solution leveraging the Opendaylight BGP LS PCEP project.

Similarly, this thesis studied a different routing protocol which is based on the semantic split of the addressing spaces, i.e., LISP. Our proposed solution foster user mobility over a rather static Internet. Open issues we plan to pursue include further evaluation of our end-to-end map registration in real scenarios, while also research the impacts of the deployment of the RLOC authorization over the RPKI repository, and study its propagation impact over the registration process.

10.2 User-centric security protection

The user-centric security protection proposed in this thesis offers a compelling and attractive solution with benefits at the user end. However, there are open issues around management and orchestration as well as of scalability. The former issue poses the requirement to further investigate solutions for managing such a hybrid architecture for the allocation and instantiation of security applications at the network edge. Proposals like NFV, and more specifically the Management and Organization architecture (MANO) are in under our radar. On the other hand, the scalability issue may be affected by the virtualization technology employed. Full VMs for security applications per user might become a problem, thus we plan to investigate on more lightweight virtualization technologies (e.g., containers).

Other open challenge regarding the user-centric offloaded security consists on the allocation and instantiation of a compound TVD along different computing nodes. A compound TVD consists of a graph of various Execution Environments EE (each hosting a security application). Hence, this declarative graph will be the input the orchestrator processes to decide where to instantiate each EE, and the proper configuration to connect them.

Similarly, in this thesis we have partially addressed the user mobility with offloaded security case. Internet User Mobility by itself remain an open challenge. We plan to further investigate on new techniques to better support seamless mobility of end user from a twofold perspective: (a) maintaining the user security close to the user, and (b) reduce the impact due to the changes at the IP communications. The former is closely related with the research line at the management and orchestration level describe above. Further research around coordination, resource allocation and security migration in heterogeneous infrastructures remain open challenges. On the other hand, the later is related with the Internet mobility traversal challenge, and its implications to the network routing at the edge and the core.

10.3 IoT security challenges

Our proposed IoT security protection targets to protect both the devices and the services that leverage them. However, there are open challenges which were not covered in this work, and are considered for our future work. The areas we plan to continue our research include: IoT model definition, model-driven construction service graphs, and the management and orchestration of elements over heterogeneous infrastructures.

References

[1] Ager, B., Chatzis, N., Feldmann, A., Sarrar, N., Uhlig, S., and Willinger, W. (2012). Anatomy of a Large European IXP. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '12, pages 163–174, New York, NY, USA. ACM.

[2] ANA Research Group (2012). Path-State Protocol (PSP). http://www.craax.upc.edu/psp.html. [Online; Accessed 2017-05-18].

[3] ARIN (2014). American Registry for Internet Numbers (ARIN). http://www.arin.net/. [Online; accessed 21-Mar-2017].

[4] Atkinson, R. and Bhatti, S. (2012). Identifier-Locator Network Protocol (ILNP) architectural description. RFC 6740, IETF.

[5] Aust, S., Prasad, R. V., and Niemegeers, I. G. M. M. (2012). Ieee 802.11ah: Advantages in standards and further challenges for sub 1 ghz wi-fi. In *2012 IEEE International Conference on Communications (ICC)*, pages 6885–6889.

[6] Bertino, E. and Islam, N. (2017). Botnets and internet of things security. *Computer*, 50(2):76–79.

[7] BGP++ (2014). http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/. [Online; Accessed 2017-05-18].

[8] BGPMON (2015). Massive route leak causes internet slowdown. http://bgpmon.net/massive-route-leak-cause-internet-slowdown/. [Online; accessed 21-Mar-2017].

[9] BGPMON (2016). Large scale bgp hijack out of india. http://bgpmon.net/large-scale-bgp-hijack-out-of-india/. [Online; accessed 21-Mar-2017].

[10] Bluetooth SIG (2017). Bluetooth Low Energy. https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/le-p2p. [Online; Accessed 2017-05-18].

[11] Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. (2012). Fog Computing and its Role in the Internet of Things. In *Proceedings of the ACM SIGCOMM 2012*, SIGCOMM '12. ACM.

[12] Bush, R. and Austein, R. (2013). The Resource Public Key Infrastructure (RPKI) to Router Protocol. RFC 6810, IETF.

[13] Chi, D. P., Secci, S., Pujolle, G., Raad, P., and Gallard, P. (2012). An Open Control-Plane Implementation for LISP Networks. *IEEE INT. Conference on Network Infrastructure and Digital Content*.

[14] Clark, D. D., Wroclawski, J., Sollins, K. R., and Braden, R. (2005). Tussle in cyberspace: Defining tomorrow's internet. *IEEE/ACM Trans. Netw.*, 13(3):462–475.

[15] Clark, Christopher and Fraser, Keir and Hand, Steven and Hansen, Jacob Gorm and Jul, Eric and Limpach, Christian and Pratt, Ian and Warfield, Andrew (2005). Live Migration of Virtual Machines. In *Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 273–286, Berkeley, CA, USA. USENIX Association.

[16] Crossler, R. E., Bélanger, F., and Ormond, D. (2017). The quest for complete security: An empirical analysis of users' multi-layered protection from security threats. *Information Systems Frontiers*.

[17] Dainotti, A., Squarcella, C., Aben, E., Claffy, K. C., Chiesa, M., Russo, M., and Pescapé, A. (2014). Analysis of country-wide internet outages caused by censorship. *IEEE/ACM Transactions on Networking*, 22(6):1964–1977.

[18] Deering, S. and Hinden, R. (1998). Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF.

[19] Dimitropoulos, X., Krioukov, D., Fomenkov, M., Huffaker, B., Hyun, Y., claffy, k., and Riley, G. (2007). AS Relationships: Inference and Validation. *SIGCOMM Comput. Commun. Rev.*, 37(1):29–40.

[20] Docker (2016). Docker - Build, Ship, and Run Any App, Anywhere. https://www.docker.com/. [Online; Accessed 2017-05-18].

[21] ETSI (2013). Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action. https://portal.etsi.org/nfv/nfv_white_paper.pdf. [Online; Accessed 2017-05-18].

[22] ETSI (2013). Network functions virtualisation (NFV) architectural framework. http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf. [Online; Accessed 2017-05-18].

[23] Farinacci, D., Fuller, V., Meyer, D., and Lewis, D. (2013). The Locator/ID Separation Protocol (LISP). RFC 6830, IETF.

[24] Fuller, V. and Farinacci, D. (2013). Locator/ID Separation Protocol (LISP) Map-Server Interface. RFC 6833, IETF.

[25] Fuller, V., Farinacci, D., Meyer, D., and Lewis, D. (2013). Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT). RFC 6836, IETF.

[26] Fuller, V., Lewis, D., Ermagan, V., Jain, A., and Smirnov, A. (2017). LISP Delegated Database Tree. draft-ietf-lisp-ddt-09.

[27] Gagliano, R. (2009). A profile for endpoint identifier origin authorizations (IOA). Internet-draft, IETF.

[28] Ganzha, M., Paprzycki, M., Pawłowski, W., Szmeja, P., and Wasielewska, K. (2017). Semantic interoperability in the internet of things: An overview from the inter-iot perspective. *Journal of Network and Computer Applications*, 81:111 – 124.

[29] Gao, L. (2001). On Inferring Autonomous System Relationships in the Internet. *IEEE/ACM Trans. Netw.*, 9(6):733–745.

[30] Gao, L. and Rexford, J. (2001). Stable internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692.

[31] GNS3 (2014). GNS3: Graphical Network Simulator. http://www.gns3.net/. [Online; accessed 21-Mar-2017].

[32] Goldberg, S., Schapira, M., Hummon, P., and Rexford, J. (2014). How secure are secure interdomain routing protocols? *Computer Networks*, 70:260 – 287.

[33] Goodell, G., Aiello, W., Griffin, T., Ioannidis, J., McDanial, P., and Rubin, A. (1999). Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *Proc. Internet Global Summit. INET 99.*

[34] Goodell, G., Aiello, W., Griffin, T., Ioannidis, J., McDanial, P., and Rubin, A. (2003). Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *Proc. Internet Society Symp. Netw. Distributed Syst. Security (NDSS) (2003).*

[35] H., T., Arkko, J.and Thaler, D., and D., M. (2015). Architectural Considerations in Smart Object Networking. RFC 7452, IETF.

[36] Handley, M., Bonaventure, O., Raiciu, C., and Ford, A. (2011). Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4271, IETF.

[37] Housley, R. (2009). Cryptographic Message Syntax (CMS). RFC 5652, IETF.

[38] Hummel, B. and Kosub, S. (2007). Acyclic Type-of-relationship Problems on the Internet: An Experimental Analysis. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 221–226, New York, NY, USA. ACM.

[39] Huston, G., Loomans, R., and Michaelson, G. (2012). A Profile for Resource Certificate Repository Structure. RFC 6481, IETF.

[40] Huston, G. and Michaelson, G. (2012). Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs). RFC 6483, IETF.

[41] IETF (2014). IETF Secure Inter-Domain Routing (SIDR) Working Group. http://datatracker.ietf.org/wg/sidr/. [Online; Accessed 21-Mar-2017].

[42] Information Sciences Institute University of Southern California (1981). Internet Protocol Specification. RFC 791, IETF.

[43] Karlin, J., Forrest, S., and Rexford, J. (2006). Pretty good bgp: Improving bgp by cautiously adopting routes. In *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, pages 290–299.

[44] Kent, S., Lynn, C., Mikkelson, J., and Seo, K. (2000). Secure Border Gateway Protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):103–116.

[45] Kirkpatrick, K. (2013). Software-defined networking. *Commun. ACM*, 56(9):16–19.

[46] Krishnamurthy, V., Faloutsos, M., Chrobak, M., Cui, J.-H., Lao, L., and Percus, A. G. (2007). Sampling large internet topologies for simulation purposes. *Comput. Netw.*, 51(15):4284–4302.

[47] Labovitz, C. (2010). China hijacks 15% of internet traffic. http://www.arbornetworks.com/asert/2010/11/china-hijacks-15-of-internet-traffic/. [Online; accessed 21-Mar-2017].

[48] Lad, M., Massey, D., Pei, D., Wu, Y., Zhang, B., and Zhang, L. (2006). Phas: A prefix hijack alert system. In *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, USENIX-SS'06, Berkeley, CA, USA. USENIX Association.

[49] Lepinski, M. and Kent, S. (2012). An Infrastructure to Support Secure Internet Routing. RFC 6480, IETF.

[50] Lepinski, M., Kent, S., and Kong, D. (2012). A Profile for Route Origin Authorizations (ROAs). RFC 6482, IETF.

[51] Lepinski, M. and Turner, S. (2011). An overview of bgpsec. draft-lepinski-bgpsec-overview.

[52] Li, S., Duan, H., Wang, Z., and Li, X. (2015). *Route Leaks Identification by Detecting Routing Loops*, pages 313–329. Springer International Publishing, Cham.

[53] Linux (2017). rsync. https://linux.die.net/man/1/rsync. [Online; Accessed 2017-05-18].

[54] LISPMob (2017). Lisp mobile — lispmob. https://lispmob.org. [Online; Accessed 2017-05-18].

[55] Maino, F., Ermagan, V., Saucez, D., and Cabellos-Aparicio, A. (2014). LISP-security (LISP-SEC). draft-ietf-lisp-sec-12.

[56] McDaniel, P. and Prakash, A. (2006). Methods and limitations of security policy reconciliation. *ACM Trans. Inf. Syst. Secur.*, 9(3):259–291.

[57] Medved, J., Varga, R., Tkacik, A., and Gray, K. (2014). Opendaylight: Towards a model-driven sdn controller architecture. In *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, pages 1–6.

[58] Merkel, D. (2014). Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239).

[59] Montenegro, G., Schumacher, C., and Kushalnagar, N. (2007). IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919.

[60] Montero, D., Yannuzzi, M., Shaw, A., Jacquin, L., Pastor, A., Serral-Graciá, R., Lioy, A., Risso, F., Basile, C., Sassu, R., Nemirovsky, M., Ciaccia, F., Georgiades, M., Charalambides, S., Kuusijarvi, J., and Bosco, F. (2015). Virtualized Security at the Network Edge : A User-centric Approach. *Communications Magazine*, pages 1–10.

[61] Nanog (2007). Detecting Route Leaks by Counting - NANOG 41. https://www.nanog.org/meetings/nanog41/presentations/mauch-lightning.pdf. [Online; Accessed 15-Jun-2017].

[62] Netfilter (2017). iptables. [Online; Accessed 2017-05-18].

[63] NetIT Research Group (2017). Overlays for Control Plane Security. http://www.netit.upc.edu/research-projects/. [Online; accessed 21-Mar-2017].

[64] NS-2 Consortium (2016). The Network Simulator — NS-2. http://www.isi.edu/nsnam/ns/. [Online; Accessed 2017-05-18].

[65] Oorschot, P. v., Wan, T., and Kranakis, E. (2007). On interdomain routing security and pretty secure BGP (psBGP). *ACM Trans. Inf. Syst. Secur.*, 10(3).

[66] OpenDaylight (2013). BGP LS PCEP Project. https://wiki.opendaylight.org/view/BGP_LS_PCEP:Main. [Online; Accessed 21-Mar-2017].

[67] OPENLISP (2014). The OpenLISP Project. http://www.openlisp.org/. [Online; Accessed 21-Mar-2017].

[68] Potaroo (2017). BGP Routing Table Analysis Reports. http://bgp.potaroo.net/. [Online; Accessed 15-Jun-2017].

[69] Qiu, S., McDaniel, P., and Monrose, F. (2007). Toward Valley-Free Inter-domain Routing. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 2009–2016.

[70] Quagga (2013). Quagga Routing Software Suite, GPL licensed. [Online; Accessed 2017-05-18].

[71] Rekhter, Y., Li, T., and Hares, S. (2006). A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF.

[72] RIPE (2014). RIPE Network Coordination Center. http://www.ripe.net/. [Online; Accessed 21-Mar-2017].

[73] RIPE NCC (2010). Youtube hijacking: A ripe ncc ris case study. http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study. [Online; accessed 21-Mar-2017].

[74] Rothenberger, B., Asoni, D. E., Barrera, D., and Perrig, A. (2017). Internet kill switches demystified. In *Proceedings of the 10th European Workshop on Systems Security*, EuroSec'17, pages 5:1–5:6, New York, NY, USA. ACM.

[75] Satyanarayanan, M., Bahl, P., Caceres, R., and Davies, N. (2009). The Case for VM-Base Cloudlets in Mobile Computing. *Pervasive Computing*, 8:14–23.

[76] Saucez, D., Iannone, L., and Bonaventure, O. (2016). Locator/ID Separation Protocol (LISP) Threat Analysis. RFC 7835, IETF.

[77] Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., and Oliveira, A. (2011). Smart cities and the future internet: Towards cooperation frameworks for open innovation. In *Future Internet Assembly 2011: Achievements and Technological Promises*, LNCS.

[78] SECURED (2015). Security at the Network Edge. http://www.secured-fp7.eu/. [Online; accessed 21-Mar-2017].

[79] SECURED (2017). The SECURED project repo. [Online; Accessed 2017-05-18].

[80] Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., and Sekar, V. (2012). Making middleboxes someone else's problem: Network processing as a cloud service. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 13–24, New York, NY, USA. ACM.

[81] Shinde, R., der Veeken, P. V., Schooten, S. V., and van den Berg, J. (2016). Ransomware: Studying transfer and mitigation. In *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pages 90–95.

[82] Siganos, G. and Faloutsos, M. (2004). Analyzing bgp policies: methodology and tool. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1640–1651 vol.3.

[83] Sigma Designs (2016). Z-Wave Public Specification. http://z-wave.sigmadesigns.com/design-z-wave/z-wave-public-specification/. [Online; Accessed 2017-05-18].

[84] Squid-cache (2017). Squid-cache. [Online; Accessed 2017-05-18].

[85] Sriram, K., Montgomery, D., Dickson, B., Patel, K., and Robachevsky, A. (2017). Methods for Detection and Mitigation of BGP Route Leaks. Internet-Draft draft-ietf-idr-route-leak-detection-mitigation-06, Internet Engineering Task Force. Work in Progress.

[86] Sriram, K., Montgomery, D., McPherson, D. R., Osterweil, E., and Dickson, B. (2016). Problem Definition and Classification of BGP Route Leaks. RFC 7908.

[87] Subramanian, L., Agarwal, S., Rexford, J., and Katz, R. (2001). Characterizing the Internet Hierarchy from Multiple Vantage Points. Technical report, University of California at Berkeley, Berkeley, CA, USA.

[88] Sundaresan, S., Lychev, R., and Valancius, V. (2013). Preventing Attacks on BGP Policies: One Bit is Enough. Technical Report GT-CS-11-07, Georgia Institute of Technology.

[89] The CAIDA UCSD IPv4 Routed /24 Topology Dataset - 01.04.2014 (2014). http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml. [Online; Accessed 2014-04-10].

[90] White, R. (2003). Securing BGP through secure origin BGP (soBGP). *Internet Protocol Journal*, 6(3).

[91] WiFi Alliance (2017). Wi-Fi HaLow. http://www.wi-fi.org/discover-wi-fi/wi-fi-halow. [Online; Accessed 2017-05-18].

[92] Wikipedia (2016). 2016 Dyn cyberattack. https://en.wikipedia.org/wiki/2016_Dyn_cyberattack. [Online; accessed 21-Feb-2017].

[93] Yannuzzi, M., Milito, R., Serral-Graciá, R., Montero, D., and Nemirovsky, M. (2014). Key ingredients in an IoT recipe: Fog Computing, Cloud Computing, and more Fog Computing. In *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on*, CAMAD '14, pages 325 – 329. IEEE.

[94] Yannuzzi, M., Serral-Gracià, R., Siddiqui, S., Serra, E., and Masip-Bruin, X. (2012). BGP Path-State Overlays as an Alternative to BGPSEC. http://www.craax.upc.edu/pdf/PSP_Demo_LACNOG.pdf. [Online; Accessed 2017-05-18].

[95] Zhang, H., Qin, Z., and Yang, Q. (2008). Design and implementation of the tpm chip j3210. In *Proceedings of the 2008 Third Asia-Pacific Trusted Infrastructure Technologies Conference*, APTC '08, pages 72–78, Washington, DC, USA. IEEE Computer Society.

[96] Zigbee Alliance (2017). ZigBee IP Specification. http://www.zigbee.org/download/standard-zigbee-ip-specification/#. [Online; Accessed 2017-05-18].

Appendix A

List of Publications

Journals

(1) D. Montero, M. Yannuzzi, A. Shaw, L. Jacquin, A. Pastor, R. Serral-Gracià, A. Lioy, F. Risso, C. Basile, R. Sassu, M. Nemirovsky, F. Ciaccia, M. Georgiades, S. Charalambides, J. Kuusijarvi, F. Bosco. "Virtualized Security at the Network Edge: A User-centric Approach," in *IEEE Communications Magazine*, vol. 53, no. 4, pp. 176-186, April 2015. DOI: 10.1109/MCOM.2015.7081092

(2) M.S. Siddiqui, D. Montero, R. Serral-Gracià, X. Masip-Bruin, M. Yannuzzi, "A survey on the recent efforts of the Internet Standardization Body for securing inter-domain routing," *Computer Networks*, Volume 80, 7 April 2015, Pages 1-26, ISSN 1389-1286. DOI: 10.1016/j.comnet.2015.01.017

(3) M. S. Siddiqui, D. Montero, R. Serral-Gracià , M. Yannuzzi, "Self-reliant detection of route leaks in inter-domain routing," *Computer Networks*, Volume 82, 8 May 2015, Pages 135-155, ISSN 1389-1286. DOI: 10.1016/j.comnet.2015.02.029

(4) Yannuzzi, M., van Lingen, F., Jain, A., Parellada, O. L., Flores, M. M., Carrera, D., Pérez, J. L., Montero, D., Chacin, P., Corsaro, A., and Olive, A. "A new era for cities with fog computing", in *IEEE Internet Computing*, vol. 21, no. 2, pp. 54–67, April 2017. DOI: 10.1109/MIC.2017.25

Conferences

(1) D. Montero, M. S. Siddiqui, R. Serral-Gracià, X. Masip-Bruin and M. Yannuzzi, "Securing the LISP map registration process," 2013 *IEEE Global Communications*

Conference (GLOBECOM), Atlanta, GA, 2013, pp. 2145-2151. DOI: 10.1109/GLOCOM.2013.6831392

(2) D. Montero and R. Serral-Gracià, "Offloading personal security applications to the Network Edge: A mobile user case scenario," *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Paphos, 2016, pp. 96-101. DOI: 10.1109/IWCMC.2016.75770409

(3) M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing," *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Athens, 2014, pp. 325-329. DOI: 10.1109/CAMAD.2014.7033259

(4) M. S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracià and X. Masip-Bruin, "Route leak identification: A step toward making inter-domain routing more reliable" *2014 10th International Conference on the Design of Reliable Communication Networks (DRCN)*, Ghent, 2014, pp. 1-8. DOI: 10.1109/DRCN.2014.6816139

(5) M. S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracià, X. Masip-Bruin and W. Ramirez, "Route leak detection using real-time analytics on local BGP information," *2014 IEEE Global Communications Conference*, Austin, TX, 2014. DOI: 10.1109/GLOCOM.2014.7037092

Appendix B

Projects

Part of the work in this thesis has been used in the following Research Projects:

European Project

- **FP7 SECURED Project**: Security at the Network Edge, SECURED Project FP7-611458, (Oct 2013-Sept 2016).

Cisco FogPoC

- Cisco - Barcelona Fog Computing PoC (May-Oct 2015).

Cisco RFP Project

- Overlays for Control Plane Security (Jan-Sept 2013).