A PRAGMATIC APPROACH TOWARD SECURING INTER-DOMAIN ROUTING

Author: Muhammad Shuaib Siddiqui
Advisor: Dr. Marcelo Yannuzzi
Co-Advisor: Dr. Xavier Masip-Bruin
Department of Computer Architecture
Universitat Politècnica de Catalunya (UPC)

A Thesis presented to the Universitat Politècnica de Catalunya (UPC) in fulfillment of the requirements for the degree of Doctor in Computer Science.

September, 2014

Beyond the stars there are worlds more
Our quest yet has more tests to pass
This existence alone does not matter
There are boundless journeys more
Do not rest on what you have
There are paradises more to explore
Why worry if you have lost one abode
There are a million addresses to claim
You are the falcon, your passion is flight
And you have skies more to transcend
Lose not yourself in the cycle of days and nights
Within your reach are feats even more
Gone is the day when I was lonesome in the crowd
Today those who resonate my thoughts are more
— **Allama Muhammad Iqbal** (1877-1938)

to Hamza Ali , Mustafa Ali and Sumayyah

Acknowledgements

I would like to extend my heartfelt appreciation to all the people who have made the development and writing of this thesis possible with their support and help.

This thesis would not have been possible without the guidance and support of my thesis supervisor, Dr. Marcelo Yannuzzi. I highly appreciate the encouragement and guidance you always extended toward me to carry on my research work. I would also especially like to thank my co-supervisor, Dr. Xavi Masip, for the invaluable good advice and support through out the tenure of my thesis. I am very grateful to Dr. Rene Serral for his insightful discussions and guidance during the development of my thesis. I also wish to express my special gratitude to Diego Montero for his support at different stages of my work. I would also like to thank all the former and current members of the ANA research group for the friendly and supportive working environment.

I would also like to express my sincere appreciation for Roque Gagliano, Keyur Patel, Fabio Maino, Rodolfo Milito, and Vina Ergaman from Cisco Systems for the valuable discussions which helped refining my work tremendously.

Words can not express how grateful I am to my parents for all the prayers and unmentioned sacrifices that they have made for me so far and continue to do so unconditionally. I owe an immeasurable debt of gratitude to my wife, Sumayyah, for her never-ending patience, unconditional support and absolute love. And most importantly, Alhamdulilah-i-Rab-bil Aalameen!

Barcelona, September 2014 M. S. Siddiqui

Abstract

Internet security poses complex challenges at different levels, where even the basic requirement of availability of Internet connectivity becomes a conundrum sometimes. Recent Internet service disruption events have made the vulnerability of the Internet apparent, and exposed the current limitations of Internet security measures as well. Usually, the main cause of such incidents—even in the presence of the security measures proposed so far—is the unintended or intended exploitation of the loop holes in the protocols that govern the Internet.

In this thesis, we focus on the security of two different protocols that play a key role both in the present and the future of the Internet. To this end, the thesis is structured in two main technical parts. In the first one, we focus on the security of the of the Border Gateway Protocol (BGP) [1], while in the second one, we concentrate on the security of the Locator Identifier Separation Protocol (LISP) [2].

The Border Gateway Protocol (BGP) is the de-facto inter-domain routing protocol in the Internet, and therefore, it plays a crucial role in current communications. Unfortunately, it was conceived without any internal security mechanism, and hence is prone to a number of vulnerabilities and attacks that can result in partial paralysis of the Internet. In light of this, securing BGP has been an active research area since its adoption and numerous security strategies, ranging from a complete replacement of the protocol up to the addition of new features in it were proposed. However, none of them were pragmatic enough to be widely accepted and only minor security tweaks have found the pathway to be adopted. Even the recent IETF Secure Inter-Domain Routing (SIDR) Working Group (WG) [3] efforts including, the Resource Public Key Infrastructure (RPKI) [4], Route Origin Authorizations (ROAs) [5], and BGP Security (BGPSEC) [6] do not counter an important set of security issues, especially, the policy related ones, such as route leaks. The main reason behind the occurrence of route leaks is the violation of the routing policies, more specifically the export policies, among the Autonomous Systems (ASes). Route leaks have the potential to cause large scale Internet service disruptions, as reported in [7] and [8]. Usually, the AS policies were largely neglected out in the past security proposals due to their confidential nature. There exist a few rudimentary solutions that can be used as a first line of defense, such as the utilization of route filters, but these palliatives become unfeasible in large domains due to the administrative overhead and the cost of maintaining the filters updated. As a result, a significant part of the Internet is defenseless against route leak attacks. In this part of the thesis, we examine the route leak problem and propose pragmatic security methodologies which a) require no changes to the BGP protocol, b) are neither dependent on third party information nor on third party security infrastructure, and c) are self-beneficial regardless of their adoption by other players. That is, our security approach offers zero entropy to the widely deployed BGP protocol. The independence from third party information avoids the security burden required for securely exchanging the information. Moreover, it does not require implementation of new protocols for interacting with the third party security infrastructure as well. Another notable characteristic of our solutions is that they take AS relationship information with direct neighbors into account for resolving the route leak problem. And more importantly, the effectiveness of our security methodologies do not depend on their mass adoption, i.e., they remain potent and improve the security of the domain implementing them even though if no other domain adopts them. In this regard, our main contributions in this part of the thesis can be summarized as follows. We develop a theoretical framework, which, under realistic assumptions, enables a domain to autonomously determine if a particular route advertisement received from a neighbor corresponds to a route leak. Based on this, we propose three incremental techniques, namely Cross-Path (CP), Benign Fool Back (BFB), and Reverse Benign Fool Back (R-BFB), for autonomously detecting route leaks. Our strength resides in the fact that these detection techniques solely require the analytical usage of in-house control-plane, data-plane and direct neighbor relationships information, which are already available within the domain. We evaluate the performance of the proposed route leak identification techniques both through real-time experiments as well as using simulations at large scale. Our results show that the proposed detection techniques achieve high success rates for countering route leaks in different scenarios.

In the other technical part, we focus our attention on securing the LISP protocol. The motivation behind LISP protocol has shifted over time from solving routing scalability issues in the core Internet to a set of vital use cases for which LISP stands as a technology enabler. However, as in the case of BGP, LISP was born without security, and therefore is susceptible to attacks in its control-plane. The IETF's LISP working group [9] has recently started to work toward securing LISP, but the protocol still lacks end-to-end mechanisms for securing the overall registration process on the mapping system ensuring RLOC authorization and global EID authorization. As a result LISP is unprotected against different attacks, such as RLOC spoofing, which can cripple even its basic functionality. Furthermore, lack of any mechanism for global EID authorization raises concerns for the practical feasibility of mobility and roaming features in LISP. For that purpose, in this part of the thesis we address the above mentioned issues and propose practical solutions that counter them. Our security proposals for LISP take advantage of the low technological inertia of the LISP protocol, i.e., our solutions leverage on the fact that the LISP protocol is not as widely adopted as the BGP protocol, and hence can be more accommodating toward security solutions proposing changes to the protocol along with using existing third party security infrastructure for establishing security. The changes proposed for the LISP protocol and the utilization of existing security infrastructure in our solutions enable resource authorizations and lay the foundation for the much needed end-to-end security.

Contents

List of Figures

List of Tables

List of Acronyms

AD	Authentication Data
AES	Advanced Encryption Standard
API	Application Programming Interface
ARIN	American Registry for Internet Numbers
AS(es)	Autonomous System(s)
ASN	Autonomous System Number
BFB	Benign Fool Back
BGP	Border Gateway Protocol
BGPSEC	BGP Security
BSM	Backend Subsystem Module
CA	Certification Authority
CLI	Command Line Interface
CMS	Cryptographic Message Syntax
CP	Cross-Path
CRLs	Certificate Revocation List
CRL	Customer Route Leak
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
DSA	Digital Signature Algorithm
eBGP	External BGP
ECDSA	Elliptic Curve Digital Signature Algorithm

EGP	Exterior Gateway Protocol
EID	End-point Identifier
EE	End Entity
ETR	Egress Tunnel Router
IANA	Internet Assigned Numbers Authority
iBGP	Interior Gateway Protocol
IETF	Internet Engineering Task Force
IOA	Identifier Origination Authorization
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
INR	Internet Number Registry
IV	Initial Vector
IRV	Internet Route Verification
IRR	Internet Route Registry
ITR	Ingress Tunnel Router
IXP	Internet Exchange Point
LACNIC	Latin America and Caribbean Network Information Centre
LISP	Locator/ID Split Protocol
Loc-RIB	Local Routing Information Base
LRC	Local RPKI Cache
MS	Map Server
MR	Map Resolver
MRAI	Minimum Route Advertisement Interval
NLRI	Network Layer Reachability Information
OPENER	Open Programmable Environment for Experimenting with Routers
OTK	One Time Key
OTP	One Time Password

PHAS	Prefix Hijacking Alert System
PKI	Public Key Infrastructure
PRL	Peer Route Leak
ps BGP	Pretty Good BGP
PSP	Path State Protocol
RA	RLOC Authorization
R-BFB	Reverse Benign Fool Back
RFC	Request For Comment
RFD	Route Flap Dampening
RIB	Routing Information Base
RIR	Regional Internet Registry
RLD	Route Leak Detection
RLOC	Route Locator
ROI	Return Of Investment
ROA	Route Origin Authorization
RP	Relying Party
RPKI	Resource Public Key Infrastructure
RR	Route Reflector
RS	Route Server
s BGP	Secure BGP
SHA	Secure Hash Algorithm
SDK	Software Development Kit
SDN	Software Defined Networking
so BGP	Secure Origin BGP
SRL	Stub Route Leak
TCP	Transmission Control Protocol
TEFIS	Testbed for Future Internet Services

Introduction Part I

1 Executive Summary and Road Map

1.1 Motivations

The security of the Internet has been one of the top priorities of researchers since its commercial adoption two decades ago. However, even today the basic necessity of availability of Internet connectivity can not be guaranteed. The disruption of Internet services due to security vulnerabilities is a frequent occurrence, but only a few incidents succeed to get mass attention—this typically depends on the scale of the service disruption, and the profiles of the affected Autonomous Systems (ASes) and the alleged attacker. Recent events such as the Youtube service breakdown [11], alleged Chinese Telecom traffic hijacking [12], Google services disruption [7], or the national level Internet service failure in Australia [8], have made the vulnerability of the Internet apparent, and, at the same time, unmasked the current limitations of Internet security measures. For the most part, the main cause of such incidents—even in the presence of the security measures proposed thus far—can be traced back to the unintended or intended exploitation of the security inadequacies in the protocols that govern the core of the Internet. In this thesis, we center our attention on the security of two different protocols that target two distinct aspects of the Internet, namely the Border Gateway Protocol (BGP) [1], and the Locator Identifier Separation Protocol (LISP) [2]. We classify this thesis in to two technical parts. In the first one, we focus our attention on the security of the BGP protocol, and in the second one, we concentrate on the security of the LISP protocol.

The BGP protocol is at the crux of the Internet as it is used to exchange reachability information among the thousands of domains that compose the Internet. All of the above mentioned incidents producing Internet service disruptions were related to vulnerabilities of the BGP protocol in one way or another. In spite of the efforts made by the research community for almost two decades, the security of the BGP protocol remains as precarious as ever. Several security proposals based on clean-slate approach (i.e., complete replacement of the BGP protocol), or addition of security features in to the protocol, or third party security infrastructure, were recommended. However, none of the main proposals made thus far has been sufficiently pragmatic to be globally adopted. While some of them required the addition or replacement of hardware elements, others needed the replacement of key software components, or they simply lacked details about their deployment and entire functioning, hence they did not convince the industrial players that needed to support an initiative of such magnitude. Apart from this, the policy related security issues in the inter-domain, such as the route leak problem which carries the potential to cause large scale Internet connectivity failure, did not receive the due attention they deserve from the research community; so much so that even the recent solutions recommended by the IETF working group incharge of securing the inter-domain routing, SIDR WG [3], including the Resource Public Key Infrastructure (RPKI) [4], Route Origin Authorizations (ROAs) [5], and BGP Security (BGPSEC) [6], do not address the route leak problem. Route leaks occur due to policy violations while exporting routes, using the BGP protocol, to a neighbor AS. Route leaks are apparently simple but hard to solve. This is because the ASes keep the information regarding their relationships and policies with other ASes confidential, which makes the identification of policy violations a challenging problem. Although there are stopgap countermeasures for the route leak problem, including route filters, Internet Route Registries (IRRs), and several BGP monitoring tools, they become impotent or unreliable in face of scalability, due to the high cost of maintenance and dependence on third party information. Hence, the main motivation of this part of the thesis is to thoroughly address the route leak problem and recommend pragmatic security solutions. One of the main motivation of this thesis is the practicality of the proposed solutions such that they do not fall prey to the technological inertia of the BGP protocol, i.e., they are efficiently adoptable in the inter-domain arena.

LISP, on the other hand, offers a new addressing and routing architecture, which was initially devised for solving the routing scalability issues in the core of the Internet. Because of the intrinsic address splitting architecture, it is also anticipated to be a technology enabler for several use cases in different areas of networking, such as rapid IPv6 deployment, high scale virtualization including support for Virtual Machine (VM) migrations, etc. Hence, the LISP protocol is envisioned to play a key role in the future Internet. Unfortunately, like BGP, LISP was conceived with little or no intrinsic security features, and thus is vulnerable to a variety of attacks. These security attacks on LISP not only jeopardize its normal operations but also undermine its potential as a technology enabler. Although another protocol was developed to improve the security of the LISP protocol, namely LISPSEC [13], it only provides security blanket to a set of LISP control-plane messages and fails to address the security needs of LISP in a larger perspective. All the LISP security solutions proposed thus far, fall short of providing any resource authorization mechanism. In absence of such mechanisms, LISP remains defenseless against a variety of attacks such as EID address or RLOC address spoofing. Furthermore, the EID holder entity is not involved in the security solutions yet which not only serves as an impediment for end-to-end security but also for global EID authorization, much required for the LISP mobility feature. Fortunately, the LISP protocol is not widely deployed yet and hence does not pose a huge technological inertia, like BGP. This fact allows us more flexibility and less constraints on the practicality of security solutions for LISP. Thus, the motivation of the second technical part of this thesis is to equip the LISP protocol with pragmatic security strategies which not only allow end-to-end security including resource authorization but also secure its future as a technology enabler.

In general, we believe that pragmatically securing the protocols that play a critical role in the routing system is a must for enhancing the overall security of the Internet. Therefore, in this thesis, we study the vulnerabilities of both protocols, including the main causes of attacks against them. Through this process, we analyze the recent security developments, covering the solutions that could be used to curb the security vulnerabilities identified. We particularly examine the unresolved security issues, and propose a pragmatic set of solutions according to each protocol.

1.2 Objectives

The main aim of this thesis is to study and enhance the security of the inter-domain routing system, by strengthening the security of the protocols that govern its functionality in the present and the future of the Internet. However, the specific objectives for the two protocols under consideration in this thesis are given next.

1.2.1 BGP Security Objectives

1. To thoroughly study and analyze the so far neglected route leak problem in the inter-domain routing, with special focus on why the so far proposed security solutions are still inadequate.
2. To develop a set of pragmatic recommendations and strategies that can be used to mitigate the route leak problem while considering the available AS policies knowledge.
3. Benefiting from the lessons learned, the proposed security solutions should not only be academically feasible but also pragmatic for real world adoption given the history of BGP security efforts.

1.2.2 LISP Security Objectives

1. To analyze the major security vulnerabilities of the LISP protocol due to lack of resource authorization mechanism, with focus on why the so far proposed security solutions are still inadequate.
2. To propose a pragmatic security framework which ensures dynamic resource authorization along with end-to-end security on the LISP control-plane.

We identify separate objectives for the two protocols mainly because of two reasons, 1) the nature of the security needs of the two protocols are substantially different, if not entirely, 2) the constraints on the practicality of a security solution varies greatly for the two protocols. In the next section, we further highlight the need for pragmatic security solutions for both, the BGP and the LISP protocols, and outline the contributions made by this thesis.

1.3 Needs and Contributions

1.3.1 The Need for Pragmatic Approaches considering Policies while Securing BGP

The Border Gateway Protocol (BGP) is the default protocol for exchanging reachability information in the inter-domain arena of the Internet. Unfortunately, the current version of the protocol (version 4) does not provide any performance or security guarantees [1]. Although BGP always remained part of the Internet research community due to several concerns related to its convergence [14–22], its churn [23–25], its limitations in terms of traffic engineering [26–29], security [30–42], policies [43–50], documented anomalies [51–56], and other issues [57–65], the recent global outages in the Internet acted as a catalyst for reviving the research focus toward its security.

As mentioned earlier, several security mechanisms and protocols have been proposed during the past two decades to improve the security of the BGP protocol. However majority of them focused on securing the BGP protocol from an operational perspective and hence left out the policy related security issues of the inter-domain routing. Route leak is one of the prominent policy related inter-domain vulnerability that occurs when the routing policies agreed between two neighbor Autonomous Systems (ASes) are not respected. This type of policy violation takes place during the route advertisement process between these ASes. More precisely, the business relationship between any two ASes steers their export and import routing policies, and a route advertised against the conceded policies is called a route leak. Route leaks can lead to partial paralysis of Internet services, and may affect both local as well as global regions. Route leaks can be either the result of a misconfiguration or a deliberate attack, and, apart from Internet service disruption, they can lead to sub-optimal routing and traffic hijacking. Therefore, route leaks can be very harmful, and they are considered a security threat for the interdomain routing system. For example, in February 2012, an Internet service failure at national level occurred in Australia, when a multi-homed ISP leaked routes learned from one of its providers to another provider [8]. In November of the same year, Google services were disrupted when one of Google's peers improperly advertised Google routes to its provider [7].

The first line of defense for preventing route leaks typically consists of utilizing route filters along with Internet Route Registries (IRRs) information, but this palliative usually becomes futile due to the high administrative cost of maintaining the filters updated. Other stopgap solutions, such as BGP monitoring tools, rely on the information collected at various vantage points, but they are only fruitful when the irregularities are observed at the vantage points themselves. In this regard, the fact that adds to the difficulty of countering route leaks using vantage points is the secrecy of the routing policies among ASes. Although several attempts have been made for inferring the relationships and the policies among ASes (see, e.g., [45, 66, 67]), more recent works are currently questioning the accuracy of these techniques [68]. This is mainly due to the fact that the knowledge base for inferring the AS relationships and their corresponding export policies is limited to the routing information available at the data collection points. Despite these efforts, only minor tweaks have finally reached an operational status in practice. More recently, the Secure Inter-Domain Routing (SIDR) [3] working group at IETF [69], has put forward several recommendations, for securing the BGP protocol, which have gained interest from industry as well as from the research community. Indeed, a couple of the recommendations have already been adopted by regional Internet registries [70][71] and several providers. However, SIDR's contributions for securing BGP, including: the Resource Public Key Infrastructure (RPKI) [4], Route Origin Authorizations (ROAs) [5] and BGPSEC [6], do not secure BGP against the route leaks. Like most of the existing proposals, SIDR WG recommendations approach BGP security from an operational perspective, and neglect the business policies among the ASes altogether. Mindful of the fact that the amount of rationally available AS policies knowledge at a domain is limited, we contend that it is fundamental to include it in the inter-domain security solutions.

The failure of the traditional countermeasures for detecting route leaks is evident from the frequent occurrences of Internet service disruptions due to these incidents [72]. Learning from the collapse of conventional solutions, we can infer that any pragmatic approach toward resolving the route leak problem should consider the following factors: 1) no or minimum possible changes to the BGP protocol; 2) no or minimum reliance on third party information; 3) real-time detection; and 4) minimum possible administrative overhead. The first factor stems from the fact that a solution requiring significant changes to control-plane protocols, such as the BGP protocol, will meet the same fate as such previous inter-domain security propositions, i.e., resistance in wide-scale adoption. The minimum reliance on third party information is important not only because of the limited reach of the information gathered at vantage points, but also because of the high administrative cost required to train and maintain the monitoring infrastructure up-to-date with the routing policies. Furthermore, serious efforts are required for trust establishment between the relying party and the third party to avoid bogus information exchanges. The real-time detection is a necessity because of the way route leaks operate. That is, detecting the route leak on the link where it occurs the first time is easier than detecting propagated route leaks (cf. Chapter 6), hence early detection of a route leak is essential. The low administrative overhead of the security solution is a vital requirement to ensure that the solution does not become a liability in face of scalability.

In this thesis, we specifically emphasize on the needs described above while attempting to resolve the route leak problem. So far in this chapter, we have highlighted the need to resolve the route leak problem and identified the essential features of a possible pragmatic solution; now we proceed to present the contributions made with respect to BGP security in this thesis.

Contributions in BGP Security

The contributions made by this thesis in enhancing the BGP include the following.

- Formal definition and description of the route leak problem, i.e., we analyze and characterize the route leak problem. Furthermore, we identify two major types of route leaks, namely, Customer Route Leak (CRL) and Peer Route Leak (PRL), and explain how, where, and why they occur.

- A theoretical framework for the detection of route leaks. We show that under realistic assumptions and routing conditions, a single AS is able to autonomously detect both types of route leaks in different scenarios. Our route leak detection framework utilizes only the standard routing information available at hand, without needing any vantage point deployed in the internetwork or third party security infrastructure.

- Based on the proposed theoretical framework we introduce three incremental real-time route leak detection techniques, namely Cross-Path (CP), Benign Fool Back (BFB) and Reverse Benign Fool Back (R-BFB). The first two techniques, CP and BFB, are based on the analytical usage of BGP's control-plane information, whereas the third technique, R-BFB, also takes advantage of data-plane traffic to provide additional information to the analytics performed on the BGP RIBs. It is worth mentioning that, none of the above route leak detection techniques, require any change to the BGP protocol.

- Performance evaluation of the proposed route leak detection techniques using large-scale simulations and real-time experiments. For the large-scale simulations, we tested the proposed route leak detection techniques on actual subgraph of global-scale ARK's Internet graph [73], i.e., all the domains, links, and the AS relationships used in our simulations, are actually present in ARK's Internet graph. For the experimental part, we deployed an inter-domain network topology on the Heterogeneous Experimental Network (HEN) [74] testbed, with the aim of testing our route leak identification techniques in a scenario that can realistically support the data-plane part. The results from our tests show that an AS is able autonomously detect route leaks in different scenarios with a high success rate using the CP, BFB and R-BFB, especially, when the three techniques are combined and used together.

The contributions made with regard to BGP security in this thesis target the route leak problem in the inter-domain routing. The proposed solution, including the route leak detection framework and techniques, is non-repellent to the existing inter-domain infrastructure as it does not require changes to the BGP protocol, does not require mass deployment of new infrastructure, and does not depend on third party information or third party security infrastructure. Most importantly, our solution remains self-beneficial for a single domain regardless of its adoption by other neighboring domains which encourages early adoption of the solution by every domain.

1.3.2 Needs and Contributions for LISP Security

The Locator/Identifier Separation Protocol (LISP) [2] 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, such as virtualization, mobility, and cloud applications. In this section, we very briefly overview the working of the LISP protocol in order to highlight its security needs in a more understandable manner. The basic idea in LISP is the decoupling of the Route LOCator (RLOC) and the End-point IDentifier (EID) spaces in the addressing scheme. LISP supports provider independent and globally unique Identifier addresses, and employs a Map-and-Encap scheme, along with an Identifier-to-Locator Mapping System to bind the two address spaces. 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 [2], this map registration process is a static procedure based on manual configurations that need to be set in advance. These configurations need to be done both on the border routers in the LISP-Site, called Egress Tunnel Routers (ETRs) and on the Map Server. Once the manual configurations are in place, each ETR will attempt to register its mappings with the Map Server. The latter can verify the requests against the predefined configuration using pre-shared keys. The pre-shared keys allow to assess the validity of the map registration, since each ETR has its own key which is shared only with the Map Server.

It is important to notice that this existing *pre-shared key* security mechanism between the ETR and the MS falls short of countering a number of relatively simple attacks, such as RLOC address spoofing. LISP lacks a procedure for ensuring whether a certain ETR is allowed to use a particular RLOC address for registering an EID prefix, i.e, it has no RLOC authorization mechanism. Furthermore, LISP has no mechanism to ensure that a certain ETR is authorized to register a particular EID prefix on the Map Server. This is mainly because the 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 are essential for the practical feasibility of mobility and roaming scenarios in LISP. The IETF's working group on LISP has recently started to work in this direction, and their first initiative is LISPSEC [13]. For the moment, LISPSEC has only focused on securing the mapping queries, i.e., the *Map-Request* and *Map-Reply* messages with the Mapping System, so the vulnerabilities mentioned above remain unsolved. In this thesis, we seek to address the above identified security needs of the LISP protocol. Overall, securing the map registration process including the EID-Holder as an entity and RLOC authorization are vital requirements, not only to avoid attacks on the LISP control-plane but also to maintain a correct operational state of the Mapping System while ensuring end-to-end security.

Contributions in LISP Security

The contributions made by this thesis for securing LISP protocol include the following.

- Introduction of a new player in the LISP working framework, the EID-Holder. The identification of the role of the EID-Holder not only paves the way for end-to-end security on the LISP control-plane but also facilitates the global EID authorization requirement.
- A secure map registration process for LISP that not only works end-to-end (i.e., it now involves the EID-Holder) but also enables dynamic map registrations. Our solution leverages on existing security infrastructure and utilizes efficient cryptographic mechanisms, which combined with actual LISP messaging, are able to enhance the overall security of the map registration process.

During this thesis work, we observed that both the SIDR WG [3] and the LISP WG [9] are targeting security aspects that involve the Internet's global routing system, but their efforts so far have been entirely disjoint. To bridge this gap, our work also explores how LISP can benefit from the security infrastructure already designed by the SIDR WG, including the Resource Public Key Infrastructure (RPKI) [4] and Route Origin Authorizations (ROAs) [5].

1.4 Thesis Structure

This thesis is arranged in four main parts including this introductory one. Next, we provide the synopsis of the rest of the thesis along with the topics covered in each chapter.

PART II

Chapter 2 briefly reviews the business relations and policies among ASes, as well as their effects in inter-domain routing, so as to facilitate the understanding of the BGP security issues described in Chapter 3. For the same purpose, this chapter also includes a brief review of the BGP protocol.

Chapter 3 outlines different types of vulnerabilities of the BGP-4 protocol. It also illustrates some of the main security vulnerabilities of the BGP-4 protocol which have lead to large scale Internet service disruptions.

Chapter 4 provides a survey on the recent contributions made by SIDR WG along with a comparison of the design principles of SIDR WG recommendations with earlier proposed solutions. This chapter also provides a detailed analysis and excess burden of the SIDR WG security recommendations along with the illustration of residual attacks which remain unresolved.

Chapter 5 lays the foundation for the studying the route leak problem in the next chapters. First, it describes two real world route leak occurrences including the related work as well. Then it dives in to the anatomy of route leaks to provide a formal definition of the problem.

Chapter 6 describes the theoretical framework for autonomously detecting detecting route leaks. Based on the theoretical framework, it presents three incremental route leak detection techniques, namely, Cross-Path (CP), Benign Fool Back (BFB), and Reverse Benign Fool Back (R-BFB).

Chapter 7 presents the simulation and experimental framework for evaluating the proposed route leak detection techniques. It also provide analysis on the results obtained from the simulations and experiments.

PART III

Chapter 8 briefly describes the LISP protocol and its working framework. It also discusses LISP existing security arsenal including its intrinsic security features and the LISP-SEC protocol in order to provide better understanding of the LISP security issues in next chapter.

Chapter 9 illustrates LISP security vulnerabilities including EID impersonation, RLOC spoofing, and lack of global EID authorization which remain unresolved in the presence of the existing LISP security measures.

Chapter 10 introduces an updated LISP working framework including a working role of the EID holder. Then, it describes an end-to-end secure map registration process based on efficient cryptographic mechanisms. This proposed secure map registration process not only enables dynamic map registrations, facilitating the much required mobility feature, but also enhances the overall security of the LISP protocol.

PART IV

Chapter 11 summarizes the main achievements of this thesis.

Chapter 12 suggests different directions for broadening the reach of the contributions made in this thesis.

Second Part Part II

2 Inter-Domain Preliminaries

This chapter briefly reviews the business relations and policies among ASes, as well as their effects in inter-domain routing, so as to facilitate the understanding of the BGP security issues described in next chapter. For the same purpose, this chapter also includes a brief review of the BGP protocol.

2.1 Business Relations among ASes

The Internet consists of thousands of inter-connected Autonomous Systems (AS). A collection of network elements and links under a single administration is called an Autonomous System (*AS*) or domain. In Fig. 2.1, AS_1 and AS_2 are two directly connected ASes. AS_1 and AS_2 would exchange their respective IP prefixes with each other such that a source in AS_1 can reach a destination in AS_2 and vice versa. Currently, there are around 45,000 ASes in the Internet, and the reachability is achieved when each AS informs all its neighbors (*directly connected ASes*) about its available IP prefixes according to its routing policy. That is, the routing policy specifies how reachability information is exchanged between any two ASes.

The business relation between any two ASes dictates the kind of policies that would be implemented on that particular link. The business relation between two ASes can be typically classified into either Provider-Customer or Peer-Peer relation [75]. In the latter case, both ASes advertise subsets of their routes, i.e., they only advertise their own or their customer's routes to each other. For the Provider-Customer relation, the Provider and the Customer are the opposite ends of the same link. In this case, the Customer will only advertise its own routes and the routes of its Customers (*i.e., Customer's Customer routes*) toward its Provider link. The Provider AS, on the other hand, will usually advertise all routes toward its customer hence providing it transit to the rest of the Internet.

For example, in Fig. 2.1, AS_3 is a Customer of AS_1 and AS_4 is a Customer of AS_2, whereas AS_1 and AS_2 are Peers. AS_1 being AS_3's Provider would advertise AS_3's routes to AS_2 which would eventually advertise them to AS_4. AS_2 would do the same for AS_4 being its Provider such that

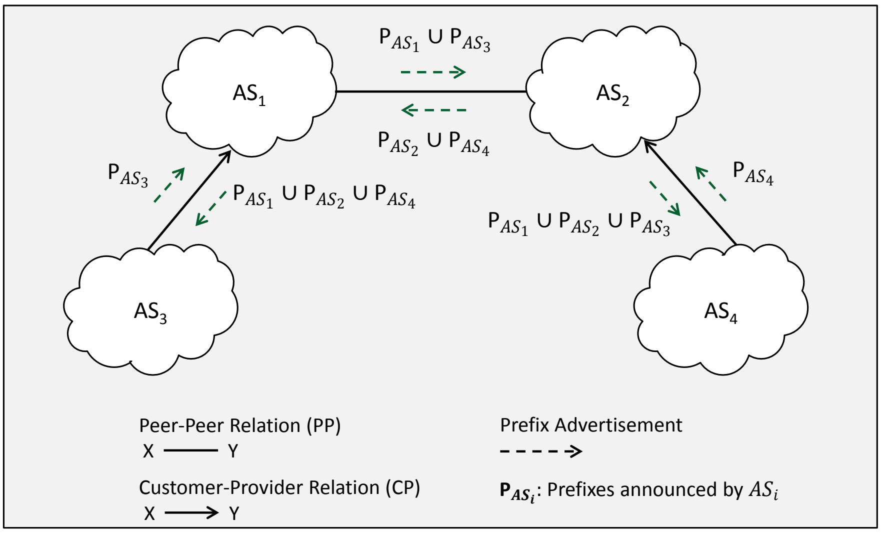


Figure 2.1: Types of business relations among ASes.

a source in AS_3 can reach any available IP prefixes in AS_4 and vice versa.

In a nutshell, the following guidelines, known as valley-free rules [45], are adopted by an AS for further advertisement of received routes:

- Routes learned from Customers are further advertised to other Customers, Peers and Providers (Fig. 2.2(a)).

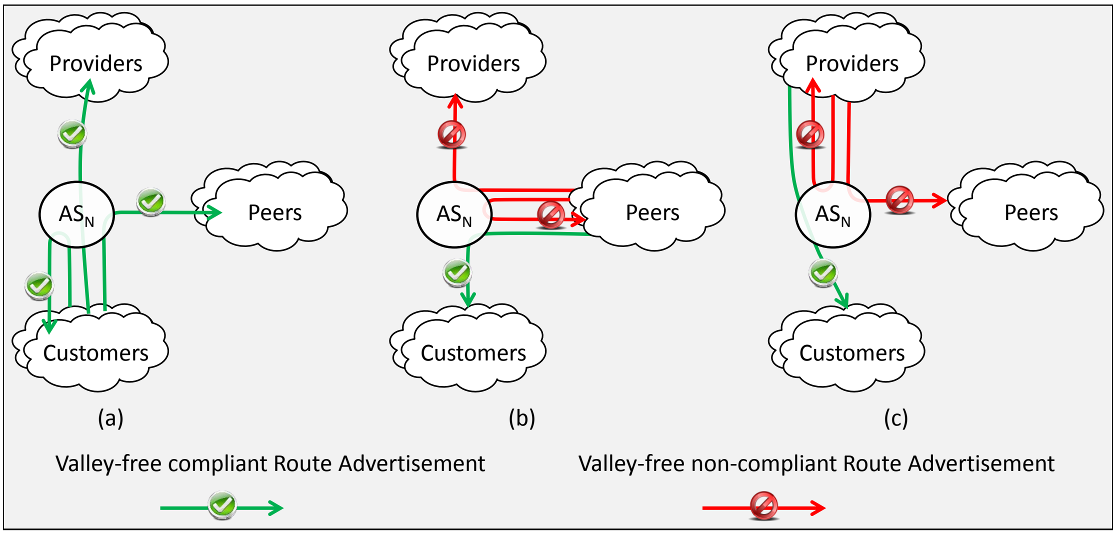

Providers
Peers
ASN
Customers
(a)
(b)
(c)
Valley-free compliant Route Advertisement
Valley-free non-compliant Route Advertisement

Figure 2.2: Route re-advertisements according to the valley-free rules: (a) for routes learnt from customers; (b) for routes learnt from peers; (c) for routes learnt from providers.

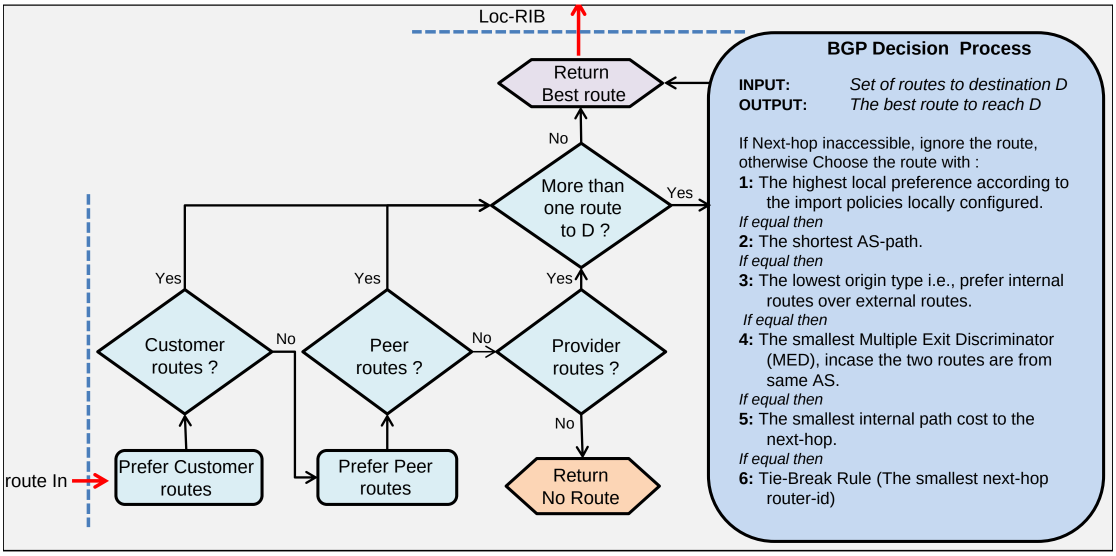
- Routes learned from Peers are further advertised to Customers only (Fig. 2.2(b)).
- Routes learned from Providers are further advertised to Customer's only (Fig. 2.2(c)).

From the business revenue perspective, a Provider charges its Customer for forwarding traffic to and from it, whereas a Peer-Peer relation may not involve financial settlement up to a certain agreed traffic ratio. Hence, in line with profit maximization objectives, an AS prefers a route learnt from a Customer (i.e., a Customer route) over a route learnt from a Peer or a Provider (i.e., a Peer or a Provider route, respectively). Similarly, an AS prefers a Peer route over a Provider route, as Peer links are usually revenue neutral as compared to the Provider one. As shown in Fig. 2.3, for any received route, a Customer route has preference over a Peer or a Provider route, and a Peer route has preference over a Provider route.

However, there exist exceptions to valley-free constraints [49, 76] as they are not upheld sometimes to accommodate customized economic models due to a complex AS relationship between the ASes. For example, for contingency connectivity, ASes usually have one or more backup links with other ASes. These backup links deliberately do not follow valley-free rules to rectify the impact of primary link failures or network congestion [77]. We contend that understanding of AS relationship and AS policies is essential to better comprehend the attacks described in next chapter. For a detailed literature on AS relationship and AS policies, the reader can refer to [75].

2.2 BGP in a Nutshell

As mentioned earlier, the protocol used for exchanging routing information between ASes is the BGP protocol. It is a path vector protocol as it dynamically maintains and advertises the path information (i.e., the path that the route has taken from the origin AS to the last AS) for any given destination [78]. BGP enabled routers are called BGP speakers, and they set up a peering session with their direct neighbors using TCP [79] as underlying transport protocol, prior to exchanging routing information through BGP updates. The BGP updates are exchanged to inform the neighbors about the recently learnt new routes or the withdrawn ones, i.e., a BGP update may consists of a list of withdrawn routes and a list of advertised routes along with their attributes. A BGP route has multiple attributes associated to it, which, as shown in Fig. 2.3, assist the receiving BGP speaker in selecting the best route following the BGP decision process [1]. The *AS_Path* attribute of a route contains the path information, in effect, a sequential list of all the ASes that a specific route passed through. Apart from detecting loops in the AS-Path, the *AS_Path* attribute also assists in calculating the AS-Path length, which is one of the metrics used in the BGP decision process for selecting the best route for a destination. The result of the BGP decision process, i.e., the best route to the destination is stored in a database called Local Route Information Base (Loc-RIB). Furthermore, some of the route attributes also allow ASes to implement their business policies by tuning their values for a route before considering it for route selection or advertising it further. For in-depth



BGP Decision Process

INPUT: *Set of routes to destination D*
OUTPUT: *The best route to reach D*

If Next-hop inaccessible, ignore the route, otherwise Choose the route with :
1: The highest local preference according to the import policies locally configured.
If equal then
2: The shortest AS-path.
If equal then
3: The lowest origin type i.e., prefer internal routes over external routes.
If equal then
4: The smallest Multiple Exit Discriminator (MED), incase the two routes are from same AS.
If equal then
5: The smallest internal path cost to the next-hop.
If equal then
6: Tie-Break Rule (The smallest next-hop router-id)

Figure 2.3: Overview of the BGP decision process to reach a destination “D”. Note that this is an elementary depiction of it, so for a detailed description of the BGP decision process the reader is referred to [1].

details of the BGP protocol itself, the reader can refer to [1], [80], and [81].

3 Inter-Domain Routing Vulnerabilities

This chapter illustrates some of the main security vulnerabilities of the BGP-4 protocol which have lead to large scale Internet service disruptions. The main motivation of this chapter is to facilitate the reader to better understand the recent security efforts of SIDR WG, presented in subsequent chapter.

3.1 Major Vulnerabilities of BGP

The BGP protocol is susceptible to a number of attacks due to the lack of an intrinsic security mechanism. Indeed, the blind trust with which two neighboring BGP speakers accept the exchanged information gives rise to vulnerabilities that can be exploited in different ways. Besides, given the complex operation of BGP, a number of BGP anomalies can even occur due to misconfigurations rather than to malicious intent. In a nutshell, the attacks in BGP can be broadly classified into three categories, namely, false information exchange attacks (e.g., false IP prefix origination and false BGP update), BGP protocol manipulation attacks (e.g., Route Flap Damping and Minimum Route Advertisement Interval attacks [32, 82]), and AS policy violations attacks (e.g., route leaks). It is important highlighting that this chapter only focuses on the well-known and frequently occurring security issues in BGP which have caused wide scale Internet service disruption. The interested readers can refer to [83] and [84] and the references therein for detailed surveys on the security issues of BGP. The main reason for illustrating only the major BGP security issues in this chapter is to provide an insight for the motivation behind the recent developments made by the SIDR WG in securing BGP, described in next chapter.

3.1.1 Advertisement of False AS-Paths

The BGP speakers exchange BGP updates for sharing the reachability information among each other. One of the most important attributes of an advertised route is its *AS_Path* attribute. It lists all the ASes, starting from the origin AS to the AS which sent the update. However, a receiving BGP speaker has no means to verify the authenticity of the AS-Path information attached to a route in the update. Manipulating the *AS_Path* attribute can affect the BGP route decision process and thus lead to hijacking or black-holing of the traffic. In Fig. 3.1, AS_1 generates a false BGP update offering one hop connectivity for IP prefix 10.1.1.0/24 owned by AS_5 to AS_2. AS_2 also receives the valid BGP update advertising IP prefix 10.1.1.0/24, but it will prefer the shortest path according to the BGP decision process (cf. Fig. 2.3). Without external means, AS_2 has no mechanisms to counter check the AS-Path information in the BGP update received from AS_1. This could result in either traffic black-holing, which is detectable as the traffic does not reach its destination, or in the worse case, traffic sniffing which is undetectable as the traffic is forwarded to the destination through a sub-optimal path, i.e., if AS_1 forwards the traffic to AS_5 via AS_3.

AS-Path Shortening: AS-Path shortening is a particular case of false AS-Path attack. In this case, an AS deliberately manipulates the AS-Path information by reducing the AS-Path length so that it becomes more favorable during the route selection process at the next hop (cf. step 2 in Fig. 2.3). It is worth mentioning that BGP legitimately allows BGP speakers to elongate an AS-Path for a route, by only prepending their own autonomous system number (ASN) in the *AS_Path* attribute for more than one time, calling it AS-Path prepending. The AS-Path prepending allows an AS to tune its policy on a link to some extent, but manipulating information other than its own in the AS-Path, or more precisely, removing another AS from the AS-Path refers to AS-Path shortening attack. As shown in Fig. 3.2, when advertising prefix 10.1.1.0/24, AS_2 prepends itself twice on its link toward AS_1 and thrice on its link toward AS_3. In this case, AS_2 treats its link with AS_3 as a backup link for prefix 10.1.1.0/24 due to high cost, hence prepends higher. AS_3 removes the AS-Path prepending performed by AS_2 and

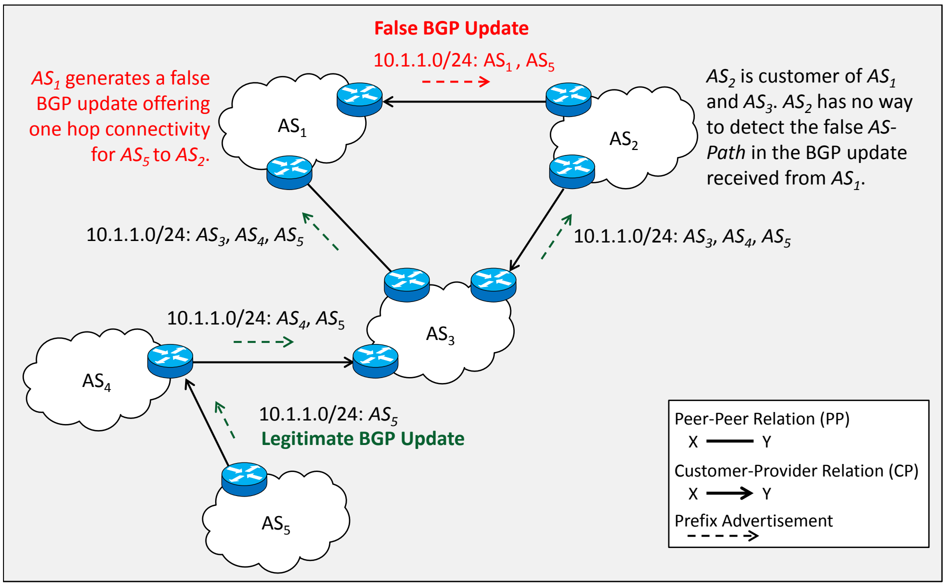

False BGP Update
10.1.1.0/24: AS_1 , AS_5
AS_1 generates a false BGP update offering one hop connectivity for AS_5 to AS_2.
AS_2 is customer of AS_1 and AS_3. AS_2 has no way to detect the false *AS-Path* in the BGP update received from AS_1.
AS_1
AS_2
10.1.1.0/24: AS_3, AS_4, AS_5
10.1.1.0/24: AS_3, AS_4, AS_5
AS_3
10.1.1.0/24: AS_4, AS_5
AS_4
10.1.1.0/24: AS_5
Legitimate BGP Update
AS_5
Peer-Peer Relation (PP)
X —— Y
Customer-Provider Relation (CP)
X ⟶ Y
Prefix Advertisement
- - - ->

Figure 3.1: False AS-Path attack.

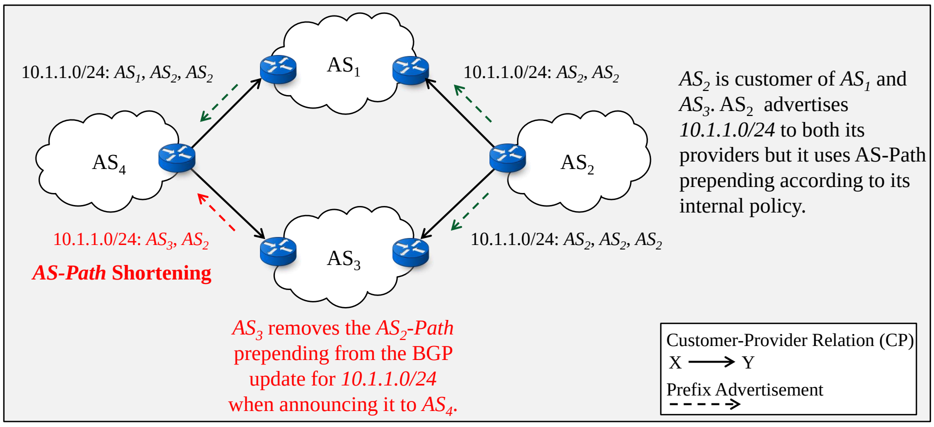

AS1
AS2
AS3
AS4
10.1.1.0/24: AS1, AS2, AS2
10.1.1.0/24: AS2, AS2
10.1.1.0/24: AS3, AS2
10.1.1.0/24: AS2, AS2, AS2
AS-Path Shortening
AS3 removes the AS2-Path prepending from the BGP update for 10.1.1.0/24 when announcing it to AS4.
AS2 is customer of AS1 and AS3. AS2 advertises 10.1.1.0/24 to both its providers but it uses AS-Path prepending according to its internal policy.
Customer-Provider Relation (CP)
X ⟶ Y
Prefix Advertisement
- - - - ->

Figure 3.2: AS-Path shortening attack.

advertises it further to AS_4 favoring its chances to get selected as the best route for prefix 10.1.1.0/24. AS_4 cannot detect the AS-Path shortened by AS_3 and following the doctrine of mutual trust, it has no other way, except to accept it, generating more revenues for AS_3 but costing more to AS_2.

3.1.2 False Route Origination

Every AS advertises the IP prefixes it owns to its neighbor ASes through BGP peering, according to its internal policies and the business relationship that it has with each neighbor. A BGP speaker accepts a received IP prefix advertisement following the implicit trust model of BGP. The BGP protocol does not define any mechanism to verify that the AS originating the IP prefix advertisement is in fact the actual owner of this prefix, hence leaving room for exploitation due to unintentional or deliberate misconfigurations. As shown in Fig. 3.3, the IP prefix 10.1.1.0/24

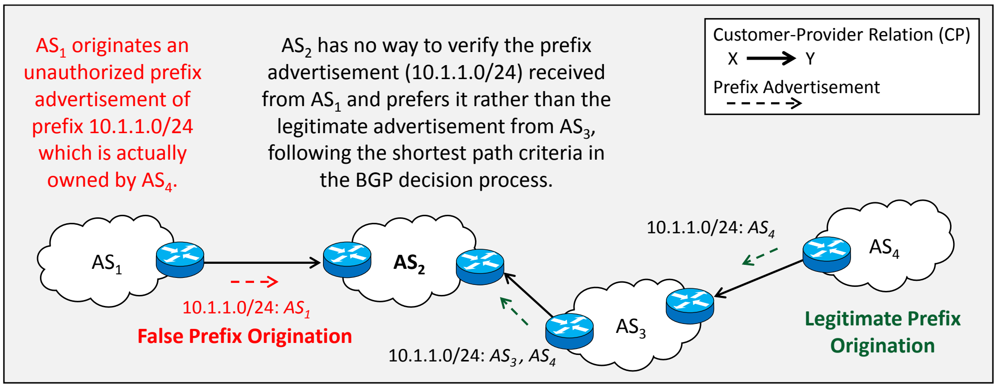

AS1 originates an unauthorized prefix advertisement of prefix 10.1.1.0/24 which is actually owned by AS4.
AS2 has no way to verify the prefix advertisement (10.1.1.0/24) received from AS1 and prefers it rather than the legitimate advertisement from AS3, following the shortest path criteria in the BGP decision process.
Customer-Provider Relation (CP)
X ⟶ Y
Prefix Advertisement
- - - - ->
AS1
AS2
AS3
AS4
10.1.1.0/24: AS1
False Prefix Origination
10.1.1.0/24: AS3, AS4
10.1.1.0/24: AS4
Legitimate Prefix Origination

Figure 3.3: False IP prefix origin attack.

is owned by AS_4, but AS_1 falsely originates the IP prefix 10.1.1.0/24 as its own to AS_2. AS_2 receives advertisements for the same IP prefix from AS_1 and AS_4 and without any out-of-band precautionary measures, AS_2 falls prey to the false advertisement. Following the BGP decision process (cf. Fig. 2.3), AS_2 will prefer the route from AS_1, since it offers the shortest AS-Path toward the destination IP prefix. This way, AS_1 successfully hijacks AS_2's traffic for AS_4.

The lack of in-built mechanisms in BGP to verify the AS origin of an IP prefix leads to the false route origination attack, resulting in either undetectable hijacking or detectable black-holing of the traffic. The false route origination attack differs from the false AS-Path attack in the sense that the latter does not lie about the origin of the prefix, but tries to inject an non-existent path in the network.

3.1.3 Route Leaks

A route leak occurs when a route gets advertised over a link by an AS, which does not coincide with the link classification [85]. We delve in to the route leak problem in much more details in Chapter 5. However, in this section, we provide a brief illustrating example to build an initial understanding of the problem. For example, in Fig. 3.4, AS_1 and AS_2 have a Peer-Peer relation. AS_3 is customer of both, AS_1 and AS_2, i.e., it is multi-homed. AS_2 has another customer, AS_4, which owns the IP prefix 10.1.1.0/24. AS_4 advertises the prefix 10.1.1.0/24 to its Provider AS_2 (Step (i)). AS_2 being a provider of AS_3 and a peer of AS_1, advertises it to both of them (Steps (ii.a) and (ii.b)).

Now if AS_3 advertises 10.1.1.0/24 to AS_1 (Step (iii)), this falls in the category of a route leak.

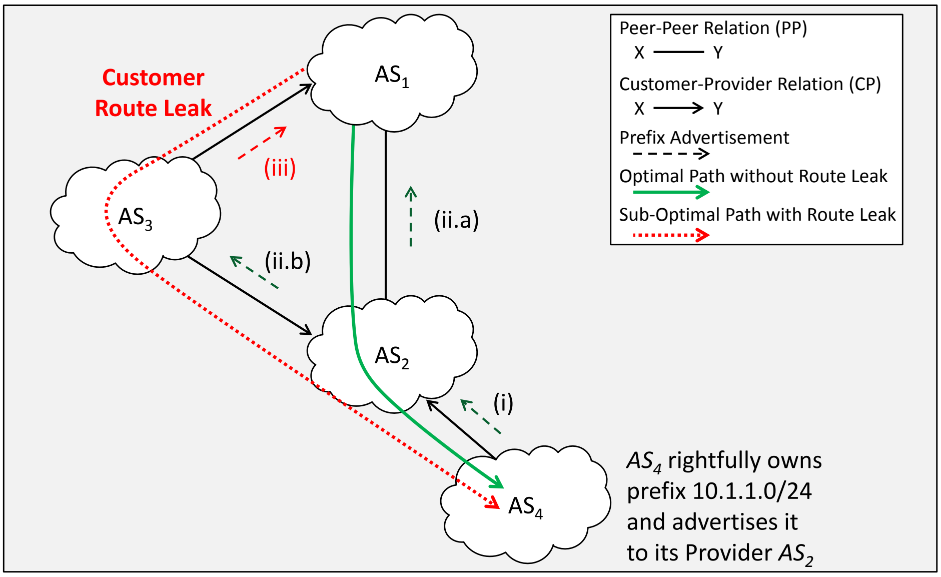

Customer Route Leak
AS1
AS3
AS2
AS4
(iii)
(ii.a)
(ii.b)
(i)
Peer-Peer Relation (PP)
X — Y
Customer-Provider Relation (CP)
X → Y
Prefix Advertisement
Optimal Path without Route Leak
Sub-Optimal Path with Route Leak
AS4 rightfully owns prefix 10.1.1.0/24 and advertises it to its Provider AS2

Figure 3.4: Route Leak on customer link.

In this case, AS_1 gets advertisements for the same prefix from AS_3 and AS_2. As mentioned in Chapter 2, customers' routes are typically preferred over peer routes, hence AS_1 selects AS_3 as its next hop for the IP prefix 10.1.1.0/24, which apart from being sub-optimal, also allows AS_3 to sniff all the traffic from AS_1 to AS_4. This may also result in congestion at AS_3 causing traffic black-holing. The increase in the AS-Path length of the route advertised by AS_3 does not help in detecting this problem because customer routes are preferred by setting the *local-pref* attribute, which is evaluated before the *AS_Path* attribute (cf. step 1 in Fig. 2.3). In other words, the route is decided before comparing the AS-Path lengths. The route leak occurring on a Customer-Provider link is termed as a Customer route leak. Moreover, route leaks are also possible on a Peer-Peer link. In Fig. 3.5, the steps (i), (ii.a), (ii.b) and (iii) illustrate how route leaks can occur on a Peer-Peer link. AS_4 advertises its IP prefix 11.1.1.0/24 to its provider. AS_1 and AS_2 learn about this prefix from their respective providers. Now, if AS_1 advertises 11.1.1.0/24 to AS_2, then AS_2 would prefer the peer route over it's provider route resulting in a route leak, causing the traffic between AS_2 and AS_4 to go through AS_1, making it vulnerable to sniffing.

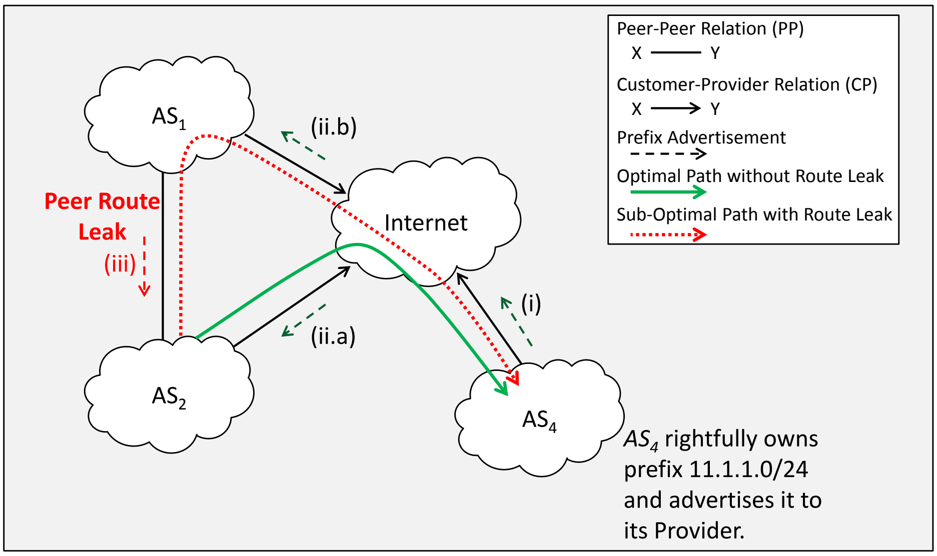

Peer-Peer Relation (PP)
X —— Y
Customer-Provider Relation (CP)
X ——> Y
Prefix Advertisement
Optimal Path without Route Leak
Sub-Optimal Path with Route Leak
AS1
AS2
AS4
Internet
(ii.b)
(ii.a)
(i)
Peer Route Leak
(iii)
AS4 rightfully owns prefix 11.1.1.0/24 and advertises it to its Provider.

Figure 3.5: Route Leak on peer link.

4 Inter-Domain Security: Recent Efforts at IETF

This chapter provides a survey on the most recent contributions made by SIDR WG of IETF. The recommendations from the SIDR WG include several ideas which are taken from the past security solutions. Thus, to put things in to perspective, this chapter presents a comparison of the design principles of SIDR WG recommendations with the ones of the earlier proposed solutions. Furthermore, this chapter also provides a detailed analysis and excess burden of the SIDR WG security recommendations along with the illustration of residual attacks which remain unresolved.

4.1 Comparison of Basic Design Principles of Past Inter-Domain Routing Security Proposals and SIDR's WG

During the past years, several proposals emerged to counter the security vulnerabilities of the inter-domain routing system. In terms of design principles, these security proposals can be broadly classified into three categories: 1) proposals solely based on the alteration of the BGP protocol, e.g., by adding extensions to the protocol, 2) proposals with radical recommendations that advocate for a complete replacement of the BGP protocol, and 3) proposals outlining extensive architectures for securing BGP. The proposals belonging to the first category (i.e., those suggesting extensions to the BGP protocol), only counter one or two specific security anomalies of BGP, hence their impact is rather limited. The so called *clean-slate* proposals in the second category provided reasonable solutions, but they did not succeed in convincing the Internet community about the need for replacing BGP. The third category of proposals have lately received the majority of the attention—especially from the industrial sector—as they provide extensive BGP security solutions contemplating a wide set of BGP anomalies. This category includes S-BGP [86], soBGP [87], psBGP [88] and IRV [89]. The recent security proposal from the SIDR WG also belongs to the third category, which describes a complete security framework for semantically securing BGP. It is important to note that the security proposals belonging to the third category also require changes to the BGP protocol, but these changes are far less disruptive as compared to the ones required by the security proposals in the other two categories. Before we survey the three pillars in SIDR proposals', namely, RPKI [4], ROAs [5] and BGPSEC [6], we will proceed to overview the main design principles on which the past solutions in the third category were based on, and compare them with the ones adopted in SIDR's proposals.

The two main common target goals dictating the design principles for a more secure BGP architecture among the past proposals include prefix origin authorization and AS-Path validation. In this regard, S-BGP's [86] Public Key Infrastructure (PKI) for establishing and maintaining verifiable security credentials, out-of-band address attestations for prefix origin authorizations, and an in-band and new optional transitive BGP attribute to support route validation, appears to be the inspiration behind SIDR's RPKI, ROA and BGPSEC, respectively. In fact, the SIDR's proposals can be seen as a refined version of S-BGP, and this represents the core of what is being standardized by the IETF [90]. However, SIDR's proposals (explained in detail in Section 4.2) describe mechanisms in far more detail and also give explicit considerations to pragmatic aspects of BGP, such as AS-Path prepending and Internet Exchange Point (IXP) transparency, as well as to partial and flexible deployments as compared to S-BGP.

The soBGP [87] proposal recommends extensive use of different types of certificates, namely, *EntityCert*, *AuthCert*, and *ASPolicyCert*, to establish trust among entities and resources, prefix origin authorizations, and existence of advertised AS-Paths, respectively. One of the fundamental design differences between soBGP and SIDR proposals is the use of web of trust instead of a PKI for validating signed objects. In this way, soBGP avoided the burden of a PKI, but was required to establish trust anchors to support its validation framework in the absence of a PKI. The ambiguous validation framework of soBGP casted doubts on the establishment of sufficient trust required for attestations and their validations. In this regard, SIDR proposals employ a hierarchical PKI for trust establishment. Another major design difference was the weak requirement of checking if the AS advertising a certain destination has a feasible route to it (with the use of ASPolicyCert) as compared to the strict AS-Path cryptographic verification employed by SIDR proposals. The design trade-offs including PKI avoidance and weak AS-Path validation was to make it less demanding in terms of processing and memory, but at the cost of weaker security. The SIDR proposals face a set of new challenges due to the hierarchical PKI and strict cryptographic AS-Path validation (cf. Sections 6.2), but it clearly does not compromise on the level of security.

The psBGP [88] proposal realizes a hierarchical PKI for AS numbers, but it suggests to use Internet Route Registries (IRRs) for authenticating the utilization of IP addresses. In this regard, psBGP inherits the shortcomings of IRRs including doubts on the authenticity and integrity of information. As mentioned earlier, SIDR proposals utilize a hierarchical PKI for both, AS numbers as well as for IP address resources, and trust is established using cryptographically verifiable certificates. Given the drawbacks of IRRs, it is obvious why SIDR did not choose an IRR-like framework for the dissemination of security credentials. Instead, the SIDR proposals accomplish the distribution of security credentials by means of a hierarchical infrastructure of repositories and local-caches (cf. Section 6.2.1). Observe that, psBGP employs strict cryptographic validation of AS-Paths, which is also the case in SIDR's proposals, however, psBGP allows partial AS-Path validation as compared to SIDR proposal's advocation of either plain AS-Path or complete cryptographically verifiable AS-Paths. The allowance of partial AS-Path validations through the utilization of confidence levels makes psBGP "partial deployment friendly", however, SIDR proposals argue that using incomplete AS-Path validations is as good as no AS-Path validation at all, since unverifiable portions of an AS-Path undermine the security feature itself (cf. Section 6.3.1).

The IRV [89] proposal is based on a query response framework that is completely separated from the BGP protocol, and it is used for achieving prefix origin authorization and AS-Path validation. According to the proposal, on reception of a BGP update, the corresponding IRV service can query the respective IRV servers in other ASes to verify the required information for prefix origin authorization and AS-Path validation. However, the IRV proposal is silent on the details of how these authorizations and validations would be realized. An interesting feature of IRV is that it does not require changes to the BGP protocol, but, for achieving the security goals, it advocates for the validation of the queries and responses for all the ASes in the AS-Path. The main drawback of this approach is that it requires an underlying functioning network that greatly reduces the protocol scalability in an Internet wide topology. A complete out-of-band on-demand security infrastructure was not an option for SIDR proposals, as they put huge emphasis on the practicality of their solution.

4.2 IETF's Secure Inter-Domain Routing Working Group Contributions

In spite of the efforts made by the research community for almost two decades, the security of the BGP protocol remains as fragile as ever. The obvious conclusion that can be drawn is that, none of the main proposals made thus far has been sufficiently pragmatic to be adopted. While some of them required the addition or replacement of hardware elements, others needed the replacement of key software components, or they simply lacked details about their deployment and entire functioning, hence they did not convince the industrial players that needed to support an initiative of such magnitude. In this adverse scenario, the IETF's Secure Inter-Domain Routing (SIDR) working group (WG) [3] has put up serious effort, and is developing recommendations for securing BGP with strong focus on practical aspects, such as partial deployments. In this regard, one of the central objectives is that there should be clear incentives for early adopters, since if not, security ends up in a sort of game where a domain only really benefits from it if all the other domains in the network are playing the game—we will delve into this later on in Section 4.3.

In a nutshell, the SIDR's proposals approach the security issues in inter-domain routing based on three pillars. The first pillar proposes a security infrastructure based on public key cryptography and is called Resource Public Key Infrastructure (RPKI) [4]. The RPKI has already been implemented by some domains and regional Internet registries, and is available for testing [70][71]. The second pillar provides mechanisms for secure route origination, and is called Route Origin Authorization (ROA) [5]. It has also been implemented, and is currently in testing phase as well. The third and most controversial pillar is still under discussion, and proposes to modify the BGP protocol for secure route propagation, and is called BGPSEC [6]. Both ROA and BGPSEC depend on the RPKI infrastructure for their operation, and therefore, for achieving their respective goals. A number of drafts and RFCs have been published by the SIDR WG detailing all the proposed mechanisms and recommendations for practical deployment and early adoption. A list of published RFCs and drafts by the SIDR WG can be found in [3]. In the rest of this section, we survey these three pillars and describe their roles in securing BGP, while illustrating their mechanisms and functioning in achieving their respective goals.

4.2.1 The First Pillar: Resource Public Key Infrastructure (RPKI)

RPKI is a vital part of SIDR proposals since it defines and provides the basic security skeleton for the other two pillars. RPKI consists of three main parts: 1) a resource allocation hierarchy; 2) a set of cryptographically protected objects; and 3) a distributed repository framework to hold these objects. Overall, RPKI mirrors the currently practiced administrative allocation hierarchy of Internet Number Resources (INRs) (e.g., IP addresses and AS numbers), where resources are distributed from the Internet Assigned Numbers Authority (IANA) [91], as root, to regional Internet registries (RIRs), and all the way down to Internet Service Providers (ISPs). However, in the case of RPKI, the resources are accompanied by X.509 certificates to form a chain of trust from top to bottom as illustrated in Fig. 4.1.

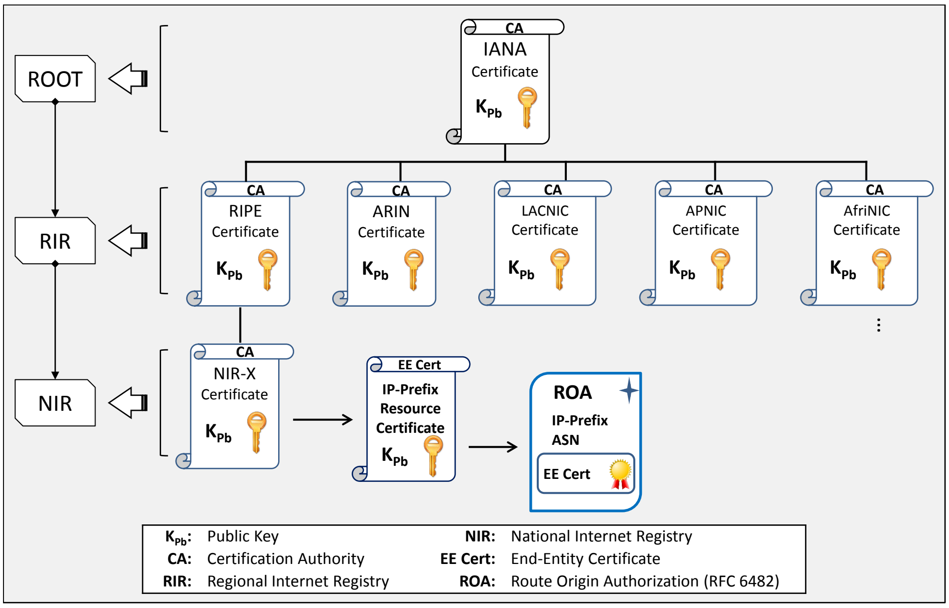


Figure 4.1: Administrative resource allocation hierarchy.

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, i.e., IP address or AS number. A Certification Authority (CA) corresponds to an entity that can further sub-allocate resources and delegate authorities using resource certificates. A CA uses a resource certificate called, "Certification Authority Certificate" (or CA certificate) to sub-allocate resources. Figure 4.1 gives an overview of the CA certificate hierarchy, e.g., IANA will have a CA certificate associated to each of the 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 ROA includes an EE certificate which enables its cryptographic verification, as shown in Fig. 4.1. These certificates and authorities 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 pillars of the SIDR proposals, i.e., ROA and BGPSEC, utilize these certificates to offer security to the exchanged information, such that the receiving party could verify the presented credentials with the help of the RPKI. Therefore, both ROA and the BGPSEC extensively rely on RPKI to achieve their goals, that is, verifying route origin advertisements, and securing route propagation updates, respectively. Each AS can have its own RPKI cache, which should be synchronized and updated regularly with the global RPKI. In fact, the global RPKI does not refer to one huge mother repository, but rather to a collection of distributed repositories which form the 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 as well as best practices RFCs related to RPKI. The RFC 6480 [4] provides detailed description of an infrastructure to support secure Internet routing. The RFC 6481 [92] describes a standard profile for a resource certificate repository structure. A complete list of RFCs can be found in [3].

4.2.2 The Second Pillar: Route 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 an ROA and publish it in the RPKI repository. This signed authority is formatted according to the Cryptographic Message Syntax (CMS) [93], and it binds the IP prefix resource with its owner's ASN by including the corresponding EE certificate inside it (see the bottom right of Fig. 4.1).

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 or not 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, in Fig. 4.2, AS_2 has RPKI presence and BGP peering sessions with AS_1 and AS_3. Let us assume that AS_1 owns IP prefix 10.1.1.0/24, so it creates an ROA using the respective RPKI EE certificate and publishes it in the global RPKI repository. Then, AS_1 advertises the prefix to AS_2. On the other hand, AS_3 tries to advertise the same IP prefix to AS_2 as its originating AS, but it cannot produce a valid ROA from any administrative resource allocation authority as it is not the rightful owner of the prefix 10.1.1.0/24. When AS_2 receives the IP prefix announcement from AS_1, it verifies against the ROA validation filters extracted from RPKI for origin validation. The existence of a valid ROA for the respective prefix from AS_1 in the RPKI not only assures AS_2 the integrity of AS_1's announcement but also assures AS_2 that AS_3's prefix announcement is false. Now AS_2 has sufficient information to make a decision according to its internal policy.

As shown in Fig. 4.2, a new protocol called Router-RPKI (Rtr-RPKI) [94], allows routers to

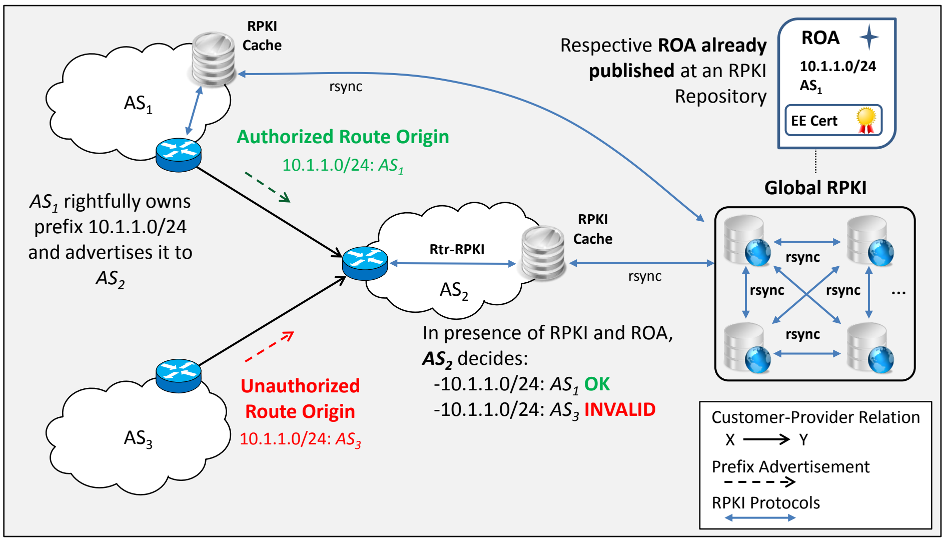

RPKI Cache
AS1
rsync
Respective ROA already published at an RPKI Repository
ROA
10.1.1.0/24
AS1
EE Cert
Authorized Route Origin
10.1.1.0/24: AS1
AS1 rightfully owns prefix 10.1.1.0/24 and advertises it to AS2
Global RPKI
RPKI Cache
Rtr-RPKI
AS2
rsync
rsync
rsync
rsync
rsync
...
In presence of RPKI and ROA, AS2 decides:
-10.1.1.0/24: AS1 OK
-10.1.1.0/24: AS3 INVALID
Unauthorized Route Origin
10.1.1.0/24: AS3
AS3
Customer-Provider Relation
X → Y
Prefix Advertisement
RPKI Protocols

Figure 4.2: Mitigating false IP-prefix origin advertisement using ROA.

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 (see Fig. 4.2). 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 6.6, the advent of Software Defined Networking (SDN) [95] 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 RFC 6483 [96].

4.2.3 The Third Pillar: Securing Route Propagation (BGPSEC)

As mentioned earlier, BGP neighbors exchange BGP updates to propagate reachability information. These BGP updates contain advertised and withdrawn routes along with their attributes. The false AS-Path vulnerability stems from the lack of verification of the authenticity of the *AS_Path* attribute of the advertised route. Essentially, securing route propagation refers to securing the *AS_Path* attribute of a particular route. The BGPSEC protocol [6] provides such mechanism, based on public key cryptography to secure the AS-Path information of an advertised route. Even though the BGPSEC protocol requires changes in the way BGP operates along with the requirement of a new BGP attribute, called *BGPSEC_Path*, it is backward compatible with the BGP-4 protocol. Furthermore, the BGPSEC protocol is only recommended for securing inter-domain routing and not intra-domain routing (i.e., routing with in the AS), implying that the BGPSEC protocol is to be practiced only on the edge routers—the eBGP routers—between different ASes. Figure 4.3 illustrates the origination and propagation of a secure BGP update from AS_N to AS_{N+1} according to the BGPSEC protocol. AS_N uses digital signatures to sign particular information (discussed later in the section) to secure the AS-Path information. The inclusion of the next-hop AS number in the signature (AS_{N+1} in this case), not only enables backward traceability all the way to the origin of the route, but also secures the forward direction of the update, hence the process shown in Fig. 4.3 is known as "Forward Signing".

In BGPSEC, a BGP speaker, now called BGPSEC speaker, has additional BGPSEC router certificates, which are essentially EE certificates. These certificates along with a pair of cryptographic keys allow the BGPSEC router to sign BGP updates on behalf of its AS. Before exchanging BGPSEC updates, the two BGP speakers have to show support for sending and receiving BGPSEC updates in their BGP Open message at the time of initiating the BGP session. After negotiating the BGPSEC capability and related security credentials with a BGP peer, a BGP speaker can originate a secure IP-prefix advertisement, i.e., a BGPSEC update, toward it. It is worth mentioning that the corresponding ROA of the IP prefix to be advertised must have already been published in the RPKI, as it is necessary for successful validation of a BGPSEC update. The BGPSEC speaker originating the BGPSEC update will have at least one Signature block and only one Signature-Segment in the Signature block, that is of its own. As shown

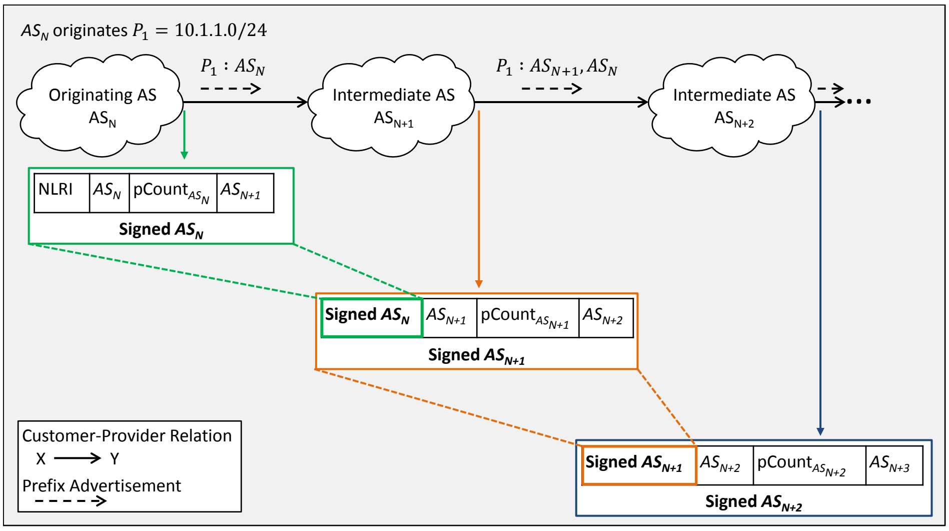


Figure 4.3: BGPSEC origination and propagation with “Forward Signing”.

in Fig. 4.4, the Signature block contains the list of all the Signature-Segments corresponding to all the ASes in the AS-Path. The BGPSEC speaker originating the BGPSEC update will construct a Secure_Path segment. The Signature-Segment is created by first constructing a *Secure_Path*. The *Secure_Path* contains of one Secure_Path segment for each AS in the path to the IP prefix specified in the update. The Secure_Path segment consists of the ASN, the pCount field and the Flags. ASN indicates the AS number of the BGPSEC speaker adding the particular Secure_Path segment in the *Secure_Path*. The pCount field is the prepend count referring to the number of repetitions of the associated ASN that the signature covers. Note that the pCount field enables a secure and optimized way of performing AS-Path prepending. Given that a BGPSEC update message does not contain an *AS_Path* attribute—but instead it has an *BGPSEC_Path* attribute—the *Secure_Path* enables backward compatibility with BGP-4, and assists in converting the *BGPSEC_Path* attribute into a BGP-4 *AS_Path* attribute whenever necessary. As shown in Fig 4.3, the Signature field of the BGPSEC updated is obtained by first concatenating a sequence of information including Target ASN, Secure_Path segment, and NLRI. The Target ASN is the AS number of the next-hop BGPSEC speaker. The NLRI corresponds to the IP-prefix being advertised. This concatenated bundle is fed to a digest algorithm, negotiated earlier, to obtain a fixed length digest value which is further fed to a signature algorithm to obtain the Signature-Segment. When a BGPSEC speaker receives a BGPSEC update, it verifies the update using a validation procedure [6]. For validation of a received BGPSEC update, a BGPSEC speaker relies on the ROA and the RPKI. The SIDR’s recommendations leave it to the discretion of ASes for interpreting the outcome of the BGPSEC update validation process according to their internal policies. This implies that ASes have the freedom to prioritize a BGP-4 update over a valid BGPSEC update for a particular IP prefix, or vice versa to satisfy their internal policies. Moreover, a BGPSEC speaker can further propagate a received BGPSEC update either as a BGPSEC update or a BGP-4 update. These

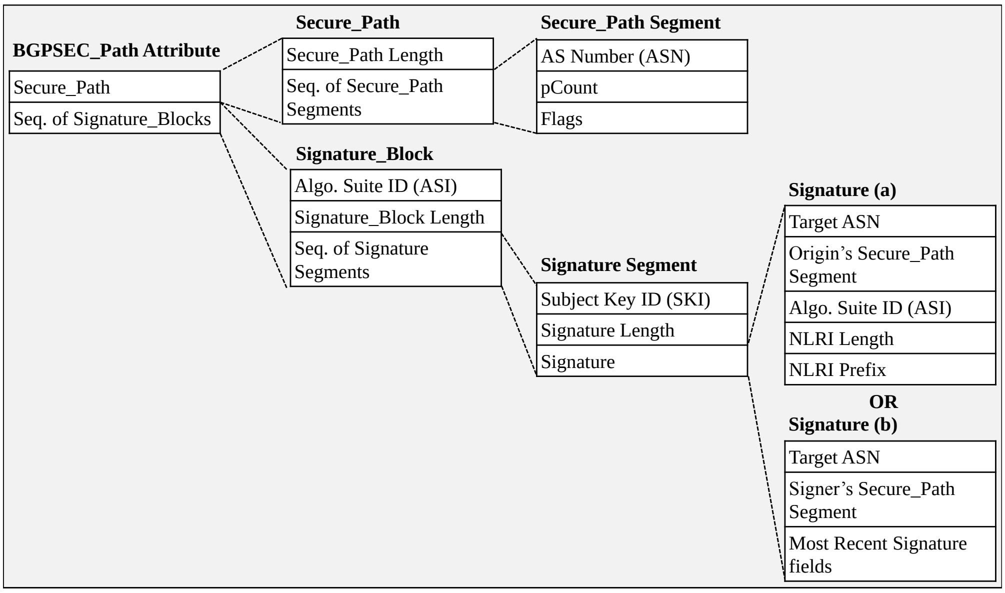


Figure 4.4: Logical structure of the *BGPSEC_Path* Attribute and its components.

flexibilities facilitate partial deployment scenarios. Now, to propagate as a BGPSEC update, the intermediate BGPSEC speaker creates a new BGPSEC update for the same IP prefix. The *BGPSEC_Path* attribute of the new update includes the received Signature-Segments along with the new Signature-Segment of the BGPSEC-speaker creating the update, prepended in front. The Signature field of the new Signature-Segment consists of the Target AS Number, the Secure_Path segment, the Flags and the received Signature fields. Figure 4.3 details the propagation of a BGPSEC update for an intermediate BGPSEC speaker AS_{N+1} toward AS_{N+2}. Hence, signing of certain portion of the BGP update enables the receiving party to verify the claimed information with the help of the ROA and RPKI.

Practical Considerations of BGPSEC

The introduction of changes in the structural and operational aspects of BGP make 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 signature verifications. However, SIDR's contributions have given explicit considerations to practical aspects of BGPSEC, such as making it backward compatible with the BGP-4 protocol. In this section, we only highlight a couple of practical considerations made by SIDR's contributions, so interested readers are referred to [97] and [98] for further information.

IXP Route Server Transparency: An Internet Exchange Point (IXP) is the place where different ISPs interconnect with each other. The Route Servers (RS) at the IXP provide an easy and efficient way of peering with multiple ASes. Usually, there are two ways an IXP can propagate routes:

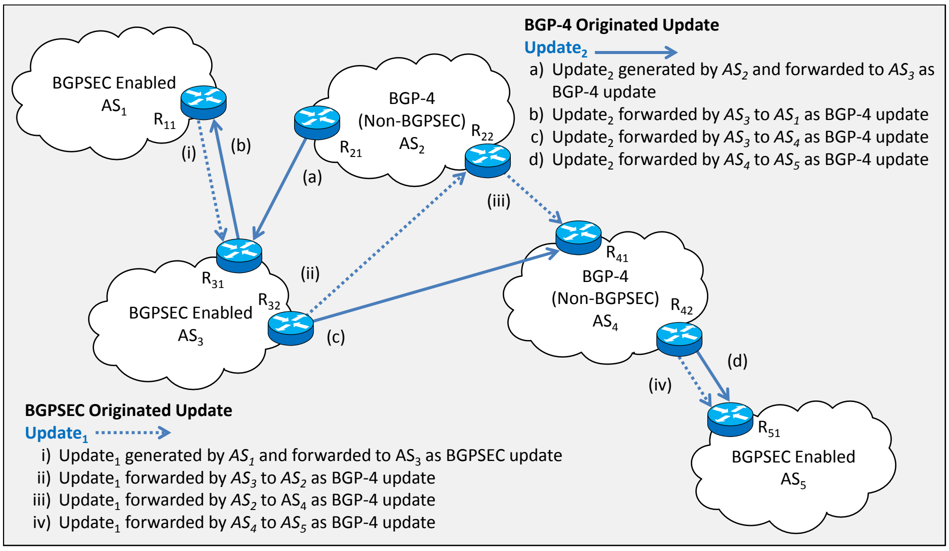
- Direct Bilateral (DB) peering through the IXP.
- Multi-Lateral (ML) peering between clients via a route server at the IXP.

On one hand, the DB peering enables more control over the selection of specific networks to peer with, by allowing to directly establish BGP sessions with the other network at the IXP, but it requires more effort and configuration to peer with all the IXP members separately. On the other hand, the ML peering eases the configuration by just establishing a single BGP session with a route server at the IXP, which is connected to all the other networks, but at the cost of limited control over selection of networks to peer with.

The IXP-RS are said to be "transparent" if they do not include their own ASN in the *AS_PATH* attribute while peering between the clients. The DB peering works unaffectedly with BGPSEC, but ML peering cannot remain completely transparent, as the sender requires the destination ASN in order to forward sign the update. This either requires the client to know in advance the ASN of all the other clients to whom it wants to peer with, or forward sign it toward the IXP-RS ASN and then the IXP-RS forward signs it further to other clients. In the latter case, an IXP-RS can partially remain transparent by putting a '0' in the pCount field when it signs. This way, IXP route servers will show up in the AS-Path but will not contribute to increasing the AS-Path length, as it is computed by summing up the pCounts in BGPSEC. In a way, BGPSEC makes the IXP route servers semi-transparent.

Partial Deployment Scenarios: The BGPSEC protocol explicitly tackles partial deployment scenarios and this may be one of the main reasons why the SIDR's proposals have a chance of adoption into the real world. Partial deployment refers to the scenario where the Internet consists of interconnected islands of BGPSEC and BGP-4 ASes.

- BGPSEC – non-BGPSEC
  - When an update originating from a BGPSEC enabled AS enters a non-BGPSEC AS, then all the signatures are stripped off. The BGPSEC update is converted into a BGP-4 update using the information in the *BGPSEC_Path* attribute (Fig. 4.5 Step (ii)).
  - A BGPSEC born update may end up as a BGP-4 update.
- Non-BGPSEC – BGPSEC
  - When a BGP-4 update is received by a BGPSEC enabled AS, then it stays a BGP-4 update. The BGPSEC enabled AS does not apply signatures to it, even if forwarding it to another BGPSEC enabled AS (Fig. 4.5 Steps (a) and (b)).
  - A BGP-4 born update stays a BGP-4 update.



BGP-4 Originated Update

$Update_2$

a) $Update_2$ generated by AS_2 and forwarded to AS_3 as BGP-4 update
b) $Update_2$ forwarded by AS_3 to AS_1 as BGP-4 update
c) $Update_2$ forwarded by AS_3 to AS_4 as BGP-4 update
d) $Update_2$ forwarded by AS_4 to AS_5 as BGP-4 update

BGPSEC Originated Update

$Update_1$

i) $Update_1$ generated by AS_1 and forwarded to AS_3 as BGPSEC update
ii) $Update_1$ forwarded by AS_3 to AS_2 as BGP-4 update
iii) $Update_1$ forwarded by AS_2 to AS_4 as BGP-4 update
iv) $Update_1$ forwarded by AS_4 to AS_5 as BGP-4 update

Figure 4.5: BGPSEC updates in partial deployment scenarios.

- BGPSEC – Non-BGPSEC – BGPSEC
  - In this case, a BGPSEC born update, gets converted into a BGP-4 update when it enters the non-BGPSEC AS. And once converted into a BGP-4 update it stays a BGP-4 update no matter wherever it is forwarded. So, if the BGP-4 converted update moves into a BGPSEC-enabled AS, it stays unsigned (Fig. 4.5 Steps (ii), (iii) and (iv)).

The BGPSEC protocol does not allow partial AS-Path information protection, therefore BGPSEC cannot be tunneled through non-BGPSEC (BGP-4) ASes. Thus, when an update goes from a BGPSEC enabled AS to a non-BGPSEC AS, the signatures of the BGPSEC update have to be stripped off, and the BGP-4 *AS_Path* attribute has to be constructed using the Secure_Path information available in the *BGPSEC_Path attribute*, since the BGPSEC protocol is backward compatible with the BGP-4 protocol. Once a BGPSEC update gets converted into a BGP-4 update, it cannot be reversed back into a BGPSEC update, even if it enters again a BGPSEC enabled AS. This is an important compromise which makes the BGPSEC protocol impotent in the presence of BGP-4 islands, though it is crucial to accommodate partial deployments.

On the boundary of a BGPSEC island, the *Secure_Path* is converted in to a BGP-4 *AS_Path* attribute. Any prepended ASN, that was collapsed in BGPSEC, will be repeated pCount number of times and any transparent route server, with pCount equal to zero, will be removed from the BGP-4 *AS_Path* attribute. Figure 4.5 illustrates different scenarios that may occur in the life of a BGPSEC update and BGP-4 update in case of partial deployment of the BGPSEC protocol.

In addition, SIDR's recommendations also allow two BGPSEC enabled ASes to negotiate an asymmetric BGPSEC communication, called simplex BGPSEC. In simplex BGPSEC, a stub AS sends BGPSEC updates and receives BGP-4 updates from its provider through a mutual trust agreement. This lifts up the burden of BGPSEC update validation from stub ASes, which constitute around 80% of the ASes in the Internet [99], encouraging early adoption for resource constrained stub ASes, as they would experience less pressure to upgrade hardware equipment.

4.3 Analysis of SIDR's Proposals

The SIDR's proposals allow to secure the prefix origination and AS-Path information in BGP by means of a global security infrastructure that relies on the PKI paradigm. The RPKI, as the global security infrastructure, enables the deployment of ROA and BGPSEC to provide a wider security blanket for inter-domain routing. However, RPKI, ROA and BGPSEC introduce extra burdens that must be taken into serious consideration. Indeed, SIDR's proposals require to fulfill certain characteristics to assure effective security, which are not only challenging but also represent huge barriers for wider acceptance. As a result, there is skepticism about SIDR's solutions and a considerable reluctance among the key actors that need to lead and push for a wide scale deployment.

In this section, we analyze the SIDR's proposals from three different perspectives, including, security analysis, deployment analysis, and adoption challenges. In the security analysis, we examine the well-known BGP attacks in presence of SIDR's solutions, and illustrate the ones that still persist while highlighting new born attacks. In the deployment analysis, we investigate the impediments faced by the three pillars proposed by SIDR in terms of practical real world implementation. That is, we analyze the size and synchronization requirements of the global RPKI repositories, as well as the adjunct issues of maintaining them scalable in a distributed manner. We also analyze the extra burdens that BGPSEC adds to the whole security solution, which basically lie on the signature generation and verification requirements as well as on the impact on router resources. Finally, we provide insight into different challenges and strategies currently being discussed to foster early adoption and gradual deployment, which aim at accelerating the acceptance of SIDR's solutions by the Internet community.

4.3.1 Security Analysis of SIDR's Solutions

The security criteria for inter-domain routing considered by the SIDR WG included empowerment of ASes to mitigate false prefix origination and false AS-Path advertisement attacks. From this perspective, SIDR's solutions technically guarantee to achieve their target with the help of the proposed security infrastructure consisting of RPKI, ROA and BGPSEC. As described in Section 4.2.2, the availability of RPKI and valid ROAs assures mitigation of false route origination attacks completely. The false AS-Path advertisement attack is not feasible anymore with the use of BGPSEC updates due to the chained signatures and verifications, as explained in Section 4.2.3. The AS-Path shortening also fails in face of BGPSEC, as the pCount field is part of the signature, and altering the signature will result in the update being dropped during the validation process. Furthermore, the SIDR's solutions also fulfill the requirement of no policy disclosure and AS-Path integrity traceable back to the origin of the announcement. However, out of the three major problems described in Chapter 3.1, namely false AS-Path advertisement, false route origination and route leaks, the latter cannot be countered by the solutions proposed by SIDR. Even though with an extensive security infrastructure, there are other BGP vulnerabilities, such as replay and coordinated attacks, that also remain unattended by the SIDR's proposals. Moreover, a new set of attacks can be envisioned based on the exploitation of the SIDR's proposals. Thus, we proceed to analyze the security of SIDR's solutions in light of these residual, unattended and new attacks next.

BGP Residual Attacks

Route Leaks: As described in Chapter 3.1.3, route leaks are a routing security problem that occur when business policies are violated. Route leaks can occur even in the presence of RPKI, ROA and BGPSEC, since they exploit the fact that customer routes are preferred over peer or provider routes, and peer routes are preferred over provider routes. Figure 4.6 illustrates the Customer route leak scenario described in Chapter 3.1.3 in the presence of the three pillars of SIDR proposal. It can be observed in the figure that even if AS_4 had published an ROA for the IP prefix 10.1.1.0/24 in the RPKI, and all the ASes in the scenario propagated BGPSEC updates, the route leak still succeeds if AS_3 advertises IP prefix 10.1.1.0/24 to AS_1. The ROA and BGPSEC update validation processes will output valid as they are legitimate (see Fig. 4.6). The peer route leak will also succeed in a similar manner. This is because the RPKI, ROA and BGPSEC secure the operations of BGP and not the business policies among the ASes. The SIDR WG views route leaks as a routing security problem, and identifies them as unresolved despite RPKI, ROA and BGPSEC [100]. Furthermore, SIDR WG has formally requested the Global Routing Operations WG [101] to define the route leak problem before attempting to address it. The recent idea of using route leak protection bits inside the BGPSEC signature segment, put forward by the GROW WG [102], is also infeasible for two main reasons. Firstly, it requires BGPSEC as a prerequisite which itself is facing resistance because of the syntactical and operational changes it incurs on the BGP protocol, and secondly, the RLP solution reveals AS policies more than what BGP already does (cf. Chapter 5.2).

Other Unattended Attacks

Replay Attacks: Replay attacks refer to malicious re-advertisement of withdrawn routes through BGPSEC updates. That is, a withdrawn route is replayed exploiting the fact that the associated certificates are still valid. In the worst case, the replay attack remains feasible until the expiry time associated with the EE certificate of the router that originated the route advertisement in the first place. The replay attack window, i.e., the time interval during which a withdrawn route can be re-advertised, can possibly be on the order of several days depending on the validity of EE certificate of the originating AS router. According to the BGPSEC protocol, the withdrawals are not signed, as it assumes transport security between any two neighbor-

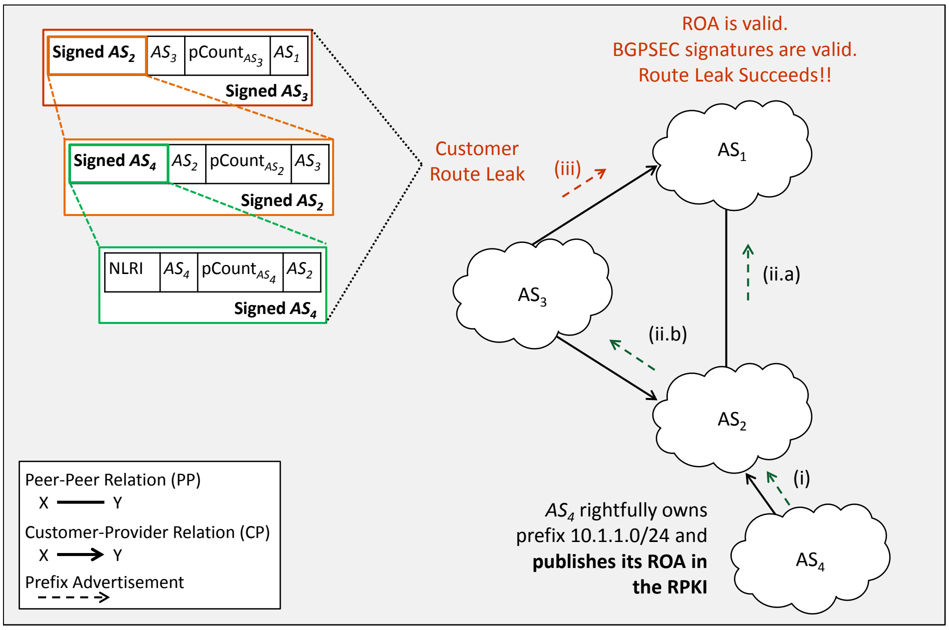

Signed AS_2 | AS_3 | pCount$_{AS_3}$ | AS_1
Signed AS_3
Signed AS_4 | AS_2 | pCount$_{AS_2}$ | AS_3
Signed AS_2
NLRI | AS_4 | pCount$_{AS_4}$ | AS_2
Signed AS_4
ROA is valid.
BGPSEC signatures are valid.
Route Leak Succeeds!!
Customer Route Leak
(iii)
AS_1
AS_3
(ii.a)
(ii.b)
AS_2
(i)
AS_4
AS_4 rightfully owns prefix 10.1.1.0/24 and publishes its ROA in the RPKI
Peer-Peer Relation (PP)
X —— Y
Customer-Provider Relation (CP)
X ⟶ Y
Prefix Advertisement
- - - ->

Figure 4.6: Route leak on a customer link in presence of RPKI, ROA and BGPSEC.

ing BGPSEC routers, which will mitigate injection of false withdrawals or replaying of stale withdrawals by an alien entity. However, this assumption fails to stop the re-advertisement of withdrawn routes by neighbor ASes within the expiration time window of the EE certificates.

One proposed way to mitigate replay attacks is by explicitly limiting the life of a route advertisement with an expiry time field inside a BGPSEC update [103]. The route originating AS includes the expire time field in its signature, whereas the ASes along the AS-Path need not to include expire time fields in their respective signatures. This solution requires a regular beaconing mechanism for refreshing the routes, i.e., the originating AS re-originates the route with new expire time to extend the life of the route propagated earlier. The route should be re-originated after a certain time interval such that all ASes in the AS-Path will receive the re-originated route with the extended expiry time before the current one expires [103]. The duration of the time interval for the re-origination of a route is an important parameter, since large time intervals will cause less BGPSEC churn, but will lead to large replay attack windows; whereas short time intervals will minimize the replay attack windows, but at the cost of more BGPSEC chattiness. Another proposed mechanism that can be used to counter replay attacks is the BGPSEC router key rollover [104]. It describes the process of replacing a router's key pairs along with a new EE certificate, hence renewing the life span of a route advertised through a BGPSEC update. It does not suffer from beaconing burden; however, it adds administrative burden of frequent rollovers in order to have a reduced replay attack window. As BGPSEC is an ongoing effort, there is no concrete indication at the moment about which mechanism will be used to provide protection against replay attacks.

Route Withdrawal Starvation: Route withdrawal starvation refers to the scenario when an AS suppresses a withdrawal update, i.e., it does not forward the withdrawal update to other ASes, to whom it had advertised this route or set of routes earlier. If the malicious AS keeps forwarding the traffic on the old route, then the traffic could be dropped further up the path and hence detected by ASes which were deprived of the withdrawal information. On the other hand, if the malicious AS forwards the traffic toward the destination through some other path, then this may not only result in sub-optimal routing but traffic hijacking as well. In the latter case, the malicious AS would be able to sniff all the traffic undetected. The transport security assumption between two neighboring BGPSEC routers works fine for shielding off external false withdrawal injections, but does not counter the potential suppression of withdrawal updates by a neighbor AS. Even though if the withdrawals are signed, route withdrawal starvation is still feasible as it occurs due to the withholding of withdrawal information rather than exploiting the semantics or operations of BGP. The main difference between route withdrawal starvation and replay attacks is that in the latter case an AS re-advertises routes that it withdrew earlier—hence exploiting the fact that the certificate associated with a route is still valid—whereas in the former case, an AS suppresses the propagation of withdrawal information. The mitigation of route withdrawal starvation is similar to replay attacks, i.e., either expire time or router key rollover mechanisms can be used to counter it.

Co-ordinated False AS-Path Attack: The BGPSEC protocol secures AS-Path information in BGP updates with the use of chained signatures and verifications, but it still falls short of countering coordinated attacks on the AS-Path information. Figure 4.7 illustrates a naive example where two ASes coordinate to propagate false AS-Path, even when BGPSEC is deployed, which can result in traffic hijacking. In this example, we assume that all the ASes employ BGPSEC, i.e., each AS signs and verifies every BGP update as described in the BGPSEC protocol. As shown in Fig. 4.7, AS_1 advertises its IP prefix P_1 to its customers, AS_2 and AS_3. According to BGPSEC, AS_1 inserts AS_2 in the target ASN field for the BGPSEC update toward AS_2, and places AS_3 in the target ASN field for the BGPSEC update toward AS_3. AS_3 does the same for AS_4 and so forth, such that the IP prefix advertisement of P_1 received by AS_7 from AS_5 contains the AS-Path AS_5—AS_4—AS_3—AS_1. If AS_2 creates a BGPSEC update of IP prefix P_1 by inserting AS_6 in the target ASN field, and then tunnels this update directly to AS_6, then it technically enables AS_6 to advertise P_1 toward AS_7 through AS_2. The BGPSEC update from AS_6 to AS_7 containing the non-existent AS-Path AS_6—AS_2—AS_1 will pass all the validation checks proposed by SIDR, including ROA validation and the verification of BGPSEC signatures along the AS-Path. Now, AS_7 has two routes for P_1, and by virtue of the shortest AS-Path preference, it will opt for the route through AS_6. AS_6 can either black-hole the traffic or undetectably sniff the traffic while sub-optimally routing it toward AS_1 through AS_5. Such attacks require a bit more sophistication and coordination between two ASes, and thus are harder to mitigate. This example clearly highlights the existence of security holes that can be exploited even after the deployment and adoption of RPKI, ROA and BGPSEC.

RFD and MRAI Attacks: The Route Flap Damping (RFD) and Minimum Route Advertisement

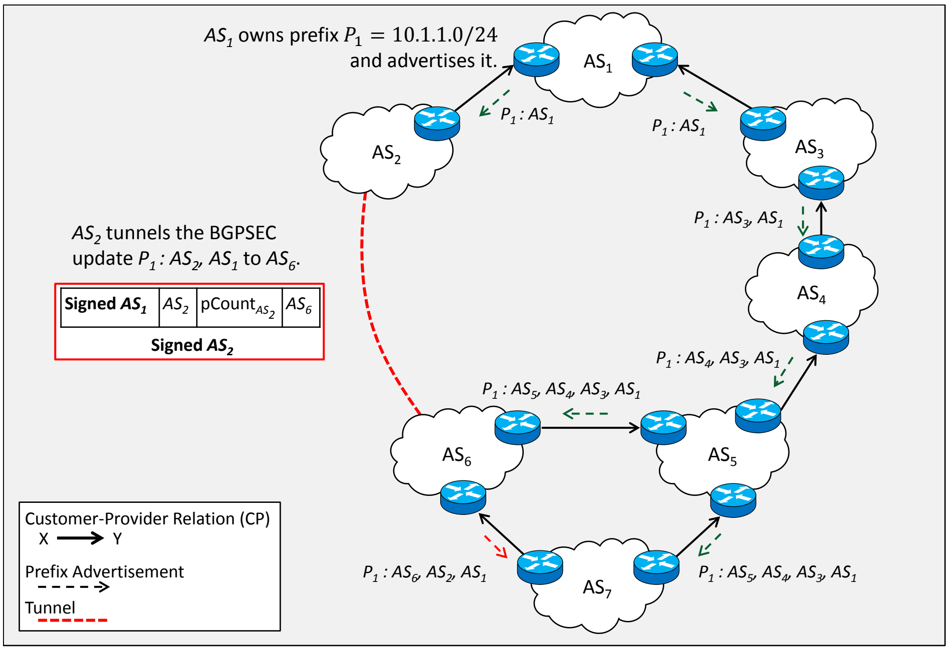


Figure 4.7: Co-ordinated false AS-Path attack on BGPSEC.

Interval (MRAI) attacks misuse the mechanisms that BGP employs to maintain route stability and to assure convergence, respectively. The RFD mechanism measures instabilities of routes, based on how frequently they are advertised and withdrawn, and blocks a route when it is unstable beyond a certain cut-off threshold. Hence, an on-the-path malicious AS can cause a victim AS to block a certain route to a destination by frequently advertising and withdrawing it. The MRAI timer puts a limit on how frequent route advertisements and withdrawals can be send to a neighbor AS. By exploiting the application of MRAI on the withdrawal sent to a neighbor, an on-the-path malicious AS can intelligently sequence advertisements and withdrawals of a route toward a victim AS such that the destination remains unreachable for the victim AS [82]. The SIDR proposals do not mitigate RFD and MRAI attacks as they are intrinsic to BGP stability and convergence mechanism but question the motives of such attacks, as the malicious AS will cause loss of revenue for itself and unreachability to a certain destination for the victim AS.

New Born Attacks

BGPSEC Functionality Downgrade: It refers to the scenario when a BGPSEC capable AS deliberately downgrades itself to BGP-4 to avoid signatures and verifications in order to launch an attack, i.e, a BGPSEC capable AS sends unsigned updates (i.e., BGP-4 updates) when it is capable of sending signed updates (i.e., BGPSEC updates). This attack succeeds by exploiting the very flexibility in SIDR's recommendations that allows a BGPSEC speaker to peer with a BGP-4 speaker for tackling partial deployment scenarios (see Section 4.2.3).

For example, let us consider the false AS-Path attack described in Chapter 3.1.1 in the context of a BGPSEC downgrade scenario (see Fig. 3.1). If we assume that the ASes along the legitimate BGP update path (i.e., AS_3–AS_4–AS_5) use the BGPSEC protocol, and AS_1 deliberately downgrades itself to BGP-4, then AS_2 receives a valid BGPSEC update and a BGP-4 update (with a false AS-Path) for the same IP prefix 10.1.1.0/24. In this scenario, the fate of the attack launched by AS_1 depends on the internal policy of AS_2, i.e, if it prefers valid BGPSEC updates over an unverifiable BGP-4 update, then the attack fails; but, if it treats them equally, or it prefers a shortest path over a valid BGPSEC update—even though it comes from an unverifiable BGP-4 update—then the attack succeeds. Furthermore, if either AS_3 or AS_4 are not BGPSEC enabled or either of them is colluding with AS_1 and downgrades to BGP-4, then the BGPSEC update will be converted into a BGP-4 update, and the attack will succeed anyway. The assumption of a BGPSEC speaker preferring a shortest path coming from an unverifiable BGP-4 update over a valid BGPSEC update is indeed rational, since a recent survey [77] among large ISPs shows that 40% of the respondents indicated that they will place the BGPSEC update information below the shortest path tie breaker in the BGP route selection algorithm (Section 9 in [1]). The results presented in this survey clearly put a question mark on the effectiveness of BGPSEC even if it is adopted (cf. Section 4.3.3).

The downgrade to BGP-4 is a possibility due to the need of partial deployment scenarios, and learning from the experience of IPv6 deployments and its adoption, the partial deployment scenarios for BGPSEC will be a reality for a long period of time.

Deviant RPKI Authorities Attack: These attacks originate when the trust among the RPKI authorities is violated. The abnormal behavior of RPKI authorities can be due to misconfiguration, malfunctioning of equipment or on the request of official authorities. A deviant RPKI authority can revoke a set of resource certificates under its administration causing several legitimate ROAs and AS-Paths to become invalid. Cooper et al. [105] argue that RPKI authorities enjoy unchecked power to revoke or overwrite resource certificates and ROAs, hence making it difficult to distinguish between abusive and normal revocations. They also highlight that in case of cross-country certification, a deviant RPKI authority can avoid any legal repercussions if it targets resource certificates and ROAs outside its legal jurisdiction. Such grave drawbacks due to compromised RPKI authority are few of the many open problems faced by SIDR's solutions.

4.3.2 Deployment Analysis of the Three Pillars of SIDR's Proposals

In this section, we examine the viability of SIDR's solutions by analyzing how the academic and industrial communities have perceived the proposals developed by SIDR. In particular, we survey the practical and theoretical studies related to the workability of the RPKI, ROA, and BGPSEC, and highlight the disagreements and challenges for their realization in the real world. For RPKI, being a distributive hierarchical repository infrastructure, we focus on the total size of the global repository and the synchronization delays among the RPKI repositories, as key gauging metrics for its successful deployment. As ROAs are signed security objects hosted in the RPKI repositories, it is appropriate to analyze their deployment impact on the security system along with the RPKI. For BGPSEC, we consider the processor and memory requirements it incurs along with the changes needed in the BGP protocol (i.e., on the software part) for a feasible deployment.

Deployment Analysis of the Security Infrastructure

The proposed security infrastructure, RPKI, is devised to support the validation of claims related to Internet Number Resources (INR) holdings (e.g., IP prefixes or AS numbers). As described in Section 4.2.1, the global RPKI consists of a set of hierarchical and publicly available repositories, each one governed by a Certification Authority (CA) Organization. In our discussion, we address the following two important issues:

- Estimation of the total number of security objects, such as certificates and ROAs, that are going to be published and stored in the distributed repositories once the solutions are completely deployed.
- Estimation of synchronization delays of security objects in terms of the global RPKI repository.

Estimation of the Global RPKI Size: Osterweil et al. [106], Kent et al. [107], and Bruijnzeels et al. [108], present different studies and scalability analysis by providing estimations about the global RPKI size, and the synchronization delays associated with the distributed repository system. Whilst these studies consider a global deployment of RPKI, ROA and BGPSEC for their estimations, it is important to remark that the different metrics required for the evaluations, such as the total number of RPKI repositories, router EE certificates and ROAs, vary largely among them. There is no general consensus or similar approximations about the total expected size and complexity of the global RPKI, which leads to drastically diverse analyses regarding the scalability and functional requirements of RPKI.

According to Osterweil et al. [106], the total expected number of objects in the global RPKI is estimated to be around 650,796 excluding the BGPSEC router certificates, i.e., considering only RPKI and ROA deployments, and around 2,650,836 in the case of complete BGPSEC deployment. This study assumes that there are 42,000 ASes in the Internet and in the worst case, every AS's CA has its own repository, hence 42,000 RPKI repositories. The number of origin attestation objects, i.e., ROAs, considering more than one prefix per ROA (excluding multi-homing scenarios) is estimated to be 273,592. In this study, the total number of BGPSEC router certificates are estimated to be around 2,000,000 based on the rough estimation that there are around 1 million eBGP routers in the Internet [109] and assuming a pair of BGPSEC router certificates per eBGP router, the total number of BGPSEC router certificates crunch up to 2 million. In contrast, Kent et al. [107] report that the total expected number of objects in the global RPKI would be around 444,645 including BGPSEC router certificates (289,501 without them). Their assumptions contemplate a total of 47,305 publication points (i.e., CAs), including 39,732 which are stub ASes, 7568 non-stub ASes, and 5 RIRs. These publications points map to 7,000 repositories, i.e., each repository can host more than one publication point belonging to different CAs. Kent et al. [107] make two crucial assumptions in their analysis; first, that the stub ASes—which constitute 80% of the Internet—outsource their RPKI repository chores; and second, that all the edge routers (i.e., eBGP routers) of an AS will use only one pair of BGPSEC router certificates per AS for signing. These assumptions drastically reduce the total number of objects in the global RPKI as compared to the estimations provided by Osterweil et al. [106].

Moreover, Bruijnzeels et al. [108] report an estimated total size of a global RPKI with ROA and BGPSEC deployed to be around 3 million objects. Apart from considering a CA per ASes, this study also considers separate CAs and related security objects for Provider Independent resources. According to Bruijnzeels et al. [108], in the worst case, the total number of CAs is around 200,000, each of which will hold 4 objects, which yields around 800,000 CA objects. Furthermore, they assume the total number of ROA objects are dependent on the number of routes in the BGP FIBs in the Default-Free Zone (~500,000), and with an aggregation factor of 3, i.e., 3 prefixes per ROA, the total estimated number of ROAs sums up to 200,000. This study estimates the total number of required BGPSEC router certificates to be around 2 million with the same line of reasoning as Osterweil et al. [106]. We can observe that Bruijnzeels et al. [108] and Osterweil et al. [106] provide relatively similar estimations regarding the total number of objects in the global RPKI, but there is a considerable difference with the estimations of Kent et al. [107] (see Figs. 4.8 and 4.9), given that they are based on different hypotheses.

The discrepancies in identifying the total size of the global RPKI found in these studies reflect that the different actors working on the solutions have different opinions and expectancies about a global deployment of the RPKI. Besides, we observe a conundrum about the total

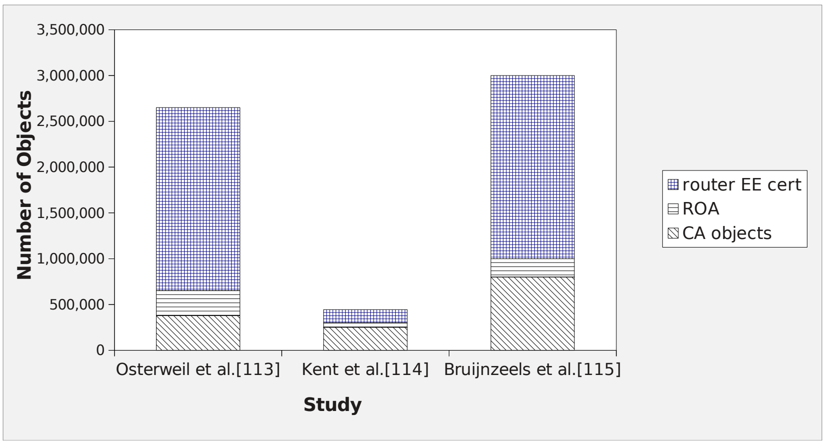


Figure 4.8: Estimated total number of objects in global RPKI.

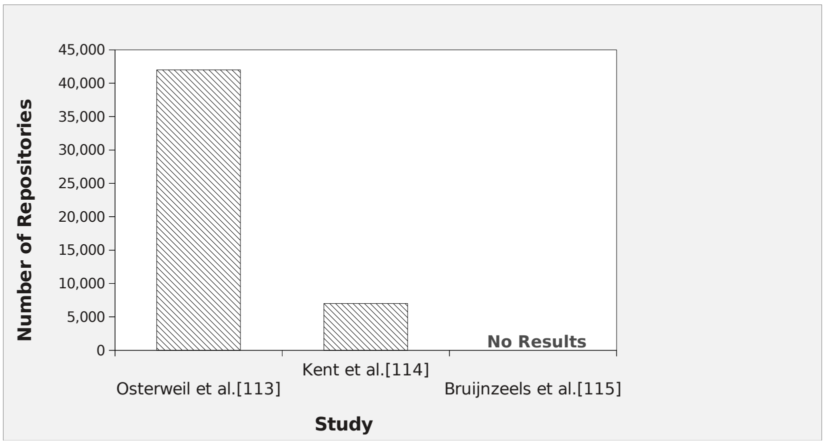


Figure 4.9: Estimated number of repositories in global RPKI.

number of BGPSEC router certificates required in a complete BGPSEC deployment. Osterweil et al. [106] and Bruijnzeels et al. [108] prefer to assume that each BGPSEC router will at least have two BGPSEC router certificates, whereas S. Kent et al. [107] considers only two BGPSEC router certificates per AS, i.e., these two BGPSEC router certificates will be shared among all the AS's eBGP routers. However, sharing a pair of certificates among several routers of an AS adds extra burden of secure distribution of these certificates among those routers. Furthermore, if a shared BGPSEC router certificate gets compromised, then all the AS's routers sharing this certificate become vulnerable to attacks, and the BGPSEC router certificate of each of these routers needs to be replaced. On the other hand, in the case of a pair of BGPSEC router certificates per eBGP router, a compromised BGPSEC router certificate will affect one particular router only. The reason behind two BGPSEC router certificates, either per eBGP router or per AS, is to provide the next BGPSEC router certificate along with the current one, in case it gets revoked, expired or compromised, to save up on the BGPSEC router certificate replacement delays. In our opinion, considering only a pair of BGPSEC router certificates for all the routers of an AS is an over simplification on part of Kent et al. [107]. However, considering a pair of BGPSEC router certificates per router acutely increases the total number of security objects in the global RPKI.

Estimation of Synchronization Delays among RPKI Repositories: Besides the total size of a global RPKI, the synchronization delay among the distributed RPKI repositories is another important metric of interest. Osterweil et al. [106] and Kent et al. [107] contemplate and evaluate the time required for local caches to actually gather a fully deployed global RPKI (i.e., the total synchronization time). They base their analysis and results on measurements taken from different large RPKI repositories to estimate the average number of objects and the average synchronization time. Therefore, an expected synchronization time per object can be derived and used for calculating the total time required for global RPKI synchronization. Bruijnzeels et al. [108] does not evaluate synchronization delays among RPKI repositories in their study.

According to Osterweil et al. [106], 628 ms/object is the average time to synchronize an RPKI object, and based on the estimated number of objects in a fully deployed RPKI, they calculate that with 42,000 repositories, the time required to gather all the objects locally is 5.04 days excluding BGPSEC router certificates, and 19.57 days including them. They conclude that this total synchronization time provides a lower bound on the estimation of the total time required to populate and synchronize a global RPKI, as the number of security objects will increase whenever certificates are revoked, or the amount of objects will double during cryptographic algorithm rollovers. Moreover, this lower bound does not take into consideration the network-latency factor. On the other hand, Kent et al. [107] estimate the total time to synchronize to be around 60 minutes, assuming 7,000 repositories, 445,000 RPKI objects, and an average time of 20 ms to synchronize an RPKI object (see figure 4.10).

The total synchronization time for a global RPKI is an important factor that directly affects the scalability of a fully deployed RPKI. We observe another discrepancy between the two studies about the estimated time required for synchronization. An important aspect that directly affects this total synchronization time of the global RPKI is the synchronization time per object value. Osterweil et al. [106] computed an average value of 628 ms/object, based on measurements taken from 10 different RPKI repositories, and 14,980 RPKI objects on average. On the other hand, Kent et al. [107] uses a measurement of 20 ms/object, based on the traces of a RIPE repository which considers at most 4,900 RPKI objects. Therefore, the total time required for synchronization obtained in these studies presents a considerable variation in the range of tens of minutes to days. This discrepancy shows again that their assumptions and estimations of the total size, and the total synchronization time for a global RPKI are far apart, envisioning two extreme possible outcomes if a fully RPKI deployment is achieved. Besides, in the worst case scenario, we foresee a serious security threat not only due to large time-windows for replay attacks but also due to the wrong validations that would occur because of

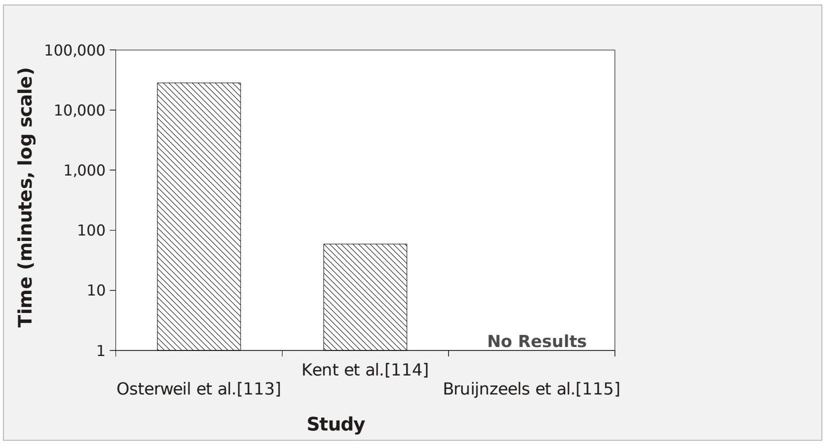


Figure 4.10: Estimated total sync time for a global RPKI.

stale security data. The latter stems from the total time (in order of days) to synchronize the global RPKI, which jeopardizes the security and consistency of the whole system.

Discussion: Approximately 3 million objects in the worst case might be considered to be a large lower-bound estimation. However, it will surely grow whenever any keys are rolled over and changed. Besides, the daily churn, i.e., the amount of new objects that are expected to be created within a period of 24 hours is another factor to consider. These factors directly impact on the scalability and consistency of the distributed repository system. We have to keep in mind that the publication and propagation time of an object in the global repository system is an important aspect that will affect the feasibility of the solution, e.g., fast global visibility of a published ROA object would be required by all the RPs in order to be able to validate a BGPSEC route. Osterweil et al. [106] recommends that if a deadline for global synchronization is considered, the object synchronization rates must be super-linear, i.e., the system must get faster as it gets more loaded. However, Kent et al. [107] project global synchronization delays (as seen by each RP doing incremental fetch) to be in the range of tens of minutes.

The estimations, evaluations and assumptions, presented in Osterweil et al. [106], Kent et al. [107] and Bruijnzeels et al. [108], regarding the security and scalability of the global RPKI provide significant insight about the divergence of views, which questions the feasibility and the potential deployment of the global security infrastructure. Observe that, even though in our discussion we only covered the estimation of a few important parameters of the global RPKI (mainly total size and total synchronization time), the discrepancies highlighted above represent a large difference of opinion among some of the major players in this arena. We believe that further in-depth evaluations and measurements of the proposed RPKI are crucial to remove the mentioned discrepancies in order to pave the way for its future deployment.

Deployment Analysis of BGPSEC

As described in Section 4.2.3, the solution proposed for securing route propagation in inter-domain routing, BGPSEC, is based on a forward signing mechanism. This solution fulfills the minimum security requirements for securing BGP announcements, however it introduces extra burdens which can be broadly classified as hardware and software challenges. The former affects directly the network hardware (e.g., the routers), and includes (1): the processing load due to the generation of signatures and their verifications; and (2) the increment on router's RIB memory in order to accommodate the BGPSEC advertisements. It is important to remark that the BGPSEC route updates are per-prefix, which amplifies these burdens. On the other hand, in terms of the software burden, BGPSEC requires changes on the current BGP protocol, which can be considered as a huge barrier in light of global acceptance. Hereafter, we present an extended discussion on the analysis of these burdens and their possible consequences over the solution's feasibility.

Estimation of Processor Load (hardware requirement): An initial estimation of the impact of BGPSEC over the processing resources in a router is presented by Sriram et al. [99]. The study employs an AMD64 Sandy Bridge Intel i7 3400MHz processor, which requires 2,530 ops/sec for signing and 2,215 ops/sec for verifying for an ECSDA-P256 signature algorithm—a benchmark of the ECSDA-P256 algorithm over the machine was used to obtain these values. The CPU cost estimation model assumes a peer session reset for a large ISP BGPSEC router with a neighbor BGPSEC router—having a customer cone of around 32,000 routes spread over AS-Path lengths between 1 and 8—to estimate the CPU cost for re-validating all the 32,000 routes with varying AS-Path lengths. It is important to note that the validation process of BGPSEC is related to the length of the AS-path, as it defines the number of signature verifications that have to be performed for a particular route update. Sriram et al. [99] estimates 34.59 seconds as the time required to re-validate all the peer routes for this particular case. This result gives insight into how much time a very large ISP router will spend in validating all the routes from a peer if the session is reseted. The result of the extra burden due to signature validation is considerable because 34.59 seconds are required for one peer session reset only at a large ISP. Furthermore, this value does not include the time required for fetching and validating certificates. An option is that the certificate fetching and verification can be done off-line and prior to the validations, hence speeding up the validation process.

Estimation of Required Memory (hardware requirement): As explained in Section 4.2.3, apart from the extra processing that BGPSEC requires, the memory capacity of the routers has to be revised as well, due to the extra information (i.e., the signatures) that a BGPSEC update contains compared with a legacy BGP update. Furthermore, the size of a BGPSEC update increases along with the number of ASes in the AS-path, as each AS must include its signed information. As a result, a BGPSEC update contains all the information required to verify and validate the AS-Path integrity, i.e., a router is able to verify if the update has traversed through all the ASes as claimed in the AS-Path.

Sriram et al. [10] present an initial estimation of the required RIB memory size for a Tier-1's Route Reflector (RR) to accommodate the BGPSEC updates. The study projects the adoption of BGPSEC based on a truncated Normal distribution model. Moreover, the estimation model considers both internal (IGP) and external prefixes (EGP) with an annual prefix growth rate of 15% for external prefixes and 5% for internal prefixes. According to their estimations, with an RSA-2048 signature algorithm, the required RIB size would be 0.51 GB for the year 2016 (anticipating the start of BGPSEC adoption), 8.30 GB in 2020 (anticipating 50% BGPSEC adoption) and 32.11 GB in 2025 (anticipating 100% BGPSEC adoption). In contrast, for the ECDSA-256 signature algorithm, the required RIB memory size is estimated to be 0.42 GB for the year 2016, 3.19 GB for the year 2020 and 11.57 GB for the year 2025. These estimations also imply RIB memory upgrades to accommodate BGPSEC operations. Table 4.1 provides an overview regarding the RIB memory requirements as of today for BGP-4, and the projected RIB memory requirements for BGPSEC for a large ISP RR. The requirement of upgrading hardware in terms of CPU and memory to cope with the proposed solution clearly implies an increase

	Total RIB size (GB)		
Year	BGP-4 (RR)	BGPSEC (RR, RSA-2048)	BGPSEC (RR, ECDSA-256)
2013	0.29	0.30	0.30
2015	0.34	0.35	0.35
2016	0.37	0.51	0.42
2018	0.44	2.23	1.05
2020	0.53	8.30	3.19
2022	0.65	18.06	6.61
2025	0.88	32.11	11.57

Table 4.1: Comparison of RIB size requirements between BGP-4 and BGPSEC (Note: This table is an excerpt from a table presented in [10]).

in CAPEX, which seems difficult to face, since the Return Of Investment (ROI) model as well as the incentives for early adoption are yet in early stages of research.

Accommodation of a New BGP Attribute (software requirement): In addition to the new software components required to implement the protocol to communicate with the RPKI local cache, BGPSEC requires changes in the BGP protocol itself. The most prominent change is the replacement of the *AS_Path* attribute with the *BGPSEC_Path* to facilitate the signature requirements (see Section 4.2.3). Another noticeable requirement is that every BGPSEC speaker must support BGP extended messages, since the size of a BGPSEC update can be large due to the accumulation of the signatures along the AS-Path. Furthermore, in order to consider the result of the BGPSEC update validation process, changes are required in the BGP best route selection algorithm. Even though, the BGPSEC protocol leaves it open for the ASes to accomplish it according to their local policies, the issue of introducing changes to the current BGP protocol could be considered a major barrier for global acceptance.

Discussion: The main deployment challenges faced by the BGPSEC solution include requirements such as router upgrades—both in terms of processing power and memory size—as well as changes to the BGP protocol for accommodating a new BGP attribute. As discussed above, contrary to the opinion of Sriram et al. [99], we argue that with BGPSEC, a session reset could be in the order of hours for a large ISP router, and therefore, BGPSEC will demand router upgrades with noticeable increase in processing power and memory size. The hardware upgrade requirements can be downplayed in presence of strong Return Of Investment (ROI) plans along with financial incentives for early adopters (cf. Section 4.3.3), however, we believe that the most difficult challenge for BGPSEC deployment is not the hardware-related part, but rather the software changes required to the ever resistant BGP protocol.

4.3.3 Global Acceptance and Adoption: Challenges and Strategies

Besides the obvious scalability aspects, the two essential prerequisites for adoption of a security proposal in the inter-domain routing system are protocol backward compatibility and flexible accommodation of partial deployments. As detailed in Section 4.2, the SIDR's proposals fulfill these two conditions, but still this seems to be insufficient. The advantages and trade-offs of BGPSEC as well as of ROA and RPKI, can be rapidly diminished due to the high burden and complexity that they incur on. Although these solutions can be considered technically feasible, their deployment or adoption is dependent on the revenue based incentives they might offer to attract early adopters, as the inter-domain business model is revenue-oriented. Apart from provably increased security offered by RPKI, ROA and BGPSEC, economic benefits are crucial for their success in the practical world. Therefore, it is important to analyze possible strategies regarding how to boost the acceptance and deployment of SIDR's proposals.

Unfortunately, merely securing the inter-domain routing system seems unlikely to provide sufficient incentives for global acceptance. Gill et al. [110] raise the question that the benefits provided by BGPSEC protocol do not become real until a large number of ASes have deployed it. Thus, it proposes a strategy that governments and industry groups can harness ISP's local business objectives and drive a global deployment. Their analysis is focused on three main perspectives which aim to impact the global deployment of the solution. First, on simplex BGPSEC to secure stubs AS. Second, they claim that convincing a small but influential set of ASes to be early adopters of the solutions would boost a faster deployment. And third, ensuring that BGPSEC influences traffic by requiring ASes to break ties between equally-good paths based on security. Similarly, Lychev et al. [111] recommend focusing on the deployment of simplex BGPSEC at stub ASes, incorporation of secure paths in AS policies, and the deployment of BGPSEC at Tier-2 ISPs for the partial deployment period, to pave the way for wide-scale adoption in the future.

We can also extract some pragmatic lessons from the success story of RPKI adoption in Ecuador [112]. The Ecuador Internet Exchange (NAP.EC) holds a unique and critical position in the Ecuadorian nation-wide network, as almost 97% of all Internet users in Ecuador are directly connected to it. Thus, the adoption of RPKI by NAP.EC will cause—or at least it will speed-up—the RPKI adoption by all the other ISPs in the country. This is basically the same strategy recommended by Gill et al. [110], in the sense that RPKI adoption by influential ASes will facilitate faster RPKI technology dissemination. However, rather than forcing the adoption on the smaller players, a consensus was built among all the ISPs, through a series of technical training and information sessions provided by LACNIC and industrial stakeholders, with the aim of securing the routing problems in the country. As a result of these efforts, almost 100% of all the IPv4 addresses allocated in Ecuador have there corresponding ROAs [112]. The main hurdle for the adoption of RPKI faced in Ecuador was the fear of the new technology, as most of the operators had little or no knowledge or experience of RPKI and ROA. This fear was overcome by raising awareness through technical information and training sessions with the help of the local RIR and a network hardware vendor. Despite this, it is not clear that the adoption strategy used in Ecuador for deploying RPKI will yield similar results in other regions, especially, with multiple stakeholders (e.g., multiple IXPs and large ISPs) and thousand fold more Internet users.

Another very important aspect is that, to effectively limit the attacks using BGPSEC, the path validity information should necessarily influence the decision process of selecting a route—the BGPSEC validation output should be considered in the BGP decision process at an appropriate priority. In this regard, the BGPSEC protocol provides flexibility to ASes to prioritize security information according to local policies. Gill et al. [77] provide a survey result with an interesting insight into how ISPs perceive secure AS-Paths over insecure ones. The survey inquired different small to large ISPs to define at what stage of the BGP decision process would they place secure AS-Paths as compared to other metrics. Surprisingly, only 9% of the respondents would prioritize security first, and 21% indicated that they would place secure AS-Paths between the *localpref* and the *shortest AS-Path* metric (recall that the *localpref* is used to prioritize routes based on local criteria). And more importantly, 40% of the respondents would place security considerations at a lower step, i.e., even below the AS-Path length. Mindful of the fact that the respondents included only a fraction of the total ISPs, the results still provide substantial insight in how BGPSEC would be treated even if it is deployed despite all the shortcomings highlighted above. Most of the major known attacks on BGP succeed with the manipulation and exploitation of the *localpref* and the *shortest AS-Path* metrics in the BGP decision process, and if 40% of ISPs are not going to consider BGPSEC validation information then the future for BGPSEC protocol does not seem very promising given the current trends in the Internet community.

5 Route Leaks

The security and reliability of the Border Gateway Protocol (BGP) [1] have been actively investigated since its adoption as the standardized inter-domain routing protocol among Autonomous Systems (ASes) in the Internet. However, the policy related security issues of the BGP have not received the due attention they deserved. Thus, policy related attacks, such as the route leak problem remains unresolved.

We recall that a route leak occurs when an AS advertises a route toward a neighbor AS that does not respect the agreed business relationship between them, as briefly described in Chapter 3. For instance, if a customer AS starts offering transit between two of its providers, then it is a route leak. Similarly, a route leak will occur if an AS advertises routes learned from one provider toward a peer AS. We will delve into the details these aspects in this chapter, but in general terms, a route leak entails a violation of the business relationship that rules the interconnection of domains. The main concern about route leaks is that they are a common occurrence, and regardless if they are due to misconfigurations or deliberate attacks, they can lead to traffic loss, sub-optimal routing, and more importantly, traffic hijacking. Hence, route leaks are capable of causing large-scale disruptions in the Internet. In the rest of this part of the thesis, we thoroughly investigate the route leak problem while proposing and evaluating a set of pragmatic solutions to resolve it.

In this chapter, we formally analyze and develop the route leak problem. We present two real world examples of route leaks. We describe different types of route leaks and explain how, where, and why they occur with the help of example scenarios. This chapter lays the foundation for the development of theoretical framework for the detection of route leaks, presented in next chapter.

5.1 Route Leaks in Real World

Internet service outages by virtue of the BGP shortcomings are frequent [72], but only a few succeed to get mass attention—in practice this typically depends on the scale of the service disruption and the profile of the victims. In this section, we illustrate two major Internet disruption incidents, that we refer to as Telstra-Dodo [8] and Google-Moratel [7]. The apparent causes behind the disruptions point out to incidents that involuntary produced route leaks. More specifically, these incidents were thoroughly analyzed, and the collected evidence boils down to the violation of routing policies between ASes. However, what could not be clarified, is if they were due to intentional (e.g., a traffic hijack attack) or unintentional misconfiguration (e.g., a fat-finger problem) over the export policies of an AS. Despite the traces and evidence left, we found that some service providers involved in these cases claimed that the issues were due to hardware failures, thereby avoiding to mention the possible case of route leaks [113]. Let us describe these two incidents, which we consider clear examples of what route leaks are and their repercussions. A country-level Internet service disruption occurred in Australia on February 23, 2012 [8], which was attributed to malfunctioning of a router. Apparently, one of Dodo's network (AS38285) edge routers exported all its internal routes to one of its providers, namely Telstra (AS1221) (see Fig. 5.1). The internal routes that Dodo advertised or leaked to Telstra included all routes learned from its other providers. These provider-learned routes enclosed all the exported routes of Optus (AS7474), PIPE Internet Exchange (AS23745, AS18398) and the Equinex Exchange (AS24115). Besides, Optus had a peer link with Telstra and, as the latter learned the route to Optus (it's peer) through Dodo (it's customer), it preferred the customer path as "the best path" (i.e., all traffic coming from Telstra toward Optus was routed via Dodo). The reason behind preferring a customer path over a direct peer link is purely economical (cf. Section 5.3). As shown in Fig. 5.1, this route leak incident turned into a snowball effect when Telstra advertised the new set of Dodo-learned routes to its provider, Telstra International (AS4637), which further advertised them to its peers and customers. Eventually, the disruption on the Internet service became visible once Telstra started forwarding large amounts of traffic toward Dodo, which was not equipped to handle the traffic volume. Therefore, the peers and customers of Telstra International also started to

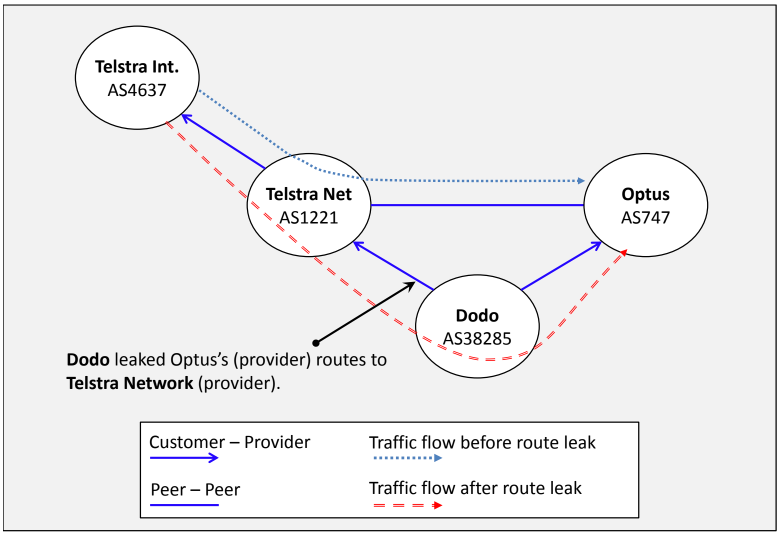

Telstra Int.
AS4637
Telstra Net
AS1221
Optus
AS747
Dodo
AS38285
Dodo leaked Optus's (provider) routes to
Telstra Network (provider).
Customer – Provider
Traffic flow before route leak
Peer – Peer
Traffic flow after route leak

Figure 5.1: Change of traffic flow in case of the Dodo route leak in April 2012.

experience the Internet service disruption. This entire event, illustrated in Fig. 5.1, occurred in less than an hour, causing large scale connectivity problems across Australia.

Another widely noticed Internet outage due to route leaks that directly affected Google's services over some portions of the Internet took place on November of 2012, and lasted for about 27 minutes [7]. In this case, Google (AS15169) experienced routing issues with its peer Moratel (AS23947). Figure 5.2 illustrates the scenario in terms of the traffic path change from the perspective of one of the affected users, CloudFlare (AS13335). They received a route toward Google through an Indonesian service provider Moratel (AS23947). This happened because Moratel exported the routes learned from its peer (Google) toward its provider (BTN-ASN), and Moratel's provider selected the leaked routes and exported them further. CloudFlare's provider, nLayer (AS4436), preferred the route received from its peer (BTN-ASN) over the old route it had toward Google through its provider, Qwest (AS209). Again, the reason behind preferring a peer route over a provider route is economical (cf. Section 5.3). The leaked routes from Moratel propagated and attracted a huge amount of traffic for Google through itself. Moratel network could not cope with such huge traffic load and eventually started dropping traffic. Whilst this problem was figured out and solved, Google's outage was seen from different segments of the Internet.

These incidents clearly expose the inefficacy of the techniques and tools available today for countering route leaks—the main ones will be outlined in next section. In summary, route leaks represent a high risk and challenging problem that requires new approaches and research efforts.

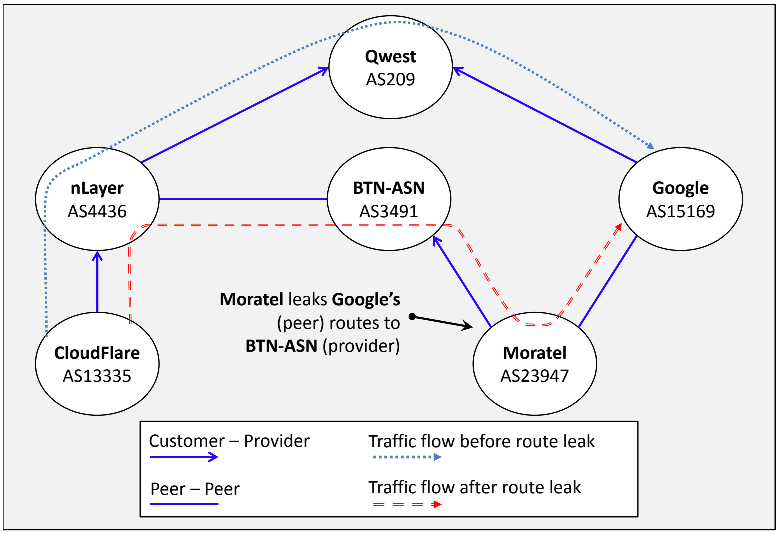


Figure 5.2: Change of traffic flow in case of the Google route leak in November 2012.

5.2 Related Work

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 [45, 66, 67]. 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. [68] 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.

In [114], Goldberg et al. studies the impact of different attacks on inter-domain routing including the export policy violation attack, i.e., route leak and show that it can be used to attract large amount of traffic in the inter-domain. They also contend that soBGP [87] and S-BGP [86] fail to encounter such attacks, however, Goldberg et al. [114] falls short of suggesting any solution to export policy violation attacks.

Sundaresan et al. [115] also investigate the export policy violation attacks in inter-domain routing, calling them *traffic attraction attacks*. In order to counter such attacks, they propose to set a flag in the BGP advertisement when it is sent to a peer AS or a customer AS. For this purpose, they introduce a new *ATTEST* attribute which is appended by all the ASes 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. Furthermore, they recommend to include the *ATTEST* attribute in the signed part of the Secure BGP (S-BGP) message to maintain the integrity of the flags set by each AS in the AS-Path. 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. The experimental results presented in [115] show that, in case of stub route leaks (i.e., when a multi-home AS leaks a route learned from provider to another provider), their solution becomes effective when more than 60% of ASes deploy the scheme. They anticipate that their scheme would perform even worse for other route leak scenarios such as peer route leak. The two main shortcomings of this scheme is that firstly, it requires changes in the BGP protocol to accommodate the new *ATTEST* attribute and secondly it 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, setting and signing the flag in the *ATTEST* attribute discloses policies more than what are revealed by the BGP protocol at present.

It is worth mentioning that the security solutions proposed by the IETF's Secure Inter-Domain Routing (SIDR) Working Group (WG) [3], namely, the Resource Public Key Infrastructure

(RPKI) [4], Route Origin Authorization (ROA) [5], and Secure BGP (BGPSEC) [6] are defenseless against route leaks, as explained in Chapter 4.3.1. This is because route leaks are not covered by SIDR's solutions, since they were not included in the original agenda of the WG. Indeed, the latter has requested the Global Routing Operations WG [101] to define the route leak problem before even attempting to address it.

Recently an idea of using Route Leak Protection (RLP) field inside the BGPSEC signatures to counter route leak problem is under discussion in the GROW WG [102]. The RLP field consists of two bits whose value is set by the AS sending the BGPSEC update to indicate the receiving AS if it is allowed to advertise the routes included in the update to its providers or peers. If the RLP field is set to 00 then the receiving AS can forward the update to its providers or peers and if it is set to 01 then the receiving AS is not allowed to forward the update to its providers or peers. Now, if an AS receives a update from its customer AS such that it observes 01 in the RLP field while unwinding and verifying the signature segments of all the ASes in the AS-Path, then it can consider this update as a route leak. Let us explain the RLP working using the topology in Fig. 5.3. According to solution, AS_4 will put 00 while advertising its IP prefix 10.1.1.0/24 toward its provider AS_2, i.e., it allows AS_2 to further advertise the IP prefix. Now, in step step (ii.b), AS_2 puts 01 while advertising the IP prefix to AS_3, i.e., disallowing AS_3 to advertise the update to its providers and peers. Now, if AS_3 leaks the route to AS_1, then AS_1 can establish it as a route leak as it will observe a 01 in the signature segment added by AS_2. The RLP solution works well for mitigating route leaks, however it suffers from two main adjunct problems. Firstly, the RLP solution will only be effective if everyone is playing

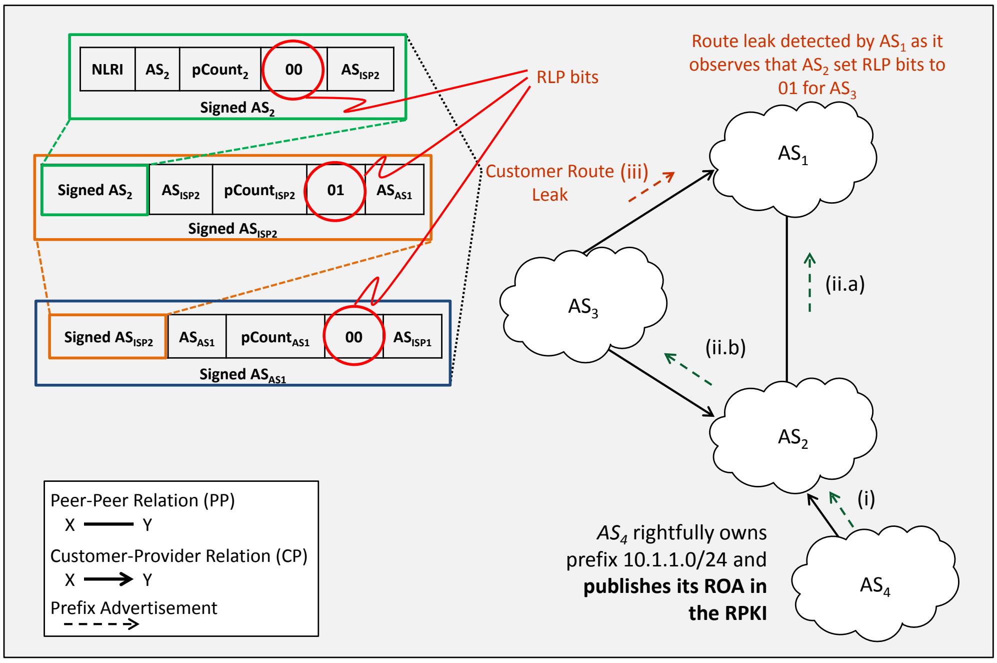


Figure 5.3: Route leak on a customer link in presence of RPKI, ROA and BGPSEC.

BGPSEC. During the partial deployment tenure, the RLP solution can deceived legitimately as BGPSEC allows BGPSEC functionality downgrade (more on this later in the section). Secondly and most importantly, the RLP solution reveals AS policies more than what BGP already does. This is because in the RLP solution, an AS has to explicitly indicate and sign if the next hop is allowed or not allowed to further advertise a particular route. The former problem puts a question mark on the robustness of the RLP solution for mitigating route leaks, however it will be more difficult to convince the industrial players for the latter one, that is to earn relaxation on the confidentiality of the AS policies.

In [85], the author attempts to provide a detection scheme for route leaks using colors along the AS-Path. The scheme suggests to color each AS-hop in the AS-Path according to the corresponding link type, i.e., an AS-hop has color “Green" if toward a provider and has color “Yellow" if toward a peer or customer. In other words, a route received from a customer must have all AS-hops marked “Green" or otherwise it is a route leak. Likewise, a route received from a peer must have all AS-hops marked “Green" except the last AS-hop marked “Yellow" or else its a route leak. In [116], the author contends that such a coloring scheme can be employed in conjunction with BGPSEC by having a signature block similar to the AS-Path signature block. This mode of implementation adds extra burden of signing and verifying the color signature block on the already resource demanding BGPSEC implementation.

Overall, 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 are out of the scope—of the agreed policies. 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. The YouTube incident in 2008 [11], and the Google incident in 2012 [7], could have been avoided if the route filters at the providers were effective. The IRRs provide an online 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.

The BGP monitoring tools, such as Nemecis, Prefix Hijack Alert System (PHAS), Pretty Good BGP (PGBGP) and Argus, 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 burden 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 pre-defined policies, whereas the latter use the data to infer the business relationships and type of peering among ASes.

In [72], the author proposed to detect route leaks by counting the number of predefined “Big Network" ASes in the AS-Path 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. One of the downsides of this technique is that it does not consider the local AS policies or AS neighbor relationship knowledge and thus not only it falls prey to generation of false positives, but also fails to detect route leaks which do not involve "Big Network" ASes in the AS-Path.

5.3 Formalizing Route Leaks

In this section, we formally describe the route leak problem and lay out the foundation for the identification of route leaks. Although, Chapter 2 provides the essential background information regarding inter-domain routing, in this section we define the terminology and the set of policies that rule the routing among ASes in the context of the route leak problem.

5.3.1 Preliminaries

A “provider link” of an AS is a link that connects it to its provider AS. Similarly, the terms “customer link”, “peer link” or “sibling link” refer to a link that connects an AS with a customer AS, a peer AS or a sibling AS, respectively. In this section, we focus on the two dominant AS relationships in the Internet, which are the customer–provider and peer–peer relationships, since the percentage of sibling relations in the Internet is comparatively negligible. However, we discuss the route leak problem in the sibling relationship case in Section 6.5.

Whilst the relationship between two ASes is business oriented which is pragmatically implemented through the BGP protocol. BGP provides complete flexibility for implementing route export or import policies according to the defined relationship, by means of several attributes associated with each advertised route. For example, a provider AS will export all its routes toward its customer ASes in order to attract traffic through its customer links. We are more interested in the export policies, as route leaks occur due to violation of business policies through these exports. The guidelines used for exporting routes (i.e., how to advertise routes depending on the type of relationship with the neighbor AS) are referred to as valley-free rules [45], and they can be summarized as follows:

Rule $\mathscr{R}$.1. *“Routes learned from Customers can be further advertised to other Customers, Peers and Providers.”*

Rule $\mathscr{R}.2$. *"Routes learned from Peers can be further advertised to Customers only."*

Rule $\mathscr{R}.3$. *"Routes learned from Providers can be further advertised to Customers only."*

Therefore, in a customer–provider relationship, the customer AS only advertises its own routes and the routes of its customers cone (*i.e., Customer's Customer routes*) toward its provider AS. A customer cone of an AS is the collection of all ASes that are reachable from an AS following only the provider–customer links. On the other hand, the provider AS advertises all routes toward its customer, hence providing it transit to rest of the Internet. In a peer-peer relation, both ASes only advertise their own or their customer's routes to each other. From the business perspective, the provider AS charges its customer AS for forwarding its traffic to and from it. Whereas in the peer–peer relation, the ASes do not charge each other for exchanging each other's customer traffic up to an agreed threshold.

Consequently, ASes prefer a route received from a customer over a route received from a peer or provider to maximize their revenues. Similarly, ASes prefer a route received from a peer over a route received from a provider for any prefix.

5.3.2 Defining Route Leaks

At present, there is no standard definition of the route leak problem in the Internet community. The working group in charge of securing inter-domain routing, namely, the SIDR WG [3], has delegated the task of defining the route leak problem to the GROW WG [101]. The reason for this is that SIDR not only considers route leaks out of their scope but also because their proposals, including RPKI [4], ROA [5] and BGPSEC [6], fail to counter route leaks. There exist some attempts in the literature from where we can extract the initial understanding of the route leak problem. In [85], the author defines route leaks as *the advertisement of a non-customer route over a peer or a provider link.*

It is worth mentioning that a route leak requires neither a false route origin claim nor a false AS-path advertisement to succeed. For example, when Dodo network leaked Optus routes toward Telstra, it neither needed to claim ownership of Optus routes nor to advertise an inexistent path toward Optus. The only violation was that Dodo advertised Optus routes toward Telstra, against the business policy set on the link between Dodo and Telstra. Therefore, a route leak can only occur when exporting routes to a neighbor AS, and the root cause is the violation of the business policy according to the link classification between the two ASes. The valley-free rules summarize the best practice guidelines for exporting routes. In this regard, the valley-free rules can be used as basis for providing an initial definition of the route leak problem.

> **Definition 1.***"If a route is advertised by an AS toward a neighbor AS, such that it is in violation of the valley-free rules $\mathscr{R}.2$ or $\mathscr{R}.3$, then the route advertisement is a route leak."*

That is, any route advertisement by an AS which infringes the valley-free rules $\mathscr{R}.2$ or $\mathscr{R}.3$ is a route leak. Note that rule $\mathscr{R}.1$ cannot be infringed, since an AS can always export customer routes independently of the business relationship with the neighbor to which it is exporting the route to. Also note that the valley-free rules are not necessarily upheld while exchanging routes under complex AS relationships, e.g., under hybrid relationships—these will be discussed later in Chapter 6.6. However, such complex relationships are quite uncommon in practice, so the above definition provides a realistic and quite general basis for our initial modeling of route leaks.

For better understanding the route leaks, let us represent the Internet as an undirected graph $G = (V, E)$, where V corresponds to set of all ASes in the Internet and E corresponds to set of all links between the ASes, then for a particular autonomous system v, let us define

$\mathbb{P}_v$: set of all the providers of v
$\mathbb{J}_v$: set of all the peers of v
$\mathbb{C}_v$: set of all the customers of v

and for x and y representing an AS or set of ASes, we define

$\mathbb{R}^I_{x,y}$: set of all the routes imported by x from y
$\mathbb{R}^E_{x,y}$: set of all the routes exported by x to y
$\mathbb{O}_v$: the set of all the routes owned by v

Mindful of the fact that route leak can occur while exporting routes toward a peer AS or a provider, we define:

$$\mathbb{R}^E_{v,j} \subseteq \mathbb{R}^I_{v,\mathbb{C}_v} \cup \mathbb{O}_v \quad where,\ j \in \mathbb{J}_v \tag{5.1}$$

$$\mathbb{R}^E_{v,p} \subseteq \mathbb{R}^I_{v,\mathbb{C}_v} \cup \mathbb{O}_v \quad where,\ p \in \mathbb{P}_v \tag{5.2}$$

as sets of routes that v can export to a peer or a provider without causing a route leak, respectively. That is an AS can only export its own routes and its customers' routes toward its peer ASes or provider ASes.

Furthermore, (1) and (2) can also be expressed as:

$$\mathbb{R}^E_{v,j} \not\supset \mathbb{R}^I_{v,\mathbb{J}_v} \cup \mathbb{R}^I_{v,\mathbb{P}_v} \quad where,\ j \in \mathbb{J}_v \tag{5.3}$$

$$\mathbb{R}^{E}_{v,p} \not\supset \mathbb{R}^{I}_{v,\mathbb{J}_v} \cup \mathbb{R}^{I}_{v,\mathbb{P}_v} \quad where,\ p \in \mathbb{P}_v \tag{5.4}$$

That is to say that the set of routes exported by an AS toward a peer AS should not contain any routes learned from peer or provider ASes. Similarly, the set of routes exported by an AS toward a provider AS should not contain any routes learned from peer, provider ASes.

Using (3), we can define route leak as:

> **Definition 2.** *"For an AS v, if $\mathbb{R}^{E}_{v,j}$, where $j \in \mathbb{J}_v$, contains a route r, such that $r \in \{\mathbb{R}^{I}_{v,\mathbb{J}_v} \cup \mathbb{R}^{I}_{v,\mathbb{P}_v}\}$, that is, $\mathbb{R}^{E}_{v,j} \cap \{\mathbb{R}^{I}_{v,\mathbb{J}_v} \cup \mathbb{R}^{I}_{v,\mathbb{P}_v}\} \neq \emptyset$, then it is a route leak"*

Thus, an AS exporting a route toward its peer AS which it learned from another peer AS or provider AS falls in the category of route leaks. Similarly, using (4) we define route leak as,

> **Definition 3.** *"For an AS v, if $\mathbb{R}^{E}_{v,p}$, where $p \in \mathbb{P}_v$, contains a route r, such that $r \in \{\mathbb{R}^{I}_{v,\mathbb{J}_v} \cup \mathbb{R}^{I}_{v,\mathbb{P}_v}\}$, that is, $\mathbb{R}^{E}_{v,p} \cap \{\mathbb{R}^{I}_{v,\mathbb{J}_v} \cup \mathbb{R}^{I}_{v,\mathbb{P}_v}\} \neq \emptyset$, then it is a route leak"*

Using the above definitions, we identify two possible types of route leaks from the perspective of an AS which wants to detect route leaks corresponding to the type of the link they occur on, namely, *Customer Route Leaks* and *Peer Route Leaks*. We proceed to describe them through examples.

Customer Route Leak: Consider the scenario shown in Fig. 5.4 (a). The AS b has a peer relation with AS a, and a provider relation with ASes c and d, i.e., c and d are customers of b. The AS c is multihomed with ASes a and b, i.e., c has two providers, a and b. Let us

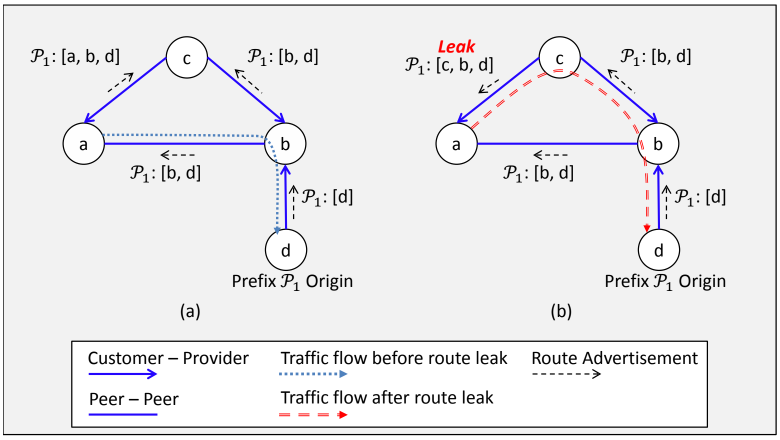


Figure 5.4: Customer route leak scenario: (a) Before the route leak. (b) After the route leak (AS c leaks a route toward its provider AS a).

consider now the propagation of a route for prefix $\mathscr{P}_1$ owned by d, i.e., d advertises $\mathscr{P}_1 : [d]$ to its provider b. Following $\mathscr{R}.1$, b forwards $\mathscr{P}_1 : [b, d]$ toward its other customer c and its peer a. In line with $\mathscr{R}.2$, a advertises $\mathscr{P}_1 : [a, b, d]$ to its customer c. The traffic for a source in a and a destination in d would follow the path $[a, b, d]$, as shown in Fig. 5.4 (a). In the case that c advertises a route learned from one provider to another provider, i.e., advertises the route for prefix $\mathscr{P}_1$ to its provider a, then a would receive two routes for prefix $\mathscr{P}_1$, i.e., $\mathscr{P}_1 : [b, d]$ via b and $\mathscr{P}_1 : [c, b, d]$ via c, as shown in Fig. 5.4 (b). As mentioned earlier, ASes usually prefer routes learned from customers over routes learned from peers. Consequently, the traffic between a and d will now follow the path $[c, b, d]$. It is worth mentioning that, although the AS-path length via b is shorter than the AS-path length via c, AS a would select the customer route, since the latter is prioritized by setting a higher value of the *local-pref* attribute, which is evaluated before the *AS-Path Length* attribute during the BGP route selection algorithm [1]. According to Definition 1, the advertisement of prefix $\mathscr{P}_1$ by c toward its provider a is a route leak, since it violates the valley-free rule $\mathscr{R}.3$.

Peer Route Leak: Let us consider now the scenario shown in Fig. 5.5 (a). The AS c is multi-homed with provider ASes a and b. AS d has a peer relation with AS e, and AS d and AS e have a customer-provider relationship with ASes a and b, respectively. AS a and AS b also have a peer link between them. Let us consider the propagation of a route for prefix $\mathscr{P}_1$ owned by AS c, i.e., c advertises the route $\mathscr{P}_1 : [c]$ to its providers. Following $\mathscr{R}.1$, a forwards $\mathscr{P}_1 : [a, c]$ to its customer d and b forwards $\mathscr{P}_1 : [b, c]$ to its customer e. By $\mathscr{R}.3$, d does not advertise the route to its peer e, and reciprocally. The traffic aimed for $\mathscr{P}_1$ originated in AS d would follow the path $[a, c]$ as shown in Fig. 5.5 (a). Now, if as shown in Fig. 5.5 (b), AS e advertises the route for prefix $\mathscr{P}_1$ to its peer d, the latter would receive two different routes for prefix $\mathscr{P}_1$, i.e., $\mathscr{P}_1 : [a, c]$ via a, and $\mathscr{P}_1 : [e, b, c]$ via e. Since d will prefer routes learned from peers over routes learned from providers, the traffic between d and c will now follow the path $[e, b, c]$. Note that similarly to the case shown in Fig. 5.4, the path length via a is shorter than the path length via e, but still d will select the peer route, since d will prioritize it by setting higher the *local-pref* value. In this example, the route $\mathscr{P}_1 : [e, b, c]$ exported by AS e toward AS d results in a route leak, given that it violates the valley-free rule $\mathscr{R}.3$. Observe that, the route leak examples shown in Figs. 5.4 and 5.5 infringe rule $\mathscr{R}.3$, but other examples can be easily elaborated infringing rule $\mathscr{R}.2$.

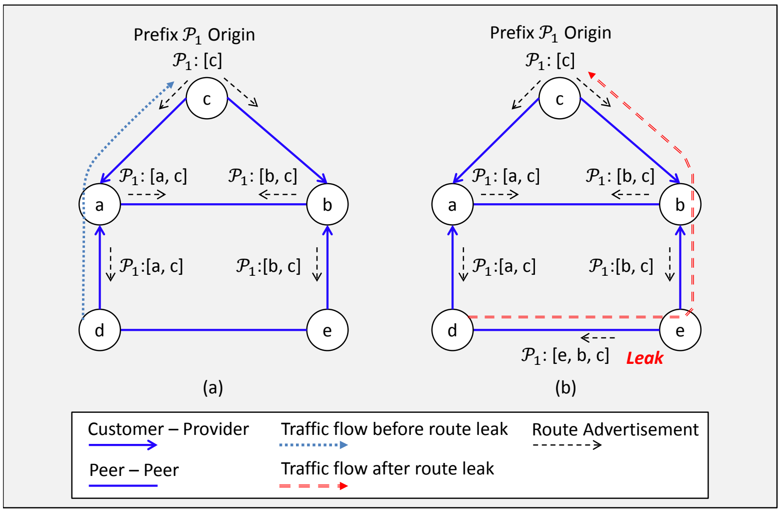


Figure 5.5: Peer route leak scenario: (a) Before the route leak. (b) After the route leak (AS *e* leaks a route toward its peer AS *d*).

6 Route Leak Detection

In this chapter, we present our route leak detection solutions which under realistic assumptions and routing conditions, enable a single AS to detect route leaks by utilizing only the standard routing information available at hand, and without needing any vantage point deployed in the internetwork or third party security infrastructure. First, we outline the theoretical framework for detecting route leaks, then based on the theoretical framework, we develop three incremental route leak detection techniques, namely Cross-Path (CP), Benign Fool Back (BFB) and Reverse Benign Fool Back (R-BFB). The first two techniques are based on the analysis of BGP's control-plane information only, whereas the third technique, R-BFB, also takes advantage of data-plane traffic to provide additional information to the analytics performed on the BGP RIBs.

6.1 Route Leak Detection Framework

The failure of the traditional countermeasures for detecting route leaks is evident from the frequent occurrences of Internet service disruptions due to these incidents. Learning from the collapse of traditional solutions, we can infer that any approach toward resolving the route leak problem should consider the following factors: 1) minimum reliance on third party information; 2) minimum possible changes to the legacy control-plane protocols; 3) real-time detection; and 4) minimum possible administrative overhead. The minimum reliance on third party information is important not only because of the limited reach of the information gathered at vantage points, but also because of the high administrative cost required to train and maintain the monitoring infrastructure up-to-date with the routing policies. Furthermore, serious efforts are required for trust establishment between the relying party and the third party to avoid bogus information exchanges. The second factor stems from the fact that a solution requiring significant changes to control-plane protocols will meet the same fate as such previous inter-domain security propositions, i.e., resistance in adoption. Then, the real-time detection is a necessity because of the way route leaks operate. As mentioned earlier, detecting route leak initiation is easier than detecting propagated route leaks, hence early detection of a route leak is essential. Moreover, two vital goals to be considered when designing a route leak detection algorithm are, to ensure a low administrative cost for maintaining the system, and that the detection technique itself does not hinder the rest of the network functions.

The identification of route leaks is the first step toward solving the route leak problem. Thus, we systematically analyze the various environments where route leaks are possible, and then propose a very simple yet powerful mechanism for their identification.

In our framework, we assume that the route leak identification analysis only uses readily available data, e.g., information obtained directly from the routing tables—that is, from the Route Information Base (RIB) of the routers. We particularly exclude from our framework 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 routes received from its neighbors. Moreover, the framework under consideration also excludes uncommon AS relationships such as sibling and hybrid relations, given that such AS relationships are relatively negligible as compared to peer-peer and customer-provider relations. However, we shall delve in to route leak detection for the sibling relationship in Section 6.5.

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

Fact $\mathscr{F}$.1.*"A route leak can only be produced by an AS on its peer or provider links".*

Given the definitions detailed in the previous chapter, we know that an AS acting as provider cannot leak a route toward its customers, since it inherently has the role of providing transit to its customers, so it can advertise "all" its routes toward them. Directly derived from $\mathscr{F}$.1 and Definition 1, we obtain the cases where a route leak is possible.

Fact $\mathscr{F}$.2.*"A route leak can only occur when an AS receives routes from a peer or a customer AS, which it imported from its respective peers or providers".*

To illustrate this fact let us consider Fig. 6.1(a). Let us assume a reference AS a in charge of identifying route leaks. Then for domain a, route leaks can only occur as a result of routes exported by its customer AS c or peer AS p. In the case that c exports routes owned by itself, then such route advertisements can never produce a route leak, since c, being customer of a, can export its own routes to its provider. Similarly, p is allowed to export its own routes to its peer a. Hence, it should be clear that the advertisement of routes owned by a customer or a peer ASes can never cause a route leak on AS a. In other words, a route leak could only occur when a customer or a peer AS exports routes that they imported from their respective neighbors. Observe that, according to $\mathscr{R}$.1, an AS can export the routes it imported from its customers toward its providers or peers, i.e., both c and p are allowed to export the routes that they imported from their customer cones toward AS a. Then, by using the facts $\mathscr{F}$.1 and $\mathscr{F}$.2

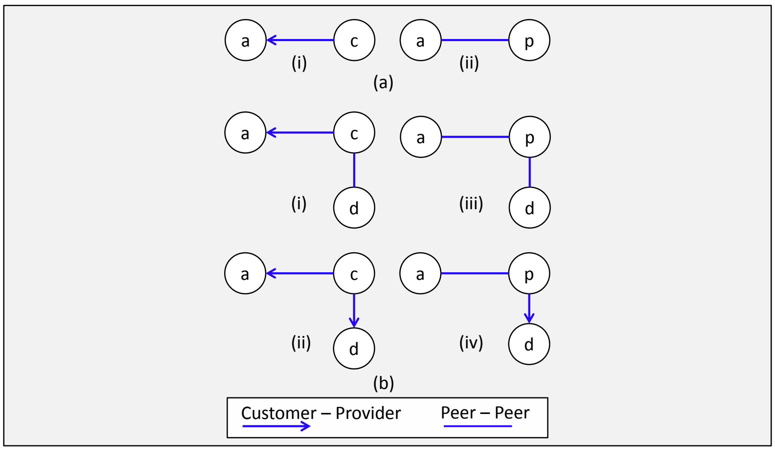


Figure 6.1: (a) Possible cases for the occurrence of a route leak on AS *a*; (b) Possible neighbor links of AS *a*'s customers and peers that can produce a route leak on AS *a*.

together, it is obvious that the possible network topologies for the occurrence of a route leak for AS *a* are the ones shown in Fig. 6.1(b). For the customer route leak case, *c* could leak either its peer or its provider routes to *a*. Similarly, for peer route leak scenario, *p* could leak either its peer or provider routes to *a*. In any route leak scenario, there are at least three ASes involved; the victim AS *V* which receives the leaked routes, the route leaker AS *L* which leaks the route, and the owner AS *O* which owns or forwarded the routes that are leaked. For example, in Fig. 6.1(b) (i), *a* is the victim *V*, *c* is the route leaker *L*, and *d* is the owner *O* of the routes that can be leaked.

It is worth mentioning that *a*, the victim, is only aware of AS relationships with its direct neighbors, but has no information about the relationships that its neighbors have with their respective neighbors. AS *a* can learn the identity of the neighbors' neighbors from the AS path information included in the route advertisements, but remains unaware of their relation. This is because an AS has limited knowledge of the network, since the relationships and policies among ASes are kept confidential. The challenge for AS *a* is thus to independently detect route leaks despite the lack of information of its neighbors' neighbors relationships.

Let us then consider a network topology scenario for generalizing the local identification of a route leak. Figure 6.2 depicts the case where our reference AS *a* is the victim (*V*) receiving new route advertisements from its neighbors. The goal is to examine under which conditions AS *a* can locally validate these advertisements prior to inserting them in the RIB and FIB tables of its routers. Domain *b* represents a neighbor that is directly connected to AS *a* by a peer-peer or customer-provider link, and it is the one that the victim *a* suspects that is responsible for leaking the routes (the leaker *L*). Furthermore, *c* (the owner *O*) is a direct neighbor of *b*, which advertises valid routes to AS *b* of the form $[c,\ldots]$ (where "…" refers to zero or more ASes in the AS-path). These routes can be potentially announced by *b* to *a*, e.g, through routes of the form $[b,c,\ldots]$. These announcements can be identified as leaks by the victim if they are against the valley-free rules. However, from *a*'s perspective, the announcements cannot be validated due

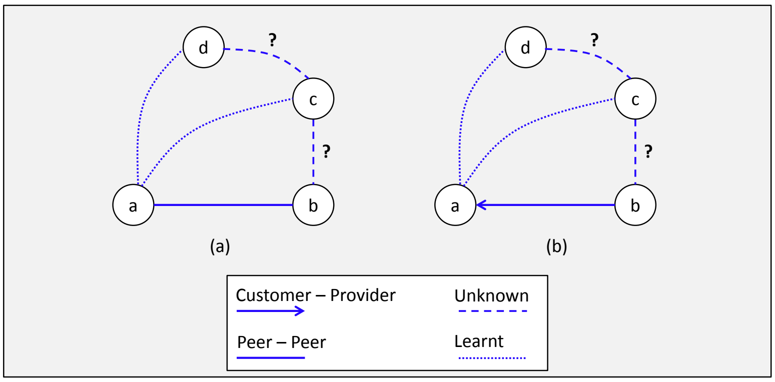


Figure 6.2: Generalized topologies for route leak detection: (a) Peer Route Leak. (b) Customer Route Leak.

to the lack of information about the type of relationship between the suspect b and its direct neighbor c.

We already stated that the minimum scenario required for a route leak occurrence contemplates three actors: the victim, the leaker, and the owner of a route. However, for the sake of generality, we consider the case when the suspect b leaks a route imported from c, but that was originated by another AS, e.g., d. Thus, the potential route that AS b would leak to the victim a would be one learned from c toward d, of the form $[c,\ldots,d]$. 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 $[\ldots,d]$. When the suspect AS b leaks the route to AS d to attract its traffic (i.e., AS b advertises to AS a a route of the form $[b,c,\ldots,d]$), the victim will be in a position to observe a new route advertisement for the same destination AS.

This reference topology and the general assumptions that we will make next shall be used in the remainder of this Chapter, while formalizing the identification of route leaks in Theorems 1 and 2.

Hypothesis $\mathscr{H}$.1. *"The state of the routing databases of the victim AS is valley-free valid before the route leak occurs."*

Remark: The purpose of our theoretical framework 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, $\mathscr{H}$.1 indicates that any route contained in the initial state of the RIBs at AS a is compliant with $\mathscr{R}$.1, $\mathscr{R}$.2 and $\mathscr{R}$.3.

Hypothesis $\mathscr{H}$.2. *"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 Fig. 6.3 (a), 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 (cf. Fig. 6.3 (a)).

> **Hypothesis** $\mathscr{H}$.3.*"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. As shown in Fig. 6.3 (b), 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) [117].

It is important to mention that, $\mathscr{H}$.2 and $\mathscr{H}$.3 also avoid the possibility of false positives. However, the assumptions that an AS prefers a customer route over a peer or a provider route and an AS prefers shorter AS-Path for a given destination may not always be true as it entirely depends on the internal policy of an AS. Hence, in scenarios where the latter two norms are violated, the occurrence of a false positive is a possibility, however the violation of the mentioned norms will reduce the effectiveness of the route leak as well.

Now, given the valley-free rules (i.e., $\mathscr{R}$.1–$\mathscr{R}$.3), and the hypotheses defined above, we proceed to formalize the conditions for detecting peer route leaks (cf. Fig. 6.2 (a)).

Theorem 1. *Let the initial state of the routing databases of an AS a contain the following:*

- *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 $\mathscr{H}$.1, $\mathscr{H}$.2, and $\mathscr{H}$.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.*

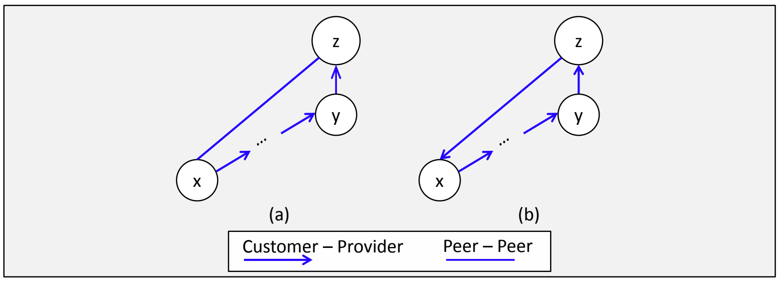


Figure 6.3: Unlikely AS relationships among ASes: (a) Hypothesis 2. (b) Hypothesis 3.

Proof. According to $\mathscr{R}.1$–$\mathscr{R}.3$, AS b could only advertise a route with AS-path $[b, c, \dots]$ 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 $[\dots, c, b]$. Now, a could only receive the route to b with AS-path $[\dots, c, b]$, iff, AS a belongs to the customer cone of AS c. This is because according to $\mathscr{R}.3$, 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 $\mathscr{H}.2$, 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 $[b, c, \dots]$ is a route leak. □

To illustrate the reach and potential application of Theorem 1, let us consider again the peer route leak example given in Fig. 5.5 (b). In practice, the route database of AS d would have a route with AS-path $[a, b, e]$ to e via b, plus the direct route $[e]$ to e in its initial state. The former is because a and b would exchange customer routes with each other. Assuming that the initial state at AS d is valley-free valid, the set up in Fig. 5.5 (b) is under the hypotheses of Theorem 1, so AS d can autonomously conclude that the route $\mathscr{P}_1 : [e, b, c]$ received from AS e is a route leak.

We proceed now to formalize the detection of customer route leaks (cf. Fig. 6.2 (b)).

Theorem 2. *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* $[\dots, c, b]$.

Under the hypotheses $\mathscr{H}.1$, $\mathscr{H}.2$, and $\mathscr{H}.3$, if AS a receives a route from its customer AS b with AS-path $[b, c, \dots]$, then AS a can identify it is a route leak.

Proof. Just as in the proof of Theorem 1, AS b could only advertise a route with AS-path $[b, c, \dots]$ 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 $[\dots, c, b]$. Now, a could only receive the route to b with AS-path $[\dots, c, b]$, iff, a belongs to the customer cone of c. This is because according to $\mathscr{R}.3$, c would advertise its provider routes only to its customers. But if a belongs to the customer cone of c, then this contradicts $\mathscr{H}.3$, 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 advertised by AS b toward AS a with AS-path $[b, c, \dots]$ is a route leak. □

It can be shown that if the initial conditions are met, then Theorem 2 applies to the example illustrated in Fig. 5.4 (b).

6.2 Cross-Path (CP) Route Leak Detection Technique

In this section, we start with one of the most straightforward approaches for detecting route leaks. In the following sections, we will incorporate additional mechanisms, which, as we shall show, will progressively improve the results in the detection. In a nutshell, 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 a short period, so as 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

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

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

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

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

updated. We further discuss the viability and impact of using route filters in Section 6.6. Now, 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 [72], 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 a AS learns a route to a Tier-1 AS or a route to any destination via Tier-1 through a neighbor customer or peer AS. In this regard, our approach is more comprehensive and encompasses the logic used in [72].

It is worth mentioning that the CP technique is effective for both unintentional and deliberate route leak attacks. Given the properties of the CP technique, i.e., it does not depend on the routing protocol and it can be run decoupled from the underlying technology. The complementary part (i.e., the remediation method) can be implemented as a third party application, such as an SDN application, thereby avoiding modification or extension of the BGP protocol. We further elaborate this point in Chapter 7.6.

6.3 Benign Fool Back

In the context of improving the performance of CP for detecting route leaks when the leaker L leaks its peer routes toward the victim V, we propose *Benign Fool Back (BFB)*. The BFB targets to improve the route leak detection in PRL cases where L and O have a peer relation. This technique exploits the commonly practiced preference of routes based on the type of relationships an AS has with its neighbors. We assume that, under normal circumstances, an AS, more specifically the leaker, follows the principle of preferring customer routes over peer and provider routes, and that it prefers a shorter AS-path route over a longer one. However, this policy might not necessarily be upheld always, such as in case of sibling AS relationships, but at least applies for a majority of them. We also assume that the ASes involved in the potential route leak incident are not using IP prefix origin verification mechanisms, such as ROA [5]. We claim that these are realistic assumptions, since most of the route leaks reported in the Internet are due to apparent misconfigurations rather than deliberate attacks, and ROA is not used by the large majority of the ASes in the Internet. To illustrate BFB, let us consider the example shown in Fig. 6.4 (a). If an AS, the potential victim V, starts receiving new routes

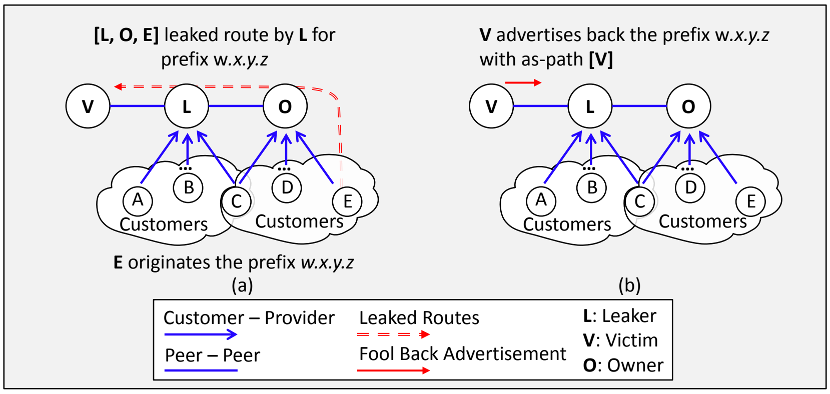

[L, O, E] leaked route by L for prefix w.x.y.z
V
L
O
A
B
C
D
E
Customers
Customers
E originates the prefix w.x.y.z
(a)
V advertises back the prefix w.x.y.z with as-path [V]
(b)
Customer – Provider
Peer – Peer
Leaked Routes
Fool Back Advertisement
L: Leaker
V: Victim
O: Owner

Figure 6.4: Benign Fool Back: (a) L leaks O's routes to V; (b) The potential victim V sends a Fool Back advertisement to L.

from a peer neighbor, the potential leaker L, for which V had never had any route for those destinations through L, then V can be suspicious of these new routes, and trigger the BFB strategy if the CP technique described in the previous section did not detect any leak. For this, V chooses one or more destination IP prefixes from the newly advertised routes by the peer L matching the following two criteria.

1. The AS-path advertised by L to reach a particular IP prefix owned by an AS E should be of the form $[L, O, \ldots, E]$, i.e., the destination IP prefix belongs to an AS E which is at least two AS hops away from L. Observe that the IP prefix is not advertised as owned by L—otherwise is not a "leak", since L can advertise its own routes to V.

2. The AS E, the owner of the selected destination IP prefix for fool back advertisement, is not a customer of both L and O.

In this framework, if V suspects this could be the result of a route leak, then V could select an IP prefix destination from the newly received suspicious routes according to the criteria defined above and advertise it back to L, that is, V could try to fool back its peer L (see Fig. 6.4 (b)). Let us assume that V chooses IP prefix $w.x.y.z$ to fool back its peer L for identifying a route leak. Once L receives the fake advertisement for $w.x.y.z$ from V, there are two options, L could either accept this route as its best path or not. If L selects the fake advertisement from V as the best route toward IP prefix $w.x.y.z$, then it would send a withdrawal for the IP prefix $w.x.y.z$ route it sent earlier toward V. On reception of the withdrawal from L, V can infer that the route received earlier from L for $w.x.y.z$ was a leak—that is, it was a non-customer route received by V on its peering link with L. This is because if $w.x.y.z$ belongs to the customer cone of L, then L would have not selected the fake route sent by its peer V, since, according to our hypothesis, customer routes are preferred over peer routes. Also observe that, the decision of choosing candidate routes that are at least two AS hops away from L increases the chances of BFB to succeed, since thanks to the shortest-path principle, the Fool Back advertisement $[V]$ for $w.x.y.z$ will prevail over the alternative peer route $[O, E, \ldots]$ at L. The AS V can run BFB

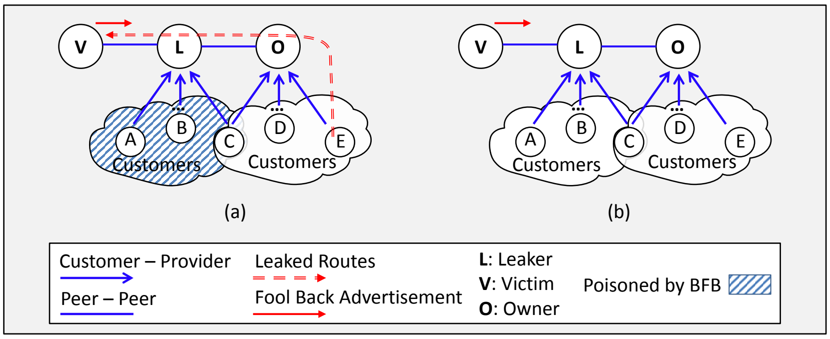


Figure 6.5: Route poisoning impact of Benign Fool Back: (a) Valid suspicion (b) False suspicion.

strategy for the all the newly received suspicious routes by carefully selecting the fool back IP prefix to detect route leaks.

Let us now consider the example when the potential victim V initiates the BFB strategy on a false suspicion. For the case of PRL, even if V sends the Fool Back advertisement to the alleged leaker L, this would not prefer it over its legitimate customer route, and hence the fool back advertisement would stay harmless in legitimate cases—this is why we call this strategy “benign”. In other words, the fool back advertisement would only poison the route for customers of L in the case that L had leaked a route to V. As shown in Fig. 6.5 (a), in case of an actual leak even if L forwards the poisoned route toward O, it will remain harmless as E belongs to the customer cone of O and it will prefer a customer route over the poisoned peer route. In the case of a false suspicion, the fool back advertisement from the V toward the suspected L causes no adverse affects, as depicted in Fig. 6.5 (b). Also observe that once the withdrawal is received by the victim, it can start the remediation actions and withdraw the Fool Back advertisement. In the case of CRL, where L leaks its peer routes toward the victim V, the BFB is not applicable. This is because L is leaking peer O routes toward its provider V, thus a fool back advertisement from the provider victim will be worthless in presence of existing peer routes.

The BFB technique allows detection of route leaks in PRL cases where the leaker mistakenly leaks the routes it learnt from its peers, however we contend that the BFB technique can be neutralized by a premeditated route leak attack. Nevertheless, the BFB technique remains effective for unintentional route leaks occurrences such as the ones due to misconfigurations.

6.4 Reverse Benign Fool Back

In the previous sections, we presented two route leak detection techniques and showed that different type of route leaks in different scenarios can be detected by using BGP intelligence available at the control-plane level only. In order to further improve the route leak detection performance, we propose to use data-plane traffic intelligence along with the control-plane in Reverse BFB. The R-BFB targets to improve the route leak detection in PRL as well as CRL cases where L and O have a peer relation. As self-explanatory from the name, this technique is based on the BFB technique described in the previous section, however the "reverse" means that it is the O who initiates the benign fool back advertisement and tries to detect a route leak occurrence. Furthermore, R-BFB utilizes both control-plane and data-plane information to counter route leaks. If an AS observes traffic through one of its peer neighbor from sources that the neighbor has not advertised through BGP, then either it could be because of an unadvertised new customer of the peer neighbor or the AS might be a collateral victim of a route leak. The reason we say collateral victim is that the alien traffic received by the AS might be due to a route leaked by the corresponding neighbor (through which the AS is receiving the traffic) to one of its neighbors. We explain the R-BFB technique with help of an example scenario shown in Fig. 6.6 (a). If L leaks the routes learned from O to V, then the traffic from AS G to AS E would follow the path $[G, V, L, O, E]$. If L has not advertised routes learned from V to O, i.e., it leaked in one direction only, then O can take measures to verify if it is a collateral victim of a route leak by using R-BFB. For this purpose, O chooses IP prefix from the unadvertised sources (i.e., AS G) and advertise them back to the AS from where it is receiving the traffic, i.e., L. The criteria to choose the unadvertised source (i.e., AS G) for the reverse benign fool back are the following.

1. There is no route of AS G advertised by L at O.
2. AS G, the selected unadvertised source for the reverse fool back advertisement, is not a customer of both L and V.
3. AS G is at least two AS hops away from L.

On receiving a fake shorter AS-Path length route advertisement for AS G from O, if L decides to choose it as its best path toward AS G, then O can be assured that it is a collateral victim of a route leak. But unlike BFB, L will not send any withdrawals toward O for AS G routes as it never advertised them to O in the first place. However, O can still sense the change of best

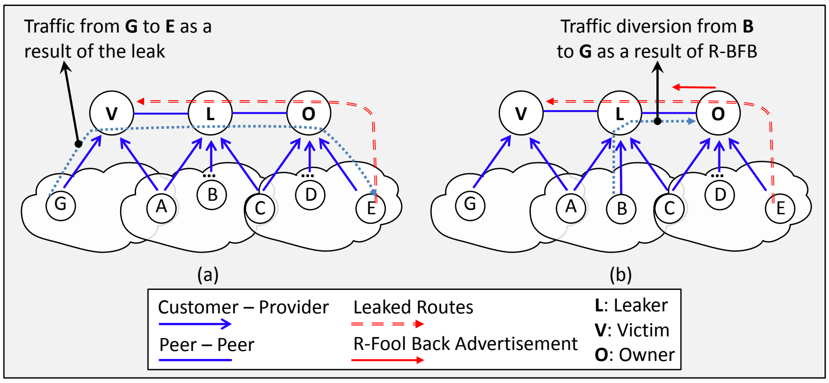

Traffic from G to E as a result of the leak
Traffic diversion from B to G as a result of R-BFB
V
L
O
G
A
B
C
D
E
(a)
(b)
Customer – Provider
Peer – Peer
Leaked Routes
R-Fool Back Advertisement
L: Leaker
V: Victim
O: Owner

Figure 6.6: Reverse BFB for PRL: (a) Traffic flow from AS G to AS E due to route leak (b) Traffic flow from AS B to AS E due to reverse fool back advertisement.

path at L for AS G, if it receives traffic destined for AS G from L, that is L accepted O's false reverse fool back advertisement. As shown in Fig. 6.6 (b), the traffic from B to G gets diverted toward O because of the reverse benign fool back advertisement instead of taking the path B, L, V, G. This confirms that the traffic $[G, V, L, O, E]$ was indeed a consequence of a route leak because if AS G was a new unadvertised customer of L, then it would not have preferred the false reverse fool back advertisement over it. The R-BFB has similar line of reasoning for verifying if an unadvertised source traffic is a fallout of a route leak as BFB, however the former depends on data-plane traffic monitoring for triggering and concluding itself.

It is important to note that the impact of the R-BFB, in terms of route poisoning of AS G, is only confined to the customers of L, in case of an actual route leak. In case of the false suspicion, the false reverse fool back advertisement of R-BFB gets discarded against a valid customer route and thus has no adverse affects. Unlike the BFB, the R-BFB is applicable to PRL as well as CRL in which L and O have peer relation, as illustrated in Fig. 6.7. Furthermore, it is worth mentioning that unlike CP and BFB, which allow an AS to detect route leaks if the AS is a direct victim, R-BFB enables an AS to detect route leaks that are not directed at the AS but are affecting it one way or the other. Furthermore, like BFB, R-BFB looses its robustness in face of a planned route leak attack, i.e., a well prepared attacker can avoid falling in the fool-back advertisement trap.

6.5 Sibling Route Leak Problem

In this section we analyze the route leak problem in the context of sibling AS relationship. Any two different ASes are said to have a sibling-sibling relation among themselves if they are under the administration of a single organization. For example, if a larger ISP acquires a smaller ISP with a distinct ASN or extends its network under a different ASN, then the relationship between the two ASes, now under the same administration, is called a sibling-sibling relationship, i.e., they are the children of the same 'mother' organization. In the

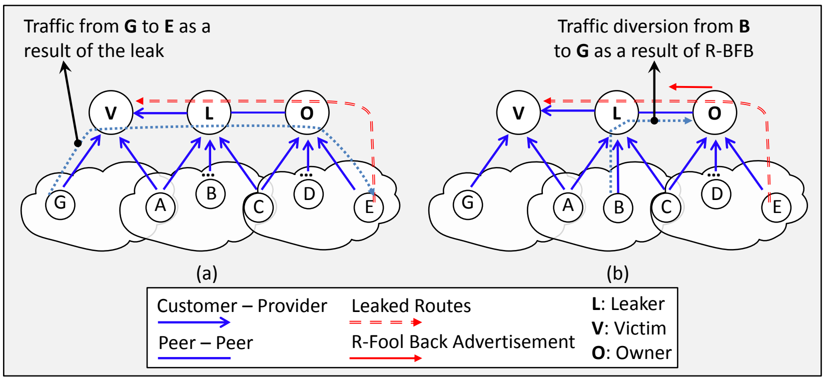

Traffic from G to E as a result of the leak
Traffic diversion from B to G as a result of R-BFB
V
L
O
G
A
B
C
D
E
(a)
(b)
Customer – Provider
Peer – Peer
Leaked Routes
R-Fool Back Advertisement
L: Leaker
V: Victim
O: Owner

Figure 6.7: Reverse BFB for CRL: (a) Traffic flow from AS G to AS E due to route leak (b) Traffic flow from AS B to AS E due to reverse fool back advertisement.

sibling-sibling relation, the ASes typically offer transit to each other. That is, sibling ASes can exchange their provider, peer and customer routes between themselves. The main reason for analyzing route leak problem in sibling relationship case separately from customer-provider and peer-peer relationships is because the valley-free route re-advertisement model, stated in Chapter 2, does not encompasses the former relationship case. That is, how to re-advertise routes learned from a sibling AS. Although, there are no hard and fast rules governing the re-advertisement of routes learned from sibling AS, the profit optimization goal of a service provider can be used to draw out economically compelling guidelines.

6.5.1 Defining Route Leak Problem for Sibling Routes

In the case of sibling relation, collective (i.e., for both ASes) revenue optimization has to be considered, as the two ASes are owned by the same organization. In that perspective, let us analyze different possible re-advertisement scenarios of sibling routes. Figure 6.8, shows a sibling-sibling relation between AS c and AS d. As shown in the figure, d forwards its provider route for prefix $\mathscr{P}_1 : [d, b]$ toward its sibling c. Now, if c further re-advertises the route for prefix $\mathscr{P}_1 : [c, d, b]$ to its provider AS a, then a would prefer the route it learned from customer c over the route $\mathscr{P}_1 : [b]$ it learned from its peer AS b. As a consequence, the traffic between a and b will follow the path $[a, c, d, b]$, i.e., c and d would be providing a transit between a and b for zero revenue. We recall that c and d being customer of a and b are paying a and b for transit to other networks. Thus, we can consider re-advertisement of sibling's provider routes to own provider as against economic convention.

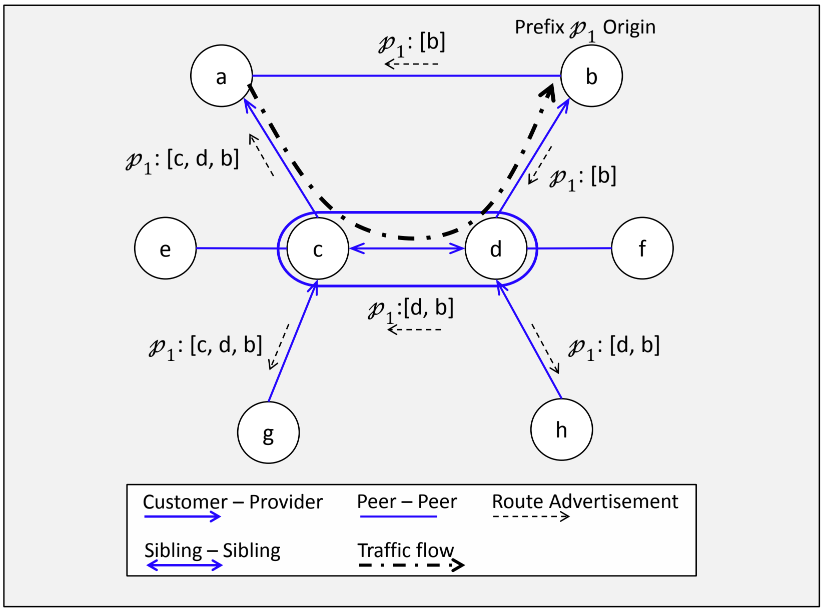


Figure 6.8: Re-advertising sibling's provider routes to own provider.

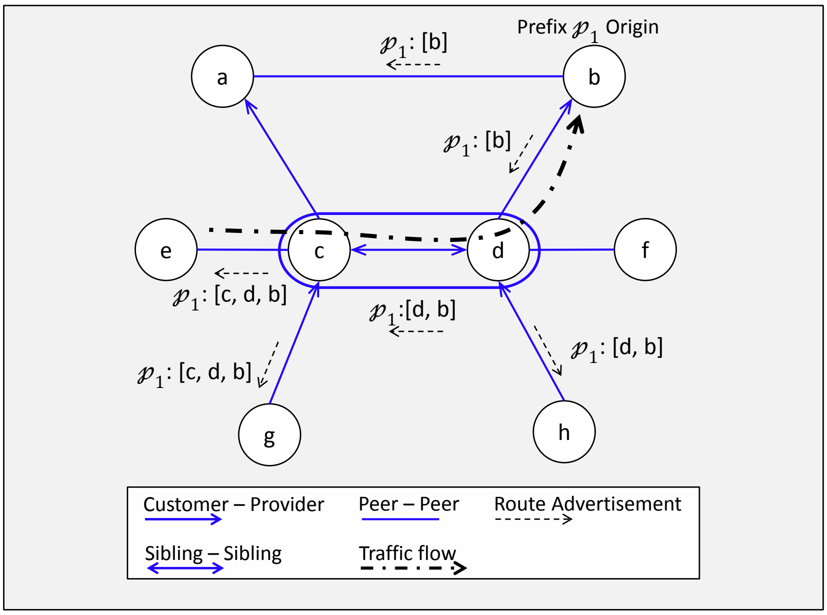


Figure 6.9: Re-advertising sibling's provider routes to own peer.

Figure 6.9, illustrates if an AS forwards sibling's provider route to its peer. That is, c re-advertises its sibling's provider route for prefix $\mathscr{P}_1 : [c, d, b]$ to its peer AS e. As shown in Fig. 6.9, as a result c and d will again be providing transit to traffic whose source and destination does not belong to either of them for zero revenue. Hence, re-advertisement of sibling's provider to

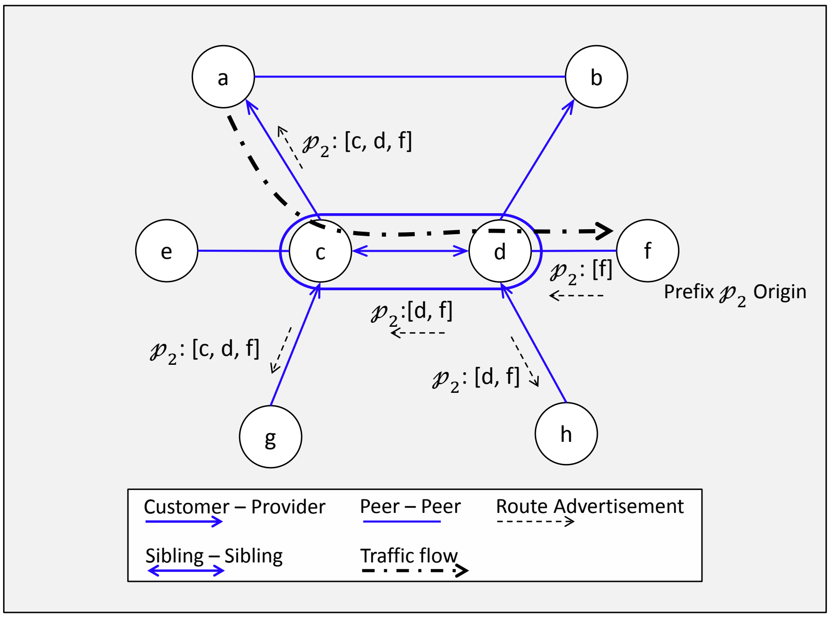


Figure 6.10: Re-advertising sibling's peer routes to own provider.

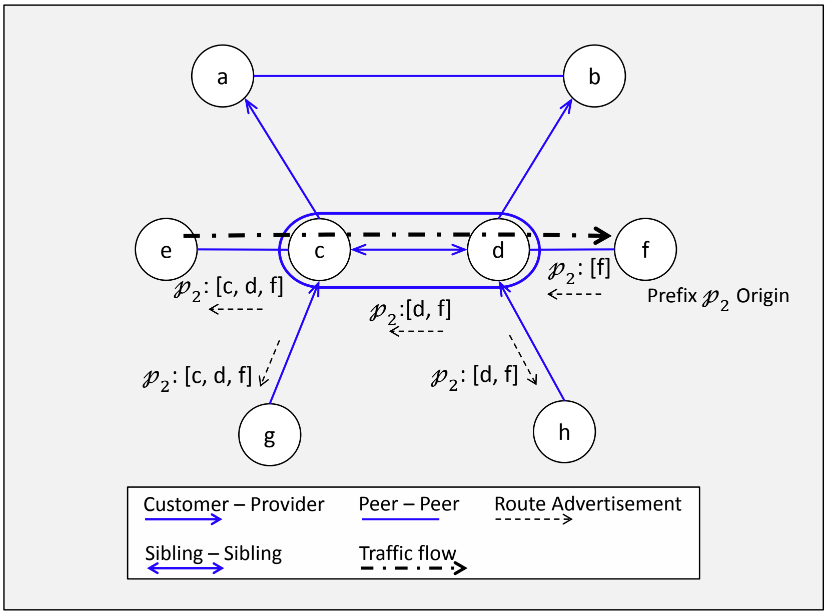


Figure 6.11: Re-advertising sibling's peer routes to own peer.

own peers is economically invalid as well.

Fig. 6.10 and Fig. 6.11 illustrate re-advertisement of sibling's peer route of prefix $\mathscr{P}_2$ toward own provider and peer ASes, respectively. In both cases, the resulting traffic flows will be revenue unfriendly. Thus re-advertisement of sibling's peer routes to own provider and peer AS is economically irrational. It is worth mentioning that re-advertisement of sibling's provider or peer routes toward own customers is economically logical as it might cause traffic between own customers and sibling's provider or peers resulting in increase of revenues.

The re-advertisement of sibling's customer routes toward own provider, peer and customers seems economically prudent as it will cause revenue generating traffic flows, as illustrated in Fig. 6.12.

Based on the above illustrated scenarios, following rules for re-advertisement of sibling routes can be considered:

Rule $\mathscr{R}$.4. *"Sibling's customer routes can be further re-advertised to own customers, peers and providers (Fig. 6.13)."*

Rule $\mathscr{R}$.5. *"Sibling's peer routes can be further advertised to own customers only (Fig. 6.14)."*

Rule $\mathscr{R}$.6. *"Sibling's provider routes can be further advertised to own customers only (Fig. 6.15)."*

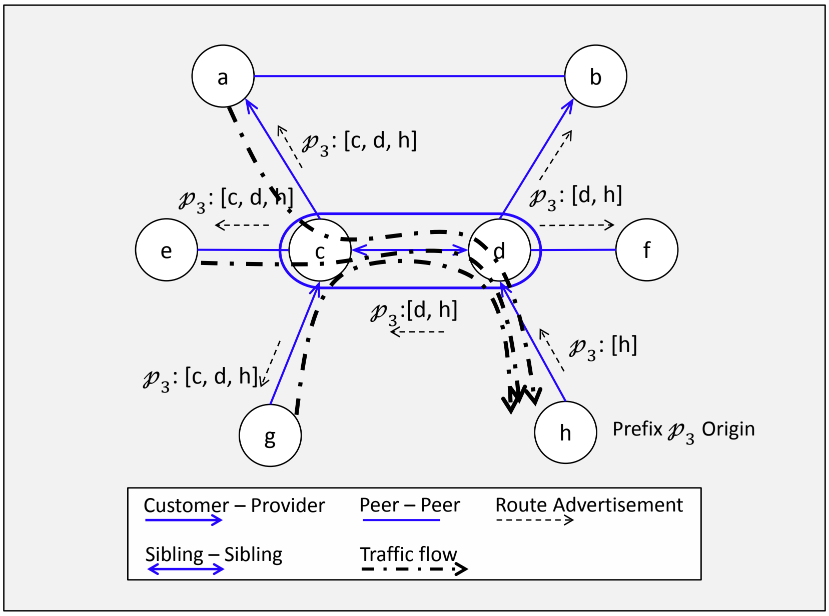


Figure 6.12: Re-advertising sibling's peer routes to own customer.

In the line of $\mathscr{R}$.4, $\mathscr{R}$.5 and $\mathscr{R}$.6, we can define route leak problem in context of sibling relationship as follows:

Definition 4. *"If a route is advertised by an AS toward a neighbor AS, such that it is in violation of rules $\mathscr{R}$.4 or $\mathscr{R}$.5 or $\mathscr{R}$.6, then the route advertisement is a route leak."*

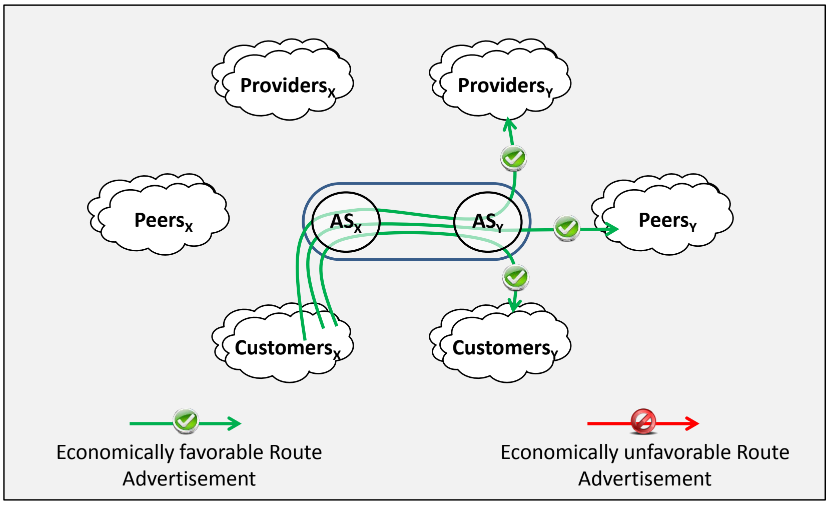


Figure 6.13: Sibling route re-advertisement for customer routes.

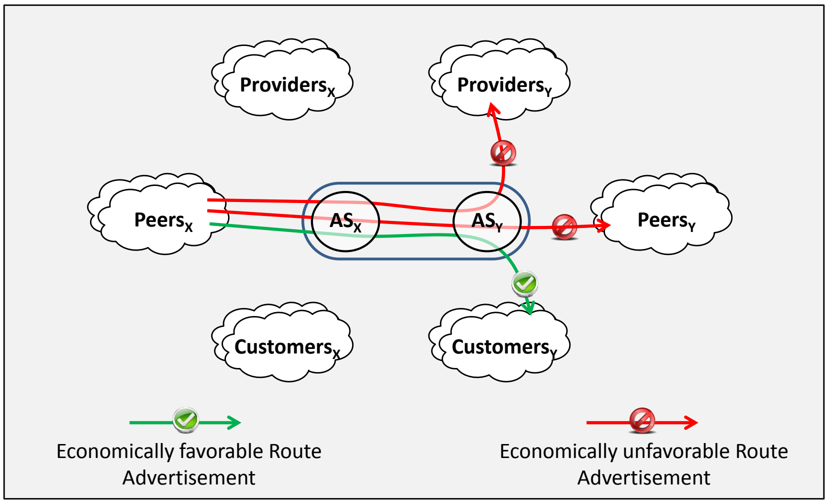

Providers$_X$
Providers$_Y$
Peers$_X$
AS$_X$
AS$_Y$
Peers$_Y$
Customers$_X$
Customers$_Y$
Economically favorable Route Advertisement
Economically unfavorable Route Advertisement

Figure 6.14: Sibling route re-advertisement for peer routes.

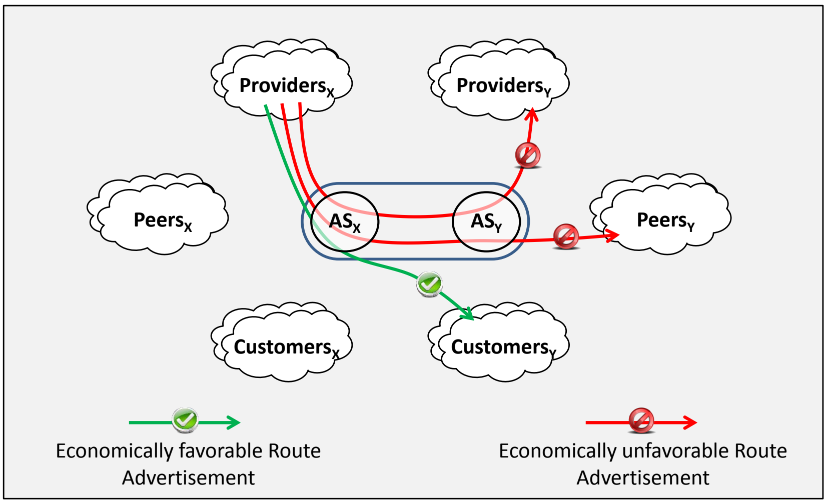

Providers$_X$
Providers$_Y$
Peers$_X$
AS$_X$
AS$_Y$
Peers$_Y$
Customers$_X$
Customers$_Y$
Economically favorable Route Advertisement
Economically unfavorable Route Advertisement

Figure 6.15: Sibling route re-advertisement for provider routes.

Given the above definition, we proceed to reasoning of route leak detection when sibling routes are leaked.

6.5.2 Route Leak Detection for Sibling Routes

The route leak detection techniques described in Sections 6.2, 6.3, and 6.4 can not be directly applied for detecting sibling route leaks given the inherent nature of the sibling AS relationship. We explain this point with the help of an example. Let us consider the network given in Fig. 6.12 and assume, for traffic engineering purposes, that c does not advertises its customer

g directly to its provider a or withdraws the route $[c, g]$ from a. However, it does advertises its customer g to its sibling d which in turn advertises to its provider b. In such a situation following routes with corresponding AS-Paths can be observed in the RIB of a including,

- AS-Path: $[b]$
- AS-Path: $[c]$
- AS-Path: $[b, d, c, g]$
- AS-Path: $[c, d, h]$
- $\ldots$

We can observe a cross-path between $[c, d]$ and $[d, c]$ for two different set of prefixes. The cross-path technique (cf. Algorithm 1) described in Section 6.2 would fall prey to false positive and output route leak detected. This happens due to lack of sibling relation information and as a consequence the cross-path treats the AS c and AS d as two separate entities. Similarly, for BFB and RBFB, lack of information of sibling relation between c and d makes the application of those route leak detection techniques doubtful. The reason we say doubtful is because, having sibling relation among any two ASes allows them to implement complex traffic engineering policies, which can not be anticipated to any point of certainty which makes it difficult for any route leak detection to perform robustly.

In this section, we defined the route leak problem in the presence of sibling relationship. However, we contend that it is important for an AS to have prior information of sibling relationship in order to detect sibling route leaks. For example, prior knowledge of sibling relationships in the cross-path technique can enable it to detect sibling route leaks. The advance information of sibling relationship will allow the cross-path technique to treat the two sibling members as one entity, thus avoiding any false positives. The Algorithm 1 can be updated to accommodate sibling route leak detection, as given in Algorithm 2.

The assumption of beforehand information of ASes which have sibling-sibling relation is not irrational. If not automated, manual effort can be made to build up a set of sibling ASes by utilizing the information available in online databases such as IRR. Although we contend that the BGP policy information available in IRRs is unreliable and not up-to-date, it is reasonable to extract sibling information based on the owner organization as it changes less frequently compared to the BGP policies [67].

Algorithm 2 CP-SIB identifies whether a new route advertisement $\mathscr{R}$ received by AS v is a leak.

```
Input:  Valley-free $RIBs$ - Routing Information Bases at AS $v$
        $\mathscr{N}_c$: Set of customer neighbors of $v$
        $\mathscr{N}_{pe}$: Set of peer neighbors of $v$
        $\mathscr{N}_{pr}$: Set of provider neighbors of $v$
        $\mathscr{S}$: Set of pairs of ASes which have sibling-sibling relationship
        $\mathscr{T}$: List of Tier-1 ASes
        A new route advertisement $\mathscr{R}$ of the form $[l, o, \dots]$.
Output: true if the new route received is a leak
        false otherwise.

if AS $l \in \mathscr{N}_{pe} \cup \mathscr{N}_c$ then
    for all $a_i \in \mathscr{R}$, where $0 < i \leq \mathscr{R}.length$ do
        if $a_i \in \mathscr{T}$ then
            $\mathscr{R} \leftarrow \emptyset$
            return true
        end if
    end for
    $\mathscr{R}' \leftarrow [\dots, o, \dots, l, \dots]$
    if $\mathscr{R}' \in RIBs$ and $(o, l) \notin \mathscr{S}$ then
        $\mathscr{R} \leftarrow \emptyset$
        return true
    end if
end if
$RIBs \leftarrow RIBs \cup \mathscr{R}$
return false
```

6.6 Open Issues

Even though our proposals can be applied in many practical situations (e.g., the Dodo-Telstra incident could have been avoided), there are still some others that might not satisfy the hypotheses of Theorems 1 and 2 given in Section 6.1, and therefore, they need further analysis. In the remainder of this Section, we discuss the reach and limitations of the contributions made in this chapter.

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 hybrid relationships. These latter refer to cases where two large ASes have different relationships between them at geographically different points of presence (PoP). 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 in this thesis 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.

Route Leak Propagation: Observe that our analysis can only be used for detecting when a route leak is initiated. Detecting route leak propagation is far more difficult than detecting its initiation. The route leak propagation refers to the scenario where the *victim* AS receives a route leak and forwards it further to its neighbors. The *victim* AS may forward the route leak to its neighbors according to the relationship it has with them, which makes it more difficult for any AS receiving the propagated route to detect it as a route leak.

A route leak propagation example is illustrated in Fig. 6.16. The AS a forwards the leaked route $\mathscr{P}_1[a,c,b,d]$ received from its customer AS c to its peer AS e, which is allowed according to $\mathscr{R}.1$–$\mathscr{R}.3$. The AS e further advertises this leaked route to its customers, including AS f. Note that neither AS e nor AS f can detect this route advertisement as a route leak, since they receive it in accordance with the relationship that they have with their corresponding neighbors. We discuss the detection of route leak propagation in Chapter 12 as future research opportunity.

Initial Valley-Free State: From an engineering perspective, the hypothesis $\mathscr{H}.1$ 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 this is guaranteed, the route filters need not be maintained 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. In any case, this method of applying and removing filters is challenging for very large providers, and without SIDR's solutions in place, it can only be achieved through a chain of trust during filter configuration. Further research is needed on how to ensure that the initial state at the potential victims is valley-free.

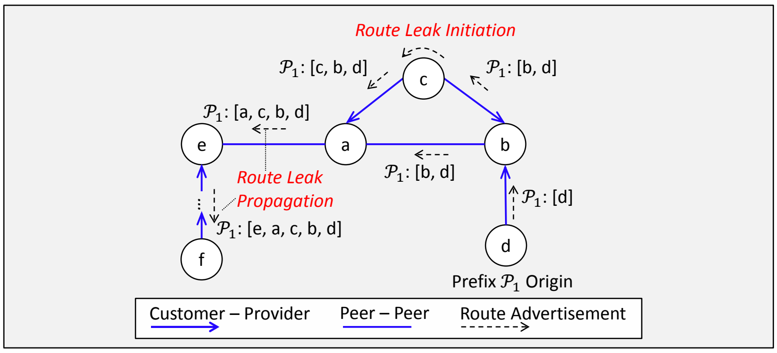


Figure 6.16: Route leak initiation and route leak propagation.

7 Simulations and Experiments

This chapter evaluates the Route Leak Detection (RLD) techniques, proposed in the previous chapter, both experimentally as well as through event-driven simulations on a large scale. For the latter, we utilized sub-graphs of the Internet graph extracted from ARK [73], and we performed simulations using NS2 [118] and BGP++ [119] on two different topologies composed of more than a thousand of ASes. For the experimental part, we deployed an inter-domain network topology on the Heterogeneous Experimental Network (HEN) [74] testbed, with the aim of testing our route leak identification techniques in a scenario that can realistically support the data-plane part. The results from our tests show that an AS is able autonomously detect route leaks in different scenarios with a high success rate using the CP, BFB and R-BFB, especially, when the three techniques are combined and used together.

7.1 Simulations Framework

In order to utilize event-driven simulations at a large scale for evaluating the proposed route leak detection techniques, a number of practical decisions were needed for our testing framework, such as considering a scaled down Internet-like topology. The Internet topologies used in our testings were extracted from the global-scale ARK's Internet graph [73]. The graph reduction technique that we used for passing from the complete ARK's Internet graph to a smaller AS topology was based on [120], and the goal in this process was twofold. Firstly, we tried to preserve some of the essential topological properties of the complete Internet graph supplied by ARK, so that the results obtained can be reasonably extrapolated to larger topologies. Secondly, and most importantly, we ensured that the graph used was actually a subgraph of the ARK graph. In other words, all the domains, links, and the AS relationships used in our simulations, are actually present in ARK's Internet graph [73].

For preserving the essential topological properties of complete ARK Internet graph, we make use of the power-law (cf. 7.1) derived in [121].

$$o_i \propto r_i^{\gamma} \tag{7.1}$$

where, r_i denotes the rank of node i, and γ denotes the rank exponent of the power-law.

We recall that the rank exponent γ is one of the fundamental properties of scale-free graphs, such as the AS-level topology of the Internet. The rank exponent γ is defined as the slope of the line obtained from the application of linear regression on the (o_i, r_i) pairs in a log-log plot. Yannuzzi et al. [122] shows that a subgraph $G_S \subset G_A$ preserves the outdegree-rank properties of graph G_A if the approximation by linear regression for subgraph G_S in a log-log plot, is the same as for G_A. We used two complete Internet topologies from ARK, one from the year 2009 (referred as Topology-2009) and the other from the year 2013 (referred as Topology-2013). The two topologies were carefully scaled-down such that the rank exponent γ is approximately same as of their respective super Internet graph, as given in Fig. 7.1 and Fig. 7.2.

Due to the constraints on the scale for carrying out event-driven simulations, we considered a single router per AS in our simulations. Observe that, RLD techniques are applied using analytics on the RIBs of all the border routers in the AS and the external advertisements that they receive, so the internal transit routes and the iBGP implications are not expected to

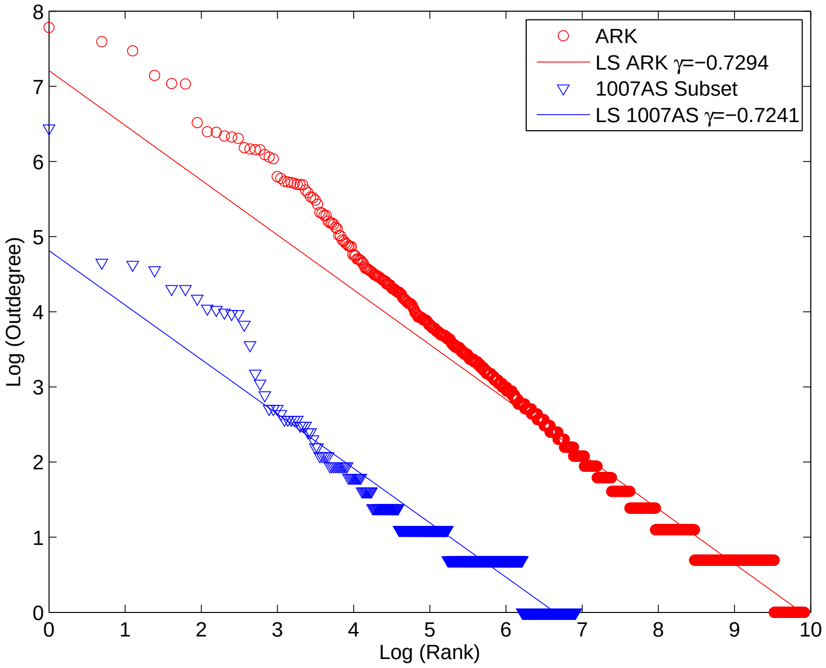


Figure 7.1: The outdegree o_i as a function of the rank r_i for Topology-2009, sorted in order of decreasing outdegree (log-log plot).

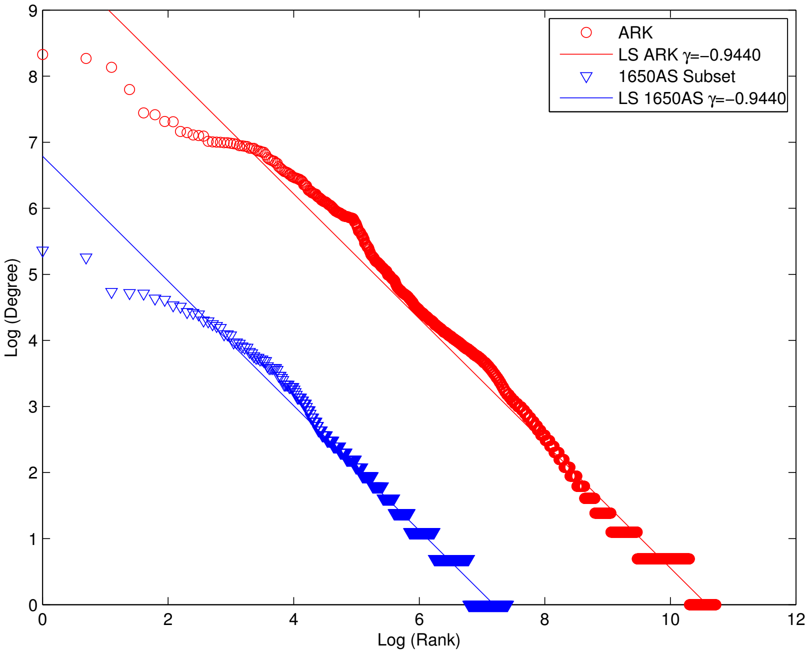


Figure 7.2: The outdegree o_i as a function of the rank r_i for Topology-2013, sorted in order of decreasing outdegree (log-log plot).

considerably influence the detection results. Indeed, with multiple border routers per AS, i.e., with more than one RIB belonging to same AS but with different Internet route views, we would actually expect improvements in the detection rates.

The simulations were setup and run using the network simulator NS2 [118] along with BGP++ [119]. BGP++ is based on the standard GNU Zebra routing software and complements NS2's lack of native BGP capabilities. All the event driven route leak simulations and detections were conducted using the high performance server cluster of the Computer Architecture Department (DAC), at Barcelona Tech (UPC).

Route Leak Scenarios

Considering the fact that, from the victim's perspective, a route leak may only be initiated by a customer or a peer neighbor, we have categorized the route leak scenarios into two groups:

- **Customer Route Leak (CRL) scenario:** this scenario includes all possible combinations of route leaks in which an AS leaks its provider's or peer's routes toward other providers (see Fig. 7.3(a)).
- **Peer Route Leak (PRL) scenario:** this scenario consists of all possible combinations of route leaks in which an AS leaks its provider's or peer's routes toward other peers (see Fig. 7.3(b)).

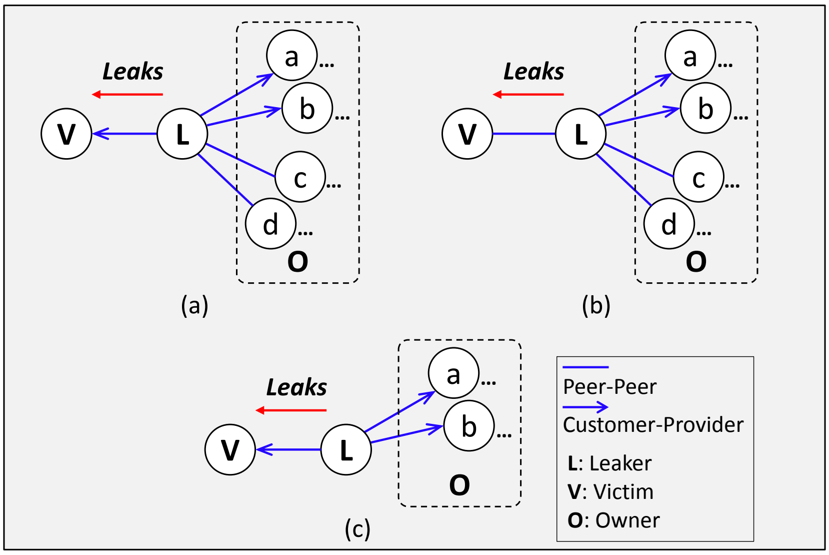


Figure 7.3: Categories of route leak scenarios: (a) Customer Route Leaks (CRLs); (b) Peer Route Leak (PRLs); (c) Stub Route Leaks (SRLs).

Apart from the above two route leak scenarios, we identify a specific case of customer route leaks, namely:

- **Stub Route Leak (SRL) scenario:** this scenario defines all possible combinations of route leaks in which a multi-homed stub AS leaks a provider route toward other providers (see Fig. 7.3(c)). We consider an AS which has no customer or peer ASes and has at least two distinct provider ASes as a multi-homed stub AS.

The classification of route leaks into CRL, PRL and SRL, will allow us to analyze the performance under different route leak scenarios, and will also facilitate understanding of the results obtained.

7.2 Topology-2009: Simulations and Results

The scaled down version of complete Topology-2009, consisted of 1007 ASes and 1753 distinct inter-domain links. In the rest of the chapter, we refer to the scaled-down version of the topology as Topology-2009.

With the business relations among neighbor ASes in Topology-2009, we first inferred the maximum number of possible route leak scenarios. We observed that 6630 different route leak cases could be studied. Thus, to evaluate the effectiveness of our RLD techniques, a total of 6630 different simulations were conducted over the Topology-2009, covering one route leak scenario per simulation.

It is worth mentioning that in some of the route leak cases, the route leak was harmless. That is, either it did not occur or even if it 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 such a route leak scenario is depicted in Fig. 7.4. 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.

We observed that out of the total 6630 route leak cases for Topology-2009, 4409 were harmful route leaks, i.e., the route leak succeeded in poisoning the RIB of the victim AS V. In the rest of this chapter, we only present the route leak detection results of the harmful route leaks in different scenarios. For the considered topology, the total number of possible CRLs add up to 2041, out of which 1292 also belong to the SRL scenario. The total number of possible PRLs are then 4409 – 2041 = 2368. The methodology used is as follows. As shown in Fig. 7.3, each route leak scenario involves three participants: 1) the Leaker AS (L); 2) the Victim AS (V); and 3) the Owner AS (O). The Leaker is the AS's router that is configured to leak the routes. The Victim is the AS that will receive the leaked routes, and the Owner is the AS whose routes were improperly advertises toward the Victim. In each simulation, the BGP protocol was initially configured according to the policies and relationships with its neighbors as obtained from ARK (i.e., compliant with the valley-free rules), and it was allowed to converge. This is important to ensure the utilization of valley-free RIBs in the initial state. Once BGP converges, the detection process is activated on the victim AS V. Once our RLD techniques are operative, we explicitly reconfigured one AS's BGP router (the leaker L) with export rules that violated the conceded relationship found in ARK—all this was done during the simulation runtime. Clearly, as new BGP updates are received, the detection technique will be analyzing them.

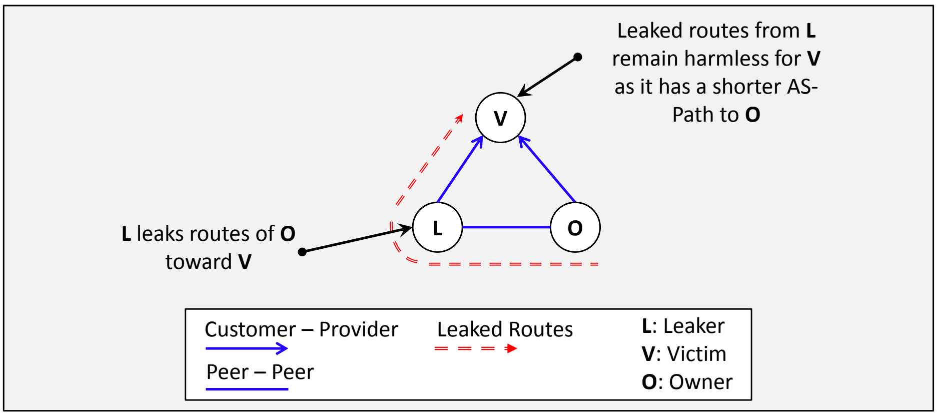

Leaked routes from L remain harmless for V as it has a shorter AS-Path to O
V
L
O
L leaks routes of O toward V
Customer – Provider
Peer – Peer
Leaked Routes
L: Leaker
V: Victim
O: Owner

Figure 7.4: General representation of one of the harmless route leak scenarios.

7.2.1 Cross-Path Technique: Results and Analysis

The overall results obtained using CP detection technique in the different route leak scenarios are summarized in Table 7.1. We observe that for all the CRL scenarios evaluated, the CP detection technique achieves a success rate of 93.34%; and, within this scenario, the success rate for the SRL cases increases up to 98.14%. On the contrary, for the PRL scenarios, the results obtained are considerably low, achieving only an overall success rate of 38.77%.

In order to provide further insight into the results, we have split the outcomes into two subcategories, depending on the Owner type. First we consider the case when L leaks the routes learned from one of its providers only, i.e., O is a provider of L (see Fig. 7.5(a)). In this case, the set of routes leaked from L to V might be potentially large, since it may include O's own routes, as well as its provider, peer and customer routes. In the second subcategory, we consider the case when L leaks the routes learned from one of its peers only, i.e., O is a peer of L (see Fig. 7.5(b)). In this second case, the set of leaked routes may include O's own routes, as well as those of its customers. Table 7.2 summarizes the detection results for the CRL, PRL, and SRL scenarios based on this classification of Owner type.

The number of cases in which O is a provider of L is 1830 for the CRL scenarios (i.e., CRL(Pr)), and 410 for the PRL scenarios (PRL(PR)), out of which the CP technique detects 97.98% and 99.76% of the leaks, respectively. We can observe that for the SRL scenarios, the RLD technique performs even better within the CRLs, with a detection success rate of 98.14%. This is because in case, the leaker is a stub AS whereas the victim, being a provider, is topologically well positioned with a broader view of the Internet and the different routes to reach the leaker. Thus, the CP technique is able to detect most of the leak cases in this scenario. Observe that the ultimate number of routes leaked by L to V will actually depend on the BGP decision process at L, and the export rules configured toward V. In general, when O is a provider of L, the routes leaked may provide reachability to a broad transit-like block of the Internet, hence

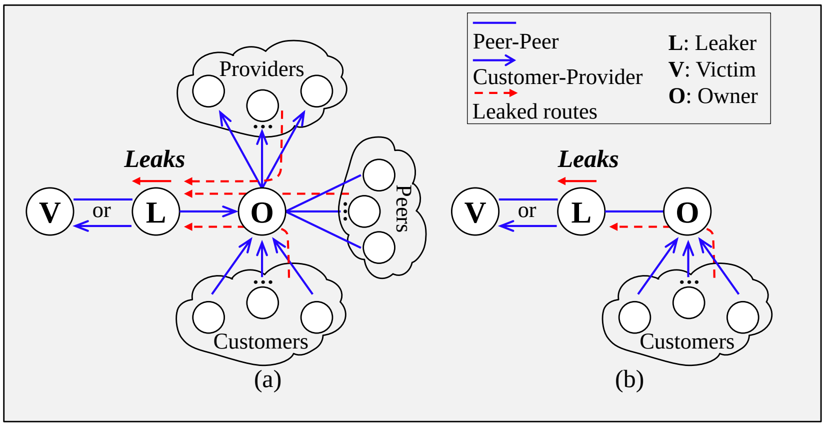


Figure 7.5: (a) Leaks toward V when O is a Provider of L; (b) Leaks toward V when O is a Peer of L.

Leak Scenarios	Total # Leaks	# Leaks Detected	% Leaks Detected
CRL	2041	1905	93.34%
PRL	2368	918	38.77%
SRL	1292	1268	98.14%

Table 7.1: Cross-Path technique: Overall route leak detection results for different route leak scenarios on Topology-2009.

observance of cross-paths is more likely. On the other hand, the number of CRLs in which *O* is a peer (i.e., CRL(Pe)) is 211, while for the PRL scenarios (PRL(Pe)) is 1958. The detection results for the CRL, and PRL scenarios in this case are, 53.08%, and 26.00%, respectively. The main reason for this low performance is that, when *L* leaks the routes learned from a peer, the number of routes announced are far less than when the leaked routes are from a provider. These routes provide reachability to a narrower stub-like block of the Internet compared to the former case. Thus, observance of cross-path is less likely, and as reflected in Table 7.2, poor detection success rates are obtained in this case.

The main conclusion that can be drawn is that CP detection technique has high accuracy when the victim is detecting leaks initiated by its customer ASes, but a more creative approach is needed for detecting route leaks initiated by a peer. Indeed, the challenge arises when *L* is a peer of *V*, and the routes leaked by *L* belong to one of its peers *O*. This is precisely the motivation for the BFB detection technique.

7.2.2 BFB Technique: Results and Analysis

We recall that BFB route leak detection technique specifically targets PRL scenarios where *L* leaks its peer routes toward *V*. Table 7.3, shows the impact of using BFB detection technique on the PRL scenarios. The results show that BFB can actually duplicate the success rate of route leak detection for PRLs. We can contend that, autonomous RLD techniques, using solely analytics on the routing information available at an AS is sufficient for detecting the large majority of the route leaks initiated by a neighbor, especially, when they are not the result of

Leak Scenarios	**Cross Path Detection**		
	# Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	1830	1793	97.98%
CRL (Pe)[b]	211	112	53.08%
PRL (Pr)[a]	410	409	99.76%
PRL (Pe)[b]	1958	509	26.00%

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

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

Table 7.2: Cross-Path technique: Detailed route leak detection results for different route leak scenarios on Topology-2009.

Leak Scenarios	# Leaks	CP	CP + BFB
CRL	2041	93.34%	93.34%
PRL	2368	38.77%	76.90%
SRL	1294	98.14%	98.14%

Table 7.3: Detection results including CP + BFB for Topology-2009

premeditated and elaborated attacks—BFB will clearly not fool a prepared attacker.

7.3 Topology-2013: Simulations and Results

Topology-2013 is extracted from a subgraph of ARK's recent Internet graph (2013) [123]. The topology was scaled down to 1650 ASes and 3744 inter-domain links while maintaining essential properties of the original Internet graph, as detailed in Section 7.1. For simulation purposes we again used NS2 [118], along with BGP++ [119]. Similar to Topology-2009 simulation setup, we considered a single router per AS and each AS's BGP router was configured according to its policies and relationships with its neighbors in the extracted topology. As a result, we were able to simulate and test the BGP behavior in the latest Internet-like topology for different route leak scenarios and evaluate their impact—observe that our approach of using a subgraph of ARK's Internet graph means that the topology that we used is actually part of the Internet.

For Topology-2013, 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. Next, we present the route leak detection results of the harmful ones for different detection techniques.

7.3.1 Cross-Path Technique: Results and Analysis

Table 7.4 shows the simulation results of CP route leak detection technique for the harmful route leaks. From the perspective of the extended classification of the route leaks, we observe a similar performance trend as in the Topology-2009 simulations results. That is, 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 performs poorly i.e., 23.73% for CRL (Pe) and 5.90% 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 Detection**		
	# Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	4773	4492	94.11%
CRL (Pe)[b]	3974	943	23.73%
Total CRL	8747	5435	62.13%
PRL (Pr)[a]	5406	5044	93.30%
PRL (Pe)[b]	2998	177	5.90%
Total PRL	8404	5221	62.12%

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

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

Table 7.4: Cross-Path technique: Route leak detection results for different route leak scenarios on Topology-2013.

7.3.2 BFB Technique: Results and Analysis

As shown in the Table 7.5, the BFB detection technique improves the route leak detection success rate of PRL (Pe) from 5.90% to 34.95%. The BFB technique proves to be very useful in PRL (Pe) case as it improves the route leak detection success rate seven-fold compared to the CP technique. However, we consider 34.95% yet low detection success rate and the main reason is that, when *L* leaks the routes learned from a peer, the number of routes announced are far less than when the leaked routes are from a provider. As mentioned earlier, these routes provide reachability to a narrower stub-like block of the Internet compared to the former case, thus, observance of cross-path is less likely.

Moreover, we observe that the overall success rate of CRL in Topology-2013 is less than what we noticed in Topology-2009. For the CRL case, the overall success rate is 93.34% for Topology-2009 and 62.13% for Topology-2013. This is mainly due to the fact that there are many more peer-peer links in Topology-2013 than in Topology-2009. The Topology-2013 has 72% customer-provider links and 28% peer-peer links (out of the total 3744 links) whereas Topology-2009 has 93.1% customer-provider links and only 6.9% peer-peer links (out of the

Leak Scenarios	**Cross Path + BFB**		
	# Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	4773	4492	94.11%
CRL (Pe)[b]	3974	943	23.73%
Total CRL	8747	5435	62.13%
PRL (Pr)[a]	5406	5044	93.30%
PRL (Pe)[b]	2998	1048	34.95%
Total PRL	8404	6092	72.48%

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

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

Table 7.5: Route Leak Detection results including CP + BFB for Topology-2013

total 1753 links). As the percentage of peer-peer links in Topology-2013 is four times more than Topology-2009, the route leak scenarios involving peer, as owner AS *O*, increase as well. Following the same line of reasoning for the low performance of CP technique (the BFB technique does not target CRL (Pe)) for CRL(Pe) as described above, the overall success rate of CRL is impacted adversely for Topology-2013.

However, the overall success rates of PRL in Topology-2009 (i.e., 76.90%) and in Topology-2013 (i.e., 72.48%) are relatively close. This is because, although there are many more PRL(Pe) cases in Topology-2013 than in Topology-2009, BFB technique performs well in both cases as it specifically targets the PRL(Pe) case.

7.4 Experimental Framework

In order to validate our RLD techniques, especially R-BFB, in a real environment, we deployed it in a testbed using a network topology with the potential to produce many different route leaks. For this purpose we used the Heterogeneous Experimental Network (HEN) [74] testbed, where we set up a network composed of 17 AS. The particular topology used is detailed in Fig. 7.6, where each AS is represented by a physical node. Each BGP router composing the topology was implemented using a Debian Linux with the well-known Quagga routing suite. In this topology, we were able to anticipate 35 CRL and 57 PRL possible scenarios. Hence we ran a total of 92 different experiments, each with one route leak occurrence. Initially, as determined by our hypotheses, all the nodes were connected and configured in line with the valley-free rules, hence 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 the *V* AS used the RLD techniques to detect the route leak based on the available BGP information.

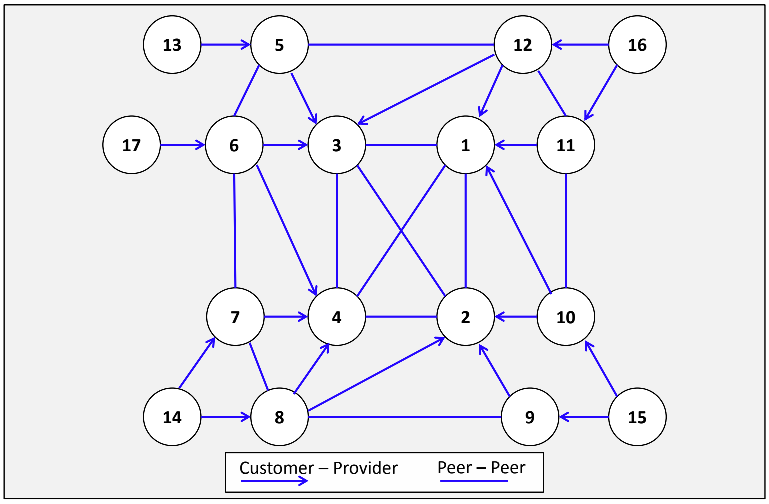


Figure 7.6: Network topology deployed at HEN for experiments.

7.4.1 Cross-Path Technique: Results and Analysis

Out of the 92 different route leak scenarios, we were able to rule out 14 leaks that were harmless. For the remaining 78 harmful route leaks, there are 21 CRL and 57 PRL route leaks. Table 7.6 shows the results obtained with the CP route leak detection technique for the 78 route leak experiments. CP performs relatively better in PRL cases with 78.9% as compared to CRL cases with 66.66% detection success. As shown in Table 7.6, in line with further classification, we observe that CP route leak detection performance is 100% in both CRL (Pr) and PRL (Pr), whereas for PRL (Pe), it detects 66.66% of the route leaks, but, for CRL (Pe) cases, simple CP identification techniques totally fails. The justification of these results is similar to the one given for the simulation results of CP technique for Topology-2009 and Topology-2013.

We observe that for the case of CRL (Pe), CP detects 0% route leaks in the experiments and 53% and 23.73% in the Topology-2009 and Topology-2013 simulations, respectively. This is because, all the 112 CRL (Pe) cases (for Topology-2009) and 943 CRL (Pe) cases (for Topology-2013), where CP detects the route leak in the simulations, the V and O have a direct peer link between them. V can infer that the O has a peer relation with the provider of its provider which is in violation of hypothesis $\mathscr{H}.2$ and thus detects the leak. Whereas, for the 7 CRL (Pe) cases in the experiments, the V and the O didn't have a direct link.

Furthermore, We also observe that for the PRL (Pe) cases, CP performs better in experiments (66.66% success rate) as compared to in simulations (26% and 5.90% success rate in Topology-2009 and Topology-2013, respectively). This is because the 24 leaks the CP detects, in the experimental case, involve the "Tier-1" of the topology (See Fig. 7.6), i.e., AS 1, 2, 3 and 4, which allows the detection of the leak. Thus, the detection was due to observance of Tier-1 ASes in the route, not because of cross-path observance which was not the case in respective simulations.

Leak Scenarios	**Cross Path Detection**		
	# Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	14	14	100%
CRL (Pe)[b]	7	0	0%
Total CRL	21	14	66.66%
PRL (Pr)[a]	21	21	100%
PRL (Pe)[b]	36	24	66.66%
Total PRL	57	45	78.90%

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

Table 7.6: Cross Path Detection: Experimental results.

7.4.2 BFB Technique: Results and Analysis

For the BFB case, we use the same experimental setup and route leak scenarios as described in Section 7.4.1 but with the addition of BFB detection technique along with the CP detection technique. Table 7.7 shows the route leak detection results for CP and BFB combined. We can observe that the BFB detection technique improves the route leak detection success rate for PRL (Pe) to 100%. As expected, the BFB detection technique does not help in CRL (Pe) route leak cases as the BFB technique fails when V is the provider of L. This is because in event of the benign fool back advertisement from V, L would prefer the peer route from O.

Leak Scenarios	**Cross Path + BFB Detection**		
	# Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	14	14	100%
CRL (Pe)[b]	7	0	0%
Total CRL	21	14	66.66%
PRL (Pr)[a]	21	21	100%
PRL (Pe)[b]	36	36	100%
Total PRL	57	57	100%

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

Table 7.7: Cross Path + BFB Detection: Experimental results.

7.4.3 R-BFB Technique: Results and Analysis

The inclusion of data-plane intelligence provides extra pair of eyes for detection of route leaks in different scenarios. For the same set of route leak experiments as in Section 7.4.1 and Section 7.4.2, Table 7.8 shows the route leak detection results of R-BFB technique on top of CP and BFB. For the topology we implemented (cf. Fig. 7.6) for real-time experiments, we are able to detect all possible route leaks with the help of all three RLD techniques including CP, BFB and R-BFB. We contend that intelligence from both control-plane and data-plane provide enough information to detect all possible route leaks.

We did not perform R-BFB experiments using our simulation environment because NS2 does not allow emulation of data-plane. Furthermore the BGP++ implementation in NS-2 only simulates the BGP control plane, but without enforcing the routing rules to the nodes, thus not allowing the generation of regular traffic through the paths as learned by BGP. Moreover, we contend that large scale event driven emulation of control-plane as well as data-plane, for a network consisting of more than 1650 ASes and 3500 inter-domain links, demands much more effort and thus is considered as future work. Hence for R-BFB, we confine our study to real-time experiments only.

Leak Scenarios	**Cross Path + BFB + R-BFB Detection**		
	# Leaks	# Leaks Detected	% Leaks Detected
CRL (Pr)[a]	14	14	100%
CRL (Pe)[b]	7	7	100%
Total CRL	21	21	100%
PRL (Pr)[a]	21	21	100%
PRL (Pe)[b]	36	36	100%
Total PRL	57	57	100%

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

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

Table 7.8: Cross Path + BFB + R-BFB Detection: Experimental results.

7.5 Conclusion on Route Leak Detection

In this chapter we evaluated the performance of the three RLD techniques including CP, BFB, and R-BFB, using simulations as well as real-time experiments. We utilized scaled down actual Internet topologies so that the route leak detection results obtained can be considered valid for real world larger topologies as well.

We observed that the CP route leak detection technique performs exceptionally well for CRL (Pr) and PRL (Pr) route leak scenarios. The success rate of CP technique is more than 90% for both, CRL (Pr) and PRL(Pr), route leak cases. For CRL (Pr), the set of routes leaked from L to V might be potentially large, since it may include O's own routes, as well as its provider, peer and customer routes, hence increases the chances of observing a cross-path at the victim AS. However, the main cause of the excellent success rate in both cases is because the victim, being a provider, is topologically well positioned with a broader view of the Internet and the different routes to reach the leaker. On the other hand, for CRL (Pe) and PRL (Pe) route leak scenarios, the CP shows low performance. The prime reason for this low performance is that, when L leaks the routes learned from a peer, the number of routes announced are far less than when the leaked routes are from a provider. Furthermore, these routes provide reachability to a narrower stub-like block of the Internet, thus the observance of cross-path is less likely.

The BFB route leak detection technique uses control-plane information and a fool back advertisement on the control-plane to detect route leak in PRL (Pe) cases. The results show that, for Topology-2009, Topology-2013 and for the experimental topology, the route leak detection success rate increases many-fold. The R-BFB route leak detection technique targets the CRL (Pe) and PRL (Pe) route leak cases by using the information both at control-plane and data-plane. The R-BFB technique enables us to successfully detect all the route leaks for different scenarios in our experimental topology.

In general, the results from our tests show that an AS is able autonomously detect route leaks in different scenarios with a high success rate using the CP, BFB and R-BFB, especially, when the three techniques are combined and used together.

7.6 Integrating Security Mechanisms in Inter-Domain Routing

The security of inter-domain routing poses multiple challenges, and any proposed security mechanism needs to be thoroughly and exhaustively analyzed and evaluated—keeping in mind all the different theoretical, operational and practical aspects of the problem. As mentioned earlier, several past security solutions could not be widely adopted mainly because they required changes in the BGP protocol for their functionality and implementation. 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, while avoiding collateral burden and causing minimum entropy. For example, the BGPSEC concept of using an inter-locked chain of signatures may be a good solution for securing route propagation in BGP, however, embedding it directly in the BGP protocol may not be the best way to implement it. One potential direction to explore is the outsourcing of BGP security, that is, to decouple security from the BGP protocol with the aim of minimizing the impact on the routers' installed base. A promising approach for outsourcing security is through Software Defined Networks (SDN) [124]. SDN enables to outsource control functions of an SDN-enabled network element to external applications, by either exposing the available capabilities through proprietary APIs or simply through a standard protocol such as OpenFlow [125]. The SDN approach offers an attractive alternative for developing outsourced security mechanisms, which could be materialized by means of an overlay network of distributed SDN controllers. However, the OpenFlow protocol lacks native support for the BGP chores. Nevertheless, under the OpenDaylight project [126], efforts are underway to develop the Southbound plugin for SDN controllers to support BGP functionalities, but these efforts are still in incubatory stages. Thus, one pragmatic option is to develop our own tool which allows us to experiment the outsourcing of security chores (e.g., RLD techniques or BGPSEC) away from BGP on non-proprietary routers, such as the open source Quagga router [127].

In light of this, we present OPENER[1] [128]—a programmable platform that provides open access to the capabilities of an open source router—specifically Quagga [127]—thereby enabling the creation of out of the box applications that can extend the existing features on Quagga routers. In order to offer an open and programmable environment, OPENER provides a set of interfaces, where the accessible internal features, e.g., routing protocols, interface management, and so on, are exposed to third-party applications. Thus, OPENER can be used as a tool that allows security outsourcing on routers to third-party applications. However, OPENER requires a few modifications on the BGP protocol, but we contend that these modifications are minor. For instance, only 88 lines of the BGP-4 code were modified in our prototype, so as to support the interception and re-injection of BGP messages, and thereby enable the outsourcing of security. For more details on the OPENER project, the interested readers are referred to Appendix C.

[1] The OPENER project was an internal project of the Advanced Network Architectures (ANA) research group at UPC. This project was initiated by Dr. Marcelo Yannuzzi and the implementation was led by Dr. René Serral-Gracià. The contributions of the author of this thesis to the OPENER project included development of backend modules and technical writing chores.

Third Part Part III

8 Locator/Identifier Separation Protocol (LISP)

In this part of the thesis, we focus our attention on the security aspects of the Locator/Identifier Separation Protocol (LISP) protocol. First, we provide a brief overview of the LISP architecture and functionality along with the existing security features. Then, we narrow down our focus on the security issues related to the lack of resource authorization mechanisms in the LISP protocol. Finally, we detail our recommendations to overcome the targeted security problems of LISP.

This chapter provides a brief overview of the Locator/Identifier Separation Protocol (LISP) protocol in order to better understand the LISP control-plane security issues and the proposed solution in the next two chapters, respectively.

8.1 Background

The current IP-based addressing schemes, namely IPv4 and IPv6, employ unified functionality of indicating the identity and location of a node in the Internet. That is, the same address is used to identify as well as to locate the node. This dual capability is the cause of many concerns with the ever increasing size of the Internet including scalability of the routing system and exhaustion of IP addresses in case of IPv4. Furthermore, the current address scheme act as an impediment to the several network features deemed essential for the future Internet including mobility, resilient communications, multihoming, efficient routing system, etc. Hence, separating the address spaces for identity and location seems to be the rightful evolution for the Internet addressing schemes.

The ID/Locator (ID/LOC) split addressing architecture, first introduced by [129], is based on the concept of splitting the Internet addressing scheme in to two address spaces, *identifier* and *locator*. That is, the identifier reveals the *who* and locator indicates the *where*. The identifier address space uniquely identifies a host (e.g., computer, mobile device, router, or a virtual object) in the network whereas the locator address space indicates the location of the host. The location of the host signifies as to which network the host is attached in the Internet. These two disjoint address spaces require a mechanism of translation from one to other. There are two predominant methodologies for address translation, *Map-and-Encap* and *Address Rewriting*. The former technique employs tunneling techniques whereas the latter uses address rewriting procedures to achieve address translation between the two logical address spaces. The examples of *Map-and-Encap* include [130], [131], [132], [133], and [134] whereas [135], [136] and [137] are examples of *Address Rewriting*. In order to resolve *who is where* currently, a global binding/mapping system is used which contains up-to-date identifier-locator mappings. Furthermore, the ID/LOC address split can be implemented either by modifying the host protocol stack (i.e., Host-based) or by equipping network elements with additional specific capabilities (i.e., Network-based). Ramirez et al. [138] provides a detailed survey on the several proposed addressing schemes based on ID/LOC split addressing architecture.

8.2 Locator/Identifier Separation Protocol (LISP)

One of the most prominent ID/Loc split based addressing scheme is Locator/Identifier Separation Protocol (LISP) [2]. LISP is a Cisco initiative promoted as an open standard through IETF LISP Working Group [9]. It separates the location and identity information of a device into Routing Locators (RLOCs) and Endpoint Identifiers (EIDs). LISP supports provider independent and globally unique Identifier addresses, and employs a network-based Map-and-Encap scheme, along with an Identifier-to-Locator Mapping System to bind the two address spaces. Another important feature is that LISP is address family agnostic, so the Map-and-Encap and Decap processes can handle mixes of IPv4 and IPv6 indistinctively. These features have made it highly flexible, and therefore, it is considered an enabler for a variety of applications.

LISP achieves the Map-and-Encap (i.e., the address translation) and Decap with the help of two border network elements, Ingress Tunnel Router (ITR) and Egress Tunnel Router. The ITR border router is responsible for performing the map-and-encap procedure for the IP packets received from the hosts (i.e., EID holders) within the domain. For every packet destined for an alien domain, the ITR consults the mapping system for an up-to-date EID-to-RLOC mapping of the EID address present in the destination field of the IP header of the received packet. Based on the EID-to-RLOC mapping, the ITR prepends another IP header in the packet with its own RLOC address as the source address and mapped RLOC address, from the EID-to-RLOC mapping query, as the destination address before pushing it out toward the Internet. That is the ITR encapsulates the EID IP header inside the RLOC IP header. The mapped RLOC address is in fact the RLOC address of the ETR of the destination domain. When the ETR receives a packet destined for itself, it strips off the outer IP header (i.e., Decap) containing the RLOC addresses and pushes the packet, now with one IP header containing the EID addresses, in to the destination domain to reach its target. This map-and-encap procedure is also referred to as LISP data-plane operation. Fig. 8.1 illustrates how an IP packet moves from one LISP site to another.

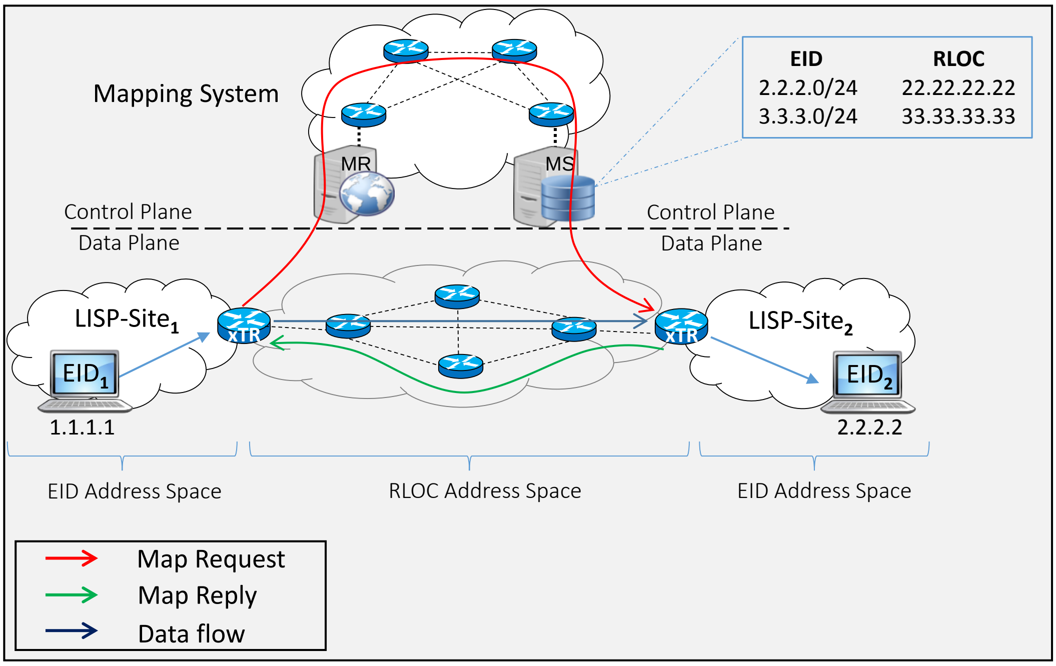


Figure 8.1: LISP overview.

The process of retrieving EID-to-RLOC mappings through a mapping system is termed as LISP control-plane operation. The LISP data-plane is not dependent on a particular mapping system and remains agnostic of the mapping system as long as the messages to query and receive responses from the mapping system remain compliant with the LISP baseline specifications. The two mapping systems considered by LISP include LISP Alternative Logical Topology (LISP-ALT) [139] and LISP Delegated Database Tree (LISP-DDT) [140]. LISP-ALT employs routing protocol for its operations whereas LISP-DDT uses a hierarchical distributed database infrastructure, similar to the DNS system. The ITR requests the mapping system for an EID-to-RLOC mapping lookup by sending Map Request message to the Map Resolver (MR). The MR alerts the Map Server (MS) responsible for containing the particular EID-to-RLOC mapping for the mapping query. The respective MS directs the mapping query to the corresponding ETR which owns the particular RLOC address. The ETR sends a Map Reply message directly to the querying ITR. The ITR caches the EID-to-RLOC mappings to avoid consulting the mapping system every time.

In Fig. 8.1, EID_1 in $LISP-Site_1$ sends an IP packet to EID_2 in $LISP-Site_2$. For that purpose the ITR of $LISP-Site_1$ initiates a Map request for which it receives a Map reply from ETR of the destination domain, $LISP-Site_2$. Once the EID-to-RLOC mapping is learned, $LISP-Site_1$'s ITR performs map-and-encap before the IP packets is on it way toward the destination.

It is worth mentioning that an ETR has to register the EID prefixes in its domain along with the associated RLOC addresses on a MS in the mapping system before those EID prefixes become reachable. The registration could be done against a single or a set of RLOC addresses, thus enabling global reachability. As currently defined in [2], this map registration process is a static procedure based on manual configurations that need to be set in advance. These configurations have to be done both on the ETRs and on the Map Server. Once the manual configurations are in place, each ETR will attempt to register its mappings with the Map Server. It does so by sending a Map Register message containing the list of EID-prefixes it claims to represent along with authentication data. The MS 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.

From the security perspective, LISP defines few intrinsic security mechanism as a first line of defense including Map Request and Map Reply nonces, Map Register authentication, and EID source check.

The ITR inserts a pseudo-randomly generated 64-bit nonce in the Map Request message which the ETR must copy in the Map Reply message in order for ITR to accept it as a genuine reply to a mapping lookup request it made earlier. LISP also suggests to use a 24-bit nonce for sending IP packets on the data-plane, i.e., the ITR inserts different nonces for different destinations in the IP packet during the map-and-encap procedure. The use of nonce attempts to provide some level of integrity to Map Request and Map Reply messages however it can be undermine by on-path as well by off-path attackers with use of brute force techniques.

The Map Register authentication refers to the authentication data included in the Map Register message. The authentication data provides a minimum security level in the map registration process. The current specification define configuring the MS and the ETR with a shared secret key to produce the authentication data using Message Authentication Code (MAC) algorithms. In this way the map register message can be authenticated. The digest information is encrypted by the shared key on the ETR and only the MS who has the same key is able to verify the authenticity of the registration message, and allow the EID-prefix registration.

Although, LISP specifications recommend that an ITR should verify the EID source of a received packet but it does not lay out any particular procedure to achieve the task.

8.3 LISP Security

The intrinsic security mechanisms of the LISP protocol leave the door wide open for several possible attacks. As mentioned earlier, if an attacker can guess the nonce of a Map Request by using brute force techniques, then it can disseminate false EID-to-RLOC bindings toward the ITR. That is, the nonce only provided a confirmation that a certain Map Reply message is in response to a particular Map Request message initiated earlier, however it provides no integrity about the content of the Map Reply message. Consequently, a valid ETR can launch a rogue attack by claiming more EID prefixes than what it owns in the Map Reply message to influence packets destined for the victim EIDs.

The existing *pre-shared key* security mechanism for map registration process between the ETR and the MS is not robust enough to mitigate rather simple attacks, such as RLOC address spoofing. This is because current LISP security mechanisms lack 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 are essential for the practical feasibility of mobility and roaming scenarios in LISP. Saucez et al. [141] provide a detail discussion on the vulnerabilities of LISP.

Apart from the internal security mechanisms of LISP including Map-Register authentication and Map-Request nonces, an additional protocol, called LISP-SEC, has been developed [13]. LISP-SEC is built around the already existing security mechanisms of LISP, and provides security measures for the *Map-Request* and *Map-Reply* messages in the LISP control-plane. A One Time Key (OTK) is securely exchanged between the ITR and Map-Resolver and another OTK between Map-Server and ETR. These OTKs are used to produce HMACs to provide origin authentication, integrity and anti-replay protection, which consequently mitigates Man-in-the-Middle attacks as well.

Prior to the secure exchange of OTK, a predefined key need to be shared between the ITR and Map-server and between the ETR and Map-server as well. That is, the corresponding shared key is used to encrypt the OTK before exchanging it securely. The requirement of out-of-band exchanges of different shared keys puts a question mark on the scalability of this scheme, furthermore a compromised shared key can put the whole security mechanism at risk. Moreover, the LISP-SEC protocol targets to provide origin authentication, integrity and anti-replay protection to the Map Request/Reply message exchanges, however it does not address the authorization requirements mentioned above. That is, LISP-SEC protocol neither has any mechanism to confirm that a particular ETR is authorized by the EID address holder to make that map registration nor any mechanism to enable MS to ensure that the requesting ETR is legitimately allowed to use that particular RLOC address for registering the EID prefixes. In the next chapter, we look at different attacks that are feasible in the absence of resource authorization.

9 LISP Security Issues

In this thesis, we narrow down our focus on LISP security issues which arise due to lack of proper end-to-end resource authorization mechanisms. The term resource authorization refers to a procedure which could confirm that a certain entity is legitimately entitled to use a certain resource such as EID prefix or RLOC address. In this regard, we illustrate two important security issues in LISP, namely, RLOC spoofing and lack of global EID authorization. A thorough LISP threat analysis could be found in [141]. The main reason for focusing on the above mentioned security issue is that they highlight the vulnerabilities which undermine LISP functionality right from the beginning, i.e., they jeopardize its normal operations as well as its potential as a technology enabler. These vulnerabilities enable the registration of false 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.

9.1 LISP Control-Plane Attacks

9.1.1 RLOC Spoofing

The RLOC spoofing attack occurs when a malicious ETR claims false ownership of a RLOC address or a set of RLOC addresses during the map registration process. The Map Server (MS) contains the mapping entries consisting of EID-to-RLOC pairs. A malicious ETR can send a Map Register request toward a MS including an incorrect RLOC and depreciate the integrity of the mapping entry. In order to counter that, the MS needs to ensure that a certain ETR is authorized to use a particular RLOC address for registering an EID prefix. The lack of such assurance can lead to different attacks by a malicious ETR, such as DoS attacks by traffic flooding.

The RLOC spoofing scenario is illustrated in Figure 9.1. The malicious ETR from $LISP-Site_1$ carries out a map registration targeting $LISP-Site_2$ by claiming 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, in this scenario at

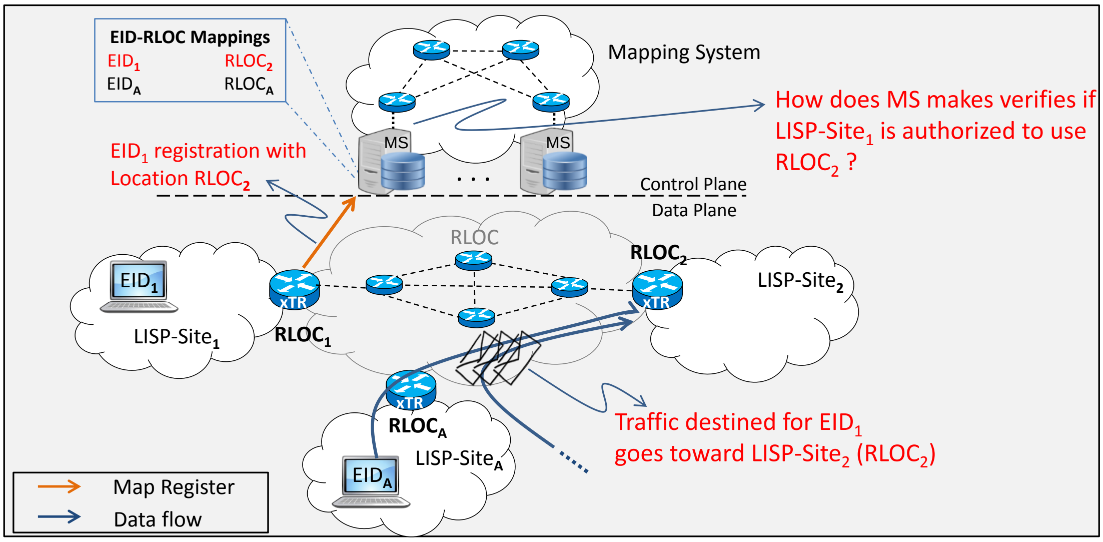


Figure 9.1: RLOC spoofing attack in LISP.

$LISP - Site_2$. The RLOC spoofing attack succeeds because the MS has no defined mechanism to verify if a particular ETR is authorized to claim a RLOC address, hence the mapping entries are compromised. Any further queries for the EID_1 prefix's locator will retrieve the wrong $RLOC_2$. Apart from the lack of RLOC authorization, the MS also does not have any mechanism to verify if an ETR is authorize to carry out map registration on behalf of a particular EID prefix. This can lead to over claiming of EID prefixes during the map registration process and corrupt the information in the mapping system.

In summary, LISP does not define a mechanism to verify the authorization of RLOCs and EID prefixes to ETRs. Any ETR can claim any RLOC or EID prefix during the registration process, which raises serious doubts on the dependability of the LISP control-plane. As described in next chapter, by proposing an adapted version of the existing Route Origin Authorizations (ROAs) [5], our solution dynamically provides both, RLOC verification and EID authorization, and effectively avoid such attacks.

9.1.2 No Global EID Authorization

The current map registration process completely alienates the EID host's role, hence making it dependent on the LISP-Site's ETRs. The Map Register message is initiated by the ETR to perform the map registration at the MS. With this approach, there is no way that a MS can verify if an ETR is authorized by an EID host to perform map registrations on its behalf—this is because it is not even involved in the process. As described in previous chapter, the current security mechanism for the map registration process is a static stop-gap solution which requires manual preconfigurations of the EID prefixes, both on the ETR and on the MS, and a shared key between them. Moreover, the exclusion of the EID host as an entity from the map registration process is not only an impediment for providing global EID authorization but also for ensuring end-to-end security.

In [142], the author proposes a solution for achieving global EID authorization, which leverages the RPKI/ROA infrastructure. The solution defines a signed object, called *Identifier Origin Authorization (IOA)*, similar to ROA, which can act as an authorization from an EID prefix holder towards a particular set of RLOCs to populate the mapping database. However, this approach burdens the EID prefix holder device with intensive cryptographic chores including signing, verifying and handling certificates.

Another noticeable observation regarding current LISP specification is its impact on the mobility of the EID host. The current LISP specification recommends to explore Mobile IP technology in the case of mobility when an EID host moves relatively fast and requires to change its RLOC attachment point while maintaining session continuity. Figure 9.2 illustrates this scenario, where an EID host, EID_1, in $LISP-Site_1$ is registered on the mapping system by its ETR, namely xTR_1 (Step 1). At certain moment, EID_1 starts communicating with an EID host, EID_2 in $LISP-Site_2$ (Step 2). Later on, EID_1 moves to another LISP-Site $LISP-Site_3$ (Step 3). In order to keep an uninterrupted communication with EID_2, EID_1 will use the shared key it has with the xTR_1 to authenticate its location update through xTR_3 (Step 4). Once authenticated and updated, xTR_1 starts tunneling the traffic coming from EID_2 towards EID_1 at its new location (Step 5).

The extra burden for using Mobile IP technology for mobility 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 burden can be avoided by recognizing the EID host as a separate entity and involving it 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 solution, 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 global EID authorization.

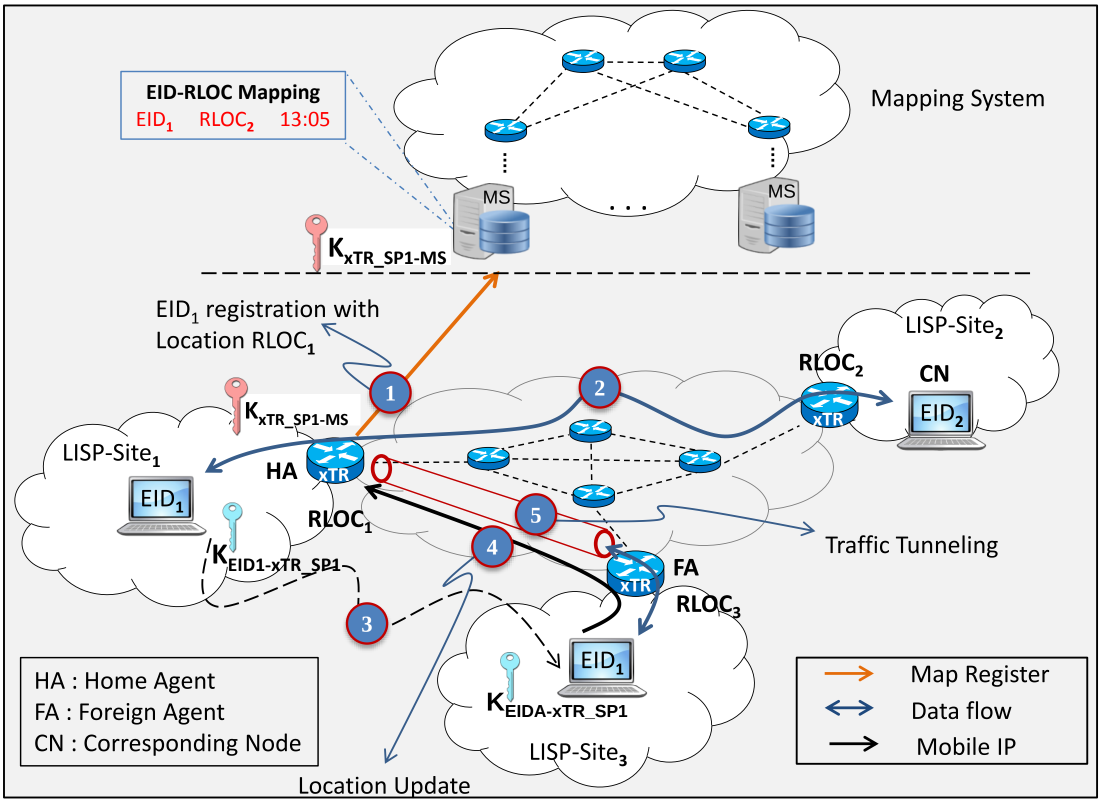


Figure 9.2: Mobility scenario with current LISP specification.

10 End-to-End Security for LISP Map Registration Process

This chapter describes our end-to-end secure map registration solution for LISP including resource authorization mechanisms. Unlike, the existing map registration process based on manual pre-configurations, our proposal enables dynamic map registration which paves the way for global EID authorization as well. Later in this chapter, we also provide a theoretical overhead analysis of our solution against the existing specifications.

10.1 Secure Map Registration Proposal

10.1.1 Prerequisites

In this section, we present some definitions and concepts that are essential to the development and explanation of 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 an RLOC or a set of RLOCs.

- **EID Authorization Process:** The mechanism by which a Map Server in the Mapping System is able to securely establish the fact that a particular ETR is authorized to register an EID prefix on its behalf.

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

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)

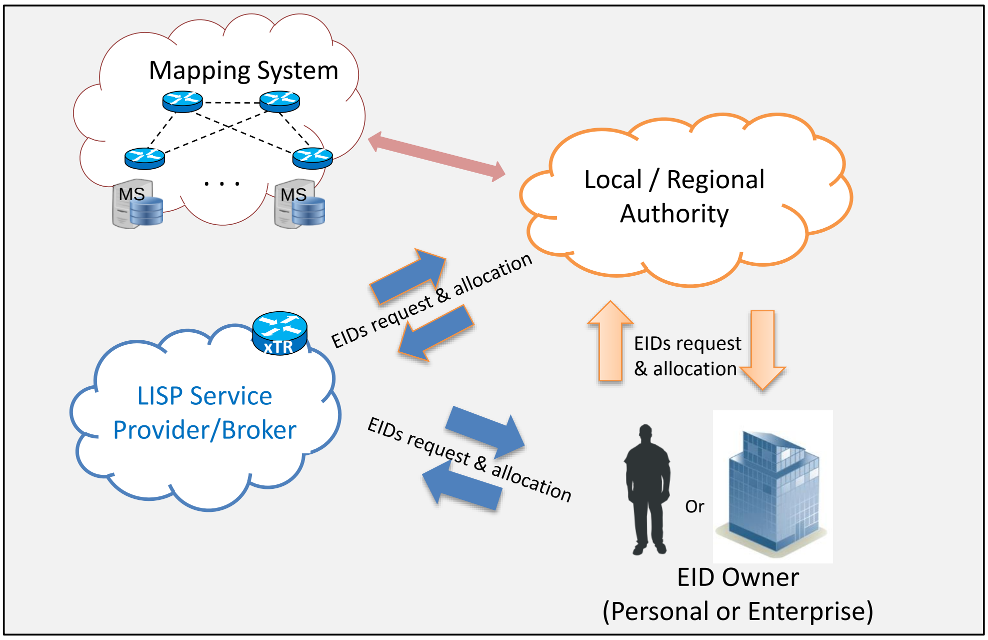


Figure 10.1: Possible scenario of EID ownership and acquisition.

and the Map Server (MSs) in the Mapping System, a new role is introduced, represented by the “user” or “host in the LISP-Site bearing the EID”, called the “EID-Holder”. In our proposal, the EID-Holder is considered independent of the service provider, which is in fact one of the main hooks of LISP. The term EID-Holder refers to the fact that the user or host is the owner of the EID prefix. The EID prefix can be acquired through a service provider, a broker, or directly from the respective regional authority (e.g., RIRs), but its ownership stays with the EID-Holder (cf. Fig. 10.1). 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.

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)

In order to enable RLOC verification, we propose to use a signed security object similar to the ROA [5] as developed by the SIDR WG [3]. We present an extension to the ROA [5] concept, which exploits the similarities between Route Origination in an inter-domain network

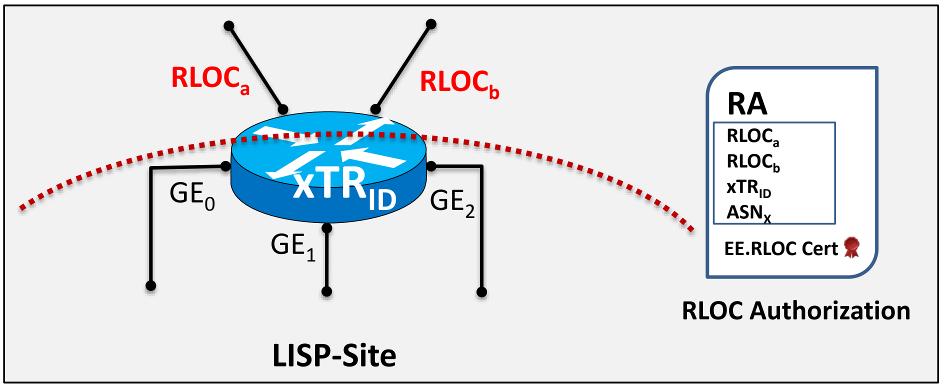


Figure 10.2: ROA inspired RLOC Authorization (RA).

with an RLOC used by an EID in a LISP-network. The ROA, as described in [5], 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. To this end, we propose an extension to legacy ROA that can be used for RLOC Authorizations. 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, as shown in Fig. 10.2. In order to ensure global and timely dissemination of RAs, we reuse the RPKI developed by the SIDR WG [3]. It is worth highlighting that, RPKI [4] has already been implemented and deployed by ARIN [71] and RIPE [70], and it is now under deployment phase in some regions. 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 [94]. 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 (cf. Fig. 10.3). 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.

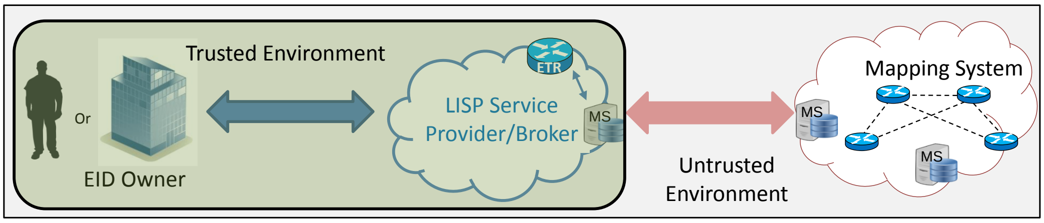


Figure 10.3: End-to-end map registrations: All trust scenario.

The second scenario assumes trust between the EID-Holder and the ETR only (cf. Fig. 10.4). 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.

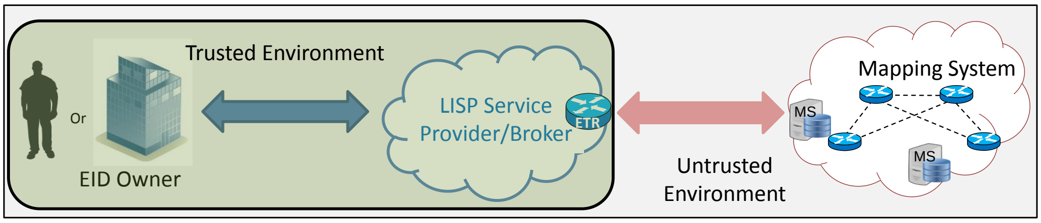


Figure 10.4: End-to-end map registrations: EID-xTR trust scenario.

The third scenario assumes no trust at all among the EID-Holder, the ETR, and the MS (cf. Fig. 10.5). This is typically the case of roaming scenarios, and requires strong security involving the three actors. We focus on this case to develop our solution since it is precisely the worst possible scenario.

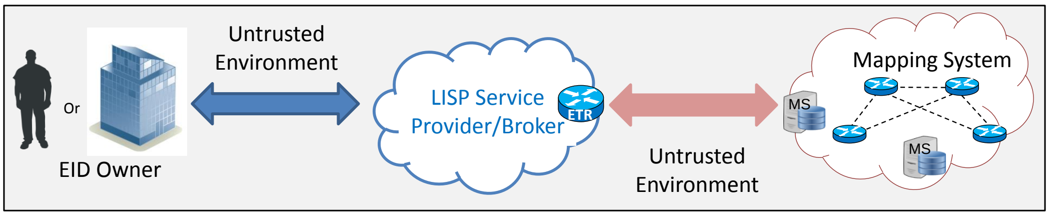


Figure 10.5: End-to-end map registrations: No trust scenario.

In the next section, we present of our dynamic and secure end-to-end map registration proposal for the no trust scenario.

10.1.2 End-to-End Secure LISP Map Registration

We divide the secure end-to-end map registration proposal 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. 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, so any party can be an attacker. In order to achieve end-to-end security and EID authorization, we propose to use a shared key between the EID-Holder and the Map Server. Later in this section, we also discuss the secure LISP Map Registration solution with the use of Public Key Infrastructure (PKI). 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 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 Fig. 10.6, where each step in solid color determines exchanged messages related with the Map Registration process, whereas the gradient steps relate to messages exchanged with the RPKI. In the first stage and prior to the Service Request, the EID-Holder must be aware of the Service Provider's ETR Identity (xTR_{ID}) from which it plans to use the service. The EID-Holder can learn about the xTR_{ID} through different means, e.g., in advance through certified templates advertised by the

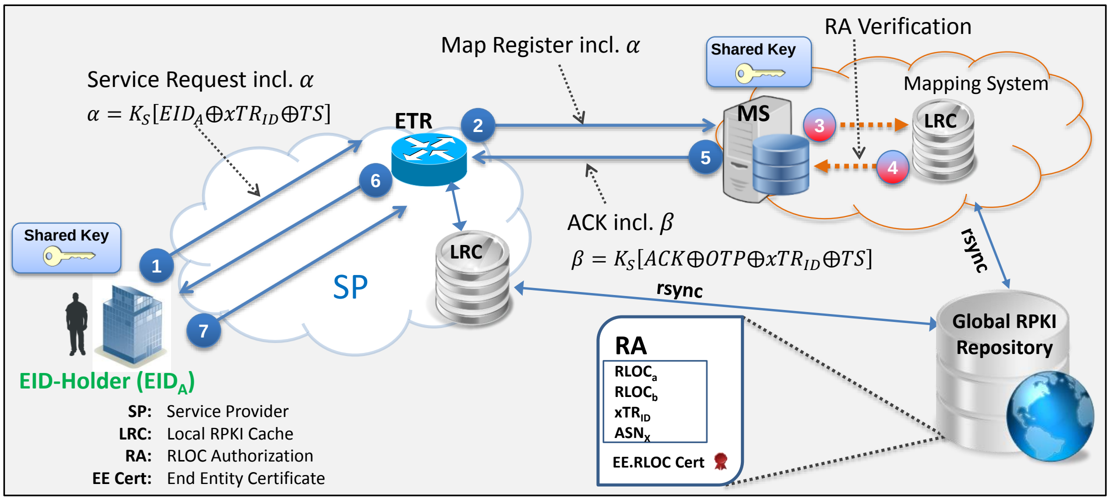


Figure 10.6: Step-by-step overview of the secure *Map Registration* process using Shared-Key between the EID-Holder and the MS.

providers, online through DHCP, by manual entry, etc. Then, the EID-Holder computes α (cf. (10.1)) by first concatenating its EID, xTR_{ID}, and a timestamp TS, and then encrypting this information with the shared key $\mathscr{K}_S$ it has with the MS.

$$\alpha = \mathscr{K}_S(EID_a \| xTR_{ID} \| TS) \tag{10.1}$$

The α is meant to be only visible to the corresponding MS in charge of the EID prefix, and will be used for the EID authorization process. The EID-Holder sends α in the *Service-Request* message to the ETR of the Service Provider, and it also adds in plain text the RLOC of the target Map Server, $RLOC_{MS}$, and its prefix EID_a (cf. step 1 in Fig. 10.6). 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 $\mathscr{K}_S$. 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 over claim EID prefixes due to the inability to produce a corresponding α.

Assuming that the Service Provider has already published the respective RAs on the RPKI for the RLOC that it plans to use during the registration, then the ETR can send a signed *Map-Register* message to the corresponding MS. The signature in the message includes α (received from the EID-Holder), its xTR_{ID}, its RLOCs, and the EID prefix it wants to register, EID_a (cf. step 2 in Fig. 10.6).

In the third stage (cf. steps 3 and 4 in Fig. 10.6), 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 (10.2), 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 $\mathscr{K}_S$ (cf. step 5 in Fig. 10.6).

$$\beta = \mathscr{K}_S(ACK \| OTP \| xTR_{ID} \| TS) \tag{10.2}$$

On receiving the ACK message, the ETR verifies the signature of the message, and if successful, it forwards only β to the EID-Holder who initiated the Service Request (cf. step 6 in Fig. 10.6). The EID-Holder verifies β using the shared key and validates its contents.

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. Only first four messages are required for the all trust scenario (see Fig. 10.3) and first five message are needed for the EID-xTR trust scenario (see Fig. 10.4) to secure the map registration process.

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.

Fig. 10.7 step-by-step illustrates the new secure map registration process when the EID holder has a verifiable digital certificate i.e., EID Certificate. Similar to the shared key secure map

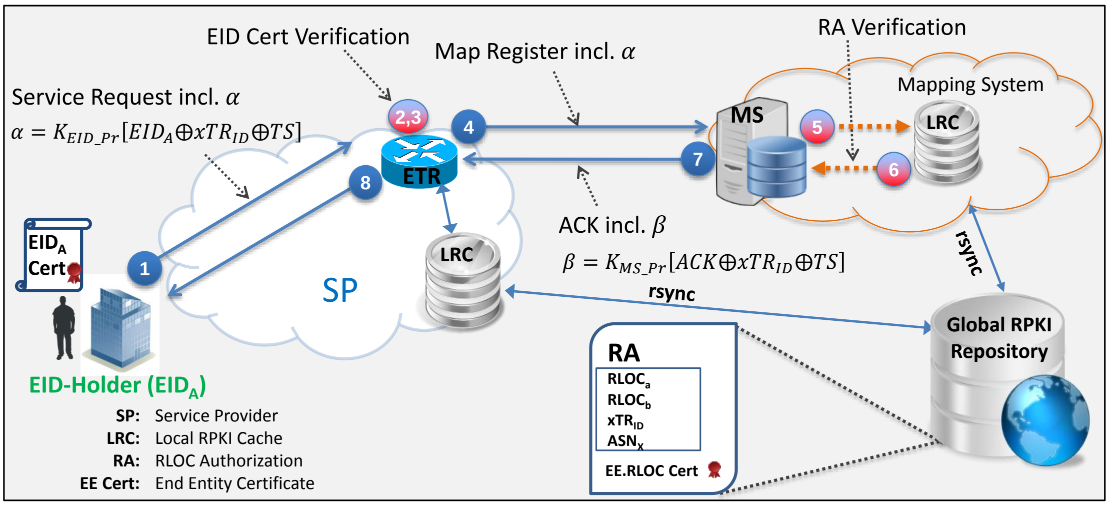


Figure 10.7: Step-by-step overview of the secure *Map Registration* process with PKI enabled EID holder.

registration process explained before, PKI enabled secure map registration process can be divided into three stages. In the first stage of this scenario, the EID holder can produce α by signing with its private key which can be used as the security credential (cf. 10.3). That is the EID holder and the respective MS need not to share a key between themselves.

$$\alpha' = \mathscr{K}_{EID-Pr}(EID_a \| xTR_{ID} \| TS) \tag{10.3}$$

Moreover, the ETR of the Service Provider can take advantage of the RPKI for verifying the *Service-Request* message (cf. step 2 and 3 in Fig. 10.7). The second stage involves sending the *Map-Register* message to MS including α (cf. step 4 in Fig. 10.7). In the third stage, the MS verifies the *Map-Register* message, α, xTR_{ID}, and the respective RA (cf. step 5 and 6 in Fig. 10.7). 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. However, in this case, the MS uses its private key for signing β (cf. 10.4). It is worth mentioning that the MS need not to include *OTP* in β as end-to-end signatures including MS, xTR and the EID holder, leave no room for Man-in-the-Middle or coordinated attack.

$$\beta' = \mathscr{K}_{MS-Pr}(ACK \| xTR_{ID} \| TS) \tag{10.4}$$

Although using end-to-end digital certificates improves the security by avoiding the shared key handling, it adds a handsome processing burden on all the players in the LISP architecture which could be highly undesirable, especially for the EID holder as a mobile node which has limited resources. Furthermore, using digital signatures and certificates at every stage does provide high level of security but it becomes impractical. Next, we provide the overhead analysis of the Shared-key version of the proposal, as it is more practical and provides reasonable security to achieve end-to-end secure LISP map registration.

10.2 Overhead Analysis

In this section, we evaluate the overhead that our solution imposes on the current LISP implementation. We examine the impact on the number of messages required to achieve secure map registrations in an end-to-end fashion. And finally, we analyze the overheads caused by different types of signatures and encryption algorithms for the Shared-Key version of the secure map registration process.

10.2.1 Overhead in the Number of Messages

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

Although current LISP specification deems sufficient to send only two messages for the Map Registration, this approach provides only poor security guarantees over the whole process. We recall that before exchanging the above mentioned map registration messages, manual pre-configuration of the shared-key and the RLOC-EID pairs is required on both ETR and MS. 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. Our solution requires a higher amount of messages, though offering significantly improved and adaptable end-to-end security. For the Shared-Key proposal, as shown in Figure 10.6, 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 10.1. The first five messages are sufficient in partially trusted scenarios, i.e., in trusted environments except between the ETR and the Map Server. Note that these include the final acknowledgement from the MS, which is optional in LISP. Indeed, in a completely trusted scenario, only the first four messages are needed to provide end-to-end security to the map registration process.

In the first stage of our proposal, the registration is initiated by the EID-Holder, which sends a *Service-Request* message towards the Service Provider (cf. Step 1 in Fig. 10.6). This is a new LISP control-plane message consisting of the following information: [EID prefix ∥ $RLOC_{MS}$ ∥ α]. Figure 10.8 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 in the Shared Key version of the proposal, we reuse the *Service-Request* message format shown in figure 10.8.

10.2.2 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

```
         0                   1                   2                   3
         0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |Type=5 |                                                       |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                         Nonce . . .                           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |                         . . . Nonce                           |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        |            Key ID             |     Encrypted Data Length     |
        +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        ~                          Encrypted Data                       ~
    +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    R   |         EID mask-len          |            Reserved           |
    e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    c   |        EID-prefix-AFI         |           MS-RLOC-AFI         |
    o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    r   |                           EID-prefix                          |
    d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |   |                          MS Locator                           |
    +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10.8: Service-Request message format.

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 10.1, which shows the overhead incurred for each message, considering the encryption type, the signature, and their sum for computing the overall overhead.

In this evaluation, we considered an IPv6 EID-prefix (128 bits), 128 bits for xTR_{ID} and a timestamp of 64 bits. This adds up to a total of 320 bits (40 bytes) for α. Once encrypted, α grows to a size between 48 and 64 bytes, plus the Initialization Vector (IV) amounting to a total between 64 and 96 bytes depending on the selected AES key depth. On top of that, the *Service-Request* 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 *Service-Request* message is 56 bytes + α (cf. Fig. 10.8 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. Table 10.1 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

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

[a] AES-CBC encryption Mode.

[b] Counting the size of the Initialization Vector (IV).

Table 10.1: End-to-end secure Map Registration Process Security Overhead

that need to be applied on the *Map-Notify* message, it is sufficient to include an encrypted acknowledgement β destined for the EID-Holder, and the signature data of the message itself (cf. message 5 in Fig. 10.6). 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 10.1.

Followed by the *Map-Notify* message, the ETR forwards β towards the EID-Holder in message number 6 (cf. Fig. 10.6). The estimated size of this message, keeping in mind the format shown in Fig. 10.8, 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 10.1.

In summary, the total security overhead of our proposed end-to-end secure map registration scheme fluctuates approximately between 952 and 1160 bytes (for the No trust scenario), as compared to the 176 to 200 bytes for the current registration process in LISP. Moreover, a basic implementation of our solution including encryption/decryption as well as signatures and verification was developed and tested. The initial results reveal the average time to complete the proposed end-to-end secure map registration process to be 0.259 seconds, compared with the average time of 0.11 seconds for the current registration process—this result excludes the time required for RA verification (i.e., Step 3 and 4 in Fig. 10.6). Clearly, enhancing the security has an associated cost, but the benefits obtained allow for a broader technological reach, especially in areas requiring mobility, where the users roam to foreign networks while keeping their original identifiers and sessions alive. With our solution, this can be achieved, along with on-the-move EID authorization and RLOC verification, without the complexities and extra burden of tunneling across protocols and mobile technologies.

Conclusions and Future Work Part IV

11 Conclusion

In this thesis, we examined the security of the BGP protocol and the LISP protocol, with special focus on the development of pragmatic solutions to improve the overall security of the inter-domain routing. Next, we present the important conclusions that can be drawn from the work done during the course of this thesis.

We discussed the solutions being developed and standardized by the IETF SIDR WG for securing the inter-domain routing system. We observed that the solutions offered by SIDR WG counter some of the existing vulnerabilities of BGP, but with the collateral burden of needing considerable software and hardware upgrades in the inter-domain routing system. More specifically, the BGPSEC protocol requires syntactical and operational changes to the BGP protocol, causing enough entropy to invite resistance from the industry. Furthermore, apart from the unresolved route leak problem, new attacks are also possible by exploiting the operational shortcomings of SIDR proposals, such as BGPSEC functionality downgrade. It is worth mentioning that the security solutions of the SIDR WG still represent work in progress, since the RPKI and ROA are currently in trial deployment phase, while the BGPSEC protocol is still under discussion.

In regard to the route leak, we presented a clear and formal definition of the problem. Our definition separated the route leak problem from the IP prefix hijack problem distinctly, unlike some of the previous works which consider both as the same problem. Furthermore, we identified two types of route leaks based on the nature of their occurrence, namely, Customer Route Leak (CRL) and Peer Route Leak (PRL).

Then, we developed a theoretical route leak detection framework based on realistic hypotheses and theorems, under which an AS is able to detect route leak initiation autonomously. The main advantages of our pragmatic approach include: a) no changes required to the BGP protocol; b) no reliance on third party information (e.g., vantage points) or third party infrastructure; and c) from an engineering perspective, route filters may be needed for an initial training period to ascertain the defined hypotheses, but their continuous maintenance is not required. Based on the route leak detection framework, we introduced a total of three real-time route leak detection techniques, namely Cross-Path (CP), Benign Fool Back (BFB), and Reverse Benign Fool Back (R-BFB). The CP technique systematically utilizes the BGP information available at hand to identify route leak following the Algorithm 1. The BFB and R-BFB are based on the ingenious and harmless fool back advertisement to detect route leaks utilizing the BGP information available at the control-plane and data-plane, respectively. The CP technique is more robust than the BFB and the R-BFB techniques as it remains effective against both unintentional (e.g., misconfigurations) and intentional (e.g., deliberate attack by a malicious AS) route leak occurrences. The BFB and the R-BFB are able to counter route leaks due to misconfigurations, i.e., when they are not the result of premeditated and elaborated attacks—BFB and R-BFB will not fool a prepared attacker.

We conducted large scale event driven simulations using scaled down actual Internet topologies, consisting of more than a thousand of ASes, as well as real-time experiments to evaluate the proposed RLD techniques. The result analysis of our simulations reveals that, using only the CP technique, high accuracy in the detection seems feasible for customer route leaks, but for the peer case, additional mechanisms seem mandatory. Moreover, our evaluation results show that BFB technique can substantially improve the detection success rate for the peer case. Furthermore, we demonstrated through real-time experiments, using data-plane based R-BFB technique, that high route leak detection success rate can be achieved. The results from our tests, which include more than 20,000 event driven simulations and 90 real-time experiments, show that an AS is able autonomously detect route leaks in different scenarios with a high success rate using the CP, BFB and R-BFB, especially, when the three techniques are combined and used together. Overall, the solutions proposed in this thesis for resolving the route leak problem in the inter-domain arena are quite pragmatic in the sense that they avoid all the main pitfalls which posed resistance to the wide scale adoption of previous security solutions. Furthermore, it is worth highlighting that our route leak solution remains self-beneficial for even a single domain regardless of its adoption by other neighboring domains.

In regard to the LISP protocol, we have presented a novel and adaptable approach for secure map registrations in LISP. Our solution 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 identify a separate role of the EID-Holder to achieve the end-to-end security blanket ensuring global EID authorization. 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, while presenting a potential adoption blueprint.

In the next chapter, we discuss the future research work based on the findings of this thesis that can contribute to the enhancement of Internet security.

12 Future Work

In this chapter, we look in to possible research directions that can be pursued on the basis of the contributions made in this thesis.

12.1 Using Smart Analytics on Locally Accessible Information

Learning from the lessons of past security proposals for BGP, in this thesis we showed that how systematic and analytical usage of locally available information, on control-plane or data-plane, can be used to enhance BGP security. The route leak detection framework presented in this thesis only utilizes information locally accessible on the control-plane or data-plane to detect route leaks. This approach avoids the pre-condition of global deployment for its success, that is, an individual AS can utilize its own resources to autonomously cater its security needs according to its requirements. The concept of using smart analytics on local information for countering vulnerabilities offers a potential research direction to secure or manage high inertia protocols or technologies in the Internet.

In this thesis, we considered a single router per AS in our simulations. Indeed, multiple border routers per AS, i.e, with more than one RIB belonging to same AS but with different Internet route views, can actually improve the route leak detection rates achieved. However, it remains to be investigated that *how* and *where* would be the best way to process and manage the information from all the border router RIBs within a domain. For example, would a centralized approach to the consumption of multiple RIBs within a domain is more efficient or a distributed one. That is, should the information be first processed and then shared with other border routers or the information should be processed at a common point within the domain, such as at a router reflector.

In terms of route leak detection, we concede that the route leak detection framework presented in this thesis is valid for detecting—under certain conditions—route leak initiations only. The detection of propagated route leaks is more difficult than detecting the route leak on the link where it initiates. This is because the *victim* AS may forward the route leak to its neighbors according to valley-free model, which makes it more difficult for any AS receiving the propagated route to distinguish it as a route leak. The detection of a propagated route leak using only BGP control-plane and data-plane intelligence is an interesting research direction.

The valley-free routing model serve as a reasonable stepping stone toward modeling the route leak problem, however, there exist exceptions to valley-free constraints [49, 76] as they are not upheld sometimes to accommodate customized economic models due to a complex AS relationship between the ASes. Hybrid AS-relationship is one such example where two large ASes have different relationships between them at geographically different points of presence (PoP). We contend that the RLD techniques presented in this thesis will stay valid in hybrid scenarios as well, since the routing information that is relevant for the route leak detection is the one contained in the routers in proximity with the occurrence of the route leak. However, the hybrid scenario becomes more interesting when BGP intelligence is pooled in from all border routers of the AS, including the ones with hybrid relationships. An intriguing research course would be to investigate the detection of intentional or unintentional route leaks in such hybrid scenarios.

12.2 Integrating Security in to BGP

This thesis focuses more on the "detection" techniques than on "remediation" techniques. However, Appendix C details an initial prototype for outsourcing BGP security chores to an overlay network with the help of OPENER. The recent developments in the networking arena, such as Software Defined Networks (SDN) can also be utilized for this purpose. SDN enables to outsource control functions of an SDN-enabled network element to external applications, by exposing available capabilities through open APIs. The SDN approach offers a promising alternative for inducting security into BGP by outsourcing the security mechanisms to distributed third party SDN solutions. We contend that a remediation solution exploiting the RLD techniques, proposed in this thesis, can be deployed as an SDN application. As mentioned earlier, the BGP/LS and PCEP project under the umbrella of OpenDaylight [126] is putting in efforts to cater the deficiency of native support for BGP in the OpenDaylight controller. However, further efforts are required to formalize a blueprint of a distributed infrastructure involving SDN controllers (per AS domain) and to identify and provide necessary north-bound and south-bound APIs to pave the way for the deployment of security applications on top of the BGP protocol, such as the RLD techniques.

The OPENER tool provides a very basic platform for experimenting with third party applications on top of open source Quagga router. In this regard, our near future research plan includes integrating and experimenting with the three RLD techniques developed in this thesis on top of Quagga using OPENER.

12.3 Enhancing LISP Security

Securing the LISP protocol is in its early stages of research. We plan to make available our solution to the open source community in LISP, and we will examine its performance in large topologies. Furthermore, we foresee following research directions with regard to LISP security based on the contributions made in this thesis.

- It is important to measure the consequential impact of adding a new player, i.e., the EID holder, on different aspect of LISP functionality.
- Another interesting direction to investigate is the impact on the Mapping System in case of user migration from one service provider to another, especially when the user has provider independent EID.

A Publications

Journals

1. M. Yannuzzi, M. S. Siddiqui, A. Sallstrom, B. Pickering, R. Serral-Gracia, A. Martınez, W. Chen, S. Taylor, F. Benbadis, J. Leguay, E. Borrelli, I. Ormaetxea, K. Campowsky, G. Giammatteo, G. Aristomenopoulos, S. Papavassiliou, T. Kuczynski, S. Zielinski, J. M. Seigneur, C. Ballester Lafuente, J. Johansson, X. Masip-Bruin, M. Caria, J. R. Ribeiro Junior, E. Salageanu, and J. Latanicki., "TEFIS: A Single Access Point for Conducting Multifaceted Experiments on Heterogeneous Test Facilities", Elsevier Computer Networks, Volume 63, Pages 147–172, 22 April 2014.

2. W. Ramirez, X. Masip, M. Yannuzzi, R. Serral-Gracia, A. Martinez, M. S. Siddiqui., "A survey and taxonomy of ID/Locator Split Architectures", in Elsevier Computer Networks Journal, Volume 60, Pages 13-33, 26 February 2014.

3. M.S. Siddiqui, D. Montero, R. Serral-Gracia, X. Masip-Bruin, and M. Yannuzzi, "A Survey on the Three Pillars that the Internet Standardization Body is Developing for Securing Inter-Domain Routing", revision submitted to Elsevier Computer Networks in May 2014.

4. M.S. Siddiqui, D. Montero, R. Serral-Gracia, and M. Yannuzzi, "Self-Reliant Detection of Route Leaks in Inter-Domain Routing", submitted to Special Issue on Fault-Tolerant Communication Networks in Elsevier Computer Networks, June 2014.

Conferences

1. M.S. Siddiqui, D. Montero, R. Serral-Gracia, M. Yannuzzi, and X. Masip-Bruin, "Route Leak Detection Using Real-Time Analytics on local BGP Information", to be presented at Globecom 2014 - Next Generation Networking Symposium (GC14 NGN), Austin, USA, December, 2014.

2. M.S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracia, and X. Masip-Bruin, “Diagnosis of Route Leaks Among Autonomous Systems In The Internet”, presented at 5th International Conference on Smart Communications in Network Technologies 2014 (SaCoNeT 2014), Vilanova i la Geltrù, Spain, June, 2014.

3. M.S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracia, and X. Masip-Bruin, “Route Leak Identification: A Step Toward Making Inter-Domain Routing More Reliable”, presented at 2014 / 10th International Conference on Design of Reliable Communication Networks (DRCN 2014), Ghent, Belgium, April, 2014.

4. D. Montero, M.S. Siddiqui, R. Serral-Gracia, M. Yannuzzi, and X. Masip-Bruin, “Securing the LISP Map Registration Process”, presented at Globecom 2013 - Next Generation Networking Symposium (GC13 NGN), Atlanta, USA, December, 2013.

5. W. Ramirez, X. Masip-Bruin, E. Marin-Tordera, M. Yannuzzi, A. Martinez, S. Sanchez-Lopez, M.S. Siddiqui, and V. Lopez, “A Techno-Economic Study of Network Coding Protection Schemes”, to be presented at Globecom 2014 - Optical Networks and Systems Symposium (GC14 ONS), Austin, USA, December, 2014.

6. W. Ramirez, X. Masip-Bruin, E. Marin-Tordera, M. Yannuzzi, M.S. Siddiqui, A. Martinez, and V. Lopez, “Improving Reliability in Multi-Layer Networks With Network Coding Protection”, presented at 18th International Conference on Optical Network Design and Modeling (ONDM 2014), Stockholm, Sweden, May 2014.

B Projects

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

Spanish Projects

- **DOMINO**: Diseño Multinivel de Nuevas Arquitecturas y Protocolos para Redes Multidominio (DOMINO), TEC2009-07041, (2010-2012)
- **GESTIONA**: Design of Management Strategies for New Network Architectures (Future Internet), TEC2012-34682, (2013-2016)

European Projects

- **FP7 Project TEFIS**: TEstbed for Future Internet Services, TEFIS Project FP7-258142, (2010-2013)
- **FP7 Project OpenLab**: extending FIRE testbeds and tools, OpenLAB Project FP7-287581, (2013-2014)

Cisco RFP Project

- Overlays for Control Plane Security (2012-2013)

Net-IT Lab Internal Project

- **OPENER**: Open and Programmable ENvironment for Experimenting with Routers

C OPENER

In this thesis we describe its basic architecture and modes of operations. We briefly discuss the initial scalability and performance tests of OPENER as a tool that were carried out in the framework of the TEFIS project[1] [143]. The target of the experiments was to use TEFIS as a debugging and performance optimization platform for the initial implementation and refining of OPENER's core modules. Later (in Section C.2), we demonstrate mitigation of false prefix origination attack using a third-party application on OPENER enabled routers.

C.1 Open and Programmable ENvironment for Experimenting with Routers (OPENER)

The Open and Programmable ENvironment for Experimenting with Routers (OPENER) is a programmable environment for open access to routers' capabilities. Hence, it allows third party applications to assist, query, complement, improve, and even change the processes and the commands running on the routers. OPENER provides a solid and extensible management interface that complements and enhances the functionalities present in routers. As a proof-of-concept, we have developed a prototype implementation in the legacy Quagga routing suite [127], enabling the development and testing of experimental applications that can seamlessly run on top of a Quagga router. The approach used by OPENER is somehow similar to JUNIPER's JUNOS SDK [144], where it is possible to independently develop an external application (third party application), deploy it, enabling the experimentation with novel protocols, algorithms, and applications directly interacting with the router internals through the provided open interface. The main advantage of OPENER is that it does not depend on any vendor specific platform, and therefore, it offers much more flexibility than solutions such as JUNIPER's SDK. To accomplish this open and programmable environment, OPENER uses REST [145] as its interface with third party applications, while exposing the selected capabilities of the

[1] TEstbed for Future Internet Services (TEFIS) is an open platform that offers a versatile combination of heterogeneous experimental facilities. It provides a single access point for conducting cutting-edge experiments on testbeds that supply different capabilities, including testbeds dedicated to network performance, software performance, grid computing, and living labs.

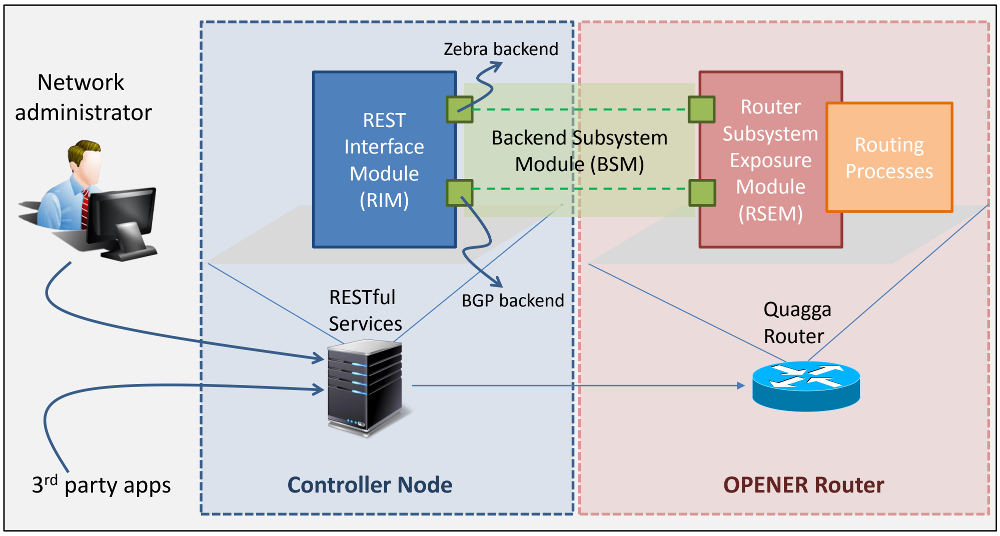


Figure C.1: Modules composing OPENER.

underlying routing system by direct access to the internal router's processes. In fact, OPENER interacts with the router's processes allowing direct operations to change the internal workings according to a requested action, thus providing programmability and a much richer interface than the regular Command Line Interface (CLI) [146]. The whole process highlighting the different parts composing OPENER is shown in Fig. C.1.

OPENER is composed of the following modules:

- REST Interface Module (RIM): It defines the exposed functionalities to third party applications through a REST interface.
- Backend Subsystem Module (BSM): To deliver an extensible and reliable system, OPENER offers a fully pluggable implementation that allows the seamless extension of the interface to other technologies, which are presented to the third party applications through the RIM.
- Router Subsystem Exposure Module (RSEM): This module exposes the necessary internal capabilities of the router to the BSM, and is used to enforce the different operations as requested by third party applications.

C.1.1 Operation Modes

As a consequence of the complexity and diversity of network sizes and topologies, OPENER is devised to work with two basic modes of operation, namely, centralized and distributed modes. Both modes of operation have their respective pros and cons. The proposed experiments in the TEFIS project targeted the pragmatic assessment of the scalability features of both modes of operation, with the goal of assisting in the decision of selecting a particular mode for a given set of demands. The next two subsections detail the centralized and distributed modes of operation of OPENER.

Centralized Mode

The centralized mode of operation of OPENER refers to the scenario where a single controller node orchestrates the OPENER enabled actions of a set of routers in the administrative domain.

Fig. C.2 shows the overall logical view of the centralized mode of operation of OPENER. As it can be observed, in the centralized mode, there is a single station controlling a set or subset of routers within the network. The OPENER user interface at the controller station, allows third party applications to access different elements of the network as per requirement. In particular, the controller node enforces the requested configuration to the end-routers, as shown by the green arrows. In addition, the controller node can also monitor the network as requested by the administrator. This monitoring can be carried out through the user interface, or by any third party application within the network. In summary, through this operation mode, OPENER enables third party application access to Quagga routers from a centralized controller node.

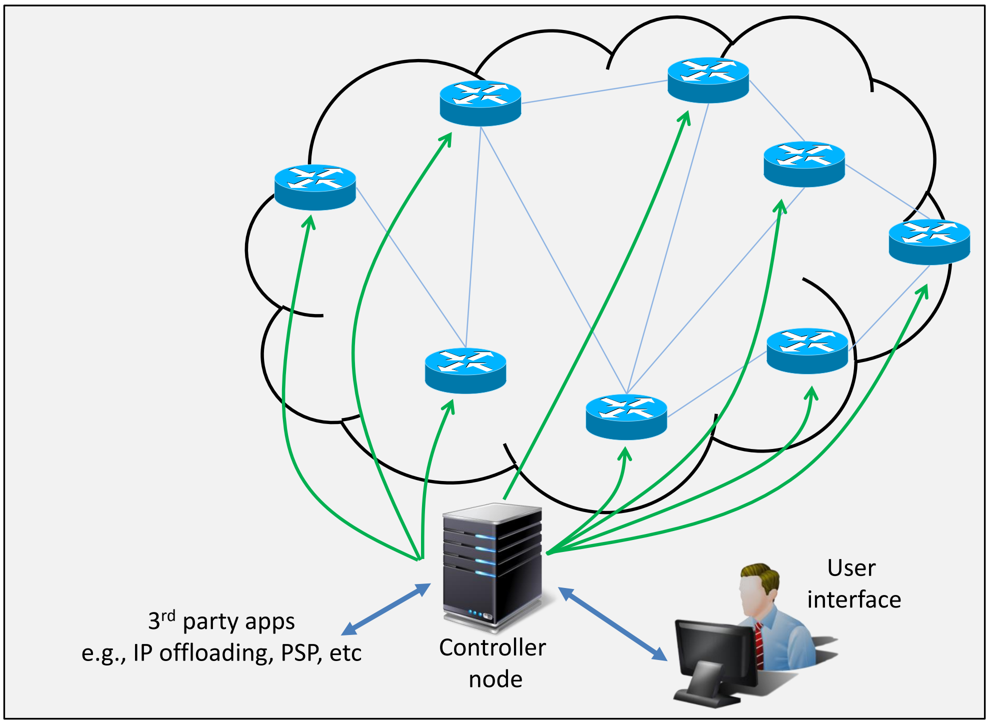


Figure C.2: OPENER centralized mode.

Distributed Mode

Opposed to the centralized operation mode, the distributed mode disperses the controller nodes among the different routers within the network. That is, each network element has its own controller, which handles the third party application tasks through the OPENER user interface. Therefore, in this scenario the controller is mainly aimed at local node configuration, without direct support for global control at least directly from the OPENER's perspective. However, this does not exclude that third party applications interact among each other in any sort of coordinated fashion, so as to effectively control the network from a global perspective. By pushing the complexity towards each element, the distributed operation mode allows higher scalability of the OPENER system, as it inherently distributes the load and resource usage incurred by the controller node among the network elements. Clearly, this comes at the cost of increased complexity for the third party applications. Fig. C.3 shows the logical overview of the distributed operation mode of OPENER. As shown in the figure, all network elements have their own third party application controllers, through which their respective tasks can be performed based on specific requirements.

At the expense of having one controller for each router, the distributed operation of OPENER can avoid the extra control traffic on the network. As mentioned earlier, this leads to a communication void between the controllers of all the routers. However, third party applications can be used to fill this void by interacting with the controller node of each router separately, hence giving a global control perspective. But in that case, the load and complexity are moved toward the third party application.

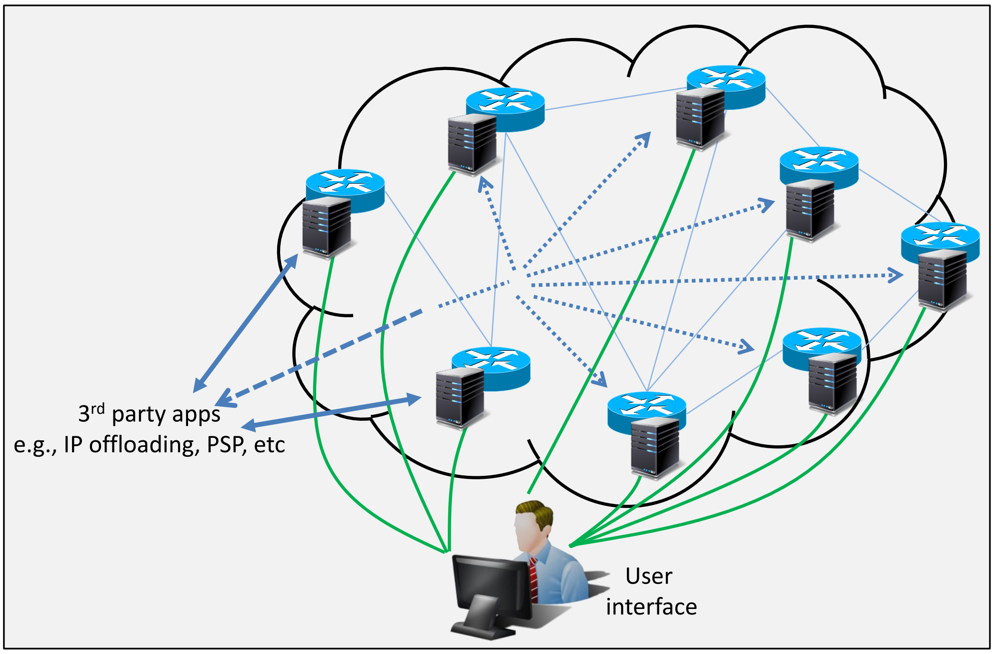


Figure C.3: OPENER distributed mode.

C.1.2 OPENER Testing using the TEFIS Platform

The testing of OPENER was carried out in two phases using the TEFIS platform [143]. The goal in the first phase of the experiment was to assess the performance and scalability of different deployments of the OPENER framework. One of the main goals of this experiment, was the optimization of OPENER to achieve deployments of realistic network sizes, including large scale deployments. In reference to this phase, TEFIS provided a comprehensive and user friendly platform allowing efficient repetition, management, and configuration of the experiment to aid exhaustive debugging. Subsequent optimizations on the experimenters' tool were evaluated through TEFIS by running a set of benchmarks iteratively, e.g., computation of query response times depending on the number of concurrent managed nodes, while assessing both the performance and the resource usage for each test. In each iteration, the code was optimized for better concurrency level using the knowledge provided by the experimental results, which at the end provided a more resilient codebase for OPENER.

The second phase of the experiment aimed at a real use case of the OPENER framework by targeting coordinated cross-layer interactions in multi-layer scenarios. To this end, a third party application was deployed and tested—an IP traffic offloading solution in this case. This application allows smart orchestration of IP and transport resources, so as to optimize their usage by offloading part of the traffic between two IP routers through a different optical path. Figure C.4 shows an example deployment, both in Kyatera and in PlanetLAB. As it can be observed from the figure, the goal of the experiment is to offload partial IP traffic when the link utilization reaches a particular threshold. This was orchestrated through OPENER

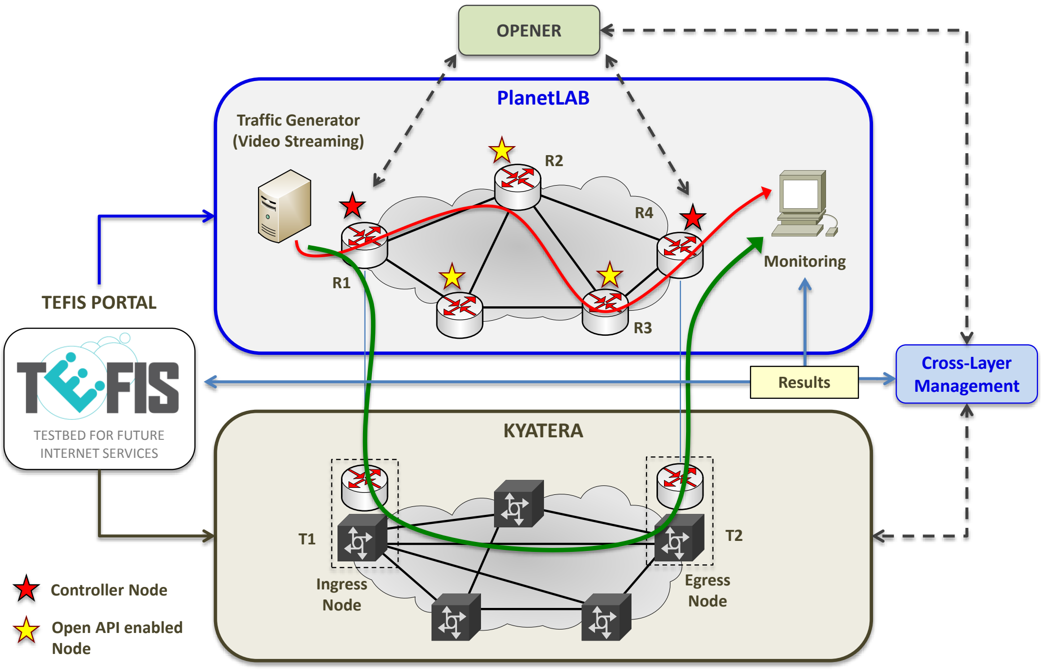


Figure C.4: Overview of the second phase of the OPENER experiment using TEFIS platform.

by a third party application which monitored the utilized bandwidth at all network links. The experiments carried out in TEFIS confirmed the adaptability of OPENER to provide coordinated control and enhanced functionality to the IP layer.

C.2 Distributed Security Overlay

In the context of outsourcing or decoupling security from the protocol, Yannuzzi et el. developed in collaboration with Cisco Systems the Path-State Protocol (PSP) [147]. This protocol provides a distributed overlay for transparently securing the BGP protocol. By "transparently" we mean that BGP is not even aware that it is being secured.

Figure C.5 depicts the basic architecture used for experimenting with the PSP protocol. The figure shows two neighboring domains and the interactions between the different components. Note that the vertical interactions between the BGP routers and the PSP controllers for IP prefix and AS-path inspection are performed through OPENER. Indeed, the OPENER platform not only enables non-disruptive control over the BGP process running on the router, but it can

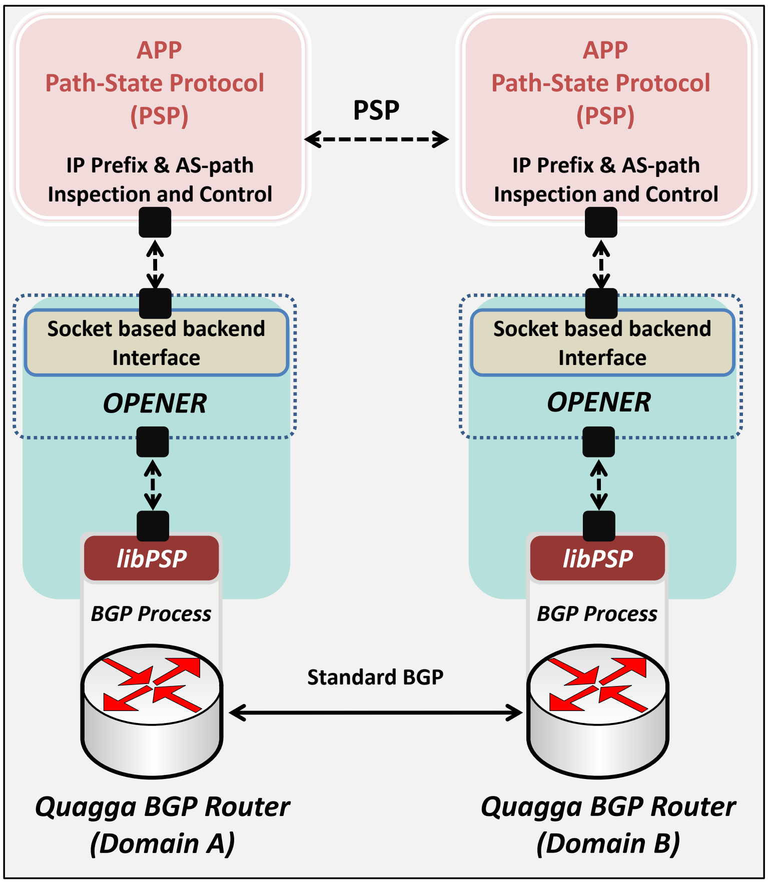


Figure C.5: The Path-State Protocol (PSP) on OPENER: An overlay solution for transparently securing BGP.

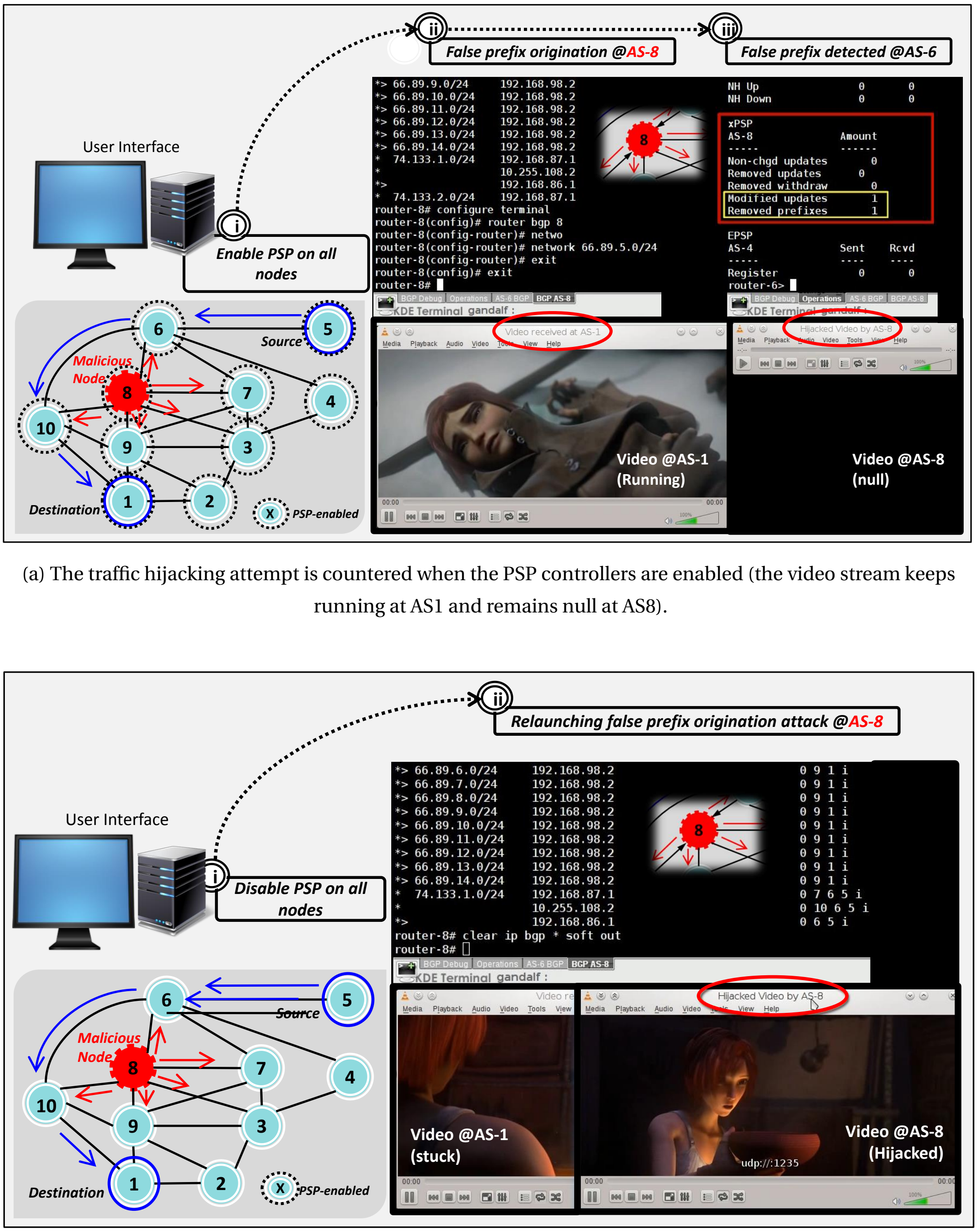
also provide the execution environment for the PSP controllers. The horizontal interactions, on the other hand, are split into two isolated control-plane flows, standard BGP messaging in the underlay, and PSP messaging in the overlay. As it can be observed in Fig. C.5, the lightweight library called "libPSP" forms the RSEM (cf. Fig. C.1) on the quagga router. The latter was specifically designed to provide the necessary capabilities and callbacks so that a PSP controller can transparently take control of the processing of BGP routing updates. For complete details of the PSP protocol and further information about this technology, we refer the reader to the PSP trials available at [147].

The use case shown in Fig. C.5 was demonstrated in a scaled trial that was exhibited at LACNOG[2] [148]. For the demonstration, we used a geographically distributed network topology between Europe and Latin America, where PSP controllers on OPENER were used for controlling ten open source routers (Quagga routers), representing the ten Autonomous Systems (AS) shown at the bottom left of Fig. C.6(a). The goal of the demo was to show the potential of an overlay solution for mitigating traffic hijacking attempts through false BGP advertisements.

In order to expose the traffic hijacking in a visible manner, we used a video stream between two ASes. The video was streamed from a source node in AS5 toward a destination host in AS1 with IP address in the range 66.89.5.0/24. The path followed by the video stream in the trials was determined by the standard BGP protocol, and as shown at the bottom left of Fig. C.6(a), the path chosen was the following: AS5 → AS6 → AS10 → AS1. In this setting, we carried out the attack twice, first with the PSP controllers enabled and then with the controllers disabled. Step (i) in Fig. C.6(a) shows that the PSP controllers are enabled and operative—note that they are in fact enabled in all the routers in the network—while Step (ii) shows how the administrator of router-8, in AS8, falsely originates the prefix 66.89.5.0/24 owned by AS1. As shown in Fig. C.6a), this prefix is advertised by router-8 to all the neighbors of AS8. Despite this, the video continues to be received at AS1, and, as it can be observed at the bottom right of Fig. C.6(a), the video stream expected at AS8 remains null. This is because the PSP controllers embed the Route Origin Authorization (ROA) functionality [5], allowing them to thwart the false prefix origin attack. This is visible in Step (iii) at the top right of Fig. C.6(a), which shows the status of the PSP counters of router-6 in AS6. In short, the PSP controllers running on AS8's neighbors inspect the routing update sent by router-8 and discard it. Figure C.6(b) illustrates the same scenario when the PSP controllers are disabled. As it can be observed, now the attack succeeds, since upon relaunching the prefix advertisement at router-8, the video stream is hijacked by AS8.

Although OPENER is an ongoing project, the initial prototype has proved to offer a valuable and useful platform for testing outsourcing of security chores away from the BGP protocol by allowing to deploy and to test third party applications on Quagga router. Furthermore, OPENER poses minimum possible entropy to the BGP protocol in enabling such an open experimental environment.

[2] The mitigation of false prefix origination using PSP trial demonstrated at LACNOG was lead by Dr. Marcelo Yannuzzi.



```
*> 66.89.9.0/24     192.168.98.2
*> 66.89.10.0/24    192.168.98.2
*> 66.89.11.0/24    192.168.98.2
*> 66.89.12.0/24    192.168.98.2
*> 66.89.13.0/24    192.168.98.2
*> 66.89.14.0/24    192.168.98.2
*  74.133.1.0/24    192.168.87.1
*                   10.255.108.2
*>                  192.168.86.1
*  74.133.2.0/24    192.168.87.1
router-8# configure terminal
router-8(config)# router bgp 8
router-8(config-router)# netwo
router-8(config-router)# network 66.89.5.0/24
router-8(config-router)# exit
router-8(config)# exit
router-8#
```

```
NH Up                     0       0
NH Down                   0       0

xPSP
AS-8                Amount
-----               ------
Non-chgd updates         0
Removed updates       0
Removed withdraw         0
Modified updates         1
Removed prefixes         1

EPSP
AS-4                Sent    Rcvd
-----               ----    ----
Register              0       0
router-6>
```

(a) The traffic hijacking attempt is countered when the PSP controllers are enabled (the video stream keeps running at AS1 and remains null at AS8).

```
*> 66.89.6.0/24     192.168.98.2                  0 9 1 i
*> 66.89.7.0/24     192.168.98.2                  0 9 1 i
*> 66.89.8.0/24     192.168.98.2                  0 9 1 i
*> 66.89.9.0/24     192.168.98.2                  0 9 1 i
*> 66.89.10.0/24    192.168.98.2                  0 9 1 i
*> 66.89.11.0/24    192.168.98.2                  0 9 1 i
*> 66.89.12.0/24    192.168.98.2                  0 9 1 i
*> 66.89.13.0/24    192.168.98.2                  0 9 1 i
*> 66.89.14.0/24    192.168.98.2                  0 9 1 i
*  74.133.1.0/24    192.168.87.1                  0 7 6 5 i
*                   10.255.108.2                  0 10 6 5 i
*>                  192.168.86.1                  0 6 5 i
router-8# clear ip bgp * soft out
router-8#
```

(b) The attack succeeds when the PSP controllers are disabled (the video stream gets stuck at AS1 and starts running at AS8).

Figure C.6: Demonstration of an PSP control overlay on OPENER for transparently securing the BGP routing protocol.

Bibliography

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

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

[3] Secure InterDomain Routing (SIDR) Working Group IETF, “http://datatracker.ietf .org/wg/sidr/,” 2013.

[4] M. Lepinski and S. Kent, “An Infrastructure to Support Secure Internet Routing,” RFC 6480, IETF, 2012.

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

[6] M. Lepinski, “BGPSEC Protocol Specification.” draft-ietf-sidr-bgpsec-protocol, February 2013.

[7] T. Paseka, “Why Google Went Offline Today and a Bit about How the Internet Works.” http://blog.cloudflare.com/why-google-went-offline-today-and-a-bit-about, Nov. 2012.

[8] G. Huston, “Leaking routes.” http://www.potaroo.net/ispcol/2012-03/leaks.html, March 2012.

[9] Locator/ID Separation Protocol (LISP) Working Group IETF, “http://tools.ietf.org /wg/lisp/,” 2013.

[10] K. Sriram, O. Borchert, O. Kim, D. Cooper, and D. Montgomery, “RIB Size Estimation for BGPSEC.” http://www.nist.gov/itl/antd/upload/BGP SEC_RIB_Estimation.pdf, May 2011.

[11] RIPE NCC, “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, November 2010.

[12] C. Labovitz, “China hijacks 15% of internet traffic.” http://www.arbornetworks.com /asert/2010/11/china-hijacks-15-of-internet-traffic/, November 2010.

[13] F. Maino, V. Ermagan, A. Cabellos, D. Saucez, and O. Boventure, "Lisp-security (lisp-sec)." draft-ietf-lisp-sec-06, October 2012.

[14] D. Pei, M. Azuma, D. Massey, and L. Zhang, "BGP-RCN: Improving BGP Convergence through Root Cause Notification.," *Computer Networks*, vol. 48, pp. 175–194, June 2005.

[15] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Improving BGP Convergence Through Consistency Assertions," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 902–911 vol.2, 2002.

[16] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," in *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '00, (New York, USA), pp. 175–187, ACM, 2000.

[17] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet Routing Instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 515–528, 1998.

[18] L. Gao and J. Rexford, "Stable Internet Routing without Global Coordination," *SIGMETRICS Perform. Eval. Rev.*, vol. 28, pp. 307–317, June 2000.

[19] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route Oscillations in I-BGP with Route Reflection," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '02, (New York, USA), pp. 235–247, ACM, 2002.

[20] K. Varadhan, R. Govindan, and D. Estrin, "Persistent Route Oscillations in Inter-domain Routing," *Computer Networks*, vol. 32, no. 1, pp. 1–16, 2000.

[21] A. Feldmann, O. Maennel, Z. M. Mao, A. W. Berger, and B. M. Maggs, "Locating Internet Routing Instabilities," in *SIGCOMM*, pp. 205–218, ACM, 2004.

[22] T. G. Griffin and B. J. Premore, "An Experimental Analysis of BGP Convergence Time," in *Ninth International Conference on Network Protocols, 2001*, pp. 53–61, IEEE Computer Society, 2001.

[23] G. Huston, M. Rossi, and G. Armitage, "A Technique for Reducing BGP Update Announcements through Path Exploration Damping," *IEEE J.Sel. A. Commun.*, vol. 28, pp. 1271–1286, Oct. 2010.

[24] N. Valler, M. Butkiewicz, B. Prakash, M. Faloutsos, and C. Faloutsos, "Non-binary information propagation: Modeling BGP routing churn," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2011*, pp. 900–905, 2011.

[25] A. Elmokashfi, A. Kvalbein, and T. Cicic, "On Update Rate-Limiting in BGP," in *IEEE International Conf. on Comm. (ICC), 2011*, pp. 1–6, 2011.

[26] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain Traffic Engineering with BGP," *IEEE Comm. Mag.*, vol. 41, May 2003.

[27] N. Feamster, J. Winick, and J. Rexford, "A model of BGP routing for Network Engineering," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, pp. 331–342, June 2004.

[28] W. Xu and J. Rexford, "MIRO: Multi-path Interdomain Routing," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 171–182, Aug. 2006.

[29] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright, "Rationality and Traffic Attraction: Incentives for Honest Path Announcements in BGP," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 267–278, Aug. 2008.

[30] L. Gao, T. Griffin, and J. Rexford, "Inherently Safe Backup Routing with BGP," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, pp. 547–556 vol.1, 2001.

[31] M. Lad, R. Oliveira, B. Zhang, and L. Zhang, "Understanding Resiliency of Internet Topology against Prefix Hijack Attacks," in *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '07, (Washington, DC, USA), pp. 368–377, IEEE Computer Society, 2007.

[32] K. Sriram, D. Montgomery, O. Borchert, O. Kim, and D. R. Kuhn, "Study of BGP Peering Session Attacks and Their Impacts on Routing Performance," *IEEE J.Sel. A. Commun.*, vol. 24, pp. 1901–1915, Oct. 2006.

[33] K. Sriram, O. Borchert, O. Kim, P. Gleichmann, and D. Montgomery, "A Comparative Analysis of BGP Anomaly Detection and Robustness Algorithms," in *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, CATCH '09, (Washington, DC, USA), pp. 25–38, IEEE Computer Society, 2009.

[34] D. Wendlandt and I. Avramopoulos, "Don't Secure Routing Protocols, Secure Data Delivery," in *In Proc. 5th ACM Workshop on Hot Topics in Networks (Hotnets-V)*, 2006.

[35] A. T. Mizrak, Y. Cheng, K. Marzullo, and S. Savage, "Fatih: Detecting and Isolating Malicious Routers," in *DSN 2005. Proceedings. International Conference on Dependable Systems and Networks, 2005.*, pp. 538–547, 2005.

[36] Fang Fang, A. B. Whinston, M. Parameswaran, and X. Zhao, "Reengineering the Internet for Better Security," *Computer*, vol. 40, no. 1, pp. 40–44, 2007.

[37] O. Nordström and C. Dovrolis, "Beware of BGP attacks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 1–8, Apr. 2004.

[38] B. Kumar and J. Crowcroft, "Integrating Security in Inter-domain Routing Protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 23, pp. 36–51, Oct. 1993.

[39] J. Karlin, S. Forrest, and J. Rexford, “Pretty Good BGP: Improving BGP by Cautiously Adopting Routes,” in *Proceedings of the 2006 IEEE International Conference on Network Protocols*, ICNP ’06, (Washington, DC, USA), pp. 290–299, IEEE Computer Society, 2006.

[40] H. Chan, D. Dash, A. Perrig, and H. Zhang, “Modeling Adoptability of Secure BGP Protocol,” *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 279–290, Aug. 2006.

[41] K. Butler, P. McDaniel, and W. Aiello, “Optimizing BGP security by exploiting path stability,” in *Proceedings of the 13th ACM conference on Computer and Communications Security*, CCS ’06, (New York, USA), pp. 298–310, ACM, 2006.

[42] M. Zhao, S. W. Smith, and D. M. Nicol, “The Performance Impact of BGP Security,” *IEEE Network*, vol. 19, pp. 42–48, Nov. 2005.

[43] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, “Characterizing the Internet hierarchy from multiple vantage points,” in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 618–627, 2002.

[44] T. Griffin, F. B. Shepherd, and G. T. Wilfong, “Policy Disputes in Path-Vector Protocols,” in *Proceedings of the Seventh Annual International Conference on Network Protocols*, ICNP ’99, (Washington, DC, USA), p. 21, IEEE Computer Society, 1999.

[45] L. Gao, “On Inferring Autonomous System Relationships in the Internet,” *IEEE/ACM Trans. Netw.*, vol. 9, pp. 733–745, Dec. 2001.

[46] F. Wang and L. Gao, “On inferring and characterizing internet routing policies,” in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC ’03, (New York, USA), pp. 15–26, ACM, 2003.

[47] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, “The Impact of Routing Policy on Internet Paths,” in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 736–742 vol.2, 2001.

[48] J. Xia and L. Gao, “On the Evaluation of AS Relationship Inferences [Internet Reachability/Traffic Flow Applications],” in *Global Telecommunications Conference, 2004. GLOBECOM ’04. IEEE*, vol. 3, pp. 1373–1377 Vol.3, 2004.

[49] S. Qiu, P. McDaniel, and F. Monrose, “Toward Valley-Free Inter-domain Routing,” in *Communications, 2007. ICC ’07. IEEE International Conference on*, pp. 2009–2016, June 2007.

[50] G. Di Battista, T. Erlebach, A. Hall, M. Patrignani, M. Pizzonia, and T. Schank, “Computing the Types of the Relationships between Autonomous Systems,” *IEEE/ACM Trans. Netw.*, vol. 15, pp. 267–280, Apr. 2007.

[51] N. Feamster, H. Balakrishnan, and J. Rexford, “Some foundational problems in Interdomain routing,” in *3rd ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, pp. 41–46, November 2004.

[52] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure, “Open Issues in Interdomain Routing: A Survey,” *Netwrk. Mag. of Global Internetwkg.*, vol. 19, pp. 49–56, Nov. 2005.

[53] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, “An Analysis of BGP Multiple Origin AS (MOAS) Conflicts,” in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW ’01, (New York, USA), pp. 31–35, ACM, 2001.

[54] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, “Detection of Invalid Routing Announcement in the Internet,” in *Proceedings of the 2002 International Conference on Dependable Systems and Networks*, DSN ’02, (Washington, DC, USA), pp. 59–68, IEEE Computer Society, 2002.

[55] T. Griffin and G. Huston, “BGP Wedgies,” RFC 4264, IETF, 2005.

[56] R. Mahajan, D. Wetherall, and T. Anderson, “Understanding BGP misconfiguration,” in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM ’02, (New York, USA), pp. 3–16, ACM, 2002.

[57] R. Perlman, “Network Layer Protocols with Byzantine Robustness.” PhD thesis, Massachusetts Institute of Technology, MIT-LCS-TR-429, October 1988.

[58] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, “Towards an accurate AS-level traceroute tool,” in *Proceedings of the 2003 Conf. on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM ’03, (New York, USA), pp. 365–378, ACM, 2003.

[59] R. Dube, “A Comparison of Scaling Techniques for BGP,” *SIGCOMM Comput. Commun. Rev.*, vol. 29, pp. 44–46, July 1999.

[60] R. Teixeira and J. Rexford, “A measurement framework for pin-pointing routing changes,” in *Proceedings of the ACM SIGCOMM workshop on Network troubleshooting: research, theory and operations practice meet malfunctioning reality*, NetT ’04, (New York, USA), pp. 313–318, ACM, 2004.

[61] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, “The Case for Separating Routing from Routers,” in *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, FDNA ’04, (New York, USA), pp. 5–12, ACM, 2004.

[62] N. Feamster and J. Rexford, “Network-wide Prediction of BGP Routes,” *IEEE/ACM Trans. Netw.*, vol. 15, pp. 253–266, Apr. 2007.

[63] R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, and L. Zhang, “Quantifying Path Exploration in the Internet,” in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC ’06, (New York, USA), pp. 269–282, ACM, 2006.

[64] N. Feamster and H. Balakrishnan, “Towards a Logic for Wide-Area Internet Routing,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 289–300, Aug. 2003.

[65] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, “Multiprotocol extensions for BGP-4,” RFC 4760, IETF, 2007.

[66] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, “Characterizing the Internet Hierarchy from Multiple Vantage Points,” tech. rep., Berkeley, CA, USA, 2001.

[67] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley, “AS Relationships: Inference and Validation,” *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 29–40, Jan. 2007.

[68] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger, “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, (New York, NY, USA), pp. 163–174, ACM, 2012.

[69] The Internet Engineering Task Force (IETF), “http://www.ietf.org/,” 2013.

[70] RIPE Network Coordination Center, “http://www.ripe.net/,” 2013.

[71] American Registry for Internet Numbers, “https://www.arin.net/,” 2013.

[72] Detecting Route Leaks by Counting - NANOG 41. https://www.nanog.org/meetings/nanog41/presentations/mauch-lightning.pdf, 2007.

[73] CAIDA’s Archipelago Measurement Infrastructure. http://www.caida. org/projects/ark/.

[74] Hetrogeneous Experimental Network. http://mediatools.cs.ucl.ac.uk/nets/hen, 2014.

[75] G. Huston, “Interconnection, Peering, and Settlements,” in *Proc. INET, June 1999*, June 1999.

[76] V. Giotsas and S. Zhou, “Valley-free violation in Internet routing–Analysis based on BGP Community data,” in *Communications (ICC), 2012 IEEE International Conference on*, pp. 1193–1197, June 2012.

[77] P. Gill, S. Goldberg, and M. Schapira, “A Survey of Interdomain Routing Policies.” Presented at NANOG’56, http://www.cs.bu.edu/~goldbe /papers/survey.pdf, October 2012.

[78] R. White, D. McPherson, and S. Sangli, *Practical BGP.* Addison-Wesley, 2005.

[79] J. Postel, “Transmission Control Protocol,” RFC 793, IETF, September 1981.

[80] J. W. Stewart, *BGPv4: Inter-Domain Routing in the Internet.* Addison-Wesley, 1999.

[81] S. Halabi, *Internet Routing Architectures.* Cisco Press, 2000.

[82] Y. Song, A. Venkataramani, and L. Gao, "Identifying and Addressing Protocol Manipulation Attacks in "Secure" BGP," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pp. 550–559, July 2013.

[83] G. Huston, M. Rossi, and G. Armitage, "Securing BGP – A Literature Survey," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, 2011.

[84] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, "A Survey of BGP Security Issues and Solutions," *Proc. of the IEEE*, vol. 98, pp. 100–122, Jan. 2010.

[85] B. Dickson, "Route Leaks – Requirements for Detection and Prevention thereof." draft-dickson-sidr-route-leak-reqts, March 2012.

[86] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 103–116, 2000.

[87] R. White, "Securing BGP through secure origin BGP (soBGP)," *Internet Protocol Journal*, vol. 6, no. 3, 2003.

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

[89] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDanial, and A. Rubin, "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)*, 2003.

[90] A. Boldyreva and R. Lychev, "Provable Security of S-BGP and Other Path Vector Protocols: Model, Analysis and Extensions," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, (New York, NY, USA), pp. 541–552, ACM, 2012.

[91] Internet Assigned Numbers Authority (IANA), "http://www.iana.org/," 2013.

[92] G. Huston, R. Loomans, and G. Michaelson, "A Profile for Resource Certificate Repository Structure," RFC 6481, IETF, February 2012.

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

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

[95] Open Networking Foundation (ONF), "https://www.opennetworking.org/," 2013.

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

[97] R. Bush, “BGPSEC Operational Considerations.” draft-ietf-sidr-bgpsec-ops, May 2012.

[98] K. Sriram, “BGPSEC Design Choices and Summary of Supporting Discussions.” draft-sriram-bgpsec-design-choices, July 2013.

[99] K. Sriram and R. Bush, “Estimating CPU Cost of BGPSEC on a Router.” IETF 82 SIDR WG Meeting, March 2012.

[100] D. McPherson, “Route Leak Attacks Against BGPSEC.” draft-foo-sidr-simple-leak-attack-bgpsec-no-help, November 2011.

[101] Global Routing Operations (GROW) Working Group IETF, “http://datatracker.ietf.org /wg/grow/,” 2013.

[102] K. Sriram and D. Montgomery, “Enhancement to BGPSEC for Protection against Route Leaks.” draft-sriram-route-leak-protection, July 2014.

[103] K. Sriram and D. Montgomery, “Design Discussions and Comaparison of Replay-Attack Protection Mechanisms for BGPSEC.” draft-sriram-replay-protection-design-discussion, September 2012.

[104] R. Gagliano, K. Patel, and B. Weis, “BGPSEC router key rollover as an alternative to beaconing.” draft-ietf-sidr-bgpsec-rollover, April 2013.

[105] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg, “On the Risk of Misbehaving RPKI Authorities,” in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, HotNets-XII, (New York, NY, USA), pp. 16:1–16:7, ACM, 2013.

[106] E. Osterweil, T. Manderson, R. White, and D. McPherson, “Sizing Estimates for a Fully Deployed RPKI,” Tech. Rep. 1120005 version 2, Verisign, 2012.

[107] S. Kent and K. Sriram, “RPKI rsync Download Delay Modeling.” IETF 86 SIDR WG Meeting, March 2013.

[108] T. Bruijnzeels, O. Muravskiy, and B. Weber, “RPKI Repository Analysis and Requirements.” draft-tbruijnzeels-sidr-repo-analysis-00, February 2013.

[109] R. Bush, “[sidr] Scaling properties of caching in a globally deployed RPKI/BGPSEC system.” http://www.ietf.org/mail-archive/web/sidr/ current/msg05351.html, November 2012.

[110] P. Gill, M. Schapira, and S. Goldberg, “Let the Market Drive Deployment: A Strategy for Transitioning to BGP Security,” in *Proceedings of the ACM SIGCOMM 2011 conference*, SIGCOMM ’11, (New York, USA), pp. 14–25, ACM, 2011.

[111] R. Lychev, S. Goldberg, and M. Schapira, "BGP Security in Partial Deployment: Is the Juice Worth the Squeeze?," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pp. 171–182, ACM, 2013.

[112] F. Mejia, R. Gagliano, C. Martinez, and G. Rada, "Implementing rpki-based origin validation one country at a time. the ecuadorian case study." draft-fmejia-origin-a-country, February 2014.

[113] R. Crozier and J. Hutchinson, "Dodo cops blame for national internet outages." http://www.itnews.com.au/News/291364,dodo-cops-blame-for-national-internet-outages.aspx, Feb. 2012.

[114] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford, "How Secure Are Secure Interdomain Routing Protocols," in *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, (New York, NY, USA), pp. 87–98, ACM, 2010.

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

[116] B. Dickson, "Route Leaks – Proposed Solutions." draft-dickson-sidr-route-leak-solns, March 2012.

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

[118] The Network Simulator - NS-2. http://www.isi.edu/nsnam/ns/, 2014.

[119] BGP++. http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/, 2014.

[120] V. Krishnamurthy, M. Faloutsos, M. Chrobak, J.-H. Cui, L. Lao, and A. G. Percus, "Sampling large internet topologies for simulation purposes," *Computer Networks*, vol. 51, no. 15, pp. 4284 – 4302, 2007.

[121] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *SIGCOMM Comput. Commun. Rev.*, vol. 29, pp. 251–262, Aug. 1999.

[122] M. Yannuzzi, R. Serral-Gracia, and X. Masip-Bruin, "Large-Scale Tests and Complexity Analysis on Path-State Vectors," Technical Report (Confidential), UPC, 2010.

[123] The CAIDA UCSD IPv4 Routed /24 Topology Dataset - 01.04.2014. http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml., 2014.

[124] What's Software Defined Networking (SDN)? https://www.sdncentral.com/what-the-definition-of-software-defined-networking-sdn/.

[125] Open Networking Foundation, "OpenFlow Switch Specification, version 1.3.2." https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.2.pdf.

[126] OpenDaylight—A Linux Foundation Collaborative Project, "http://www.opendaylight.org/."

[127] Quagga Routing Suite, "http://www.nongnu.org/quagga/," 2013.

[128] OPENER, "http://www.craax.upc.edu/opener," 2013.

[129] J. Chiappa, "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture." http://mercury.lcs.mit.edu/ jnc/tech/endpoints.txt, 2000.

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

[131] D. Massey, L. Wang, B. Zhang, and L. Zhang, "A scalable routing system design for future internet," *ACM SIGCOMM Workshop on IPv6*, August 2007.

[132] F. Templin, "The IPvLX Architecture." draft-templin-ipvlx-08, November 2007.

[133] R. Whittle, "Ivip (Internet Vastly Improved Plumbing) Architecture." draft-whittle-ivip-arch-04, September 2010.

[134] X. Zhang, P. Francis, J. Wang, and K. Yoshida, "Scaling IP Routing with the Core Router-Integrated Overlay," in *In ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, pp. 147–156, IEEE Computer Society, 2006.

[135] C. Vogt, "Six/one router: A scalable and backwards compatible solution for provider-independent addressing," in *Proceedings of the 3rd International Workshop on Mobility in the Evolving Internet Architecture*, MobiArch '08, (New York, NY, USA), pp. 13–18, ACM, 2008.

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

[137] M. Wasserman and F. Baker, "IPv6-to-IPv6 Network Address Translation (NAT66)." draft-mrw-behave-nat66-02, May 2009.

[138] W. Ramirez, X. Masip-Bruin, M. Yannuzzi, R. Serral-Gracia, A. Martinez, and M. S. Siddiqui, "A survey and taxonomy of id/locator split architectures," *Comput. Netw.*, vol. 60, pp. 13–33, Feb. 2014.

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

[140] V. Fuller, D. Lewis, V. Ermagan, and A. Jain, "LISP Delegated Database Tree." draft-ietf-lisp-ddt-01, September 2013.

[141] D. Saucez, L. Iannone, and O. Bonaventure, "LISP Threat Analysis." draft-ietf-lisp-threats-10, July 2014.

[142] R. Gagliano, “A profile for endpoint identifier origin authorizations (IOA),” internet-draft, IETF, Mar. 2009.

[143] TEFIS, “http://www.tefisproject.eu,” 2013.

[144] JUNIPER’s JUNOS SDK, “https://developer.juniper.net/content/develop-overview/junos-sdk/getting-started.page,” 2014.

[145] RESTful Web services, “http://www.ibm.com/developerworks/webservices/library/ws-restful/,” 2013.

[146] Command Line Interface, “http://en.wikipedia.org/wiki/command-line_interface,” 2013.

[147] Path-State Protocol (PSP). http://www.craax.upc.edu/psp.html, 2012.

[148] LACNOG 2012. http://www2.lacnic.net/sp/eventos/lacnicxviii/index.html, Montevideo, Uruguay, Oct./Nov. 2012.