Politecnico di Milano
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

Study and measurements of the RPKI deployment

Relatore: Prof. Giacomo VERTICALE

Tesi di laurea di:
Daniele IAMARTINO Matr. 795906

Anno Accademico 2014–2015

To those who made this possible

Acknowledgments

First of all, I would like to thank my advisor, professor Giacomo Verticale, who supported and patiently advised me while I was continuously changing my mind about this thesis. He also allowed me to work on this topic that I proposed, even though it is outside of his current research works.

A very special thanks to Cristel Pelsser and Randy Bush, the Internet researchers who introduced me to this topic when I was an intern at IIJ-II in Tokyo, gave me guidelines for doing these measurements and helped me in publishing initial results of this work. Thanks also to Rob Austein, who gave me a lot of answers on doubts that I had while working with *rcynic* and RPKI in general while I was at IIJ-II.

I owe many of my technical skills, interests and «*Internet culture*» to a student association of my university, POuL (*Politecnico Open unix Labs*), where in the last 6 years I've met some of the brightest, passionated and skilled «*hackers*». In particular I would like to give special thanks (in random order) to Alessandro, Stefano, Michele, Massimo, Riccardo, Federico, Emanuele, Radu, Luca, Andrea, Fabrizio, Edoardo, Sgrammaticato, Salvatore, Sara and Fabio.

Other thanks for people in *NECST* laboratory (Stefano Zanero, Federico Maggi), *Tower of Hanoi* computer security team («CTF captain» Alessandro Barenghi) and my internship's teammates at Google UK (Pierre, Florian, Claudiu, Luca).

Finally I would like to thanks Yukino for the cheering and support provided while writing this thesis, as well as my family for supporting me during my studies and my working experiences.

Abstract

BGP, the *interdomain* routing protocol of the Internet, was designed long ago without considering security. In the last decade, the Internet started to experience a growing number of «routing incidents». These incidents can be the result of malicious attacks, but more often they are due to operational mistakes, or «*misconfigurations*» made by some network operator. Because of the forwarding mechanisms of BGP, it is easy for a single operational mistake to propagate and cause problems to a lot of networks, resulting in intercepting or *«blackholing»* other networks traffic.

In order to address this, prefix origin validation, based on the *Resource Public Key Infrastructure* (RPKI), was proposed and developed.

Many discussions are going on about RPKI: doubts on its usefulness, as well as doubts on quality of data present in its repositories.

In this thesis we would like to achieve two goals. The first goal: give a clear overview on types and reasons of routing incidents, as well as discussing existing solutions to address them. The second goal: analyze the deployment of RPKI and the quality of the data contained in it, via measurements from several points of view. This second goal will provide to the reader and to network operators an exhaustive analysis on the quality of RPKI.

In our results, we show problems detected in the history (such as RIR's RPKI repositories not operationally reliable), as well as current errors in the registration of RPKI resources. However, we also show how RPKI deployment is positively increasing, the number of problems is decreasing, what are the causes of this problems and why they are not so serious. Moreover, we present the work we've done in parallel with this thesis in order to help fixing these problems by publicizing measurements.

Sommario

BGP, il protocollo di *interdomain routing* di Internet, fu progettato in passato senza considerarne la sicurezza. Nell'ultimo decennio, Internet è stato colpito da un numero crescente di «incidenti di routing» (o «routing incidents»). Questi incidenti possono essere il risultato di attacchi malevoli, ma più spesso sono causati da errori umani o errori di configurazione generati da operatori di reti. A causa dei meccanismi di inoltro di BGP, è facile che un singolo errore di configurazione possa propagarsi e causare problemi a tante altre reti, arrivando ad intercettare o interrompere il traffico destinato ad altre reti.

Per risolvere ciò, la «prefix origin validation», basata sulla *Resource Public Key Infrastructure* (RPKI), è stata proposta e sviluppata.

Molte discussioni sono state spese riguardo ad RPKI: dubbi sulla sua utilità, così come dubbi sulla qualità dei dati presenti nei suoi *repositories*.

In questa tesi vogliamo raggiungere due scopi. Il primo scopo: fornire una panoramica completa sui motivi e sui tipi degli incidenti di routing, oltre a discutere le attuali soluzioni per fronteggiarli. Il secondo scopo: analizzare lo sviluppo di RPKI e la qualità dei dati in esso contenuti, tramite misurazioni da diversi punti di vista. Con questo secondo scopo forniremo al lettore ed agli operatori di reti una analisi esaustiva sulla qualità di RPKI.

Nei nostri risultati, mostriamo problemi rilevati nell'analisi storica dei dati (come ad esempio il fatto che i repository RPKI dei RIR non sono operativamente affidabili), così come mostriamo errori attualmente presenti nella registrazioni di risorse RPKI. Tuttavia mostriamo anche come lo sviluppo di RPKI sta crescendo positivamente, il numero di problemi sta decrescendo nel tempo, analizziamo le cause di questi problemi e spieghiamo perché non sono così gravi. Inoltre, presentiamo il lavoro di pubblicazione delle misure, fatto in parallelo alla tesi con lo scopo di aiutare la comunità a risolvere questi problemi.

Contents

Introduction **1**

1 Background and problem analysis **4**
1.1 The Internet routing and BGP 4
1.1.1 Routing 4
1.1.1.1 Intradomain (or «interior») routing 4
1.1.1.2 Interdomain (or «exterior») routing 5
1.1.2 The Border Gateway Protocol (BGP) 6
1.2 Address space assignment 7
1.2.0.1 The Regional Internet Registries (RIRs) 7
1.2.0.2 Legacy address space 9
1.3 Types of routing incidents 9
1.3.1 Hijacks 9
1.3.1.1 Prefix hijacks 10
1.3.1.2 Sub-prefix hijacks 10
1.3.1.3 Path-shortening hijacks 10
1.3.2 Route leaks 11
1.3.2.1 Sub-prefix route leak 11
1.3.3 «*Self de-aggregation*» 12
1.4 Impact of routing incidents 12
1.4.1 Traffic blackhole 12
1.4.2 Traffic interception 12
1.4.3 High load and crashes 12
1.5 Examples of routing incidents 13
1.5.1 Indosat incident (2014) 13
1.5.2 Pakistan YouTube incident (2008) 13
1.5.2.1 The incident 13
1.5.2.2 Mitigating the hijack 14
1.5.2.3 Conclusions 15
1.5.3 DEFCON 16 - «Stealing The Internet» (2008) 16
1.5.3.1 The attack 16

1.5.3.2 Bypassing Internet Routing Registries 16
1.5.3.3 Sending traffic back to the hijacked network 17
1.5.3.4 Hiding the attack 18
1.5.3.5 Conclusions 19
1.5.4 Spammers using unallocated IP space («*IP squatting*») . . . 19
1.5.4.1 The attack 19
1.5.4.2 AS Telelatina case of IP squatting 20
1.5.4.3 Internet Routing Registry bypass 20
1.5.4.4 Conclusions 21
1.5.5 Moratel route leak (2012) 21
1.5.5.1 The incident 21
1.5.5.2 Conclusions 22
1.5.6 Telekom Malaysia route leak (2015) 22
1.5.6.1 The incident 22
1.5.6.2 Conclusions 22
1.5.7 BGP Optimizer causing fake routes (2015) 23
1.5.7.1 The incident 23
1.5.7.2 Amazon AS example 23
1.5.7.3 Conclusions 24
1.5.8 BGP Hijacking for Cryptocurrency Profit (2014) 24
1.5.8.1 Bitcoin mining 25
1.5.8.2 BGP 25
1.5.8.3 Conclusions 26
1.5.9 Incidents summary 26
1.6 Common reasons for accidental routing incidents 26
1.6.1 IGP Redistribution 26
1.6.2 Wrong BGP community 27
1.6.3 Reliance on upstream filtering 27
1.6.4 Old configuration 27
1.6.5 Typing error 27

2 Available defenses **28**
2.1 Monitoring and mitigating hijacks 28
2.1.1 Data sources 28
2.1.1.1 BGP collectors 28
2.1.1.2 Looking glass servers 29
2.1.2 Getting alerts 29
2.1.2.1 Issues with BGP monitoring 29
2.1.3 Fighting hijacks by *de-aggreating* prefixes 30
2.1.4 Contacting the network causing problems or upstream 30

2.2 BGP prefix filtering . . . 30
2.2.1 Prefix filter list . . . 31
2.2.2 Access lists . . . 31
2.2.3 Route map . . . 32
2.2.4 BGP Maximum-Prefix filter . . . 32
2.2.5 BGP communities for route leaks . . . 32
2.2.6 Limits of the prefix filtering . . . 32
2.2.6.1 Prefix filtering works only on customer links . . . 32
2.2.6.2 Manually maintain filters . . . 33
2.2.6.3 Automatically building filters . . . 33
2.2.6.4 Hard to apply in Internet eXchange Points . . . 33
2.2.6.5 Low incentives in deploying filtering . . . 34
2.2.6.6 Filtering «failing silently» . . . 34
2.3 Internet Routing Registries (IRRs): Overview and issues. . . . 34
2.3.1 Issues with IRRs . . . 35
2.3.1.1 Data is (mostly) not authoritative . . . 35
2.3.1.2 Lack of authentication . . . 35
2.3.1.3 Quality of the data . . . 36
2.3.1.4 IRR registration and *IRR-based* prefix filtering . . . 36
2.4 Resource Public Key Infrastructure (RPKI) . . . 37
2.4.1 RPKI-based BGP route origin validation . . . 38
2.4.2 RPKI-based origin validation vs prefix filtering and IRRs . . 39
2.4.3 Preventing routing incidents with RPKI-based origin validation 40
2.4.3.1 ROA Maximum length benefits . . . 40
2.4.3.2 Prevent incidents examples . . . 40
2.4.3.3 Prevent hijacking of unused address space . . . 40
2.4.4 BGPSEC: Path validation with RPKI . . . 41
2.4.5 RPKI issues and need for measurements . . . 42
2.4.5.1 ARIN's agreements legal issues . . . 42
2.4.5.2 Fear of court orders and law enforcement . . . 43
2.4.5.3 Revealing business relations . . . 43
2.4.5.4 Unclear quality of RPKI repository data and Related Works . . . 44
2.4.5.5 Local policies and dropping invalid announcements . 44

3 Measurements on RPKI **45**
3.1 RPKI repositories registration analysis . . . 45
3.1.1 Current registration status . . . 45
3.1.2 Registration trends among repositories . . . 46
3.2 Validating BGP data against RPKI repositories . . . 47

3.2.1 Data sources 48
3.2.2 Validation process and measurement infrastructure 48
3.2.2.1 Optimizations 49
3.2.2.2 Classifying data 50
3.2.2.3 Graph generation 51
3.2.3 Overview on validated BGP announcements 51
3.2.4 Counting prefixes 52
3.2.5 Reasons of invalidity 55
3.2.6 Taking prefix coverage into account 56
3.2.7 Effect of using a single BGP route collector/monitor 59
3.2.8 Provider with a «shadowing ROA» 61
3.2.8.1 Example scenario 61
3.2.8.2 Measuring «shadowed» prefixes 61
3.2.8.3 Missing announcements 63
3.2.9 Maximum length problems 63
3.2.10 Conclusions 64
3.3 Analysis on the quality of ROAs 64
3.3.1 Problematic ROAs classification 64
3.3.2 Analysis results 65
3.3.3 Other problem ROAs 67
3.3.4 IPv4 address space transfers 68

4 Other measurements related to RPKI **70**
4.1 Providing measurements to network operators 70
4.1.1 Structure of the website 70
4.2 Root and TLDs DNS nameservers measurements 72
4.2.1 DNS root servers 72
4.2.2 TLD name servers 73
4.3 Related Work: RPKI coverage of websites 74

Conclusions **75**

Bibliography **84**

List of Figures

1.1 Map of Regional Internet Registries. (Source: *wikipedia.org*, License: CC BY-SA 3.0) 8
1.2 Regional Internet Registries and IP addresses allocation hierarchy (source [89]). 8
1.3 Illustration of the generic route leak described in section 1.3.2 11
1.4 Main Autonomous Systems involved in the Pakistani YouTube incident 14
1.5 Traffic flow few minutes after the hijacking of the Pakistani YouTube incident 15
1.6 Topology of the ASes, announcement and traffic before the prefix hijacking. AS numbers 10, 20, etc... are falsified for illustration purposes. 16
1.7 AS Topology in case of a «normal» sub-prefix hijack 17
1.8 AS Topology after the complete hijack 18
1.9 Pattern in the announcement of IP prefixes by time. Seen with RIPE NCC ATLAS tool [21], source: [96]. 20
1.10 AS Topology of the Moratel's `8.8.8.0/24` route leak. 21
1.11 AS Topology after the route leak 22
1.12 AS Topology of the Telekom Malaysia's route leak 23
1.13 AS Topology of the Enzu Inc. route leak for the Amazon's prefix. Note: the topology has been slightly modified from the original, for illustration purposes. 24

3.1 Number of authenticated ROA files below the six trust anchors. The discontinuous increases in number of ROAs observed for RIPE NCC occur during key rollovers. LACNIC and APNIC face a loss of valid ROAs for roughly seven months, likely due to an expiration of the manifest related to their trust anchor certificate. There is a hole in our data, for all trust anchors between July and August 2013. 47
3.2 Number of (IP prefix, Origin AS) pairs not covered by ROA record, valid invalid or containing AS_TRANS/AS_SET as origin. 51

3.3 Percentage of (IP prefix, Origin AS) pairs correctly validating against some ROA, failing validation or containing AS_SET or AS_TRANS as origin. Note: the numbers shown in the table under the plot are percentages relative to the total of the four categories shown, not referring to all existing announcements. 52
3.4 Total number of unique prefixes announced, compared to the number of prefixes covered by some RPKI ROA record. 54
3.5 Percentage of prefixes that, after RPKI-based origin-validation are classified as «valid», «invalid» or «valid and invalid». The last category is regarding prefixes which are valid for some announced origins and invalid for others. Note: the numbers shown in the table under the plot are percentages relative to the total number of prefixes covered by some RPKI ROA. 54
3.6 Breakdown of invalid prefixes, by failing cause, as seen by *Route Views LINX* monitor. Note: the numbers shown in the table under the plot are percentages relative to the total number of «invalid only» and «valid and invalid» prefixes for the given time snapshot. 56
3.7 Validity status of routes seen by *Route Views LINX* monitor between June 2012 and August 2015. The first (green, pink and yellow) bar shows the status of prefixes independently from the existence of covering prefixes. The second bar (blue, red and grey) illustrates the reachability of a prefix considering that an invalid prefix might be covered by another valid or «ROA not found» prefix. Some relevant data points in table 3.3. 58
3.8 Some of the top ASes announcing unreachable prefixes, sorted by number of unreachable prefixes announced, for the data from 15 August 2015 of Route Views LINX monitor. 59
3.9 Reachable and Unreachable prefixes from different RouteViews monitors. 60
3.10 Number of invalid prefixes due to maximum length problem, invalids with wrong origin AS but correct AS on the AS_PATH and invalids for other reasons. Note: the numbers show in the table under the graph are percentages relative to the total number of «invalid only» and «valid and invalid» prefixes. 62
3.11 Number of address blocks transferred for each month from October 2012 to November 2015 in the RIPE NCC region. Data from [29]. . 68

4.1 Screenshot from our website [23] reporting quality of ROAs in RPKI repositories. 71

4.2 Screenshot from our website [23] , showing an example of a «Problem ROA» and its related BGP announcements. 72
4.3 RPKI-coverage among all TLDs name servers and among ccTLDs only. 74

List of Tables

1.1 Summary of all incidents seen in section 1.5 26

2.1 Summary of all types of incidents seen in section 1.5, and whether they could have been prevented by using RPKI-based route origin validation. 41

3.1 Deployment status of the registration of IPv4 addresses on September 1, 2015 (data from our RPKI data archive presented in section 3.2.1) compared to the allocation of IPs by these RIRs on the same day ([10–14]). 46

3.2 Counters and percentages of prefixes classified as «valid», «invalid» or «valid and invalid» for relevant data points relative to figure 3.5. . 55

3.3 Relevant data points relative to figure 3.7. 58

3.4 Reachable and Unreachable prefixes from different RouteViews monitors on 24 September 2013 60

3.5 Reachable and Unreachable prefixes from different RouteViews monitors on 20 August 2014 60

3.6 Quality of ROA records registered in the 5 RIRs, computed with data from Route Views LINX monitor of 15 August 2015. The categories shown are described in section 3.3.1. Note: the numbers shown here are relative to number of *unique* ROA *records*. 66

Introduction

The *Border Gateway Protocol* (BGP), the *interdomain* routing protocol of the Internet, is often mentioned as «*the glue that holds the Internet together*» because of its crucial role in exchanging network reachability information. BGP allows any host and router on the Internet to have its packets routed and delivered through different organizations or administrative domains, achieving the goal of interconnecting together all isolated networks part of the Internet.

BGP is also the result of a pragmatical solution designed decades ago, without taking security into account. With the growing number of networks connected to the Internet, the problem of securing BGP, or interdomain routing in general, became more and more important over the years.

In the last decade, the Internet started to experience a growing number of «routing incidents». As today, these incidents are happing on a daily basis, sometimes affecting few isolated networks and sometimes causing worldwide problems. These incidents are often the result of operational mistakes (or «mis-configurations») made by a single network operator, while in other few cases they are the result of malicious attacks. In this work we would like to focus mainly on the operational mistakes, where network operators often «mis-originate» IP prefixes belonging to other organizations, like the well-known *«Pakistani YouTube»* incident of 2008 (section 1.5.2).

A number of solutions have been developed over the years in order to solve this problem, such as manual prefix filtering or filtering based on «*Internet Routing Registries*» data [1]. The latest solution, standardized by the *Internet Engineering Task Force* (IETF) is route origin validation based on the RPKI, or *Resource Public Key Infrastructure* [68]. With RPKI, a network operator can cryptographically sign statements about which IP prefixes can be originated by his network, and add them in a *public key infrastructure* repository managed by each *Regional Internet Registry*.

The deployment of RPKI is suffering of common problems in the deployment of new technologies, as well as problems specific to RPKI itself.

Many network operators are holding on RPKI deployment due to: limited knowledge of it, doubts on its efficacy or doubts on the quality of the data currently contained. In fact, some of the early works measuring RPKI data quality shown

a lot of bogus information registered [98]. These and other discussions resulted in operators debating whether RPKI is ready for deployment and whether it is safe to start dropping BGP announcements considered invalid by RPKI-based validation.

In this thesis we would like to achieve two main goals. Firstly we would like to provide a clear overview on origin and reasons for routing incidents, analyze several of them in detail and study all current solutions to monitor, prevent or mitigate them. This will provide to the reader and researchers of routing security a good overview on what are these incidents and why it is not so easy to prevent them. Secondly we will analyze the deployment of RPKI and the quality of the data contained in it via measurement from several points of view. This second goal will provide to the reader and to network operators an exhaustive analysis on the quality of RPKI, in order to reduce doubts and encourage deployment.

We already published [61] and presented [60] most of these results. In this thesis our goal is to provide a complete view of the routing incidents problem, and more deeply analyze some of the data already published, in an up-to-date version.

Regarding a complete overview on the routing incidents, the problem of BGP «mis-originations» and «hijacks» has longly been discussed in several research works, such as [51, 73]. Many of these works analyze specific issues but often fail to give a complete overview of the problems in routing security and current solutions. In [73] most of the types of incidents are presented, but the authors fails to give historical examples that could better explain reasons of the incidents described. [50] is probably the most complete «overview» work we can find, but still lacks on explaining some types of incidents (such as *self de-aggregations*) and skip what we consider important details (such as discussing about the IRRs).

Regarding measurements of RPKI deployment and quality or RPKI data, there have been a number of browsers of RPKI data, such as [63], [46] and [88]. [46, 63] provide snapshots of route validation in specific deployments, while [88] is for analyzing the PKI certificates alone. Here we go further as we study route validation over an extended period of time, by using an historical archive of RPKI data collected since 2012. In addition, we provide statistics regarding the RPKI infrastructure, and the registration of resources and events caused by the operation of the infrastructure. We try also to analyze the origin of observed problems in RPKI, as well as helping in fixing them.

From the RPKI measurements point of view, [100] is probably the nearest work to ours, but did not provide exhaustive details about reasons behind these invalids, did not consider prefix coverage, nor shown an exhaustive history showing the trends in the deployment. Moreover, [100] was published in 2012, when deployment had just started.

In the results of our work, we show several problems that we detected in the history of RPKI (such as RIR's RPKI repositories not operationally reliable), as

well as current errors in the registration of RPKI resources. However, we also show how RPKI deployment is positively increasing, the number of problems is decreasing, what are the causes of this problems and why they are not so serious.

The thesis is organized as follows. In chapter 1 we introduce interdomain routing, possible types of routing incidents and we deeply analyze and discuss several historical incidents. In chapter 2 we present existing solutions for monitoring, preventing or mitigating routing incidents and we discuss about their limitations or problems. In chapter 3 we measure RPKI deployment from several points of view, as well as study the different causes of mismatch between route advertisements and RPKI registrations. In chapter 4 we present the work we done in publicizing some of the problems we found and we also discuss other «minor» measurements related to RPKI deployment.

Chapter 1

Background and problem analysis

1.1 The Internet routing and BGP

1.1.1 Routing

Networks running the Internet Protocol use a *table-based* routing. This means that each router in a network uses a table containing a list of «routes» in order to determine which interface to use and which host to forward a packet to for every given destination. A «route» entry in the routing table is usually composed by a *destination prefix* and the *interface* and *host* to forward the packet to for all destination addresses starting with such prefix. The routing table of each router has to reflect how everything is connected at any given time in order to successfully forward packets.

These routing tables could be manually maintained for very small networks, but they are usually updated by a ***dynamic routing protocol***.

Routing protocols are responsible for disseminating up-to-date network reachability information between routers, and for choosing a path to reach any available destination of the network. When the topology changes, the routing protocol will take care of updating all routing tables accordingly.

Routing can be classified in two main classes: ***Intradomain Routing*** and ***Interdomain Routing***.

1.1.1.1 Intradomain (or «interior») routing

Intradomain routing protocols are those used within a single *Autonomous System* («AS» or «ASes» in the following).

«An Autonomous System is a connected group of one or more IP prefixes run by one or more network operators which has a single and clearly defined routing

policy» [53].

One way of doing this is by using a ***distance vector*** routing protocol, like the *Routing Information Protocol* (RIP) [74]. The RIP protocol basically works by broadcasting the contents of the routing table of each router periodically over every connection and listening for other routers to do the same. All routes are added to the routing table and later broadcasted again. Every route has a «hop count» that indicated the distance to the destination network, so routers have a way to select the best route for the same destination.

Another approach is the one of ***link state*** routing protocols, like the *Open Shortest Path First* (OSPF) protocol [77]. The OSPF protocol keeps a topology map of the network and sends updates to the other routers only when something changes. When an update is sent, all routers recompute the topology map using the Dijkstra's Shortest Path First algorithm. OSPF takes into account the cost (link bandwidth) rather than the number of hops. The *Intermediate System To Intermediate System* (IS-IS) protocol [47] is also an example of another very popular link state routing protocol.

Interdomain routing protocols are often abbreviated as «IGP».

1.1.1.2 Interdomain (or «exterior») routing

Periodically broadcasting all the routes of all routers in all networks, or keeping topology information about every single link connection of each network isn't possible for the entire Internet because it's not a scalable approach and it's also not needed.

Therefore we need interdomain routing protocols, being able to relay *aggregated* routing information between different ASes.

Let's assume we are in a given AS «`AS1`» and we want to send a packet to another AS «`AS2`». We don't need complete routing information about all existing links of both ASes at each routing point. What we need is routing information about how to reach a router on the «border» of `AS1`, then how to reach `AS2`, and from there the information for reaching the destination inside `AS2`. Because of the «bordered» structure of Internet's *Autonomous Systems,* this solution is often optimal and allow to scale the routing process.

In other words, an interdomain routing protocol treats autonomous systems as a single nodes and determine the routing among them, without considering topology information inside each autonomous system.

ARPANET/Internet went through using several interdomain routing protocols, like the Gateway-to-Gateway Protocol (GGP) in 1982 [54], the Exterior Gateway Protocol (EGP) in 1984 [76] and finally the Border Gateway Protocol (BGP) since 1989 [70].

Exterior routing protocols are often abbreviated as «EGP».

1.1.2 The Border Gateway Protocol (BGP)

BGP is «*de facto*» *interdomain* routing protocol currently used by all ASes on the Internet to advertise routes. BGP is often mentioned as «*The glue that holds the Internet together*» because of its crucial role in exchanging network reachability information between different organizations that would be otherwise isolated networks not able to communicate with each other.

BGP is the result of a pragmatical solution to a problem, standardized and updated multiple times: in 1989 [70] («BGP Version 1»), in 1995 [85] and finally in 2006 [87] («BGP-4»).

Describing in detail all mechanisms, state machines and *tie-breakers* of BGP is outside the scope of this document. In the following we will try to give a general overview of the protocol.

BGP is a *path-vector* routing protocol which can be used by an AS for «announcing» or «advertising» to all its neighbors all the IP prefixes it can reach. A *route announcement* is the equivalent of requesting other networks to send traffic for the announced destination prefix.

Typically, an AS (identified by an «AS number») is connected (or is «peering») to other ASes (or «peers») with a direct physical link and communicating with them using BGP on top of TCP/IP. The AS is originating announcements for its own IP prefixes, sending them to other ASes, as well as receiving announcements from peers and forwarding them to other peers.

Each time an AS is receiving and forwarding a route announcement, it is also appending its own AS number to the «AS path» (or «`AS_PATH`») attribute contained inside the announcement message.

An AS will receive multiple announcements for the same IP prefix going through different paths and with different attributes. Each AS has to choose which announcement to use for forwarding traffic to the announced destination IP prefix. This is done according to several common or custom «*tie-breakers*». The main and most important rule is selecting the announcement with the shorter `AS_PATH` length. After an announcement is selected for the given prefix, an entry is added to the routing table of the BGP router, forwarding traffic to the selected announcement's path.

Once an IP packet is forwarded to the BGP router, the router will lookup in the routing table the *longest-prefix match* entry for the destination address of the packet, and will send it to the next-hop specified in such entry.

It is important to note how, thanks to this *longest-prefix match* mechanism, all announcements of more specific prefixes are always going to have precedence on less-specifics.

For example, let's say that a BGP router «`A`» is receiving multiple announcements for `10.0.0.0/16` from different peers `B`,`C`,`D`. Router `A` chooses to select the

announcement coming through peer `B` and creates an entry in the routing table for such prefix, using `B` as *next-hop* address. Once a packet, containing `10.0.1.1` as destination IP address, is coming to router «`A`», the lookup in the routing table is resulting in having the packet forwarded to `B`. However, if `C` is announcing to `A` prefix `10.0.1.0/16` , another entry will be added in the routing table of router `A`, resulting in the previous packet to be forwarded to `C` instead of `B`.

For this reason it is often said that «more-specific prefixes always win» within BGP. This is an important fact that we will have to remember while discussing about *hijacks* in the following.

Note: In the following we will often use the notation «`/m`» (such as «`/16`» or «`/24`») where `m` is the prefix length of a generic IP prefix.

1.2 Address space assignment

The key feature allowing scalability to the Internet routing is the *aggregation* of address blocks. For this reason, it's not possible to allocate single addresses, but only «blocks» of addresses, or *«subnets»*, which could be of different sizes. Here we briefly present the Internet address allocation hierarchy, as it will be considered later when discussing about authorities for IP address allocations.

On September 1981, the RFC for the Internet Protocol specification [83] was published in the IETF.

Since the publication of the specification, several organizations started to request address space.

In this first phase of address allocation, *Jon Postel* (the editor of the specification) was taking care of manually allocating address blocks for each organization, depending on their size.

The new Internet Protocol started operating on the famous «*ARPANET flag day*» of the 1st January 1983, using a *class-ful* addressing system.

In order to regulate and properly track the growing allocation of addresses, in 1990 the IETF created the *Internet Assignment Numbers Authority* (IANA).

1.2.0.1 The Regional Internet Registries (RIRs)

Between 1992 and 2005 ***Regional Internet Registries*** (RIRs) were gradually created in order to better serve the Internet communities outside the United States [89] (see figure 1.1):

- 1992 - RIPE NCC (Europe, Russia, the Middle East, and Central Asia)
- 1993 - APNIC (Asia, Australia, New Zealand, and neighboring countries)
- 1997 - ARIN (United States, Canada, parts of the Caribbean region, and Antarctica)

- 2002 - LACNIC (Latin America and parts of the Caribbean region)
- 2005 - AfriNIC (Africa)

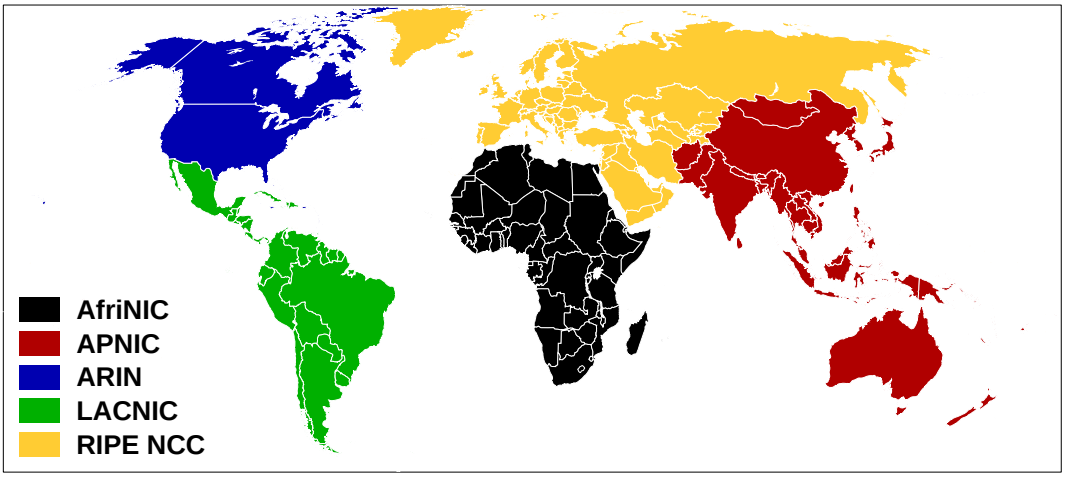

Figure 1.1: Map of Regional Internet Registries. (Source: *wikipedia.org*, License: CC BY-SA 3.0)

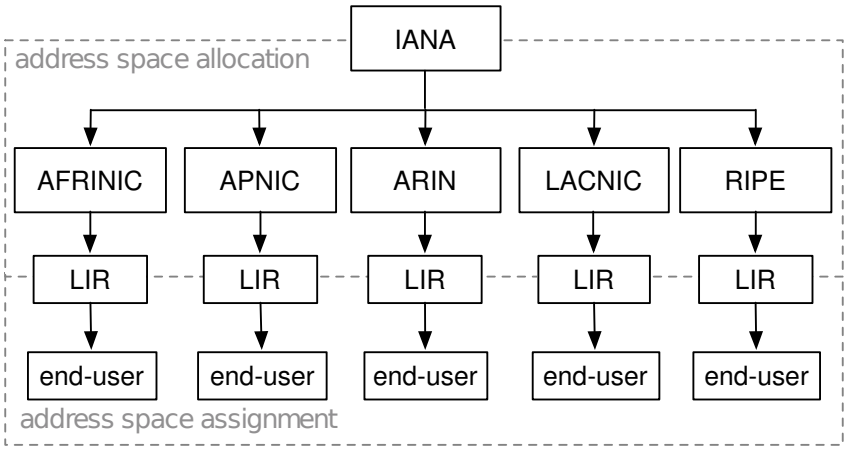

Figure 1.2: Regional Internet Registries and IP addresses allocation hierarchy (source [89]).

The current address allocation hierarchy is shown in figure 1.2 and works as following [89]:

- IANA is in charge of all the address space, which is divided in blocks and allocate to each RIR (typically a /8 block for IPv4).
- Each RIR will divide its address block in smaller blocks and assign them directly to the so-called «*Local Internet Registries*» (LIRs), which are mainly *Internet Service Providers* (ISPs).

- Each LIR can then divide its address space for using by itself or for sub-allocating to other organizations.

In some cases there is also an intermediate authority between the RIRs and LIRs, called *National Internet Registry* (NIR). An example of NIR is *JPNIC*, the Internet Registry of Japan.

1.2.0.2 Legacy address space

When RIRs were created, some of the previous holders of address space accepted to move their address space under the control of the local RIR, however several organizations didn't. This created what today is called the «*legacy address space*», that is a group of address blocks not controlled by any RIR. RIRs keep track of these blocks and provide limited services (*reverse DNS*, *WHOIS*, ...), but they **have no authority on these addresses**. Some organizations (ex: Stanford university), «*returned*» IP addresses back to RIRs [66, 89], but they are a small percentage.

It is important to note how, normally an organization gets a «**delegation**» for using IP address space from a RIR (thus, the address space is not owned by the organization itself).

1.3 Types of routing incidents

The BGP protocol was designed back in 1989 without considering security, and because of its purpose and use, it was often considered a «*walled garden*»: a place where only operators can play, so there was no fear of security.

However, routing incidents due to configuration errors, also known as «*mis-configurations*», have been present on the Internet for over a decade [26] and are very common nowadays, as we will show in the following sections. The scale of these incidents can range from few network prefixes being unreachable to a slowdown of Internet communications in an entire geographic region.

Issues that we care about BGP can be classified mainly into *hijacks* and *route leaks* [50].

1.3.1 Hijacks

Because BGP have no mechanism to verify who is the current holder AS of a given IP prefix, a prefix hijacker can exploit this by originating a prefix allocated to some other AS. These hijacks could be the result of accidental «mis-configurations» or planned «attacks».

In other papers these *hijacks* are sometimes also called *«mis-originations»*, to underline the unintentional nature of the event. In the following we will however

always call them as «hijacks», since the final result appears the same, intentional or not.

Several types of hijacks are possible, depending on the type of BGP announcement used.

1.3.1.1 Prefix hijacks

In this kind of hijacks, a given AS a is announcing on BGP a given prefix p, which was legitimately allocated to a. An hijacker AS b originates then the exact same prefix p and propagate the announcement to other ASes on the Internet.

Other ASes may then select the announcement made by the hijacker b, according to local policies (for example: if the AS_PATH length is shorter than the original announcement of a).

The result will be that some of the traffic sent to addresses contained in prefix p may be sent to AS b instead of AS a.

An example of prefix hijack will be shown in section 1.5.1.

1.3.1.2 Sub-prefix hijacks

Like in the previous case, a given AS a is announcing prefix p, allocated to a. An hijacker AS b originates then another prefix r, which is a sub-prefix of the victim IP prefix.

With «sub-prefix» we mean that the prefix r is longer than p and it's addressing a part of the address space of p. For example `131.175.121.0/24` is a sub-prefix of `131.175.0.0/16`.

Other ASes receiving the announcement will be forced to prefer AS b for all the traffic for addresses belonging to sub-prefix r, since BGP always prefers routes over more-specific prefixes as main tie-breaker rule. For this reason this attack is even more dangerous since it may potentially result in hijacking all the traffic destined to the hijacked sub-prefix.

An example of sub-prefix hijack will be shown in section 1.5.2.

1.3.1.3 Path-shortening hijacks

Both prefix and sub-prefix hijacks could also be combined with a *path-shortening attack*, that is when the hijacker is announcing a fake AS path, where the origin AS of the announcement is the correct one. However, the AS path advertised by the hijacker is fake, and used only for two reasons:

- Hiding the hijacker by showing the correct AS as origin in the path.
- The advertised AS path might be shorter than the original one, thus forcing other ASes receiving the route to prefer it.

1.3.2 Route leaks

Route leaks are a different kind of routing incident, were no bogus route is announced. Instead, the announced route is legitimate and really used by the perpetrator of the leak, but it's *«leaked»* to too many neighbor AS.

On BGP, a route announcement is basically a request for traffic: it means telling to everyone: *«I can send traffic to this destination, please send me traffic and I will forward it»*.

For example, AS a might have a direct BGP peering with a big content provider AS b, as well as another direct connection with an upstream provider ISP c and a customer d, as we can see in figure 1.3.

a will get a BGP announcement from content provider b for prefix p , and will forward the announcement to the customer d, thus allowing the customer to transit on a for reaching b.

a have no interest in forwarding the route learned from b to the upstream ISP c, because a doesn't want to work as a transit provider between b and c, thus getting a lot of traffic to deliver.

However, it may happen that (due to *mis-configurations*), a could ***leak*** to c the route learned from b, attracting a lot of traffic for c to itself.

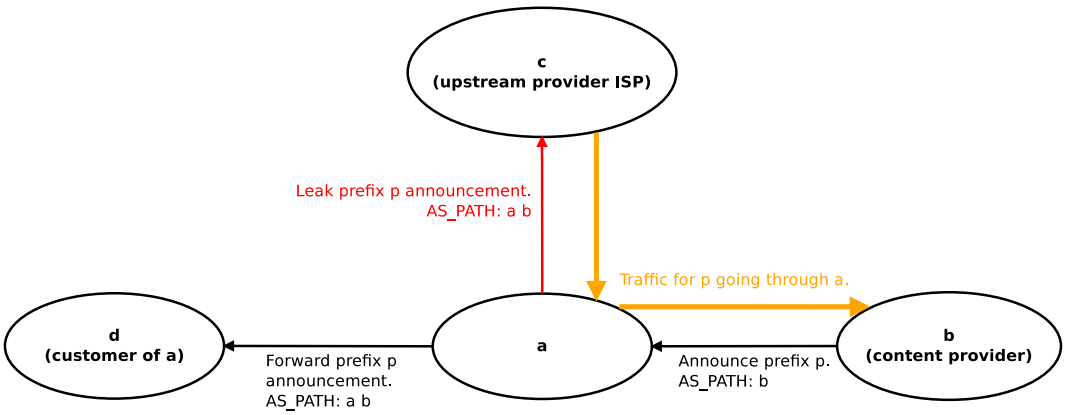

Figure 1.3: Illustration of the generic route leak described in section 1.3.2

For this reasons, contents providers peering directly with a lot of networks are more likely to be affected by route leaks, as discussed in [72].

1.3.2.1 Sub-prefix route leak

A route leak could become more effective when the perpetrator AS is *de-aggregating* the original prefix p into a sub-prefix r and leaking it to the wrong neighbor, since the more-specific priority will attract even more traffic, making this case similar to a ***sub-prefix hijack***.

1.3.3 «*Self de-aggregation*»

Another type of incident, that we may call «*self de-aggregation*», is when an AS starts to *de-aggregate*[1] it's own prefixes into a lot of more specifics ones (usually «leaked» from internal configurations), thus improperly increasing the global BGP table size and computation load on other routers.

1.4 Impact of routing incidents

The two main possible results of an *hijack* or *route leak* are a ***traffic blackhole*** or ***traffic interception*** [50]. High load and crashes of routers are also a possible *«side-effect»* of an incident.

1.4.1 Traffic blackhole

A **traffic blackhole** is when an hijacking or leaking AS is attracting traffic and dropping it, resulting in packets not able to reach the final destination and *end-users* noticing network outages.

1.4.2 Traffic interception

Traffic interception is when an hijacking or leaking AS is attracting traffic and later forwarding it to the legitimate AS. This could be the result of an attack as well as a mis-configuration. In the case of an attack, this could allow the attacker to silently intercept the traffic without both end users to notice.

We should also note that some routing incidents may initially result in *traffic interception*, and later degenerate into a *traffic blackhole* due to the huge amount of traffic flowing through some low-capacity links or hardware unable to process the amount of data incoming.

1.4.3 High load and crashes

Many routers running BGP are already heavily loaded due to the rapid growth of the Internet and the rate of BGP announcements incoming. A mis-configuration event may generate thousands of new prefixes, increasing dangerously the load on many BGP routers due to the high rate of BGP updates [73]. High loads may also lead to crash or *denial of service* of the router.

Another reason for routers crashing may also be related to the total amount of prefixes received by the router. On 12 August 2014 many routers on the Internet crashed after receiving a large amount of (legitimate) new prefixes *de-aggregated* by

[1] *de-aggregation* is the act of splitting a large prefix into several smaller ones covering the same address space. For example: `192.168.0.0/16` can be *de-aggregated* into two /17 prefixes: `192.168.0.0/17` and `192.168.128.0/17`, or into four /18 prefixes, etc...

a large provider [40, 67]. The *de-aggregation* event leaded many router to reach the limit of available records for storing prefixes in the internal $TCAM$[2] memory.

1.5 Examples of routing incidents

1.5.1 Indosat incident (2014)

On 2 April 2014, Indosat (`AS4761`), one of Indonesia's biggest ISPs which was normally originating about 300 prefixes, began to originate 417038 new prefixes normally announced by other ASes [93]. The mis-origination lasted several hours and all the prefixes originated were of the same length of the original ones, but announced with `AS4761` as origin. Only a small percentage of prefixes were propagated far from Indosat's AS, keeping the incident relatively isolated.

Given the type of announcements it's very likely to be the result of an operational issue or mis-configuration related to a re-propagation of internal routes as explained in section 1.6.1.

This could be considered a typical example of **prefix hijacking** or «mis-origination», where an AS mis-originated prefixes of other ASes by mistake, resulting in attracting some traffic of other ASes.

These attacks are quite common, and happen every month from different ASes around the Internet nowadays. Sometimes the scale of this problems it's limited thanks to filters of upstream providers, while other times the impact might be larger, depending on the specific case of topology and policy of providers.

1.5.2 Pakistan YouTube incident (2008)

1.5.2.1 The incident

One of the most famous Internet routing incidents is probably when *Pakistan Telecom* took *YouTube* offline [34].

On 24th February 2008, after an anti-Islamic video was published on YouTube, Pakistani authorities demanded that YouTube must be censored within Pakistan [52].

At the time, YouTube (`AS36561`) was announcing `208.65.153.0/22` as address space.

In order to apply the censorship, Pakistan Telecom (`AS17557`) decided to announce the sub-prefix `208.65.153.0/24` to its customer ASes in Pakistan (*Aga Khan University*, *Lahore Stock Exchange*, *Allied Bank Pakistan*, ...).

[2] *Ternary Content-Addressable Memory* (TCAM). Content-addressable memory allows software to provide content and to recall the address of the memory. Such operations are much faster using binary CAM than RAM. Ternary CAM allows an additional field to be used, such as a mask.

By announcing a more specific prefix, Pakistan Telecom ensured that all the traffic going to the `208.65.153.0/24` subnet would have been intercepted and dropped.

However, due to a mistake in the configuration, Pakistan Telecom also sent this announcement to PCCW (`AS3491`), a large ISP providing connectivity to Pakistan Telecom and several other ASes worldwide. PCCW received, accepted the announcement and made it the most preferred route for addresses in that range (because more-specific than the original /22 of YouTube). PCCW also re-announced the prefix to other ASes around the world, who also forwarded elsewhere.

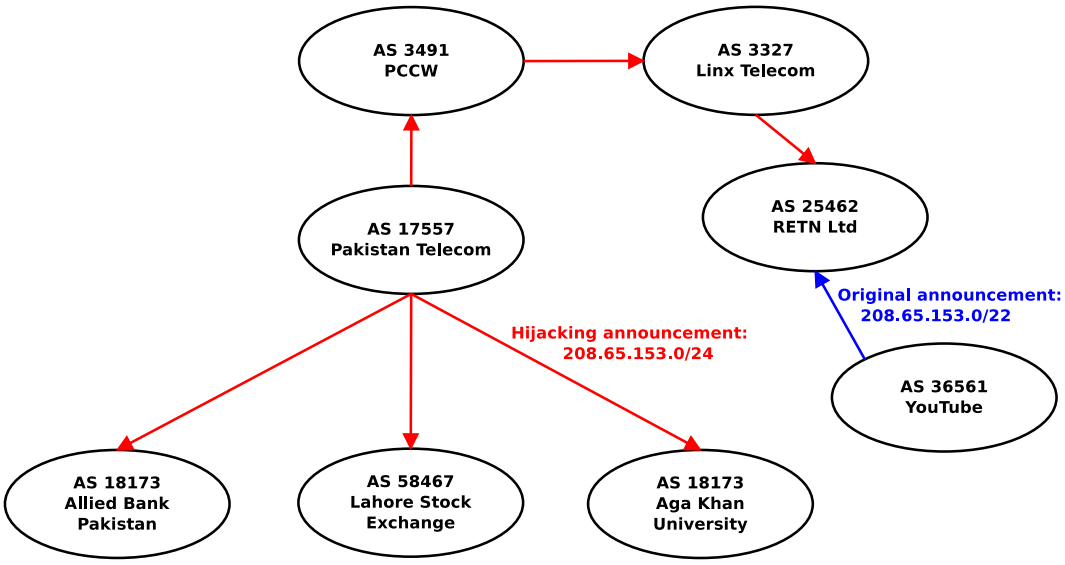

Figure 1.4: Main Autonomous Systems involved in the Pakistani YouTube incident

Time passes and the hijacked announcements gets gradually propagated in several ASes.

1.5.2.2 Mitigating the hijack

After 80 minutes from the hijack, YouTube engineers started to announce the same `/24` prefix, re-gaining some of the traffic.

Some ASes received both the Pakistani and the YouTube new `/24`, but the Pakistani one was still the most preferred, for example because in some cases the `AS_PATH` length was shorter for the Pakistani announcement rather than the YouTube one.

For this reason, 90 minutes after the hijack, YouTube engineers also started announcing two `/25` prefixes, covering the same address space of the `/24` prefix (`208.65.153.0/25` and `208.65.153.128/25`), in order to gain precedence also where `AS_PATH` length was choosing the wrong announcement.

The problem of using `/25` prefixes is that several ASes may reject announcements with prefixes longer than 24, in order to avoid filling the routing table (which has a limited size) with too-specific announcements.

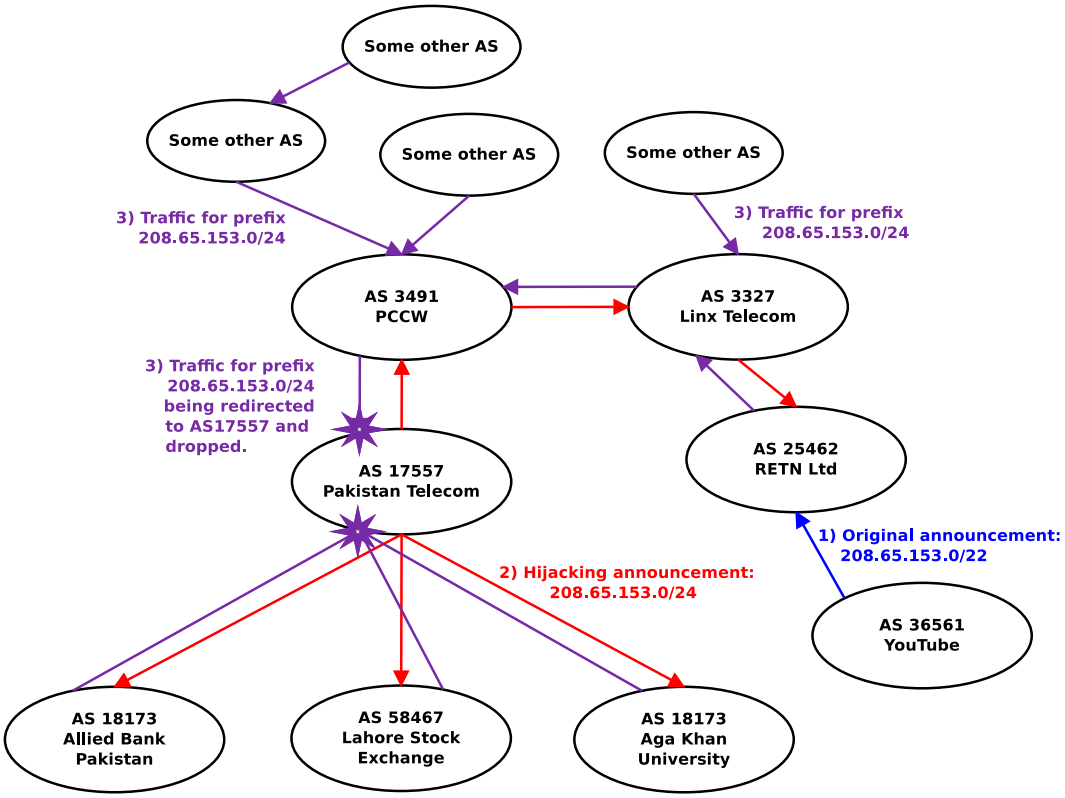

Figure 1.5: Traffic flow few minutes after the hijacking of the Pakistani YouTube incident

Two hours from the hijack start, luckily for the YouTube engineers, PCCW was contacted and manually did a withdrawn of the problematic prefix finally solving the problem.

1.5.2.3 Conclusions

It has been estimated [34] that about two-thirds of the Internet was sending traffic to Pakistan Telecom instead of YouTube for about 80 minutes for most ASes and nearly 2 hours for few other ASes.

This incident is an example of accidental **sub-prefix hijacking**, because a more specific prefix has been used and the source AS of the announcement was not the original one. Moreover, this case was particularly problematic due to the prefix length of 24 that made harder than usual to re-gain traffic back to the origin. Due to the high popularity of the website, the incident also slightly raised the awareness on routing security problems in the Internet community.

Looking back at the incident we may say that, other than Pakistan Telecom, PCCW should be also considered responsible for the incident, because it did not filter an announcement from a direct customer (Pakistan Telecom). Because BGP relies on a *«transitive-trust model»*, validation between customer and provider is very

important. A provider AS should agree with customers about advertised prefixes, in order to setup filters on incoming and outgoing announcements and prevent this kind of incidents.

1.5.3 DEFCON 16 - «Stealing The Internet» (2008)

1.5.3.1 The attack

At DEFCON conference, in 2008, Alex Pilosov and Tony Kapela demonstrated an example of «*man in the middle*» (MITM) attack to the BGP protocol, involving a **sub-prefix hijacking** and **as-path mangling**, in order to intercept all the Internet traffic going to the DEFCON conference's network [81], while hiding the attack.

An Internet Service Provider (located in Las Vegas, `AS22773`) reserved a `/22` IPv4 prefix for the conference network connectivity (`24.120.56.0/22`). In figure 1.6 there is an illustration of the topology before the attack.

In the demonstration, the speakers use a router they control in another AS (located in New York, `AS26627`) to announce a more-specific `/24` prefix (`24.120.56.0/24`) in order to steal the traffic directed to the conference network.

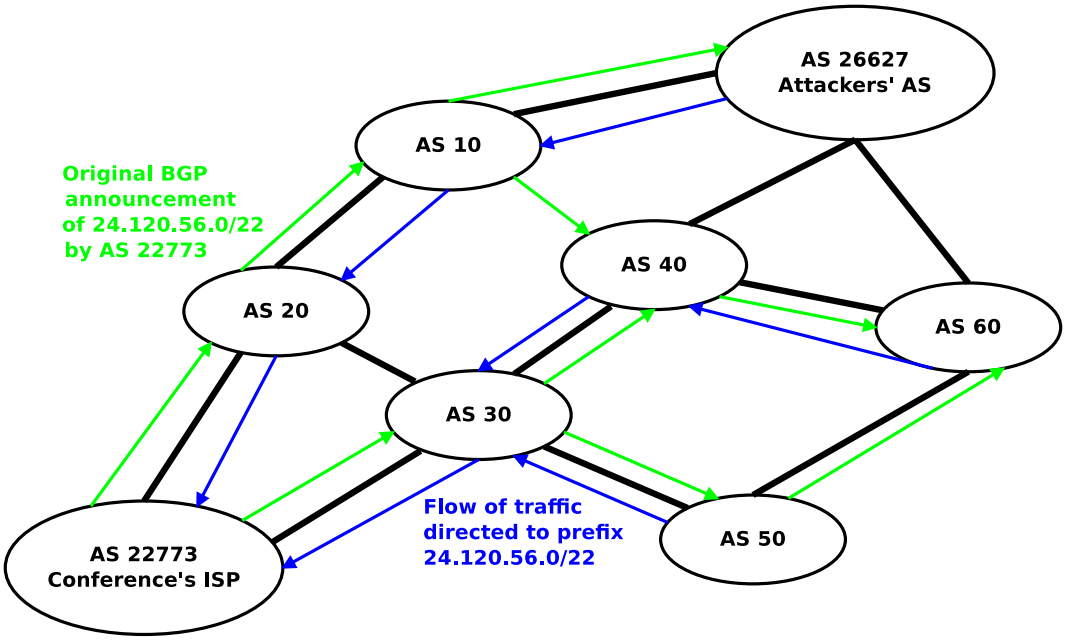

Figure 1.6: Topology of the ASes, announcement and traffic before the prefix hijacking. AS numbers 10, 20, etc... are falsified for illustration purposes.

1.5.3.2 Bypassing Internet Routing Registries

As we will explain in section 2.3, it's possible to use a public «*Internet Routing Registry*» (IRR) to lookup which should be the AS number allowed to announce a given prefix. The information contained in the IRRs is commonly used by other

network operators to setup filters on incoming BGP announcements in order to prevent mis-originations. However, Pilosov and Kapela also managed to create a valid entry in one of the routing registries (ALTDB IRR [2]), listing their AS (`AS26627`) as allowed for announcing the more-specific prefix.

1.5.3.3 Sending traffic back to the hijacked network

The speakers didn't only wanted to get all the traffic for the DEFCON's network, but they also wanted to send the traffic back to the conference's network after intercepting it, in order to hide the attack and intercepting data of ongoing connections.

In figure 1.7 we can see an illustration of what would be the result of a «normal» sub-prefix hijack, by simply announcing the `/24` prefix to all peers of the attacker AS. Because the most specific always win, and there is a valid IRR entry, the hijack would be successful, but there would be no way for routing packets back from the attacker to the ISP.

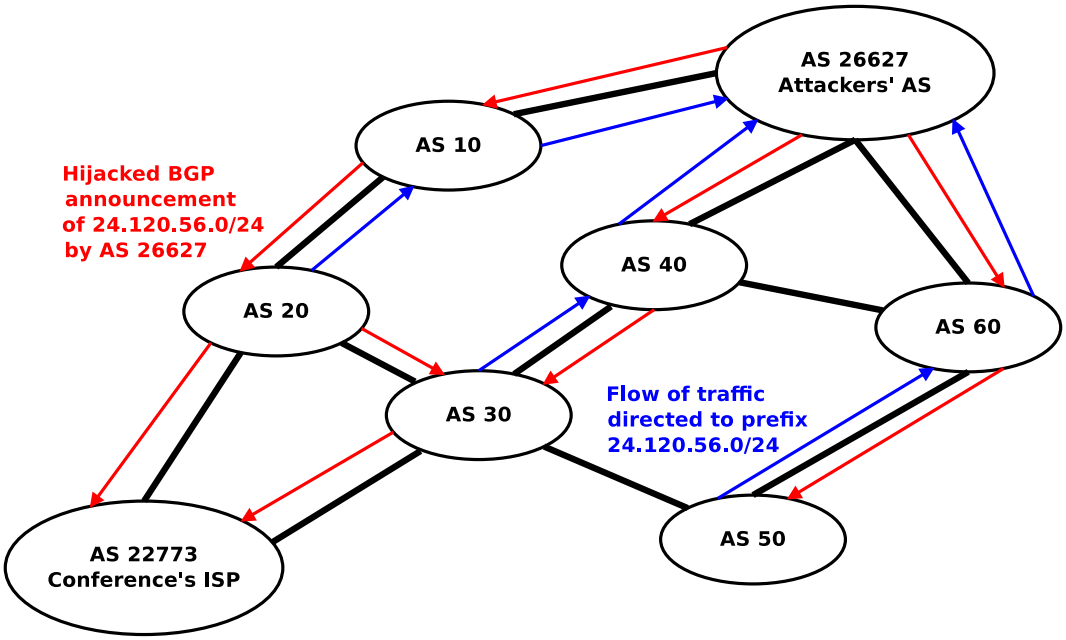

Figure 1.7: AS Topology in case of a «normal» sub-prefix hijack

One of the most important rules of BGP is: an AS that receives an announcement containing its own AS number is supposed to drop the announcement, since it may represent a loop of a previously sent announcement coming back on a different path, and accepting it may create a routing loop (section 9.1.2. of [87]).

The speakers decided to exploit this property of BGP and they modified the BGP `AS_PATH` by «*prepending*» the AS number of all the AS on the path from their AS to the ISP's AS. In this way they managed to send the hijacked announcement

attracting traffic for the conference network to every AS, except those on the path to the ISP. Figure 1.8 shows the result of such announcement.

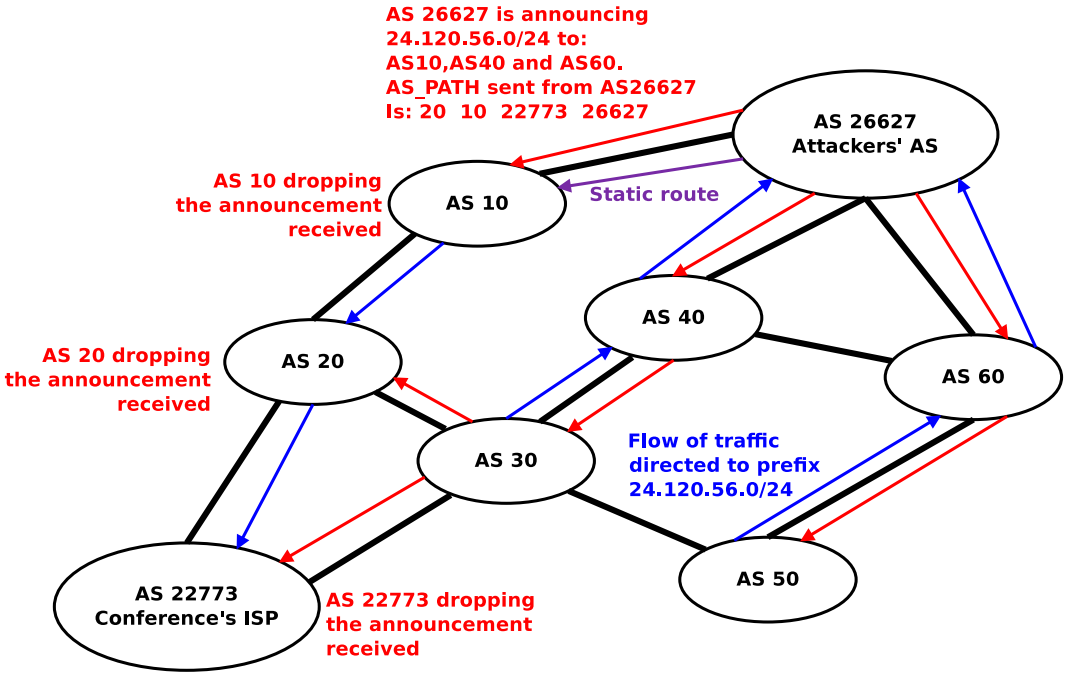

Figure 1.8: AS Topology after the complete hijack

After this, they configured a static route for the hijacked `/24` in their AS, using the first AS on the path back to the conference's ISP as next-hop of the route. In this way they managed to get the hijacked traffic to be forwarded back to the correct destination.

1.5.3.4 Hiding the attack

Someone running the *traceroute* tool from a machine on the Internet, using one of the IP addresses of the hijacked `/24`, could however very trivially notice the attack, since *traceroute* will show all the hops up to the attacker's AS, and then down to the DEFCON's network.

Every host that forwards an IP packet will reduce its TTL value by one. Once the value becomes 0, the host currently holding the packet will send an ICMP time exceeded message back to the sender. The *traceroute* program works by sending ICMP or UDP packets with an increasing TTL value in the IP header, in order to get replies back from all the hosts/routers along the path to the destination. This also means that a router may be able to hide some hosts after itself just by increasing the TTL value (and by recomputing the IP header checksum).

For this reason, the speakers also decided to mangle IP packets transiting through their AS, by increasing the *Time To Live* (TTL) value of the IP packets, in order

to hide the hops after their AS, making harder for an external user to notice the hijack.

1.5.3.5 Conclusions

This attack still have few limitations:

- The ASes on the path from the attacker to the hijacked AS will still send traffic to the correct destination, so it's not possible to intercept traffic directed to addresses in those ASes
- Only the **incoming** traffic can be intercepted: the traffic going from the conference's network to the Internet it's still following normal paths. However, as the speakers shows, it is possible to intercept sensible data as well as poisoning DNS replies and perform many other common MITM attacks.
- The attack could have been detected, if the hijacked AS had monitored its prefixes using publicly-collected BGP data (using data sources or services that we will discuss in section 2.1).
- The attack could still have been detected from the partial *traceroute* output or by noticing the sudden *Round-Trip-Time* change, with a detailed analysis.

This attack type is interesting, because it shows how easily one could not only misoriginate a prefix, but also fake the `AS_PATH` and bypassing the common «protection» used by network operators (filtering based on IRR data).

1.5.4 Spammers using unallocated IP space («*IP squatting*»)

1.5.4.1 The attack

The use of email to send unsolicited messages (like advertising), also known as «***spam***» or «***spamming***», has always been a problem on the Internet since the popularity of email spread around the world.

Avoid getting spam is quite a complicated problem to solve, however one of the most effective solutions nowadays is the use of an *«IP reputation list»,* that is a list containing IP prefixes know for sending spam. System administrators can then download or lookup these lists in order to setup a check of the source IP address of an incoming email, and eventually drop it.

Some spammers did however found a way around IP reputation lists. Firstly they look for some IP space which is not allocated yet to any AS, or existing IP space allocated to some AS which had not been announcing it for a long time. They then select a prefix in this IP space and they announce it from some AS that they control (or that has been compromised by them).

This technique is also known as «*IP squatting*» and has been observed and studied in several research works [84].

1.5.4.2 AS Telelatina case of IP squatting

An example of IP squatting is what has been observed by engineers running a BGP monitoring service, *BGPMon* [7], in 2014 [96].

`AS15078` Telelatina has not been announcing any prefix since January 2011, but in 2014 it suddenly started to announce around 1000 new prefixes in few weeks of time. Figure 1.9 shows a clear pattern where `AS15078` is announcing 8 new prefixes for few hours, probably while using them for sending spam emails. Few hours later the AS is withdrawing the prefixes, then announcing other 8 new prefixes and repeating the cycle.

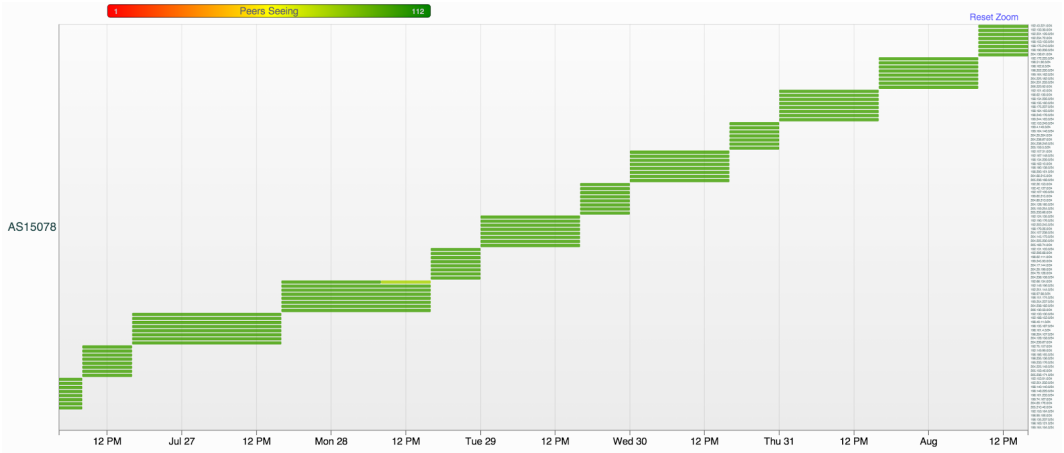

Figure 1.9: Pattern in the announcement of IP prefixes by time. Seen with RIPE NCC ATLAS tool [21], source: [96].

1.5.4.3 Internet Routing Registry bypass

BGPMon engineers analyzed who was providing transit to the Telelatina AS and they found out that the only provider was Hibernia (`AS5580`) [7], a provider known for applying filtering on incoming BGP announcements.

What has been found out later is that the spammers controlling Telelatina's AS, managed to create valid route objects in the RADB IRR[3] [19], since RADB IRR allows to register arbitrary route objects without any authorization. Hibernia's AS was then using RADB among with several other IRR databases in order to automatically build filters on incoming BGP announcements, thus allowing the spammers to propagate their BGP announcements.

[3]Note: see section 2.3 for IRR details.

1.5.4.4 Conclusions

This is yet another example of a **prefix hijack**, slightly different to the others because the announced IP space is not being announced by anyone else, so that it might be harder to notice.

1.5.5 Moratel route leak (2012)

1.5.5.1 The incident

Moratel (`AS23947`), is a local ISP in Indonesia handling small amounts of traffic for few customers. Moratel gets its connectivity from PCCW (`AS3491`), an international ISP.

On 6 November 2012, Moratel started «*leaking*» a route [80] to prefix `8.8.8.0/24` of Google (`AS15169`), a well-known prefix containing the Google's public DNS address [9]. Moratel was sending to PCCW a BGP announcement with Google as origin AS and Moratel as transit for reaching Google (`8.8.8.0/24`, AS_PATH: `23947 15169`).

Because of internal routing policies, PCCW decided to prefer the route for `8.8.8.0/24` received from Moratel instead of the one received from Google [50]. A lot of providers setup policies in order to prefer routes announced by customers when possible, since the more traffic is sent through the customer, the more revenue is possible for the provider. In figure 1.10 there is an illustration of the incident topology.

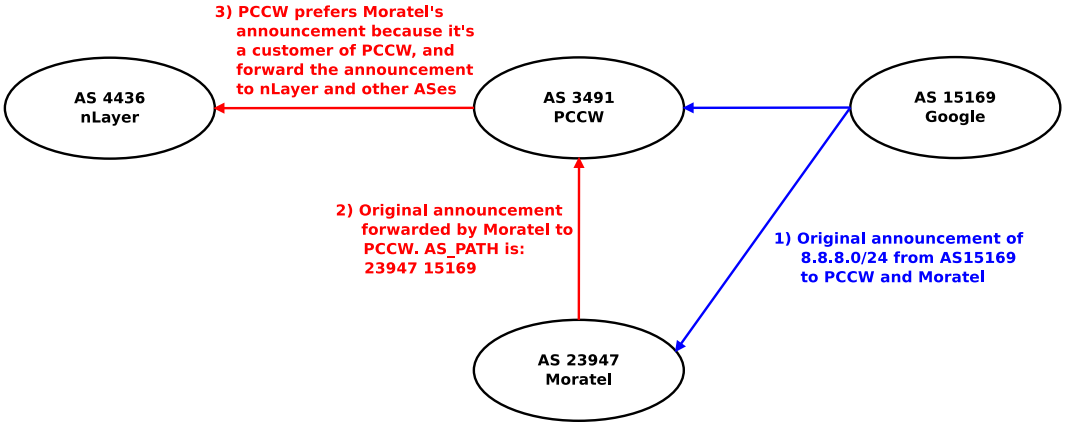

Figure 1.10: AS Topology of the Moratel's `8.8.8.0/24` route leak.

The route leak didn't propagated much far from Hong Kong (where PCCW is the *incumbent* provider), however about 3-5% of the Internet was affected [80].

As 2012, the Google's public DNS address `8.8.8.8` was handling more than 70 billion requests a day [27]. For this reason, as soon as the route leak propagated, a large amount of traffic was directed to Moratel, thus overloading the capacity of

transit links, bringing the Moratel's network down, as show in figure 1.11.

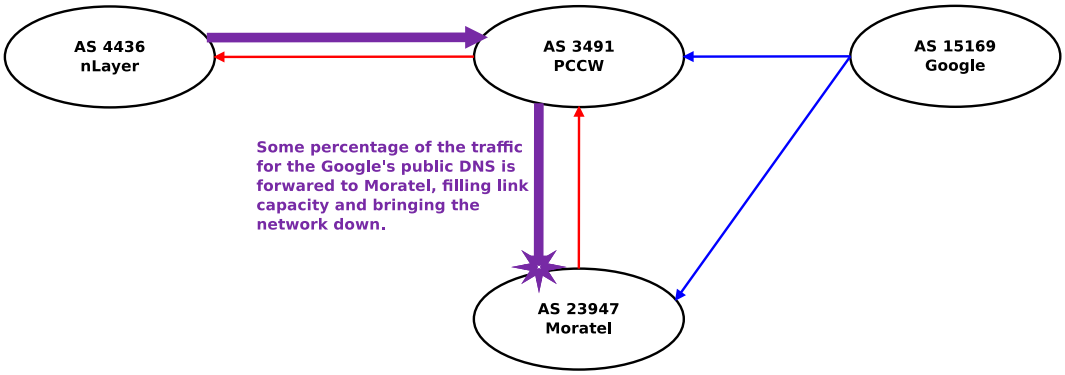

Figure 1.11: AS Topology after the route leak

About 30 minutes later the problem was solved after contacting the Moratel's network operators, who withdrawn the route [80].

1.5.5.2 Conclusions

This is an example of a ***route leak***, that managed to be spread due to a specific situation and upstream provider's policy. The reason of the route leak have been confirmed to be a mis-configuration «*caused by an unexpected hardware failure*» [80].

1.5.6 Telekom Malaysia route leak (2015)

1.5.6.1 The incident

On 12 June 2015, Telekom Malaysia (`AS4788`), started leaking 179,000 routes to Level3 (`AS3549` – formerly known as Global Crossing) whom in turn accepted these and propagated them to their peers and customers, causing significant packet loss and Internet slow down in all parts of the world [71,94]. The type of incident is very similar to what we already described in section 1.5.5. The main difference with the Moratel's incident is that, in this case the upstream provider was a *tier-1* ISP with hundreds of customers around the world.

The networks more affected from this incidents were those peering directly with Telekom Malaysia like: Amazon, Google and Facebook. Figure 1.12 shows a part of the route leak topology.

1.5.6.2 Conclusions

This is yet another case of ***route leak***, in this case involving a lot of prefixes rather than a single one. The incident caused a lot of packet loss due to packets being forced to transit through Telekom Malaysia and lasted for about 2 hours [94].

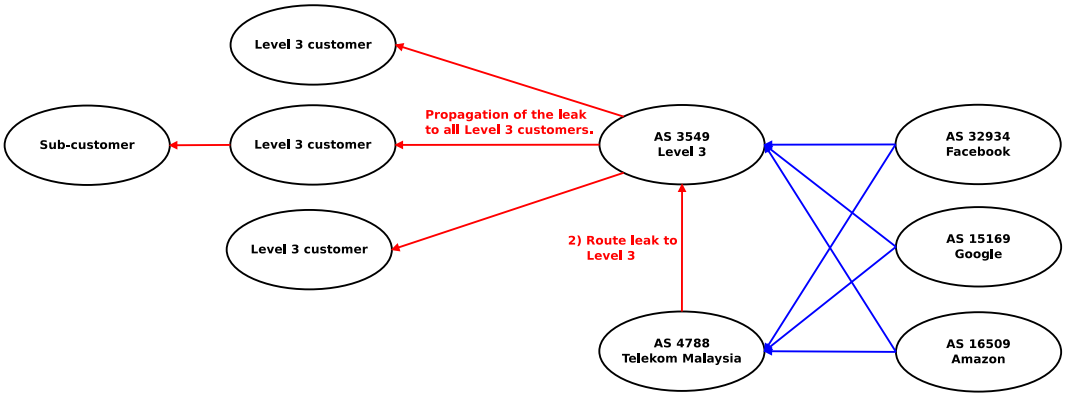

Figure 1.12: AS Topology of the Telekom Malaysia's route leak

1.5.7 BGP Optimizer causing fake routes (2015)

Another cause of incidents seems to be due to BGP optimizers, which are appliances used inside the network of an AS for *traffic engineering* purposes. A BGP optimizer usually works by breaking an aggregate block of prefixes into two (or more) prefix sets, in order to select paths inside a network and optimize the internal network traffic.

1.5.7.1 The incident

On 27 March 2015, more than 7000 prefixes (belonging to roughly 280 ASes) were **sub-prefix hijacked** by someone who was announcing more specifics [97].

The unusual fact about this attack is that the more specific prefix was apparently announced by the correct origin AS number, thus looking as legitimate.

At the time, Enzu Inc. (`AS18978`) had installed a *BGP optimizer* in its network, which was breaking some prefixes into more specifics for internal *traffic engineering* purposes. These more specifics leaked from the internal network to the upstream AS: *Los Angeles Internet Exchange* (`AS40633`). For this reason the more specific announcement, similarly to a route leak, had the correct origin AS, but an AS path attracting all the traffic to pass through `AS18978`.

1.5.7.2 Amazon AS example

For example, Amazon Inc. (`AS14618`) was usually announcing the prefix `23.20.0.0/15`.

During the attack, the prefix `23.20.112.0/20` appeared with the following `AS_PATH`: `4608 24130 7545 6939 40633 18978 3257 14618`. However, Amazon Inc. never announced a longer prefix than the `/15` one.

As shown in figure 1.13, the `/20` sub-prefix was leaked to the Los Angeles Internet Exchange, which accepted the announcement and propagated the prefix, forcing all the Internet to select the leaked route (because more specific than the original one).

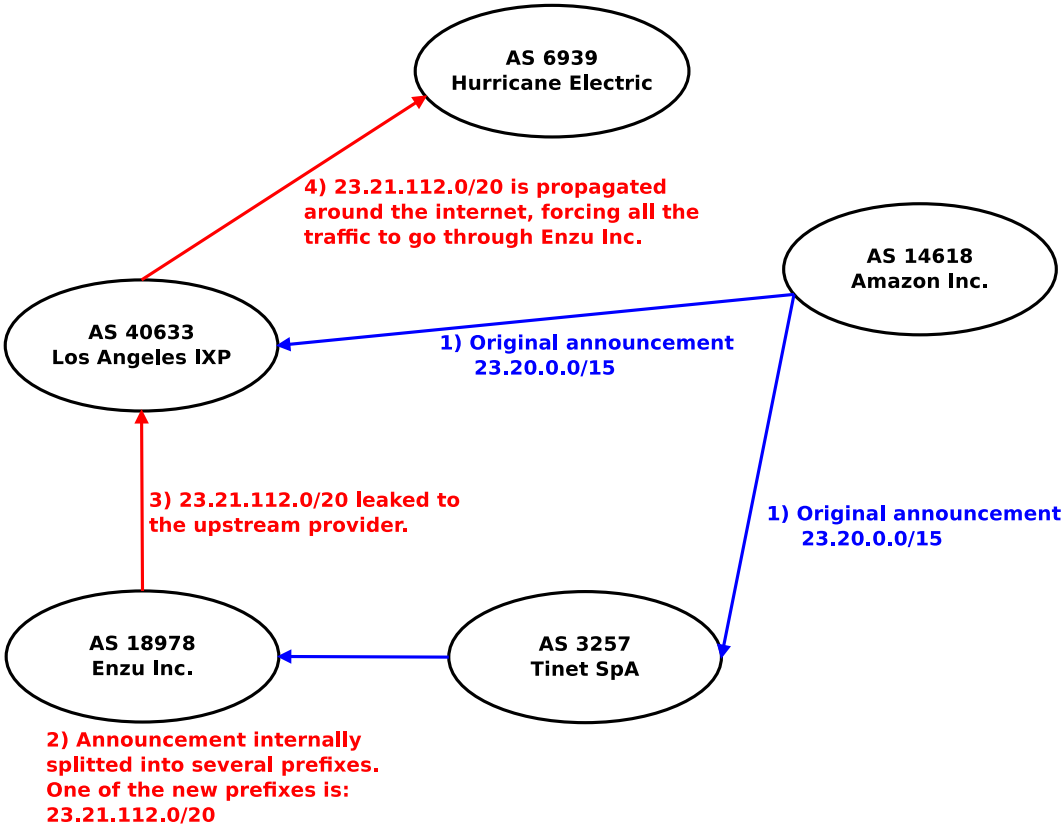

Figure 1.13: AS Topology of the Enzu Inc. route leak for the Amazon's prefix. Note: the topology has been slightly modified from the original, for illustration purposes.

1.5.7.3 Conclusions

This incident is classified as a ***sub-prefix route leak*** (since a real path towards the correct AS was existing). This was more dangerous than a «normal» *route leak* because the more specific prefix propagated and forced all the traffic to flow through Enzu Inc., due to the most-specific preference of BGP.

In this case the upstream provider (the Los Angeles Internet Exchange), could not be accused of accepting the route, since it couldn't be able to know that Enzu Inc. can or can't provide transit to other ASes.

1.5.8 BGP Hijacking for Cryptocurrency Profit (2014)

A research team part of *Dell SecureWorks*, discovered that between February and May 2014, an unknown entity repeatedly hijacked traffic destined for networks belonging to large hosting companies [69, 95]. The goal of the malicious attack was diverting the traffic from *bitcoin miners* to relay the result of their computation and divert the monetary benefit of their work.

It has been estimated that the attackers earned about $83,000 in more than four months [69].

1.5.8.1 Bitcoin mining

Explaining details about how the bitcoin network and mining process work is outside the scope of this work; however we can briefly summarize as following:

In the bitcoin network, «*mining*» is the act of solving a long computational problem, which is rewarded with the creation of some bitcoin value.

In order to get the smallest reward amount, the computation power required is becoming more and more high. For this reason several users shard together in a «*mining pool*». The purpose of a mining pool is to share computation power to achieve the goal of mining «*a block*». The reward is then split among miners.

In a mining pool, all miners contact a central «*pool server*», which is able to control all actions of miners, as well as getting result of computations. The communication between *miners* and the *pool server* is done using the *Stratum protocol* [69], which is a JSON-based TCP-connection.

In this case, BGP was exploited by the attacker in order to hijack TCP connections going to a *pool server* and sending commands to *miners* in order to redirect them to another pool server controlled by the attacker, thus stealing result of computations.

1.5.8.2 BGP

There aren't many public details about this attack (except for the list of prefixes that were affected [69]), however it is known that the attack was a ***sub-prefix hijack*** where the `AS_PATH` has been spoofed similarly to the 2008 DEFCON attack (section 1.5.3). The main difference to the DEFCON's attack is that here the attacker also managed to hide the origin AS of the attacks by «*AS path prepending*» of fake informations.

For example the attacker was appearing to forward a more-specific route announcement with *Amazon Inc.* as origin AS, even though there was no such route coming from *Amazon Inc.*

These announcements were then forwarded to the *Toronto Internet Exchange*, where other ASes received and accepted the announcement. Many ASes do not use prefix filters on announcements coming from Internet Exchanges, but only *Max-Prefix* filters in order to prevent accepting announcement from network leaking or mis-originating a lot of prefixes [95]. For this reason the announcements were easily propagated.

1.5.8.3 Conclusions

This attack is interesting not only because the attackers managed to silently hijack TCP connections over a period of months, but they also managed to spoof the origin AS of the announcement. This will be important when we will discuss about the current solutions for origin-validation and their limits.

1.5.9 Incidents summary

Table 1.1: Summary of all incidents seen in section 1.5

Incident	Original prefix	Sub-prefix	Correct origin AS	AS_PATH mangled	Result
Indosat (1.5.1)	Yes	No	No	No	Blackhole
Pakistan YouTube (1.5.2)	No	Yes	No	No	Blackhole
DEFCON16 (1.5.3)	No	Yes	No	Yes	Interception
IP squatting (1.5.4)	No	Yes	No	No	Interception
Moratel (1.5.5)	Yes	No	Yes	No	Interception
Telecom Malaysia (1.5.6)	Yes	No	Yes	No	Interception
BGP Optimizer (1.5.7)	No	Yes	Yes	No	Interception
Canadian bitcoin (1.5.8)	No	Yes	Yes	Yes	Interception

1.6 Common reasons for accidental routing incidents

In the following we report results of a survey conducted in [73].

Most of the prefix hijackings are often caused by errors in the configuration of *route redistribution* internally to an AS, but bugs and typing errors are also common.

1.6.1 IGP Redistribution

The most common cause of hijackings and route leaks is probably IGP redistribution [73].

In short, redistribution allows a network operator to specify which routes learned from interior routing protocols (like OSPF or IS-IS), should be advertised to external BGP peers. There are a lot of ways to achieve this goal, here there are two examples (taken from [73], based on *Cisco IOS*), which are error-prone:

- `redistribute igrp 100 route-map igrp2bgp` : This tells the router advertise everything in IGP tables that matches the route map igrp2bgp. If an operator forget to specify the route-map part of the command or gets the route-map itself wrong, all the prefixes in the IGP table would be leaked to BGP. If the IGP table contains all Internet prefixes propagated into IGP from another BGP router, this will result in leaking all known Internet prefixes to the BGP peers.

- `aggregate-address 192.168.0.0 255.255.0.0 summary-only` : This tells the router not to advertise any subsets of the prefix. However, if *«summary-only»* is forgotten, all the more-specific prefixes in the routing table could be

advertised. If redistribution is not done correctly, it can lead to a large number of faulty prefix advertisements.

1.6.2 Wrong BGP community

A BGP community is basically a label attached to a BGP announcement, which could be read by some upstream AS and used for taking some decisions about how to treat the announcement.

A common use of a BGP community is to communicate to the upstream provider to receive and use the announced route, but avoid forwarding it to any other AS (also known as «*no-advertise* community» [38]).

For this reason, an incorrect or missing BGP community may get some route announcements to be propagated beyond of where they were intended.

1.6.3 Reliance on upstream filtering

In some cases, some ASes were found to announce some prefixes for various reasons, on the assumption that the upstream provider would filter them, but the provider was not doing so [73].

1.6.4 Old configuration

In few cases it was found that some operators changed a router configuration, but later forgot to commit changes to a stable storage. After a reboot, the router changed configuration and leaked prefixes. Because of how the common router configuration interfaces works, it's often not easy to notice that the current configuration is not being saved.

1.6.5 Typing error

A lot of BGP routers are still manually configured on the *command-line-interface* nowadays, so it is often possible that a typing error in a BGP peering configuration may lead in announcing the wrong prefix.

Chapter 2

Available defenses

2.1 Monitoring and mitigating hijacks

Although it might not be considered a «solution» to the problem of mis-configurations and attacks, one of the most common practices used by network operators is monitoring the global BGP traffic in order to spot anomalies regarding their own prefixes. This could be done manually, automatically with some scripts or delegated to some external monitoring service.

2.1.1 Data sources

2.1.1.1 BGP collectors

There are two main public projects providing historical BGP data: **Route Views** [25], which is maintained by the *University of Oregon*, and the **Routing Information Service** (RIS) [22], which is maintained by the European RIR: *RIPE NCC*.

These services have several routers running BGP in several big ASes (ISPs, IXPs) around the world, recording incoming BGP announcements from all peers and saving them in a public archive accessible via FTP protocol.

In each router running BGP, there is a table called the «Routing Information Table», or **RIB**, which contains the list of all *routes* received from all peers.

Every few hours, a complete dump of the RIB table is saved and uploaded a public archive.

These archives may also contain «updates» files for every 5-10 minutes interval. These are a raw dump of all the BGP announcements as they were received by the peers in a period of few minutes.

In some cases it's also possible to have a live data stream of the BGP announcements collected by each router, without any publication lag [6].

2.1.1.2 Looking glass servers

Some networks, commonly ISPs and IXPs, are also providing the so called «*looking glass servers*», which are public-accessible services for running network diagnostic tests (like ping, traceroute, ...) or for the purpose of viewing routing information as they are known to the looking glass provider.

These services can be provided with custom web interface or via «read-only» console access (telnet or SSH protocol) to some server or router in the provider's network.

2.1.2 Getting alerts

When there is a mis-configuration or an attack, a network operator is interested to be alerted if his network is affected and if any customer may have problems in reaching the announced prefixes.

The operator may setup scripts for pulling data from public available archives of BGP data and verify that the IP prefixes are being received by the BGP collectors correctly, that is: there are no more specifics or announcements from different origin ASes.

As alternative, several companies are specialized in providing BGP monitoring and alerting services for other networks. Some examples of this are: *BGPMon* [7], *Dyn* [8] or *ThousandEyes* [4].

These services often make use of some heuristics such as looking for peaks in generation of more specific prefixes in the global announcements, in order to spot some other incidents that may be undetected. This was for example done in the incident of section 1.5.4 and was also used in [73] for detecting mis-configuration events.

2.1.2.1 Issues with BGP monitoring

A minor problem is present with this approach too: because of how BGP forwards announcements and because the number of BGP collectors is quite limited, there is always the possibility to miss the detection of an hijack or route leak event. Not every BGP announcement is forwarded everywhere.

For example, a customer of an ISP might be hijacking a prefix, propagating the announcement to all other customers of the ISP too; but the announcement might be filtered by the upstream transit provider of the ISP. If no BGP route collector is peering with the ISP or customer ASes, this event might be undetected but still causing geographically-limited issues.

2.1.3 Fighting hijacks by *de-aggreating* prefixes

A common temporary solution to an hijack or route leak can be the announcement of more specific prefixes (prefix *de-aggregation*) in order to fight the attack by gaining precedence in the routing tables. The Pakistani YouTube incident that we saw in section 1.5.2 is a clear example of this solution.

This solution is so common that apparently some network operators are on purpose announcing short-length prefixes on BGP, in order to be ready to *de-aggregate* them in more specifics in case of an hijack.

An issue with this «solution» is that many operators are filtering BGP announcements with length longer than 24, in order to avoid overloading the routing table size with «*BGP churn*». For this reason, if the hijacking is using a /24 prefix, this method may not work for worldwide propagation. The Pakistani YouTube incident is again a good example of this problem too.

2.1.4 Contacting the network causing problems or upstream

As we have seen in chapter 1, most of the routing incidents are mis-configurations. For this reason, contacting the network operators of the perpetrating AS can be one of the most effective solutions to the problem. This is in fact what happened for example in 1.5.5.

By querying the RIR's registries with the *WHOIS* protocol [43] it's usually possible to find email addresses and phone numbers for contacting the network operators of a given AS.

When the perpetrating AS is not reachable or when the incident is likely to be malicious, an alternative is to find and contact the upstream connectivity providers of the perpetrating AS. An example of this are sections 1.5.2 and 1.5.4.

Part of being a network engineer is having relationships with other network engineers around the world. For this reason there are some *mailing lists* where is possible to ask for help for other network operators in contacting the perpetrating AS or mitigating the incident. The «*North American Network Operators Group*» (*NANOG*), have a famous *mailing list [17]* which is a good example of this practice.

2.2 BGP prefix filtering

The most effective solution in preventing most of the hijacks and route leaks is prefix filtering, which is a *whitelisting* technique used to filter out bogus BGP announcements coming from a BGP peer. In [51] it was found that if every Internet provider with at least 25 customer ASes were to deploy prefix filters properly, this would prevent a large fraction of the Internet's ASes from perpetrating route leaks or hijacks.

With prefix filtering, a provider can filter all announcements coming from a customer peering with it, in order to accept only announcements containing known prefixes and/or with a given AS path structure. This can very effectively prevent all hijacks and route leaks by blocking them before they get forwarded to any AS.

«Prefix filtering» is a general idea, which could be applied in several ways. By considering some of the most commonly used router OSes: *Cisco IOS* and *GNU Quagga*, the main types of filtering are: **prefix filter lists**, **access lists**, **route maps** and **maximum-prefix filter**.

2.2.1 Prefix filter list

For each peer connected to a given router, it is possible to define a *prefix-list* for filtering incoming (or outgoing) BGP announcements.

For each BGP announcement received, each entry of the prefix list is analyzed. Each entry of the prefix list contains an IP prefix to match, a maximum/minimum length of the prefix match, and an action to perform: ***accept* or *drop*** the announcement.

Once an entry is matched, the action is performed and no other entries are processed. If no entries match the announced prefix, then the default action is performed.

One important characteristic of a prefix list, is that it is matching the IP prefix in the announcement, regardless on the origin or AS path.

A prefix list can be used to perform some simple filtering tasks:

- For a provider, to filter BGP announcements of a «stub-AS» customer, known to always announce the same prefixes.
- Filtering well-known IP prefixes that should never be announced on BGP (also known as «BGP bogons»), like the private IPv4 addresses [86].
- For filtering all prefixes with length longer than N, in order to avoid receiving «BGP churn». This can be done for example by creating a rule for allowing prefix `0.0.0.0/0` , maximum length N and drop everything else.

Prefix filter lists are commonly configured using the «`ip prefix-list`» command.

2.2.2 Access lists

For each peer, it is also possible to define one or more *access lists*. An access list can be used to match the structure of the AS path received in the BGP announcement, using a *regular expression* syntax.

For example, «`ip as-path access-list myACL permit ^200_300`» will add to the access list «myACL», a rule for accepting BGP announcements containing an AS path starting with `AS200` followed by `AS300`.

2.2.3 Route map

The *route map* is the most flexible type of filtering: with a *route map*, an AS can specify a filter matching a specific announcement (specific prefix, AS path or other attributes) and a list of **actions** to be taken upon a match. Commonly an action could be accepting or dropping the announcement, but setting various BGP parameters, such as local preference, MED and community is also possible.

A route map can make use of prefix filter lists or access lists as matching rules.

2.2.4 BGP Maximum-Prefix filter

When an *hijack*, *route leak* or a *self de-aggregation* happens, a huge number of prefixes may be announced by the perpetrator to the peers, thus increasing the load on the peering routers, which have to process the vast amount of prefixes. In order to avoid this, several router OSes created the «BGP Maximum-Prefix filter», which is a threshold on the maximum number of prefixes that a peer is allowed to announce. Once the maximum number is reached, the peering session is interrupted and all prefixes received from the peer are withdrawn.

For a provider peering to a customer known for announcing only few prefixes, this is an effective way to prevent large *self de-aggregations*, *route leaks* or *hijacks.*

For a customer peering to a provider point of view, this filter might not be effective since the provider may provide the full BGP table to the customer.

Also, knowing the proper value to setup in this filter and updating it when needed is often challenging.

2.2.5 BGP communities for route leaks

As a side note slightly related to filtering: some BGP communities can be partially used to prevent ***route leaks***.

As we already said in section 1.6.2, there is a well known BGP community («*no-advertise* community» [38]) that can be used by an AS to signal that the receiver must not re-advertise the route to any other peer, thus «preventing» route leaks in theory.

Of course BGP communities may be stripped out by the receiver or mis-configured by the sender, in which case any other AS receiving the leak may not be able to discard the bogus route.

2.2.6 Limits of the prefix filtering

2.2.6.1 Prefix filtering works only on customer links

Prefix filtering is usually performed by a provider to filter announcements from customer ASes. This is because prefix filters are built on the assumption that the

filtered AS will announce only a small number of IP prefixes to the filtering AS.

Prefix filtering is not typically used by a customer AS to filter BGP announcements from transit providers or settlement-free peers [50], since they will announce to the customer a complete BGP table.

If a transit provider is accepting and forwarding an hijacked prefix, customers of the provider and other transit providers will receive the hijacking and forward it, propagating it everywhere. This is in fact what happened for example in the YouTube's incident of section 1.5.2, where the provider or Pakistani Telecom, PCCW, did not filter the hijack and forwarded it to other transit providers and customers, propagating it to all the Internet. Little could have be done in terms of filtering by other ASes (for example by *Linx Telecom*) to prevent the incident.

2.2.6.2 Manually maintain filters

If manually performed, prefix filtering might be challenging: the provider have to update the filters each time that a customer is adding a new network block. If the provider is a *tier-1* ISP with a smaller ISP as customer, the number of prefixes to filter and periodically update might become quite high for manual maintenance.

From the customer point of view, manual filtering might be problematic because nearly every change on the external announcements may require interaction with the upstream provider.

2.2.6.3 Automatically building filters

Prefix filters can be automatically built by exploiting information present in the Internet Routing Registries (IRRs, explained in section 2.3). As we will discuss later (in section 2.3) this may in theory solve the maintenance problem, but in practice can't always be effective since some operators don't want to rely on IRR data, and others may not use it for registering their own prefixes. From the point of view of some network operators, applying prefix filtering by using stale or bogus data might be more dangerous than not applying it.

2.2.6.4 Hard to apply in Internet eXchange Points

In Internet eXchange Points (IXPs), since the BGP peering is not in a customer-provider relationship, most network operators do not have prefix filters on BGP peering sessions, but only use Max-Prefix filters in order to protect against large leaks [95]. This is even more dangerous since the probability of receiving an hijack or route leak is more likely in an IXP where there are so many peers exchanging announcements between them.

2.2.6.5 Low incentives in deploying filtering

For a provider, there are no strong incentives in deploying prefix filters towards a customer, since the only incentive would be protect the rest of the Internet from attacks by its own customers. [50]

For example, we can look at the Pakistani YouTube incident of section 1.5.2. YouTube's AS had no way for filtering the hijacking announcement and preventing it from spreading all over the Internet. The only AS that could have prevented the incident by using prefix filtering is PCCW, by filtering its own customer, Pakistan Telekom. However, PCCW had no strong incentives on doing that.

2.2.6.6 Filtering «failing silently»

According to [73], another problem in implementing prefix filtering is that it «fails silently». This means that because of the filtering in place on an upstream provider, the customer (perpetrator of a mis-configuration) might not notice that it is announcing bogus information.

2.3 Internet Routing Registries (IRRs): Overview and issues.

The *Internet Routing Registries (IRRs)* are public databases where network operators can publish and retrieve information about routing policies of autonomous systems. The main purpose of these registries is to use the data present for automatically build BGP prefix filters on peering links.

A single *Internet Routing Registry* (*IRR*) «database» is a compressed textual file containing several entries, or «objects», written using the *Routing Policy Specification Language* (RPSL) [31, 32] syntax. The RPSL syntax allows to declare several type of objects:

- Maintainer object (`mntner`): It's an object holding the ownership of other objects
- Autonomous System object (`aut-num`): It's an object declaring an autonomous system and its import/export policies of routes.
- Route object (`route` or `route6`): contains an IP prefix, description fields and reference to the origin AS number.

[75] provides more information on the common use of RPSL objects on IRR registries.

There are nearly 30 IRRs at the time of writing [1], managed by several independent organizations, including ISPs, RIRs and others. Each IRR is providing *WHOIS* and *FTP* access for querying or downloading its database.

In order to prevent single point of failures, each IRR is also downloading other IRRs databases and re-uploading them on its own *FTP* site.

A network operator can make use of some or all IRRs by downloading them, extracting route objects relative to BGP peers and building prefix filters for each of them. This can be performed with a number of well-known tools, such as *IRR ToolSet* [15].

2.3.1 Issues with IRRs

2.3.1.1 Data is (mostly) not authoritative

IRRs started as a voluntary effort made by several companies and operators. Because of their nature, there is no way for the maintainers of the registries to verify the authoritativeness of the information submitted. For this reason, in several IRRs, basically everyone can publish any route object and claim the ownership of any prefix.

This was in fact largely demonstrated by the incidents that we mentioned in section 1.5.3 and 1.5.4, where attackers managed to register rogue route objects for prefixes that they didn't own, in order to bypass prefix filters on upstream providers.

In the last years, RIPE, APNIC, AfriNIC and AIRN also started to provide a IRR for their customers. Three out of four apply a mechanism which ensures a high level of certainty that the route-objects covering the respective RIR's managed space are authenticated in a proper way. ARIN does not validate against their RIR database at the moment and LACNIC has no IRR.

This means that as today, only a part of the address space can be authoritatively registered in an IRR, keeping the need for public IRRs to exist.

2.3.1.2 Lack of authentication

Some public IRRs allows operators to update their RPSL objects by sending an email to an automated program managing the IRR, which will process the email body containing the new object, and will update the database.

Various authentication scenarios for this email message are possible, depending on what was specified in the existing maintainer object:

- **No authentication**: Any email sent to the IRR is accepted.
- **Password** authentication: The existing maintainer object contains the hash of a password. When sending the email for updating the object, the *clear-text* password must be included in the email.
- **Email sender** authentication: The email address of the sender, who is updating the object, must match a regular expression specified in the existing maintainer object.

- **PGP** authentication: The maintainer object contains a reference to a PGP public key and the email message must be signed using the private key associated to the public one.

Except for the PGP case, this shows how insecure can be a submission of an entry to an IRR.

2.3.1.3 Quality of the data

IRR data has been known for having bad quality: in most registries nearly anyone can submit objects about any prefix, as we discussed in previous section 2.3.1.1.

Moreover, some operators register objects in an IRR, but later decide to don't keep the information up-to-date, since the upstream provider is not using those information for filtering. Other operators may register information in multiple IRRs and keep up-to-date information only in one of them, or change IRR and leaving stale data in the old one [62]. This is increasing the amount of churn present in the IRR registries. Having the same prefix listed in different registries might be even more challenging, since there is no clear way to know which object to trust.

These problems are well known by many network operators, who in turn prefer to avoid using that data rather than risking outages when building filters from incorrect data.

According to [92], in 2009: 46% of all the prefixes in the routing table had a valid route object. In 2013, according to [62], 71% of the prefixes announced on BGP had a valid direct match to an entry in one of the IRRs. This means that if everybody would use strict IRR filtering, 29% of the prefixes on the Internet would be unreachable.

When looking at this result, we should also consider that some IRRs have a much better quality than others. For example, while *JPIRR* is nearly completely full of valid route objects, *ARIN*'s valid route objects are less than 60% [62].

IRRs also contain a minor percentage of invalid objects that are very likely to be the result of «*proxy registration*s» by providers for customers [62]. This may happen for example when a provider registers route objects for customer's prefixes, but then the customer is announcing the prefixes from a different origin AS. The opposite is also possible: the customer is registering a prefix in the IRR, but the provider is announcing it from its own AS.

2.3.1.4 IRR registration and *IRR-based* prefix filtering

Even though, according to [62], a large (but not complete) percentage of BGP prefixes are registered in IRRs, this is not an indicator of IRRs usage. A customer may register all its routes in an IRR, but the provider may never setup filters to use that data.

By looking back at historical data of IRR registrations [18], for example, we can see that for incident of section 1.5.7, several route object were present in the RADB public registry. The same is true for the Moratel incident of section 1.5.5 (even though these entries were *proxy-registered* by another AS). During the Pakistani YouTube incident, a *proxy-registered* entry in RADB was also present for Pakistan Telecom.

A recent experiment performed on some Europeans IXPs shown how a lot of networks are not doing IRR-based filtering in IXPs [44]. In the experiment, an AS which had registered all prefixes in the IRR for a long time, started to announce a sub-prefix hijack for a prefix of another network. This was performed in 7 different IXPs against a total of 1369 peers and resulted in 931 peers (68% of them) accepting the hijacked sub-prefix, thus demonstrating that 68% of them are not using IRR data for filtering the announcements.

By looking at these examples, as well as the amount of daily hijacking and route leaks incidents, we can deduce that there are still many networks not building prefix filters from IRR data or not registering data in the IRRs.

Given the reputation of IRR data quality, it's very likely that a lot of operators are registering prefixes, but only few of them trust the use of that data for prefix filtering, unless directly specified by the peer.

2.4 Resource Public Key Infrastructure (RPKI)

The RPKI is an X.509 based hierarchy congruent with the Internet IP address allocation administration, the IANA on top, then Regional Internet Registries (RIRs), and ISPs, ... It is the substrate on which *origin* and *path validation* are based. It is currently deployed by all five RIRs: AfriNIC, APNIC, ARIN, LACNIC, and RIPE.

RPKI-based origin validation uses RPKI data to allow routers to verify that the AS originating an IP prefix is in fact authorized to do so. This is not cryptographically checked, as a BGP update message does not carry signatures, so can be violated. But it should prevent the vast majority of accidental prefix and sub-prefix hijacks on the Internet today.

RPKI-based origin validation is in shipping code from Cisco and Juniper, and others soon.

We are mostly interested in three types of objects. As described in [68], **Certification Authority** (CA) certificates within the PKI attest to IP address space and AS number ownership. Each resource holder who is delegating some of its resources or needs to publish ROAs should have a CA certificate. Each resource certificate attests to an allocation of resources to a resource holder, so entities may have multiple CA certificates. *End-entity* (EE) certificates are issued by resource holder CAs to

delegate the authority attested by their allocation certificates. The primary use for EE certificates is the validation of *Route Origination Authorizations* (ROAs), signed objects which provide an explicit authorization by an address holder that a given AS is permitted to originate routes to a set of prefixes. The private key corresponding to each end-entity certificate is used to sign exactly one object, and each object is signed with only one key so there is one EE certificate for each ROA.

A **Route Origination Authorization** (**ROA**) is an RPKI object which verifiably asserts that a specified AS is authorized to originate BGP announcements for a given set of prefixes [68]. A ROA is composed of an AS number, a list of IP prefixes, and for each prefix, a maximum length. The maximum length is a macro to authorize the AS to advertise more specific prefixes than the original prefix, up to the length as specified. For example a ROA may contain prefix `10.0.0.0/16` with maximum length `17` and origin `AS42`. This means that any announcement of prefix `10.0.0.0/16` or more specific prefixes coming from an different origin will be rejected. However, because of the maximum length value, this also means that `AS42` is authorized to announce `10.0.0.0/16` and `10.0.0.0/17` but not `10.0.0.0/18` or any more longer prefix.

2.4.1 RPKI-based BGP route origin validation

RPKI-based BGP route origin validation is performed as following:

- A copy of all the RPKI repositories is periodically downloaded by an AS into an organization-wide «cache» of RPKI data. The synchronization of this cache is performed using the *rsync* program.
- Inside the AS network, Relying Parties (RPs) download RPKI data from the local cache and validate all objects in all repositories by cryptographically verifying all signatures on all objects and skipping all objects not signed correctly.
- The RP then extract all ROAs from the validated repository and, using the *rpki-rtr* protocol [37], sends to a router a list of tuples (IP prefix, maximum length, origin AS number).
- The router will store in memory the list of tuples and periodically receive from the RP some differential updates for new entries.
- Once a BGP announcement is received on the router, the router will execute a longest-prefix match of the received IP prefix in the list of tuples and will outcome different results for the announcement:
  - **ROA not found**: There is no ROA with the given prefix or a less-specific one covering the announced prefix.

- **Valid announcement**: There is at least one ROA with a matching prefix (exact length or less-specific), correct maximum length and origin AS of the announcement matching with the one in the ROA.
- **Invalid announcement**: One or more ROAs are matching the prefix (exact length or less-specific), but none of them have the correct maximum length and/or the correct origin AS.

Depending on the validation result of the announcement, different policies can be implemented. A common policy to adopt in this initial phase of deployment is: higher «local preference» value for **valid** announcements is configured, lower preference for **ROA not found** announcements and drop of invalid prefixes [35].

In this way, for the same prefix, a **valid** announcement covered by an RPKI ROA will be preferred over an announcement from a different origin (preventing hijacks), but prefixes not covered by RPKI will still be accepted, allowing gradual deployment.

2.4.2 RPKI-based origin validation vs prefix filtering and IRRs

Compared to prefix filtering, RPKI-based origin validation is different:

In prefix filtering, the goal is to decide which prefixes can be received through a peering link, whatever is the AS path or origin. The goal of the filtering is to protect the network from the peer.

In RPKI-based origin validation, the goal is to choose which announcement is valid or not, looking at the origin AS in the AS path, regardless of where it came from.

For this reason, while prefix filtering can only be used for filtering announcements from customers, RPKI-based origin validation can be used to filter announcements regardless of the neighbor. For this reason it avoids the problem of low incentives that we discussed in section 2.2.6.5 and allows any AS to filter invalids coming from any other AS on the Internet, regardless of the peering topology.

A major difference with IRRs data is that, while prefix filtering can't be properly applied unless all prefixes of the peers are registered in an IRR, RPKI-based origin validation can be performed also when only part of the prefixes are registered in the RPKI, thus encouraging deployment.

Another important detail to mention is that for IRR-based filtering an external program must be used to build prefix filters to later install on the router. Instead, for RPKI-based origin validation, major router vendors are providing an implementation on the router OS itself.

2.4.3 Preventing routing incidents with RPKI-based origin validation

Table 2.1 shows which of the incident types that we presented in section 1.3 could be prevented by using RPKI-based origin validation.

2.4.3.1 ROA Maximum length benefits

A *self de-aggregation* or a *sub-prefix route leak* can be prevented with origin validation if the maximum length field of the ROA is properly configured.

For example, an AS announcing `10.0.0.0/16` and `10.0.0.0/17` could register a ROA for `10.0.0.0/16` with maximum length 17. When, due to an incident, the AS starts to *de-aggregate* into more specifics, like `10.0.1.0/24` and `10.0.2.0/24`, other networks will not accept such prefixes because of the maximum length field in the ROA.

However, if the AS is normally announcing prefixes with length 24 and using a ROA `10.0.0.0/16` with maximum length of 24, a *self de-aggregation* can cause incidents.

The same discussion can be applied for *sub-prefix route leaks*.

2.4.3.2 Prevent incidents examples

Of the types of incidents listed in section 1.5, 6 out of 8 could have been prevented with RPKI-based origin validation. Moratel 1.5.5 and Telecom Malaysia 1.5.6 are the only ones that couldn't have been prevented because they are **route leaks**. This is sadly the limitation of RPKI-based origin validation: it is not possible to prevent *route leaks*. *Route leaks* may be solved by proper use and filtering of BGP communities, as discussed in section 2.2.5.

As we said, the purpose of RPKI-based origin validation is to prevent accidental hijacks or accidental self de-aggregations, not attacks. For this reasons, attacks such as the ones of section 1.5.3 and 1.5.4 could have been performed even with RPKI, in case the attackers had spoofed the origin AS (*path-shortening hijack*).

2.4.3.3 Prevent hijacking of unused address space

An interesting detail is the use of AS number 0 in a ROA, which is described in [55].

A ROA stating that a given prefix can be announced by `AS0`, is meaning that the prefix and any more specifics can't be announced by any AS on BGP (unless there is another other valid ROA for the given prefix saying the contrary).

A network operator could use a ROA containing `AS0` in order to protect an unused address space from all types of hijack and route leak shown in table 2.1.

Table 2.1: Summary of all types of incidents seen in section 1.5, and whether they could have been prevented by using RPKI-based route origin validation.

Incident types	**More specific prefix announced**	**Correct origin AS**	**RPKI-based origin validation could prevent?**
Prefix hijack	No	No	Yes
Route leak Path-shortening hijack	No	Yes	**No**
Sub-prefix hijack	Yes	No	Yes
Sub-prefix route leak *Self de-aggregation*	Yes	Yes	Yes (if proper maximum length in the ROA)

This, for example, could have prevented the incident of spammers shown in section 1.5.4.

Moreover, in the future IANA or RIRs may register ROAs with `AS0` for all known *bogon* prefixes[1].

2.4.4 BGPSEC: Path validation with RPKI

RPKI-based origin validation is designed to prevent only accidental incidents, not malicious attacks where the attacker may manipulate the AS path.

The definitive solution to secure the AS path in accidental or malicious hijacks and route leaks, currently being standardized by the IETF is *BGPSEC*.

With *BGPSEC*, a BGP receiver can cryptographically validate that the originating autonomous system is truly authorized to announce the IP address prefix. Moreover, the receiver can also verify that the ASes through which the announcement passed were indeed those which the sender/forwarder at each hop intended.

Standardization work is also ongoing in order to enable *BGPSEC* to prevent route leaks too [91].

BGPSEC works as presented in the following example:

- `AS1` generates a private/public key pair for its BGP router, which is signed by its CA, part of an RPKI repository
- `AS1` generates an announcements of prefix `10.0.0.0/16` for `AS2`, sign it using its own private key and forwards.
- `AS2`, who will receive the announcement, will have a copy of RPKI repositories which it can use to verify the signature on the BGP announcement just received from `AS1.`

[1] A *bogon* prefix is a route that should never appear in the Internet routing table. These are typically private IP ranges for RFC1918 [86] and other reserved ranges. These prefixes are typically filtered on BGP ingress prefix filters.

- `AS2` will then append its own AS number on the AS path on the signed announcement of `AS1` , will prepare and will sign the new announcement for AS3 using its own private key.
- `AS3` receives the announcement and repeat the same process of `AS2`.

Each time the announcement is being forwarded, new signatures are added on top of existing ones, thus allowing the final receiver to unpack each signature and verify that the announcement went exactly through the path shown in the AS path field.

This approach have sadly two main problems [50] which slow down deployment:

- Each router have to perform «online» cryptography, that is: each announcement must be validated and re-signed. This requires more powerful hardware in routers, other than new firmware for implementing the new protocol.
- With BGPSEC it is not possible to validate the correctness of the path unless **all** ASes along the path are signing the announcement. In the previous example, if `AS2` is not signing the announcement, `AS3` or a following `AS4` have no way to be sure that the AS path has not being mangled between `AS1` and `AS3`, or that `AS3` is not originating a fake announcement.

RPKI-based origin validation is in one sense a first step in working towards BGPSEC, which will allow operators to already have experience with RPKI repositories when deploying BGPSEC.

2.4.5 RPKI issues and need for measurements

Registration in the RPKI and route origin validation is gradually positively increasing. In the following we will present some of the issues raised while RPKI started to be deployed. These are some of the reasons that pushed some network operators to delay the RPKI deployment, by considering that the effort to be spent to solve these issues is not worth the benefits. Lack of perceived need, combined with insufficient manpower and expertise in the area [99, 101] are example of issues shared with the adoption of other technologies as well. RPKI is however also slightly affected by legal issues, revealing business relationships and doubts on the quality of the data.

2.4.5.1 ARIN's agreements legal issues

In the North American region, ARIN requires any operators wanting to use the ARIN RPKI data to sign a Terms of Service Agreement that includes an indemnification clause [89]. This is causing legal teams of several organizations to hesitate on the implications of RPKI use (ex: [49, 90]). This may also in some way slow down deployment since each organization have to wait for the agreement to be processed

by their legal team before getting access to the repositories. Moreover, ARIN requires **legacy resource holders** (see section 1.2.0.2) to sign an *Legacy Registration Services Agreement* (LRSA) in order to be eligible to register their resources in the RPKI [89]. With the ongoing depletion of IPv4 addresses (widely discussed in [89]), legacy address space holders are in a delicate position and might be interested to prevent ARIN from taking the address space back. For this reason, this agreement is also probably preventing many organizations holding legacy address space to register RPKI resources due to possible legal implications.

2.4.5.2 Fear of court orders and law enforcement

In 2011, after a series of events (described in detail in [78]), RIPE NCC was ordered by a Dutch court to «lock» registration activities regarding certain address blocks related to criminal activities of the *DNSChanger malware*. RIPE NCC complied the court order, but this raised an issue on RIRs that can be controlled by countries. While RIPE NCC is a geographical entity covering multiple countries, it is based in Amsterdam, Netherlands; and for this reason it could be subject to Dutch government control.

About one year after the court order, RIPE NCC also filed a summons against the Dutch government [28].

This event raised several issues regarding whether RIR's actions could be controlled by a government, and did also affected discussions about RPKI which was starting deployment at the time of the court order. With RPKI, a RIR could perhaps revoke and re-issue certificates in RPKI at the request of a law enforcement agency in some distant jurisdiction, resulting in influencing the global routing.

The DNS infrastructure is already subject to legal orders to take down websites [82], however a DNS name is still just a reference to a resource in a network. In a future where *Secure BGP* is deployed, such attacks could be very effective in completely taking down an entire network with a single law-enforcement operation.

Ad discussed in [78]: «*In other words, the RPKI debate is exactly about the possibility of linking address registries to routing activity*».

The IETF community have perceived these issues and discussion about how to address this problem are currently ongoing [36], as well as papers discussing issues of mis-behaving RPKI authorities [39].

Ultimately, we could summarize this debate as general disadvantages for *Public Key Infrastructures*, since *DNSSEC* and *SSL* PKIs have very similar issues.

2.4.5.3 Revealing business relations

In [101] it is discussed how one of the issues raised by operators on RPKI is the potential for *ROA* files to «reveal business relations», since they associate a prefix

to an AS which is authorized to announce it.

One example scenario described in [101] is when «*two large CDNs serve secretly as backups for each other. [...] Similarly, smaller CDNs that rely on third party networks (e.g. using Verisign for external DoS mitigation) may see such information as damaging to their reputation. Despite this, in both cases, RPKI would publicly reveal these setups*».

Looking glasses and BGP collectors are different from RPKI data, because they can provide insight into peering relations but only after the event has occurred, while RPKI exposes more information in advance.

2.4.5.4 Unclear quality of RPKI repository data and Related Works

Prior to our work, a measurement from 2011 [98] was reporting more invalids rather than valid prefixes when using RPKI-based origin validation. Another measurements in 2012 from the same author [100] resulted in a more positive outcome, but did not provided many details about reasons behind these invalids, nor an history showing the trends in the deployment. Moreover, [100] was published in 2012, when deployment had just started.

More recent and closest works to ours are [63] and [46]. They provide snapshots of route validation in specific deployments. Here we go further as we study route validation over an extended period of time. In addition, we provide statistics regarding the RPKI infrastructure, and the registration of resources and events caused by the operation of the infrastructure.

2.4.5.5 Local policies and dropping invalid announcements

In RFC7115 [35], it is suggested to always drop invalid announcements, rather than applying a lower preference. This is because sub-prefix hijackings would be still possible if invalids are accepted and this would go against the purpose of RPKI validation.

However, [64] discussed how invalids should be accepted due to the amount of them that would result in loosing connectivity if dropped. Many operators started to wonder if dropping invalids could be a safe policy, looking at the amount of reported invalids by previous works.

Moreover, other operators discussed about the lack of policy knobs for handling origin-validated announcements [90].

Chapter 3

Measurements on RPKI

3.1 RPKI repositories registration analysis

Before looking at real BGP announcements compared to available RPKI data, we want to take a closer look to the RPKI repositories alone, in order to understand the current effort on the registration of resources.

3.1.1 Current registration status

In table 3.1 we present the number of IPv4 host addresses (/32s) covered by ROAs by each RIR publication point, compared to the number of IPv4 host addresses allocated from IANA to each RIR.

Here we can observe that while ARIN has allocated most of the address space, it lags far behind the other Northern RIRs in registrations, the same is also true for APNIC. RIPE NCC is currently the leader in terms of both absolute and relative amount of allocated address space covered by ROAs, and LACNIC is quite active. The global percentage of IPv4 address space covered by a ROA is 6.03%.

Table 3.1: Deployment status of the registration of IPv4 addresses on September 1, 2015 (data from our RPKI data archive presented in section 3.2.1) compared to the allocation of IPs by these RIRs on the same day ([10–14]).

Regional Internet Registry	**Number of IPv4 host-addresses covered by a ROA**	**Number of IPv4 host-addresses totally allocated**	**Percentage coverage**
RIPE NCC	151,728,434	812,429,160	18.67%
ARIN	33,704,192	1,695,121,920	1.98%
LACNIC	26,418,882	190,357,760	13.87%
AfriNIC	4,226,048	79,582,976	5.31%
APNIC	3,479,808	860,793,088	0.40%
Total	**219'557'364**	**3'638'284'904**	**6.03%**

3.1.2 Registration trends among repositories

In figure 3.1 we show, for each RIR, the **number of ROA files** authenticated by *rcynic* [20] between March 2012 and September 2014 using a log-scale on the y-axis.

By counting the number of authenticated ROA files instead of prefixes or host addresses, we may not show a precise view of each repository. However this choice, together with the use of the log scale, was made because our goal here is to analyze trends on the registration of ROAs and compare various repositories, rather than give exact numbers.

On the graph we can observe the following details:

- Between July and August 2013 there is a one-month data hole, due to a problem in our data collection system.
- We started collecting ARIN's data only starting from August 2014, because of ARIN's legal barriers placed on RPKI use[1].
- LACNIC data is interrupted from the end of December 2012 to mid August 2013. We believe the reason for this is the expiration of the manifest file related to their trust anchor. This resulted in making LACNIC trust anchor and thus all ROAs in the repository to become invalid.
- APNIC repository also had a similar expiration event for seven months between

[1] As we said in section 2.4.5.1, ARIN has a policy of providing access to the data only to those who have signed a document.

January and August 2013. The fact that this went undetected is operationally quite disturbing.

- Between November 2012 and February 2013 we can see the effects of key rollover on the RIPE data.
- We started to collect AltCA (also known as «CA0») repository data on August 2013.
- We can observe regular drops in the number of ROAs for AltCA because this data is hosted on a machine that is regularly disconnected from the Internet for extended periods of time, giving time for object to expire without being renewed on time.

On 2 March 2015, AfriNIC suffered of the same operation mistake of LACNIC and APNIC related to the manifest file expiration. While performing our study, we detected the event, contacted AfriNIC and managed to solve it in few hours [59].

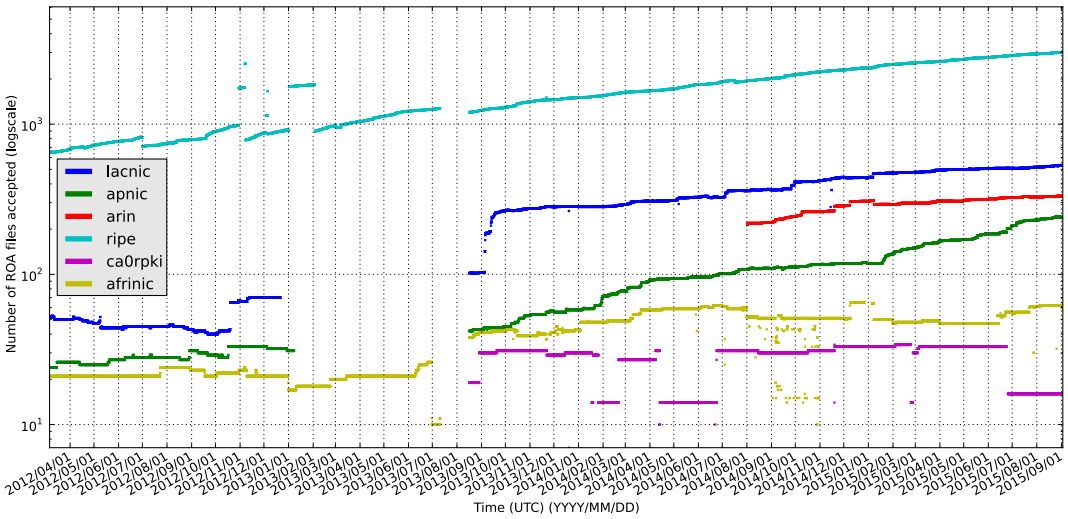

Figure 3.1: Number of authenticated ROA files below the six trust anchors. The discontinuous increases in number of ROAs observed for RIPE NCC occur during key rollovers. LACNIC and APNIC face a loss of valid ROAs for roughly seven months, likely due to an expiration of the manifest related to their trust anchor certificate. There is a hole in our data, for all trust anchors between July and August 2013.

3.2 Validating BGP data against RPKI repositories

Our goal is to find out how many BGP announcements and prefixes announced on BGP are actually covered by a valid ROA record present in an RPKI repository, and how many valid, invalid or «not found» announcements or prefixes are there. We want to perform these measurements on the current data, but explore historical data too.

3.2.1 Data sources

This study relies on a private archive of RPKI data: for every hour from March 2012 to August 2014, a server running *rcynic* [20] has been downloading all RPKI repositories, validating the data contained in them and archiving it.

As the IANA has not been allowed to provide an RPKI root, we chose trust anchors following the recommendation of the IETF *SIDR* working group [68], using the *rcynic* tool [20] to download ROAs from the RIPE, LACNIC, AfriNIC, APNIC and CA0 (Also known as «*AltCA*») trust anchors, with two exceptions. For legal reasons, we only have ARIN data starting from August 2014 and we add the CA0 data. CA0/AltCA is the trust anchor for some legacy and experimental address space that ARIN would not register.

For the BGP data we use BGP RIB table dumps archived by the RouteViews project [25], for the same period. RouteViews RIB dumps are available every two hours for several locations of route collectors. We mainly use the data collected from the *London Internet Exchange* (LINX), since it is peering with a lot of other ASes, thus increasing our view on global BGP announcements. However we also consider the *ISC*, *Sao Paulo*, *Sidney* and *WIDE* RouteViews collectors later.

3.2.2 Validation process and measurement infrastructure

Our RPKI-based origin-validation on historical RPKI and BGP data is performed using a program we developed with Python language. In the following we describe at high level the various steps for performing the origin validation:

- In order to have an historical view of the trends, but at the same time avoid processing too much data, we decide to pick a RIB dump from RouteViews archives every 30 days.

- After choosing a RIB dump, our program goes through all archived RPKI data and selects a complete archive of authenticated RPKI data with an archive *timestamp* older than the RIB dump, but as close as possible in time. Because RIB dumps are taken every two hours and our RPKI archives are taken every hour, there is no risk in validating against stale information.

- Each RPKI data archive is containing all RPKI file objects previously authenticated by *rcynic*. Once the RPKI archive is selected, all ROA files are extracted and each ***record*** (consisting of IP prefix, maximum length and authorized origin AS) is added in a *prefix trie*, using the IP prefix as key and the ROA record data as value. Loading ROA records can be performed thanks to the use of Python bindings of *rcynic* libraries. The *prefix trie* will have several leaves and nodes, each of them containing one or more ROA records data.

- At the same time, the selected BGP RIB dump, which is encoded in MRT format [33] is processed using the BGPdump program [5]. BGPdump converts it in a textual format decoding each announcement and its information, which are mainly: *IP prefix*, *peer AS*, *peer IP* and *AS path*.

- For each BGP announcement decoded by BGPdump, we search the associated ROAs in the *prefix trie* by using a longest prefix match and we validate the announcement as following:

    1. If no such node is found, we mark the announcement as «ROA not found»
    2. If a longest-prefix match is found, for each ROA record present in the node, we check if the max length of the node covers the announced prefix and if the AS number specified in the ROA record is equal to the origin AS number of the announcement. If these conditions are met, the ROA record validates the route announced for the prefix, so we mark the announcement as «valid».
    3. If a longest-prefix match is found, but all ROAs contained in the found node are not satifying the conditions described in point 2, we move upward in the tree to the parent node until we found at least one ROA record correctly matching the announcement. If such ROA record is not found, we mark the announcement as «invalid».

- Once the announcement is classified as «valid», «invalid» or «ROA not found», the result of such validation (together with the information about which ROA records has been used for such validation), is saved in an output CSV file containing the list of validated announcements.

In a RIB dump, there are very often multiple announcements for the same prefix, as a monitor may learn the same prefix from different BGP peers. The validation result for each of the announcement might be different depending on the origin AS on the AS path. For this reason we validate each announcement separately and write the result of this validation process in a CSV file.

3.2.2.1 Optimizations

All the analysis on each dump has to be performed only once in order to obtain the statistics we want to show in the following. For this reason, spending a lot of time in parallelizing the process of validation RIB dumps was not a priority for our work.

However, in order to gradually develop this tool and add new features and details to the validation output, we had to find ways to save time in the processing each RIB dump.

For example, the RIB dump have announcements (typically about 10 million of them) sorted by announced prefix. Because for the same IP prefix we have to perform the exact same lookup and find the same result in the *prefix trie,* we keep a «cache» dictionary mapping the last IP prefixes looked up in the *prefix trie* to its ROA records. In this way, after the first announcement for a prefix is looked up in the *trie*, we can validate following announcements much faster (in nearly $O(1)$, compared to $O\left(log\left(n\right)\right)$).

Another example: the extraction of RPKI archives may take some time. For this reason we perform it in a parallel process taking care of gradually extracting RPKI archives that will be used for the next RIB dumps to be processed.

Moreover, looking at the slow and incomplete alternatives for processing BGP RIB dumps from Python, we decided to use BGPdump program instead. Because decoding a complete dump may take several minutes, we structured our program so that it can process decoded BGP announcements in pipeline while BGPdump is running, saving more time.

3.2.2.2 Classifying data

After for each RIB dump we generate a validated RIB dump (in the form of a CSV file), we need to perform a long number of classifications (that we will present in the following sections). Most of these classifications are based on the idea of counting how many prefixes with a certain characteristic are present in the RIB dump. The goal of other of these classifications is to find out how many prefixes have a certain characteristic in at least one or more announcements.

For example, we need to classify which are the prefixes which are «valid only» (valid in all announcements where we find them), «invalid only» (invalid in all announcement where we find them) or «valid and invalid» (valid in some announcements, invalid in other announcements).

In order to perform such classification, we made extensive use of Python's *hash tables* (in the form of the «*set*» and «*dict*» data types). For example, for classifying valid/invalid/valid&invalid, we go through each announcement and we check if we already classified the prefix in one of the first two categories by doing a lookup in the relative *sets*. If the prefix was in the «valid only» set and an invalid announcement is found, the prefix is moved in the «valid and invalid» set and so on.

After all classifications are completed for a single RIB dump, counters on the number of prefixes in each *set* are saved in a temporary file and used later for building graphs.

3.2.2.3 Graph generation

Once we have counters for all classifications of prefixes and announcements, another Python script is generating several graphs (stacked bars and line plots) using Python's MatPlotLib library [16]. Each graph is generated by specific code selecting specific counters computed in the previous phase.

3.2.3 Overview on validated BGP announcements

A validated RIB dump will have several announcements of the same prefix from different origin ASes, or from the same origin AS. For this reason we then extract from the validated RIB dump only a single announcement per *(IP prefix, origin AS)* pair. By doing this on the RIB taken from LINX route collector, we get what can be observed in figure 3.2 and figure 3.3.

Note: from now on we will consider IPv4 and IPv6 prefixes together, so every percentage or count that we will present will include both of them.

In the last data from August 2015 we have 585'272 pairs announced. Among these pairs, 36'762 of them (6.28%) are covered by some ROA.

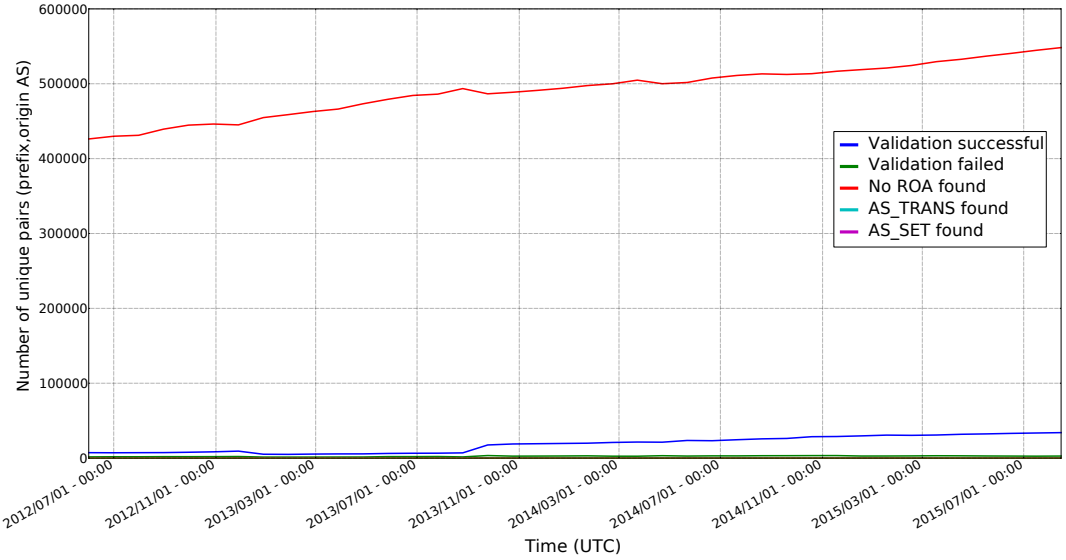

Figure 3.2: Number of (IP prefix, Origin AS) pairs not covered by ROA record, valid invalid or containing AS_TRANS/AS_SET as origin.

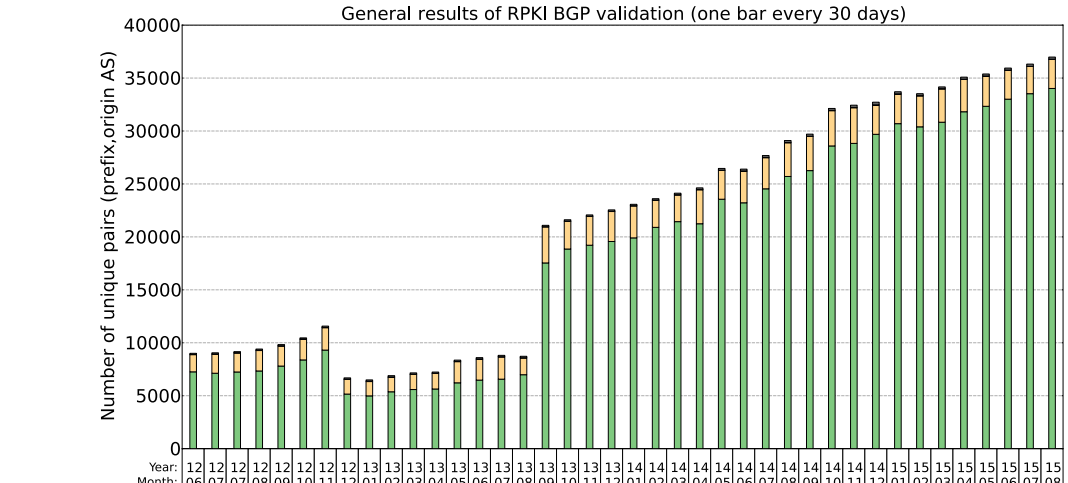

Figure 3.3: Percentage of (IP prefix, Origin AS) pairs correctly validating against some ROA, failing validation or containing AS_SET or AS_TRANS as origin. Note: the numbers shown in the table under the plot are percentages relative to the total of the four categories shown, not referring to all existing announcements.

BGP announcements containing AS_SET are excluded from our analysis because the function of AS_SET is deprecated with the deployment of RPKI [65] and can lead to ambiguity in the origin AS of the announcement.

We also exclude announcements with `AS23456` (or «AS_TRANS») as origin AS, because it should not be announced on BGP and it's the result of some misconfiguration in the usage of 4-octect AS numbers by few operators.

According to the data from August 2015, only 199 pairs (0.03%) contain an AS_SET, and only 22 pairs contains AS_TRANS as origin AS.

In figure 3.3 we are considering only pairs covered by RPKI, and we can see the percentage of valid and invalid pairs. Here we can note that 7.46% of the RPKI-covered pairs are failing the RPKI-based origin validation process.

We can see a very significant decrease of pairs being validated between December 2012 and August 2013. This is due to the problem in the LACNIC and APNIC repositories mentioned in section 3.1.2. During this period, all ROA files of the LACNIC and APNIC repository disappeared. Thus, all the prefixes which were previously covered by these ROAs become labelled as «*ROA not found*».

3.2.4 Counting prefixes

So far we have been looking at *(IP prefix, origin AS)* pairs, but it is more useful to count by considering IP prefixes as unit of measurement, and it is also common

practice for measurements regarding BGP.

In figure 3.4 we can see the total number of unique prefixes taken from the previous dataset, compared to the number of unique prefixes covered by a valid ROA record.

In figure 3.5 we can see, among the prefixes covered by RPKI only, which is the percentage of prefixes valid in all announcements, invalid in all announcements or valid in some announcements and invalid in others.

Table 3.2 shows some of the relevant data points of figure 3.5. By considering unique prefixes we can observe that 6.31% of the global prefixes are covered by RPKI.

Again, because a prefix can be announced from different origin ASes, it's possible that some announcements for a prefix are valid and some others for the same prefix are not. However, the amount of these «valid and invalid» prefixes is very low: in the dump of August 2015 we can count only 25 of them.

These «valid and invalid» prefixes might be the result of several things, such as a prefix hijack or a prefix anycasted from two ASes with ROAs missing for some of the potential origin ASes. Among these 25 prefixes, one have the invalid announcement sourced from a private origin AS number, that in theory should not be leaked to the global BGP traffic. 9 of these 25 prefixes are anycasted, in fact the failing AS and the valid AS have very similar or same AS names. Others of these «valid and invalid» prefixes are probably the result of IP space transfers, that we will discuss later in section 3.3.4.

We can also note, from data of table 3.2, that not only the percentage of invalid prefixes is decreasing, but the number of them is decreasing too. This means that problems are being solved and not that there are more valid prefixes making invalids a smaller percentage.

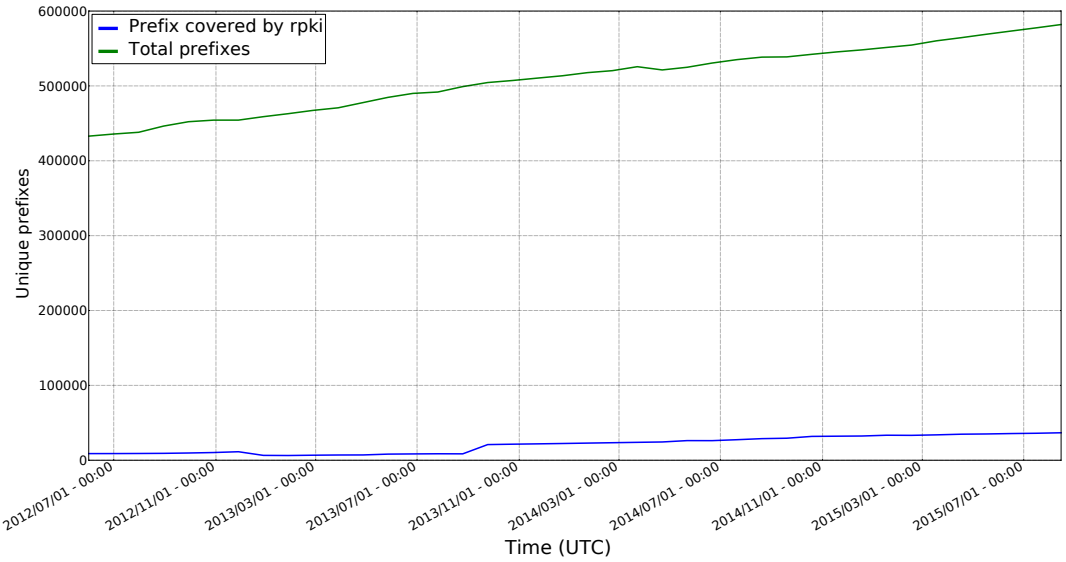

Figure 3.4: Total number of unique prefixes announced, compared to the number of prefixes covered by some RPKI ROA record.

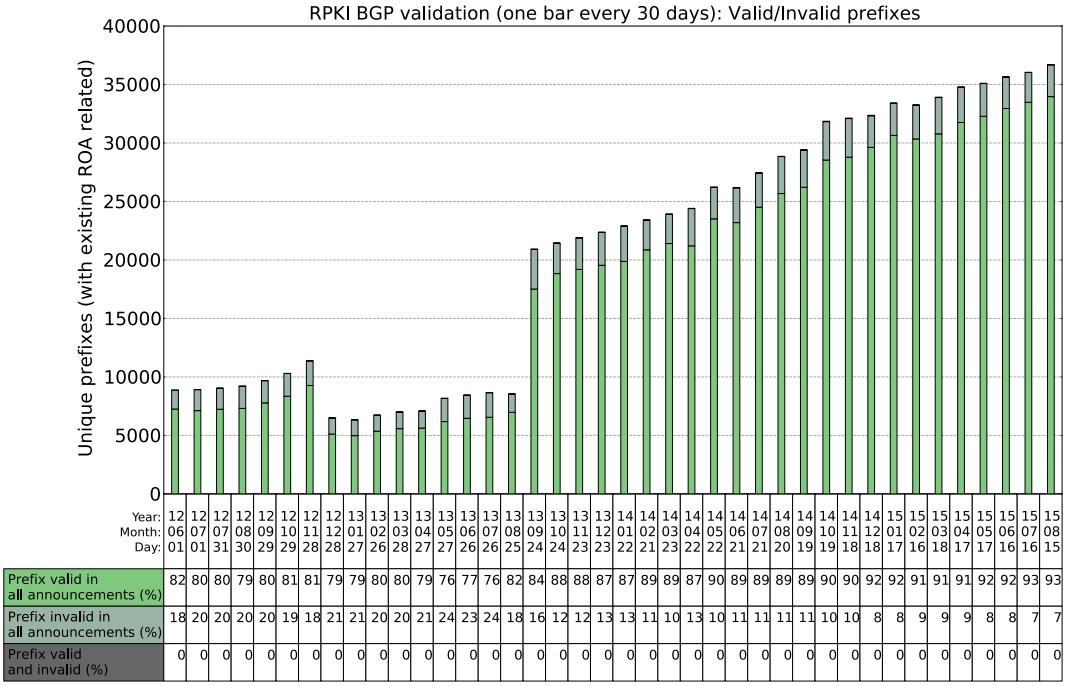

Figure 3.5: Percentage of prefixes that, after RPKI-based origin-validation are classified as «valid», «invalid» or «valid and invalid». The last category is regarding prefixes which are valid for some announced origins and invalid for others. Note: the numbers shown in the table under the plot are percentages relative to the total number of prefixes covered by some RPKI ROA.

Table 3.2: Counters and percentages of prefixes classified as «valid», «invalid» or «valid and invalid» for relevant data points relative to figure 3.5.

Date	Total number of unique prefixes seen	Percentage of RPKI-covered prefixes among all unique prefixes seen	Valid prefixes	Invalid prefixes	Valid and Invalid prefixes	Percentage of Invalid prefixes among RPKI-covered prefixes
2012/06/01	432'905	2.05%	7'253	1'622	0	18.28%
2012/11/28	454'294	2.51%	9'269	2'100	13	18.45%
2012/12/28	458'897	1.42%	5'117	1'366	16	21.02%
2013/09/24	504'488	4.15%	17'513	3'395	12	16.23%
2014/05/22	524'891	5.00%	23'516	2'686	31	10.24%
2014/07/21	535'101	5.13%	24'510	2'916	18	10.63%
2015/02/16	554'544	6.00%	30'345	2'882	30	8.67%
2015/08/15	581'963	6.31%	33'963	2'712	25	7.39%

3.2.5 Reasons of invalidity

Even with a measurement by unique prefixes we still see 7.39% of *RPKI-covered* prefixes failing validation (2'737 of them), and few others valid only in some announcements and invalid in others (25 of them).

In order to try to find out the reason of these invalids[2], we split them by reason of invalidity. In order to do that, we look at the reason why the ROA record(s) present in the longest-prefix matching node of the prefix trie does not match the announcement under validation. The result of such analysis is shown in figure 3.6.

We divide the failed validations into three categories:

- **Invalid maximum prefix length**: For example, the monitor receives an announcement for `10.1.2.0/24` but the ROA record covers only `10.1.0.0/16`, maximum length 16.
- **Invalid origin AS number**: The monitor receives an announcement by `AS666` for `10.1.2.0/24` but the ROA record authorize `10.1.2.0/24` only from `AS42`.
- **Both maximum length and origin AS number**: At least two ROA records are found in the longest-prefix matching node for the prefix, one or more of them failing on AS number, the other(s) failing on maximum prefix length; or there is a single ROA failing for both reasons. This may cover a lot of different causes and we don't have enough information to classify them.

We can see that mismatched maximum length is the most widespread cause for invalids. There are less invalids due to non-matching origin ASes.

[2]Note: in the following we will consider both «invalid only» and «valid and invalid» prefixes of the previous section.

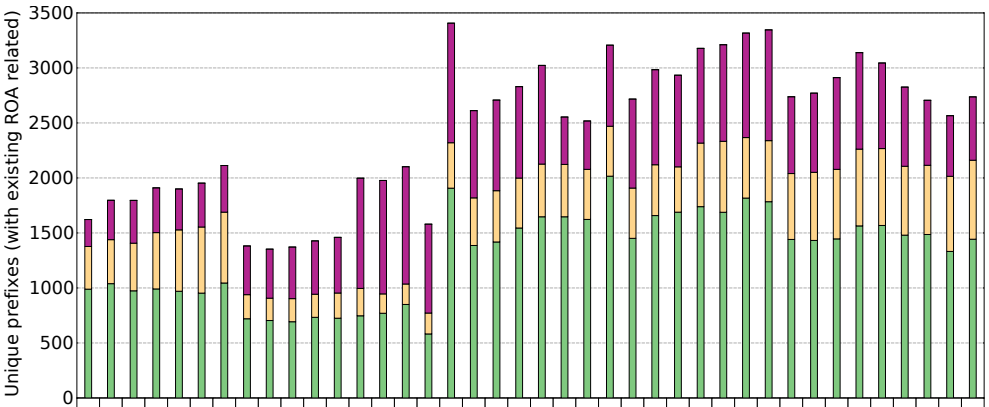

Figure 3.6: Breakdown of invalid prefixes, by failing cause, as seen by *Route Views LINX* monitor. Note: the numbers shown in the table under the plot are percentages relative to the total number of «invalid only» and «valid and invalid» prefixes for the given time snapshot.

3.2.6 Taking prefix coverage into account

We found invalid prefixes and we shown reasons of invalidity.

It is often assumed that operators who validate advertisements will drop invalids, so the next question that we want to answer: «*Is the presence of these invalid prefixes so problematic for the deployment of RPKI-based origin-validation?*».

In order to better understand the effect of the invalid dropping policy on reachability, we cannot simply look at prefixes separately. We need to consider the ***coverage* of invalids by other prefixes**.

Let's assume that a BGP border router receives the same routes as our LINX monitor and it drops all «invalid only» prefixes. In addition, in the deployment phase, we expect operators to also accept announcements for prefixes with no ROA. If a prefix is «valid» or «valid and invalid», we consider it as reachable, because it means that at least one valid announcement for that prefix was present. When a prefix is marked «invalid only», the router will drop the announcement, but there are some cases where it could be reached anyway:

- The invalid prefix is **up-covered by another valid** prefix. For example: announcement of `10.1.2.0/24` is invalid, but `10.1.0.0/16` is also announced and valid, so the router can reach `10.1.2.0/24` anyway exploiting the covering valid announcement.

- The invalid prefix is **completely down-covered by other valid** prefixes. For example: announcement of `10.1.0.0/16` is invalid, but `10.1.0.0/17` and

`10.1.128.0/17` are also announced and valid.

- The invalid prefix is **up-covered by a «ROA not found»** prefix. For example: announcement of `10.1.2.0/24` is invalid, but `10.1.0.0/16` is also announced and there is no covering ROA for the latter.

So we can finally say that a given prefix is **reachable** if it is «ROA not found», «valid only», «valid and invalid» or «invalid only» covered as in one of the three cases above. Instead, when a prefix is «invalid only» and there is no coverage of any of the cases described above, we can say that it is **unreachable**.

In figure 3.7 we can see a graph composed by two bars for each time slot:

- The bar on the left side is exactly the same already seen in figure 3.5 and shows the number of «valid», «invalid» and «valid and invalid» prefixes
- The bar on the right side shows the reachability of prefixes considering coverage.

Table 3.3 is reporting relevant data points of figure 3.7.

In the last dump from 15 August 2015, we can note that 65.19% of «invalid only» prefixes are in fact **reachable**. They are «rescued» by another «valid» prefix or a «ROA not found» prefix.

We previously said that 7% of *RPKI-covered* prefixes were invalid. If we consider coverage we can now say that among all *RPKI-covered* prefixes, only 2.57% (944 prefixes) are actually unreachable.

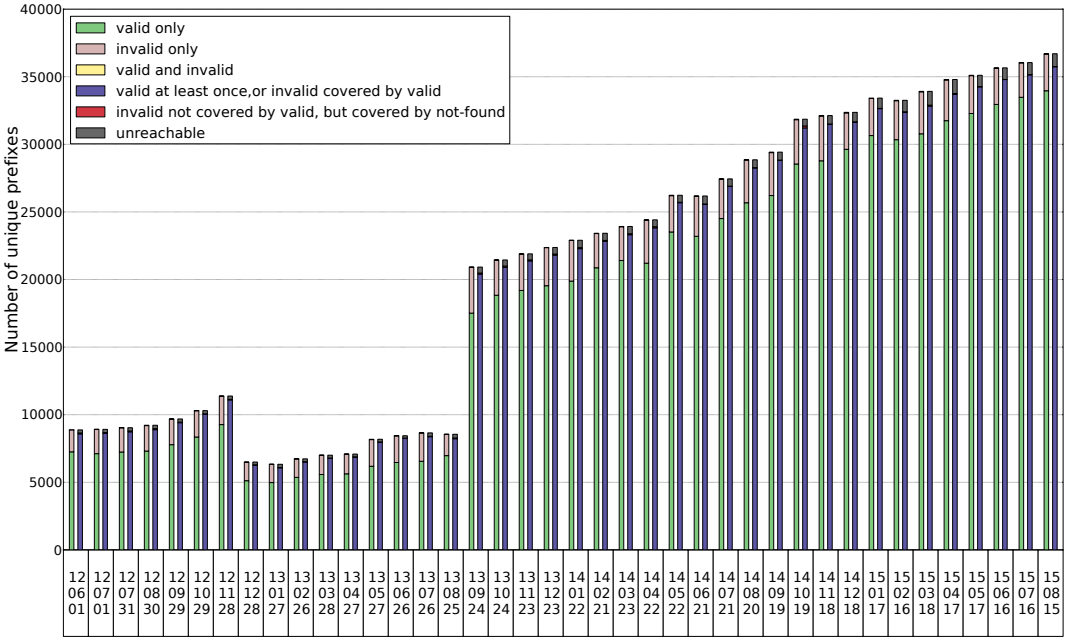

Figure 3.7: Validity status of routes seen by *Route Views LINX* monitor between June 2012 and August 2015. The first (green, pink and yellow) bar shows the status of prefixes independently from the existence of covering prefixes. The second bar (blue, red and grey) illustrates the reachability of a prefix considering that an invalid prefix might be covered by another valid or «ROA not found» prefix. Some relevant data points in table 3.3.

Table 3.3: Relevant data points relative to figure 3.7.

Date	**Reachable prefixes among *RPKI-covered* prefixes**	**Unreachable prefixes among *RPKI-covered* prefixes**	**Percentage of invalid prefixes covered**
2012/06/01	8'648	227	86.00%
2012/11/28	11'137	245	88.33%
2012/12/28	6'290	209	84.70%
2013/09/24	20'478	442	86.98%
2014/05/22	25'710	523	80.53%
2014/07/21	26'903	541	81.45%
2015/02/16	32'402	855	70.33%
2015/08/15	35'756	944	65.19%

When considering the last dump of August 2015, these 944 unreachable prefixes are announced by 219 unique AS numbers. Moreover, 20 of these ASes account for 58% of these unreachable prefixes. In figure 3.8 we show some of the ASes announcing more unreachable prefixes. The AS announcing more invalids for this

snapshot is a datacenter operator in Germany.

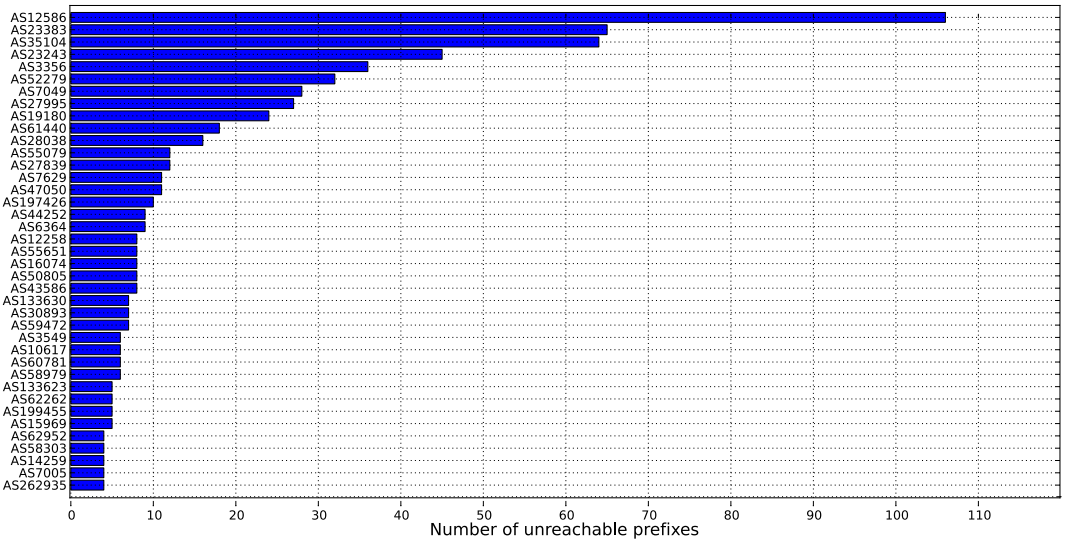

Figure 3.8: Some of the top ASes announcing unreachable prefixes, sorted by number of unreachable prefixes announced, for the data from 15 August 2015 of Route Views LINX monitor.

3.2.7 Effect of using a single BGP route collector/monitor

Up to now we considered the data from the *RouteViews LINX* monitor. This monitor is interesting because it receives a lot of heterogeneous announcements.

However, we would like to verify if our observations are highly dependent on that monitor. For that purpose, we consider 4 additional RouteViews collection points: **ISC** (Palo Alto CA, USA), **SAOPAULO** (Sao Paulo, Brazil), **SYDNEY** (Sydney, Australia), **WIDE** (Tokyo, Japan).

Figure 3.9 compares the number of reachable and unreachable prefixes at these different locations. The main difference between monitors is that they do not receive announcements for the same amount of prefixes (see Table 3.4 and Table 3.5). However, the percentage of *RPKI-covered* prefixes seen is very similar. We think that in order to detect specific events, it might be better to combine the data from all monitors, but for the purpose of our measurements it's enough to consider one of the biggest. The percentage of unreachable prefixes due to an invalid origin is almost the same at any of the 5 locations considered.

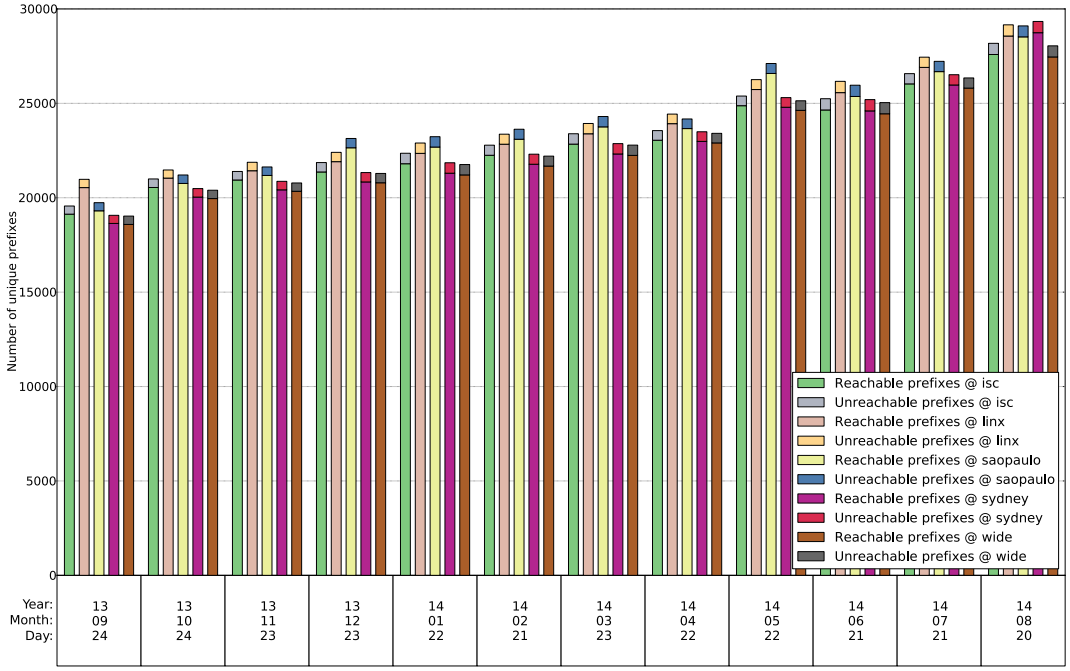

Figure 3.9: Reachable and Unreachable prefixes from different RouteViews monitors.

Table 3.4: Reachable and Unreachable prefixes from different RouteViews monitors on 24 September 2013

Monitor	**Total number of unique prefixes seen**	***RPKI-covered* reachable prefixes seen**	***RPKI-covered* unreachable prefixes seen**	**Percentage of unreachable**	**Percentage of *RPKI-covered* prefixes**
ISC	501,186	19,130	432	2.25%	3.90%
LINX	504,773	20,537	438	2.13%	4.15%
SAOPAULO	494,730	19,308	430	2.22%	3.98%
SYDNEY	500,523	18,639	430	2.30%	3.80%
WIDE	486,403	18,586	442	2.37%	3.91%

Table 3.5: Reachable and Unreachable prefixes from different RouteViews monitors on 20 August 2014

Monitor	**Total number of unique prefixes seen**	**RPKI-covered reachable prefixes seen**	**RPKI-covered unreachable prefixes seen**	**Percentage of unreachable**	**Percentage of RPKI-covered prefixes**
ISC	540,197	27,587	591	2.14%	5.21%
LINX	538,926	28,565	593	2.07%	5.41%
SAOPAULO	547,554	28,521	580	2.03%	5.31%
SYDNEY	538,378	28,741	596	2.07%	5.44%
WIDE	528,883	27,457	588	2.14%	5.30%

3.2.8 Provider with a «shadowing ROA»

What is the reason of such broad prefix coverage and what is the origin of these invalids? In order to answer this question we have to look more carefully at the AS path of the invalid BGP announcements.

In some cases, the BGP announcement is invalid because the origin AS of the announcement is not matching the one specified in the ROA, but we are able to find the correct AS number specified in the ROA on the AS path of the announcement, before the wrong origin AS.

This indicates that the up-stream provider registered a covering prefix in a ROA record, but did not create a ROA record for its customer's sub-allocation.

3.2.8.1 Example scenario

Let's see an example of this scenario:

- An ISP (`AS42`), in order to protect its own prefix `10.0.0.0/16`, registers a ROA with the following data: prefix `10.0.0.0/16`, maximum length 16, origin `AS42`.
- The ISP also allocates a sub-prefix of its address space, `10.0.1.0/24`, to a customer in `AS666`. The customer is peering with `AS42` and using it as transit provider.
- The customer announces its sub-prefix `10.0.1.0/24` from `AS666`.
- Another AS (or in our case, the BGP route collector), receives an announcement for `10.0.1.0/24` with AS path: `100 200 42 666`
  - The only longest-prefix match ROA existing for this prefix is the ROA registered by the ISP. For this reason, the customer's announcement is classified as «both maximum length and AS number» invalid, since the origin AS is wrong and the prefix is longer than the allowed length of the maximum length value specified in the ISP's ROA.
  - If the ISP did register the same ROA with maximum length equal to 24, then the customer's announcement would still be an «invalid origin AS number» invalid.
  - However, because the customer is using the ISP as transit, we can find the correct AS on the AS path of the customer's announcement.

3.2.8.2 Measuring «shadowed» prefixes

For this reason, finding the correct AS on the AS path of an invalid announcement is a strong clue that some upstream provider is making a customer's prefix invalid. An alternative hypothesis might be considering these invalid announcements

as hijacks. This is very unlikely because, since the announcement is going through the correct AS, it could be very easily filtered by standard prefix filters.

In order to measure these cases, we take each «invalid only» and «valid and invalid» prefix presented in section 3.2.4, and we divide them into three categories:

- **Maximum length only** invalids (therefore with correct origin AS)
- **Wrong origin AS** (and eventually also maximum length), but **correct AS on AS_PATH**: prefixes for which we verify that the correct AS specified in a ROA record for the prefix is present in at least one of the announced AS paths.
- **Other invalids**: invalids due to wrong origin AS, but the correct AS is not on the path.

Figure 3.2.8 shows the result of such classification.

By looking only at the last snapshot from August 2015, results are that 53% of invalids are due to «*maximum length only*», 24% are invalids with wrong origin AS but correct AS on path, and 23% are invalids due to wrong origin AS but correct AS not on the path.

This means that, if we exclude «maximum length only invalids», when we see a prefix coming from the wrong origin AS, in 50% of the cases we can find the correct AS in one of the AS paths of that prefix.

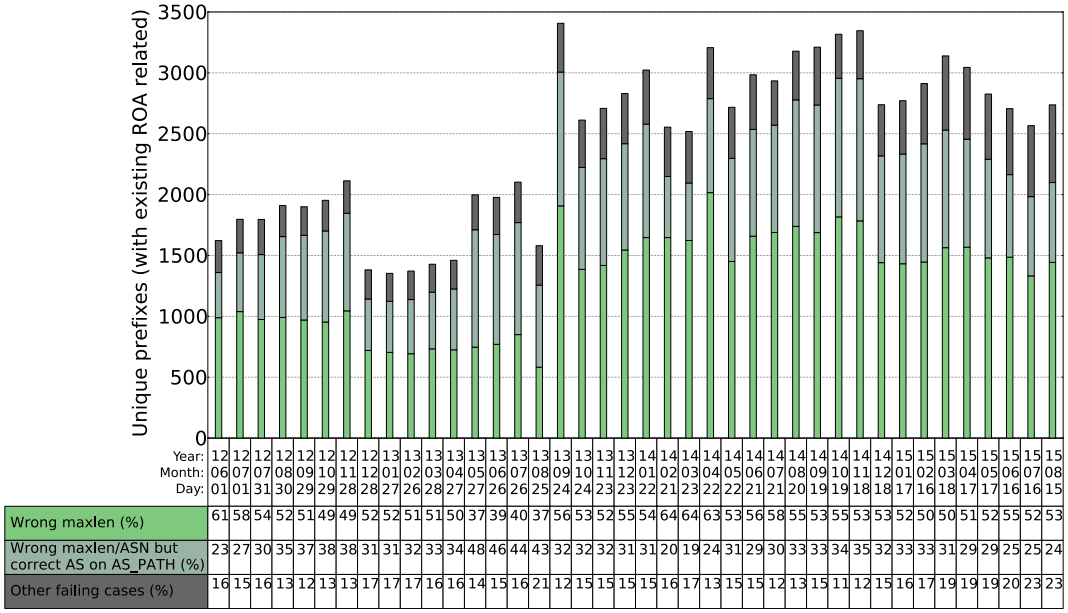

Figure 3.10: Number of invalid prefixes due to maximum length problem, invalids with wrong origin AS but correct AS on the AS_PATH and invalids for other reasons. Note: the numbers show in the table under the graph are percentages relative to the total number of «invalid only» and «valid and invalid» prefixes.

3.2.8.3 Missing announcements

Because the customer (`AS666` in the example of section 3.2.8.1), might be *multi-homed* with other transit providers, in some cases our BGP monitor collector may receive only some of the announcements sent by the *multi-homed* customer. This means that for some prefixes it is possible that the correct AS is present on the AS path, but our BGP collector is not receiving it. As result, it is possible that the 50% previously stated is a lower-bound on the percentage of this cases, which might be even more frequent than measured.

3.2.9 Maximum length problems

As we shown in the previous section 3.2.8, as well as in section 3.2.6 (about prefix coverage), there are a number of invalids which are «rescued» by another covering prefix. Moreover, we found that several of these invalids are the result of a «shadowing ROA» of a provider, making a customer invalid.

However, the maximum length alone is still the dominant reason of invalidity. By manually looking at some of these maximum length invalids in detail, and by thinking about other possible cases, we ended up with this list of possible reasons behind them:

- **DDoS protection**: It is common practice, during a DDoS attack to de-aggregate a `/24` or `/32` prefix in order to blackhole it on the border of the AS and avoid congesting the internal network.
- **iBGP to eBGP prefixes**: An ISP may have a single public AS number, while internally it is using BGP with private AS numbers for providing connectivity to some customers. Normally the announcements of the customer don't need to exit the ISP's network, but in some cases (traffic engineering, ddos, ...), the ISP may allow these more specific prefixes to be announced outside.
- **Backup links** or **multi-homed**: An AS is receiving a lot of traffic on a given address, so it decides to de-aggregate a `/24` and announce it over a link going through a different upstream provider.
- **Experiments** by researchers or by network operators testing RPKI by creating bogus ROAs.
- **Missing knowledge about RPKI validation**: Some operators experimenting RPKI, may not have clear the validation method of RPKI or they may be confused because of its difference with commonly-used prefix filter lists.

3.2.10 Conclusions

Even though the «shadowing provider» case is quite common, the «maximum length only» invalids alone are still the overwhelming invalid cause, and could be easily fixed by submission of correct ROA records by organizations. Most invalids today are probably a result from operators learning a new technology and have not yet developed good procedures. By monitoring the validity of their prefixes they should be able to learn from their mistakes and fix them. RIRs and researchers could also publish these problems and notify those who should fix them.

Moreover, even though there are a number of invalid prefixes, they are decreasing over time, most of them are «rescued» by other prefixes covering them and those which are not covered are mostly announced by few ASes.

3.3 Analysis on the quality of ROAs

In the previous section we shown the trends in the RPKI from the point of view of the BGP announcements. Here instead, we would like to do the same analysis from the point of view of the ROAs in the repositories.

Are there a lot of ROAs causing a lot of invalids? Are there just few ROAs causing a lot of invalids? Why are there ROAs causing announcements to become invalid? These are some of the questions that we try to answer in this section.

Note that in the following we will use the word «**ROA**» or «**ROAs**» for referring to what is usually called «ROA record»: that is a single *(IP prefix, Maximum length, Origin AS)* record in a «.roa» file.

3.3.1 Problematic ROAs classification

In order to perform this analysis, we considered again a single BGP RIB dump from 15 August 2015 and we extracted all ROAs used in the validation of each announcement of the RIB dump. Then, we built a database mapping each ROA record to the list of all BGP announcements making use of that specific ROA.

Note that due to our analysis method, we decided to count only unique ROA records. This means that if there are one or more «.roa» files containing the same *(IP prefix, Maximum length, Origin AS)*, we will count the record only once.

As result of this process, we divided ROAs into four categories: «satisfied ROAs», «questionable ROAs», «problem ROAs» and «other problem ROAs»:

- **Satisfied ROAs**: All the ROAs used only in BGP announcements passing validation, thus working correctly and not causing any problem.
- **Questionable ROAs**: ROAs used in some BGP announcements passing validation correctly, but also in BGP announcement failing validation, thus par-

tially causing problems. We list here ROAs where at least one invalid announcement is due to one of the following reasons:

  - A ROA is covering the announcement, but the ***maximum length*** field of the ROA it's incorrectly set. The case represents with high probability an error by the operator in creating the correct ROA, or *de-aggregating* something without checking what is registered in the ROA.
  - The origin AS of the announcement doesn't match the AS in the ROA, but the correct AS (listed in the ROA) is **present on the AS path**. This case represents with high probability a provider who registered a ROA for its prefix, but forgot to create a ROA for each customer who is announcing part of the provider's address space. In is the ***shadowing ROA*** discussed in section 3.2.8.

- **Problem ROAs**: Are ROAs used only in BGP announcements failing validation, thus not working correctly and causing prefixes to become unreachable. We list here ROAs where at least one of the failing announcements is failing due to one of the two reasons already listed for «Questionable ROAs».

- **Other problem ROAs**: ROAs that are causing problems, but we are not able to classify them because none of the announcements related is matching one of the two cases described above: *maximum length* or *shadowing ROA* problem. They are causing announcements to fail, but we are not sure if they are wrong.

  - For example, if the origin AS of all announcements related to a given ROA doesn't match the AS in the ROA, but the correct AS is **not** on the AS path, we can't deduce surely the cause of the problem.

3.3.2 Analysis results

In table 3.6 we present the result of this classification for the 15 August 2015 RIB dump from LINX monitor. Here we list the total number of unique ROA records registered in each repository and the various categories of problems described above.

For each of the four categories, we also count the number of unique autonomous system numbers seen among ROAs of each category. This is useful in order to understand how many problematic ROAs are the result of a registration by the same organization.

Moreover, we also present the number of «Unused ROAs», that is the number of ROA records registered in a repository, but never used during the validation process.

By looking at this data we can see how, in general, the RPKI deployment is going well. These results confirms the trends that we saw in the previous sections.

A significant percentage of unused ROAs can be observed, probably as result of many operators doing experiments with the deployment of RPKI.

ARIN have very few prefixes registered (and also the smallest percentage of address space covered, as seen in section 3.1.1) and it looks to be the one with the highest percentage of problematic ROAs.

For several repositories we can observe how many problematic ROAs are in fact the result of the same AS with multiple invalid ROAs, but at the same time we can't spot a single «evil AS» generating all the problematic ROAs.

Table 3.6: Quality of ROA records registered in the 5 RIRs, computed with data from Route Views LINX monitor of 15 August 2015. The categories shown are described in section 3.3.1. Note: the numbers shown here are relative to number of *unique* ROA *records*.

	AfriNIC	**ARIN**	**APNIC**	**LACNIC**	**RIPE NCC**
Total unique ROA (records)	113	472	1'497	1'675	14'010
Satisfied ROAs	94 (83.18%)	369 (78.17%)	1234 (82.43%)	1'300 (77.61%)	11'781 (84.08%)
Satisfied ROAs' unique ASes	20	120	175	229	2243
Questionable ROAs	6 (5.30%)	25 (6.77%)	37 (2.47%)	82 (4.89%)	221 (1.87%)
Questionable ROAs' unique ASes	3	8	19	26	104
Questionable ROAs shadowing invalids	5	21	1	41	88
Problem ROAs	0 (0.00%)	22 (5.96%)	18 (1.20%)	22 (1.31%)	83 (0.70%)
Problem ROAs' unique ASes	0	14	14	16	51
Problem ROAs shadowing invalids	0	16	6	6	48
Other problem ROAs	2 (1.76%)	20 (5.42%)	16 (1.06%)	54 (3.22%)	223 (1.89%)
Other problem ROAs' unique ASes	2	13	12	39	120
Unused ROAs	11 (9.73%)	36 (7.62%)	192 (12.82%)	396 (23.64%)	1702 (12.14%)

3.3.3 Other problem ROAs

We have been manually looking into all 315 «Other problem ROAs» cases from all RIRs listed in table 3.6, in order to deduce the source of these problematic ROAs. Our most significant findings are:

- 52 ROAs are registered and announced (incorrectly) from two different AS numbers, but the descriptive field associated to these ASes (in the *WHOIS* query result) is very similar or equal. This may be the result of several situations, mainly:
  - Use of the wrong AS in the ROA registration process
  - Announce of the same prefix from two different ASes for anycast purposes, but forgetting to register a ROA for the correct AS.
- 4 ROAs containing a private AS number in the ROA or one of the invalid announcements
- About 5 ROAs are very likely to be the result of ongoing research experiments on the propagation of invalid prefixes
- 2 ROAs are part of RIPE NCC RPKI beacons
- 18 ROAs are the result of IPv4 address space transfers, that we will discuss in the next section 3.3.4.
- A well-known Austrian hosting provider registered about 40 ROAs using its own AS number as the only authorized origin. All the prefixes registered in these ROAs are however announced by Level3, a well known transit provider. Level3 is probably originating these prefixes as a service for the hosting provider. This could be possible for example when an AS would like to act as provider for a geographically far AS, and want to use Level3 as wholesale connectivity provider for such peering.
  - We found 2 other ROAs similar to this case too, one involving *TATA Communications* and one involving *CenturyLink,* other well-known transit providers.

Other cases not listed here are very likely to be the result of a shadowing ROA (as seen in section 3.2.8) where:

- The originating AS is not using the *address space provider* as transit provider but the ROA's maximum length is correct.

- Another possibility is that there are announcements containing the *address space provider* on the AS path, but we are not able to receive such announcements from our BGP monitor because of specific propagation dynamics of those announcements.

Yet other cases may be the result of hijacks or, more likely, *address space transfers*.

3.3.4 IPv4 address space transfers

With the ongoing shortage of allocable IPv4 address space, some organizations started to sell and buy address space from others, creating a market for IPv4 addresses.

RIPE NCC is trying to keep tracks of these transfers by recording data about them [29]. In figure 3.11 we can see for each month from October 2012 to November 2015 the amount of IP prefixes (or address blocks) being transferred, and we can observe the increasing trend.

In [41, 42] it is discussed for example how some ASes in Romania sold address space to other ASes in Iran, but failed to stop announcing the sold address space after the sale. As result, the new owners had to announce more specific prefixes and registered ROAs for the prefixes bought.

These problems in transfers resulted in a number of invalid announcements that are classified as «Other problem ROAs» in our classification.

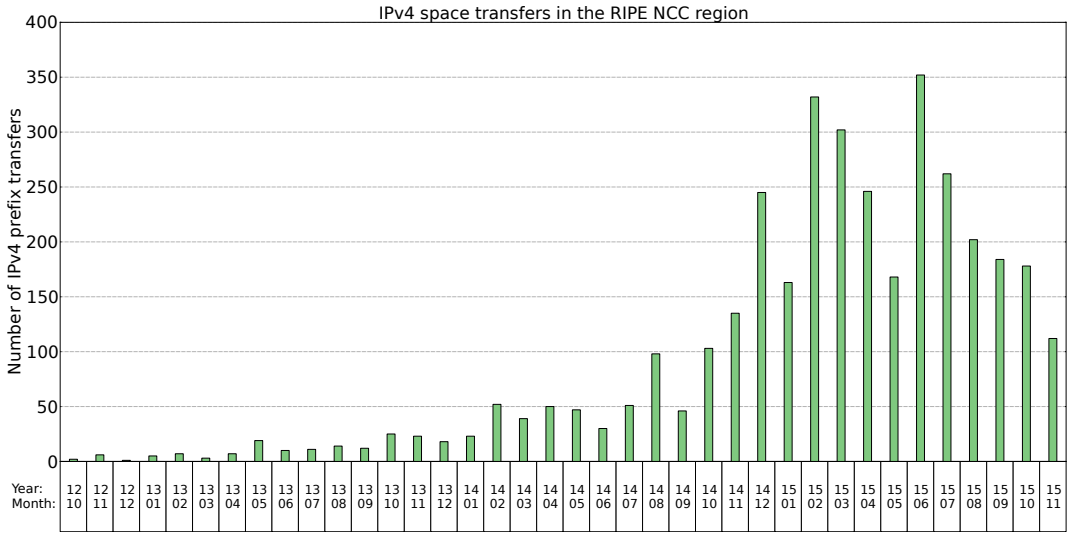

Figure 3.11: Number of address blocks transferred for each month from October 2012 to November 2015 in the RIPE NCC region. Data from [29].

In fact, we intersected RIPE NCC transfer data with the 233 «Other problem ROAs» from RIPE NCC and we found 18 ROAs that could be the result of address transfer effects.

RIPE NCC data about transfers is not reporting the AS numbers of the entities, but only the names of organizations who exchanged the address space. For this reason we had to manually check each transfer record and ROA. Moreover, we found that many transfers have been done through an intermediary organization. This makes even harder to match transfer data against ROA information, since we have no common information.

We don't have data about transfers in other regions, but looking at many of the other unclassified «Other problem ROAs», it is very likely that transfers are affecting many other ROAs in other regions too.

Chapter 4

Other measurements related to RPKI

4.1 Providing measurements to network operators

In order to help in cleaning RPKI from problematic ROAs, on February 2015 we built a website [23] providing a list of problematic ROAs, divided by RIR. We publicized the website on well known blogs [56] and mailing lists [57, 58].

The ROAs listed on the website are divided in «Questionable ROAs» and «Problem ROAs» in the same way as we explained in section 3.3.1, and for each problematic ROA there is a list of all valid and invalid BGP announcements related to such ROA. For each BGP announcement we also show detected problems, such as the shadowing ROA case or the wrong maximum length and suggest possible fixes.

On the top of the page, a visitor have the possibility to enter its own AS number and find any problematic ROA listed.

The website has been built using HTML5, Bootstrap and AngularJS [3], as discussed in the next section.

Figure 4.1 and 4.2 show some screenshots of the website.

According to our access logs (after excluding web crawlers, bots and others), the page has been viewed by nearly 400 unique IP addresses in the period from February to November 2015.

In figure 3.5 we not only see a decreasing percentage of invalid prefixes, but also a decreasing ***number*** of invalid prefixes. This means that some operators fixed problematic ROAs or announcements, and we believe that some of these fixed problems might have been the result of our website.

4.1.1 Structure of the website

The data shown on the website is produced as we did for our graphs and analysis of chapter 3. A BGP RIB dump from Route Views *LINX* monitor is validated using

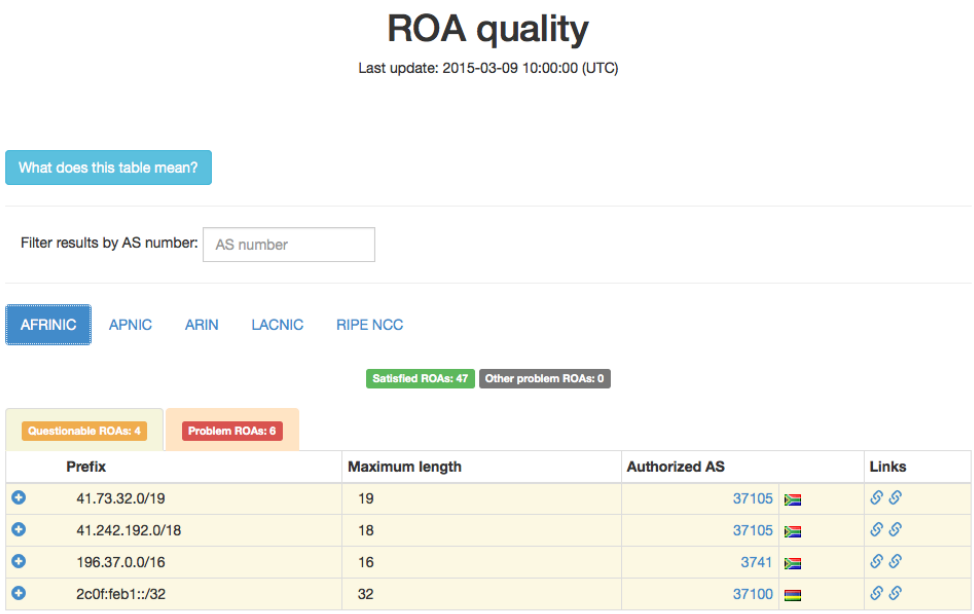

Figure 4.1: Screenshot from our website [23] reporting quality of ROAs in RPKI repositories.

the latest dump of RPKI repositories processed by *rcynic* [20]. The validated RIB dump is then converted into a Python dictionary mapping each ROA record to the list of all valid and invalid BGP announcements related to such ROA. ROAs are then split into the categories shown in section 3.3.1 and stored into a *JSON* file.

One important feature of our website was to show, not only the problematic ROAs, but also all BGP announcements related to them. The goal of this design choice was to be more precise and clear in presenting the problems caused by a ROA.

A common approach to do this could be building a dynamic web application that, on a query for a specific ROA, would return a list of announcements to present on the page. Due to time constraints in the development and limited resources on the server side, we decided to build a static web page.

A simple static HTML page, containing all ROAs (~500 ROAs) and all BGP announcements (~100'000) would however be very big in size, would render an excessive amount of data and thus use and excessive amount of memory on the client side.

By using AngularJS [3], we are able to let the client to download the compressed *JSON* file containing all ROAs and announcements (about 400KB when served compressed), and later render on the page only the list of ROAs of a specific registry (typically less than 300). Only when a specific ROA is selected, then the list of all announcement related to that ROA is loaded from the JSON file in memory and rendered on the page.

Everyday, a periodic job is called to fetch the latest RIB dump and generate a

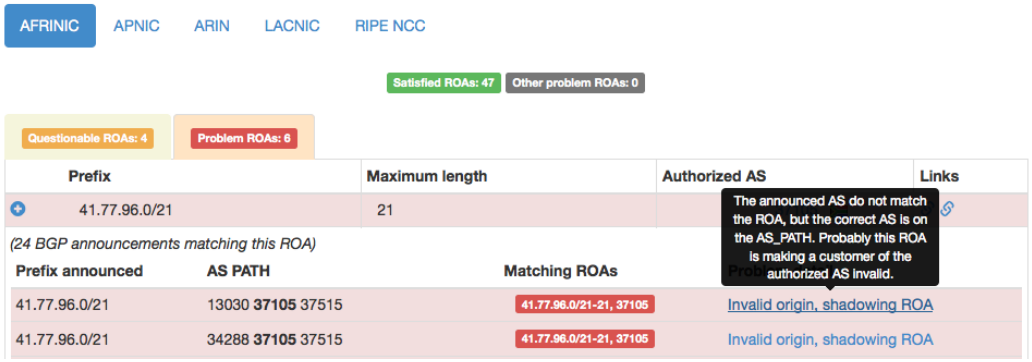

Figure 4.2: Screenshot from our website [23] , showing an example of a «Problem ROA» and its related BGP announcements.

new JSON file in order to keep the page updated.

4.2 Root and TLDs DNS nameservers measurements

The DNS infrastructure is famously know to be a critical service for the Internet. Over the years, the IETF created DNSSEC, a group of specifications with the purpose to protect certain kinds of information provided by DNS from being altered. DNSSEC [45] can provide origin authentication of DNS data, authenticated denial of existence, and data integrity, but not **availability** or confidentiality.

A routing incident involving one of the DNS root server's prefixes or one of the *Top-Level Domain* (TLD) name server's prefixes may cause a significant outage all over the Internet.

If a ROA is registered for these prefixes, RPKI-based origin validation could help in preventing accidental hijacks affecting them.

For this reason, we decided to measure the RPKI-coverage of DNS root servers and TLD name servers prefixes.

The following analysis is based on data from 20 November 2015.

4.2.1 DNS root servers

There are currently 13 DNS root servers worldwide, named with letters from *A* to *M*, managed by several different organizations, such as *Verisign, Inc.*, *RIPE NCC*, *Internet Systems Consortium, Inc.* and many others. Each one of these DNS root servers is represented by an IPv4 and an IPv6 addresses. Behind one of these addresses there is no single machine, but one or more clusters of them.

Each one of the addresses of these DNS root servers are part of prefixes anycasted on BGP by announcing them from different locations around the world, in order to allow extensive redundancy and low latency.

As result of our analysis we found that only one root server, **root server «K»** managed by *RIPE NCC*, is correctly covered by a ROA for both the IPv4 and IPv6 addresses.

Root server «I», managed by *Netnod* have a ROA correctly covering only the IPv6 address, while all other root servers have no ROA at all for any of their addresses.

One of the possible reasons preventing the deployment of RPKI on these prefixes could be that 6 out of 13 of these root servers (A,C,E,F,G,H) are currently using IPv4 addresses which are part of the *legacy address space* discussed in section 1.2.0.2. For this reason, the companies managing them might have legal issues in deploying RPKI, as discussed in section 2.4.5.1.

4.2.2 TLD name servers

As 20 November 2015, there are 1'108 TLDs registered [24] and 247 of them are *Country-Code Top-Level Domains* (ccTLDs, for example: it, us, fr, jp, ...).

In our analysis we found:

- For 256 TLDs there is at least one name server of the TLD with an address covered by a prefix correctly origin-validated with RPKI
- 27 (2%) TLDs have *all* nameservers covered by a ROA.
- «.com», «.net» and «.org», which together represent more than half of the TLDs used in the top 1 million *ALEXA's* website ranking, are still not covered. The TLDs «.com» and «.net» are currently maintained by *Verisign Inc.*, who is well known for holding on RPKI adoption due to doubts on its scalability [79].
- 166 ccTLDs have at least one name server covered, that is 67% of ccTLDs. These are the following: `ad, ae, al, am, ao, aq, ar, as, at, au, aw, az, ba, be, bf, bg, bi, bj, bm, bn, bt, bv, bw, by, cg, ch, ci, ck, cl, cm, cr, cu, cv, cw, cy, cz, de, dj, dk, dm, do, dz, ee, eg, er, es, et, eu, fi, fj, fo, fr, gb, gd, ge, gg, gh, gl, gm, gn, gp, gr, gs, gu, gw, gy, hk, hr, ht, hu, id, ie, il, im, ir, is, it, je, jm, jo, ke, kg, kh, km, kw, kz, la, lb, li, lk, lr, ls, lu, lv, ly, ma, mc, md, mg, mk, mm, mr, mt, mu, mv, mw, my, mz, nc, ne, ng, ni, nl, no, np, nr, nu, nz, pa, pe, ph, pl, pm, pn, ps, pt, pw, py, qa, re, rs, rw, sa, sc, sd, se, sg, si, sj, sm, sn, so, sv, sy, sz, td, tf, th, tj, tn, to, tr, tt, tw, tz, ua, ug, uy, va, ve, vg, vn, wf, yt, za, zm.` 59 out of these 166 are the result of ccTLDs using a RIPE NCC secondary DNS service [30], which is using addresses covered with RPKI ROAs.
- None of the addresses of any of the nameservers is covered by an invalid prefix. This is an interesting finding, showing how the invalids that we discussed before are not related to critical infrastructure such as protecting DNS servers.

Figure 4.3 shows the data just described.

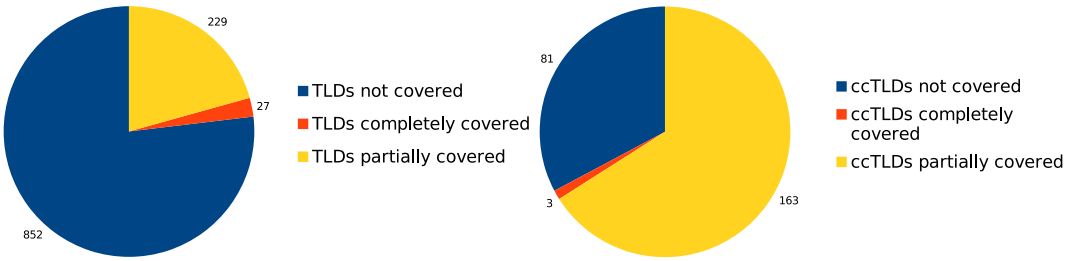

Figure 4.3: RPKI-coverage among all TLDs name servers and among ccTLDs only.

4.3 Related Work: RPKI coverage of websites

In the context of getting a complete view of the RPKI deployment, it is interesting to briefly report the results shown in [101], where it is discussed about RPKI-coverage of web server addresses.

Protecting prefixes containing web server addresses with RPKI is important in order to prevent websites outages caused by accidental hijacks. Moreover, RPKI may be useful to partially prevent malicious attacks against the SSL/PKI infrastructure shown in [48].

By taking the list of the top 1 million most popular websites from *Alexa* ranking, the authors of [101] found that only 6% of the web server prefixes are currently covered by RPKI. However, less popular content is more likely to be covered by RPKI, in fact when considering only the top 100'000, about 4% of them are covered. The paper also shows how CDN providers are likely the cause for the scarce deployment. In the results is also shown how only one out of 16 CDN providers, is *partially* using RPKI for few prefixes.

Conclusions

In this work we studied *interdomain* routing security and presented measurements on the deployment and quality of data in the RPKI repositories, towards the goal of deploying RPKI-based origin validation.

In the study of routing security we explained possible types of routing incidents, their consequences, we analyzed in detail 8 «historical» examples of them and we tried to understand common causes of accidental incidents.

We discussed existing solutions to face routing incidents and, in this context, we shown how RPKI-based origin validation is an important method to prevent accidental routing incidents at the moment, even though it is deploying slowly due to technical and non technical reasons.

We measured the historical registration deployment trends in RPKI repositories. We observed that Europe and Latin America are leading today, with many ROAs registered. Regarding the RIRs RPKI infrastructure, there were serious problems. The entire dataset became unavailable for extended periods of time for a couple of RIRs. We then quantified the state of origin-validation deployment by validating historical BGP data against historical RPKI repository data from the same period. We found that, currently, about 6% of the BGP prefixes announced on BGP are covered by a correctly registered ROA in the RPKI, with an increasing trend.

By considering IP prefix coverage, we found that even though there are a number of invalid prefixes (currently in a decreasing trend), most of them are «rescued» by another valid or «ROA not found» prefix covering them. This is an interesting result that shows that dropping invalid prefixes is probably less «dangerous» than what is often thought by some network operators.

Moreover, we analyzed types of invalid prefixes and tried to understand what is causing them. We found that several cases are probably due to operators who misunderstood the purpose of «maximum length field» in a ROA or forgot what they registered while announcing more specifics. However we also found that in many cases, invalids are due to a provider mis-registering a ROA which is making customer's prefixes invalid. These measurements highlights the need for operators to monitor the status of their prefixes with regard to what is registered in the RPKI. In addition, customers should make sure that their provider registers the prefixes

they have been allocated or should perform the registration themselves.

We then performed measurements from the point of view of ROA records and we confirmed how many ROAs are being mis-registered by upstream providers, as well as other cases such as address transfers.

We shown how we collected and published these results in order to solve these problems and we finally commented some RPKI measurements related to DNS and the web servers.

Future work might look into measuring invalid prefixes from multiple monitors with a higher resolution, in order to better understand registration trends and try detecting hijacks affecting prefixes correctly registered in RPKI repositories.

Bibliography

[1] http://www.irr.net/.

[2] ALTDB IRR website. http://altdb.net/. Accessed: 2015-10-15.

[3] AngularJs. https://angularjs.org/. Accessed: 2015-10-26.

[4] BGP Route Monitoring | ThousandEyes. https://www.thousandeyes.com/solutions/bgp. Accessed: 2015-10-26.

[5] BGPdump tool. https://bitbucket.org/ripencc/bgpdump/wiki/Home.

[6] BGPmon - Network Security group at Colorado State University. http://www.bgpmon.io/. Accessed: 2015-10-26.

[7] BGPMon website. https://www.bgpmon.net/. Accessed: 2015-10-04.

[8] Get Informed About Internet Outages - DynDNS. http://dyn.com/internet-alerts//. Accessed: 2015-10-26.

[9] Google Public DNS. https://developers.google.com/speed/public-dns/. Accessed: 2015-10-17.

[10] IPv4 Prefixes Delegated by AfriNIC.

[11] IPv4 Prefixes Delegated by APNIC.

[12] IPv4 Prefixes Delegated by ARIN.

[13] IPv4 Prefixes Delegated by LACNIC.

[14] IPv4 Prefixes Delegated by RIPE NCC.

[15] IRR ToolSet. http://irrtoolset.isc.org/.

[16] Matplotlib. http://matplotlib.org/. Accessed: 2015-11-26.

[17] North American Network Operators Group (NANOG). https://www.nanog.org/. Accessed: 2015-10-26.

[18] RADB IRR Archive. `ftp://ftp.radb.net/radb/dbase/archive/`.

[19] RADB IRR website. `http://www.radb.net/`. Accessed: 2015-10-15.

[20] rcynic RPKI validator. `http://rpki.net/wiki/doc/RPKI/RP/rcynic`.

[21] RIPE NCC - Atlas. `https://atlas.ripe.net/`. Accessed: 2015-10-15.

[22] RIPE NCC: Routing Information Service. `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris`. Accessed: 2015-10-04.

[23] RPKI.me. `http://www.rpki.me/`. Accessed: 2015-10-15.

[24] TLD list. `https://www.iana.org/domains/root/db`.

[25] University of Oregon Route Views Project. `http://www.routeviews.org/`. Accessed: 2015-10-04.

[26] Post in NANOG mailinglist: "Wow, AS7007!". `http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html`, 1997.

[27] Google Public DNS: 70 billion requests a day and counting. `https://googleblog.blogspot.jp/2012/02/google-public-dns-70-billion-requests.html`, 2012. Accessed: 2015-10-18.

[28] Summons of the RIPE NCC Against the State of the Netherlands. `https://www.ripe.net/publications/news/about-ripe-ncc-and-ripe/summons-of-the-ripe-ncc-against-the-state-of-the-netherlands`, 2012.

[29] IPv4 Transfer Statistics. `https://www.ripe.net/manage-ips-and-asns/resource-transfers-and-mergers/transfers/ipv4/ipv4-transfer-statistics`, 2015. Accessed: 2015-11-19.

[30] Secondary DNS Service for ccTLD Operators . `https://www.ripe.net/publications/docs/ripe-659`, 2015. Accessed: 2015-11-19.

[31] C Alaettinoglu et al. Routing Policy Specification Language (RPSL). RFC 2622, RFC Editor, 1999.

[32] L Blunk et al. Routing Policy Specification Language next generation (RPSLng). RFC 4012, RFC Editor, 2005.

[33] L Blunk et al. Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format. RFC 6396, RFC Editor, 2011.

[34] Martin Brown. Pakistan hijacks YouTube. `http://research.dyn.com/2008/02/pakistan-hijacks-youtube-1/`, 2008.

[35] R Bush. Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI). RFC 7115, RFC Editor, 2014.

[36] R. Bush. RPKI Local Trust Anchor Use Cases. Internet-Draft draft-ietf-sidr-lta-use-cases-03.txt, IETF Secretariat, June 2015.

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

[38] R Chandra, P Traina, and T Li. BGP Communities Attribute. RFC 1997, RFC Editor, 1996.

[39] Danny Cooper, Ethan Heilman, Kyle Brogle, Leonid Reyzin, and Sharon Goldberg. On the risk of misbehaving rpki authorities. In *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, page 16. ACM, 2013.

[40] Jim Cowie. Internet Touches Half Million Routes: Outages Possible Next Week. `http://research.dyn.com/2014/08/internet-512k-global-routes/`, 2014.

[41] Jim Cowie and Doug Madory. Change of Address: Routing Issues of Transferred IPv4 Addresses. `https://ripe70.ripe.net/wp-content/uploads/presentations/61-27-Change_of_Address_Cowie-1.pdf`, 2015.

[42] Jim Cowie and Doug Madory. IPv4 Address Market Takes Off. `http://research.dyn.com/2015/04/ipv4-address-market-takes-off/`, 2015.

[43] L Daigle. WHOIS Protocol Specification. RFC 3912, RFC Editor, 2004.

[44] Marco d'Itri. BGP security at internet exchanges: A practical experiment. `https://ripe71.ripe.net/presentations/33-bgp-experiment-ripe71.pdf`, November 2015.

[45] D Eastlake. Domain Name System Security Extensions. RFC 2535, RFC Editor, 1999.

[46] Michael Fincham. RPKI, NZNOG 2014. `https://hotplate.co.nz/archive/nznog/2014/rpki/`. Accessed: 2015-11-19.

[47] International Organization for Standardization. Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473). 2002.

[48] A Gavrichenkov. Breaking https with bgp hijacking. BlackHat, 2015.

[49] Wes George. Adventures in RPKI (non) deployment. `https://www.nanog.org/sites/default/files/wednesday_george_adventuresinrpki_62.9.pdf`, October 2014.

[50] Sharon Goldberg. Why is it taking so long to secure Internet routing? *Communications of the ACM*, 57(10):56–63, 2014.

[51] Sharon Goldberg, Michael Schapira, Peter Hummon, and Jennifer Rexford. How secure are secure interdomain routing protocols. *ACM SIGCOMM Computer Communication Review*, 41(4):87–98, 2011.

[52] Government of Pakistan. Website blocking order. `http://research.dyn.com/content/uploads/blog/pakistan_blocking_order.pdf`. Accessed: 2015-10-04.

[53] J Hawkinson and T Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930, RFC Editor, 1996.

[54] Robert Hinden and Alan Sheltzer. Gateway-to-Gateway Protocol (GGP). RFC 823, RFC Editor, 1982.

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

[56] Daniele Iamartino. Quality of ROAs in RPKI Repositories. `https://labs.ripe.net/Members/daniele_iamartino/quality-of-roas-in-rpki-repositories`, March 2015. Accessed: 2015-10-26.

[57] Daniele Iamartino. Quality of ROAs in RPKI repositories. `https://mailman.nanog.org/pipermail/nanog/2015-February/073584.html`, February 2015. Accessed: 2015-10-26.

[58] Daniele Iamartino. Quality of ROAs in RPKI repositories. `https://www.ripe.net/ripe/mail/archives/routing-wg/2015-February/002954.html`, February 2015. Accessed: 2015-10-26.

[59] Daniele Iamartino and Randy Bush. Urgent problem in AfriNIC RPKI repository. `https://afnog.org/pipermail/afnog/2015-March/002085.html`, 2015. Accessed: 2015-11-19.

[60] Daniele Iamartino, Cristel Pelsser, and Randy Bush. Study of BGP Route Origin Registration and Validation. `https://ripe69.ripe.net/`

presentations/103-route-origin-validation.pdf, 2014. Accessed: 2015-11-19.

[61] Daniele Iamartino, Cristel Pelsser, and Randy Bush. Measuring bgp route origin registration and validation. In *Passive and Active Measurement*, pages 28–40. Springer, 2015.

[62] Akmal Khan, Hyun-chul Kim, Taekyoung Kwon, and Yanghee Choi. A comparative Study on IP Prefixes and their Origin ASes in BGP and the IRR. *ACM SIGCOMM Computer Communication Review*, 43(3):16–24, 2013.

[63] Jac Kloots. SURFnet. http://rpki.surfnet.nl/. Accessed: 2015-11-19.

[64] Jac Kloots. RPKI Routing policy decision-making, a SURFnet perspective. https://blog.surf.nl/en/rpki-routing-policy-decision-making-a-surfnet-perspective/, 2014.

[65] W Kumari and K Sriram. Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP. RFC 6472, RFC Editor, 2011.

[66] Xavier Le Bris. Status of Legacy IPv4 Address Space. https://labs.ripe.net/Members/xavier/status-of-legacy-ipv4-address-space, September 2011.

[67] Robert Lemos. Internet routers hitting 512K limit, some become unreliable. http://arstechnica.com/security/2014/08/internet-routers-hitting-512k-limit-some-become-unreliable/, 2014.

[68] M Lepinski et al. An Infrastructure to Support Secure Internet Routing. RFC 6480, RFC Editor, 2012.

[69] Pat Litke and Joe Stewart. BGP Hijacking for Cryptocurrency Profit. http://www.secureworks.com/cyber-threat-intelligence/threats/bgp-hijacking-for-cryptocurrency-profit/, 2014.

[70] K Lougheed and Y Rekhter. A Border Gateway Protocol (BGP). RFC 1105, RFC Editor, 1989.

[71] Doug Madory. Global Collateral Damage of TMnet leak. http://research.dyn.com/2015/06/global-collateral-damage-of-tmnet-leak/, 2014.

[72] Doug Madory. Use Protection if Peering Promiscuously. http://research.dyn.com/2014/11/use-protection-if-peering-promiscuously/, 2014.

[73] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding bgp misconfiguration. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 3–16. ACM, 2002.

[74] G Malkin. RIP Version 2. RFC 2453, RFC Editor, 1998.

[75] D Meyer et al. Using RPSL in Practice. RFC 2650, RFC Editor, 1999.

[76] D.L. Mills. Exterior Gateway Protocol (EGP). RFC 904, RFC Editor, 1984.

[77] J Moy. OSPF Version 2. RFC 2328, RFC Editor, 1998.

[78] M. L. Mueller. In important case, RIPE NCC seeks legal clarity on how it responds to foreign court orders. `http://www.internetgovernance.org/?p=504`, 2011. Accessed: 2015-10-26.

[79] Eric Osterweil, Terry Manderson, Russ White, and Danny McPherson. Sizing estimates for a fully deployed rpki. Technical Report 1120005 version 2, 2012.

[80] T Paseka. Why Google went offline today and a bit about how the internet works. `https://blog.cloudflare.com/why-google-went-offline-today-and-a-bit-about/`, 2012.

[81] A Pilosov and T Kapela. Stealing the Internet: An Internet-scale man in the middle attack. `https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf`, 2008.

[82] Dave Piscitello. Guidance for preparing domain name orders, seizures & takedowns. *Thought paper. ICANN (Mar. 2012)*, 2012.

[83] Jon Postel et al. Internet protocol. RFC 791, RFC Editor, 1981.

[84] Anirudh Ramachandran and Nick Feamster. Understanding the network-level behavior of spammers. *ACM SIGCOMM Computer Communication Review*, 36(4):291–302, 2006.

[85] Y Rekhter et al. A Border Gateway Protocol 4 (BGP-4). RFC 1771, RFC Editor, 1995.

[86] Y Rekhter et al. Address Allocation for Private Internets. RFC 1918, RFC Editor, 1996.

[87] Y Rekhter et al. A Border Gateway Protocol 4 (BGP-4). RFC 4271, RFC Editor, 2006.

[88] Andreas Reuter, Matthias Wählisch, and Thomas C Schmidt. RPKI MIRO: Monitoring and Inspection of RPKI Objects. In *Proceedings of the 2015 ACM*

Conference on Special Interest Group on Data Communication, pages 107–108. ACM, 2015.

[89] Philipp Richter, Mark Allman, Randy Bush, and Vern Paxson. A Primer on IPv4 Scarcity. *ACM SIGCOMM Computer Communication Review*, 45(2):21–31, 2015.

[90] Job Snijders. Golden Prefixes. `https://ripe69.ripe.net/presentations/46-jobsnijders_ripe69_golden_prefixes.pdf`, November 2014.

[91] K Sriram, D Montgomery, B Dickson, K Patel, and A Robachevsky. Methods for Detection and Mitigation of BGP Route Leaks. Internet-Draft draft-ietf-idr-route-leak-detection-mitigation-01.txt, IETF Secretariat, October 2015.

[92] Andree Toonk. How accurate are the Internet Route Registries (IRR). `https://www.bgpmon.net/how-accurate-are-the-internet-route-registries-irr/`, 2009.

[93] Andree Toonk. Hijack event today by Indosat. `https://www.bgpmon.net/hijack-event-today-by-indosat/`, 2014.

[94] Andree Toonk. Massive route leak cause Internet slowdown. `https://www.bgpmon.net/massive-route-leak-cause-internet-slowdown/`, 2014.

[95] Andree Toonk. The Canadian Bitcoin Hijack. `https://www.bgpmon.net/the-canadian-bitcoin-hijack/`, 2014.

[96] Andree Toonk. Using BGP data to find spammers. `https://www.bgpmon.net/using-bgp-data-to-find-spammers/`, 2014.

[97] Andree Toonk. BGP Optimizer Causes Thousands Of Fake Routes. `https://www.bgpmon.net/bgp-optimizer-causes-thousands-of-fake-routes/`, 2015.

[98] Matthias Wählisch. One Day in the Life of RPKI . `https://labs.ripe.net/Members/waehlisch/one-day-in-the-life-of-rpki`, 2011. Accessed: 2015-11-19.

[99] Matthias Wählisch. Preliminary Results of Survey about RPKI/DNSSEC. `https://tools.ietf.org/agenda/92/slides/slides-92-sidr-4.pdf`, 2015. Accessed: 2015-11-19.

[100] Matthias Wählisch, Olaf Maennel, and Thomas C Schmidt. Towards detecting bgp route hijacking using the rpki. *ACM SIGCOMM Computer Communication Review*, 42(4):103–104, 2012.

[101] Matthias Wählisch, Robert Schmidt, Thomas C Schmidt, Olaf Maennel, Steve Uhlig, and Gareth Tyson. RiPKI: The Tragic Story of RPKI Deployment in the Web Ecosystem. 2015.