# Improving Internet Routing Security: From Origin Validation to Path Validation

Nils Miro Rodday

ISBN (print): 978-90-365-6100-6
ISBN (digital): 978-90-365-6101-3
DOI: 10.3990/1.9789036561013
URL: https://doi.org/10.3990/1.9789036561013

Typeset with LaTeX. Cover design and print by Gildeprint, The Netherlands.

# IMPROVING INTERNET ROUTING SECURITY: FROM ORIGIN VALIDATION TO PATH VALIDATION

DISSERTATION

to obtain
the degree of Doctor rerum naturalium (Dr. rer. nat.)
at the Universität der Bundeswehr München
on the authority of the president of the Universität der Bundeswehr München,
Prof. Dr. mont. Dr.-Ing. habil. Eva-Maria Kern,
and the degree of Doctor at the University of Twente,
on the authority of the rector magnificus of the University of Twente,
prof. dr. ir. A. Veldkamp,
on account of the decision of the graduation committee,
submitted on Friday, January 19, 2024,
accepted at the faculty of computer science at the Universität der Bundeswehr
München on Wednesday, March 6, 2024,
publicly defended on Thursday, March 21, 2024 at 14:00

by

Nils Miro Rodday

born on March 12, 1989
in Tönisvorst, Germany.

This dissertation has been approved by:

**Supervisors:**
Prof. Dr. rer. nat. G. Dreo Rodošek
Prof. dr. ir. A. Pras

**Co-supervisor:**
Prof. dr. ir. R.M. van Rijswijk-Deij

*To my grandma.*

# Acknowledgments

Many people have contributed to the success of my PhD and I am thankful to each and everyone!

I am very grateful to have such a supportive family that continuously encouraged me throughout the past six years. Thanks, Mom and Dad, for setting me on the right track and believing in me. I know that it cannot have been easy. To my siblings, Lara and Lars, thanks for supporting me and being with me all along. I am able to rely on you whenever necessary, and I hope you feel the same about me. Dear Grandma, I promised you that I would finish, and I did. I hope you are well, wherever you are.

I am particularly thankful to my supervisors at the two universities. I was lucky enough to have you, Aiko, as a supervisor of my master's thesis. Thanks for leaving the door open after I decided to go into industry and reminding me of the opportunity to pursue a PhD when I was looking for a change. Moreover, thanks for supporting me even though you were already enjoying retirement. Gabi, I appreciate you for providing a space in Munich where I could explore the things I am interested in and giving me the flexibility I needed. You managed to push me to a level that enabled me to finish this work. I have overcome the often-cited Valley of Tears and your guidance has been much appreciated. Roland, although we had little interaction at the beginning of my PhD, I mainly relied on you for feedback and content-related questions during the past two years. I am very thankful for your support and hope to be able to catch up to your quick reasoning at some point.

I am grateful for the opportunity to have supervised many talented students over the years. Lukas Kaltenbach, Paul Friedemann, Kai Hamich and Nils Höger were particularly important for this work with their contributions.

Many thanks to my fellow PhD students in Munich and Enschede for the most valuable exchange and sharing of experiences. Klement, I would not have survived the past six years without our regular beer & pizza evenings out. You are a very good friend! Raphael, you truly impressed me with your ability to achieve your goals and find solutions in a timely manner under challenging circumstances. Your logical thinking inspires me. The three of us had the same mindset, and I am glad we all made it in the end! Thanks to all my colleagues at the two chairs: Flo, Sinclair, Alex, Tobi, Julius, Daniel, Christian, Sigi, Leandro, Bernd, Wouter, Ramin, Moritz, Luuk, and Mattijs. Ricardo, in memoriam, you provided me with the opportunity to present my master's thesis work at my first scientific conference and contributed to paving the way for my future career. Thank you!

While looking for a topic during my first two years, I was lucky to contact you, Matthias. We worked on many problems throughout the past years, and I had the privilege of learning a lot from you. Thanks for allowing me to work with you and your team, pushing me when necessary, and teaching me the skills to perform exact and accurate measurements. Without you, I would still be looking for a topic!

Running measurements is only possible when the required infrastructure is readily available. Randy, I appreciate your support in providing the required prefix space to run large-scale measurement campaigns. Thanks to Ethan and especially Italo for allowing me to use the PEERING testbed for our experiments and troubleshooting for countless hours. Thanks to the RIPE Atlas team for their technical support and the Atlas community for providing many millions of credits to conduct my research.

Finally, thanks to everyone who enriched my life throughout these past six years outside of the university and family circles. Clemens & Jitka, I had a great time hiking numerous trails with you. And our nights out weren't that bad either! Christian, thank you for the many insights into the world of vertical farming. I enjoyed developing technical solutions that were immediately put to the test. Moreover, while trying to distract myself from working on the PhD, I had the privilege to work as a paramedic on many ambulance shifts with excellent colleagues. I thoroughly enjoyed spending my time with you and learning from the best!

# Abstract

The Internet has become an integral part of all of our lives. Many people, especially younger generations, cannot even imagine a world without it as it is present in every aspect of their lives. Hence, it is imperative that the Internet is reliable and secure.

While many think that the Internet is one coherent whole, it is indeed a network of networks. Each network is called an Autonomous System (AS) and has the freedom to decide on the preferred protocols and technologies within its boundaries. To communicate among each other, the one protocol and de-facto standard used is called Border Gateway Protocol (BGP). It facilitates the exchange of Internet Protocol (IP) address information and allows ASes to know to which destination the traffic has to be routed. It is equivalent to the exchange of zip codes among cities to know where packets have to go.

The Internet developed from a small research project between three universities where trust was taken for granted to a worldwide and complex infrastructure. With more than 74,000 participants, we can no longer rely on each other's good intent. The protocol, however, has largely remained the same regarding security. Therefore, several attack vectors continue to exist that allow for the redirection of traffic which in turn allows for its manipulation or the unavailability of entire parts of the Internet.

Throughout the past decade, attempts have been made by the Internet Engineering Task Force to secure inter-domain routing. Origin validation provides a way for ASes to check whether the sender of a received BGP announcement is allowed to do so. The Resource Public Key Infrastructure (RPKI) implements origin validation by binding IP prefixes to AS numbers and is under deployment since 2011.

More recently, path validation algorithms have attempted to secure not only the legitimacy of the BGP announcement's origin but also the entire path the announcement travelled. The advantage of such technology is that the receiving party can infer whether an announcement was manipulated on the way and take appropriate actions. There is, however, uncertainty whether building on the existing RPKI infrastructure is the best option and which path validation algorithms under which constraints should receive the most attention.

Therefore, the goal of this thesis is to *assess whether we can build on top of the RPKI with algorithms securing the AS path to improve overall routing security.* To achieve our goal and improve Internet routing security we perform large-scale measurements and simulations on the deployment of origin validation, default routes, and path validation algorithms.

Our *first contribution* summarizes existing related work and identifies gaps. We categorize the significant publications in the field and show where more work is needed. Research within the realm of RPKI is divided into RPKI Route Origin Authorization (ROA) measurements, RPKI Route Origin Validation (ROV) measurements, and RPKI resilience. We find that existing approaches fail to correctly identify RPKI ROV filtering ASes and, therefore, include false positives that lead to a skewed adoption rate of RPKI-protected ASes.

In our *second contribution*, we develop an improved RPKI ROV identification methodology by relying on extensive data plane measurements. We develop a strict approach that reduces false positives and makes inferences more accurate. Moreover, we compare seven Relying Party (RP) software implementations and recommend Routinator as the best solution there currently is for operators. While we are able to increase accuracy in our RPKI ROV measurements, we noticed measurement artifacts that could potentially be attributed to default routes.

In our *third contribution*, we turn to default routes and improve upon two existing methodologies. We implement these measurements as continuous measurements and present our results on a website. Our newly derived datasets allow for the exclusion of ASes with default routes installed from RPKI ROV measurements and, therefore, further improve the results. Moreover, we develop a way to identify middleboxes, which helps sanitize measurement data even further. While working on origin validation techniques such as the RPKI, we understand that the Internet could be more secure only by deploying additional security mechanisms to secure the path of the BGP announcement.

In our *fourth contribution*, we focus on path validation algorithms and how they can be built on top of the existing RPKI infrastructure to provide a high level of additional security while maintaining a high likelihood of adoption. Previous attempts wanted maximum security at the expense of usability and never succeeded. We perform simulations for two algorithms called Autonomous System Provider Authorization (ASPA) and AS-Cones with different deployment scenarios to tell which path validation algorithm offers the most security benefits at minimal operational cost. ASPA allows for more security compared to AS-Cones and for both algorithms, adoption is only required in a few ASes to provide an overall benefit for the whole inter-domain routing infrastructure. In addition, we develop a BGP topology generator that allows the emulation of arbitrary topologies within the NIST BGP-SRx software suite.

In summary, we improve Internet routing security through these contributions by adding path validation to the existing origin validation deployments. Our findings have been made publicly available, datasets are open-sourced, and source code has been published on publicly available repositories.

# Zusammenfassung

Das Internet ist zu einem festen Bestandteil unseres Lebens geworden. Viele Menschen, insbesondere jüngeren Generationen, können sich eine Welt ohne das Internet gar nicht mehr vorstellen. Es ist in jedem Aspekt des Lebens vertreten. Daher ist es unerlässlich, dass das Internet zuverlässig und sicher ist.

Während viele denken, das Internet sei ein großes Ganzes, ist es in Wirklichkeit ein Netz von Netzen. Jedes Netz wird als Autonomes System (AS) bezeichnet und hat die Freiheit über die verwendeten Protokolle und Technologien zu entscheiden. Für die Kommunikation untereinander wird als Protokoll und De-facto-Standard das Border Gateway Protocol (BGP) verwendet. Es ermöglicht den Austausch von Internet-Protokoll (IP)-Adressen und die Weiterleitung von Daten an das richtige Ziel, vergleichbar mit dem Austausch von Postleitzahlen zwischen Städten, um die Zustellung von Paketen zu gewährleisten.

Das Internet entwickelte sich aus einem kleinen Forschungsprojekt zwischen drei Universitäten bei dem Vertrauen als selbstverständlich vorausgesetzt wurde, zu einer weltweiten, komplexen Infrastruktur. Mit mehr als 74.000 Teilnehmern ist Vertrauen nunmehr schwerlich gegeben. Die Sicherheit von BGP hat sich jedoch nicht verändert. Deshalb gibt es weiterhin mehrere Angriffsvektoren, die eine Umleitung des Datenverkehrs ermöglichen, welche zur Manipulation des Datenverkehrs oder der Nichtverfügbarkeit ganzer Teile des Internets führen.

In den letzten zehn Jahren wurde verstärkt durch die Internet Engineering Task Force versucht das Inter-Domain Routing zu sichern. Die Ursprungsvalidierung bietet eine Möglichkeit für Autonome Systeme zu überprüfen, ob der Absender einer empfangenen BGP-Ankündigung dazu berechtigt ist. Die Resource Public Key Infrastructure (RPKI) implementiert die Ursprungsvalidierung seit 2011 durch Bindung von IP-Präfixen an AS-Nummern. Zuletzt wurde mit Algorithmen zur Pfadvalidierung versucht den gesamten Pfad der Ankündigung zu sichern. Der Vorteil einer solchen Technologie besteht darin, dass die empfangende Partei erkennen kann, ob eine Ankündigung auf dem Weg manipuliert wurde, und entsprechende Maßnahmen ergreifen kann. Es besteht jedoch Unsicherheit darüber, ob der Aufbau auf der bestehenden RPKI Infrastruktur die beste Option ist und welche Pfadvalidierungsalgorithmen unter welchen Bedingungen die meiste Aufmerksamkeit erhalten sollten.

Ziel dieser Arbeit ist es daher, *zu prüfen, ob wir auf der RPKI mit Pfadvalidierungsalgorithmen aufbauen können um die Routing-Sicherheit insgesamt zu verbessern.* Zur Erreichung des Ziels führen wir umfangreiche Messungen und Simulationen zum Einsatz von Ursprungsvalidierungsalgorithmen, Standardrouten und Pfadvalidierungsalgorithmen durch.

In unserem *ersten Beitrag* fassen wir die bestehenden Arbeiten zusammen und zeigen Lücken auf. Wir kategorisieren die wichtigsten Veröffentlichungen auf diesem Gebiet und zeigen Forschungsbedarf auf. Bestehende Forschung auf dem Gebiet der RPKI ist unterteilt in RPKI Route Origin Authorization (ROA) Messungen, RPKI Route Origin Validation (ROV) Messungen, und RPKI-Resilienz. Da aktuelle Ansätze nicht in der Lage sind RPKI ROV filternde ASes korrekt zu identifizieren und daher False-Positives enthalten, sind die angegebenen RPKI Adoptionsraten zu hoch.

In unserem *zweiten Beitrag* entwickeln wir eine verbesserte RPKI ROV Identifikationsmethode, die sich auf umfangreiche Messungen mit RIPE Atlas stützt. Wir entwickeln einen regiden Ansatz, der False-Positives reduziert und genauere Ergebnisse zulässt. Außerdem vergleichen wir sieben Implementierungen von Relying Party (RP) Software und empfehlen Routinator als die beste Lösung, die es derzeit für AS-Administratoren gibt. Während wir in der Lage sind die Genauigkeit unserer RPKI ROV Messungen zu erhöhen, haben wir Messartefakte, die möglicherweise auf Standardrouten zurückzuführen sind, festgestellt.

In unserem *dritten Beitrag* wenden wir uns daher den Standardrouten zu und verbessern zwei bestehende Methoden. Wir führen diese Messungen als kontinuierliche Messungen durch und präsentieren unsere Ergebnisse auf einer Website. Unsere neu abgeleiteten Datensätze ermöglichen den Ausschluss von ASen mit installierten Standardrouten aus den RPKI ROV Messungen und verbessern somit die Ergebnisse weiter. Außerdem entwickeln wir eine Methode zur Identifikation von Middleboxen, was dazu beiträgt, die Messdaten noch weiter zu bereinigen. Während der Arbeit an Techniken zur Ursprungsüberprüfung durch RPKI wurde klar, dass die Sicherheit des Internet Routings nur durch den Einsatz zusätzlicher Sicherheitsmechanismen zur Absicherung des Pfads erhöht werden kann.

In unserem *vierten Beitrag* konzentrieren wir uns auf Algorithmen zur Pfadvalidierung und wie diese auf der bestehenden RPKI-Infrastruktur aufgebaut werden können, um ein hohes Maß an zusätzlicher Sicherheit zu bieten und gleichzeitig eine hohe Wahrscheinlichkeit der Annahme zu gewährleisten. Frühere Versuche wollten maximale Sicherheit auf Kosten der Benutzerfreundlichkeit erreichen und waren nie erfolgreich. Wir führen Simulationen für zwei Algorithmen namens Autonomous System Provider Authorization (ASPA) und AS-Cones mit verschiedenen Einsatzszenarien durch, um festzustellen, welcher Pfadvalidierungsalgorithmus die meisten Sicherheitsvorteile bei minimalen operativen Kosten bietet. ASPA bietet im Vergleich zu AS-Cones eine höhere Sicherheit. Bei beiden Algorithmen ist die Einführung nur in einigen wenigen ASen erforderlich, um einen Gesamtnutzen für die gesamte Inter-Domain-Routing-Infrastruktur zu erhalten. Darüber hinaus veröffentlichen wir einen BGP-Topologie-Generator der die Emulation beliebiger Topologien innerhalb der NIST BGP-SRx Software-Suite ermöglicht.

Zusammenfassend lässt sich sagen, dass wir die Sicherheit des Internet-Routings durch diese Beiträge verbessern indem wir die Pfadvalidierung zu der bestehenden Ursprungsvalidierung hinzufügen. Unsere Ergebnisse wurden öffentlich zugänglich gemacht, die Datensätze sind als Open Source verfügbar und der Quellcode wurde in öffentlich zugänglichen Repositories veröffentlicht.

# Samenvatting

Het internet is een integraal onderdeel van ons hele leven. Vooral jongeren kunnen zich een wereld zonder internet niet meer voorstellen, aangezien het internet bij deze generatie onderdeel is van bijna elk activiteit en interactie. Daarom is het van cruciaal belang dat het internet betrouwbaar en veilig is. Hoewel velen denken dat het internet één samenhangend geheel is, is het daadwerkelijk een netwerk van netwerken. Elk netwerk wordt een Autonoom Systeem (AS) genoemd en heeft de vrijheid om te beslissen over de voorkeursprotocollen en technologieën binnen zijn grenzen.

Het Border Gateway Protocol (BGP) wordt van netwerken gebruikt om onderling te communiceren en is de facto standaard op het internet. Het maakt de uitwisseling van internetprotocollen (IP)-adresinformatie mogelijk en stelt AS'en in staat om te weten naar welke bestemming het verkeer moet worden geleid. Het is vergelijkbaar met het uitwisselen van postcodes tussen steden die het versturen van pakketten onderling mogelijk maken.

Het internet is ontstaan uit een klein onderzoeksproject tussen drie universiteiten waar vertrouwen als vanzelfsprekend werd beschouwd. Hedendaags kunnen we met ruim 74.000 deelnemers niet meer vertrouwen op elkaars goede bedoelingen. Echter, het protocol is qua veiligheid grotendeels hetzelfde gebleven. Daarom bestaan er nog steeds verschillende aanvalsvectoren die het omleiden van verkeer mogelijk maken, wat op zijn beurt kan zorgen voor manipulatie of de onbeschikbaarheid van hele delen van het internet. Het afgelopen decennium zijn er pogingen ondernomen door de Internet Engineering Task Force om routering tussen netwerken te beveiligen. De validatie van de informatie over de oorsprong van adressen op het internet biedt de AS'en de mogelijkheid om te controleren of de afzender van een ontvangen BGP-aankondiging ook daadwerkelijk hiervoor is geautoriseerd. Sinds 2011 implementeert De Resource Public Key Infrastructure (RPKI) dit door een reeks van IP-adressen te koppelen aan AS-nummers.

Meer recentelijk hebben padvalidatie-algoritmen geprobeerd niet alleen de legitimiteit van de oorsprong van de BGP-aankondiging te beveiligen, maar ook het hele traject dat een aankondiging heeft afgelegd. Het voordeel van een dergelijke technologie is dat de ontvangende partij kan verifiëren of een aankondiging onderweg is gemanipuleerd en hierop kan reageren. Er bestaat echter onzekerheid of voortbouwen op de bestaande RPKI-infrastructuur de beste optie is en welke padvalidatie-algoritmen, onder welke beperkingen, de meeste aandacht moeten krijgen.

Daarom is het doel van dit proefschrift om te beoordelen of we met algoritmen die het AS-pad beveiligen boven op de bestaande RPKI kunnen bouwen om de alge-

hele beveiliging van het routeringsysteem te verbeteren. Om ons doel te bereiken en de veiligheid van internetroutering te verbeteren, voeren we grootschalige metingen en simulaties van de inzet van oorsprongsvalidatie, standaardroutes, en algoritmen voor padvalidatie uit.

Onze *eerste bijdrage* vat bestaand gerelateerd werk samen en identificeert tekortkomingen. Wij categoriseren de meest belangrijke publicaties op dit gebied en laten zien waar er nog meer werk nodig is. Onderzoek binnen dit domein van RPKI is onderverdeeld in RPKI Route Origin Authorization (ROA) metingen, RPKI Route Origin Validation (ROV) metingen, en RPKI-veerkracht. Wij constateren dat bestaande methodes er niet in slagen om AS'en die ROV filters toepassen correct te identificeren. Deze methodes bevatten daarom foutpositieven die tot een verkeerde weergave leiden van het aantal door RPKI beschermde AS'en.

In onze *tweede bijdrage* ontwikkelen we een verbeterde RPKI ROV-identificatie methodologie door te vertrouwen op uitgebreide metingen op de data-plane. Wij ontwikkelen een strikte aanpak die foutpositieven vermindert en de identificatie van beschermde AS'en nauwkeuriger maakt. Bovendien vergelijken we zeven Relying Party (RP) software implementaties en adviseren Routinator als de beste oplossing die er momenteel is voor operators. Door ons werk aan ROV-technieken werd ons duidelijk dat de beveiliging van internet routering alleen kan worden verbeterd door het introduceren van extra beveilig-matregels.

In onze *derde bijdrage* kijken we naar de beveiliging van *default routes* en verbeteren twee bestaande methodologieën. De metingen die we hiervoor implementeren worden continu uitgevoerd en we publiceren onze resultaten live op een website. De nieuwe dataset die we hiermee verzamelen maken het mogelijk AS'en met geïnstalleerde default routes van RPKI ROV-metingen uit te sluiten en daardoor de nauwkeurigheid verder te verbeteren. Daarnaast ontwikkelen we een manier om middleboxes te identificeren, waardoor verder kunnen worden opgeschoond.

In onze *vierde bijdrage* concentreren we ons op padvalidatie-algoritmen en hoe ze bovenop de bestaande RPKI-infrastructuur kunnen worde toegepast. Het doel is om een hoog niveau aan extra beveiliging te bieden, terwijl de kans op adoptie groot blijft. Eerdere pogingen wilde maximale veiligheid ten koste van de bruikbaarheid en werden daarom nooit op grootte schaal uitgerold. Wij voeren simulaties uit voor twee algoritmen genaamd Autonomous System Provider Authorization (ASPA) en AS-Cones met verschillende implementatiescenario's om te onderzoeken welke padvalidatie-algoritme de meeste beveiligingsvoordelen bieden bij, tegelijkertijd, minimale operationele kosten. ASPA zorgt voor meer veiligheid vergeleken met AS-Cones en voor beide algoritmen geld dat adoptie slechts in een paar AS'en is vereist om voor een verbetering van de gehele routeringsinfrastructuur te zorgen. Daarnaast ontwikkelen we een BGP-topologiegenerator dat de simulatie van willekeurige topologieën binnen de NIST BGP-SRx softwarepakket mogelijk maakt.

Door padvalidatie toe te voegen aan de bestaande route-origin-validatie dragen onze vier bijdragen bij aan de verbetering van de beveiliging van internetroutering. Onze bevindingen zijn openbaar beschikbaar, datasets zijn open source en de broncode is gepubliceerd op openbaar beschikbare repositories.

# Contents

CHAPTER 1

# Introduction

The Internet connects us to people and services almost anywhere on earth and has become a cornerstone of modern society. While most people perceive it as a coherent whole, it really is a network of networks interconnected by a system to exchange information, commonly known as *routing*. The routing protocol of the Internet is called Border Gateway Protocol (BGP), and it facilitates the exchange of reachability information at its very core. When the Internet was born more than 40 years ago, trust among the three initial participants was implicit. Throughout the past decades, the Internet grew to more than 74,000 so-called Autonomous Systems (ASes), each managing a set of IP prefixes. They are independent of each other on the inside regarding the choice of protocols and technologies but interconnect with each other on the outside via BGP. Due to the Internet's size, connecting ASes from all over the world with opposing political views and interests, implicit trust has become a lack of trust among participants. Although BGP has been known to be inherently insecure for many years, it has never been replaced, nor could its security drawbacks be fixed satisfactorily. Attacks on BGP allow malicious traffic redirection or denial of service. They have been the focus of many research and newspaper articles [1]–[4].

Due to the immense growth of the Internet and the resulting trust issues among participants that lead to attacks with severe impact [5], [6], regulatory bodies recognize the need for extra regulations in this field. The European Parliament resolution of June 10, 2021, on the European Union's cybersecurity strategy for the digital decade "recalls the necessity of better protection of the Border Gateway Protocol (BGP) in order to prevent BGP hijacks [...]" [7]. The USA's Federal Communications Commission (FCC) launched an inquiry into Internet routing vulnerabilities in February 2022 that targets explicitly BGP vulnerabilities and calls for clarification. This is because Russian network operators have been suspected of diverting sensitive traffic to Russia by exploiting BGP vulnerabilities [8]. The FCC points out that the introduced threats "expose U.S. citizens' personally-identifiable information, enable theft, extortion, and state-level espionage, and disrupt otherwise-secure transactions" [9]. Moreover, the German Bundesamt für Sicherheit in der Informationstechnik (BSI), as a nationwide authority on cybersecurity, is aware of the BGP threat landscape and works on scenarios in anticipation of future attacks and their countermeasures [10].

In summary, more attention is required to secure BGP. But how does BGP work, why is BGP inherently insecure, and why have these problems not been solved over the past ~30 years?

## 1.1 BORDER GATEWAY PROTOCOL

We must go back to how BGP evolved to answer these questions. The Internet started as a testbed to exchange information between three universities. Trust between the entities was implicit and taken for granted. However, today, BGP is used between more than 74,000 entities, and knowing and trusting everyone is impossible. We highlight the increase in the number of ASes, which are the participants in the routing infrastructure, and active prefixes (contiguous block of Internet Protocol (IP) addresses) throughout the past decades in Figure 1.1. Let us consider a simple analogy: protecting bicycles against theft. It appears plausible that we would not need to lock our bikes within the University of the Bundeswehr Munich campus as it is enclosed by barbed wire, and we trust everyone within. But when we bike into Munich, it would be unwise not to use additional security measures to ensure we can ride back home. In our example, the Internet can be compared to the city of Munich, as we do not know most of its population and therefore do not know whether they are trustworthy. As a result, we need to enforce additional security measures to keep those with ill intent in check.

The increasing number of participants not only changed the level of trust but also created tremendous complexity when needing to alter parts of the technology, such as upgrading the protocol while maintaining compatibility and supporting legacy equipment. Therefore, changes in BGP take years and sometimes decades to propose, agree upon, and implement.

As mentioned before, BGP facilitates the exchange of reachability information. Consider a packet that we would like to send from Munich to Berlin. Before sending the packet, we need to specify the zip code, among other things. Within each city,

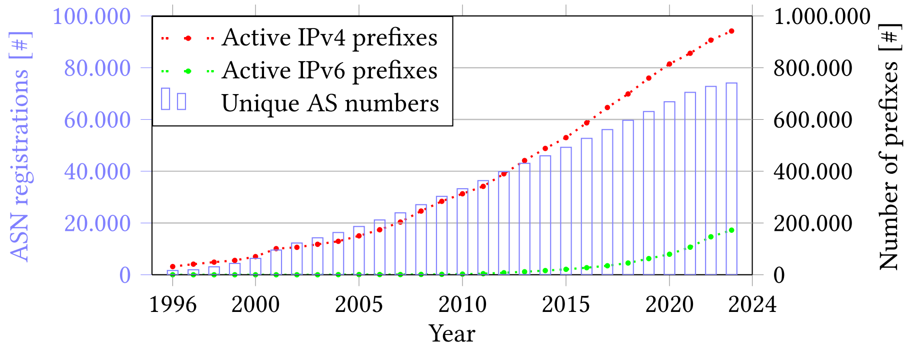


Figure 1.1: Unique Autonomous System Numbers (ASNs) advertised in BGP on each January 1, seen from AS131072 (APNIC R&D). We observe more and more unique ASNs since the beginning, a trend which has slowed down slightly during the last couple of years but remains present. Moreover, we see the number of IPv4 prefixes continuously increasing while the number of IPv6 prefixes exhibits the same trend since 2015.

the street name is used to localize the final destination further. Each city in BGP is represented by an Autonomous System (AS) with a unique numerical identifier. Examples of ASes are AT&T (AS7018), Deutsche Telekom (AS3320), and KPN (AS1136). To route packets between ASes, we need zip codes or IP prefixes in BGP that are announced by the origin AS to its neighbors. The neighbor knows, after receiving the BGP announcement, that traffic destined for an IP address within the received IP prefix range should be sent towards the announcing AS. Within the boundaries of each AS, any arbitrary routing mechanism can be used to route traffic arriving at the exterior routers to the final IP address within the AS. BGP is only responsible for inter-domain routing.

## 1.2 BGP VULNERABILITIES

Now that we discussed how BGP works in its most rudimentary fashion, we turn to its weaknesses. The current BGP version 4 suffers primarily from BGP prefix hijacks, route leaks, and path manipulation attacks. We explain all of these types of attacks in detail in Chapter 2 but provide a brief overview here to allow for a proper understanding of the research goal and questions following hereafter.

**BGP prefix hijacking**, in short BGP hijacking, describes the announcement of an IP prefix in BGP by a different AS than the owner. This is comparable to two cities claiming the same zip code. The packet would either arrive in city A or B. It may, therefore, be sent to an unavailable or impersonated destination. Unauthorized announcements can be mitigated via origin validation. The Resource Public Key Infrastructure (RPKI) implements origin validation by attesting via cryptographically-signed objects that an AS is indeed allowed to announce a specific prefix. ASes implementing filtering techniques based on RPKI will not accept RPKI-invalid prefix announcements and, therefore, traffic will only flow towards the correct origin.

**Path manipulation** attacks are severe and intentional attacks on routing behavior. In this scenario, an attacker misdirects traffic by maliciously making itself part of the route. To stay with our zip code analogy, it can be compared to a third city, claiming that all packets that must go to city A first must pass through their logistics center before arriving at the final destination. This allows the attacker to inspect and manipulate all packets and eventually forward or discard them. Path validation algorithms are designed to mitigate this attack by providing information on the announcement's path. One could think of a GPS tracker in every packet such that the receiving end would notice when the packet to city A went through city B before arrival. While, in theory, Border Gateway Protocol Security (BGPsec) [11] does protect the path, it is due to the missing support of partial deployment and the added computational overhead on routers that it is not expected to be deployed in the foreseeable future. While the use of origin validation can mitigate prefix hijacks, path manipulations can be mitigated by path validation if it is deployed.

**Route leaks** are created by AS operators that accidentally forward announcements to neighbors for whom the announcement is not intended. In our example, city A would advertise a new highway with a lower road charge to get packets to city B.

However, in reality, it is not a highway but a dirt road, and it quickly becomes congested, and packets get stuck. Route leaks create unexpected routing behavior and lead to redirection and, therefore, traffic delay, which can also cause the destination's unavailability. While path manipulations can be mitigated by path validation, route leaks can be mitigated by a subset of path validation algorithms, namely path plausibility algorithms. These are, however, not yet fully developed and require further assessment.

We have now seen how BGP works in its most rudimentary fashion. We have discussed BGP vulnerabilities and how they became more problematic due to a change in trust among ASes as the Internet became successful. Moreover, there are existing solutions to tackle a subset of the problems, like the RPKI or BGPsec, but are these technologies deployed and helpful in solving the problems above? And how could we add path validation on top of the existing origin validation to secure inter-domain routing?

## 1.3 OBJECTIVE, RESEARCH QUESTIONS AND APPROACH

In this thesis, we want to contribute to making Internet routing more secure. The objective of this thesis is:

> *To assess whether we can build on top of the RPKI with algorithms securing the AS path to improve overall routing security.*

It is clear that if the RPKI is not successfully deployed, additional security mechanisms that build on top of the RPKI, such as path validation and path plausibility algorithms, have no chance of adoption. We have identified three research questions. We list them in the following paragraphs and describe how we address them.

ASes typically do not share internal information on the deployment of specific technologies. To see how successful the adoption of RPKI is, the operator community relies on measurements instead. On the one hand, it is crucial to know how many ASes protect their address space by creating RPKI objects. On the other hand, we also need to know how many operators use these RPKI objects to filter RPKI-invalid routes, namely, perform RPKI Route Origin Validation (ROV). Answering the first part is easy, as data can be obtained from public repositories. The second part is significantly more challenging as it requires inferring private router policy configurations. Our literature analysis shows that current approaches have drawbacks and cannot sufficiently pinpoint Route Origin Validation (ROV)-filtering ASes. We therefore state the following research question:

> **RQ 1**—*How can we improve the identification of RPKI ROV deployment in an AS without running into the problem of wrong attribution?*

We address this research question in Chapter 4.

**Approach to RQ1.** We analyze existing methodologies based on the control plane, the data plane, and heuristics. For each of them, we report the shortcomings and state why they cannot accurately pinpoint ROV-filtering ASes. Next, we develop a measurement approach that uses controlled data plane experiments to identify ROV-filtering ASes. We validate our findings with publicly available information.

During the ROV deployment measurement phase, we find that default routes potentially affect our measurements. One assumption of RPKI-based measurements is that if RPKI ROV is deployed, connectivity to RPKI-invalid routes is not present due to filtering. Default routes invalidate that assumption and potentially falsify our measurement results. To mitigate that shortcoming and eliminate results that are not reliable, we phrase the following question:

**RQ 2**—*How can we identify default routes that are present in an AS?*

We address this research question in Chapter 5.

**Approach to RQ2.** Firstly, we determine the extent to which RPKI measurements are affected by default routes. Secondly, we extend two methodologies from the literature to identify default routes. Thirdly, we perform extensive measurements with our set of vantage points using one of the two methodologies. To increase the coverage of our experiments, we apply the second methodology, which we implement as an ongoing effort. Both measurements together allow us to quantify the error of our RPKI ROV measurements by looking into which ASes have default routes installed.

We focus on origin validation by performing RPKI measurements in RQ1 and default route measurements to quantify the RPKI measurement error in RQ2. To secure the origin and avoid path manipulation attacks and route leaks, we intend to contribute to the current state-of-the-art by investigating possible path validation approaches. We know BGPsec as a prominent domain representative. It was standardized in 2017, but deployment never succeeded in the real world. We learned that the design of a security mechanism should not require global deployment before starting to provide benefits. Instead, security additions supporting partial adoption are much more likely to be deployed.

The Internet can be represented as a tree. It consists of quite a small number of transit providers higher up in the tree and a huge number of leaf ASes at the bottom of the tree. During our studies in RQ1, we learned that many transit providers are doing RPKI ROV filtering and RPKI adoption among transits is much higher compared to leaf ASes. Therefore, adoption of RPKI ROV filtering is more likely the higher a node resides within the tree.

One path validation mechanism proposed within the Internet Engineering Task Force (IETF) is Autonomous System Provider Authorization (ASPA). It is also called a path plausibility algorithm since it offers slightly weaker security guarantees at the benefit of supporting partial adoption and it is designed as an out-of-band mechanism. It relies on relationship information between ASes. Each node has to create an ASPA object that contains information about its providers. ASPA builds a validation tree using the bottom-up principle. In addition to the bottom-up approach, we investigate the inverse direction, the top-down approach. A very recent IETF proposal is called AS-Cones. Each AS creates an AS-Cones object containing its customer cone. Therefore, only ASes higher up in the tree must create such objects. Since the RPKI adoption among nodes higher up in the tree is much more common, the idea is to see a much more significant impact of path plausibility algorithms early on without the requirement that every AS creates cryptographic objects. We thus ask:

> **RQ 3**—*What are the advantages and disadvantages for different deployment scenarios of path plausibility algorithms to improve inter-domain routing security?*

We address this research question in Chapter 6.

**Approach to RQ3.** We analyze existing path plausibility approaches and show their benefits and shortcomings. We introduce the path plausibility concept in addition to path validation, which promises higher adoption rates at the expense of weaker security guarantees. We compare ASPA and AS-Cones algorithm deployment strategies in a simulation testbed and make recommendations as to which deployment strategy yields the most benefit for the overall routing security of the Internet. Moreover, we provide the design of our BGP topology generator to emulate network topologies.

## 1.4 ORGANIZATION AND MAIN CONTRIBUTIONS

The remainder of this thesis is divided into six chapters. Figure 1.2 shows them in a schematic overview. In this section, we briefly highlight the main contributions of each chapter.

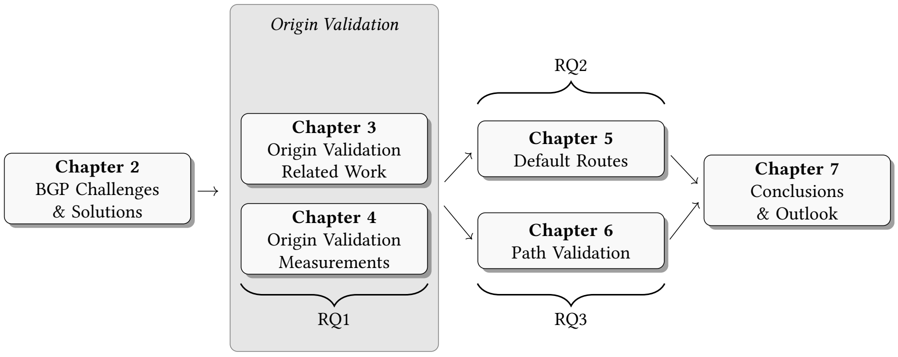


Figure 1.2: Thesis outline

## Chapter 2: BGP Challenges & Solutions

To enable the reader to follow this thesis's central chapters, we need to introduce some basic concepts and provide background information. We start with a brief Internet history and continue with the workings of BGP. Next, we highlight BGP challenges and discuss solutions proposed by literature and within the IETF. We identify why many solutions have remained theoretical proposals and have never seen the light of day. This is particularly important as we work on BGP security improvements in the following chapters and use the lessons from previously failed approaches.

## Chapter 3: Origin Validation - Related Work

Before developing our approaches and methods, it is essential to gain an understanding of the existing domains within the RPKI measurement field. To provide an overview of existing research in the RPKI domain, we develop a classification scheme and categorize more than 40 papers. Additionally, we deepen our understanding of the RPKI and highlight many proposed solutions. Standardized algorithms by the IETF often integrate several good ideas from previous proposals to obtain a result that suits many needs. Hence, it is meaningful to understand earlier proposals and point out why specific ideas have achieved support within the IETF, while others were not considered for standardization. Based on the identified categories, we can put our contributions into context and identify gaps that are being addressed in the following chapters. The main contributions of this chapter are that we:

- Present an RPKI classification scheme.
- Classify existing RPKI measurement research accordingly and present an extensive survey.
- Point out gaps and contradictions in origin validation security research.

This chapter is based on the following peer-reviewed publication:

- N. Rodday, I. Cunha, R. Bush, E. Katz-Bassett, G.D. Rodosek, T.C. Schmidt, and M. Wählisch, 2023. The Resource Public Key Infrastructure (RPKI): A Survey on Measurements and Future Prospects. In Transactions on Network and Service Management (TNSM). [12] *Accepted for publication*

## Chapter 4: Origin Validation Measurements

Several RPKI ROV identification methodologies have been discussed in the previous chapter, and major shortcomings have been pointed out. Our goal in this chapter is to develop a rigorous methodology with a very low false positive rate. We opt for data plane measurements instead of heuristics or control plane measurements. We also perform controlled instead of uncontrolled measurements. The main contributions of this chapter are that we:

- Show why simple end-to-end HTTP measurements falsely attribute ROV deployment to ASes under test if transits in between are filtering.
- Develop a new data plane methodology based on controlled measurements using RIPE Atlas that makes strong inferences for ASes directly peering with our announcement sites (1 AS hop) and weak inferences for longer paths (2+ AS hops).
- Consider Internet Exchange Point (IXP) traversals and build an include list for ASes seen on invalid paths to differentiate between partially and fully filtering ASes.
- Present up-to-date results and confirm that deployment has increased.

Our work triggered a discussion within the IETF on the timing requirements for RPKI relying parties [13]. This chapter is based on the following peer-reviewed publications:

- N. Rodday, I. Cunha, R. Bush, E. Katz-Bassett, G.D. Rodosek, T.C. Schmidt, and M. Wählisch, 2021. Revisiting RPKI Route Origin Validation on the Data Plane. In Proceedings of the Network Traffic Measurement and Analysis Conference (TMA), Virtual. [14]
- N. Rodday, R. Bush, I. Cunha, E. Katz-Bassett, G. Dreo Rodosek, T.C. Schmidt, and M. Wählisch, 2019. Extending RPKI ROV Measurement Coverage, Poster @ 19th Internet Measurement Conference (IMC), Amsterdam, The Netherlands. [15]
- P. Friedemann, N. Rodday, G.D. Rodosek, 2022. Assessing the RPKI Validator Ecosystem. In Proceedings of the 13th International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain. [16]

## Chapter 5: The Impact of Default Routes on Origin Validation Measurements

RPKI ROV measurements rely on the assumption that if an AS is filtering, connectivity to an RPKI-invalid prefix range is terminated. Default routes undermine this assumption as the AS would send traffic to the upstream regardless of whether a route entry is present. Therefore, default routes hinder correct RPKI ROV inferences and need to be quantified to determine the impact they have on RPKI ROV measurements. This chapter aims to develop a default route identification methodology and run measurements to show how significant the impact of default routes is. The main contributions of this chapter are that we:

- Extend prior measurements based on the *path-poisoning* methodology by considering IPv6 and using dual poisoning of the prepended AS path to increase efficiency.

- Add RIPE Atlas vantage points to the *path-poisoning* methodology where available. This allows us to not only identify a default route but also to infer its direction.
- Increase coverage of the *not-announced prefix* methodology by adding NLnog vantage points.
- Introduce a threshold to decide on the ratio of routes before an AS will be flagged deploying a default route.
- Present our findings and up-to-date results as part of an ongoing measurement study on our website https://www.defaultroutes.net. All datasets are publicly available.

This chapter is based on the following peer-reviewed publication:

- N. Rodday, L. Kaltenbach, I. Cunha, R. Bush, E. Katz-Bassett, G.D. Rodosek, T.C. Schmidt, and M. Wählisch, 2021. On the Deployment of Default Routes in Inter-domain Routing. In Proceedings of the ACM SIGCOMM Workshop on Technologies, Applications, and Uses of a Responsible Internet (TAURIN), Virtual. [17]

## Chapter 6: Path Validation

Origin validation provides a strong security mechanism to mitigate BGP prefix hijacks. It does not, however, mitigate path manipulation attacks or route leaks. Path validation or path plausibility algorithms are needed to check the AS path attribute for correctness to avoid the attacks above. This chapter aims to implement and evaluate the ASPA and the AS-Cones algorithm in different deployment scenarios. The main contributions of this chapter are that we:

- Implement the ASPA and AS-Cones algorithms from their respective IETF drafts in a Python simulation framework and publicly release all source code and other artifacts.
- Evaluate both algorithms regarding potential deployment strategies for *route leak* mitigation and ASPA deployment for the *forged-origin prefix hijack* mitigation and provide recommendations as to which ASes to incentivize to deploy the algorithms first to yield the most benefit.
- Discuss weaknesses in the design of both algorithms and provide recommendations on how the algorithm could be improved and deployment accelerated.
- Propose a topology creation framework based on the NIST BGP-SRx software suite and allow the creation of arbitrary topologies based on a directed input graph up to a size of 55,000 containers.
- Implement the AS-Cones IETF drafts in the Quagga routing daemon.

The chapter is based on the following peer-reviewed publication:

- N. Rodday, G.D. Rodosek, A. Pras, R. van Rijswijk-Deij, 2024. Exploring the Benefit of Path Plausibility Algorithms in BGP. In Network Operations and Management Symposium (NOMS), Seoul, South Korea. [18] *Accepted for publication*

- N. Rodday, G.D. Rodosek, 2023. BGPEval: Automating Large-Scale Testbed Creation. In Proceedings of the 19th International Conference on Network and Service Management (CNSM), Niagara Falls, Canada. [19]

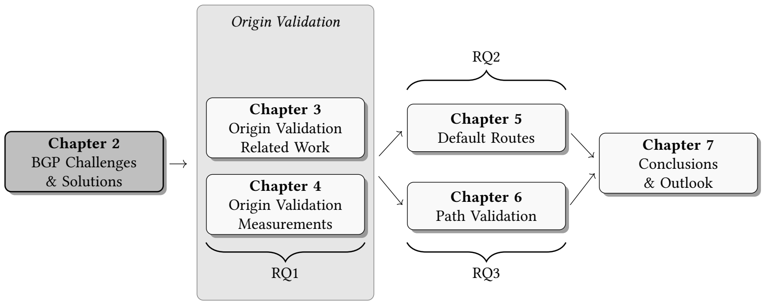
- N. Rodday, R. van Baaren, L. Hendriks, R. van Rijswijk-Deij, A. Pras, and G. Dreo, 2020. Evaluating RPKI ROV identification methodologies in automatically generated mininet topologies. In Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT), Barcelona, Spain. [20]

## Chapter 7: Conclusions and Outlook

In this thesis's last chapter, we revisit the research goal and draw conclusions. Here, we also discuss potential future research directions.

CHAPTER 2

# BGP Security Challenges and Solutions



*In this chapter we explain the basic principles of the inter-domain routing infrastructure and its problems, as well as proposed solutions. These lay the groundwork for the following chapters. Firstly, we provide a brief history section. Secondly, we explain the difference between inter-domain and intra-domain routing and focus on the basic functionality of BGP. We highlight how ASes are able to obtain AS numbers and prefix space, and detail BGP route propagation and route selection process. Thirdly, we discuss BGP hijacking, the Internet Routing Registry (IRR) and RPKI. We talk about path manipulations and path validation algorithms as a mitigation. In addition, we focus our attention on route leaks and path plausibility algorithms and provide a comparison of BGP challenges and solutions. Lastly, we introduce measurement tools and platforms that are used repetitively throughout this thesis.*

## 2.1 HISTORY

Back in the middle of the 20th century, computational power was very costly and sought after. Computers were massively inferior compared to systems of the 21st century. Since there was great demand, but little computational power at different locations, researchers wanted to interconnect computers to aggregate computational power and share information quickly. This required computers to become remotely accessible. To tackle these problems, J. C. R. Licklider and Bob Taylor developed the idea of the Advanced Research Projects Agency Network (ARPANET) in the late 1960s [21]. They published an article with funny cartoons describing many of the features of the future ARPANET [22]. Subsequently, in 1969, a contract was awarded to BBN Technologies to create a prototype with four participating universities: University of California, Los Angeles (UCLA), University of California, Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah. However, the official birth date of the Internet is considered to be 17 years later, in 1983, when the TCP/IP protocol suite was made the official standard in all military computer networking in the US [23]. This truly allowed devices from different manufacturers to communicate with each other. In the same year, numerical IP-addresses were supplemented with domain names. The Domain Name System (DNS) was born, which is essential for human-readable addressing on the Internet, e.g. *www.unibw.de* [24]. During the second half of the 1980s the Internet grew substantially from around 1,000 connected hosts in 1984 to 100,000 in 1989 [25]. Since then the inter-domain routing infrastructure has continued to grow tremendously and interconnects many more universities, people, and continents than initially envisioned. The number of worldwide Internet users in January 2023 is estimated to be around 5.16 Billion [26].

## 2.2 INTER-DOMAIN ROUTING

The Internet is comprised of many independently organized ASes. An Autonomous System is defined in RFC 1930 as: "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" [27]. It is referenced by a number that is obtained from the Internet Assigned Numbers Authority (IANA). Examples are AT&T (AS7018), Orange (AS5511), Deutsche Telekom (AS3320), etc.

There is a differentiation between *intra*-domain routing and *inter*-domain routing. Figure 2.1 shows the separate domains. *Intra*-domain routing refers to the fact that each AS is in control of its own infrastructure and decides which hardware, *intra*-domain routing protocol, also called Interior Gateway Protocol (IGP), etc., it wants to use within its boundaries. Examples are Open Shortest Path First (OSPF), Routing Information Protocol (RIP), Intermediate System to Intermediate System (IS-IS). The best path is usually the shortest path. Such a domain is illustrated in Figure 2.1 as a grey cloud containing yellow links.

*Inter*-domain routing describes the communication between different ASes. In

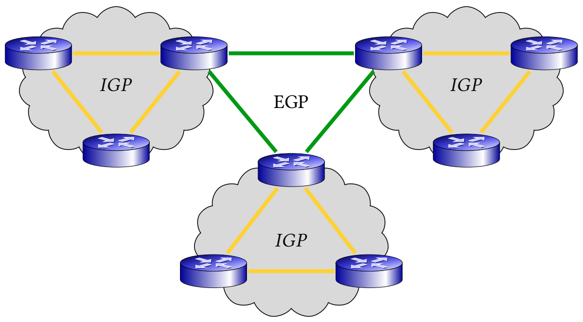


Figure 2.1: Separation of Interior and Exterior Gateway Protocols

order to communicate externally, an *inter*-domain routing protocol, also called Exterior Gateway Protocol (EGP), needs to be chosen. The de-facto standard is the Border Gateway Protocol (BGP) in its most recent version 4 [28]. There are some alternatives available, such as EGP [29] (a protocol) and IDRP [30], but they do not carry practical relevance. Figure 2.1 shows the inter-domain connectivity as green links inbetween ASes.

Within the *inter*-domain routing infrastructure we typically differentiate between two types of domains: *Transit* domains, which allow peers to route traffic via their AS, and *stub* domains, which do not allow for transit and are connected to at least one transit provider.

**Border Gateway Protocol.** BGP is a path-vector protocol that comes in two forms, internal Border Gateway Protocol (iBGP) and external Border Gateway Protocol (eBGP). Routers communicating via BGP are called BGP speakers. When two BGP speakers communicate with each other, they are referred to as *peers*. BGP is used to communicate the availability of Network Layer Reachability Information (NLRI), *i.e.,* IPv4 and IPv6 prefixes, between peers. For example, $AS1$ communicates to $AS2$ the availability of the IPv4 prefix `1.2.3.0/23`, as illustrated in Figure 2.2. $AS2$ continues to propagate the IPv4 prefix via $AS3$ to $AS4$. The ASN of the announcing AS is always prepended to the AS path attribute. The notation specifies, that the origin AS remains on the very right hand side of the AS path attribute. From now on, $AS2$, $AS3$, and $AS4$ have an entry in their routing tables and are able to send traffic for the IPv4 prefix `1.2.3.0/23` to $AS1$. This section refers to BGP v4, specified in RFC

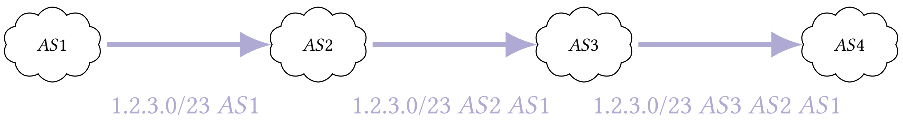


Figure 2.2: Border Gateway Protocol Propagation

4271 [28], if not mentioned otherwise. However, there are several Request for Comments (RFC) updates available that specify details of the workings of BGP [31]–[39]. BGP v4 supports five different BGP message types:

1. **Open Messages**: An Open Message is used to create a BGP peering session after the TCP session has been established. It contains a value for the hold timer until the session expires, among other things.
2. **Update Messages**: The most common message type. It is used to exchange reachability information between peers. Throughout this work, we focus on the content of Update Messages.
3. **Keep-Alive Messages**: These messages are sent to keep a BGP peering session alive, if it is not actively used to exchange information. It renews the hold timer.
4. **Notification Messages**: These messages are only sent in case of an exception. The BGP and TCP sessions terminate afterwards.
5. **Route-Refresh Messages**: If the route refresh capability advertisement has been received by a peer, the router sends these messages to renew the advertised routes.

**Neighbor States.** The internal workings of BGP can be represented in a Finite State Machine (FSM). Six states have been defined:

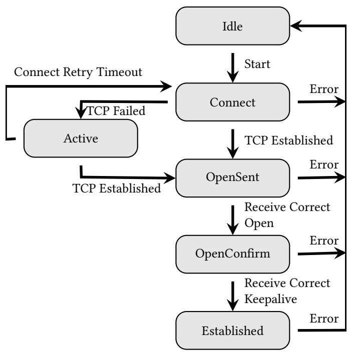


Figure 2.3: BGP Finite State Machine

1. Idle: Initial state
2. Connect: Waiting for TCP connection to be completed
3. Active: Initiated a TCP connection but waiting for answer from peer
4. OpenSent: Open message was sent
5. OpenConfirm: Waiting for keep-alive after Open message exchange
6. Established: A working BGP session

Only in the Established state, will BGP be able to exchange update messages and perform route selection.

**Longest Prefix Match.** An underlying principle in BGP is the use of the longest prefix match. If there are two or more routes available towards a target, BGP will always choose the most specific option available. For example, a packet would need to be routed to the destination `1.2.3.4` and the routes shown in Listing 2.1 are present in the RIB. BGP would prefer the /24 network, since it is more specific.

Listing 2.1: BGP Longest Prefix Match

```
1.2.3.0/16 1234 8888
1.2.3.0/24 6789 2222 <-- chosen for destination 1.2.3.4
```

**BGP attributes.** Due to the design of BGP as a path-vector protocol, and in contrast to the use of a metric in distance-vector protocols such as RIP and Interior Gateway Routing Protocol (IGRP), BGP does not propagate a cost vector in its update messages. Instead, it relies on so-called BGP attributes. There are well-known mandatory attributes (e.g., origin, AS path, and next hop) and well-known discretionary attributes (e.g., local preference). Every router must be able to parse these attributes, but only the mandatory ones are expected to be present in every update message. Moreover, there are optional transitive (e.g. communities) and optional non-transitive (e.g. cluster list) attributes. Since they are optional, they do not need to be recognized by a router. However, transitive attributes are meant to be forwarded by the receiving party, while non-transitive attributes can be safely ignored.

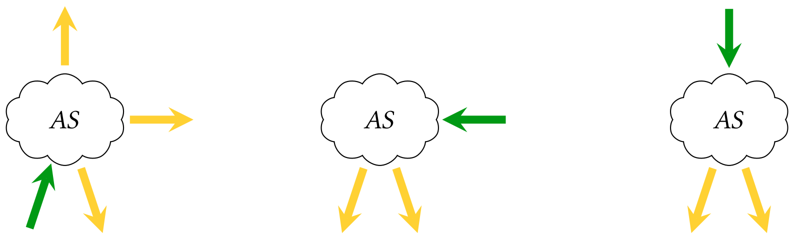
Since the AS path already contains all previous AS hops, it provides the opportunity to easily detect routing loops. If an AS observes its own ASN in the BGP AS path attribute, it simply discards the update message. Following this method, BGP implements its *loop detection* and prevention mechanism.

When advertising routes to peers, operators sometimes wish to attach information to some announcements to control the behavior of downstream of upstream routers. BGP communities [40] are meant to serve that purpose. They are typically described as two 16-bit representations, separated by a colon: (0–65535):(0–65535). The first section specifies the ASN, the second one the community, which may carry individual meaning. Since the number of ASNs is constantly increasing, larger ASNs with 32-bits have been standardized. In order to make communities usable for them as well, RFC 8092 [41] specifies Large Communities with a 96-bit representation. Semantics of communities can be different for iBGP and eBGP. An operator could define a community value that is added to routes when they enter the AS and is considered in all internal BGP routers thereafter. BGP communities with global significance are:

- Internet: Advertise to any peer (0:0)
- No-Export: Do not advertise to any outside BGP peer (65535:65281)
- No-Advertise: Do not advertise to any BGP peer (65535:65282)
- No-Export-Subconfed: Do not advertise to any BGP peer outside the local AS (65535:65283)

Another example is the Blackhole community, which was recently added in RFC 7999 [42]. When it is used, it triggers blackholing at remote ISPs and can therefore be used to distribute the information that traffic for a specific prefix is unwanted and should be discarded. The IANA maintains a list of registered well known communities at an online resource [43].

**Valley-free routing / Gao-Rexford model.** On the one hand, in-bound traffic is controlled by advertising routes towards peers. On the other hand, out-bound



(a) Routes learned from customers are exported to any other peer.

(b) Routes learned from peers are only exported to customers.

(c) Routes learned from providers are only exported to customers.

Figure 2.4: Export policies according to the Gao-Rexford model.

traffic is controlled by selecting routes received from peers and including them into the Route Information Base (RIB). The *Gao-Rexford model* [44] describes the business relationships between ASes and how ASes should design their policies to avoid route divergence and the resulting route oscillation [45], [46]. Such behavior degrades the end to end performance of the inter-domain routing infrastructure. If we apply the Gao-Rexford model to real world interconnections, we observe three basic scenarios, as illustrated in Figure 2.4. Firstly, a route learned from a customer would be propagated to any other peer, since the customer pays for any forwarded traffic. Secondly, if the route was received by a peer, it is only propagated to customers, since they would pay for the traffic transmitted via that link. Forwarding traffic to lateral peers would not yield any monetary reward. Forwarding to upstreams would even cost money. Thirdly, a route learned by a provider is forwarded to customers only for the same reason. Forwarding to peers would make the AS in question pay for traffic transmitted by a peer. Forwarding to another upstream would make the AS pay on both sides and has therefore to be avoided.

If aforementioned rules are not adhered to, which might happen due to fat-fingering or misconfiguration via other means, so-called *route leaks* occur. These situations are usually financially undesirable for the leaking AS. Measures can be implemented in BGP by tagging routes with BGP communities on ingress and execute filters on egress. A detailed description of route leaks is presented in Section 2.5.3.

**Route Decision Process.** There exist multiple routing tables within a BGP speaker:

- Adj-RIBs-In: Contains unprocessed routing information received by the local BGP speaker from peers.
- Loc-RIB: Contains the routes that have been selected after the local BGP speaker's decision process has been applied.
- Adj-RIBs-Out: Contains routes that will be advertised to peers within UPDATE messages of the local BGP speaker.

The decision process for installing received routes that are within Adj-RIBs-In into the Loc-RIB table and selecting a subset of those routes for export to peers within the Adj-RIBs-Out table is outlined as follows:

- Phase 1 - Calculation of Degree of Preference: The exact nature of determining the local degree of preference depends on local policies. It is also called *local_pref*. Every operator is free to weight attributes differently. A route with a higher *local_pref* would take precedence over a route that has a lower value assigned.
- Phase 2 - Route Selection: Once a degree of preference for a received route has been calculated, route selection is commenced. Some routes might be excluded, e.g. when the next_hop attribute contains an address that is unreachable or when the local AS number is detected in the AS path attribute. Routes must also be resolvable. This ensures that only valid routes are entered into the Loc-RIB routing table. For each set of destinations that have at least one feasible route in Adj-RIBs-In, the BGP speaker identifies the route that has (i) the highest degree of preference, (ii) is the only route to that destination, or (iii) is selected by a tie-breaking rule. Tie-breaking rules come into action when two routes within Adj-RIBs-In have the same degree of preference and there would be no clear winner otherwise. Many tie-breaking rules have been defined, e.g. the index of the AS number of the announcing AS. They are applied in a sequential manner until one route wins over the other. Once a route has been chosen, it is installed in the Loc-RIB table.
- Phase 3 - Route Dissemination: Route dissemination happens when Phase 2 is completed, routes within the Loc-RIB have changed, or a new BGP speaker is connected. Local policies are applied to select routes from within the Loc-RIB table that should be propagated to peers. These routes are installed in the Adj-RIBs-Out table. Route aggregation is an important step that is applied during dissemination to reduce the amount of BGP announcements and also the number of entries within each RIB. Two prefixes, e.g. 1.2.**2**.0/24 and 1.2.**3**.0/24, may be aggregated to 1.2.2.0/**23**. Route aggregation is optional, but is commonly applied to reduce the number of update messages and save memory in routers.

**Prefix and ASN delegation.** The IANA is a standards organization that is responsible for global IPv4 and IPv6 prefix allocation and ASN allocation, amongst other things. These Internet resources are delegated from the IANA to the Regional Internet Registries (RIRs). From there on Internet resources are further delegated to National Internet Registries (NIRs), Local Internet Registries (LIRs), Internet Service Providers (ISPs), and other organizations. Figure 2.5 highlights the delegation schematically and shows the geographical distribution of the five RIRs.

**Internet Exchange Points.** ASes need to be present in multiple locations in order to peer with each other. To establish a physical presence in many different locations can be costly and peering with many different ASes can be complex and error prone. Maintaining multiple peering relationships results in significant management overhead. IXPs are entities that aim at reducing such cost and complexity. They can be seen as a marketplace where all parties come together and meet. On a technical

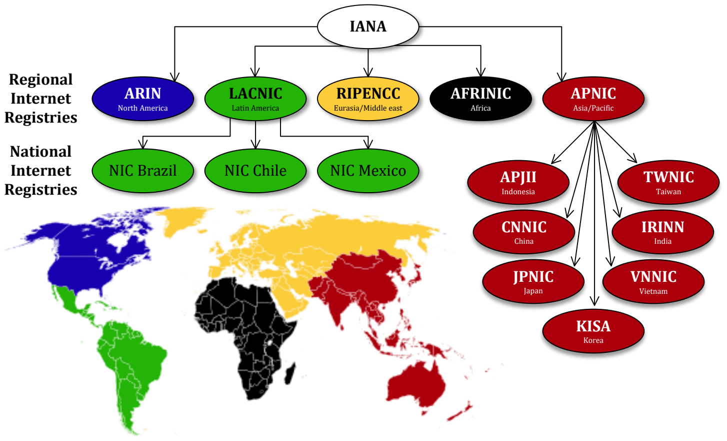


Figure 2.5: Geographic breakdown of RIRs, modelled after [47]

basis, they allow mostly for two models: Firstly, direct peering between two ASes, while the IXP facilitates the peering. Secondly, peering with a route-server, which is run by the IXP, that provides access to many connected ASes simultaneously. Such route-server peerings shorten the AS path, which is why IXPs are considered a factor for Internet topology flattening [48], [49]. It is also only necessary for an IXP client to maintain a single BGP peering connection, while receiving routes from many different ASes. Moreover, IXPs implement security features that might be too costly or complex for a small AS to implement by themselves. In summary, IXPs are a method to gain wide connectivity while being cost-effective.

**Default Free Zone.** Routing tables are a collection of available resources, announced by different peers. If a packet needs to be routed to a certain destination, but there is no matching entry in the routing table, connectivity would not be possible. To avoid scenarios in which smaller ASes do not have a full routing table and would therefore not be able to reach all parts of the Internet, default routes are installed. They function as a gateway of last resort. If no entry could be found in the routing table that matches the requested destination, the packet is sent to the peer defined in the default route. This is typically the primary upstream of an AS.

The Default Free Zone (DFZ) is described as a collection of ASes in which not a single AS has a default route installed. Tier-1 providers are expected to reside within the DFZ. Bush *et al.* [50] have shown back in 2010 that default routes are very well present in the DFZ.

## 2.3 ROUTING SECURITY BEST PRACTISES

**CIA triad.** The CIA triad illustrates the main security goals: Confidentiality, Integrity, and Availability. Confidentiality is a set of rules that limits access to information. It describes who is allowed to access what. Integrity is the guarantee that information is trustworthy and accurate. Availability is the assurance of the resource being accessible by authorized entities. Additionally, authentication verifies the identity of a person/system and authorization determines the access right.

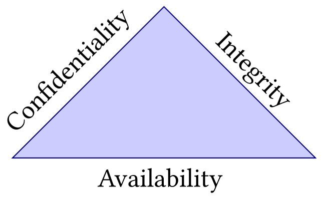


Figure 2.6: CIA triad

**Confidentiality & Integrity of BGP Messages.** The aforementioned security goals of BGP can be endangered, if BGP sessions are compromised. BGP sessions are by default not protected and send clear text messages from one peer to another. They run on top of Transmission Control Protocol (TCP) on port 179. It is therefore easily possible as an attacker to read the content, compromising the *confidentiality* of the message. It is also possible to alter the content of a BGP message while it is in transit, compromising the *integrity* of the message. As a countermeasure, BGP communication can be protected by encrypting the channel. Internet Protocol Security (IPSec) offers such functionality [51]. Alternatively, BGP traffic can be tunnelled via *e.g.*, a Virtual Private Network (VPN) or a Secure Shell (SSH) tunnel. However, performance might be significantly degraded when applying such tunnelling mechanisms as they add cryptographic overhead and slow down operations. Another drawback is certificate management and the additional complexity introduced by key-rollovers. Static keys that never change are considered a bad practice. Keys should be rolled on a regular basis as a key compromise might go unnoticed. However, IPSec deployment has its drawbacks. Once IPSec is deployed, an attacker could send many authentication requests towards a BGP peer using IPSec in order to render it offline. Such a Denial of Service (DoS) attack endangers the *availability* of the BGP protocol.

**Spoofing.** Since there is no authentication, BGP peers could be *spoofed* by a third party. Spoofing describes a process in which an illegitimate third party imitates the original sender. The destination cannot differentiate between a legitimate and an illegitimate source and treats the packets the same. To avoid spoofing of BGP peers, shared secrets for BGP sessions are used. RFC 2385 [52] describes the use of the Message Digest Algorithm 5 (MD5) digest for authentication inserted into the BGP packets. The shared secret can be configured with a single line of code in a BGP speaker's configuration file. A different shared secret should be used per peering session, which might lead to an overhead in administration.

Additional security to protect the confidentiality, integrity, and availability of BGP comes at an additional cost. As in many other scenarios, this tradeoff has to be carefully considered. While the aforementioned security challenges focus on attacking the protocol itself, we will investigate logical BGP security challenges throughout the following paragraphs. We therefore assume that the BGP messages themselves are delivered as intended and remain unchanged.

**Best Practises.** An initiative to make routing more secure is Mutually Agreed Norms for Routing Security (MANRS). An implementation guide is available in [53]. MANRS is a set of guidelines that are considered best practises to avoid routing problems. Participating ASes are expected to keep their IRR data, amongst other things, up-to-date. The guide is mostly written as a hands-on tutorial and contains many examples from router configuration files.

It is generally good practise to use route tagging. When a route is received on an ingress router, it is tagged with a specific community attribute depending on the peering relationship. On egress, this information is used to only allow the routes through that carry the correct BGP communities. Without tagging, the knowledge of where the route came from would be lost. Implementing this methodology allows the operator to easily follow the export policies illustrated in Figure 2.4, see Subsection 2.2 for more details.

**Prefix filters.** Such policies to filter incoming and outgoing BGP announcements are enforced on a router using *prefix filters.* None of these are bullet-proof concepts and rather try to lower the risk of BGP threats with the limited tool-set available to operators. We will highlight the most common filtering techniques and explain their limitations:

Prefix filters are regularly built based on Internet Routing Registry (IRR) data. An AS should only accept routes if a matching route object could be found in the IRR. The IRR is known to be insecure and can therefore not be used to mitigate intentional hijacks since the attacker could possibly inject arbitrary route objects, but it does help to avoid accidental hijacks. We explain the IRR in detail in Subsection 2.4.2. It is common to set a threshold for the expected amount of prefixes for each neighbour, called *Maximum-Prefix.* Should some routes suddenly be deaggregated resulting in a flood of more specific BGP announcements, the mechanism kicks in to terminate the peering session or raise a warning. Also massive route leaks can be prevented with this feature. The *peer lock* mechanism describes the acceptance of prefixes only for specific upstreams of a protected AS. For example, AS1 would like to participate in a peer lock mechanism, it would communicate the allowed upstream(s) to the peer locking AS. From now on, the peer locking AS would not accept any routes of the protected AS via another upstream than the legitimated one. This mechanism requires knowledge of the allowed upstreams of a peering AS and is deployed in collaboration with the peering AS. The *"bignetworks filter"* describes filtering of tier-1 providers in received BGP announcements. There are only a few ASes that are considered as tier-1 providers. These global networks would usually not buy transit from a peer, especially not from a smaller customer. Therefore, if a route is received via a peer or a customer and it contains a tier-1 provider, it should be discarded. Listing 2.2 displays the tier-1 providers [54].

Listing 2.2: Tier-1 ASNs

```
174       # Cogent Communications       3491  # PCCW Global
209      # CenturyLink Communications  5511  # Orange
286/3257 # GTT Communications           6453  # Tata Communications
701/2828 # Verizon                      6461  # Zayo Group
1239     # Sprint                       6762  # Telecom Italia Sparkle
1299     # Arelion (formerly Telia)     6830  # Liberty Global
2914     # NTT Communications           7018  # AT&T
3320     # Deutsche Telekom             12956 # Telxius
3356     # Lumen Technologies
```

*Bogon ASNs* should be filtered. They are ASNs that carry specific meaning and should not appear in public BGP routes. Listing 2.3 displays the specific numbers, from [55]. If any of these ASNs appear in an AS path, a configuration lapse happened.

Listing 2.3: Bogon ASN List

```
AS0                          # RFC 7607
AS23456                      # RFC 7607
AS[64496..64511]             # RFC 5398
AS[65536..65551]             # RFC 5398
AS[64512..65534]             # RFC 6996
AS[4200000000..4294967294]   # RFC 6996
AS65535                      # RFC 7300
AS4294967295                 # RFC 7300
AS[65552..131071]            # IANA reserved
```

*Bogon prefix filtering* is similar to bogon ASN filtering. The IETF has also specified several prefix ranges for private or experimental use. These should not be announced within the public domain and can be safely filtered. Listing 2.4 displays the specific numbers, from [55]:

Listing 2.4: Bogon Prefix List

```
0.0.0.0/8+          # RFC 1122 'this' network
10.0.0.0/8+         # RFC 1918 private space
100.64.0.0/10+      # RFC 6598 Carrier grade nat space
127.0.0.0/8+        # RFC 1122 localhost
169.254.0.0/16+     # RFC 3927 link local
172.16.0.0/12+      # RFC 1918 private space
192.0.2.0/24+       # RFC 5737 TEST-NET-1
192.88.99.0/24+     # RFC 7526 6to4 anycast relay
192.168.0.0/16+     # RFC 1918 private space
198.18.0.0/15+      # RFC 2544 benchmarking
198.51.100.0/24+    # RFC 5737 TEST-NET-2
203.0.113.0/24+     # RFC 5737 TEST-NET-3
224.0.0.0/4+        # multicast
240.0.0.0/4+        # reserved
```

*Long ASN paths* are created when excessive path prepending is used. Section 2.5.1 details the problem. To mitigate such problems, long AS paths should be filtered. The operator is free to set a threshold of choice, but 50 is commonly considered a sufficient length, while the average AS path length observed on the Internet is 4.4 [56]. *Small prefixes* should be rejected. The RIPE Routing Working Group recommends the smallest commonly routed prefix sizes to be /24 and /48 for IPv4 [57] and IPv6 [58], respectively. Prefix filters should be designed to reject anything that is more specific. *Default routes & own prefixes* should also be rejected, unless a default route was specifically requested. Moreover, one should reject any announcements for their own prefix ranges as such present an obvious routing error.

## 2.4 ORIGIN VALIDATION

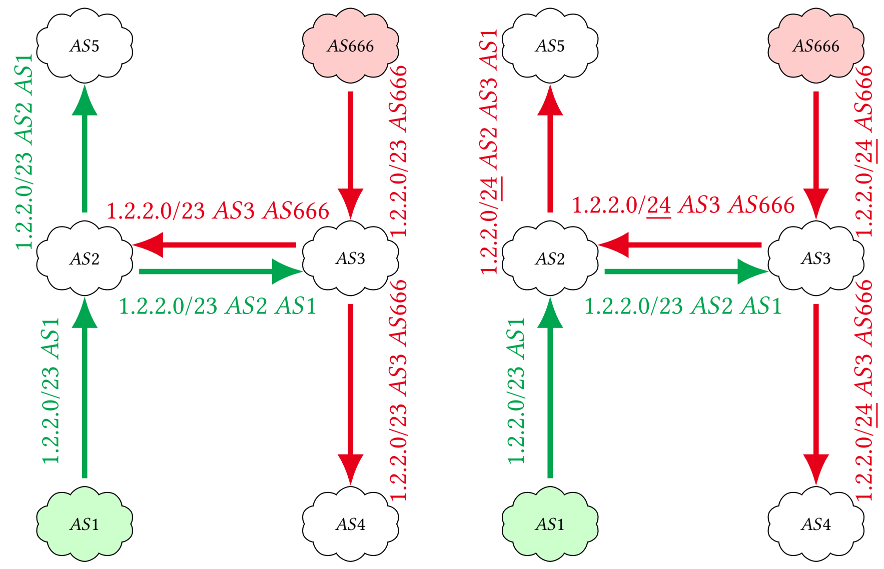
Within the inter-domain routing system ASes obtain resources, *i.e.,* ASNs, IPv4 and IPv6 prefixes, through the IANA. Since BGP does not provide any security mechanisms, it is assumed that the legitimate AS announces its BGP prefix to its peers. However, intentionally or unintentionally, operators sometimes announce prefix space that was assigned to other ASes and is not owned by themselves. Such behavior is called a BGP prefix hijack.

### 2.4.1 BGP Prefix Hijacking

Because of the lack of proof of address ownership, prefix hijacking is possible. We differentiate between exact prefix hijacks and more specific prefix hijacks. Another attack, in which unassigned address space is used for malicious purposes, is called prefix squatting.

**Exact prefix hijack.** An exact prefix hijack describes the announcement of the exact same resource into the BGP routing system. Figure 2.7a illustrates $AS1$ as the legitimate origin of IP prefix `1.2.2.0/23`. The regular propagation is shown in Figure 2.2. In our hijacking example, an additional $AS5$ and an attacker $AS666$ are added. $AS666$ announces the exact same prefix as the legitimate origin $AS1$. Since peers cannot differentiate between legitimate and illegitimate announcements, they accept and propagate the best path, according to local policy. This usually means that the announcement with the shorter path wins for otherwise equal parameters.

In an exact prefix hijack, the origin that is closer to the targeted ASes will usually receive the traffic. It is therefore possible that only parts of the BGP ecosystem will send traffic to the hijacker, while other parts will remain to send traffic towards the legitimate origin. In our particular example in Figure 2.7a, $AS2$ is closer to $AS1$. The AS path is therefore shorter and only contains a single hop. $AS2$ would therefore side with the announcement of $AS1$. $AS5$ which sits 'behind' $AS2$, receives the legitimate announcement. $AS3$ receives the illegitimate announcement of $AS666$. $AS3$ is closer to $AS666$ and would therefore choose the announcement of $AS666$. $AS4$ sits 'behind' $AS3$ and only sees whatever is forwarded from $AS3$ towards $AS4$. In this case, the announcement forward would be irrelevant, as the traffic for the prefix `1.2.2.0/23` is sent to $AS3$ in any case, with $AS3$ preferring the attacker instead of the legitimate origin.



(a) Exact prefix hijack: *AS*1 is the legitimate origin, *AS*666 announces the exact same IP prefix. ASes will typically pick the shortest route in such a case. *AS*3 chooses *AS*666, *AS*2 chooses *AS*1. *AS*4 depends on *AS*3 in any case, no matter which route is advertised, while AS5 depends on AS2.

(b) More specific prefix hijack: *AS*1 is the legitimate origin, *AS*666 announces a more specific IP prefix. All ASes that receive the more specific announcement will choose *AS*666 as destination for traffic. The announcement travels further compared to the exact prefix hijack.

Figure 2.7: BGP prefix hijack types

**More specific prefix hijack.** A more dangerous variant of prefix hijacks are more specific prefix hijacks. BGP is built on a mechanism called longest prefix matching, see Section 2.2. Traffic will always be sent to the most specific prefix found in the routing table.

We illustrate such a scenario in Figure 2.7b. *AS*1 remains to announce `1.2.2.0/23` as the legitimate origin. The attacker, *AS*666, announces a more specific prefix `1.2.2.0/`**`24`**. Since the announcement is more specific, it is forwarded within the whole inter-domain routing infrastructure. From now on, every AS within the illustrated topology picks the attacker's route for the prefix `1.2.3.0/24` and all traffic towards that prefix range is forwarded to the attacker. *AS*1 also receives the more specific prefix, but should not accept it, as it is common understanding that one's own prefixes should not be accepted when received from external peers.

It is within the nature of a more specific prefix that it does not cover the full range of the covering prefix. The legitimate less specific prefix continues to propagate (not shown in Figure 2.7b for simplicity), but packets on the data-plane will be routed via the more specific route announcement. To attract all traffic of the covering prefix, the attacker would need to announce a second more specific prefix `1.2.`**`3`**`.0/24`. If the upstream, in our case *AS*3, would perform prefix aggregation, both prefixes `1.2.2.0/24` and `1.2.3.0/24` would be forwarded as `1.2.2.0/23`, leading to the previously introduced scenario of an exact prefix hijack.

More specific prefix hijacks are used as a common countermeasure against DoS attacks. The anti-DDoS service providers, such as Akamai or Cloudflare, use such announcements to divert traffic to prefix ranges under attack to their scrubbing centers, before forwarding legitimate traffic to the origin AS. In this case, anti-DDoS service providers are authorized to perform such more specific prefix hijacking.

There are also limits to more specific prefix hijacks. Since there is common understanding that subnets smaller than /24 and /48 in IPv4 and IPv6, respectively, should not be routed on the public inter-domain infrastructure, a /25 announcement in IPv4 would very likely be filtered. This holds true for ill-intentioned announcements and well-intentioned announcements from anti-DDoS service providers.

**Prefix squatting.** A prefix space can be unassigned, as well as assigned but currently unused by the owning AS. Prefix squatting describes an attack in which an unauthorized AS announces such unassigned or dormant prefix space [59]. In contrast to prefix hijacking, in a prefix squatting attack the prefix is currently not announced by the legitimate owner. It is mostly used for spamming purposes or other illegal activities which require a frequent change of IP addresses [60]. IP addresses (or blocks) are ranked with a reputation by companies such as Spamhouse, etc. These blacklist providers quickly pick up squatted IP address blocks and add the respective blocks to the blacklists. Once they are added to the blacklist, the IP ranges become worthless since traffic that originates from there will be rejected by the destination. For example, an email that is sent from an IP range on a blocklist will be rejected by the receiving email provider.

### 2.4.2 Internet Routing Registry

In order to mitigate the aforementioned BGP prefix hijacking and squatting attacks, prefix ownership information is required. An operator needs to be able to check which AS is allowed to announce which resource. The Internet Routing Registry (IRR) provides such information. The IRR is a distributed database, intended to "help debug, configure, and engineer Internet routing and addressing" [61]. It currently comprises 18 geographically distinct databases, each run by a different entity. The information contained is helping network operators to easily find who to contact in case of routing problems. It also allows for building prefix filters and therefore making routing more secure, which is why many networks require IRR objects in their peering agreements.

There are many object types within the IRR. We highlight the three most relevant ones for this work in the following paragraphs:

**as-set** An as-set has a name and a list of members. It can be used instead of listing each of the members individually. For example, an AS could create an as-set for a customer-cone in a specific region. Instead of iterating over each and every single customer AS in its filters, the as-set can be used as a reference. To apply a rule to multiple regions, the respective as-sets can be entered into the filter. Instead of changing rules in prefix filters when customers are added or removed, the AS simply edits the as-set object. This feature significantly reduces complexity of prefix filter maintenance.

Listing 2.5: as-set

```
as-set: as-cust-cone-europe
members: AS1, AS2, ...
```

**aut-num** An aut-num object binds an AS number, *e.g.,* AS47065, to a symbolic name of the AS, *e.g.,* PEERING-TESTBED-AS47065. Moreover, the object provides valuable contact information as well as information about import and export policies. Such public display allows other operators to see peering policies and make adjustments to their announcements, if necessary. Listing 2.6 shows an excerpt. aut-num objects are usually much longer and contain a multitude of additional data.

Listing 2.6: Excerpt of AS47065 aut-num object

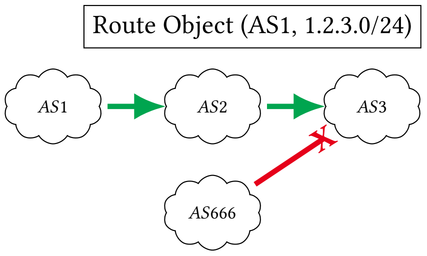
```
aut-num:        AS47065
as-name:        PEERING-TESTBED-AS47065
mp-import:      afi ipv4.unicast from AS-ANY accept ANY
mp-export:      afi ipv4.unicast to AS-PEERING-TESTBED announce ANY
tech-c:         DUMY-RIPE
admin-c:        DUMY-RIPE
notify:         noc@peering.ee.columbia.edu
```

**route** A route object links an IP prefix to an AS number. If an announcement is received, the router can check whether an IRR route object exists that would allow the origin AS to make such an announcement. This is very similar to the idea of RPKI, except that IRR information is insecure and is not backed up by a certificate hierarchy that would be able to certify that the information is correct.

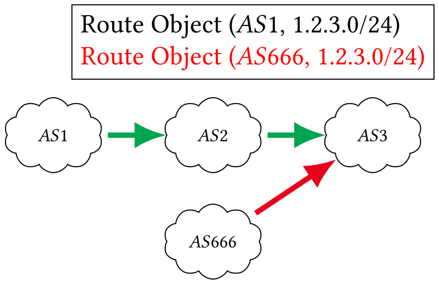
Listing 2.7: route object

```
route:   147.22.0.0/16
origin:  AS47065
```

The IRR is used to extract peering information [62], detect BGP hijacks [63], [64], and infer AS relationships from IRR routing policies [65]. Each object can be accessed via a web interface or an Application Programming Interface (API). Therefore, objects can be automatically fetched and used to build prefix filters to mitigate hijacks and route leaks and increase the security of inter-domain routing. Unfortunately, the data contained in the IRR is not cryptographically secured and is known to contain a lot of bogus information [66]. Moreover, some ISPs do not like to add information to IRR in order not to not leak sensitive information [67]. There are repositories where attackers can add objects without authorization and hence compromise the integrity of the data. An as-set also used within prefix filters might be altered



(a) Attack unsuccessful: IRR Filter prevents route installation.


Route Object (AS1, 1.2.3.0/24)
Route Object (AS666, 1.2.3.0/24)
AS1
AS2
AS3
AS666

(b) Attack successful: Attacker adds an IRR object and the IRR filter does not filter the route anymore.

Figure 2.8: IRR filtering

by the original creator at a later stage. The operator relying on the as-set is not likely to notice such a change as it is the nature of as-sets to be dynamic and adjustable to change. as-sets are known to contain outdated information, as old customers are not removed. An intentional attack that is planned and carefully executed, including the manipulation of IRR data, would therefore still succeed. However, the vast majority of irregularities in inter-domain routing stem from misconfigurations. The IRR is a very useful tool in filtering these misconfigured announcements.

Figure 2.8a shows how an IRR filter prevents the installation of a route into the local RIB of *AS*3. If *AS*666 creates an IRR route object legitimizing its own AS to announce the prefix 1.2.3.0/24, as displayed in Figure 2.8b, the announcement would be accepted by *AS*3 and the filtering can be circumvented. This is possible since IRR objects can be added without authorization of the legitimate owner.

### 2.4.3 Resource Public Key Infrastructure

To work around this problem and bind ASes to prefixes in a cryptographically secure manner, several approaches have been proposed [68], [69], and ultimately the Resource Public Key Infrastructure (RPKI) emerged. Development of the RPKI started in April 2008 within the Secure Inter-Domain Routing (SIDR) working group, while RPKI was standardized as RFC 6480 [70] in 2012. Four of the five RIRs started to deploy RPKI in January 2011, and the American Registry for Internet Numbers (ARIN) in September 2012 [71]. It describes an architecture in which authorities attest to objects containing security-relevant information, similar to Secure Sockets Layer (SSL)/Transport Layer Security (TLS) and DNS/DNS Security Extensions (DNSSEC). An Internet draft detailing a RPKI repository analysis and requirements can be found in [72]. Sizing estimates for a fully deployed RPKI were made in [73].

We explain the components of the RPKI using Figure 2.9. The IANA holds the whole IPv4 & IPv6 address space and assigns subnets to the five RIRs. Each RIR acts within the RPKI as a trust anchor via a self-signed root certificate. This has been a controversy from the beginning, as the Internet Architecture Board (IAB) originally recommended a single trust anchor [74], but later revisited its statement saying proven by operational practise five Trust Anchors (TAs) also work well [75]. It is therefore possible for each RIR to attest to prefix space managed by another RIR. The same argument led to proposals suggesting to use a Dalskov protocol [76] as a distributed threshold signature model instead, in which only a set of RIRs could jointly sign End-Entity (EE) certificates and delegate address space. Unfortunately, due to the introduced complexity of such proposals, they have never been considered for deployment.

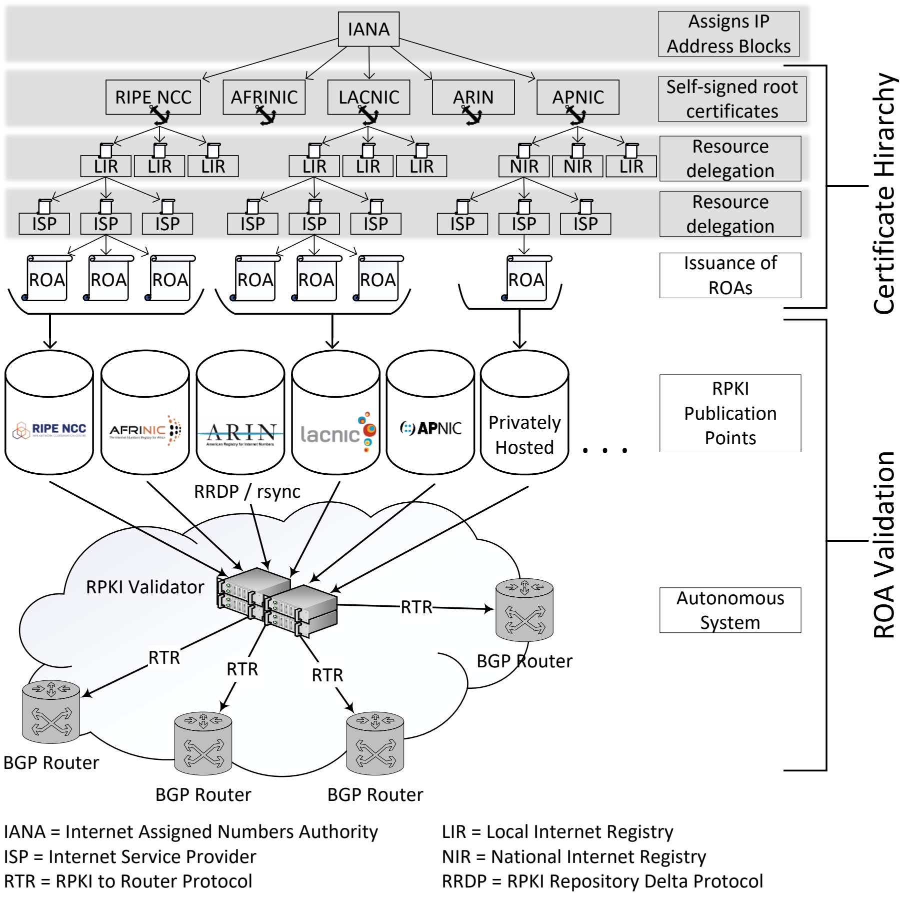


Figure 2.9: RPKI architecture. The IANA assigns resources to the five RIRs which act as trust anchors via self-signed root certificates. They delegate address space to ASes who publish ROAs. ROAs are stored in publication points and will be fetched & validated by RPKI validators. The result is provided to BGP routers.

Smaller subnets are delegated by each RIR via EE certificates and within its geographic region to a LIR or NIR. From there the process is repeated until typically an ISP or another AS of some sort that would like to actively use the address space is reached. The AS in the end of the chain uses the EE certificate to create a Route Origin Authorization (ROA) object to publicly announce their control over a prefix range. ROAs are published in RPKI publication points. A ROA contains, amongst other things, the following information:

Listing 2.8: Route Origin Authorization object

```
ROA: Prefix/Max-Length, ASN

For example:
ROA: 147.22.0.0/16-24, AS 47065
```

In the aforementioned example, AS47065 is authorized to announce any BGP route for the prefix `147.22.0.0` with a length of /16 to /24. Such an announcement would be rendered RPKI valid. Announcing `147.22.0.0/24` from AS1111 would be RPKI invalid, since the origin AS is not legitimized in the ROA. Announcing `147.22.0.0/24` from AS47065 would render RPKI unknown, given that no other covering ROA is issued. It should be noted, that the example ROA is non-minimal, since its max-length attribute allows for different subnet sizes. To protect unassigned or assigned, but unannounced address space, the RIRs and organizations can issue AS0 ROAs [77]. If a prefix range is included in an AS0 ROA, it should not be routed by any network.

The RPKI is based on a manual certification process. For each AS an operator needs to manually select other ASes they want to delegate resources to and those ASes need to publish ROAs. This process is implemented via digital X.509 certificates, called resource certificates [78]. It is complex, time-consuming and error-prone. It was recently proposed by Gilad et al. [79], [80] to move from a manual certification model to a de-facto ownership model. In de-facto ownership, an AS is considered the owner if it used the address space consistently over a long period of time. On the one hand, such change in semantics would significantly weaken the security guarantees of RPKI, at least temporarily until proper RPKI-signed ROAs are created. On the other hand, it would significantly speed up the process with which ROAs are created. Adoption of RPKI would likely increase much faster. Due to the described security drawbacks, the proposal has not been considered for adoption.

**Hosted vs. delegated model.** To meet the needs of different ASes, RIRs offer two separate models for resource delegation. Firstly, the hosted model, which offers a web portal in which RIR-members can easily log in to and manage their resources and create ROAs. Secondly, the delegated model, which allows for much more flexibility. However, the AS needs to run its own Certificate Authority (CA) and manage all resources by itself. It may or may not create a publication point. If only the CA is run, the RIR publication point can be used. This is called publish-in-parent and is currently implemented for production by RIPE NCC [81].

On the one hand, the hosted model abstracts most of the complexity but private keys remain with the RIR. Moreover, an AS needs to log into the different portals if it wants to manage resources issued by different RIRs. On the other hand, the delegated model serves as a single point of control and allows for more flexibility. Resources are managed within a single CA software, namely Krill [82] from NLNet-Labs and RPKI Toolkit [83] from Dragon Research Labs. Private keys do not leave the AS running the CA software. However, we should bear in mind that the RIR can at any time revoke any key, no matter where such key is stored. The AS is dependent on the RIR in any case. Considering these pros and cons, it might be worth the effort for larger organizations to run their own CA and publication point, while for smaller ASes the hosted solution seems to provide a better trade-off.

**Relying party software.** ROAs are fetched by so-called relying party software, also called RPKI validators, from the RPKI publication points. Relying Party (RP) software is available in multiple flavours: RPSTIR2 [84], OctoRPKI [85], Routinator 3000 [86], FORT-Validator [87], rpki-client [88], and rpki-prover [89]. The RPKI-Validator 3 [90] and Rcynic [91] have been discontinued. Some of these validators provide a web-GUI to manually check objects in the RPKI. RPKImancer[92] is designed to analyse RPKI objects on the command-line. RPKI data can be easily visualized via tools like RPKI MIRO [93] and RPKIVIZ [94].

The RPKI validators use either rsync or RPKI Repository Delta Protocol (RRDP) to pull the data from the repositories. rsync is an old protocol that is known to have some drawbacks: It consumes quite a lot of memory and CPU. This is particularly problematic as it makes rsync susceptible to DoS attacks. Additional problems are the lack of library support and the difficulty of publishing objects atomically. Therefore, the IETF pushes the replacement of rsync with RRDP in the future [95]. RRDP was designed to mitigate previously-mentioned shortcomings and supports Hypertext Transfer Protocol Secure (HTTPS) Content Delivery Network (CDN) infrastructure to increase resilience during content provisioning. It is a state-of-the-art protocol standardized by the IETF in 2017 as RFC 8182 [96].

Within one AS, the operator usually deploys at least two RP software instances for redundancy. The validation process is commenced in an *out-of-band* fashion. It does not consume BGP router resources. Once validation has finished, an output called Validated ROA Payload (VRP) is produced. This is a simple text file that contains all ASNs to prefix combinations that were cryptographically validated. The output can be easily digested by BGP routers. Transfer to BGP routers is performed via the RPKI to Router (RTR) protocol [97]. Proprietary protocol implementations as well as Open Source implementations, such as RTRlib [98], [99], StayRTR [100], and RTRTR [101] are available. Once the VRP list has been received by BGP routers, it can be used to perform BGP prefix origin validation [102], also called route origin validation. Routes can be flagged RPKI valid, invalid, or unknown. Local policies might de-preference RPKI invalid announcements or reject them. The problem with de-preferencing is that a more specific prefix hijack will still work due to a lack of alternative routes. Therefore, rejection of RPKI invalid prefixes is recommended.

A very important concern of operators is the stability of the routing infrastructure and that failure of components does not lead to outages. Outages would imply lost connectivity and as a result monetary loss and customer complaints. Therefore, the RPKI follows a soft-fail paradigm. Should RPKI components fail and therefore RPKI validation not be possible, routes always default to the unknown status. They will be handled by the BGP route selection process as if RPKI was not implemented. A short discussion about the soft-fail mechanism and the proposal of an alternative can be found in the ROVER approach [103]–[105]. Further shortcomings have been discussed by Geoff Huston [106].

RPKI-covered prefix space has been continuously increasing throughout the past 10 years. Du *et al.* [66] investigated data consistency between IRR and RPKI and found datasets to be inconsistent in 27.4% and 61.4% of cases for Réseaux IP Européens (RIPE) IRR and Routing Assets Database (RADB), respectively. The findings highlight the need for the RPKI as a cryptographically proven database with accurate information.

Du *et al.* [107] continued to investigate the level of conformity of participants of the MANRS project. Participants of MANRS are requested to implement several security mechanisms. One required activity is the registration of prefixes either in the IRR or the RPKI. The Internet Health Report (IHR) of the Internet Initiative Japan (IIJ) [108] was used as an underlying source of information. In May 2022, small and medium ASes participating in MANRS are more likely to originate only RPKI-valid announcements (60.1%) compared to non-MANRS participants (24.7%). MANRS participants also originate much fewer RPKI-invalid prefixes (23.6%) compared to non-MANRS participants (68.1%). For larger networks, the gap is much smaller. Large ASes participating in MANRS also propagate RPKI-invalid announcements at a lower rate.

## 2.5 PATH VALIDATION & PATH PLAUSIBILITY

### 2.5.1 Path Manipulations

Exact and more specific prefix hijacks can be mitigated by cryptographically supported origin validation. This is true for the previously-introduced scenarios when an unauthorized AS announces an IP prefix. The attacker therefore appears as the origin, which allows for origin validation algorithms to check if the origin is indeed allowed to make such an announcement.

**Artificial links.** Path manipulations are attacks in which the announcing AS changes parts of the AS path attribute. Such a change could be the insertion of an artificial link in the inter-domain routing topology such that the attacker claims to be directly connected to the legitimate origin. Figure 2.10 illustrates such a scenario.

Instead of announcing the prefix as the origin AS, $AS666$ claims to be directly connected to $AS1$. Since there is no check on the path and there is no relationship information available that can be used to properly identify whether $AS666$ would indeed have a peering link with $AS1$, the AS path is assumed to be correct. $AS1$ is the legitimate origin of the prefix `1.2.2.0/23`, therefore origin validation algorithms would render the announcement as RPKI valid. The path manipulation is successful.

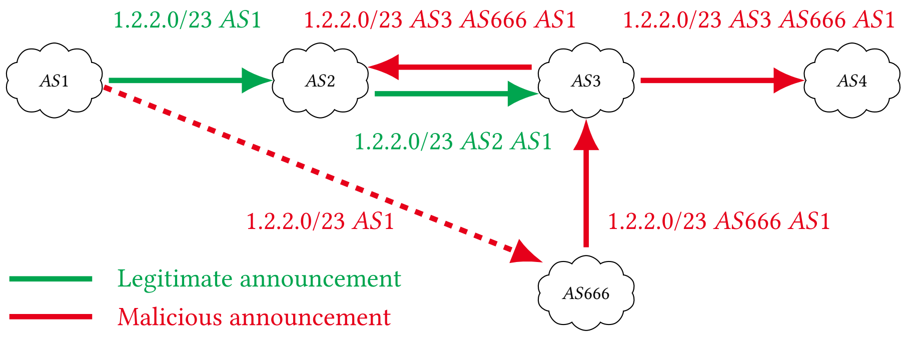


Figure 2.10: Path manipulation attack, also called forged-origin prefix hijack: *AS*1 is the legitimate origin, *AS*666 announces the same IP prefix. Since origin validation would filter the announcement of illegitimate *AS*666, the attacker creates an artificial link to the origin *AS*1 by manipulating the AS path attribute, in particular by inserting *AS*1 as the origin.

Depending on whether an exact prefix or a more specific prefix was used in the announcement during the path manipulation, the propagation will be different. For the exact prefix, propagation will be a result of the closeness of the targeted AS to one of the announcing ASes in the topology. The shorter the link, the more likely it is that an AS will fall victim to the attacker. Propagation of the more specific prefix will be global, as BGP uses the longest prefix match principle.

**Path prepending.** Aforementioned methods looked at path manipulation from an attacker perspective. It is, however, also used much more frequently as a legitimate traffic engineering tool [109], [110]. The longer an AS path, the more likely it is to receive a lower score by the receiving ASes. It will therefore be less likely to be chosen as a preferred route. Such mechanism is useful if an AS has multiple peerings but would like to preference one over the other. It does not, however, want to lose connectivity over the less preferred option. By prepending its own AS multiple times to the AS path, ASes can artificially increase the length of the AS path and achieve such de-preferencing, given that all other factors remain the same.

Consider the example in Listing 2.9. AS1234 has AS5555 as an upstream and AS1111 as a settlement-free peering relationship. In order to save money and attract more traffic via the settlement-free peer, AS1234 prepends its own ASN many times to the AS path to de-preference announcements sent over the upstream. At the same time it would send the regular announcement to the settlement-free peer. Since longer AS paths are usually less preferred, traffic would be shifted to the settlement-free peer and the origin AS saves money.

Listing 2.9: BGP path prepending

```
1.2.3.0/24  1111  1234  <-- Settlement-free peer
1.2.3.0/24  5555  1234  1234  1234  1234  1234  ...  <-- Upstream
```

A problem with this traffic engineering approach is the strict memory limitations of routers, since longer routes consume more memory. Moreover, if the AS path length exceeds 255, routers could crash. A prepending bug caused a worldwide outage in 2009 [111]. In this particular case, the operator inserted the ASN instead of the count that the tool expected in order to generate the path prepending output. Since the AS in question was AS47868, the count was a multiple of what was possible for the routers to process. Internally, one Byte was used to represent the expected count value and no boundary checks were in place. Entering 47868 led to a modulo 256 operation, which resulted in 252 as a final number. Hence, AS47868 got prepended 252 times to the AS path. Cisco IOS routers had a bug that caused a crash when exceeding path lengths of 255, leading to an outage with a severe stability impact on the inter-domain routing infrastructure. The IETF is currently working on path prepending guidelines to avoid such problems in the future [112].

**Path shortening.** When path prepending is used excessively, it is easy for an attacker to craft announcements that appear meaningful, using path shortening. The shortened AS path would have a regular length compared to the much longer legitimate path. Listing 2.10 provides an example of a path shortening attack.

Listing 2.10: BGP path shortening

```
1.2.3.0/24  1111  1234  1234  1234  1234  1234  1234  1234  1234  1234  1234  1234
            1234  1234  1234  1234  1234  1234  9999  <-- Prepended route
1.2.3.0/24  666  1111  1234  9999  <-- Artificially crafted route
```

AS9999 originates `1.2.3.0/24`. AS1234 performs excessive path prepending to de-preference the path for some reason. The attacker AS666 can easily craft an announcement that would be preferred since it provides a much shorter path.

### 2.5.2 Path Validation

The RPKI provides origin validation and is designed to bind IP prefixes to AS numbers in a cryptographically secure manner. It allows operators to efficiently mitigate exact and more specific prefix hijacks. However, any attack in which the announcement carries the legitimate origin AS in the very beginning of the as path attribute will validate properly within RPKI and go unnoticed. To be precise, an attacker could arbitrarily change the AS path, *i.e.,* inject or delete AS numbers, as long as the origin remains the same. Recently, an attacker hijacked Amazon's address space by forging the AS path and claiming to be an upstream of an Amazon ASN [113], [114]. To detect and mitigate such path manipulation attacks, the AS path itself needs to be validated on the receiving end. Figure 2.11 illustrates where origin validation and path validation are yielding benefits.

Butler *et al.* surveyed BGP security issues and mitigation methods in 2009 [1], followed by Huston *et al.* in 2010 [2]. Both are very comprehensive survey papers that

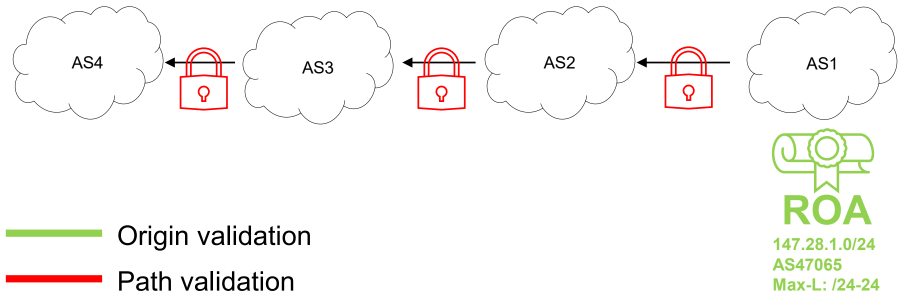


Figure 2.11: Origin validation binds resources to ASNs. Path validation ensures that an announcement took the path contained in the BGP AS path attribute. Path validation requires origin validation to work properly.

provide excellent summaries on the topic. We highlight the most relevant proposals in the following sections and provide a comparison of BGP security technologies in Section 2.6.

### Secure BGP (S-BGP)

Secure BGP (S-BGP) was proposed by Kent *et al.* in 2000 [115]. It was never implemented as proposed, but laid the foundation for the development of BGPsec, which we explain later in great detail. It is therefore important to understand how S-BGP was designed in order to understand the later standardized BGPsec. S-BGP provides several security guarantees [2]. It ensures that

1. the integrity of all UPDATE messages in transit between BGP speakers is guaranteed.
2. the UPDATE message was indeed sent by the indicated peer.
3. the information sent is more up-to-date than earlier received versions.
4. the UPDATE message was intended for that particular receiver.
5. the BGP peer is authorized to send the UDPATE message on behalf of the peer AS.

To achieve these security goals, it leverages IPSec to protect data integrity of BGP sessions, is based on a central trust model in which two Public Key Infrastructures (PKIs) certify the ownership of IP prefixes and ASNs, and introduces a new BGP attribute called *attestations*. Both PKIs are rooted at Internet Corporation for Assigned Names and Numbers (ICANN). The first PKI is used for address allocation, which is now implemented in a different form by the RPKI. It testifies as to who is allowed to use which prefix ranges [116]. A second PKI is used for ASN assignment. Issuing certificates for ASNs is a two step procedure: Firstly, ASNs are assigned by ICANN. Secondly, the organization that had the ASN assigned creates and signs a certificate binding the ASN to a public key. The workflow has the disadvantage that if *e.g.,* the name of an organization changes, a new certificate would have to be issued [117]. Each BGP speaker relies on a distinct public key which is reflected in a subordinate certificate.

**Route attestations.** X.509 certificates are used to prove ownership and are communicated to routers in an *out-of-band* fashion, while route attestations are done with digital signatures in an *in-band* fashion within the new *attestation* attribute. Since digital signatures of all nested route attestations within received UPDATE messages have to be checked during the verification procedure, and outgoing BGP UPDATE messages have to be signed, additional strain is put on routers to perform the cryptographic operations. This is considered a significant drawback. Previously in the year 2000 it was questionable whether the performance capabilities of routers were sufficient to run S-BGP in production [2], [118]. Performance of devices has significantly increased since then, however, but the discussions and arguments against BGPsec putting additional load on routers using *in-band* processing, which we focus on later in Section 2.5.2, have not changed.

The following items are required [115] to validate that the origin is authorized to make such an announcement, which implements *origin validation*, and that each subsequent AS on the as path is authorized to advertise the route received, which represents the *path validation* part:

- 1 address attestation from each organization owning an IP prefix.
- 1 address allocation certificate from each organization owning an IP prefix.
- 1 route attestation from every S-BGP speaker (or its AS) along the path.
- 1 certificate for each S-BGP speaker along the path to check the signatures on the route attestations.

**Unsupported partial adoption.** Other drawbacks of the design of S-BGP are unsupported partial adoption scenarios. There will not be a flag day in which all routers on the Internet suddenly support a new technology. Therefore, proposed solutions need to take into consideration that partial adoption is a mandatory case that needs to be supported. Since S-BGP is designed to provide strong cryptographic proof that the whole AS path was not altered and therefore requires each hop within the as path to provide a route attestation, it is incompatible by design with partial adoption scenarios in which only a single router does not support the new technology.

### Pretty Secure BGP (psBGP)

Pretty Secure BGP (psBGP) was proposed by van Oorshot *et al.* in 2005 [117]. Just like its predecessor S-BGP, it uses IPSec to protect the integrity of BGP sessions. It also uses a centralized trust model, in particular a single-level PKI, for ASN authentication. However, CAs at the top of the hierarchy are the RIRs, not ICANN. The RIRs are expected to cross-sign their public keys. Each AS creates and signs a SpeakerCert, SessionCert, and a prefix assertion list. They are independent of the name of an organization and as long as the ASN remains the same, there is no need to reissue a certificate. Also, BGP speakers share a single public key. As a consequence, psBGP has less of a certificate management overhead compared to S-BGP.

The prefix assertion list is an ordered list and contains prefix assertions for the AS itself, as well as endorsements of IP prefixes for neighbours. In contrast to previously published BGP security proposals, *i.e.,* S-BGP and Secure Origin BGP (soBGP), a decentralized model for verifying IP prefixes is deployed based on the prefix assertions that are received by neighbours. To validate prefix assertions, a rating mechanism is proposed. Each AS rates every other with a value between 0-1. RIRs are expected to be trusted, therefore receive a rating of 1, most ASes should be neutrally trusted with 0.5. The summary of resulting values provides heuristics as to what degree an announcement can be trusted. psBGP is offering a possibly simpler operation regarding prefix ownership attestation at the expense of security guarantees. With RPKI currently being deployed it is more than questionable whether a decentralized approach would indeed have been simpler.

psBGP uses the same approach as S-BGP for AS path verification, except that the bit-vector method by Nicol *et al.* [119] is deployed. It therefore also inherits the biggest drawback, namely the increased *in-band* processing capacities needed on routers.

An important element that was a driving factor for its development was the need to support incremental (partial) deployment. psBGP supports such partial deployment, but is described by Huston *et al.* as "needlessly complex and bears much of the characteristics of making a particular solution fit the problem, rather than attempting to craft a solution within the bounds of the problem space" [2].

### Border Gateway Protocol Security (BGPSec)

The first Border Gateway Protocol Security (BGPsec) specification was published in 2011 [120] by Matt Lepinski and underwent further development during the following years. It was finally standardized in 2017 as RFC 8205 [11]. Many people contributed to the standardized design. There are several other associated RFCs *i.e.,* RFC 8206–RFC 8209 and RFC 8654, which detail parts of the algorithm and necessary components [38], [121]–[124].

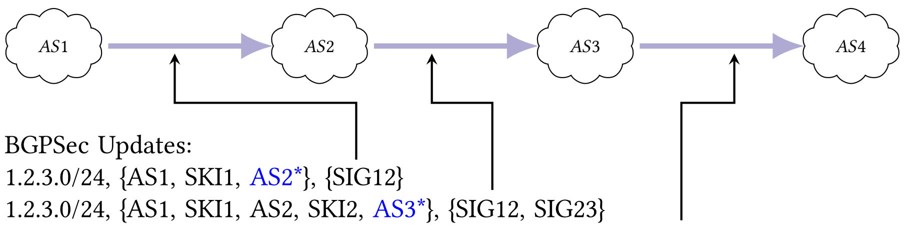
BGPSec was created based on the learning from prior work in the field of path-validation *i.e.,* S-BGP [115], soBGP [125], and psBGP [117], [126], amongst other work. In contrast to previously-mentioned methods, BGPSec no longer needed to solve the problem of origin validation. It was standardized after the RPKI was already deployed and is designed to work in conjunction with it. The RPKI provides a rich infrastructure and thoroughly-designed architecture which has proven its stability and advantages in practice. Developing a security solution on such a basis seems to yield more fruitful results as complexity is reduced by not trying to solve all problems at once.

**Design goals.** BGPSec aims at providing strong cryptographic proof that every AS within the AS path of an UPDATE message has explicitly authorized the route to the subsequent AS in the path. An AS can therefore be sure that the announcement traveled exactly the way it is described in the UPDATE message. To provide such strong security guarantees, verification of incoming announcements and forward-signing of outgoing announcements is required. A threat model for BGP is presented in RFC 7132 [127]. It details threats such as man-in-the-middle attacks, path manipulations, and compromised router private keys, amongst other things. BGPSec is about formal verification, not routing policy violations, hence it is not capable in its original version to detect route leaks.

**Prerequisites.** It is assumed that the BGPSec capability was negotiated between two peers. Amongst other things, it involves increasing the 4,096 octets Protocol Data Unit (PDU) limit for updates to 65,535 octets [38]. The AS path attribute is replaced by the potentially much larger BGPsec_Path attribute [11].

**Forward Signing.** In our example in Figure 2.12, AS1 wants to propagate a BGPSec update message to AS2. Firstly, the eBGP router within AS1 creates a hash over {1.2.3.0/24, AS1, AS2} to obtain a hash value. SHA-256 [129] can be used for that purpose. Secondly, the router uses its private key to create a cryptographic signature Sig12 over the previously-obtained hash value via the ECDSA-P256 algorithm. If key management within the AS seems too complicated to provide for dedicated private keys per eBGP router, the same private key (AS key) can be used. Thirdly, the update is propagated including the following information: 1.2.3.0/24, {AS1, SKI1},{SIG12}. The first item in the list is the prefix itself. The second item is the list is the AS path, which also includes a Subject Key Identifier (SKI) for each AS. The SKI points to the public key which is later required for verification of the submitted signature. The third item is the signature section. Each AS that signs and forwards the announcement attaches its own signature to the existing ones. Sriram and Montgomery formally describe the process as [128]:

$$\begin{gathered} P, [AS_1, SKI_1, AS_2, SKI_2, ..., AS_{n-1}, SKI_{n-1}, \\ AS_n, SKI_n, AS_{n+1}], [Sig_{12}, Sig_{23}, ..., Sig_{n-1,n}] \end{gathered} \tag{2.1}$$



BGPSec Updates:
1.2.3.0/24, {AS1, SKI1, AS2*}, {SIG12}
1.2.3.0/24, {AS1, SKI1, AS2, SKI2, AS3*}, {SIG12, SIG23}
1.2.3.0/24, {AS1, SKI1, AS2, SKI2, AS3, SKI3, AS4*}, {SIG12, SIG23, SIG34}
* Next hop AS is signed over but not included in the forwarded BGPSEC update

Figure 2.12: Forward Signing in BGPSec, modelled after [128]

It is important to point out that the information that $AS_n$ signs over includes the next AS hop, *i.e.,* $AS_{n+1}$, hence the term forward signing. But the next hop AS is not included in the forwarded AS path:

$$\begin{aligned} P, [AS_1, SKI_1, AS_2, SKI_2, ..., AS_n, SKI_n], \\ [Sig_{12}, Sig_{23}, ..., Sig_{n,n+1}] \end{aligned} \tag{2.2}$$

**Signature verification.** Once an AS receives a BGPSec message, the validation procedure starts. It terminates either with valid or invalid. The procedure contains many checks of the BGPsec_PATH that can be found in detail in RFC 8205 [11]. We limit ourselves here highlighting the workings of the verification of the signature block. Each signature in the signature block is validated via the public key of the signing AS, found under the location included in the SKI. The validation procedure starts with the most recently-added segment and concludes with the least recently-added segment. If no algorithm suite is found to match the one supported by the validating BGP speaker, the unsecured AS path is reconstructed from the secure AS path, the transmission is stripped of any BGPsec content, and the update treated as if it were a legacy BGP transmission.

**Drawbacks.** BGPsec mitigates path manipulation problems introduced in the beginning. The operation of BGPsec, however, comes with some drawbacks that need to be taken into consideration when deployment of this security solution is evaluated. In contrast to RPKI, BGPsec works in an *in-band* fashion. The eBGP router is required to validate the chain of signatures of incoming BGPsec updates and needs to forward sign the outgoing update message. This requirement introduces significant processing overhead that impacts how many updates can be processed by currently-deployed routers in a given amount of time. Sriram and Montgomery [128] worked on minimizing the cryptographic processing overhead of BGPsec, while Bagdonas [130] further improved BGPsec performance. However, *in-band* processing is conceptually required and therefore remains a problem.

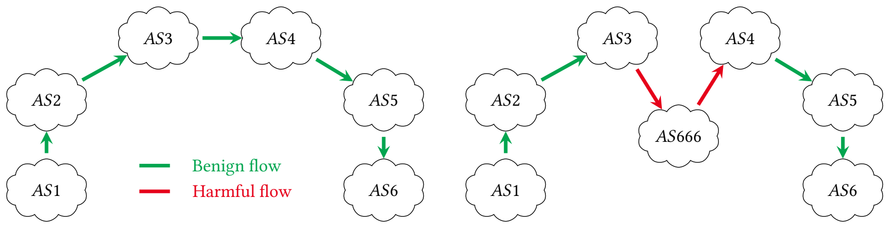
BGPsec also only works if each router along the way supports the security extension. If only a single eBGP speaker is not capable of conducting the verification and signing procedures, the whole chain becomes broken. As a result, BGPsec is downgraded to regular BGP communication. Therefore, it is not really applicable in a *partial deployment* scenario. There will not be a flag day during which everyone switches their BGP security extension on. However, incremental deployment is possible. The more eBGP speakers support BGPsec, the higher the likelihood that a short path lies only within the protected range of eBGP speakers. Such routes would be fully protected. Since large transits are present in a very high percentage of BGP paths, transits would need to support BGPsec deployment first to enable smaller stub ASes to participate. Lychev *et al.* [131] pointed out that precisely because of this, BGPsec and S-BGP have limited security benefits over the RPKI.

**Outlook.** According to Ignas Bagdonas, global end to end deployment is not realistic but we are likely to see limited domain deployments [132]. This is a very optimistic view, given the drawbacks that have been detailed before. Huston *et al.* state that: "In practical terms this is an unlikely scenario, and the current experience with the uptake of a revised version of BGP that supports 32-bit AS number values suggests that the public Internet has considerable inertia and is very resistant to adopting changes to BGP" [2]. Firstly, hardware would have to be developed and optimized for BGPsec deployment. Secondly, an AS would need to manage the complexity of key management and BGPsec configuration. Thirdly, at least two peers need to agree to deploy BGPsec or possibly more when longer paths are intended to be protected. We therefore believe that our focus of attention should lie on *out-of-band* solutions instead.

### 2.5.3 Route Leaks

Routing within the inter-domain infrastructure commonly follows the Gao-Rexford model [44], also known as valley-free routing, see Section 2.2, in particular Figure 2.4. A route received by a customer is forwarded to all peers regardless of relationship, a route learned from a peer is only forwarded to customers, and a route learned from a provider is only forwarded to customers.

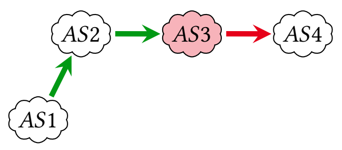
Such a rule set is summarized as routing policy. Previously-introduced attacks focused on illegitimate announcements and path manipulations, hence they used BGP actively in an unintended way. In contrast to these attacks, route leaks [133] occur unintentionally as they usually cause a monetary loss for the offending party. ASes do not announce prefixes to specific peers, as they would otherwise have to pay for traffic that does not serve their own customers. Figure 2.13a illustrates proper routing behavior, while Figure 2.13b illustrates a routing valley. *AS*666 is paying both upstreams, *AS*3 and *AS*4, for forwarding their traffic. The upstreams would always prefer a customer route as it yields monetary reward. However, *AS*666 does not gain any benefit in this scenario and pays for forwarding the traffic between both upstreams. This is a very unfortunate scenario that *AS*666 seeks to avoid. Such a scenario is called a route leak.



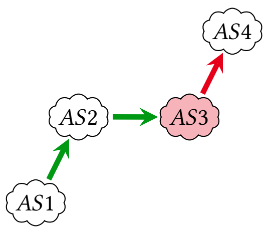
(a) Valley-free routing (b) Routing valley

Figure 2.13: Routing behavior according to the Gao-Rexford model. Valley-free routing represents proper behavior, while a routing valley constitutes a policy violation and results in monetary loss for the leaking AS.

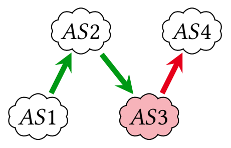
Based on the previously introduced rules in Section 2.2, in particular Figure 2.4, we are able to determine four possible scenarios in which anomalies happen. We illustrate them in Figure 2.14.



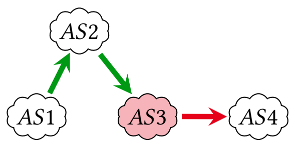
(a) AS3 leaks route from peer to peer



(b) AS3 leaks route from peer to provider



(c) AS3 leaks route from prov. to provider



(d) AS3 leaks route from provider to peer

Figure 2.14: Route leak scenarios

1. A route received from a peer is forwarded to another lateral peer (a).
2. A route received from a peer is forwarded to a provider (b).
3. A route received from a provider is forwarded to another provider (c).
4. A route received from a provider is forwarded to a lateral peer (d).

More details regarding the different route leak classifications can be found in RFC 7908 [133].

### 2.5.4 Path Plausibility

Path plausibility algorithms provide slightly fewer security guarantees compared to path validation algorithms. This is because in path validation the AS path is validated to be exactly the way it is specified in the AS path attribute, *e.g.,* BGPSec. In path plausibility, a graph of allowed paths is constructed via external relationship information. The AS path is then checked as to whether it lies on an allowed path or not. There is no cryptographic proof that the announcement traversed the specified ASes in the AS path. However, if the presented path lies on an allowed path, it is fine to parse the announcement as each neighbour has been authorized by its peers. Using this principle, path plausibility algorithms are able to spot forwarding paths in announcements that have not been cleared by the respective neighbours and cannot therefore be verified. Since path plausibility algorithms rely on relationship information, they can also be used to mitigate policy violations, *i.e.,* route leaks. A known drawback that applies to all path plausibility algorithms is requiring ASes to publish relationship information. This information is sometimes perceived

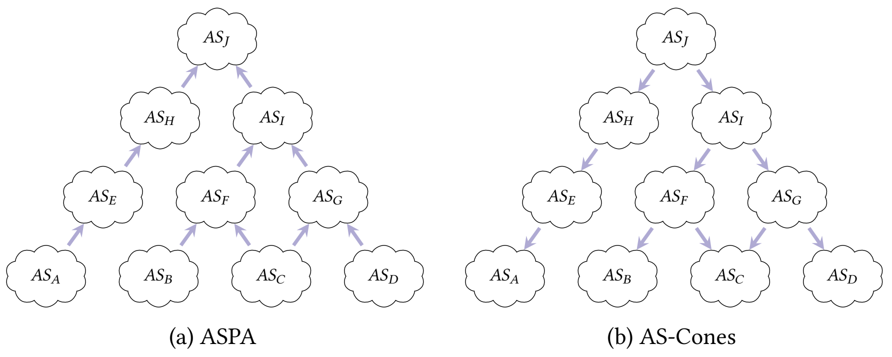


Figure 2.15: Comparison of ASPA and AS-Cones object creation

as business critical information and an AS might not want to publicly share this information. Precisely this sharing is required for the algorithms to work. Therefore, not only technical, but also marketing efforts are required to convince operators to participate in such new security proposals.

### Secure Origin BGP

soBGP was proposed by White *et al.* within the IETF in 2003 [134] and is the first proposal in the path plausibility domain. Just like S-BGP it uses IPSec to protect BGP session integrity. Moreover, a PKI is used for verifying IP prefix ownership. The concept for ASN authorization is, however, different. soBGP makes use of a web-of-trust model for authenticating AS public keys. The level of trust is defined via a metric calculated per AS adjacency. Each AS publishes a list of its AS neighbours, signed by the local AS. The local AS only uses a single public key for all BGP speakers. Since a BGP peering session consists of two participants we observe the following cases: If there are both, (X,Y) && (Y,X) then the pair is trustable. If there is only one pair (X,Y) || (Y,X) it is less trustable.

IXPs were considered a significant problem for soBGP as they are transparent on the network layer and there is no information published on adjacencies between the IXP clients. Therefore an announcement received via an IXP peer would be less trustable. This is a disadvantage.

soBGP can be used to filter bogon routes, but cannot be used to filter route leaks as there is no policy component integrated. The algorithm allows *out-of-band* processing and detached filter management. Moreover, partial adoption scenarios are supported.

### Autonomous System Provider Authorization (ASPA)

A recent variant is called Autonomous System Provider Authorization (ASPA). ASPA is currently heavily discussed within the IETF [135], [136]. It is designed to prevent path manipulation attacks and mitigate route leaks. Additionally, it is able to mitigate the forged-origin prefix hijack. The first IETF draft was proposed in 2019, with lively discussions on mailing lists and during RIPE & IETF meetings since then. The need to find a security solution that solves the problems introduced in the previous sections, without relying on in-band processing and being partially deployable is apparent.

ASPA requires relationship information to work properly. A Customer Autonomous System (CAS) signs and publishes a single ASPA object containing information about its upstream providers, called Provider Autonomous System (PAS), and non-transparent route servers at IXPs it peers with. If no upstream provider exists, the ASPA object would contain AS0 instead. Only mutual transits, but not lateral peers, are included in the objects. ASPA objects are published within the RPKI repositories and conform to the template for signed RPKI objects [137]. Using the RPKI with its existing infrastructure seems to be a very reasonable solution, since adoption of ASPA will be much easier and quicker.

Listing 2.11: Autonomous System Provider Authorization object

```
ASPA: CustomerASID, ProviderASSet

For example:
ASPA: AS47065, {{AS3130, IPv4}, {AS20940}, {AS714}}
```

We illustrate the (artificially crafted) content of an ASPA object in Listing 2.11. We removed components such as version and object identifier, amongst other things, for clarity. Essentially, it carries the CAS number followed by a sequence of PAS numbers. Optionally, an IP version can be specified per PAS to limit authorizations to a certain address family. Our example shows three legitimized PASs, with AS3130 only being allowed to forward IPv4 prefixes. Instead of inserting each and every AS manually by hand, AS-Sets can be used, see Section 2.4.2.

Since the RPKI infrastructure is used to hold and distribute ASPA object, RPKI validators are expected to be extended to also validate ASPA objects. The CA-software Krill [82] already supports ASPA object creation, while the latest version of the RP-software Routinator [138] supports ASPA validation. Moreover, several prototype implementations already exist. The process is equivalent to RPKI objects. First of all, the cryptographic validity of all objects themselves needs to be checked. To guard against a maliciously-acting neighbour that might remove their own ASN from the AS path, the receiving AS needs to check that the last AS in the AS path contains the relevant neighbour. Once these checks have passed, the content of the AS path is validated on a logical level. The outcome of the validation procedure is RPKI-like: valid, invalid, or unknown. The procedures themselves for upstream and downstream paths are precisely described in [136].

The ASPA objects are accumulated and their content is represented as a tree in Figure 2.15a. For each link between two ASes in the AS path, the described relationship in the constructed tree is checked. If a hop is unauthorized, *i.e.,* an ASPA object exists but the provider is not listed, the procedure halts with invalid. If a hop could not be found since an ASPA object is missing, the outcome is unknown. If all hops could be found and their order respects the Gao-Rexford model, the outcome of the procedure is valid. It is recommended that invalid routes are rejected. Valid paths should be preferred over unknown.

ASPA is not able to protect against prefix hijacking of their own upstream, since the upstream is legitimized by the customer and origin validation would also validate when the prefix matches the legitimized origin. However, such an attack is highly unlikely as there is a business relationship that the upstream benefits from and does not want to endanger. An accidental hijack, however, would remain a possible threat.

### AS-Cones / ASGroups

Another recent proposal is AS-Cones [139]. It was originally proposed to provide a secure alternative to the insecure ASSET objects but was quickly reused as the contained relationship information are designed for use as a path plausibility algorithm. The problem with ASPA is that it requires each and every leaf AS to issue an ASPA object. RPKI has the same requirement and continues to get deployed, but efforts to deploy the technology have been under way for more than a decade. In order to speed up deployment of path plausibility algorithms, AS-Cones takes an opposite approach regarding the issuance of attestations. It requires providers to create AS-Cone objects, containing the customers, see Listing 2.12.

Listing 2.12: AS-Cones object

```
AS-Cone: ProviderASID, CustomerASSet

For example:
AS-Cone: AS47065, {AS1234, AS20940, AS714}
```

Considering a BGP dump from all RouteViews [140] collectors for 24 hours on October 1, 2022, we obtain a graph with 74.110 ASes and classify each link using the CAIDA AS relationship dataset (as-rel2) [141]. We observe that the vast majority of ASes (62768) are stubs, without peering or customer relationships. While ASPA requires involvement from each of these ASes, AS-Cones does not have such a requirement. Instead, only the providers would be involved in the issuance of cryptographic objects. Since larger ASes are more likely to have a greater workforce, a 24/7 Network Operation Center (NOC), and more specialized security personnel, a quicker adoption is assumed.

Quicker adoption is realized at the expense of security. AS-Cones relies on relationship information and is capable of performing the same detection tasks as ASPA. It detects accidental prefix hijacks and route leaks. However, due to its inversion of cryptographic attestations, a significant security drawback is introduced: AS-Cones is not capable of providing security against the BGP forged-origin hijack attack introduced in Figure 2.10. At first, the attack itself would be detected. But any provider could add artificial customers in their AS-Cone, therefore rendering such an attack as valid. Since the provider signs and publishes the AS-Cone object, it is the only one in control regarding the content. Also, in contrast to ASPA, there is no business relationship between the misbehaving provider and the victim. The victim would be a random AS that is not a customer of the provider.

Because of this security drawback, we consider AS-Cones as an intermediate step towards the deployment of ASPA. Deployment of AS-Cones can be much quicker and help to provide greater security compared to RPKI in a short time frame. In the long run, however, ASPA would need to be deployed in order to mitigate the forged-origin prefix hijack attack.

A very recent proposal, called ASGroups, modifies the AS-Cones concept slightly by improving upon the ASN.1 formal notations, simplifying the validation concept, and changing the opt-out behavior [142]. One thing to consider is that operators are sometimes hesitant to share too many details about their customer cone since this is business-critical information. This might be a disadvantage in the adoption process for algorithms relying on such information.

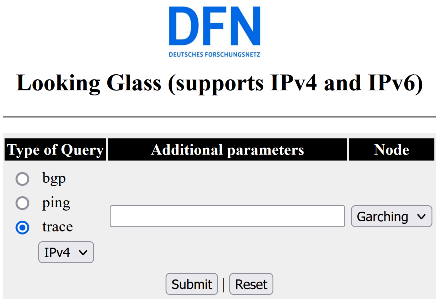
## 2.6 COMPARISON OF SECURITY SOLUTIONS

In this section we provide a comparison of previously-introduced security mechanisms.

Table 2.1: Comparison of BGP security extensions

| | Origin validation | | Path validation | | | Path plausibility | | |
|---|---|---|---|---|---|---|---|---|
| Characteristic | IRR | RPKI | S-BGP | psBGP | BGPSec | soBGP | ASPA | AS-Cones |
| Incidental hijacks | ✓ | ✓ | ✘ | ✘ | ✘ | ✓ | ✘ | ✘ |
| Intentional hijacks | ✘ | ✓ | ✘ | ✘ | ✘ | ✓ | ✘ | ✘ |
| Path manipulations (except origin manipulation) | ✘ | ✘ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Path manipulations (including origin manipulation) | ✘ | ✘ | ✓ | ✓ | ✓ | ✓ | ✓ | ✘ |
| Route leaks | ✓ | ✘ | ✘ | ✘ | ✘ | ? | ✓ | ✓ |
| Partial deployment | ✓ | ✓ | ✘ | ✘ | ✘ | ✓ | ✓ | ✓ |
| Out-of-band processing | ✓ | ✓ | ✘ | ✘ | ✘ | ? | ✓ | ✓ |
| Complexity | ↓ | → | ↑ | ↑ | ↑ | ↑ | → | → |
| Adoption likelihood | ↑ | ↗ | ↓ | ↓ | ↓ | ↓ | ↗ | ↘ |

Incidental hijacks carrying the wrong origin ASN are fairly easy to detect. The IRR provides prefix to ASN information in a cryptographically insecure manner, the RPKI provides the same set of information in a cryptographically secure way. soBGP would also have been capable of detecting incidental hijacks as it was also intended to provide a secure prefix to ASN mapping. Intentional hijacks are harder to mitigate as the IRR is insecure and a sophisticated attacker could simply add an entry to an IRR database that legitimizes the invalid origin. Cryptographic proofs within the RPKI eliminate such a possibility. Path manipulation attacks need more advanced path validation or path plausibility algorithms in order to be mitigated. Path validation offers a higher level of security since an announcement traveled the exact same way as shown in the AS path. Within path plausibility it is sufficient to create a tree of adjacencies that legitimizes the path an announcement took. However,



(a) Web interface

```
Router: Garching
#show bgp 137.193.0.0/16

Mon Jan 15 15:04:12.381 CET
BGP routing table entry for 137.193.0.0/16

Last Modified: Dec  7 13:12:23.592 for 5w4d
Paths: (3 available, best 1)
  Advertised IPv4 Unicast paths to peers:
    188.1.37.90 188.1.239.86 188.1.243.94 ...
  Path #1: Received by speaker 0
  Advertised IPv4 Unicast paths to peers:
    188.1.37.90 188.1.239.86 188.1.243.94 ...
  65208
```

(b) Result (truncated)

Figure 2.16: DFN Looking Glass

there is no cryptographic proof that the information is correct. For path validation, each AS needs to participate, and for path plausibility AS relationship information is required. The only difference here is that ASPA is capable of preventing AS origin forgery, while AS-Cones is not capable of preventing such attack. Route leaks can be detected and mitigated by either IRR filters (not cryptographically secured) or path plausibility algorithms (cryptographically secured), both relying on AS relationship information.

Partial deployment is not possible with path validation algorithms, as paths only get secured once each and every eBGP speaker along the path participates in the verification and signing process. Out-of-band processing is implemented by most modern algorithms and is expected to significantly increase the likelihood of adoption.

## 2.7 MEASUREMENT TOOLS AND PLATFORMS

Within this chapter, we use many measurement tools and frameworks on the control plane as well as data plane. On the control plane we use Looking Glasses, Route Collectors, and the Peering Testbed, while on the data plane we make use of RIPE Atlas. These require an introduction to fully understand the methodology explained at a later stage.

**Looking Glasses.** Whenever changes on the control plane, namely within BGP, are made, we want to understand whether such changes were deployed as intended or if unforeseen behavior occurred. If the changes had not been deployed as intended, measurement results would be impacted or possibly invalidated. It is therefore crucial to be able to check control plane information in ASes under test or at upstream to know whether announcements propagate properly. This is challenging, since the majority of ASes do not offer public access to their internal systems. Some ASes, especially upstreams and larger transits, thankfully offer such services. They are called *Looking Glasses*. It is public access to an internal BGP router that allows queries to the RIB. Using that access it is possible to see if an announced route is

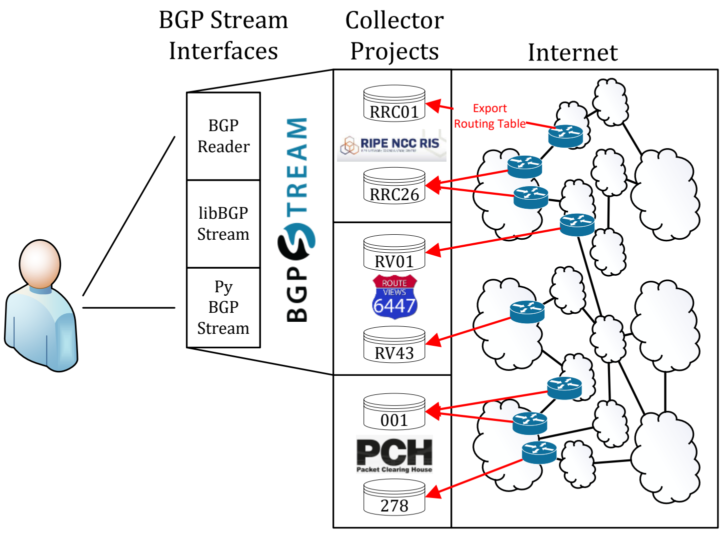


Figure 2.17: Route Collector Architecture

received at all, preferred by the router, or discarded. Unfortunately, the access is manual via a web interface, as shown in Figure 2.16a. In addition, the output in the form of RIB entries cannot be retrieved in an automated manner and requires manual analysis, see Figure 2.16b. The queried prefix range `137.193.0.0/16` belongs to the Universität der Bundeswehr München, but it does not have its own AS registered with IANA and is instead represented towards the inter-domain routing infrastructure by Deutsches Forschungsnetz (DFN). We observe only one path towards that range announced by AS65208 within the DFN looking glass. AS65208 is a private ASN, according to RFC 6996 [143].

There are many looking glasses available [144]. Some major router manufacturer such as Cisco, Juniper, etc., provide a looking glass feature, but the interfaces are different and there is no standardized means of accessing these looking glasses altogether. The Center for Applied Internet Data Analysis (CAIDA) has developed a wrapper project that allows to access a multitude of looking glasses provided by different ASes via a single interface. Periscope [145] provides an API to retrieve measurements status, search by type of measurement, and create new measurements. It is a powerful tool that has saved many hours of manual debugging work for operators.

**Route Collectors.** Since looking glasses are a very limited way of gaining access to routing information and require many hours of manual debugging, they are not suited for automated measurement campaigns. Instead, many researchers rely on *Route Collector* projects.

Route collector projects are geographically-distributed BGP peers that collect and store BGP RIB dumps and BGP update messages. The route collector projects require the collaboration of operators as the data that is fed into the route collectors needs to come from ASes in the wild. Since debugging and monitoring becomes much easier when automated, several ASes participate. The first collector service is called RIPE Routing Information Service (RIS) [146] and was established in 1999. It provides access to the data either via RIS live, a real time BGP streaming API, RIS Raw data that allows fetching of data dumps in Multi-Threaded Routing Toolkit (MRT) format, and RISwhois. The University of Oregon Route Views Project [140], founded in 1997, has 35 BGP collectors peering with 342 ASes which held a total of 321.163.844 routes, as of March 2022. The Packet Clearing House (PCH) [147] collects BGP data from 239 IXPs around the world, as of March 2022. The service has been operational since 2011 and allows the downloading of BGP data in MRT format. They also provide daily snapshots of the 'show ip bgp' command for each route collector. Isolario [148] appears to have been discontinued as neither website nor BGP repositories work anymore. Running such a collector project is a time consuming and money intensive undertaking.

Each collector project maintains its own infrastructure and relies on ASes to export their BGP RIB to the collectors. The number of peers that export routing information is key. The more data is collected by a certain collector instance, the more valuable the data becomes as the view of the Internet becomes less biased. Based on that information, snapshots are provided via web portals.

A user is (in most cases) able to access the collector data for free. Since there are multiple collector projects, each with its own benefits, a researcher would have to connect manually to each of them (and there are hundreds) to fetch required data for analysis. Most provide telnet access for live debugging and an archive to download MRT files. This procedure is usually automated by every researcher on their own. Scripts for scraping data might be shared together with publications but are mostly tailored towards the researchers' need. Therefore, a project called BGPStream was initiated by CAIDA based at the University of California's San Diego Supercomputer Center [149].

It provides several interfaces to the data from RIPE RIS and Routeviews: a command line tool; BGPReader [150]; a Python library PyBGPStream [151]; and a C-library libBGPStream that allow users to automate measurement processes. To obtain BGP data via BGPReader for the prefix `137.193.0.0/16` between start and stop is as easy as:

Listing 2.13: BGP Reader command

```
bgpreader -m -k '137.193.0.0/16' -w 1621468800,1621476000
```

**The PEERING Testbed.** The PEERING testbed [152], which is run by participants from the Columbia University, University of Southern California, and Universidade Federal de Minas Gerais, provides a platform for researchers to participate in the inter-domain routing infrastructure. Instead of performing measurements on historic data that cannot be controlled by the measuring party, the PEERING testbed allows the announcement of BGP prefixes. It therefore enables for active experiments. It has several IP ranges reserved for use within experiments (184.164.224.0/19, 138.185.228.0/22, 204.9.168.0/22, and 2804:269c::/32) and owns several ASNs (47065, 61574, 61575, 61576, 263842, 263843, 263844, 33207). All peering

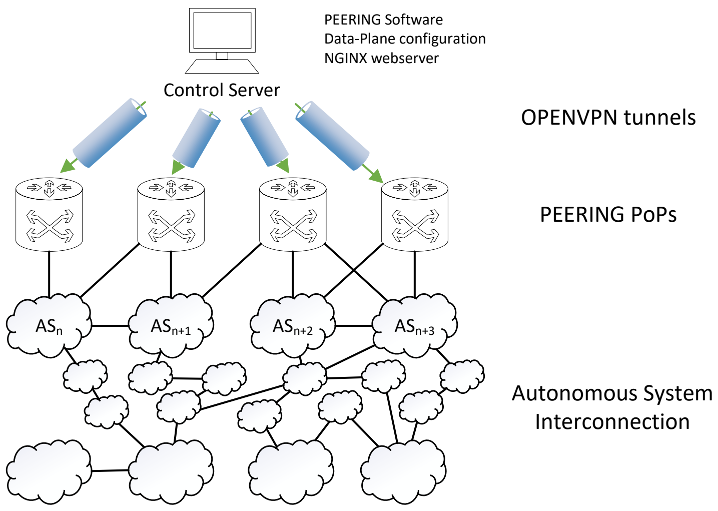


Figure 2.18: PEERING Testbed Architecture

sessions are established via AS47065. However, traffic forwarding capacities are quite limited and project proposals including things such as Internet-wide scans have to be discussed with the maintainers before performing such measurements.

Once an experiment is approved by the maintainers, the experimenter is assigned resources and is able to connect to the testbed via OpenVPN. Connectivity to the PEERING testbed is provided by OpenVPN tunnels from the controlling host to each Point of Presence (PoP), see Figure 2.18. Bird is used as a BGP routing daemon to exchange BGP traffic between nodes. The announcement is depicted inside the OpenVPN tunnel via a green arrow. Each PEERING PoP provides connectivity to a different set of ASes, which might be interconnected themselves via additional peering agreements or IXPs.

Several safety nets are built into the PEERING testbed to avoid accidental hijacks and unwanted interruptions of the routing infrastructure. For each client, a filter list containing the allowed prefix ranges has to be created. The measurements cannot announce any range which is not present in this client-side filter list, but also on the server-side measures are in place that limit the abilities of experimenters. Since the PEERING testbed relies on the goodwill of its peering partners to forward announcements without payment, it tries to reduce the risk as much as possible.

AS47065 is currently present at 5 different IXPs around the world (Amsterdam Internet Exchange, Phoenix-IX , SIX Seattle, IX.br São Paulo, IX.br Belo Horizonte). It has 457 active peering sessions at different 9 PoPs [153]. However, not all of the PEERING PoPs are stable and able to be used throughout large scale measurements. The most stable PoPs, which also provide the richest connectivity as well as bandwidth, are Amsterdam and Seattle. Some other PoPs might only have a single

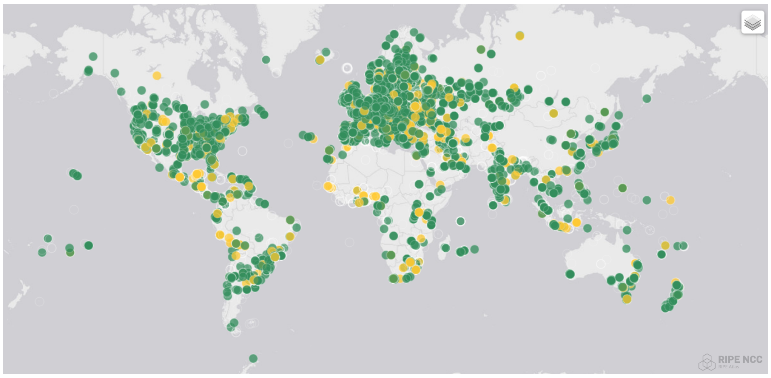


Figure 2.19: RIPE Atlas coverage [155]

upstream peering session and have very limited capacities.

**RIPE Atlas.** RIPE Atlas is a global Internet measurement platform that currently covers 3,775 ASes in IPv4 and 1,789 ASes in IPv6 in 172 countries. In contrast to the previously-introduced frameworks, it works on the data plane. Within these ASes, 12,775 probes are deployed, each connected to a home router, data center switch, or similar. The distribution of probes is shown in Figure 2.19. Probes are either more powerful anchor nodes (804) or small Internet of Things (IoT) devices based on NanoPI NEO Plus2 boards (11.971) that are cheap but fairly reliable [154]. All of them are remotely managed by RIPE Network Coordination Centre (NCC) and allow the issuing a vast variety of data plane packets, such as TCP, User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP).

RIPE Atlas measurements are paid for in credits. On the one hand, each participant receives credits for hosting probes. Anchor nodes create many more credits compared to IoT probes. The more measurements executed via a certain probe, the more credits are generated. On the other hand, participants pay with credits for running their measurements on other probes. During the time of this thesis, we hosted two RIPE Atlas IoT probes in different locations to support the community and become acquainted with the technology.

RIPE Atlas has a very active community. Since the gathered credits from our own probes were not nearly enough to support our large-scale measurements, we asked for support on the mailing list. Within a matter of minutes, several people transferred hundreds of millions of credits into our research account.

Since RIPE Atlas provides probes for self-installation, many probes are present at the edge of the Internet topology, also called eye-ball networks.[1] The existing coverage is not evenly distributed and not every AS is covered by RIPE Atlas (only about 5% of ASes). RIPE NCC initiated a software probe alternative that aims to mitigate these shortcomings.

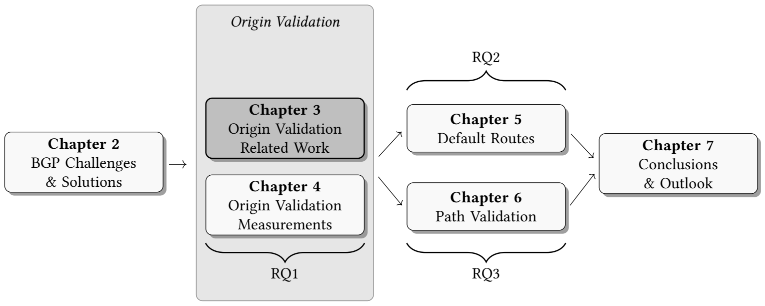
[1]https://atlas.ripe.net/results/maps/network-coverage/

RIPE Atlas provides a web interface to manually configure and run measurements. To automate measurements, an API can also be used. Python libraries to create measurements and fetch measurement data are also available.

**NLnog.** The Netherlands Network Operator Group (NLnog) is a non-profit organization to help improve operational experience of large-scale networks. One particular interesting project in the realm of this dissertation is the NLnog Ring. Similar to RIPE Atlas, the operators of the NLnog Ring support each other by granting access to their own machines within their infrastructure in exchange for access to all other machines participating in the project. By simply contributing one ring node, an operator gains access to quite powerful machines in many other networks that can be used for debugging and trouble-shooting. Since this is a project that is run by rather large-scale networks, most ring nodes are present within the core of the Internet. It is a requirement that the contributed node is also present in the DFZ, which does not, however, always seem to be the case.

CHAPTER 3

# Origin Validation - Related Work



*The previous chapter introduced the basic concepts and focused on the looming BGP problems as well as proposed solutions. We have seen that many proposals for security solutions have been made, but problems remain largely unsolved. This chapter looks further into the details of existing research and introduces a system-of-knowledge that will allow us to classify existing work. In more detail, we identify three major categories, which we further split into more fine-grained compartments. Furthermore, we deepen our understanding as to why many security proposals have never been considered for adoption and what the main obstacles are. We published the study on which this chapter is based in a journal [12].*

## 3.1 SYSTEM-OF-KNOWLEDGE

In order to categorize related work, we introduce a system-of-knowledge that allows us to classify existing research contributions. We identify three major domains: ROA measurements, ROV measurements, and RPKI resiliency. *ROA measurements* deal with research that started very early on with the deployment of RPKI. It contains methods on how to measure coverage of IP prefix space by ROAs. Moreover, the design of ROAs themselves raises some security weaknesses that we address. Research on *ROV measurements* started a bit later. It contains methodologies on how to identify ASes that use RPKI data to filter BGP announcements. Lastly, we discuss research dealing with *RPKI resilience*. In detail, we focus on problems dealing with the centrality of the RPKI infrastructure, inconsistencies in RP software, circular dependencies between BGP and RPKI, and very recent attacks on the RPKI infrastructure. Understanding the existing body of literature surrounding RPKI deployment helps us to define the shortcomings of existing methodologies and improve upon them in our own methodologies in the following chapters. We show the classification of more than 40 scientific publications in Figure 3.1. Moreover, we support claims and findings with a multitude of additional resources.

## 3.2 ROA MEASUREMENTS

The first domain we identified during our study is ROA measurements. Table 3.1 shows all relevant publications within this domain. According to the deployment identified in each scientific contribution, we observe the increasing coverage of IPv4 prefix space by ROAs throughout the past years. To further partition existing re-

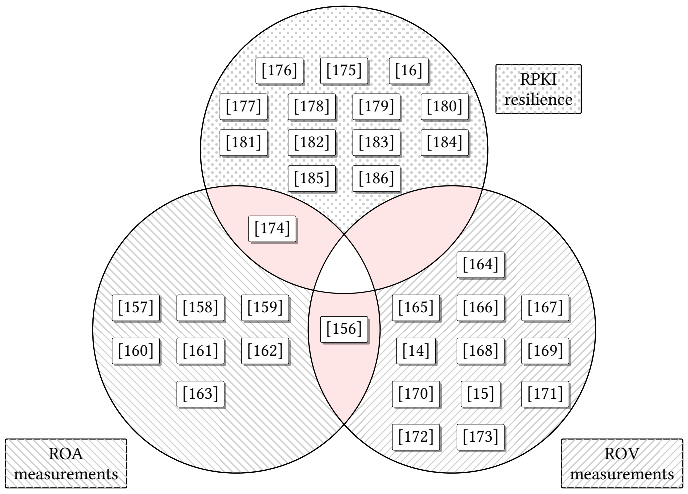


Figure 3.1: Publications per category.

Table 3.1: Comparison of ROA measurement methodologies sorted by year

| Reference | Measurement-Period | Longitudinal | Deployment [%] |
|---|---|---|---|
| Wählisch *et al.* [157], 2012 | April 1 - 30, 2012 | ✘ | 2.00 |
| Iamartino *et al.* [174], 2015 | March 2012 - August 2014 | ✓ | 5.41 |
| Wählisch *et al.* [158], 2015 | Several weeks in 2014/2015 | ✘ | 6.00 |
| Gilad *et al.* [156], 2017 | July 2016 | ✘ | 6.50 |
| Gilad *et al.* [159], 2017 | June 1, 2017 | ✘ | 7.60 |
| Chung *et al.* [160], 2019 | 2011 - 2019 | ✓ | 12.10 |
| Hlavacek *et al.* [161], 2021 | February 26, 2019 | ✘ | 14.16 |
| Li *et al.* [162], 2022 | January 1, 2022 | ✘ | 35.00 |
| Oliver *et al.* [163], 2022 | June 2019 - March 2022 | ✓ | 35.00 |

search, we identified the following fields: (i) how to measure ROA coverage; (ii) the problem of the max-length attribute (loose, minimal and hanging ROAs); and (iii) the use of AS0 ROAs.

### 3.2.1 ROA Coverage

In 2012, Wählisch *et al.* [157] published the first scientific publication dealing with measuring the RPKI. They compared the BGP updates for April 2012 with all available ROAs, which covered roughly 2% of address space at the time. It turned out that 20% of the verifiable routing table was invalid. Since operators were not very familiar with the technology yet, a more detailed analysis revealed that most of these invalids could be attributed to misconfigurations of the ROAs. It was still very early on in the roll-out of the RPKI and the default policy for handling RPKI invalids was not yet to drop them. Hence, not much harm was done during operation. However, the content of the RPKI was perceived as inaccurate and operators were hesitant to deploy the RPKI and rely on its information to filter routes.

From March 2012 to August 2014, Iamartino *et al.* [174] conducted a comprehensive study on RPKI ROA measurements. For a measurement window of two years, the authors used every two-hour snapshots of historical BGP data and hourly snapshots of RPKI ROA data and performed the RPKI validation procedure on each data point in time. While at the beginning of the measurements RPKI ROA coverage was determined to be at 2.05%, it rose to 5.41% in the end. The finding confirmed ROA creation amongst operators. Another finding was that 80% of prefix space that validated to RPKI invalid was still reachable. These RPKI invalid ranges were covered by RPKI valid or not-found prefix ranges. The finding implied that RPKI ROV filtering was not widely deployed and therefore did not have a great impact yet. A central outcome of the study was the recommendation to turn on dropping of RPKI invalids. The results were consistent with [187], [188]. Kloots [188] studied how much traffic would be lost by enabling RPKI ROV when RPKI invalid prefixes would be discarded.

In 2015, a subsequent study was released by Wählisch *et al.* [158], [189] analysing the percentage of the Alexa top 1M domains protected by RPKI. The results showed that 94% of webserver prefixes were uncovered by RPKI while only 6% were covered. From the covered ones, only 0.09% were RPKI invalids. The authors attributed the invalids to misconfigurations. Interestingly, more popular websites were less protected by RPKI compared to less popular ones. Popular websites are commonly hosted within CDNs and those did not yet support RPKI features. The authors analysed and classified 199 ASes as CDNs, but only a single CDN using two ASNs used prefixes covered by the RPKI.

At the beginning of usage of the RPKI, automated monitoring systems were not in place. The previous studies highlighted the need for such monitoring systems to observe ROA creation continuously and provided the methodologies to implement monitors. Much work has been done since then and we are now able to rely on the National Institute of Standards and Technology (NIST) RPKI deployment monitor [190], the MANRS ROA Stats Tool [191], or the Cloudflare RPKI monitoring tool [192] to continuously monitor ROA prefix coverage.

In 2019, Chung *et al.* [160] published a longitudinal analysis of the development of the RPKI. Similar to Iamartino *et al.* [174] in 2015, they looked at longitudinal RPKI and BGP data. They correlated eight years of RPKI data, containing all published ROAs, with BGP data from public collectors throughout the measurement window. The code and results of the analysis are available online [193]. In addition to the public datasets, they also used a private dataset from the Akamai CDN, a large Anti-DDoS provider. The Akamai data was stripped of all private BGP announcements, to avoid a biased view of routing [194], and merged into the overall dataset. Overall, 12.1% of IP address space was covered by ROAs in 2019, which has greatly increased since then. Instead of only comparing ROA coverage, the authors also looked at VRP, the output of the RPKI relying party software after RPKI objects have been cryptographically validated. On February 20, 2019, they found RIPE NCC (16.04%) to have the highest percent of ASes that have VRPs published, followed by LACNIC (9.33%), APNIC (8.14%), AFRINIC (3.30%), and ARIN (1.47%). Overall, RPKI ROA coverage of prefix space was increasing throughout all five RIRs. However, the uptake differs quite significantly when comparing RIRs amongst each other.

### 3.2.2 Loose, Minimal, and Hanging ROAs

In addition to working on ROA coverage, Iamartino *et al.* [174] also looked into the reasons behind the RPKI invalids they found in their study. As suggested by others before, misconfigurations were expected to be the underlying issue since operators were not yet fully acquainted with the workings of the RPKI. The percentage of invalids due to invalid max-length dropped from 61% to 54% between March 2012 and August 2014. Moreover, invalids due to an invalid ASN dropped from 24% to 18%. On the flip side, invalids due to incorrect max-length and incorrect ASN rose from 15% to 27%. It was clear that further development of RPKI tooling was necessary and training of operators needed more time to bear fruit.

Gilad *et al.* [156] introduced in 2017 the concept of *loose* ROAs. They found roughly 30% of prefixes covered by such insecure ROAs, with the organizations issuing such ROAs remaining vulnerable to prefix hijacking. A *loose* ROA describes a ROA that is not strict enough and includes more specific prefixes compared to the ones currently announced in BGP. E.g. AS1 makes the following BGP announcement:

```
1.2.3.0/16, AS PATH: AS1
```

The ROA that is present in the RPKI allows for more specific prefixes to be announced in BGP. This is, because the max-length attribute (/16–24) is too widely defined:

```
ROA: 1.2.3.0/16-24, AS1
```

The RPKI only performs *origin validation.* Therefore, a more specific prefix hijack in combination with a path prepending attack, as introduced in Section 2.5, would render the announcement RPKI valid and propagate throughout the whole inter-domain infrastructure. The attacker AS666 could announce the following more specific prefix in BGP:

```
1.2.3.0/24, AS PATH: AS666 AS1
```

The displayed AS path carries the origin on the very right side. That origin is legitimated by the previously introduced ROA to announce the prefix `1.2.3.0/24`. The attacker AS666 creates an artificial link and claims to have received the route from AS1. Since the announcement is valid and also more specific, it would be chosen by all ASes. The RIPE NCC team published a blog post back in 2011 detailing problems of the max-length attribute [159]. However, the creation of *loose* ROAs makes sense in certain scenarios. In order to be able to react quickly to Distributed Denial of Service (DDoS) attacks, ROAs are already created for DDoS providers, although not yet used operationally on a BGP level. A ROA creation in the event of an attack would simply take too long, since the RPKI is working in an *out-of-band* fashion and would require each RP software to download, validate and push new ROAs to edge routers. Traffic engineering is also another valid reason. Operators need to be able to quickly shift traffic for a multitude of reasons and use smaller prefix ranges to achieve this goal. To reduce the amount of misconfigured ROAs and make operators aware of their mistakes, the authors released ROAlert. The tool is not available anymore. It used WhoIs data to inform operators via email of their *loose* ROAs. Other problems that hinder RPKI adoption are upwards and downwards dependencies. Smaller prefix ranges of a larger subnet are typically given to customers by a larger organization. If the larger prefix would have a ROA, but the smaller customer prefix ranges within that larger range would not, customer prefixes would be rendered RPKI invalid. Therefore, larger organizations must coordinate with their customers and convince them to issue ROAs first. Such a manual process is labour intensive and takes time. As a solution to solve such blocking issues they propose to use wildcard ROAs. These are designed to 'punch holes' in a larger prefix range by allowing any other AS to announce predefined subprefixes. The idea was never considered for adoption.

In 2017, Gilad *et al.* [195] continued to study problems surrounding the max-length attribute in a follow-up paper. The authors suggested the use of minimal ROAs, which are defined as only covering the prefix ranges that are announced in BGP. Minimal ROAs have been defined in RFC 7115 [196]. In order to provide guidelines for operators, the IETF also released RFC 6907 [197], detailing how ROAs should be created. Moreover, in 2011 the RIPE NCC [159] had also published advice on using the max-length option. Despite the aforementioned standards and advice, many ROAs are not minimal in practice. To mitigate this issue, the authors published a Python script that fetches RPKI and BGP data for a certain point in time, iterates over all ROAs and outputs them as minimal, such that they meet the announced BGP prefixes. The tool has been released in [198]. Another recommendation they provide is to change the design of user interfaces of RIR portals. Only experienced users should be given the option to actually change the max-length attribute, while for others it is recommended to only create minimal ROAs. If all ROAs were adjusted to be minimal, the amount of RPKI data in repositories would increase by 23%. To carry this discussion into the IETF, a draft was published [199].

Chung *et al.* [160] use VRPs instead of ROAs as a metric for their study. When ROAs are validated in RP software, VRPs are created. It is important to highlight that the counts of ROAs and VRPs do not necessarily match. A ROA might contain multiple prefixes, while there can only be a single VRP per prefix. Therefore, the count of VRPs is usually higher compared to the amount of ROAs. In their study, an instance of ROA deaggregation is clearly visible, shown in Figure 3.2. The blue line represents the VRP count of Asia-Pacific Network Information Centre (APNIC). The spike was caused by an introduction of an erroneous new management system at APNIC that caused a temporary increase by 13,000 VRP entries, more than doubling the previously present amount of VRPs. Similar results would be expected if the suggestion by Gilad *et al.* [195] to only produce minimal ROAs would be implemented. During their longitudinal study, Chung *et al.* found in the early days of the RPKI roughly 20.76% of RPKI covered BGP announcements to be invalid, confirming earlier research. Many years later, in 2019, only 2.25% - 5.39% of RPKI covered BGP announcements remained invalid. The authors attributed the steady decrease over time to training and monitoring services that were deployed by RIRs. They reported another drop in RPKI invalids in 2018, most likely caused by IXPs that adopted RPKI filtering and forced their customers to fix their RPKI invalids to avoid filtering of their announced BGP prefixes. Chung *et al.* confirmed the results of previous research regarding the reasons for invalid announcements: In 48% - 51.5% BGP announcements were too specific and hence not covered by the issued ROAs. Another reason was the use of a different ASN for the origin of the announcement compared to the one authorized in the RPKI. The authors offered a few likely causes. Firstly, although the same business manages both ASNs, the ASN in the ROA has not been changed. Secondly, a prefix was lent to a customer, but the ROA was not updated. Thirdly, ASes tried to protect against DDoS attacks and forgot to issue the correct

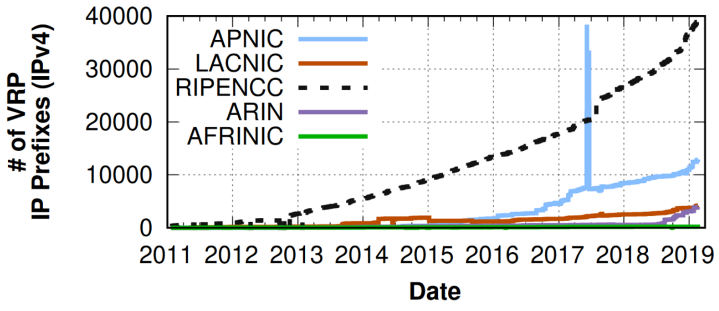


Figure 3.2: Validated ROA Payload throughout the years. [160]

ROAs for the DDoS protection service provider, which they point out is rarely happening. Fourthly, other reasons are responsible for the invalid announcements - this includes possible hijacks. Only 11.2% of prefixes in VRPs use the max-length attribute, which is a decreased use compared to the beginning of the RPKI. The possible causes for RPKI invalid BGP announcements are mostly attributed to misconfigurations, not to actual hijacks. Also, misconfigurations typically last much longer compared to hijacks. Earlier research called for the removal of the max-length attribute, but Chung *et al.* argued that a trade-off needed to be considered. Non-minimal ROAs with a longer max-length still protect against misconfigurations, albeit they do not protect against intentional hijacks. However, the presented methodology has shortcomings and is not capable of making the distinction between misconfigurations and intentional hijacks. They concluded that further research is required.

Hlavacek *et al.* [161] worked in 2021 and in an extended version in 2022 [200] on a more detailed differentiation between misconfigurations of ROAs and actual traffic hijacks. Their research was motivated by the suspicion that invalid ROAs are a root cause to prevent wider ROV adoption on the Internet. During a comparison of RPKI ROAs and BGP announcements they found inconsistencies and attributed the majority of conflicts to misconfigurations instead of actual hijacks. By looking at historic BGP events they found BGP hijacks to be usually short-lived. Similar to Gilad *et al.* in ROAlert [156], they notified the responsible operators via WhoIs contact data of their findings. At first, 760 emails were sent, with a follow-up campaign containing 180 additional emails. In addition to informing the operators of their findings, a questionnaire was attached, asking for possible causes. Most operators replied that the detected errors were simply because of misconfigurations, while others acknowledged the findings but did not see the need for change. The tool was publicly released [201].

Li *et al.* [162] continued to work on the max-length attribute in 2022 by proposing so-called *Hanging* ROAs. Their proposal aims at increasing the performance of compression in RPKI objects. Compared to the previously introduced *minimal* ROA proposal in [195], their proposal applies a bitmap-based encoding scheme which compresses the total size of ROA payloads in RRDP by 26.6%. It also significantly reduces the synchronization cost of RP software. Since objects are smaller, a transfer on a 10 Mbps link would improve the performance (measured in time) of the currently used max-length and the minimal ROA approach by 41.3% and 50.4%, respectively. A disadvantage is that the proposed changes would require adoption from RIRs, RP software developers, and CA-software developers. Considering the small amount of RPKI data that needs to be transferred during an update process, it is questionable whether the additional effort made by a change of the compression algorithm within the RPKI infrastructure is worth the benefit of saving some seconds during transmission. Obviously, it is always good to improve, but a change in infrastructure requires a complex plan to transition from one version to another. Unfortunately, the authors failed to provided additional information on a possible transition period from the currently-used approach to the proposed hanging ROA approach.

### 3.2.3 AS0 ROAs

Oliver *et al.* [163] used the Spamhaus' Don't Route or Peer (DROP) list [202] to investigate the usage patterns of 712 BGP prefixes. The DROP list is regularly used to identify malicious address space and to drop traffic from the blacklisted ranges. Firstly, they found that 32% of prefixes were added to the IRR only a month before they appeared on the blacklist. Once again, it was confirmed that the IRR is unreliable and easy to tamper with. Secondly, they found that attackers usually prefer unallocated or unannounced prefix space and refrain from targeting RPKI-protected address space. To further increase the protection of unallocated or unannounced address space, the authors suggested using AS0 ROAs. This type of ROAs has been specifically designed by the IETF for such a purpose and carries special meaning. A RIR is able to flag unallocated address space and organizations are able to flag assigned but currently unused address space in the RPKI. If an AS receives a route that is covered by an AS0 ROA, it should not install or propagate the route any further, otherwise the RPKI would render announcements as RPKI unknown, allowing the operator no judgement on the legitimacy of the announcement. 70.1% of the unrouted prefix space that is covered by non-AS0 ROAs is owned by only three organizations. The creation of AS0 ROAs would therefore be a fairly small effort by three organizations to improve the overall inter-domain routing security. The authors suggested that RIRs and operators reconsider the use of AS0 ROAs within their domains.

We conclude that in the beginnings of the RPKI, research had focused on proposing measurement methodologies to measure IP prefix coverage of the RPKI. These methodologies are currently implemented in automated monitoring tools such that operators are always able to tell the current state of RPKI coverage. Additionally, *loose* ROAs have been identified and new variants, such as *minimal* and *hanging* ROAs proposed. Finally, *AS0* ROAs allow the protection of unallocated or unannounced prefix space and should be deployed by RIRs and ASes.

Table 3.2: Comparison of ROV measurements sorted by year

| | | Plane | | Experiment Type | | | |
|---|---|---|---|---|---|---|---|
| Reference | Measurement Period | Control | Data | Controlled | Uncontrolled | Approach | Longitudinal |
| Gilad *et al.* [156], 2017 | July 2016 | ✓ | ✘ | ✘ | ✓ | BGP dump analysis | ✘ |
| Reuter *et al.* [165], 2018 | February 20–27, 2017<br>May 11–17, 2017<br>August 1–7, 2017 | ✓ | ✘ | ✓ | ✘ | BGP dump analysis with route injection | ✘ |
| Hlavacek *et al.* [166], 2018 | February 2017–June 2017 | ✓ | ✓ | ✓ | ✘ | Traceroute & TCP SYN | ✘ |
| Cartwright-Cox [167], 2019 | N/A | ✘ | ✓ | ✓ | ✘ | ICMP scans | ✘ |
| RPKI WebTest [170], 2019 | on-demand, discontinued | ✘ | ✓ | ✓ | ✘ | HTTP | ✘ |
| Rodday *et al.* [15], 2019 | August 22–29, 2019 | ✓ | ✘ | ✓ | ✘ | Control plane extensions | ✘ |
| Cloudflare [173], 2020 | on demand | ✘ | ✓ | ✓ | ✘ | HTTP | ✘ |
| Testart *et al.* [171], 2020 | April 1, 2017–January, 22 2020 | ✓ | ✘ | ✘ | ✓ | Statistical approach | ✓ |
| Huston *et al.* [168], 2020 | June 1–20, 2020 | ✘ | ✓ | ✘ | ✓ | HTTP | ✘ |
| Rodday *et al.* [14], 2021 | July 2–19, 2021 | ✘ | ✓ | ✓ | ✘ | HTTP & Traceroute | ✘ |
| Chen *et al.* [169], 2022 | May 1-31, 2021 | ✘ | ✓ | ✘ | ✓ | Traceroute | ✘ |
| Hlavacek *et al.* [172], 2023 | June 8 and 10, 2022 | ✓ | ✓ | ✓ | ✘ | BGP dump + Traceroute | ✘ |
| RoVista [203], 2023 | December 24, 2021–September 12, 2023 | ✘ | ✓ | ✘ | ✓ | IP-ID side-channel technique | ✓ |

## 3.3 ROV MEASUREMENTS

The previous section focused on methodologies to determine the amount of prefix space that is currently protected by ROAs. These methodologies allow us to draw conclusions on how useful the RPKI could potentially be. However, protecting prefix space alone will not change the level of routing security. ASes also have to deploy RPKI filtering in order to make an impact on the routing infrastructure. They need to filter or depreference RPKI invalid announcements. Such process is called BGP prefix origin validation, or Route Origin Validation (ROV). The following section focuses on methodologies to measure RPKI filtering in the wild. They aim at answering which ASes perform RPKI filtering.

Two major areas in RPKI ROV measurement methodologies exist: control plane and data plane measurements. Measurements can be active or passive, controlled or uncontrolled. *Active* measurements describe the injection of artificial traffic into a system in order to observe a result, while *passive* measurements rely on existing information without the need to interfere with ongoing processes. *Controlled* measurements describe a measurement setup in which the measurement variables are controlled by the experimenter, while measurement variables lie outside of such control for *uncontrolled* measurements. Table 3.2 summarizes prior work.

### 3.3.1 Control Plane Measurements

Gilad *et al.* [156] published a ROV measurement methodology in 2017 using passive control plane measurements. Within RouteViews BGP collector dumps, they filter for an AS that originates an RPKI invalid and a non-invalid (*i.e.,* not found or valid) BGP announcement. The obtained data is further filtered for routes that have exactly one transit AS inbetween the origin and the BGP collector. The transit AS is flagged as ROV-enforcing if two conditions are met: Firstly, the AS is dropping the RPKI invalid announcements, but continues to forward the RPKI non-invalid ones; Secondly, the action is performed by the AS for three different destination ASes.

Their results show that three out of the top 100 ASes deploy ROV-filtering. In addition to the measurement study, they performed a survey amongst network operators asking how relevant of a topic ROV is within their AS. 84.09% respond that they are not using ROV at all, while 10.23% are de-preferencing RPKI invalid announcements. Only 5.68% are using the RPKI to drop invalid routes. They also present the concept of collateral benefit and collateral damage. *Collateral benefit* refers to an AS sitting behind another AS that performs ROV-filtering. The AS in question does not perform the ROV-filtering itself, but it benefits from the upstream or peer that is only forwarding RPKI-valid and RPKI-unkown prefix ranges. Therefore, it does not receive RPKI-invalids via the peering link in question in the first place. Precisely this action can also have negative consequences for the downstream AS, which is called *Collateral damage*. If the upstream filters and removes RPKI-invalid BGP announcements, a downstream peer might not receive certain routes and is therefore disconnected from parts of the inter-domain routing infrastructure. In another scenario, two routes, a covering prefix and a hijacked more specific prefix are forwarded to a downstream peer. The covering prefix is originated by the legitimate origin and either RPKI-valid or RPKI-unknown. The hijacked prefix is RPKI-invalid. The upstream itself does not perform ROV and therefore forwards both announcements to the downstream peer. Assuming that the downstream peer filters the RPKI-invalid announcement, it would still send the traffic for the filtered prefix towards the upstream since the covering prefix exhibits the same path. The upstream, however, does not perform RPKI-filtering and therefore forwards the traffic for the filtered subnet to the RPKI-invalid origin. In this particular case, RPKI filtering at the downstream is without any effect.

Reuter *et al.* [165] revisited earlier work by Gilad *et al.* [156] and showed that the obtained results were heavily based on the chosen set of BGP collectors. The previously-introduced concept of collateral benefit caused measurement errors that resulted in an incorrect attribution of ROV filtering. Instead of relying on *uncontrolled* measurements, they argued for *controlled* measurements. This change was intended to reduce the amount of independent variables present in the measurement setup and increase control over the experiments. Their study identified three ASes that deployed ROV filtering. The positive identification of these three ASes was confirmed by their operators. In order to allow for continuous monitoring of ROV deployment, the measurements were installed as an ongoing study [204]. In March 2021, they identified 118 ASes as deploying ROV. Since then the live monitoring system has no longer been available.

Testart *et al.* [171] introduced a statistical method to infer ROV-filtering from control plane data. They used three datasets, one longitudinal BGP dataset from 04/2017–01/2020 every first day of a month; one BGP dataset that rebuilds RIBs in five minute intervals for 09/2019; and accordingly a RPKI dataset containing validated ROAs for each day in the same month. Their methodology first identifies so-called full-feeders within the BGP collector data. A full-feeder is defined as an AS that reports more than 75% of the globally-visible routing table to a collector. Within that dataset, they looked for ASes that report significantly fewer (only up to 20%) RPKI-invalid prefixes compared to the set of full-feeders. They identify 21 ASes as ROV-filtering. By using publicly-available announcements, they confirmed correctness for five of their inferences. A possible bias of the presented methodology is *collateral benefit*. Consider multiple ASes that are within the same customer cone of a parent that performs RPKI-filtering. All of the downstream ASes will likely report significantly fewer RPKI-invalid routes to the BGP collectors. Overall, the authors clearly identified increasing RPKI usage on the Internet. Their methodology worked, since the statistical difference between ASes that adopt and do not adopt RPKI validation is significant. Only very few ASes deploy RPKI-filtering while the vast majority do not. With the forthcoming RPKI deployment, statistical differences will likely become smaller. Therefore, it will be hard to perform such a methodology at a later stage of RPKI deployment.

Gray *et al.* [205] proposed BeCAUSe in 2020. The algorithmic framework is using Bayesian computation to infer network properties for ASes. One of the use-cases for validating their approach is to identify ROV-enforcing ASes.

### 3.3.2 Data Plane Measurements

Cartwright-Cox [167] proposed a methodology to measure RPKI adoption on the data plane in 2018. Firstly, excessive ICMP scans coming from an RPKI-valid source address target the whole IPv4 address space and identify responsive hosts. Secondly, an additional ICMP probe is sent towards the previously responsive hosts, this time from an RPKI-invalid address range. The number of replies received is recorded by a control server. If a reply is observed from an RPKI-valid range, but not from an RPKI-invalid range, ROV-filtering is assumed. The methodology has a significant drawback. It does not allow the pinpointing of ROV-enforcing ASes. Filtering could happen in any transit AS along the way. It is therefore not possible to attribute filtering to a single AS. However, precisely the attribution to a single AS is attempted with this methodology. Updates to this study were presented at NLNOG Day, in September 2019, and at RIPE 80, in May 2020 [206].

Huston *et al.* [168] performed a similar study on the data plane in order to determine the level of protection of end user devices by the RPKI. Many end user hosts query via HTTP a server that has been assigned an IP address from a BGP experiment prefix. The BGP experiment prefix remains for 36 hours in an RPKI valid state and is swapped to an RPKI invalid state for the next 12 hours, before the cycle repeats. Any difference in reachability is attributed to ROV filtering. Opposed to Cartwright-Cox [167], Huston *et al.* [168] aim at identifying the share of protected end users and do not attempt to attribute ROV-filtering to the AS hosting the end user probe. They report end user protection to be as high as ∼17%. Many transit providers are suspected to perform RPKI filtering instead of stub networks.

The RPKI WebTest [170] was a website to test whether the ISP of an end user deployed ROV filtering. It was operated by RIPE NCC and created to increase awareness for RPKI filtering. A user could simply access the website in their browser. Two Hypertext Transfer Protocol (HTTP) queries would be created. The first query towards an RPKI valid address range, and a second query to an RPKI invalid address range. If both queries were successful, the ISP of the user was considered not to deploy ROV. If the invalid address range could not be reached, but the valid one succeeded, RPKI filtering was inferred. The Cloudflare [173] RPKI ROV project uses the same methodology. Moreover, it is possible to issue pull requests on Github to update the list of RPKI ROV enforcing ASes. Since the process requires manual intervention from AS operators, only few are expected to update their RPKI ROV status. Similar to Cartwright-Cox [167], the RPKI WebTest and the Cloudflare RPKI ROV project do not consider transit ASes. Any AS inbetween could have performed the filtering. To inform the user that their own ISP is not filtering is a correct conclusion, since the invalid range could be reached. However, informing the user that the local ISP is filtering might be an incorrect conclusion, since any transit towards the RPKI invalid range could perform the filtering. In such a case, the attribution of RPKI filtering could be incorrect.

Hlavacek *et al.* [166] summarized existing ROV measurement methodologies in 2018. To confirm detection rates, they repeated the study by Reuter *et al.* [165] using controlled *control plane* measurements. As an improvement in order to achieve a higher level of accuracy, the authors argued for data plane measurements instead of relying on control plane measurements. They also altered the methodology in such a way that not only one prefix pair (valid/invalid) is announced at a time from one AS, but they instead additionally announce the inverse of that combination from another AS, effectively creating always one valid and one invalid announcement per prefix from competing ASes. During the data plane measurements, the authors used traceroutes issued from RIPE Atlas probes. Moreover, they inferred a share of protected end hosts, by relying on TCP-SYN packets sent to the top 1,25M Alexa domains. Four ASes were identified as ROV-enforcing via control plane measurements, while the data plane measurements yielded 12 ASes. By looking at the TCP replies received from the top 1.25M Alexa hosts, they flagged 201 TCP end points as protected. Shulman *et al.* [207] used the same methodology in June 2022 and reported an extraordinarily high adoption rate of 37.8% of ASes that enforce ROV.

van Hove *et al.* [208] ran a short measurement campaign in 2022. They announced two prefix ranges, a RPKI valid /23 range from Sydney and another RPKI invalid /24 range from Amsterdam. The assumption was that if RPKI filtering was fully deployed, there should not be any traffic arriving at the RPKI invalid prefix range, although it is more specific. To test the hypothesis and measure incoming traffic they set up two RPKI publication points, one within the RPKI valid, the other within the RPKI invalid prefix range. RP software would attempt to connect to the IPs of the two publication points and generate traffic. Any other vantage points generating traffic, such as RIPE Atlas nodes, would have also been a viable option. 75% of traffic was observed at the RPKI valid prefix range, compared to 25% of traffic at the RPKI invalid prefix range. Since the location and the degree of connectivity of the announcing ASes are very important for such measurements, it is hard to generalize these findings.

In 2022, Chen *et al.* [169] published the most recent methodology to identify ROV-enforcing ASes. The connected-assumption is a problem that Reuter *et al.* [165] identified to make rigorous inferences of ROV-enforcing ASes. The announcing AS must be directly connected to the AS under test. It significantly reduces the amount of ASes that judgements can be made for, since a direct peering relationship is required. Chen *et al.* aimed to remove the connected-assumption to include a higher number of ASes into their measurements. They also resorted to uncontrolled measurements, instead of controlled ones, to be able to work with more data. The methodology starts with selecting roughly 6,000 publicly available RPKI-invalid BGP prefix-origin pairs from several different ASes. The data is accessed via BGPStream [149] and made available by RIPE RIS [146] and Routeviews [140]. The methodology has similarities to Gilad *et al.* [156]. Prefixes covered by other legitimate announcements and multi-homed prefixes are removed from the dataset. Next, they try to find RPKI-valid prefixes that are originated by the same origin ASes. Measurements were run on several days. On average they find 350–500 prefix-origin pairs where the previously introduced conditions are met. It is important to highlight, that the only things these prefix-pairs are required to have in common are that one prefix is RPKI-invalid and the other is RPKI-valid, and that both are announced by the same origin. For each prefix pair ZMap [209] is used to find active hosts. Next, traceroutes are issued from 200 randomly selected RIPE Atlas [210] and perfSONAR [211], [212] probes towards each active host in both the RPKI-valid and the RPKI-invalid prefix range. The obtained traceroutes are then translated into AS paths. The resulting paths either show equality or inequality. If they divert from each other, ROV is attributed to the ASes en route and the Stein Variational Gradient Descent (SVGD) model is used to obtain a probability score per AS. In earlier research, Gray *et al.* [205] used a similar model for the identification of Route Flap Damping (RFD). From 11,074 ASes that forward RPKI-valid announcements 28% deploy ROV (n=3107), 43% do not deploy ROV (n=4716), 3% partially deploy ROV (n=357), and 26% are unknown (n=2894). A ground-truth dataset from is-bgp-safe-yet [173] is used for validation. The authors report 100% precision and 100% recall.

Using uncontrolled measurements is quite unreliable. It has been shown by Reuter *et al.* [165] that traffic engineering, but also other factors, can heavily skew the experiments. We have to consider results from such methodologies carefully, since it is unclear whether they would indeed attribute ROV filtering correctly. Some IXPs also deploy ROV, according to their public announcements. Since they are transparent on the as path, incorrect attribution becomes even more realistic.

Hlavacek *et al.* [172] investigate ROV deployment at routeservers at IXPs via controlled control plane and data plane experiments on June 8 and 10, 2022. Similar to [14], [166], the control plane data is relying on Routeviews [140] BGP dumps, while the data plane data is acquired by running traceroutes from RIPE Atlas probes. They introduce divergence points as a new term for ASes that are found as ROV-filtering. In their work, they split their results into seven categories, each which has a different confidence level. More than 27% of ASes are reported to filter RPKI invalid prefixes. Interestingly, they find IXPs to not block hijacks, although these IXPs are filtering based on RPKI ROV. This is, because many peers rely on direct peering sessions at IXPs that would circumvent the filtering features enabled by the IXP. They find out of 15 tier-1 providers 11 to be correct inferences and 4 non-verifiable. False negatives were not reported. Compared with the previously introduced Cloudflare measurements [173], they find 75% overlap. For the APNIC measurements [168] they report 79% overlap.

Li *et al.* [164] published their latest advancements in the field of ROV identification in 2023. During a period of December 24, 2022, to September 12, 2023, they ran measurements every four hours using their new approach called RoVista. It is based on the IP-ID side-channel techniques which provides insights into the connectivity between two remote hosts. The IP-ID field is a specific field in the IPv4 header. The method does not require control over the remote hosts that are within the control of the ASes under test. Hence, they were able to measure 28k ASes. Results are grouped into no filtering, inbound filtering, and outbound filtering. All measurements are executed in an uncontrolled fashion as the authors rely on RPKI invalid BGP announcements obtained from BGP collector dumps. Firstly, ZMap [213] is used to scan the RPKI invalid prefix ranges and find hosts that reply correctly to TCP SYN packets. They report 0.7% of the global routing table to be RPKI invalid. Moreover, they further sanitize the data and find 31 test nodes that can be used for their experiments placed within RPKI invalid prefix space. Secondly, the authors identify virtual vantage points via ZMap by isolating nodes that send RST packets as a reply to a TCP SYN/ACK packet. In 28,314 ASes they find 1,396,407 virtual vantage points. Since RST packets allow the tracking of an increase in the IP-ID counter of the sending host, the method relies on sending traffic between pairs of test nodes and virtual vantage points. A packet delivery from a test node towards a virtual vantage point is triggered by spoofed data plane packets. If the packet successfully arrives at its destination, the IP-ID counter is increased and the method infers a successful connection. By using the overall amount of test nodes and virtual vantage points, the authors calculate an ROV protection score per AS. Similar to other methods, RoVista might report full protection while the AS itself is not filtering if the AS under test receives collateral benefit. 63.8% of all ASes are reported as having received some kind of RPKI protection, while 12.3% are fully protected by RPKI ROV.

In summary, we have discussed many approaches to identify ROV filtering using control plane as well as data plane measurements. Precise wording is key as many proposals mix the measured protection of the end host probe with the attribution of ROV filtering for the AS hosting such probe. We note that many proposals fail to correctly attribute ROV to filtering ASes.

## 3.4 RPKI RESILIENCE

We identify the following primary fields as being relevant to the RPKI resilience domain: (i) centrality of the infrastructure; (ii) Relying Party (RP) inconsistencies; (iii) circular dependencies and usability; and (iv) attacks.

Table 3.3: Comparison of RPKI resilience research sorted by year

| Reference | Topic |
|---|---|
| Cooper *et al.* [177], 2013 | Whacking of ROAs |
| Heilman *et al.* [175], 2014 | Consent via .dead object |
| Iamartino *et al.* [174], 2015 | LACNIC/APNIC outage for 9 months |
| Liu et al [181], 2015 | RPKI risks categorization |
| Hari *et al.* [176], 2016 | Blockchain proposal |
| Yan *et al.* [178], 2018 | CA-software suggestions |
| Kristoff *et al.* [180], 2020 | 90% of RPs not falling back to rsync |
| Shrishak *et al.* [182], 2021 | Threshold-based delegation |
| Friedemann *et al.* [16], 2022 | Comparison of RP software |
| Hlavacek *et al.* [183], 2022 | Stalling of RP software |
| van Hove *et al.* [179], 2022 | Vulnerabilities in RP software |
| Hlavacek *et al.* [185], 2022 | Attacking DNS to harm RPKI |
| Fontugne *et al.* [184], 2023 | Delays in the RPKI ecosystem |
| Hlavacek *et al.* [186], 2023 | RP threshold analysis |

In prior work, Liu *et al.* [181] worked on aggregating existing research within the RPKI resilience domain. They proposed the separation into technical, economical, and political risks. A condensed version of their paper is available as an IETF draft [214]. Previous work is listed in Table 3.3.

### 3.4.1 Centrality of the Infrastructure

The inter-domain routing infrastructure is a distributed system, which has proven to be a good design choice to reduce complexity and split power between different entities. The RPKI itself relies on a hierarchical architecture. Even in 2008, before the RPKI was productively deployed, the Internet Governance Project pointed out that the RIRs would likely be targeted by those who try to regulate the Internet [215]. The RIRs hold the TAs. A compromised root would allow for compromising the trust and security within a whole geographical region. It would jeopardize the whole security architecture and bring back attacks that the RPKI is here to protect against. Subsequently, in 2011, the RIPE NCC wanted to clarify how it would have to react in the event of a foreign court order [216].

The research community started to be aware of the problem of the centrality of the RPKI infrastructure in 2013. One assumption that is heavily relied on is that RPKI authorities are trusted entities. They never misbehave, they never go rogue. Precisely this assumption is questioned Cooper *et al.* [177]. They focused on issues which are created by the trust and power that is invested into the RPKI authorities by design. What would happen if a root CA started to revoke child certificates without a reason? The damage that could be done to the whole ecosystem would be immense as the RPKI is built on chains of delegation. A revocation at the top would result in invalidating the whole chain. To motivate their research, the authors argue that "there is ample evidence of authorities [...] being hacked [217]–[219], misconfigured [220], or compelled by government agencies to delete information (e.g. DNS takedowns [221]) or attest to bogus information" [222].

**Whacking of ROAs.** A term introduced by Cooper *et al.* is the *whacking* of ROAs. Resource Certificates (RCs) are capable of holding many prefixes and could therefore cover larger address blocks. Whacking describes an attack in which the CA itself overwrites an existing RC with a fraudulent intent. The majority of address space remains within the RC, but the address space relating to the (great-)grandchildren ROA that is supposed to be whacked is removed from the RC. Since a cryptographic validation of the ROA fails from now on, RP software would not include the prefix space in the Validated ROA Payloads (VRPs). As a result, the BGP announcement carrying the prefix space covered by the ROA becomes RPKI-unknown. If another organization issued a covering ROA for the targeted prefix space, the BGP announcement might also become RPKI-invalid. The attack itself would be hard to detect, since it only targets a single, specific ROA. It does not cause collateral damage that would be easy to spot. Revoking the whole RC, containing many more prefixes, would result in an attack that is much less stealthy.

In 2014, Heilman *et al.* [175] continued to work on the problem of the overarching power of RPKI authorities [177], [223]–[225]. In order to create social and legal pressure, they propose to deploy a transparency mechanism that would allow the community to notice if illegitimate changes were made. A looming problem, however, is the differentiation between legitimate and illegitimate changes. The RPKI is not a static system, but changes on a regular basis. Certificates are issued and revoked, delegations take place, and ROA are created or removed. To understand which underlying reason caused the change, be it dispute, censorship, or business arrangements, is a hard problem. They also propose that security audits might increase the level of trust. This recommendation has been implemented by RIRs like the RIPE NCC [226].

In order to ask for consent before a ROA is invalidated by changing a RC, Heilman *et al.* present the *.dead* object. The RPKI is based on a tree structure. Starting at the leaf which is represented by a ROA, each entity on the path towards the root is requested to sign a *.dead* object that acknowledges the change of the RC to remove a particular resource. The signing procedure is a technical implementation to describe consent in removing the resource. Once implemented, such a process would add transparency, since resources that are removed without previously running the acknowledgement process would be easy to identify. The downside of the approach is a heavily complicated structure and revocation process. Moreover, one has to consider that this is merely a transparency, not a security mechanism. Missing *.dead* objects would not impact an RPKI root with malformed intent. It holds the absolute power and could simply overwrite the existing RC, no matter what.

**Cross-country certification.** Cooper *et al.* [177] also identified the risk of cross-country certification. Address space is often delegated to customers in other countries and, hence, other jurisdictions. In case of legal disputes between the two entities, a court from one country might decide on address space that is delegated to a company in another country. Such a scenario would make it very costly, maybe even impossible, for the company in another jurisdiction to enforce their interests.

**Blockchain.** In 2016, Hari *et al.* [176] proposed a blockchain-based approach to replace the currently-deployed centralized RPKI structure for signing and delegating resources with a distributed model. The underlying algorithm is capable of creating a 1 MB block every 10 minutes, which translates to 3–7 transactions that can be processed. The model works for the current RPKI, considering the average volume of ROA changes, but fails during peak times where many ROAs are changed by a single or multiple ASes. In such a case, a delay would be introduced before ROAs would be properly included into the chain. Such a delay is undesirable, since ROA propagation and validation within RP software would fail as long as the queue has not been processed. To make the problem worse, not only ROAs, but also BGP updates would have to be included in the chain if BGPSec functionality was to be supported with blockchain technology. BGP had on average 9,000 BGP updates per second in 2016. Using the proposed algorithm, the blockchain approach is much to inefficient to support that many transactions per second. Another problem is the growth of the chain that comes with continuous use. The more data is incorporated, the larger the chain becomes.

Nonetheless, blockchain is a promising technology on which other researchers previously focused their attention. Back in 2009, Haeberlen *et al.* [227] evaluated the use of blockchain technology to implement a secure log of BGP transactions. The idea was to consult the log whenever analysis of BGP problems was necessary. Paillisse *et al.* [228] proposed in 2018 to implement Proof-of-Stake as a consensus algorithm. With such a model, larger ASes that hold more address space would become more powerful compared to smaller ones. Mastilak *et al.* [229] surveyed existing blockchain variants and showed how they can be applied to technologies present in the Internet.

Although it might sound like a good idea to use blockchain in the first place, there is a conceptual gap between BGP as an information-hiding protocol and blockchain as a transparent chain of transactions. Many operators might be hesitant to adopt a technology that publicly records potentially business sensitive information.

**Threshold-based approach.** Shrishak *et al.* [182], [230] continued to work on the overarching power of RPKI authorities in 2020 and 2021. In the current architecture, each RIR is technically able to issue and sign prefix delegations for the whole IPv4 and IPv6 address space. This is due to the fact that there are five trust anchors, one with each RIR. To limit the power of RPKI authorities, Shrishak *et al.* proposed to use a threshold-based approach for resource delegation, with the Dalskov *et al.* [76] protocol as a threshold signature model. Only joint signatures of resource delegations would be possible. One RIR could no longer delegate prefix space assigned to another RIR. A major argument would be legal requirements. If a court ordered a RIR within its authority to sign or withdraw certain cryptographically secured objects, the RIR could no longer comply by itself. It would therefore not be possible for a single court to impact the security of the inter-domain routing infrastructure that easily. Also, multiple compromised RIRs are a very unlikely scenario, while compromising a single RIR might be feasible for *e.g.,* a state-sponsored actor. When the threshold-based model would be deployed, three out of five RIRs are required to agree on a certain action in order to obtain a majority vote. It becomes highly unlikely that such a scenario is realistic for malicious behavior.

The RPKI supports two models for certificate management: hosted and delegated. With the hosted model, the RIR provides a web interface in which the resource holder is able to issue ROAs and delegate resources. All private keys remain with the RIR. In the delegated model, the resource holder uses a CA software, *i.e.,* Krill [82], in order to control prefix delegate and ROA issuance for multiple RIRs. See Section 2.4.3 for more details. The threshold-based approach only works with the hosted model since it requires all keys to be at the same location.

The proposed approach would indeed limit the power of RPKI authorities. However, it would also introduce a significant amount of complexity during the delegation and signing procedures. The threshold-based approach would require 20,000 signatures per day. It would be able to cope with average days, but would fail to deal with increased numbers of signatures during peak hours. These peak hours would introduce delays, which negatively impact the time until the BGP protocol converges, based on new RPKI data.

### 3.4.2 Relying Party Inconsistencies

Other important things to consider are inconsistencies that could potentially be present during a validation interval of RP software. Cooper *et al.* [177] highlighted that ROAs could be missing during the validation process of RP software due to a corrupted file system, delayed renewal, or the unavailability of a RPKI repository. Consider a valid ROA that covers a BGP announcement. If that ROA is removed from the RPKI on purpose, the BGP announcement will become either RPKI unknown or RPKI invalid. It becomes RPKI unknown if there is no other covering ROA present and it becomes RPKI invalid if there is a covering ROA from a different organization. If RP software is not able to include the address space covered by the ROA into the VRP, we observe the same result. The authors argue that it is very important that RP software has a full set of ROAs to take into consideration when performing the validation process. Different sets of ROAs for different RP software instances would lead to inconsistencies between ASes and therefore to different routing decisions. An attack in which the attacker is trying to create a different view for different participants of the Internet is called a *mirror-world* attack.

Heilman *et al.* [175] pointed out that manifests have an expiration date. A manifest is a document in which the hashes of ROAs are tracked such that RP software can easily spot which items within the RPKI repository have changed. RP software would only fetch the changed objects to save resources. If a manifest expires, all content will be excluded from the validation process. Moreover, most RPKI repositories are managed by RIRs. A RIR can therefore edit the manifest, simply excluding a certain ROA. With such an attack it would also be possible to target only a single AS that fetches data, even a single RP software instance, resulting in a *mirror-world* attack.

To avoid such problems, Heilman *et al.* proposed the use of hash-chained manifests. An updated manifest would always carry the hash of the previous manifest, such that RP software can easily track changes. They also proposed that manifests should not expire, but instead become stale. Once stale, a missing-information alarm should be raised, indicating the staleness. In order to reduce the cryptographic overhead, they argued that only manifests should be signed, but not ROAs. The use of a collision-resistant hash in combination with signing the manifests provides a high-enough level of security. A side-effect would be the obsolescence of Certificate Revocation Lists (CRLs), while ROAs and RCs would no longer expire. The proposal remains entirely academic. There has not been an IETF draft and therefore no discussion amongst operators whether such changes would indeed be for the better. Heilman published his dissertation [231] on the topic, which contains much additional information. Also in 2014, Kent *et al.* [232] submitted an IETF draft called "Suspenders: A Fail-safe Mechanism for the RPKI". The draft proposes a solution to the issue of changed RPKI objects and monitoring.

In 2015, Iamartino *et al.* [174] found that LACNIC and APNIC repositories were running expired X.509 TA certificates on production systems for a period of roughly nine months. During the outage, all BGP announcements covered by the affected ROAs were rendered RPKI unknown. All these incidents highlight the need for monitoring systems. Back then, such systems were not in place. In the current day, many more monitoring systems are deployed and outages of the RPKI infrastructure are detected more rapidly [233]–[236].

Kristoff *et al.* [180] investigated RP software behavior in 2019. The measurement setup consists of one child, two grandchildren CAs, and three publication points under a single RIR CA. During the measurement phase they collect data of RP software, such as RP software name, IP address, originating ASN, reverse DNS records and timestamps. They report 20% less traffic on one of their publication points compared to the parent. Hence, different RP software instances work with a different set of data. During the one year of measurements, the number of RP software instances increased from 25–100 to 75–250. In the same dataset, the number of Facebook's RP software instances increased from 0 to almost 70 in 2020. According to their observations, overall RPKI deployment continued to increase. Most operators rely on one or two distinct RP software implementations to perform the RPKI validation procedures. They also deploy the RP software within their own AS instead of relying on third party CDNs, most probably to reduce the risk of an overall outage. 20% of RPs software only sync 20 times or less per day. Hence, changes in the RPKI will take quite a while to propagate until all ASes receive up-to-date information and BGP converges based on filtering RPKI-invalids. Another point is that the majority of RP software that accounts for 90% of traffic also do not fall back to rsync, which is recommended by the IETF standards. In response to the arguments present in the publication by Kristoff *et al.* , NLnet Labs' Martin Hoffmann argued that the fall back to rysnc would most probably take down the rsync channel, since it is not build to cope with such a massive amount of requests [237]. Therefore, Routinator does not fall back to rsync. He also raised the point that cached data will be used until the expiration date has been reached, and therefore the maintainer of the repository has sufficient time to fix the issues.

In 2023, Fontugne *et al.* [184] investigated delays within the RPKI infrastructure and performed two experiments. Firstly, a prefix pair /24 for each RIR is announced from an AS surrounded by ROV filtering ASes. The control prefix remains static and the test prefix is swapped regularly to render the BGP announcement RPKI valid or invalid. The experiment is conducted for eleven months. Secondly, three /24 test prefixes from RIPE NCC are announced by three networks. All ROA states are changed daily. They record the time for the user query in the RIR portal, ROA signing, ROA publication, and RP validation till the information contained in the ROA is productively deployed within an AS. They find that ROA creation varies significantly across RIRs, from a few minutes to over an hour till ROAs reach publication points. A possible cause is that the underlying processing mechanisms differ, *e.g.,* batch processing or similar. ROA deletion takes longer compared to ROA creation to reflect in BGP. As expected, most delays for ROA creation come from RP software implementations that are pulling ROAs from publication points in different time intervals.

## 3.4.3 Circular Dependencies and Usability

The RPKI was invented to secure inter-domain routing, yet it is using inter-domain routing technology to transfer information. It relies on TCP/IP to issue RCs, publish ROAs, transfer data from the RPKI repositories to the RP software, and forward the resulting VRP to BGP routers. Cooper *et al.* [177] pointed out that this is a circular dependency between BGP and RPKI that might lead to problems. BGP propagates reachability information. The RPKI is a security add-on to filter such reachability information. If routes are filtered based on RPKI information, some RPKI repositories might become unreachable. Such behavior is suspected to lead to a downwards spiral in which overall connectivity is impacted.

Yan *et al.* [178] developed use-cases that CA software must be have the capability to support. As discussed in Section 2.4.3, a RIR is theoretically able to delegate any resources within the whole IPv4 and IPv6 range, although the address space is assigned to another RIR. Based on their use-case study, they proposed the integration of alerting mechanisms into CA software. An alert should be raised if a CA is tasked to sign resources under the management of another CA. However, during a key-rollover operation precisely such an alert would be triggered. Hence, they suggested the exclusion of this case from the altering mechanism. The addition of such an alerting mechanism would not prevent an attacker from delegating address space under control of another CA. A simple warning can be ignored. A resourceful attacker could even implement changes into the open-source software to alter the behavior. It is, however, useful against simple operational mistakes.

### 3.4.4 Attacks / Threat Models

In 2022, Hlavacek *et al.* [183] presented an attack on the RPKI that aims at stalling RPKI RP software with the goal to prevent certain ROAs from being processed. It has two components. Firstly, during the update procedure of RP software the attacker tries to cause packet loss using a specific time pattern that is in sync with the refresh interval of the RP software. Secondly, the RP software is stalled by the creation of very deep sub trees in RPKI repositories. These very long delegation chains consume too many resources during the validation procedure, such that the RP software cannot finish the validation process. As a result, the cached RPKI data expires after a certain period of time and is excluded from the list of VRP. Subsequently, the attacker is able to perform a BGP hijack and attract traffic. The attack was announced as a major security flaw in the RPKI architecture that would significantly reduce the overall level of security. An intense discussion sparked on the IETF routing mailing list [238], pointing out that the authors of RFC 7132 [127] already foresaw such an attack: "An attacker could create very deep subtrees with many ROAs per publication point [...]". In detail, the attack works as follows: Firstly, the attacker needs to obtain the IP address of the legitimate RP that queries the RPKI repositories. They set up a publication point of their own, or they obtain access to an existing one. Since a RP software should fetch ROAs from all publication points, it should also contact the publication point of the attacker. The attacker is now in possession of the IP address of the target RP software. Secondly, the attacker is attempting to blacklist the obtained IP address at the repository holding the ROAs that are targeted for exclusion from the VRP list. This is implemented using a simple trick. Repositories typically use rate-limiting as a protection against DoS. The attacker therefore sends many requests with a spoofed source address towards the repository. The spoofed source address is the legitimate IP of the RP software. It is soon blacklisted and the IP address cannot contact the repository any longer. One thing to consider is that DNS resolvers are typically anycasted. Therefore, the origin for the packets containing the spoofed source address needs to reside within the same anycast domain as the repository. The authors report that 47% of repositories are vulnerable to such an attack. A challenge in performing the attack is the precise timing that is required to be successful. The RP software tries to contact the RPKI repositories at predefined intervals. These are different per RP software [180]. The attacker is required to undermine every single connection attempt as otherwise, cached items would be updated and the expiry timer restarts. Moreover, intentionally forcing an IP address on a blacklist many times in a row would make an attack pattern visible. The second part of the attack relies on the SlowLoris attack [239]. Once a connection is established the server responds very slowly. In addition, many connections are opened, each having its own timeout counter. 53.01% of manifests have an expiration of

24 hours or less. Hence, the attacker is combining the two previously-mentioned methods to stall the RP software longer than 24 hours. Upon expiry of the manifest, all ROAs included within will be removed from the RP software cache and are not included in the VRP that is forwarded to routers. At this point, the attacker can hijack the targeted IP prefix. As a mitigation, the authors suggest implementing a maximum depth of 32 for delegation chains.

Mirdita *et al.* [240] report on 4,344 relying party software instances that were deployed in June 2022. They find exploitable bugs in Routinator and OctoRPKI via means of black box testing that were fixed by the respective developers.

van Hove *et al.* [179] performed a penetration test on several parts of the RPKI infrastructure assuming that an RPKI publication point is compromised. The methodology and results are also available as an IETF draft [241]. rsync has several drawbacks and the IETF already prefers RRDP [242]. Hence, the authors applied the OWASP Top10 REST security vulnerabilities [243] on RRDP connections, which run over HTTPS. Additionally, the Extensible Markup Language (XML) security considerations [244] are applied to the XML formatter, since the payload is formatted in XML. Moreover, an attack using a gzip bomb is performed [245]. Out of 15 attacks, the validators under test were at least susceptible to one. In the aftermath of the study, a Coordinated Vulnerability Disclosure (CVD) process, led by the Dutch NCSC-NL, was initiated. Most vulnerabilities have been fixed, but the disclosure process itself was controversially discussed on the SIDROPS mailing list [246].

Hlavacek *et al.* [185] focused on studying the DNS infrastructure and its implications for RPKI resilience in 2022. Their measurements require DNS data and since the authors do not control the requesting relying party instances, debugging posed a significant problem. They create nested publication points to require the relying party software to issue many DNS queries until the last object is derived from the repository. They track relying party software by redirecting the request to randomly generated subdomains. In order to resolve the subdomain, the relying party software would need to contact the nameserver of the experimenter and therefore the would establish a link between the IP address of the relying party software and the IP address of the resolver. All measurements were conducted between April 2021 and September 2021. They report that 63% of ASes that deploy multiple relying party software instances use DNS resolvers from a single AS. Only 42.8% use a single DNS resolver. This behavior poses an issue when an attacker block access to that single resolver, as the relying party would not be able to receive complete information. As a result, RPKI ROV would be impacted by an attacking the DNS resolution.

Within this section, we have discussed problems related to the centrality of the RPKI architecture, namely the whacking of ROAs, cross country certification and proposals such as blockchain-based approaches and threshold-based approaches as a mitigation to parts of the problem. Several studies have focused on the inconsistency of data in RP software, opening the possibility for a mirror world attack, while others have pointed out that circular dependencies between BGP and RPKI can be problematic. Finally, we looked at attacks that can be launched against repositories or RP software in order to hinder proper RPKI workflow. It turned out that RP software can be stalled, which leads to expired ROAs, and that regular OWASP Top10 attacks succeed against the RRDP protocol.

## 3.5 CONCLUDING REMARKS

Throughout this chapter we have studied many related research works in the three domains: ROA measurements, ROV measurements, and RPKI resilience. The presented proposals are required to understand our approaches in the coming chapters.

Within the ROA measurements category, measurement methodologies have been proposed to reliably identify which IPv4 and IPv6 address space is covered by RPKI ROAs. These methodologies have been implemented in automated monitoring tools such as the NIST deployment monitor [190]. Moreover, we discussed why the normal ROA procedure is suboptimal as it leaves room for prefix hijacks, if the max-length attribute is specified incorrectly, causing a *loose* ROAs. The IETF is currently working on improvements by defining minimal ROAs [196]. *Hanging* ROAs are a concept that improves the compression of ROAs to speed up performance of the RPKI infrastructure. It has not been implemented. There is also the suggestion to increase the use of *AS0* ROAs, which are used to protect unassigned address space and assigned but currently unused address space, to avoid prefix squatting.

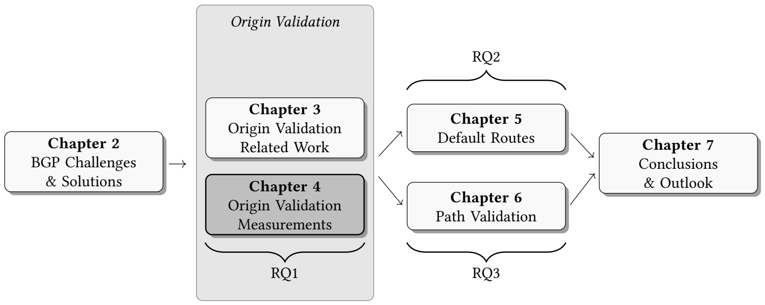
Within the ROV measurement domain, we focused on measurement methodologies to identify BGP prefix origin validation, also called RPKI Route Origin Validation. We are able to identify ROV either on the control plane or the data plane. Measurements can be either controlled or uncontrolled, while we prefer controlled measurements as results are more reliable. Such measurements are always limited by the amount of probes that can be used. They are also dependent on the set of probes. We highlight that some methodologies attribute ROV falsely, as they do not consider transit ASes or IXPs that deploy ROV filtering.

Finally, we looked at RPKI resilience, in which we summarize discussions about the RPKI architecture and its associated problems. Several works have raised the issue of the centrality of the RPKI infrastructure. Since RIRs have the absolute power, they could potentially *whack* ROAs. Moreover, legal issues arise from the choice of architecture. Blockchain proposals try to replace the current workings of the RPKI publication points, and a threshold-based approach aims at introducing a consensus algorithm in signing ROAs. RIRs would only be able to commonly delegate resources, which would remove many problems stemming from the centrality, but would also introduce a lot of additional complexity into the signing procedure. Inconsistencies in the data that RP software obtains from repositories are a looming problem. If RP software cannot obtain a full set of available ROAs, a *mirror world* attack can be mounted. Additional work has been done in inferring the delays introduced at the several spots within the RPKI architecture that lead to delayed routing changes in the BGP ecosystem. Very recent work focused on attacking the RPKI infrastructure itself, either via stalling RP software to achieve expiration of ROA objects or via attacking the RRDP protocol to make delivery of ROAs to RP software unfeasible.

Overall, many research proposals have not been adopted and remain academic as they introduce additional complexity that cannot be tolerated in a routing infrastructure that is already very complex.

CHAPTER 4

# Origin Validation Measurements



*In Chapter 3 we saw the many different directions which existing research has explored. We also saw that existing methodologies of RPKI Route Origin Validation measurements have failed to precisely identify ROV-filtering ASes. In this chapter we propose a new methodology to identify Route Origin Validation. In particular, we extend previous methods on the control plane and develop a new method using data plane measurements to pinpoint ROV-filtering ASes. Our approach is both controlled and rigorous; controlled in that we only rely on information that we are able to influence and rigorous in that we are able to exactly pinpoint filtering to ASes without diluting results via third-party filtering. We also compare existing RPKI relying party software and evaluate the coherence of results amongst one another as well as their performance. Our work has been published as two conference papers [14], [16] and a conference poster [15]. Additionally, we presented RPKI measurement methods in a NOMS tutorial*[a]*. Moreover, our measurement framework has been publicly released*[b]*.*

[a]https://noms2020.ieee-noms.org/program/tutorials.html
[b]https://github.com/nrodday/TMA-21

## 4.1 INTRODUCTION

The previous chapters provided background on origin validation, in particular the RPKI, and related technologies and highlighted existing work in the field. Many drawbacks of existing RPKI ROV identification methodologies that have been pointed out and show that obtaining knowledge about the current adoption is of vital importance but current methodologies lack the ability to allow for bullet-proof conclusions. In this chapter we are going to propose a new method to identify RPKI ROV. We deem it equally important to understand the current deployment of RPKI ROV as it allows conclusions as to how likely it is that new security solutions are implemented and accepted in an existing and complex ecosystem. In particular, we think of path validation algorithms for future deployments. On the one hand, if origin validation, and therefore RPKI as a representative of such, as a first step towards a potentially more secure future is not being adopted, it follows that no one will start to deploy an even more complex technology to protect the whole BGP path. On the other hand, if RPKI is not only used to protect prefix space via RPKI ROAs, but also actively used to drop RPKI invalid announcements, it is much more likely to see path validation algorithms being deployed in the future. Moreover, path validation techniques are only useful if origin validation is already in place and cryptographic objects for path plausibility algorithms are envisioned to use the same RPKI infrastructure. It is therefore key that such infrastructure is productively deployed and accepted by the operator community.

We identified three major directions within the RPKI research community dealing with: (i) ROA measurements; (ii) ROV measurements; and (iii) RPKI resilience. ROA data is publicly available and provided by RPKI repositories. If that data is combined with public BGP dumps available via BGP route collector projects, we are able to obtain a comprehensive picture of the current state of RPKI protected prefix ranges. Tools such as the NIST RPKI deployment monitor [190] implement the aforementioned strategy and supply live data on the current rate of protection. On January 14, 2024, RouteViews indicated 47.81% of prefixes were valid, 0.92% were invalid, and 51.27% did not have a covering ROA.

Measuring ROV adoption is much more challenging. BGP is an information hiding protocol and there are a multitude of reasons why a certain route is potentially preferred over another one or becomes unavailable altogether. We have seen in the previous chapter that researchers have tried to identify via different methods which ASes are deploying RPKI ROV [156], [165], [166]. The underlying problem is the inference of private router configurations. Methodologies typically rely on passively collected BGP data or on active experiments that inject a RPKI valid and a RPKI invalid prefix announcement from the same origin AS in order to observe the difference in handling these two announcements by the same AS under test. Methodologies can be designed for the control plane or the data plane, or a combination of both. Measurements can also be controlled or uncontrolled. Reuter et al. [165] have shown that incorrect attribution is common within uncontrolled measurements that only rely on control plane information. To reduce the amount of independent variables, controlled measurements should be preferred as they exert control over BGP announcements and changes of RPKI objects within the RPKI infrastructure.

## 4.2 LIMITATIONS OF STATE-OF-THE-ART MEASUREMENTS

Reuter *et al.* [165] developed a rigorous methodology on the control plane for the identification of RPKI ROV. However, although their methodology is very precise, it only allows judgements to be made about a small fraction of ASes on the Internet. This is because two requirements are enforced on any AS that is tested:

Firstly, each AS is required to be directly connected to the PEERING testbed [153]. This is called the *connected assumption*. This is to avoid the influence of any intermediate ASes with the propagation of the RPKI anchor and experiment prefixes. If the AS under test is directly connected to the PEERING testbed, we can be sure that any announcement issued from the PEERING testbed is directly received by the AS under test and therefore included in the local decision-making process.

Secondly, any measurement study is limited by the number of vantage points. With respect to RPKI ROV measurements on the control plane, the amount of BGP collector data is crucial. In order to make rigorous inferences on the control plane, the amount is further decreased by enforcing the so-called *visibility assumption*. It requires each AS under test to export its RIB periodically to a BGP route collector.

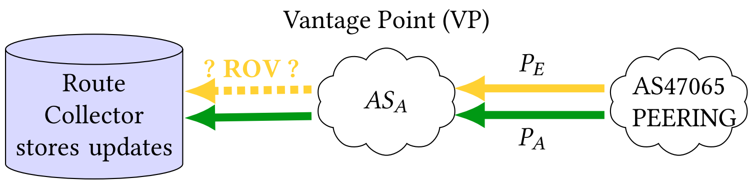


Figure 4.1: Controlled control plane measurements. Two prefixes are announced, an anchor prefix $P_A$ and an experiment prefix $P_E$. Data is only used if the anchor prefix is visible. The experiment prefix is swapped between RPKI valid and RPKI invalid.

Figure 4.1 illustrates the regular operation of the measurement setup. The PEERING testbed announces two prefixes to the AS under test. One prefix is the RPKI valid anchor prefix and it remains valid throughout the duration of the experiments. The other prefix is an experiment prefix that is swapped from RPKI valid to RPKI invalid in a predefined schedule. $AS_A$ under test is only considered if it exports the anchor prefix to a BGP collector.

**Removing the Connected Assumption.** The connected assumption is a strong limitation as it limits the ASes that we are able to measure to direct BGP peers of the PEERING testbed. While the connected assumption enables absolute confidence in the results, it limits the methodology to only 86 ASes out of 74k, or 0.1%. We propose to remove the limitation, but only in the cases where each and every AS along the propagation path is a vantage point, see Figure 4.2. If $AS_A$ exports both routes, the

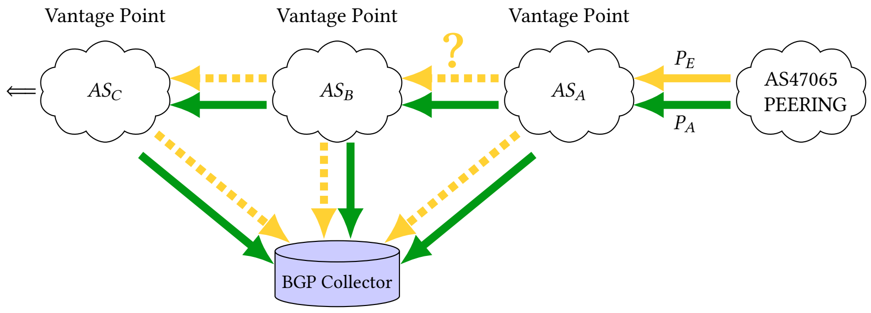
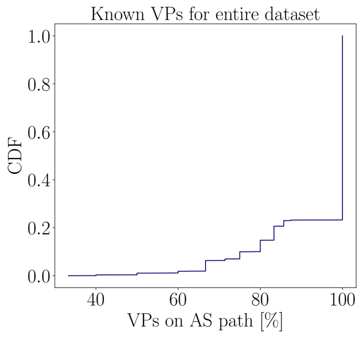


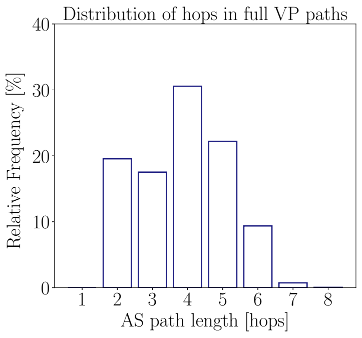
Figure 4.2: Removing the connected assumption. Two prefixes are announced from the PEERING testbed, the anchor RPKI valid, the experiment prefix swapping between RPKI valid and invalid. We extend prior measurements by also making judgements about longer path as long as all ASes are vantage points and therefore export their RIB tables to BGP collector projects.

RPKI valid anchor prefix and the RPKI invalid experiment prefix, to a route collector, we conclude that it does not perform RPKI ROV. Obviously, for a longer chain, the methodology is only capable of identifying the first AS that performs RPKI ROV filtering. Subsequent ASes do not receive the experiment prefix anymore and can therefore not discard it, even if they do deploy RPKI ROV.

Removing the connected assumption weakens the obtained results, as we might



(a) Known VPs for entire dataset



(b) Distribution of hops in full VP paths

Figure 4.3: Meaningfulness of proposed extensions. We observe that the majority of BGP paths in our dataset contain vantage points. The average path length is 3.87 hops. BGP paths longer than six hops are extremely rare. The dataset was collected between August 22–29, 2019 for our anchor and experiment prefixes announced from the PEERING testbed.

observe partial adoption in some corner cases. An AS might apply RPKI ROV on one link, but not on another. If we observe $AS_A$ to export the anchor and the experiment prefix to a route collector, but it only forwards the anchor to $AS_B$ and filters the experiment prefix, $AS_B$ would only be capable of exporting the anchor to the route collector. Such behavior would incorrectly attribute RPKI ROV to $AS_B$, while $AS_B$ might not be filtering at all and instead $AS_A$ performs partial filtering. We explore partial adoption scenarios later in our methodology section.

**Relaxing the Visibility Assumption.** Any study of the control plane is limited by the number of vantage points exporting data to public BGP collectors. This becomes even more crucial in the context of ROV measurements because they require path-specific analysis and ROV deployment is still limited.

Previous measurements only considered ASes that exported BGP data to collector projects. Unfortunately, collector projects only capture a fraction of the overall ASes. To widen the scope of measurements we propose to also consider isolated non Vantage Points. In Figure 4.4, $AS_B$ is surrounded by Vantage Point ASes, $AS_A$ and $AS_C$. We also know, because of prior measurements, that $AS_C$ does not perform RPKI ROV. Hence, we are able to tell whether $AS_B$ drops RPKI invalid routes without having direct access to its exported RIB. Similarly to the removal of the connected assumption, it introduces the uncertainty of partial adoption and therefore lowers the level of certainty of our inferences.

**Extension of Range.** Overall, our proposed extensions increase the potential range of the experiments by 474%. While the original methodology could only consider 86 ASes, we add 166 ASes by removing the connected assumption and add another 156 ASes by relaxing the visibility assumption. Figure 4.5 shows the improvements in a horizontal bar plot.

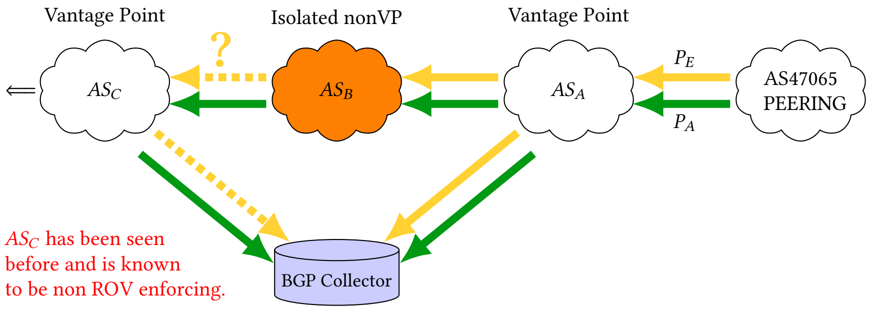


Figure 4.4: Relaxing the visibility assumption. Two prefixes are announced from the PEERING testbed, the anchor RPKI valid, the experiment prefix swapping between RPKI valid and invalid. We extend prior measurements by also making judgements about longer paths including isolated non Vantage Points, as long as all other ASes are vantage points. Additionally, we need to be sure that downstream ASes to the non vantage point AS under test are not performing RPKI ROV filtering.

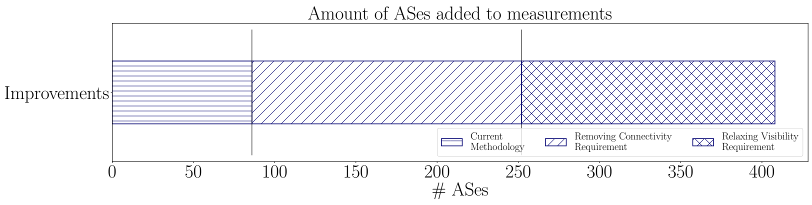


Figure 4.5: Improvements: The original methodology includes 86 ASes, removing the connected assumption adds 166 ASes, relaxing the visibility assumption adds another 156 ASes. The overall improvement is 474%.

It is clear that the range of these measurements remains limited, considering the overall number of ASes on the Internet, which is currently about 74,110. However, an increase in the range of experiments conducted from the PEERING testbed enables us to obtain a greater sample size and therefore generalize findings.

## 4.3 INCORRECT ATTRIBUTION OF DATA PLANE MEASUREMENT METHODOLOGIES

This section highlights issues introduced by end-to-end RPKI ROV identification methodologies such as ICMP and HTTP-based measurements. Cartwright-Cox [167] used ICMP scans in his methodology, which was published in 2019. The RPKI WebTest [170] and the online tool set up by Cloudflare [173] both use HTTP queries. Both proposals only allow a binary result. The target is either reachable, or not. This constitutes a problem when performing RPKI ROV measurements on paths that are further than a single hop away from the probe, since intermediate ASes could be filtering, but the methodology would wrongly attribute the filtering to the AS under test. For example, the RPKI WebTest and Cloudflare online-based test both issue HTTP requests to an RPKI valid and an RPKI invalid range. If the RPKI valid prefix could be reached, but the RPKI invalid one could not be reached, they report to the user that the ISP of the user supports RPKI ROV. This inference might be incorrect, it could also be the upstream, another transit AS or a transparent IXP that is filtering. End-to-end methodologies therefore do not allow the pinpointing of ASes for ROV filtering.

**Reproduction of HTTP measurements.** In order to show wrong attribution we reproduce the measurements. Similar to the RPKI WebTest [170] and Cloudflare [173] online test, we announce two distinct /24 prefixes. To observe potential differences that stem from the location of the announcement, we use 11 separate PEERING testbed PoPs [247], which are geographically distributed according to Table 4.1. We announce our anchor and experiment prefixes to all peers of the respective PoP (directly and via routeservers) to obtain the best connectivity possible. To avoid upstream filtering of our routes, we create IRR objects and present each upstream of a PoP with a Letter of Authorization (LoA) to clear our prefix ranges.

Table 4.1: Points of Presence at the PEERING testbed.

| Name | Upstream/IXP | Geographic Location |
|---|---|---|
| ams01 | AMS-IX* | Amsterdam, NL |
| clemson01 | Clemson University | Clemson, USA |
| cornell01 | Cornell University | Cornell, USA |
| neu01 | Northeastern University | Boston, USA |
| isi01 | Los Nettos Regional Network | Los Angeles, USA |
| gatech01 | Georgia Institute of Technology | Atlanta, USA |
| grnet01 | GRNet | Thessaloniki, GR |
| seattle01 | Seattle-IX* | Seattle, USA |
| uw01 | University of Washington | Seattle, USA |
| ufmg01 | Rede Nacional de Ensino e Pesquisa (RNP)* | Sao Paulo, Brazil |
| utah01 | Utah Education Network | Salt Lake City, USA |

* Our AS is also directly connected to the route server of the IXP.

After a period of debugging and elimination of connectivity and filtering issues, we are able to conduct the planned measurements. Measurements are scheduled for a whole month and executed once per day. We set up an HTTP server (nginx) to serve HTTP queries and use RIPE Atlas to send HTTP requests from all over the world to each of the prefix ranges. The traffic from all prefixes is forwarded to our control server hosting the HTTP server. To conduct ethical experiments, each probe carries a disclaimer in the payload section that links to a website explaining our experiments and providing the possibility to complain and opt-out. Throughout our entire measurement period we received no such requests.

Similar to [170] and [173], if the HTTP probe could reach the server within the RPKI valid prefix range, but not within the RPKI invalid prefix range, ROV is attributed to the AS that hosts the RIPE Atlas probe.

Firstly, we verified whether multiple probes within the same AS would lead to the same results. Most of our ASes in Figure 4.6 are only covered by one probe (2398), the second largest share in green is displaying multiple probes per AS, however results are consistent among the probes (between 946-1025). Exceptions are displayed with the red bars which indicate when multiple probes for an AS were present and delivered inconsistent results (between 0-79). We conclude that the consistency of results per AS is very high. Either an entire AS can or it cannot reach a certain prefix range.

Secondly, we plot the ratio of the number of failed HTTP requests to prefixes of invalid routes and the number of successful HTTP requests to prefixes of valid routes in Figure 4.7. We observe connection failures in about 60% of the cases for the prefixes announced to all peers at AMS-IX. The operators of AMS-IX have been filtering RPKI invalid prefixes on route server sessions since October 20, 2017 [165]. Therefore, data plane probes that would only have connectivity via route server

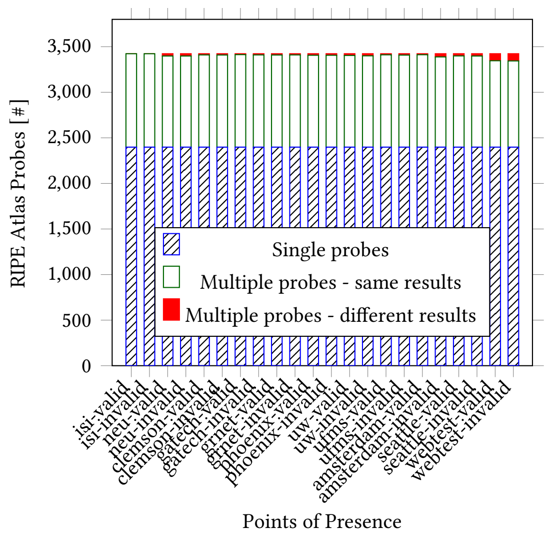


Figure 4.6: Consistency of probe results. For the majority of Atlas probes it is only a single probe within an AS (blue bars). When multiple probes are present, the connectivity results are mostly the same (green bars). In very few cases results differ for multiple probes within the same AS (red bars).

sessions are not able to reach their destination. During our measurement windows of 31 days, we started announcing a dedicated prefix pair to only the route servers on day 17. Figure 4.7 clearly shows that 80% of AMS-IX members do not opt out from the RPKI filtering service (red line). Therefore, they do not receive the RPKI invalid prefixes and cannot connect to the IP addresses within.

Next, we show why simple end-to-end measurement methodologies fail to attribute ROV correctly. We create a hitlist of probes that were able to reach the RPKI valid but not the RPKI invalid prefix range. Table 4.2 displays an excerpt of the hitlist for day 20 which is representative for other days in this study. At the top of the list we identify five probes from New Zealand. Manual investigation revealed that all probes are sitting behind the same upstream provider AS38022 REANNZ-NZ-A. This is a strong indicator that the upstream provider and not the ASes of these probes themselves are enforcing ROV. We contacted the operator of AS38022 to confirm our findings. The ASes hosting the probes are not filtering based on RPKI, but AS38022 performs RPKI filtering.

**Reproduction of ICMP measurements.** We also reproduced the ICMP scans by Cartwright-Cox [167]. Again, we used the PEERING testbed to announce RPKI valid and RPKI invalid prefix ranges from different PEERING PoPs. According to our results, filtering transits have a great impact on this methodology. If the filtering transit is closer to the PEERING testbed within the AS path, many more ASes are flagged as ROV enforcing. Additionally, the location of the PEERING PoP influenced meas-

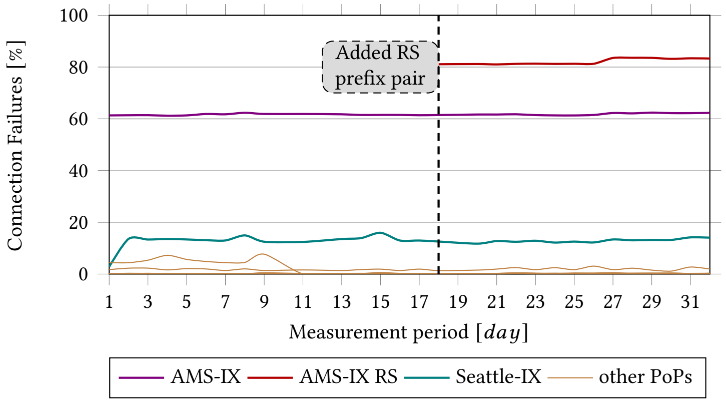


Figure 4.7: Comparison of the reachability of HTTP measurements initiated from RIPE Atlas probes to RPKI valid and invalid routes, which were announced at different locations in January 15–February 15, 2020. Connectivity is impacted by ROV-filtering at AMS-IX and, to a lesser extent, at Seattle-IX.

urement results. Depending on the PoP, measurement results for the same set of ASes change. This is unfortunate, as a measurement result should be independent from the location where it is executed. It highlights the fact that intermediate ASes performing RPKI filtering appear on some paths but not on others, depending on the location of the announcement. One more thing to consider is Reverse Path Filtering (RPF). Filtering upstreams drop data plane packets directly upon reception. The methodology, however, assumes that the return packet is not transmitted correctly. In reality the forward path is not working; therefore, attribution of ROV is not working as intended and false positives are introduced.

In summary, we have reproduced HTTP and ICMP end-to-end measurements. Our results show that filtering is sometimes attributed to the wrong ASes, since fil-

Table 4.2: Hitlist for HTTP methodology for day 20. We observe five probes at the top to reach the RPKI valid but not RPKI invalid prefix range, implying RPKI filtering. They are, however, connected to the same upstream AS38022 that performs the RPKI filtering for them.

| ProbeID | ASN | AS Name | Country | Score |
|---|---|---|---|---|
| 21069 | 9433 | MASSEY-AS | NZ | 11 |
| 306 | 45131 | REANNZ-OFFICE | NZ | 11 |
| 93 | 681 | ERX-KAWAIHIKO | NZ | 11 |
| 12197 | 9431 | AKUNI-NZ | NZ | 11 |
| 26218 | 9433 | MASSEY-AS | NZ | 11 |
| ... | | | | |

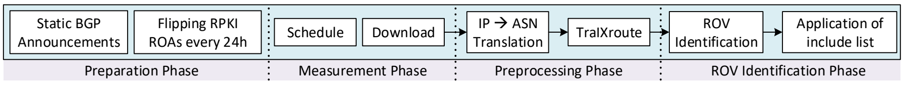


Figure 4.8: Measurement Phases

tering is often performed at the upstream that provides connectivity to leaf ASes. These methodologies do not reliably allow the inference of ROV filtering and they are especially unsuitable to pinpoint ASes that are supposed to perform RPKI filtering.

## 4.4 METHODOLOGY AND SETUP

In order to reliably pinpoint ASes that are filtering RPKI invalid announcements, we propose a new *data plane* methodology. Instead of looking at the path as a whole and obtaining binary results for the entire path, we propose a hop-wise approach. We use extensive and reproducible data plane measurements via traceroutes issued from RIPE Atlas [210]. Our measurements are controlled and active. We use the PEERING testbed [247] to announce our BGP prefixes and deploy an RPKI child-CA to create and alter RPKI objects for our prefix ranges. To this end, our parent CA delegates the IP resources used in the experiments to our child-CA. The four phases of our study can be seen in Figure 4.8. Figure 4.9 illustrates the architecture of the technical components required for the experiments.

**Preparation Phase.** During the preparation phase our measurements are set up. We select five PoPs from the PEERING testbed, see Table 4.3, and announce static BGP prefix pairs to each of them. Each pair consists of an anchor as well as an experiment prefix. Both ranges have IRR entries and are cleared with the respective upstreams. Additionally, the chosen ranges are always neighbouring ranges, *e.g.,* `147.28.12.0/24` and `147.28.13.0/24`. This is to avoid unexpected behavior with old filtering rules and other issues. For each PoP we announce to all peers, except for PoPs with IXP connectivity, such as AMS-IX and Seattle-IX. Here, a second prefix pair is announced to only the IXP route servers in order to make inferences

Table 4.3: Points of Presence at the PEERING testbed.

| Name | Upstream/IXP | # Direct Peers |
|---|---|---:|
| ams01 | AMS-IX* | 123 |
| gatech01 | Georgia Institute of Technology | 1 |
| grnet01 | GRNet | 1 |
| seattle01 | Seattle-IX* | 72 |
| uw01 | University of Washington | 1 |

* Our AS is also directly connected to the route server of the IXP.

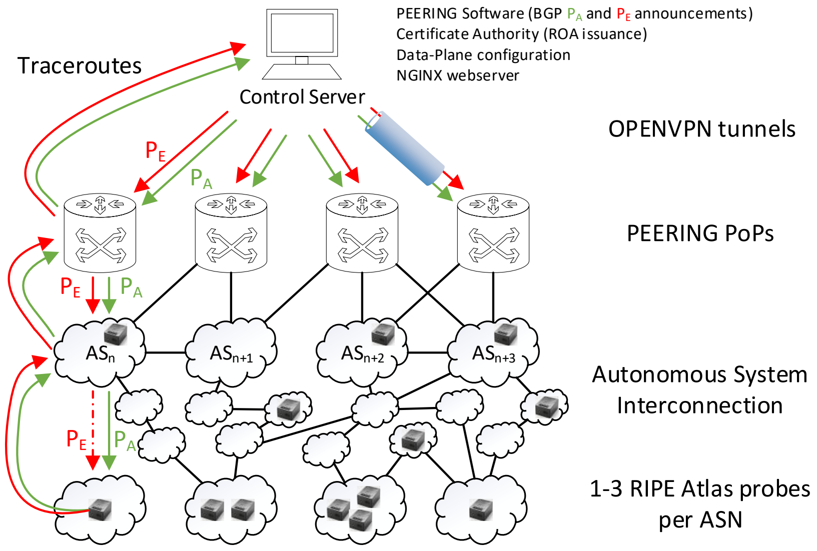

PEERING Software (BGP $P_A$ and $P_E$ announcements)
Certificate Authority (ROA issuance)
Data-Plane configuration
NGINX webserver
Control Server
Traceroutes
OPENVPN tunnels
PEERING PoPs
$P_E$
$P_A$
$AS_n$
$AS_{n+1}$
$AS_{n+2}$
$AS_{n+3}$
Autonomous System Interconnection
1-3 RIPE Atlas probes per ASN

Figure 4.9: Experiment setup. The control server connects via OpenVPN tunnels to the PEERING PoPs, which announce the prefix ranges. RIPE probes in ASes send traceroutes towards our control server.

about their behavior. In total, we utilize 14× /24 IP prefixes.

We sign and publish the ROAs for our experiments at a predefined schedule within our child-CA, see Figure 4.10, which was recorded by RIPEstat [248]. The anchor prefix remains RPKI valid, while the experiment prefix is swapped between an RPKI valid and invalid state to test for RPKI-related filtering. This behavior is clearly visible as the horizontal bar for the anchor prefix `147.28.12.0/24` remains green, which implies that many collectors export this prefix, while the representation of the RPKI state for the experiment prefix `147.28.13.0/24` changes to red, representing only few collectors that export this prefix. The ROAs are swapped at midnight, but since RPKI propagation and the resulting changes in BGP route propagation and convergence, as well as exporting the new routes to route collectors, consume some time, a small delay is visible.

Originally, we planned to adopt the ROA schedule from [165] which has an eight hours swapping interval. However, during the analysis of our first trial runs, we observed inconsistencies in the results, which lead to a manual investigation. Our initial assumption that all RPs fetch and update their routers within a short period of time does not hold. We observed that some ASes still maintain routes according to stale ROAs. To account for these long adoption times we changed the ROA schedule to a longer period of 24 hours valid and 24 hours invalid. This gives RPs ample time to update their routers. Our observations triggered a discussion within the IETF SIDROPS working group that led to a draft to narrow down timing parameters for the RPKI supply chain [13]. Kristoff *et al.* [180] later published a paper dedicated to timing parameters in the RPKI.

**Measurement Phase.** In our data plane measurements, we randomly select three RIPE Atlas probes per AS that have RIPE Atlas coverage. In total, we select 5,537 probes in 3,694 ASes. We therefore cover each AS with 1.49 probes on average. From each probe we send traceroutes to the `.1` address within the anchor and experiment prefix range per PoP, *e.g.,* `147.28.12.1/24` and `147.28.13.1/24`. Since we flip ROAs every 24 hours, a single measurement run takes 48 hours to complete. Our five measurement runs are performed on 17 days, July 2–19, 2021. While it should theoretically only take 10 days, we experience unforeseen outages either within the PEERING testbed or the RIPE Atlas infrastructure that we explain in detail in the lessons learned in Section 4.6. For the impacted days, measurements are extended for another 24 hours and redone, data is discarded. For all experiments we use RIPE Atlas APIv2 in combination with the Cousteau Library [249] for scheduling. In Figure 4.9 we show the experiment setup.

**Preprocessing Phase.** Before beginning to process the data, we download all available datasets via the RIPE Atlas API to our local processing facilities. This process only takes minutes, the data is provided in json format. We take a two step approach. Firstly, we need to translate traceroute paths into ASN paths. This is done by mapping IP addresses to ASNs. Secondly, we perform an analysis of the data that highlights IP addresses that belong to IXPs and as such identify traces that crossed IXP facilities.

A single measurement run contains 38.5k IP addresses. We extract them from the provided json files and apply the following procedure to obtain AS paths, see Listing 4.1. Firstly, we identify 6.5k IP addresses from private IP address ranges, see Section 2.3 and RFC 1918 [250]. These are not considered further, since they should not appear in public routing in any case and cannot be mapped to an AS since they are reused within any arbitrary network. We find that in 5,174 traces, 3,380 (65%) paths include at least one private IP address. The ratio of private IP addresses in public traces is surprisingly high, which highlights the fact that many misconfigurations exist. Secondly, we identify 23 IP addresses from the PEERING testbed. Thirdly, we use Team Cymru [251] to attempt mapping of the remaining 32.4k IP addresses to ASNs. We are able to resolve 30.3k addresses via this method. Fourth, the remaining addresses are run through a tool called PyASN[1]. We use BGP

[1]https://pypi.org/project/pyasn/

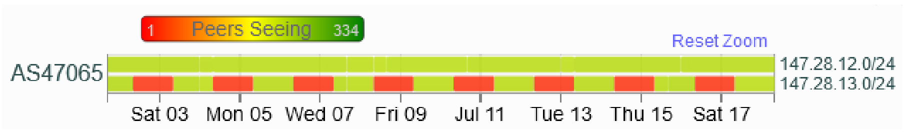


Figure 4.10: RIPEstat Routing History for one of our prefix pairs. Visibility of the anchor prefix remains stable while visibility of the experiment prefix drops significantly when ROA configuration leads to invalid routes.

collector dumps from RIPE RIS [146] and Routeviews [140] within PyASN during our measurement window to find ASNs that announced the IP address ranges in which our remaining IPs are contained. We are able to resolve another 200 addresses. Finally, 1.8k IP addresses remain that cannot be resolved by any means.

Listing 4.1: IP to ASN mapping for a single run.

```
~38.5K IP Addresses
↪ ~6.1k Resolved to private
 ↪ ~23 Resolved to PEERING testbed
  ↪ ~30.3k Resolved with Team Cymru
   ↪ ~0.2k Resolved with pyasn
    ↪ ~1.8k Remain unresolved
```

Next, we need to sanitize the obtained AS-level paths. We apply three simple steps:

1. If the adjacent (left and right) IP addresses belong to the same ASN, we remove the private IP addresses in-between from the AS path.
2. If the adjacent (left and right) IP addresses belong to the same ASN, we remove unresponsive hops (*i.e.,* "*").
3. If an AS is prepended multiple times (*i.e.,* sequence of duplicate ASNs), we merge them into a single instance.

Listing 4.2 presents an example of our ASN path sanitization method. We observe an AS path obtained from a RIPE Atlas traceroute in the first line. It contains the same AS multiple times, with private and unresponsive hops inbetween. After applying our methodology, we observe the result in the second line. The sanitized AS path no longer contains artifacts and redundant information. Such a path can be used for our ROV methodology.

Listing 4.2: Reduction of an AS Path

```
48147, private, 48147, 200612, 3257, *, 3257, 209, 2722, 47065
48147, 200612, 3257, 209, 2722, 47065
```

However, many paths do not end up as AS paths that fulfil our requirements. We exclude all traces that: (*i*) do not have adjacent (left and right) public IP addresses to a private IP address; and (*ii*) AS paths that exhibit multiple unresponsive hops in sequence where the first and last unresponsive hop are connected to different ASes. In such cases, we cannot be sure that there have not been single or multiple additional ASes inbetween that did not respond entirely.

To filter out measurement noise, we execute the exact same measurement three times shortly after each other within one measurement run. Afterwards, we verify that the obtained traceroute results for each source and destination pair are consistent throughout the three samples. We observed variations in only 0.5–1% of our measurements. They can mostly be attributed to load balancers within the same AS that have different IPs assigned. These do not change the translated AS-level paths, since all such IPs are mapped to the same AS. In some instances, inter-domain changes happen, which changes the obtained AS-level path and makes the traceroutes unusable for our ROV inferences.

After translating traceroutes to AS-level paths, we attempt to identify traces that crossed an IXP. IXPs are a challenge for ROV inference as they work transparently on the network layer but might deploy ROV by themselves. It is also common policy that ASes do not insert their own AS into the BGP AS path attribute. ASes sitting behind those route servers would receive *collateral benefit*, although the IXP itself does not show in the BGP AS path attribute and therefore cannot be identified. We attempt to identify IXPs to differentiate between ROV filtering that is performed by an AS and ROV filtering done by the routeserver.

To facilitate IXP detection in traceroute paths, we use a tool called TraIXroute [252]. Two main mechanisms are used to perform such identification: Firstly, the use of IXP membership datasets which contain IXP peering Local Area Network (LAN) addresses and AS-to-facility mappings. Secondly, the detection of IXP prefixes. The underlying data sources are PeeringDB [253], PCH [254], and Routeviews [140]. Nomikos *et al.* [255] show how TraIXroute is used to uncover IXP peerings.

Before implementing TraIXroute into our methodology chain, we want to evaluate its performance and detection capabilities. We use the RIPE Atlas measurements towards a prefix pair that is announced to only the route servers of the AMS-IX. By limiting our analysis to only this BGP peer, we are certain that RIPE Atlas measurements must cross the IXP facilities. Surprisingly, TraIXroute only identifies 1.7% of our paths as traceroutes that have crossed an IXP facility. This is a very low detection rate, since all of the traces are actually running via the IXP. We confirm the results with a manual investigation. It reveals that the vast majority of traces running via the AMS-IX do not contain the IP addresses around the IXP hops. These are essential for correct identification, therefore making it impossible for the tool to correctly determine IXP paths. For traceroutes towards the anchor prefix announced to all AMS-IX peers, TraIXroute flags roughly 10% of traces as having crossed IXP facilities. This also includes other IXPs, not only AMS-IX. We conclude that routes via other peers at AMS-IX appear to traverse other IXPs more often. These other IXPs also respond with IP addresses to ICMP TTL exceeded messages, making it easy for the tool to correctly flag IXP crossings. In summary, TraIXroute is an addition to our tool chain, but it does not yield the anticipated benefit in identifying IXPs with high confidence. In some cases, we might therefore still cross IXPs that remain undetected by our method.

**ROV Identification Phase.** Since we perform *controlled* experiments, we only consider measurements in which the anchor prefix could be reached throughout the whole measurement period and the experiment prefix could be reached during the period where the ROA status of the experiment prefix rendered the BGP announcement valid. We therefore have sufficient visibility to proceed with ROV identification. In addition, the paths of both the anchor prefix and the experiment prefix during the valid period must exhibit the same sequence of ASes. This is to avoid unexpected routing behavior. If any of the above requirements are not met, data is discarded. RIPE Atlas probes are not always alive or have sufficient capacity to run our experiments for all the PoPs for which we announced prefix pairs. As a result, the usable data per PoP varies. In general, we selected 5,362 probes to execute our measurements. 5,143 probes (95.91%) were able to execute traceroutes towards all of our announced prefix pairs for each PoP. After applying the two requirements introduced before, 2,000–2,500 probes (37–47%) remain that deliver usable data for our inference methodology.

Within this subset of data, we adhere to the following procedures to classify ASes. If a probe is able to reach the anchor prefix but not the experiment prefix during the invalid period, at least one AS en route must be enforcing RPKI ROV. If a probe reaches the the anchor prefix and the experiment prefix during the invalid period, but the path towards the experiment prefix diverges from the anchor prefix path, an AS on the anchor prefix path must be enforcing RPKI ROV. If the probe is able to reach both the anchor and the experiment prefix during the invalid period, and both paths remain the same, no RPKI ROV is deployed in any AS in the path. Alternatively, the RPKI ROV deploying AS could have default routes installed, which still provides data plane connectivity for the prefixes filtered on the control plane. This would introduce false negatives. We investigate default routes in Chapter 5.

In order to more precisely differentiate the many scenarios that result from our experiments, we introduce six categories. We show our six cases with simple examples in Table 4.4. The first three cases respect the *connected assumption* from earlier work [165]. Every AS is required to directly connect to the origin network. Since we use the PEERING testbed to announce our prefixes, only ASes directly connected to the PEERING testbed are considered. Such inferences we call *strong inferences*. The following three cases relax the *connected assumption* and expand measurement coverage, as proposed in Section 4.2. This allows the inference that RPKI ROV also functions for ASes not directly connected to the PEERING testbed. Inferences made in these three categories are considered *weak inferences*. Within each row in Table 4.4, we show the current ROA state of the experiment prefix. Our ROA configuration either leads to a valid or an invalid state of the BGP announcement.

*1 hop—Full reachability without route divergence:* This is the default case. All four traceroutes are able to reach their targets and the BGP path is exactly the same. No ROV is deployed. We add the single AS on the experiment prefix's AS path to the include list. ASes added to the include list are underlined in Table 4.4.

*1 hop—Invalid fail:* All four traceroutes reach their targets and exhibit the same path, except the traceroute towards the experiment prefix when our ROA configuration renders the BGP announcement RPKI invalid. We do not observe an answer for this traceroute. The red font in Table 4.4 shows the non-responsive traceroute in this case. This behavior is attributed to RPKI ROV and the AS under test is flagged as ROV enforcing.

Table 4.4: Overview of heuristics to detect ROV using `traceroute` to the anchor ($P_{\text{anchor}}$) and experiment ($P_{\text{experiment}}$) prefix. Bold ASNs deploy ROV, underlined ASNs are added to our include list, a green AS path represents a path when the route to our experiment prefix is valid, a red AS path illustrates a path change because of an invalid route.

| | ROA | | Traceroute | | Example AS paths | |
|---|---|---|---|---|---|---|
| **Case** | State | ROV | $P_{\text{anchor}}$ | $P_{\text{exp}}$ | Probe ⟶ Valid Prefix | Probe ⟶ Invalid Prefix |
| ***1 hop*** | | | | | | |
| Full reachability w/o route divergence | Valid | ✘ | ✓ | ✓ | [111 - 47065] | [111 - 47065] |
| | Invalid | | ✓ | ✓ | [111 - 47065] | [111 - 47065] |
| Invalid fail | Valid | ✓ | ✓ | ✓ | [111 - 47065] | [111 - 47065] |
| | Invalid | | ✓ | ✘ | [**111** - 47065] | [111 - "*" - "*"] |
| Route divergence | Valid | ✓ | ✓ | ✓ | [111 - 47065] | [111 - 47065] |
| | Invalid | | ✓ | ✓ | [**111** - 47065] | [111 - 222 - 47065] |
| ***2+ hops*** | | | | | | |
| Full reachability w/o route divergence | Valid | ✘ | ✓ | ✓ | [111 - 222 - 333 - 47065] | [111 - 222 - 333 - 47065] |
| | Invalid | | ✓ | ✓ | [111 - 222 - 333 - 47065] | [111 - 222 - 333 - 47065] |
| Invalid fail | Valid | ✓ | ✓ | ✓ | [111 - 222 - 333 - 47065] | [111 - 222 - 333 - 47065] |
| | Invalid | | ✓ | ✘ | [**111** - 222 - 333 - 47065] | [111 - "*" - "*" - "*" - "*"] |
| Route divergence | Valid | ✓ | ✓ | ✓ | [111 - 222 - 333 - 47065] | [111 - 222 - 333 - 47065] |
| | Invalid | | ✓ | ✓ | [111 - 222 - **333** - 47065] | [111 - 222 - 666 - 444 - 47065] |

*1 hop—Route divergence:* Similar to the previous case, all four traceroutes reach their targets and exhibit the same path, except the traceroute towards the experiment prefix when our ROA configuration renders the BGP announcement RPKI invalid. Unlike the previous case, we do not observe unresponsive hops in-between, but find that the traceroute is received and answered, but via a different path. We infer ROV in the directly connected AS under test. The reason for this behavior is that the AS under test could: (*i*) deploy RPKI filtering but has a default route installed that provides connectivity to an upstream provider. The data plane packets responding to the traceroute are therefore forwarded to the upstream, which in turn forwards the packets via its own routing table to our control server. Hence, we observe a longer and indirect AS path; (*ii*) The AS under test could deploy partial filtering. While it filters or depreferences RPKI invalid announcements on the peering session with the PEERING testbed, it does not filter RPKI invalid routes on other links, such as an upstream peering link. Therefore, announcements received via an upstream can either be the only ones connecting the AS under test to our control server, or receive a higher preference via the upstream when the direct announcement was depreferenced. In any case, the AS under test deploys RPKI ROV filtering.

*2+ hops—Full reachability without route divergence:* This is the default case for multiple hops. All four traceroutes are able to reach their targets independently of our ROA configurations and the BGP path is exactly the same. No AS en route performs RPKI-based filtering. We add ASes on the path to the experiment prefix to the include list. ASes added to the include list are underlined in Table 4.4.

*2+ hops—Invalid fail:* Inferences for longer paths are more tricky. Based on our proposal from Section 4.2 we relax the *connected assumption*. We therefore enforce the *visibility assumption* and require every AS on the anchor path to be a vantage point. Since we are performing data plane measurements, they need to host a RIPE Atlas probe. With the probe in place, we are able to iteratively verify which ASes still have connectivity, and therefore do not filter RPKI invalid announcements, and at which point in the AS path connection is lost. Moreover, we use data from the strong 1-hop inference cases. If an AS was already flagged as RPKI filtering, any traceroutes that cross the filtering AS are discarded. By means of exclusion, we are able to identify the filtering AS.

*2+ hops—Route divergence:* Similar to the previous route divergence case, all traceroutes reach their target, but the traceroutes for the experiment prefix during our ROA configuration for the invalid time frame exhibit a different AS path. In order to isolate the divergence, we strip the anchor prefix' AS path prefix and suffix from the experiment prefix AS path. Since the divergence could potentially include more than one AS, we only consider traces that leave us with exactly one AS within our isolated route divergence. We apply this additional restriction since it would not be possible to tell with a high level of certainty within which AS in the remaining path ROV is deployed. By only considering cases that leave us with a single divergence point, we are sure that this particular AS is performing RPKI ROV. In Table 4.4, AS333 is flagged as ROV-enforcing.

**Include List.** An AS could potentially activate RPKI ROV on some peering links, but not on others. Such behavior is called partial filtering. To account for the difference between fully and partially filtering ASes, we make use of an include list. ASes that have been seen forwarding RPKI invalid announcements are added to the include list. If the same AS is observed filtering based on RPKI, we conclude that this particular AS filters on some peering sessions, but not on other. It is therefore only partially filtering.

**Internet Exchange Points.** In addition to partial filtering, we look for IXPs in traceroutes. The methodology has been outlined in the previous section. We are therefore able to differentiate between direct ROV filtering at an AS and indirect ROV filtering performed by route servers at an IXP.

## 4.5 RESULTS

We show the results of our ROV identification methodology in Table 4.5. The numbers indicate the amount of ASes in the respective category. The presented results include all prefix pairs that were announced via all five PEERING PoPs. The first section called *1 hop* contains measurements that satisfy the *connected assumption*. We find ≈41 ASes *without route divergence*. The other two subcategories *invalid fail* and *route divergence* together contain many more ASes. The reason for these numbers is ROV filtering at routeservers of the AMS-IX and Seattle-IX. The IXPs cover the majority of the PEERING testbed's direct peers, see Table 4.3.

Table 4.5: Results of ROV data plane measurements for all PoPs [# ASes]

| | Measurement Run | | | | |
|---|---|---|---|---|---|
| Case | 1 | 2 | 3 | 4 | 5 |
| ***1 hop*** | | | | | |
| Full reachability w/o route divergence | 43 | 43 | 42 | 41 | 37 |
| Invalid fail | 181 | 181 | 182 | 175 | 181 |
| Route divergence | 15 | 15 | 15 | 13 | 12 |
| ***2+ hops*** | | | | | |
| Full reachability w/o route divergence | 803 | 798 | 775 | 731 | 711 |
| Invalid fail | 2 | 2 | 2 | 2 | 1 |
| Route divergence | 11 | 10 | 11 | 7 | 8 |
| Total unique ROV | 206 | 205 | 202 | 194 | 199 |
| Added to include list | 630 | 628 | 626 | 628 | 587 |
| Partially filtering | 60 | 58 | 54 | 55 | 48 |
| Fully filtering | 146 | 147 | 148 | 139 | 151 |

The second section called *2+ hop* contains measurements that relax the *connected assumption*. We are able to consider longer paths in this category. 731–803 ASes are not filtering based on RPKI. However, we also find ≈9–13 ASes in the two subcategories *invalid fail* and *route divergence* together that do deploy RPKI ROV. One might be surprised by the low number of positive inferences. This is due to the fact that our methodology very cautiously and conservatively relaxes the connected assumption. We enforce many restrictions on longer paths. It is designed to be rigorous and therefore optimized to find true positives, but without integration of false positives. The high level of certainty is, however, traded for a lower amount of inferences, which might lead to false negatives. This fact was also correctly pointed out by Hlavacek *et al.* [172].

In total, with 5,537 RIPE probes covering 3,694 ASes, we identify 194–206 ROV-enforcing ASes. Out of those, 48–60 have been observed forwarding invalid announcements on some peering sessions, but filtering according to RPKI on some other peering sessions, and are therefore partially filtering. We classify the remaining 139–151 ASes as fully filtering. They have only been observed filtering, but never forwarding invalid announcements.

**ROV at Route Servers.** For all ASes peering directly with the PEERING testbed at AMS-IX or Seattle-IX in addition to route server peerings (1 hop case), we are able to make more detailed inferences, see Table 4.6. In order to differentiate between filtering at the IXP or the AS itself, we announced a prefix pair to *route servers only* and an additional prefix pair to *all peers excluding route servers*. This is important as any AS can opt-out of RPKI ROV filtering at route servers at any time. If a ROV inference is made via the *route servers only* prefix pair, but not via the *all peers excluding*

Table 4.6: IXP filtering for 1-hop *invalid fail* and *route divergence* cases of ROV inferences for AMS-IX and Seattle-IX members [# ASes]

| PoP | Total ROV | Route server | Direct | Both |
|---|---|---|---|---|
| AMS | 165 | 160 | 9 | 4 |
| Seattle | 33 | 33 | 2 | 2 |

*route servers* prefix pair, we are able to conclude that ROV filtering is performed by the route servers of the IXP. This is the case for 160 AS members of AMS-IX and 33 AS members of Seattle-IX. IXPs also offer direct peering at their facilities. However, ROV is only deployed on route servers, direct peerings cannot be impacted by filtering mechanisms at the IXP level. If filtering on direct peering links is detected, it is deployed within the AS under test. We detected filtering directly in the AS under test for 9 AMS-IX members and 2 Seattle-IX members. For 4 AMS-IX members and 2 Seattle-IX members, both measurements yielded a positive outcome. These ASes deploy RPKI filtering within their own AS and benefit from filtering via route servers.

IXP identification is more challenging for other PoPs without direct route server peerings and all 2+ hops cases. In these cases, we need to rely on the identification of IXPs via TraIXroute. In addition to flagging an AS as ROV enforcing, we also tag the whole traceroute dataset with an IXP tag, if the TraIXroute analysis yielded a positive outcome. This is to show that an inference might be impacted by IXP crossings. Such IXP crossings were identified for two ASes flagged as ROV enforcing. Related traces cross the Digital Realty Internet Exchange.

**Sanitization.** In order to filter out measurement noise and reduce the likelihood of false positives in our measurements, we consider the deviation between the different measurement runs. 89% of ASes are positively identified in three or more measurement runs. The high amount of recurring inferences shows the repeatability of the measurements and the reproducibility of results. A false positive would be introduced if during our RPKI valid ROA configuration all traces complete successfully, and during the RPKI invalid ROA configuration all traces except the experiment prefix would complete successfully. Random route changes or errors in the RIPE Atlas processing might be responsible for such noise. Moreover, data plane packets might simply be dropped by load balancers and other middleware, leading to unresponsive hops. To avoid such issues and make the measurements even more rigorous, we enforce the additional restriction that a ROV inference has to be made in three or more runs before the AS is flagged as ROV-enforcing in our results.

**Validation.** We apply a manual process for validating our findings. ROV inferences were confirmed with data from whois records, public Twitter announcements, PeeringDB information, and operator mailing lists. For our 1 hop cases, out of 174 ASes that support IPv4 and have direct peering sessions with the PEERING testbed, 73 ASes host at least one RIPE Atlas probe. Since we apply the requirement that any flagged AS needs to be positively identified in three independent measurement runs, our methodology flags 10 ASes (9 at Amsterdam and 1 unique additional AS at Seattle) in the 1-hop case as ROV-enforcing. We manually vetted 9 out 10 as true positives. For the other AS we could neither confirm nor deny the hypothesis.

The majority of ROV inferences, 156 (AMS-IX) and 31 (Seattle-IX) ASes in the 1-hop case, stem from route servers, see Table 4.6. Since we observed 8 ASes at both IXPs, we identify 174 unique ASes to benefit from filtering at route servers. Naturally, all of these ASes are assumed to be members of either Seattle-IX or AMS-IX. We checked our assumption and found that it does not hold. 6 ASes for Seattle and 8 ASes for Amsterdam that were identified via route server prefixes are not members of the respective IXPs. We confirm the integrity of the underlying traceroute data and proper working of our mapping methodology and find both to be correct. Therefore, we assume remote peerings via *e.g.,* Multiprotocol Label Switching (MPLS) tunnels to be the underlying cause that hides topology information in traceroute data. With remote peerings, an AS is able to connect to an IXP facility via the company's services without having to be physically present at the IXP. We found a company called IX Reach to offer such services at the IXPs that are included in our measurement data.

We also verify results of our 2+ hops inferences and are able to confirm 6 out of 12 inferences as true positives. We are not able to confirm, nor deny our remaining 6 inferences. 2 out of 12 inferences also carry an IXP tag. A final judgement on whether it is the AS itself, or the IXP inbetween, is currently not possible. Hence, these two inferences could potentially be wrong.

## 4.6 DISCUSSION AND LESSONS LEARNED

**Measurement infrastructure.** We experience outages in prefix propagation with the PEERING testbed. Since continuous propagation of the prefix ranges is a requirement for performing stable experiments, we repeat experiments that are interrupted. Every prefix range that will be used within the PEERING testbed needs to be cleared with the upstreams and thoroughly tested before performing experiments. RIPE Stat [248] helps to understand if a prefix propagates correctly. We experience many propagation errors that are solved by manual investigation and debugging. Every range also needs to have an IRR entry as prefix filters are automatically built from IRR data. The stability of VPN tunnel from the control server towards the PEERING PoPs needs to be closely monitored. We find that VPN connections are sometimes dropped which leads to interruptions of the enclosed communication of Bird-daemons between our control server and the PEERING PoP. As a consequence propagation of prefixes is stopped via the PoP that drops the VPN connection. Therefore, one needs to closely monitor the VPN sessions, see Figure 4.9. If a session is closed unexpectedly, it needs to be immediately reinitiated. The PEERING PoPs Amsterdam and Seattle have by far the most members and therefore richest connectivity. If prefix space is limited, it makes sense to focus on these PoPs for experiments.

Similar to the PEERING testbed, we also experience issued with large scale measurements on the RIPE Atlas platform. Some measurements are simply not executed or mysteriously terminated by the engine. Such problems result in gaps in our measurement data which could only be fixed by extending the respective measurement period and rerunning the experiments. The RIPE Atlas support could unfortunately not determine the cause for such unexpected failures. Our measurement rely on quite a strict schedule as there are many dependencies between the different components (BGP announcements, RPKI ROA schedule, and RIPE Atlas measurements). If a measurement is delayed by the RIPE Atlas engine it might already reach into the next RPKI ROA state. This is very problematic, especially when the problem is not detected. To account for such delays, we recommend using a grace period around the scheduled time of execution. Before using RIPE Atlas data, we also recommend checking that the actual stop date lies within the boundaries set for the particular measurement. For our measurements, we verified manually that measurements were always executed within the desired time frame.

**Middleboxes.** Middleboxes are known to cause problems in data plane measurements [256]. They reply to an ICMP TTL Exceeded message, although they should theoretically act transparently on the network layer. This is done for security reasons to avoid the leakage of potentially confidential information about internal network topologies. To avoid traceroute data that contains replies from middleboxes we have to make sure that traceroutes actually reached the control server within the PEERING testbed. The safest option to achieve that goal is to deploy server-side logging and correlate the traceroute data with logs obtained from the control server. This obviously increases the complexity of measurements to a great extent, but ensures that only proper information is taken into account. Due to technical limitations, we drop around 1% of data plane packets during capture on our control server. The packets that fall within that 1% will not show up in our logs, although those traces have reached the PEERING testbed and data should be used. To avoid such mistakes, we deploy a second check. We also consider the second last hop in all traceroutes, which lies within the PEERING testbed. The second last hop must always contain a PEERING LAN address. The set of PEERING LAN addresses is known to us such that we are able to compare against the IPs in the traceroutes. We are therefore able to confirm whether a traceroute has actually reached our control server or was tampered with by a middlebox.

**IP address to ASN mapping.** We highlight our mapping procedure in Section 4.4. A reliable mapping is quite challenging but essential for the following ROV inference procedures as otherwise false positives are introduced. ASes connect via border routers that speak BGP. The interface that is used for the BGP communication has an IP address assigned. These IP addresses are often shared between peering partners, *e.g.*, $AS_1$ lends $AS_2$ an IP address from its peering LAN for the BGP session. Both IP addresses are from the same subnet. This makes it very hard to perform proper attribution as both routers would be mapped to the same ASN. We confirm the problems outlined above in some of our traces. By manually looking up the reverse DNS entries we are able to determine the correct AS. To minimize the effect of wrong

IP to ASN mapping, we recommend the use of multiple services consecutively, see Section 4.4.

**ROA schedule.** Reuter *et al.* [165] used an 8 hours ROA schedule in 2018, which we originally planned to use as well. However, during trial runs of our measurement campaign, we noticed inconsistent router behavior which triggered a manual investigation. We assumed in the beginning that the RP software deployed in the various ASes fetches ROAs and updates its cache and therefore also BGP routers within a short period of time. That assumption did not hold. We experience quite long delays between publication and actual use of our ROAs. Therefore, some ASes make routing decisions based on stale ROAs which falsifies our results. To allow ASes more time to fetch, digest, and distribute new ROA information, we extend our ROA schedule. We repeat our tests with a very conservative 24 hour window and not further encounter the previously described delays. Our observation leads to a discussion within the IETF SIDR Operations (SIDROPS) working group. As a result, a draft is published with the goal to explicitly state timing parameters for implementers [13]. In the following years, two research items have been published that explicitly deal with our findings. Kristoff *et al.* [180] measure RP delays while Hlavacek *et al.* [172] focus on the different delays from ROA publication until BGP routing decision are impacted.

**Limitations.** Our measurement methodology has several limitations that we would like to highlight, such that future research is able to mitigate some of the shortcomings: (*i*) We explained the need for proper IP to ASN mapping. If IP addresses are mapped to the wrong ASNs, simply because the third-party service provides wrong inputs, the results will contain wrong attributions. (*ii*) Induced path changes are a problem for all measurements that perform active BGP experiments [257]. We could not use the algorithm in [257] since it requires full visibility on all surrounding ASes. We require visibility along the BGP path, *e.g.,* every AS along the path must be a vantage point, but we do not require full knowledge about each and every adjacent AS to the ASes specified in the BGP path. This requirement makes the proposed algorithm practically unusable. It is only possible to fulfil such a requirement in a simulation or limited mock scenario. Moreover, the approach in [257] assumes that only a single root cause is responsible for the change in routing behavior. Again, a very unrealistic scenario considering that BGP had on average 300k–900k BGP updates per day in 2021. Lastly, a change of the ROA state might not only trigger ROV filtering in a single AS, but possibly in adjacent ASes without vantage points that in turn send updates to other ASes. (*iii*) We use TraIXroute to identify IXPs on traceroute paths. If TraIXroute does not detect an IXP within a traceroute, an AS under test might possibly be flagged as ROV enforcing, while the IXP actually performed filtering and the AS under test received *collateral benefit.* (*iv*) We use an include list to differentiate between partially and fully filtering ASes. Each AS that has been seen forwarding RPKI invalid BGP routes is added to the include list. If the same AS is observed filtering based on RPKI invalid ROAs, it is flagged as partially filtering. Due to the limited set of vantage points, we might not be able to add an AS to the include list The differentiation between fully and partially filtering ASes might therefore be biased towards fully filtering ASes. Just because we could not observe peering sessions where filtering for a particular AS was disabled does not mean that such links do not exist. Positive evidence of RPKI filtering can therefore not be generalized for all peering links of an AS.

## 4.7 RPKI VALIDATORS

The previous sections highlighted that RP software, also called RPKI validators, do not fetch, process, and distribute RPKI information to BGP routers within the expected timing intervals. We are therefore interested to know which validator solutions exist and how they are different from each other. Simply put, we would like to know which validator should an operator prefer? How reliable, easy to deploy and use, and resource intensive are they? Most importantly, we would expect all validators to fetch all ROAs from all available publication points and provide the same set of VRP as output. We test these hypotheses and find deviations.

In parallel to the rise of RPKI validator development around 2018/2019, we saw in Section 3.2.2 a steady increase in the amount of VRPs processed by RPKI validators, see Figure 3.2. We updated the Figure in 2021 and confirmed a steady and steep increase, see Figure 4.11. The RPKI is picking up and more and more operators need to make a decision as to which validator they should pick within their infrastructure.

There are seven validator implementations available. All of them are released under an open source license, *i.e.,* Berkeley Software Distribution (BSD), Massachusetts Institute of Technology (MIT), or Internet Systems Consortium (ISC), and hosted on Github. We illustrate the timeline of validator development in Figure 4.12. The very first implementation was sponsored by RIPE NCC in 2011. They developed the RPKI Validator. Another implementation called RPSTIR followed in 2015. Since RPKI gained more traction around 2018, multiple implementations followed in the coming years. Once multiple alternatives were present in the market, it was decided

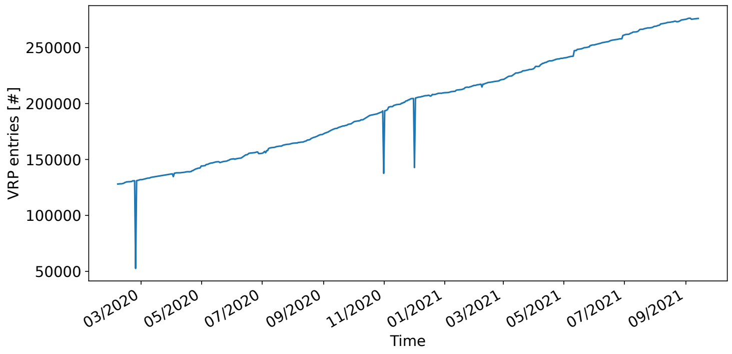


Figure 4.11: VRP entries from March 2020–September 2021. We observe a steady increase of VRP entries. The drops relate to outages in the validator software producing the underlying data [258].

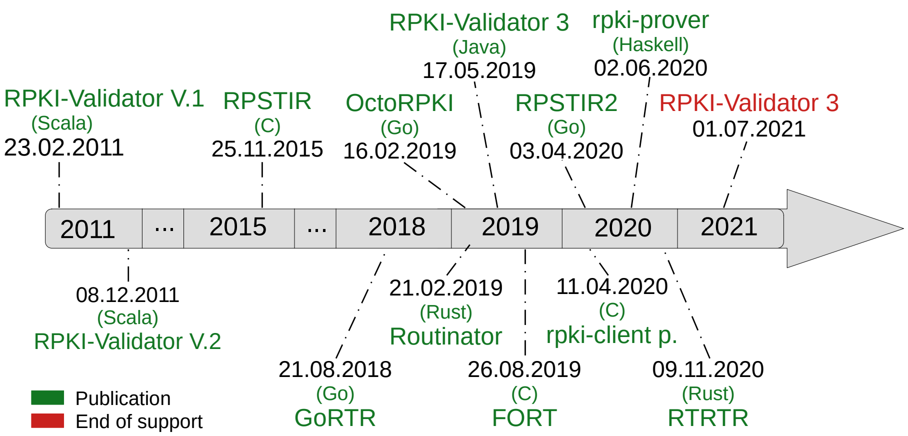


Figure 4.12: Timeline showing major milestones of validators and separate RTR servers.

by RIPE NCC to discontinue with the development of the RPKI Validator in 2021. We highlight the available alternatives throughout the following paragraphs.

**RPSTIR2.** BBN Technologies developed the first version of RPSTIR [259] in 2015. RPSTIR is implemented in C. Around 2017, development and maintenance was continued by ZDNS instead as evident by the commit history of the project. In 2020, RPSTIR2 [84] was released which is written in the Go language. It implements the majority of RPKI RFCs. The validator does not ship in releases, but has to be compiled from the development repository by hand. In addition to RPSTIR2, RPKIVIZ [94] was developed as a frontend to dissect RPKI contents more easily via a browser interface.

**OctoRPKI.** In 2019, an additional RPKI validator was developed by Cloudflare. OctoRPKI [85] is also written in the Go language. In order to move the VRP from the RPKI to the BGP router, the authors also developed GoRTR [260]. It implements RFC 6810, and later RFC 8210 [97]. GoRTR is known to be productively used by Cloudflare, Telia, NTT, GTT, and Cogent [260].

**Routinator 3000.** Routinator [261] was developed by NLnet Labs in 2019. It is implemented in Rust and supports a standalone version of the user interface in version 0.10.1. An RTR server is integrated. In addition to the integrated RTR server, NLnet Labs provides a standalone version to serve as a proxy for larger networks [101]. NLnet Labs does not only develop Routinator as a RP software but also develops an RPKI CA software called Krill [262]. It allows operators to maintain their resources and create ROA, delegations, and the like in an easy to use interface. Routinator is supported via a Discord [263] channel and a public mailing list [264]. Extensive documentation makes installation and maintenance easy [265]. Updates are shared via Twitter [266].

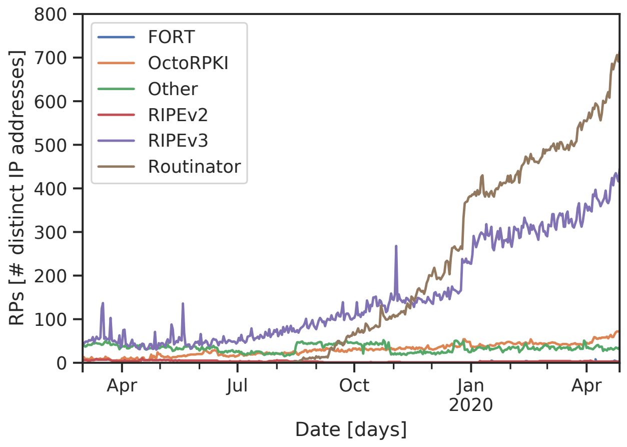


Figure 4.13: Distribution of RPKI validator software. Kristoff *et al.* [180] set up a publication point and recorded incoming pull requests from RP software in 2020. Routinator is most frequently used.

**FORT-Validator.** In 2019, the FORT project [267] RPKI validator was released. It is written in C and was developed by a secure routing initiative from the RIR Latin America and Caribbean Network Information Centre (LACNIC) and NIC.MX from Mexico. Unfortunately, due to a resource shortage, it will only receive critical updates until the end of 2023 [87].

**rpki-client.** OpenBSD project developed and maintains an RPKI validator called rpki-client [268]. The first release was shared in 2019 for OpenBSD, but became usable on other platforms as well in 2020.

**rpki-prover.** Mikhail Puzanov, the main developer of RIPE NCC's RPKI Validator, released rpki-prover [89] as a side project in 2020. It is written in Haskell and is the only RPKI validator that is developed by an individual without an affiliation to a larger organization.

In 2020, Kristoff *et al.* [180] created a publication point for measuring timing parameters within the RPKI. Since every RPKI validator needs to obtain ROAs from each and every publication point, they were able to log all connections of RPKI validator software and created Figure 4.13 from it. We are clearly able to see that Routinator is preferred by operators, followed by the RPKI Validator V3, which was discontinued one year later, and OctoRPKI on third place. All other RPKI validators only obtain insignificant shares.

### 4.7.1 Experiment setup

To compare the performance of seven validators, we aim at parallel execution within our measurement testbed. We could use a virtualized testbed with Kernel-based Virtual Machine (KVM) as a hypervisor, and run multiple virtual machines in parallel. In this scenario we cannot control the scheduling mechanism of the hypervisor and are not entirely sure whether a certain Virtual Machine (VM) would receive more resources compared to another. Our results would therefore eventually be biased. As an alternative, we opted for execution of our measurements on dedicated hardware that provides the exact same environment for each RPKI validator. To this end, we deploy seven Raspberry Pis 4B with 4 GB of Random-Access Memory (RAM). All of the Raspberries are connected to a Cisco Catalyst 2960-S switch with an uplink capacity of 90 Mbit/s. Since some validators require a 64bit OS, we install a 64bit base operating system on a master device with Raspberry Pi OS lite from August 20, 2020 the former Raspbian [269]. The image could easily be copied to all slaves via mirroring the Secure Digital (SD) cards. Since SD cards are usually much slower compared to Hard Disk Drive (HDD)/Solid State Disc (SSD) technology, we fitted all Raspberries with enterprise SD cards. Each SD card has 160 MB/s read and 60 MB/S write speed. Afterwards, the different RPKI validators are installed manually on each device. The versions can be found in Table 4.7. In addition to our Raspberry Pi testbed, we use a server equipped with 112 cores and 700 GB of RAM for long-term tests. The version for validators included in our tests are displayed in Table 4.7.

**Evaluation criteria.** We perform a comparison of RPKI validators using several criteria, see Table 4.8. Most of them are self-explanatory, *e.g.*, quality of documentation, ease of installation, functionality, but some others require a more detailed explanation. Moreover, we try to limit the subjectiveness as much as possible, but a certain amount of subjectivity remains in such evaluations. Each criterion is ranked on a score from one to five. Furthermore, each criterion is weighted from one to three

Table 4.7: RPKI validator versions

| Name | Version |
|---|---:|
| RPKI Validator | 3.2-2021.04.07.12.55 |
| OctoRPKI | 1.2.2 |
| Routinator | 0.8.3 |
| FORT-Validator | 1.5.0 |
| rpki-client | 7.1p |
| RPKISTIR2* | June 2021 |
| rpki-prover* | June 2021 |

* Versioning was not available at this point in time.
We instead state the date when the code was checked out for tests.

according to its overall importance. Hence, the higher the final score, the better the validator performs in comparison to the other candidates.

A single run of a validator includes all necessary parts of the validation process, *i.e.,* fetching ROAs, cryptographically validating all ROAs, and generating the VRPs. The validators need different lengths of time to complete the process. In order to make Central Processing Unit (CPU) utilization throughout the whole process comparable, we have to normalize the CPU utilization. We do this by applying Equation 4.1. $CPU_{norm}$ yields the CPU utilization for each validator normalized to 15 minutes runtime.

$$CPU_{norm} = \frac{CPU_{average} * duration}{15min} \tag{4.1}$$

We expect all validators to provide the same VRP output as it should not depend on the chosen solution which VRPs are generated. There is no ground-truth available as validator output varies for each point in time and no single validator can be considered absolutely reliable. We therefore consider the majority vote of RPKI validator implementations as correct and discuss outliers in our results section. If a validator is found to deviate from the majority vote, the AS using that validator would have a different view of the BGP ecosystem as it might filter more or fewer routes compared to other ASes. We call this criterion *Validator deviation.*

Each validator builds a cache and only fetches the delta to its previously acquired cache for a new validation run. This feature saves a significant amount of load on RPKI publication points and eliminates the burden of transferring redundant information. Moreover, the validation on the client side is much quicker. Therefore, one more thing we test each validator for is differences in results when a validator starts with an empty cache or with a full cache. We call this criterion *Cache vs. Fetch.* However, one should bear in mind that during actual operation of an RPKI validator, an empty cache will only be present during the very first time the validator is used. From then, it will always use the cached information.

### 4.7.2 Comparison

With an overall score of 250/270, we find Routinator to perform best. On the other side of the spectrum ranks RPSTIR2 with only 104/270, the lowest amongst all candidates. There is also a significant gap between RPSTIR2 and the penultimate candidate, implying quite large differences in quality. We summarize our findings in Table 4.8. For each validator, we discuss the individual results in the following paragraphs starting from the bottom to the top:

**RPSTIR2.** This validator marks the lower end of the comparison. It ranks at place seven with 104/270 points. The installation process of RPSTIR2 is quite cumbersome and cannot be readily deployed. It requires manual compilation for each end system and relies on a specific OpenSSL version that has to be built from source as well.

Table 4.8: Comparison of all validators. The highest scores achieved in each case are highlighted. Overall, Routinator ranks the highest.

| | **Weight** | **RPKI-Val.** | **RPSTIR2** | **OctoRPKI** | **Routinator** | **FORT-Val.** | **rpki-client** | **rpki-prover** |
|---|---|---|---|---|---|---|---|---|
| **Installation** | | | | | | | | |
| **Quality of documentation** | 3 | 4 | 2 | 4 | 5 | 4 | 2 | 4 |
| **Installation possibilities** | 2 | 4 | 2 | 5 | 5 | 5 | 4 | 3 |
| **Installation steps quantity** | 2 | 5 | 1 | 4 | 4 | 5 | 4 | 5 |
| **Installation duration** | 1 | 3 | 2 | 5 | 5 | 5 | 3 | 5 |
| **Dependencies** | 1 | 3 | 1 | 4 | 5 | 4 | 2 | 5 |
| **ARM-Installation** | 2 | 5 | 1 | 4 | 5 | 5 | 5 | 1 |
| **Intermediate score** | 55 | 46 | 17 | 47 | **53** | 51 | 37 | 40 |
| **Performance** | | | | | | | | |
| **Validation runtime** | 3 | 4 | 1 | 2 | 5 | 4 | 4 | 5 |
| **CPU utilization/time** | 3 | 2 | 1 | 5 | 5 | 3 | 3 | 5 |
| **Network utilization** | 3 | 5 | 3 | 5 | 5 | 1 | 3 | 5 |
| **Max. RAM consumption** | 3 | 3 | 1 | 2 | 4 | 5 | 5 | 4 |
| **Average RAM consumption** | 2 | 3 | 2 | 3 | 3 | 5 | 5 | 3 |
| **Max. system memory increase** | 1 | 3 | 1 | 5 | 4 | 3 | 5 | 4 |
| **System drive stress** | 1 | 5 | 1 | 5 | 2 | 3 | 3 | 5 |
| **Intermediate score** | 80 | 56 | 24 | 58 | 69 | 55 | 63 | **72** |
| **Validation Results** | | | | | | | | |
| **Validator deviation** | 3 | 5 | 1 | 5 | 5 | 5 | 5 | 3 |
| **Cache vs. Fetch** | 2 | 5 | 2 | 5 | 3 | 5 | 4 | 5 |
| **Intermediate score** | 25 | **25** | 7 | **25** | 21 | **25** | 23 | 19 |
| **Code** | | | | | | | | |
| **Update freq. + code changes** | 1 | 4 | 2 | 3 | 5 | 4 | 5 | 4 |
| **Support** | 1 | 3 | 2 | 3 | 5 | 4 | 5 | 5 |
| **LoC (Complexity)** | 1 | 2 | 3 | 5 | 4 | 2 | 3 | 5 |
| **Intermediate score** | 15 | 9 | 7 | 11 | **14** | 10 | 13 | **14** |
| **Applicability** | | | | | | | | |
| **Update effort** | 3 | 3 | 5 | 4 | 5 | 4 | 4 | 5 |
| **Configurability** | 2 | 3 | 2 | 3 | 5 | 5 | 2 | 2 |
| **Provision of summaries** | 1 | 5 | 2 | 3 | 5 | 1 | 3 | 4 |
| **Intermediate score** | 30 | 20 | 21 | 21 | **30** | 23 | 19 | 23 |
| **Functionality** | | | | | | | | |
| **MAN-Page and help** | 3 | 1 | 2 | 3 | 5 | 5 | 4 | 3 |
| **SLURM support** | 2 | 5 | 2 | 5 | 5 | 5 | 1 | 1 |
| **Logging capabilities** | 2 | 3 | 2 | 3 | 5 | 5 | 3 | 2 |
| **User interface** | 2 | 5 | 1 | 1 | 4 | 1 | 1 | 3 |
| **Single and server execution** | 2 | 2 | 2 | 5 | 5 | 5 | 1 | 1 |
| **RTR server integration** | 2 | 4 | 4 | 4 | 5 | 5 | 1 | 5 |
| **Intermediate score** | 65 | 41 | 28 | 45 | **63** | 57 | 26 | 33 |
| **Overall ranking** | 270 | 197 | 104 | 207 | **250** | 221 | 181 | 201 |
| **Rank** | | 5 | 7 | 3 | **1** | 2 | 6 | 4 |

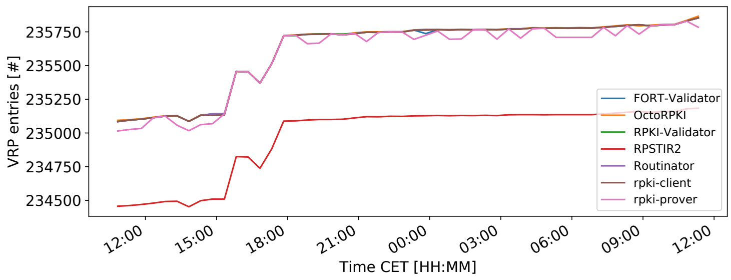


Figure 4.14: Comparison of VRP results of all validators in 24 hours. We observe that most validators obtain the same results while RPSTIR2 reports roughly 600 entries less.

Building RPSTIR2 is not straightforward, as we identify several dependency problems. After contacting the developer, we are able to resolve the issues. Moreover, MySQL 8 is required before the validator functions properly. Overall, the configuration of RPSTIR2 is not well documented and many options are not clearly defined.

During our VRP tests, we find RPSTIR2 to deviate on average about 1,000 VRPs from to the majority vote of all other validator implementations. Figure 4.14 illustrates our finding. The figure shows the absolute amount of VRP entries in comparison to other validators. Due to some additional VRPs and other missing VRPs, the overall difference is shown as 600 VRPs. The trend itself seems to follow the overall set of results, but the deviation required manual investigation. It turns out that the implementation of RPSTIR2 is built in such a way that the same protocol (RRDP or rsync) is set for all descending publication points of one RIR. If a RIR has RRDP defined, the validator will not perform a fallback to rsync if a publication point further down in the chain does not support RRDP. Such behavior explains the missing VRPs, since the ROAs from such a publication point will not be fetched and validated. We consider this choice of implementation a major drawback. The protocol should be chosen per publication point instead. Moreover, the software produces different results when run from scratch with an empty cache and during regular operation with a full cache.

Another thing to consider is the performance of the validator. A single validation takes between 25 and 90 minutes on a Raspberry Pi. The inefficient use of a relational database that is used to manage all RPKI objects might be the underlying issue. We confirm the finding by running RPSTIR2 on a powerful server. As expected it was faster but in comparison to other validators the results remained the same.

**rpki-client.** The rpki-client validator achieves a score of 181/270 and is therefore placed at rank 6. The OpenBSD project had the intention of building a reliable and easy to use validator, which they achieved. It relies on an external RTR server for operation and offers reliable results with a comparable performance. Optional features are mostly missing, this product concentrates on the essentials.

**RPKI Validator.** The RPKI Validator by RIPE NCC was discontinued in 2021.

However, we include the validator in our evaluation as it is still the second most used validator in the field. It ranks in 5th place with a score of 197/270. The validator is provided as a Docker container and ships with a RTR server. Several precompiled binaries are available. Documentation is sufficient and the installation process is easy enough to follow.

**rpki-prover.** The validator ranks on 4th place with a score of 201/270. The installation is easy, binaries for several platforms are provided. Also the performance of the validator is good. Unfortunately, we observe some deviations in the VRP results. Similar to RPSTIR2 they stem from insufficient fall backs for certain publication points and in some instances, unstable RRDP connections. However, the impact is much smaller compared to RPSTIR2. Figure 4.14 shows the small deviations from the majority vote of the other RPKI validators. The problem has been reported and a bug fix was included in June 2021 [270]. Several features are planned to be added to the validator in the future, which are missing at the time of our evaluation.

**OctoRPKI.** The validator is placed 3rd and scores 207/270. Installation is easy and comfortable, as an install from Debian 11 stable repository is available. Alternatively, a docker image is also provided. In December 2020 the project lost its main developer from Cloudflare [271]. At the time of our evaluation it was unclear where the project was headed but it appears Cloudflare could find a replacement, as development has continued in 2023. The VRP output is stable and comparable with the majority vote of other validators.

**FORT-Validator.** Our results show that the FORT-Validator is ranked 2nd with a total of 221/270. Installation is easy and the candidate is included in Debian repositories. In addition, other options are provided for installation. Unfortunately, there is no web user interface. The validator has to be configured from within a terminal. What is surprising is the high network utilization. While Routinator consumed 376 MiB and rpki-client consumed 1,043 MiB under the same conditions, FORT validator managed to transfer 5968 MiB. Extensive documentation is provided but the project has been limited to critical bug fixes since 2021. That condition has not changed in 2023 [87]. It is therefore questionable if the validator receives sufficient feature support that justifies new deployments.

**Routinator 3000.** The first place in our comparison is obtained by Routinator with a score of 250/270. NLnet Labs provides continuous support and implements new features on a frequent basis. A bug report in version 0.9.0 [272] was fixed immediately and the developers communicated the fix to the community shortly after [273]. The installation is easy and the documentation is well prepared. In addition, the performance of Routinator is excellent. Even on our Raspberry Pi evaluation boards it consumes very few resources and only takes between three to six minutes for the initial validation. During 2023 Routinator announced support for ASPA, a new algorithm for path plausibility. This demonstrates that the development of Routinator is at the forefront in BGP security. We therefore recommend deploying Routinator within a production environment.

## 4.8 CONCLUDING REMARKS

In this chapter, we dealt with RPKI ROV identification methodologies and RPKI validator software.

Firstly, we showed limitations of state-of-the-art ROV inference methodologies. We proposed control plane extensions to remove the *connected assumption* and relax the *visibility assumption*. Our extensions improve the amount of ASes that judgements can be made for by 474%.

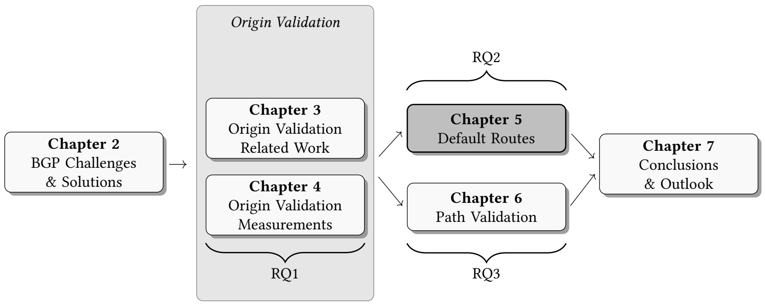
Secondly, we showed how end-to-end data plane methodologies fail to correctly attribute ROV to single ASes. ASes under test receive *collateral benefit* from upstream ASes that perform RPKI filtering. Since end-to-end measurement methodologies via *e.g.,* HTTP only provide a boolean outcome, either a packet arrived or not, a detailed analysis is impossible.

Thirdly, to improve upon the current state-of-the-art, we develop our own ROV inference methodology based on data plane measurements. We use controlled experiments and announce two BGP prefixes on the control plane via the PEERING testbed, an anchor prefix and an experiment prefix. Moreover, we establish our own child-CA and publish ROAs according to a predefined schedule on the management plane. On the data plane, we use RIPE Atlas to send traceroutes towards our prefixes during the valid and the invalid time periods. We translate the obtained IPs into AS paths by using Team Cymru and PyASN. Additionally, we detect IXP crossings in traceroute data using TraIXroute. Finally, we classify the obtained results into six cases. The first three respect the *connected assumption* of previous research, the other three relax the *connected assumption* and allow judgements on longer AS paths. In order to avoid false positives, we enforce very strict requirements on the use of paths of a length greater than two. Our validation shows that the proposed methodology is capable of identifying ROV-filtering ASes in the wild with a very high accuracy. Moreover, the methodology is able to pinpoint ASes that perform the filtering and contributes to the current state-of-the-art in improving the attribution of ROV filtering to single ASes. In addition, by using an include list we are able to tell whether an AS is fully or partially enforcing for RPKI ROV. In our measurements from 5,537 vantage points in 3,694 ASes, we infer that ROV is deployed in 206 unique ASes: 10 with strong confidence, 12 with weak confidence, and 184 indirectly adopting ROV via filtering by IXP route servers, of which 146 are fully and 60 are partially filtering. The measurements were conducted in February 2020, adoption is likely to be much higher now.

Fourthly, we take a look at all available RPKI validators and perform a comparison. We find significant differences in the output of two RPKI validators which would lead to a different view within the AS that uses the validator software. The bugs have been reported to the developers. Our evaluation shows that Routinator performs best. We therefore recommend this validator for new RPKI deployments.

CHAPTER 5

# The Impact of Default Routes on Origin Validation Measurements



*The previous chapter presented our RPKI ROV-inference methodology. During our study, we have seen that RPKI ROV measurements appear to be influenced by the presence of default routes. Throughout this chapter, we identify default routing in ASes in order to quantify the impact on RPKI ROV measurements. To this end, we extend two existing methodologies. Moreover, we provide a new methodology to infer middleboxes in ASes. Our measurement methodology and results have been published as a conference paper [17]. We implement parts of our study as ongoing measurements at* `defaultroutes.net`*. We are pleased that other researchers have already requested access to our data to continue their research. Furthermore, we have been contacted for follow-up work to improve the methodologies. Our measurement code has been publicly released*[a].

[a]https://github.com/nrodday/TAURIN-21

## 5.1 INTRODUCTION

In inter-domain routing, each BGP edge router is responsible for selecting the next best hop for each and every packet that is about to be transferred to the next AS. Outbound policies are controlled by accepting or rejecting prefix announcements from peers, see Section 2.2. Each accepted prefix receives a numerical value as Local-Pref which specifies the priority of the route. BGP will always choose the longest prefix match to forward traffic. If two or more options with the same subnet length are available, the route with the higher LocalPref is selected. This decision process is controlled by BGP policies.

It may happen that an AS has no route in its RIB for a destination to which a packet needs to be routed. That is, because it never received a BGP announcement containing a covering prefix and therefore does not have a full routing table and a complete view of the inter-domain infrastructure, or it decided to reject certain prefix ranges due to local policy. In any case, the AS would not know where the packet is supposed to go to and the client requesting communication with that particular destination would be unable to communicate. To avoid such scenarios, operators can deploy *default routes.* They are used as a last resort when no other possibility has been found in the routing table and attempt to still provide connectivity. An AS would usually define a default route for a certain upstream and rely on the upstream to know where the packet should be further forwarded. Default routes are therefore assumed to be more common in smaller ASes. By definition, ASes participating in the Default Free Zone (DFZ) should not have a default route installed.

Throughout this chapter we aim at developing methodologies to identify the presence of default routes in ASes. We assume default routes can influence our RPKI ROV identification measurements. Chapter 4 highlighted our RPKI ROV identification methodology. At its core, it relies on a prefix pair consisting of an anchor and an experiment prefix. When the ROA configuration is swapped to render the experiment prefix invalid and the route towards that prefix becomes unreachable, we infer ROV in the AS under test. The methodology therefore assumes that non-connectivity during the invalid phase stems from filtering. If connectivity remains, we assume that RPKI ROV is not deployed. But what happens if the AS under test deploys a default route in addition to RPKI ROV filtering? The RPKI invalid route would be filtered and is not present in the local RIB but the data plane packets remain to get forwarded to the upstream to which the default route points. Our methodology therefore infers non ROV, whereas the AS is actually performing RPKI filtering. The presence of default routes jeopardizes the security introduced by deploying RPKI and introduces false negatives into our ROV measurements.

To measure default routes, we extend two methodologies introduced in prior work. (*i*) Bush *et al.* [50] introduced the *Path-poisoning* methodology. And (*ii*) Hlavacek *et al.* [166] introduced the *not-announced prefix* methodology.

## 5.2 METHODOLOGIES AND MEASUREMENT SETUP

The first methodology is called *path-poisoning* and was proposed by Bush *et al.* [50] in 2009. It relies on a poisoned BGP path attribute to avoid installing a route into the routing table of an AS under test. The second methodology is called *not-announced* prefix methodology and simply does not announce a prefix range while performing data plane tests. Both methodologies have in common that no covering prefix must be present. A covering prefix would provide reachability while the experiment prefix is not installed in the RIB itself. If a data plane packet is sent towards the experiment prefix and no entry for the experiment prefix could be found, a larger subnet (covering prefix) is used before a default route comes into play. We must therefore ensure, before starting any experiments, that no covering prefix exists for our experiment prefix. Since we only have a limited amount of prefix space for our experiment prefixes, we conduct such tests manually before starting the measurements.

### 5.2.1 Path-poisoning

**Overview.** The path-poisoning methodology has two essential parts. Firstly, we announce an experiment prefix. The experiment prefix carries the AS under test in its BGP path attribute. This creates an artificial link between our AS and the AS under test. The experiment prefix propagates throughout the inter-domain infrastructure and is installed in all ASes except the AS under test. The AS under test will not install the experiment prefix because of BGP loop prevention. The mechanism is designed to detect routing loops and avoids accepting routes that already carry their own AS number in it. Since every AS except the AS under test has installed the route, only the AS under test would *not* be able to reach IP addresses contained within it. Secondly, we use source address spoofing to send data plane packets towards an IP address within the AS under test which carries the experiment prefix address as a source address. We control the sending server and record replies. To establish communication, the AS under test needs to reply to an IP address from within the experiment prefix range, which it has not installed. If, despite this, receive a reply to our data plane packet, we are able to conclude that the AS under test must have reached our IP address via an alternative path. That alternative path could be a covering prefix (which we made sure does not exist before running the experiment) or a default route as a gateway of last resort. Since all other ASes, including the upstream, have the experiment prefix installed, they would forward the reply packet just like any other packet towards its destination. As a result, we are able to identify the presence of a default route in the AS under test.

A major advantage of this methodology is that it can be used to identify default routes in any AS participating in the inter-domain infrastructure as long as they announce at least one prefix range that contains at least one active host that replies to incoming data plane packets, such as ICMP Time To Live (TTL) exceeded messages.

**Extension.** During the measurements that were performed in 2009, Bush *et al.* [50] had a /16 subnet at their disposal. Split into /24 networks, they were able to perform

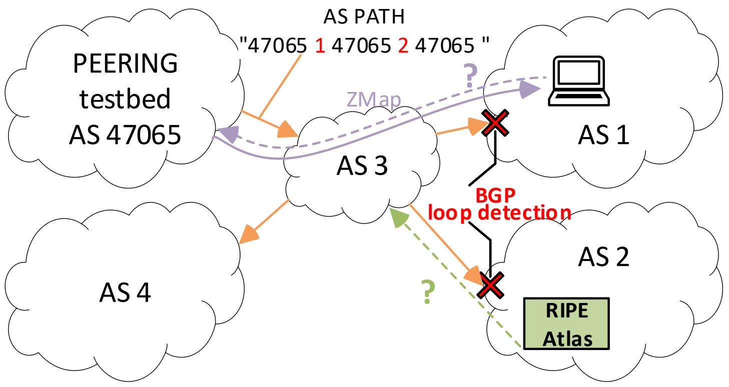


Figure 5.1: Path-poisoning methodology. *AS*1 and *AS*2 are included in the AS path and trigger BGP loop prevention in their own AS. As a result, *AS*1 and *AS*2 do not have connectivity without a default route to the PEERING testbed, while *AS*3 and *AS*4 do.

256 experiments at the same time. Hence, efficiency in the use of prefixes was less important. For our experiments, we are able to use four /24 networks in IPv4 and four /48 networks in IPv6. A single experiment testing only one AS for a default route, takes about three hours. Testing all 72,004 ASes present in the CAIDA AS relationship dataset [54] adds up to six years of continuous testing without interruptions. Such a long time frame does not make any sense as configurations within ASes change over time. In order to speed up the process of default route identification, we add two instead of one AS into the poisoned path, cutting the overall time by half. To avoid artificial links between two different ASes that do not exist in reality, we insert our own PEERING testbed AS inbetween. Only two ASes can be added with the PEERING testbed at a time. Further timing improvements are to be expected when more ASes can be tested simultaneously.

The improvement in efficiency comes with two problems: Firstly, if two ASes happen to be related we might poison a leaf AS as well as the upstream. If, for example, the upstream does not have a default route installed but the leaf AS deploys a default route, data plane packets would still not arrive at our control server. That is because the data plane packet arrives at the upstream, but the upstream has no route towards the destination and would discard the packet. It is therefore important to only poison two ASes at the same time that have no peering relation. We use the CAIDA AS relationship dataset [54] to ensure that the two ASes we pick satisfy this criterion. Secondly, filtering of announcements containing poisoned ASes is twice as likely to happen, according to [274]. We could not confirm this finding as we did not observe any negative side effects in our measurements.

**Setup.** Similar to our previous RPKI measurements, we use the PEERING testbed [152] to announce the experiment prefixes. We maintain a control server which runs the PEERING testbed client software. Our control server connects via Open

Virtual Private Network (OpenVPN) to the Amsterdam PEERING PoP. We create IRR objects for all our prefixes and clear the prefixes with the upstreams at our PoP.

The overall architecture is shown in Figure 5.1. We announce an IPv4 and IPv6 experiment prefix for each AS under test. Some ASes might deploy a default route only for IPv4 and not for IPv6, or the other way round. The announcement is poisoned with two ASes under test in the BGP AS path attribute. $AS3$ receives the update, installs the announced network into its RIB and forwards the route to its peers. $AS4$ receives the route from $AS3$ and also installs it. Upon reception of the update, BGP loop prevention is triggered within $AS1$ and $AS2$. Their ASNs are included in the path and the route is therefore rejected. Once the topology converges, $AS1$ and $AS2$ do not have connectivity towards the experiment prefix range, while $AS3$ and $AS4$ do have connectivity. $AS1$ and $AS2$ were only able to reach the PEERING testbed if default routes were deployed. The final methodology has eight steps:

**1) Announce prefix, wait 20 min.** We announce the experiment prefix via the PEERING testbed. We then wait for 20 minutes for BGP to propagate our new route.

**2) Look-ahead test with ZMap.** To find active hosts within the ASes under test, we query them in order to test for proper reachability. These hosts will be stored in a list as reference points for later use when the announcement is modified with the poisoned ASes in the BGP path. The discovery process has two steps: Firstly, we use IPv4 [275] and IPv6 hitlists [276] to most efficiently identify active hosts within the prefix ranges announced by the ASes under test. E.g. $AS1$ is tested. We look for prefix announcements of $AS1$ with CAIDA's BGP Reader command line utility. Once we find prefix ranges, we search the hitlists for IP addresses within those ranges. If we hit one or multiple addresses, we use ZMap [209], [213] to query these hosts. Positive replies indicate reachability and the host IP is stored in our reference list for $AS1$. If we cannot find any addresses within the hitlists, we perform a full subnet scan with ZMap. This process is repeated until we have gone through all announced address space of $AS1$ until reachable IP addresses are identified. Since the tool is optimized for Internet-wide network surveys, it uses raw-sockets and is therefore capable of sending out many packets in a very short period of time. We make sure not to overload the PEERING testbed or trigger ICMP filtering by limiting the amount of outgoing packets per second.

**3) Look-ahead test with RIPE Atlas.** Bush *et al.* originally only used outgoing data plane packets from a control server towards IP addresses announced by the AS under test. This, however, only allows for a boolean answer. Either an IP address is reachable, or it is not. We add RIPE Atlas [210] into the methodology, where available. The advantage is testing in the opposite direction. We perform a look-ahead test from the RIPE Atlas probe within the AS under test towards the prefix range announced by our control server. Instead of merely receiving a boolean reply, we are able to determine the path the outgoing packet took and are therefore able to identify the upstream to which the packet was forwarded. We used the RIPE Cousteau library [249] for scheduling measurement and parsed the results with RIPE Sagan [277].

**4) Withdraw prefix, wait 90 min.** Once look-ahead tests are completed, we withdraw the experiment prefix and wait for the 90 minutes cool down period. This is to avoid triggering RFD [205] once the prefix is reannounced. If RFD was not considered, a BGP route might not propagate as intended and measurement assumptions would be violated.

**5) Announce poisoned prefix, wait 20 min.** We poison the AS path with the two ASes we plan to test during the experiment run and reannounce the prefix in BGP. Figure 5.1 shows the poisoned BGP path. All ASes except the poisoned ones will install the BGP route in their table. As a result, only the poisoned ASes do not have connectivity if no default route is present. Similar to step one, we wait for 20 minutes for BGP to propagate the route.

**6) Validation test with ZMap.** We use our previously-established list of reference IP addresses that were reachable during step two. Each and every IP is queried again via ZMap and results are stored in our database. On the one hand, if an IP address is not reachable, no route towards our experiment prefix range is present. On the other hand, if a reply is still received by our control server, a default route must be present.

**7) Validation test with RIPE Atlas.** Not only does a validation test with RIPE Atlas allow confirmation of the presence of a default route, but it also allows the identification of the upstream to which the default route pints. We repeat the RIPE Atlas test from step three. If the probe is able to reach our control prefix, we know that a default route is installed and are also able to tell its direction. If data plane packets cannot reach our control server, no default route is installed.

**8) Withdraw prefix, wait 90 min.** We withdraw the experiment prefix. Before reusing the prefix for the next measurement run, we need to make sure that a cool down period of 90 minutes elapses to avoid RFD in the next measurement run.

### 5.2.2 Not-announced Prefix

**Overview.** The methodology was first presented by Hlavacek *et al.* [166] in 2020. At its core, the methodology relies on the fact that an AS will not be able to forward a packet to an upstream, or any other AS of that matter, if there is no route available. Therefore, the methodology uses a reserved subnet that is currently not announced. As explained before, no covering prefix is allowed to exist. The AS forwards a packet to the target defined in the default route only if a default route is present. Hence, we are able to identify default routes by sending data plane packets, in particular traceroutes, from within an AS under test towards a not-announced prefix range and observe whether they are forwarded to the upstream, or they are not. Figure 5.2 shows the experiment setup.

The underlying requirement of this methodology is access to a vantage point that is capable of issuing data plane packets from within the AS under test. This significantly limits the range of the experiments. An advantage of the methodology

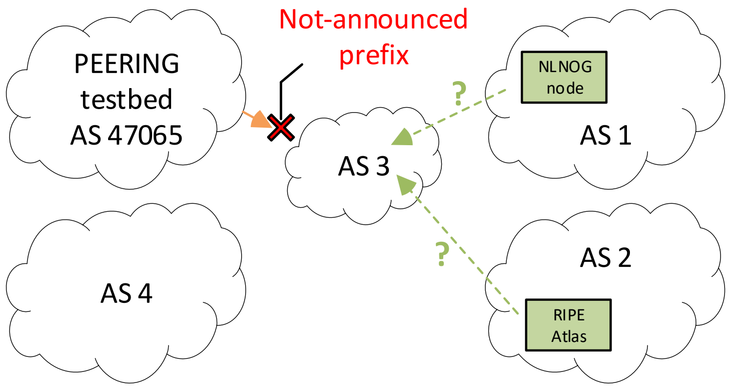


Figure 5.2: Not-announced prefix methodology. *AS*1 and *AS*2 host data plane probes that send traceroutes towards a withdrawn prefix range. *AS*4 does not host a probe and cannot be tested.

is the very fast execution of measurements. It is entirely possible to run all measurements within all ASes that should be tested in parallel, therefore reducing the time until results are obtained to a fraction compared to the *path-poisoning* methodology. A typical run only takes about 30 minutes for all ASes covered by a vantage point. Previous work [166] used RIPE Atlas to perform the experiments, which covers 3,699 ASes. In addition, no prefix ranges are required to run these experiments. Hence, access to a BGP facility like the PEERING testbed is also not necessary, which reduces complexity of measurements tremendously. The experimenter could simply use an assigned but currently unannounced prefix range for their experiments as long as no covering prefix exists.

**Extension.** In prior work, an AS is flagged as having a default route when a single `traceroute` of any vantage point within the AS under test reaches an upstream or peer. Unfortunately, inconsistencies amongst vantage points are not considered. We discover such inconsistencies where multiple vantage points within the same AS deliver different results. Default routes could only be deployed for a subset of the network, *e.g.*, a geographical area or a logical area within the network. Depending on where the vantage point is located, the results will turn out to be positive or negative. Moreover, ICMP packets could be filtered while TCP or UDP packets might still be forwarded via default routes. We developed a threshold-based approach in which the experimenter is able to individually define a threshold that abstracts multiple measurement results to a boolean result for the whole AS.

Moreover, prior work only used RIPE Atlas. We extend the range of the experiments by adding NLnog vantage points, see Section 2.7. Access to NLnog Ring nodes was kindly granted for our measurements by the managing parties. NLnog covers 467 ASes in total and coverage of our experiments is increased by 193 ASes for IPv4 and 234 ASes for IPv6. To execute measurements on all nodes simultaneously, NLnog offers a ring-all command which encapsulates the actual command to be executed. The command, however, was not usable for our purpose as we needed to obtain the output traceroute from each individual session. Instead of relying on the ring-all command we developed a tool that logs into all machines simultaneously and executes the traceroute command. Output is afterwards received and stored in RIPE Atlas format. It is therefore easy to reuse our NLnog tool to expand other measurements that already use RIPE Atlas. We publicly released our tool on Github [278].

Lastly, we perform experiments for IPv4 and IPv6.

**Setup.** Our experiment setup is depicted in Figure 5.2. Since the not-announced prefix methodology does not require any BGP interaction, the experimenter can simply query RIPE Atlas and NLnog vantage points to execute traceroutes towards an unannounced prefix range and evaluate the results.

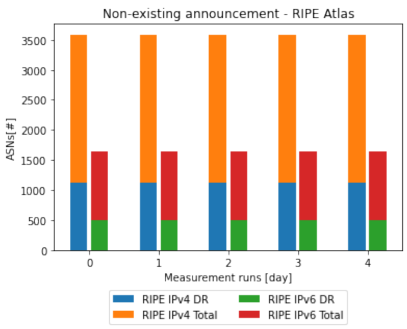
### 5.2.3 Classification of Network Tiers

We need to categorize ASes in order to make judgements about whether default routes are more likely to be deployed at the core or the edge of the Internet. The UCLA dataset [279] was used in previous work by Bush *et al.* [50] but is no longer available. We classify ASes into three tiers according to a commonly accepted definition and create our own dataset: Tier-1 ASes do not need to buy any connectivity from other ASes and achieve global connectivity by themselves. Tier-2 ASes need to buy transit for some parts of the Internet but achieve a high level of connectivity already via (settlement-free) peering arrangements. They also sell connectivity to smaller ASes. Tier-3 ASes are leaf ASes that always require an upstream to connect them to the outside world. They do not have peering relationships.

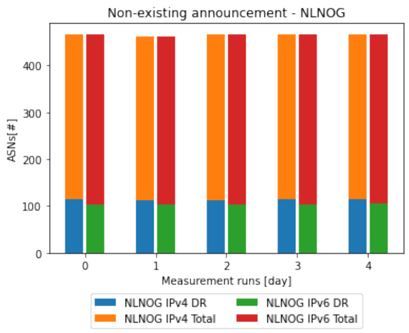
Our methodology relies on AS relationships. Hence, we need to use a dataset that contains relationship information for as many ASes as possible and with high quality information contained within it. There are two datasets available: The CAIDA AS relationship dataset [54] and the ProbLink dataset [280]. Both methodologies [281], [282] create the datasets from publicly available collector data from RIPE RIS [146] and RouteViews [140]. Jin *et al.* [282] claim that their ProbLink dataset is 27% more accurate for inferring complex relationships.

We analyze both datasets to see which best fits our classification. CAIDA's AS relationship dataset contains 72,004 ASes, while the ProbLink dataset contains 44.695 ASes. We contact the authors of the ProbLink dataset to learn more about how their algorithm for inferring complex relationships works and find that ProbLink's inferences are calculated based on a single day of BGP data. CAIDA's AS relationship dataset is calculated based on multiple days. Therefore, CAIDA's AS relationship dataset contains many more ASes and many more relationships between them. Since a large coverage is key, we decided to use CAIDA's AS relationship dataset.

Our process for classifying ASes within the AS relationship dataset works as follows: CAIDA provides an input clique, which contains by definition tier-1 providers. These are manually vetted and we label them as such. In total, we label



(a) Not-announced - RIPE Atlas



(b) Not-announced - NLnog

Figure 5.3: Consistency of not-announced prefix results. Default Routes distribution remains the same for five measurement days. RIPE Atlas has a higher total of probes capable of IPv4 probing than IPv6. NLnog nodes are capable of both IPv4 and IPv6 probing.

19 ASes as tier-1. Next, all ASes that either have a peering relationship with another AS or provide connectivity to another AS are labelled as tier-2. All remaining ASes are labelled as tier-3. When our procedure is finished, we obtain 19 (0.03%) tier-1, 11,325 (15.73%) tier-2, and 60,660 (84.25%) tier-3 ASes.

For comparison, Bush *et al.* [50] classified into large, small and stub networks and used the UCLA dataset. They classify 255 (0.08%) as large ASes, 1,361 (4.11%) as small ASes, and 31,517 (95.12%) as stub ASes. We observe that many more ASes are classified as tier-2 in our approach. A possible reason could be that the Internet is more and more flattening [283], [284]. Such flattening is prevalent since peering relationships between ASes are more common these days. The reason is simple: ASes aim at reducing cost by preferring settlement-free peering relationships compared to purchasing upstream connectivity.

## 5.3 RESULTS

Our findings are summarized in Table 5.1. For the path-poisoning methodology we start our measurements in December 2020. Since this methodology is very time consuming, every AS is only tested once. Both measurement campaigns are implemented as ongoing measurements, but the path-poisoning methodology had to be discontinued in 2022 as the experiment prefixes were needed elsewhere by the owner. The not-announced methodology only requires access to and credits for RIPE Atlas and NLnog. It is therefore much simpler and easier to reproduce. We perform experiments with the not-announced methodology in February 2021 for five consecutive days and find results to be consistent, see Figure 5.3. We compare the results for both methodologies in Table 5.1 with a limited dataset dated to June 2021.

Table 5.1: Default route results from different methodologies.

| | **Not-announced Prefix** | | | | | | **Path-Poisoning** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **RIPE Atlas** | | | **NLnog** | | | **PEERING Testbed** | | |
| AS Tier | # Tested ASes | | Default Routes | # Tested ASes | | Default Routes | # Tested ASes | | Default Routes |
| ***IPv4*** | | | | | | | | | |
| 1 | 11 | (0.30%) | 0% | 3 | (0.64%) | 0% | 15 | (0.91%) | 33.33% |
| 2 | 1909 | (53.30%) | 23.31% | 306 | (65.66%) | 17.97% | 1001 | (61.26%) | 51.45% |
| 3 | 1662 | (46.40%) | 41.14% | 157 | (33.70%) | 39.49% | 618 | (37.83%) | 66.02% |
| *Sum* | 3585 | (100%) | 31.52% | 466 | (100%) | 25.11% | 1634 | (100%) | 56.79% |
| ***IPv6*** | | | | | | | | | |
| 1 | 10 | (0.61%) | 10% | 3 | (0.64%) | 33.33% | 15 | (0.91%) | 53.33% |
| 2 | 974 | (59.46%) | 25.67% | 306 | (65.66%) | 16.99% | 1002 | (61.24%) | 32.83% |
| 3 | 654 | (39.93%) | 39.76% | 157 | (33,70%) | 33.76% | 619 | (37,85%) | 21.49% |
| *Sum* | 1638 | (100%) | 31.2% | 466 | (100%) | 22.75% | 1636 | (100%) | 28.73% |

## 5.3.1 Preliminaries

**Consistency check.** During our evaluation of the *not-announced prefix* methodology we discover inconsistent behavior when multiple probes are residing in a single AS. In AS3320, Deutsche Telekom GmbH, we observe 190 RIPE Atlas probes. While 186 probes are not able to reach any other AS, four probes manage to get forwarded to an upstream/peer and reported its IP addresses. Each of these four probes have connectivity via a different peer. Hlavacek *et al.* [166] do not consider inconsistent behavior within their study. They classify an AS as having a default route if only a single probe is able to reach another peer. This could potentially lead to false positives, as a ratio 4/190 seem to pretty clearly state that no default route is present for the vast majority of probes. But why are four probes capable of reaching an upstream/peer? We assume default routes in parts of the network or specific geographic regions to be responsible. In any case, the experimenter should be able to judge whether they want to classify an AS as having a default route or not. We therefore introduce a threshold value. Depending on the study conducted, the experimenter can choose a different value to render an AS as having a default route once the threshold is surpassed. It describes the minimum number of probes needed to flag an AS relative to the maximum number of probes available in the AS under test.

We illustrate the impact of the introduced threshold value on the overall results in Figure 5.4. Multiple probes may deliver different results. Additionally, as discussed in Chapter 4.6, mapping of IP addresses to ASNs might not always perform perfectly without problems. It might seem that a default route is deployed as the traceroute reports an IP address that belong to another AS. However, the IP address might still be within the same AS and mapping might be wrong, if the identification of the border between two ASes proves to be a challenge [285]. Therefore, no default route is deployed, but it only seems that way. Figure 5.4 shows that the smaller the threshold, the more ASes are identified as having a default route. If only a single probe is sufficient to flag the AS as deploying a default route, it is likely

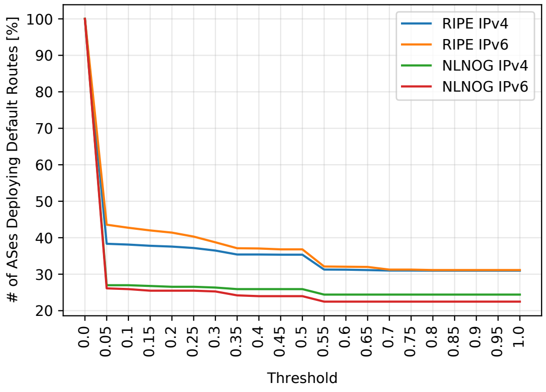


Figure 5.4: The relative ratio of ASes identified as deploying default routes (y-axis), based on a threshold value (x-axis). The more ASes required to provide a positive result, the lower the identification rate. Some ASes only feature two probes with contrary results, hence the decline at 0.5.

that false positives are included. We observe a decline for a threshold value of 0.5. This scenario is important for a draw between the number of probes reporting to reach an upstream and not reaching an upstream. E.g. two probes are deployed, but only one is reporting a default route. Setting the threshold above 0.5 will require an absolute majority of probes to report a default route. For our experiments using the *not-announced prefix* methodology, we use a threshold value of 0.55.

This is contrary to the *path-poisoning* methodology where a single probe is sufficient to flag an AS as deploying a default route.

**Coverage.** RIPE Atlas provides coverage for 11 of 19 tier-1 ASes, 1909 tier-2 ASes and 1662 tier-3 ASes. From a relative perspective, NLnog covers more tier-1, but fewer tier-3 ASes. Since NLnog is a collaboration platform between network operators, we suspect mostly operators who participate have more staff available, hence larger ASes. Many of the RIPE Atlas probes are IPv4 only, while a fraction supports both IPv4 and IPv6. NLnog nodes are required by definition to support IPv6. As a result, we observe the same amount of ASes in IPv4 and IPv6 that can be tested via NLnog vantage points.

**Middlebox identifications.** During the evaluation we find replies to probes that seem to have successfully reached the PEERING testbed even though we did not announce the prefix range in BGP. It should therefore be impossible for a packet to reach the final destination. This behavior triggers a manual investigation and we find middleboxes to be the underlying issue. In order to filter out probes sitting behind middleboxes we set up a measurement campaign.

On the control plane we announce a /24 network via the PEERING testbed. On the data plane we choose 10,105 RIPE Atlas probes that are IPv4 capable and currently connected. Moreover, they also need to have the *public IPv4 address* field filled in their probe's meta data [286] information. This information is crucial as many probes are sitting behind Network Address Translations (NATs) and the public IP of the router and not the private IP address of the probe itself will be seen at our PEERING testbed destination. At the same time, we record all received traffic at the PEERING testbed for the announced /24 network. The idea is to filter probes that claim in their traceroute to have reached the PEERING testbed, while in fact they did not.

Out of 10,105 selected probes, 9,530 actually participated. From these 9,530 participating probes, 9,474 (99.41%) show in their traceroute to have reached the PEERING testbed. Based on the captured traffic at the PEERING testbed, we are able to confirm 9,045 (95.47%) probes have delivered correct measurement data. However, 322 (3.4%) probes could not be found in the recorded traffic sample, while tcpdump reported 0% of dropped packets during capture. We were still able to confirm these probes as they had reached the PEERING testbed upstream router as a second-last hop, which we could identify by the limited set of IP addresses assigned to them. We could not confirm 107 (1.13%) probes. During manual investigation, they all exhibit the same properties: Very short paths, a few private hops, followed by the final destination. We could not observe the PEERING testbed router anywhere. As a result, we find these 107 RIPE Atlas probes to sit behind a middlebox. The middlebox replies to traceroutes (ICMP TTL exceeded messages) with the destination IP address. We suspect the reduction of traffic and hiding of topology information to be the reasons for this behavior. Identification of these RIPE Atlas probes behind middleboxes is not only crucial for our experiments, but is essential for many other measurements as they indicate successful reachability, while in fact they did not get any further than the middlebox itself. We propose to add a flag into the RIPE Atlas meta data information that allows easy elimination of such probes from measurements, *e.g.,* via the API.

### 5.3.2 Comparison of Vantage Points, Methodologies, and Prior Work

**Comparison of RIPE Atlas and NLnog vantage points.** RIPE Atlas probes identify default routes at a higher ratio compared to NLnog probes for IPv4 and IPv6, see Table 5.1. We have access to RIPE Atlas and NLnog vantage points in 273 ASes. In 250 (91.6%) ASes, results are identical, while in 23 (8.4%) ASes results differ. We are able to reproduce these findings for different measurement runs.

**Comparison of not-announced prefix and path-poisoning methodologies.** For IPv4, we are able to compare the results of 601 ASes. 271 (45.11%) ASes show the same results. 162 (26.95%) ASes show different results, albeit the measurements were executed from the same RIPE Atlas probe. We suspect that the time of measurement execution might be a, or at least a partial, reason. The not-announced prefix measurements are performed at the same time while the *path-poisoning* measurements are conducted over several months. Operators possibly changed default route configurations in the mean time. 103 (17.13%) ASes hosted RIPE Atlas probes which send traceroutes that were not able to leave the AS. But they did successfully reply to ICMP echo requests during the *path-poisoning* measurements. 65 (10.81%) ASes remain that show different results. The threshold of the *not-announced prefix* methodology was set to 0.55 which would render an AS as not having a default route if two probes were present but only one managed to obtain upstream connectivity. The *path-poisoning* only requires a single probe to leave the AS, which explains why there is a deviation between the two methodologies.

**Comparison with prior work.** Bush *et al.* [50] report that during the AS path-poisoning, 74.8% of ASes were able to consistently respond to probe packets from the experiment prefix. 20.4% of the ASes could not reply successfully and 4.3% of ASes answered only for a subset of IP addresses, but never all of them. We show in Table 5.1 that in our measurements during the poisoning period, 57.25% of ASes responded successfully. We should, however, bear in mind that the overall number of tier-3 ASes that were tested in the 2009 study was much higher and could therefore potentially weight results differently. They report 17.1% of large ISPs, 44.5% of small ISPs, and 77.1% of stub ASes to deploy default routes.

Hlavacek *et al.* [166] report 768 (46.37%) ASes out of 1,656 tested ASes deploying default routes. In our experiments, we covered all available ASes within RIPE Atlas and found default route deployment with 31.52% to be lower. We suspect three main reasons for this deviation: Firstly, the selection of RIPE Atlas probes might have influenced results. We cover all ASes that are available via RIPE Atlas. If Hlavacek *et al.* tested for a subset of ASes that contained more tier-3 ASes and fewer tier-2 ASes, the relative default route deployment is expected to be higher. Secondly, we introduce the threshold value to reduce false positives (where single RIPE Atlas probes report connectivity while the vast majority of probes does not). Thirdly, prior work does not account for middleboxes. These will introduce false positives as they

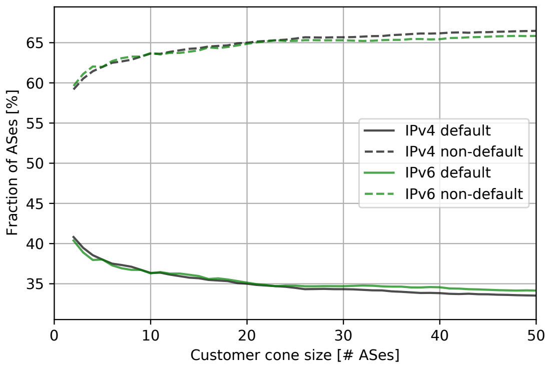


Figure 5.5: Default route presence in relation to costumer cones size. The data was obtained with RIPE Atlas and a threshold at 0.55. We observe that the smaller the AS the more likely it deploys a default route. Results for IPv4 and IPv6 are almost identical.

suggest connectivity while the packet was indeed answered by the middlebox.

Overall, we find default routes to be more often present in smaller ASes compared to larger ASes. The smaller the customer cone of an AS, the higher the likelihood that a default route is deployed, see Figure 5.5.

## 5.4 CONCLUDING REMARKS

During performing our RPKI ROV measurements we discover that default routes could potentially lead to false negatives within our RPKI measurement results. Consequently, we start to extend two existing methodologies, the *not-announced prefix* methodology and the *path-poisoning* methodology. We extend the *not-announced prefix* methodology with NLnog vantage points and run the experiments for IPv4 and IPv6. In addition, we extend the *path-poisoning* methodology with dual poisoning to make measurements more efficient.

Our results show that default routes are present in many more ASes than anticipated. Even ASes within the DFZ, in which NLnog vantage points reside, use default routes as a means to avoid interruption of communication. We see that default routes are much more common in smaller ASes compared to larger ASes. The smaller the customer cone size, the higher the likelihood that a default route is deployed.

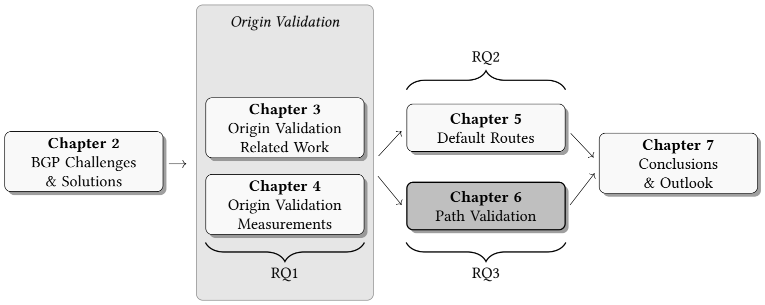
Moreover, we find that middleboxes tamper with traceroute results obtained via data plane vantage points. 107 out of 10,105 RIPE Atlas probes are sitting behind middleboxes. These reply with the destination IP address instead of transparently forwarding the packets to the next router. Such middleboxes lead to falsified results, not only in our default route measurements, but in other data plane experiments as well.

We publicly release the code and datasets of our study [278]. Additionally, we set up a measurement website `https://www.defaultroutes.net` which reports the status of our weekly ongoing experiments. The website allows the adjustment of the threshold value for all obtained measurement data for the not-announced prefix methodology.

The identification of default routes allows us to eliminate such ASes from the overall dataset of ASes to test for RPKI ROV. Our middlebox inferences are also used by other researchers for Route Flap Damping (RFD) measurements.

CHAPTER 6

# Path Validation



*In Chapter 4 we focused on the development of a RPKI ROV measurement methodology and in Chapter 5 we quantified the presence of default routes in ASes. This chapter continues our endeavor by moving from origin validation to path validation and path plausibility algorithms. Origin validation, namely the RPKI, is currently being deployed, but only path validation and path plausibility algorithms allow for protection against path manipulation attacks and route leaks. This chapter uses simulations to compare two new path plausibility algorithm proposals within the IETF. We focus on implementing these two new proposals, namely ASPA and AS-Cones, and propose several improvements. Moreover, we compare both algorithms regarding their protection against route leaks and the forged-origin prefix hijack and evaluate several deployment scenarios. Based on our results, we recommend where these algorithms should first be deployed. Our work has been published as two conference papers [18], [19] and a conference poster [20]. The topology generator framework, found in Appendix A, and our simulation results have been publicly released.*

## 6.1 INTRODUCTION

The previous chapters highlighted several BGP security issues. In particular, BGP exact and more specific prefix hijacks can be mitigated using the RPKI. However, the forged-origin prefix hijack as a path manipulation attack cannot be mitigated by the RPKI. Path validation algorithms or path plausibility algorithms are required to mitigate such path manipulation attacks. The IETF standardized BGPSec [11] in 2017 to provide strong path security, see Section 2.5.2. Unfortunately, it is not expected to be deployed in the near future as BGPsec suffers from several drawbacks, such as incompatibility with partial deployment and additional cryptographic overhead introduced into routers.

Route leaks are another attack vector that we have not yet considered in previous chapters. We introduced route leaks in Section 2.5.3 as policy violations. ASes are usually expected to act according to the Gao-Rexford model [44]. If the model is violated by an AS forwarding a route in an in-compliant manner, financially undesirable situations occur for the leaking AS. Moreover, traffic could potentially be forwarded to a small leaking AS that cannot cope with large volumes of traffic and would render the destination unavailable. These errors occur when prefix filters are mistakenly updated. Neither the RPKI [70] nor BGPsec [11] are designed to solve the route leak issue.

This chapter focuses on two path plausibility algorithms currently discussed within the IETF. ASPA [135], [136] has been discussed for many years and has matured. AS-Cones [139] was proposed in 2020 but did not receive much attention. It is, as a result, also less mature. The fact that both path plausibility algorithms use the existing RPKI infrastructure to publish and distribute their cryptographic objects is expected to accelerate the deployment. ASes are not required to make groundbreaking changes to their infrastructure. Path plausibility algorithms are, therefore, considered more lightweight than path validation algorithms such as BGPSec. Within ASPA and AS-Cones, each object carries relationship information of either the authorized providers, in the case of ASPA, or the customer cone served by the AS, in the case of AS-Cones.

In this chapter we intend to evaluate both proposals regarding their performance for route leak and forged-origin prefix hijack mitigation. To this end, we design and implement a BGP topology generator framework that allows us to generate arbitrary topologies with the NIST BGP-SRx software suite. During our study, we cannot overcome the limit of 55,000 containers within our framework due to technical limitations. Therefore, we use simulations to perform our ASPA and AS-Cones evaluations. Our BGP topology generator framework can be found in Appendix A.

In particular, we focus on different deployment scenarios to make recommendations as to which ASes should be incentivized to deploy path plausibility algorithms first.

## 6.2 METHODOLOGY

Our methodology aims to achieve two goals: Firstly, we study the performance of path plausibility algorithms, namely ASPA and AS-Cones, regarding their ability to mitigate route leaks and forged-origin prefix hijacks. Secondly, we study both algorithms in different deployment scenarios in order to conclude how these algorithms should be deployed to have the highest impact. Both algorithms were implemented in a Python BGP simulation testbed. Cohen et al. [287] used a BGP simulator and published their results but did not open-source their implementation. As a result, Brand and Posen reimplemented the BGP simulator in [288] in order to reproduce the previous study and open-sourced their implementation in [289]. We use their BGP simulator and extend the environment with our route leak and forged-origin prefix hijack scenarios. Meanwhile, BGPy [290] was published, which looks promising but could not be considered for our study.

**Simulation graph.** Our simulations are performed with the CAIDA as_rel2 dataset [141] from October 1, 2022. It represents the inter-domain routing infrastructure, including the ASes and their inter-connectivity. Based on that dataset, we create a directed graph with NetworkX [291], a network analysis library in Python. We define a directed edge to represent a customer–provider relationship. A peering relationship is represented by a directed edge in both directions. The obtained graph has 74,110 nodes that represent the ASes. The labels of the nodes are the respective ASNs. In total, our graph consists of 110 tier-1, 11,237 tier-2, and 62,768 tier-3 ASes. The tier classification strategy is applied from [17]. A tier-1 AS has no provider relationships, a tier-2 AS has both customer and provider relationships, and a tier-3 AS has no customer relationships at all.

**Route leak scenario.** The simulation testbed is designed to simulate prefix hijacking scenarios. Since we evaluate a different attack within the testbed, namely route leaks, we need to adjust the scenario. Route leaks are misconfigurations that randomly take place. Therefore, for each trial, we choose a *victim AS* and a *leaking AS* randomly from the overall set of ASes. ASes selected during one trial may appear in another trial. Each AS in the graph has a *Default Policy* assigned that controls its routing behavior. It represents correct behavior according to the Gao-Rexford model. Hence, no route leaks will occur within the graph with this policy applied. Only the leaking AS is set to *Route Leak Policy*. In contrast to the *Default Policy*, the *Route Leak Policy* does not respect the Gao-Rexford model and propagates announcements regardless of relationship. This behavior creates route leaks. In addition, both policies implement the following BGP preference rules for route selection:

1. Local preference (customer - 1, peer - 2, provider - 3)

2. Route length

3. ASN of first hop as tie-breaker

Table 6.1: Computational time with 250 CPU cores @ 2.4 Ghz

| Trials | Single trial [sec] | Single scenario [days] |
|---:|---:|---:|
| 10 | 5.3 | 0,6 |
| 100 | 12.2 | 1.4 |
| 1,000 | 30.7 | 8.4 |
| 10,000 | 242.0 | 84* |

* Anticipated time.

It depends on the degree of connectivity of the leaking AS how many ASes will be affected by the leak. The larger the leaking AS, the more ASes will fall victim to the leak. Furthermore, if the leaking AS is directly connected to a tier-1 provider, many more ASes will be affected, as the resulting routes will be relatively short. On the other hand, if the leaking AS is only connected to the rest of the infrastructure via a single upstream provider, via which it also receives the same route, the leak will have no effect as the upstream will filter the announcement due to BGP's loop prevention mechanism. We only propagate a single route announcement during each simulation run, starting with the victim as the origin.

To observe how many ASes were affected by the leak, we use the *route leak success rate* as a metric. The input consists of all RIB tables of all ASes in the graph. The graph is used as an oracle to know whether an AS contains the leaked route. More specifically, we look for the leaking AS in the AS path attribute of the installed RIB entry. The *route leak success rate* is calculated by the share of affected ASes vs. unaffected ASes. Before running the experiments, we must establish a baseline against which to compare. These simulation runs do not have any security mechanism in place. We run our measurements with 10, 100, 1,000, and 10,000 trials and execute

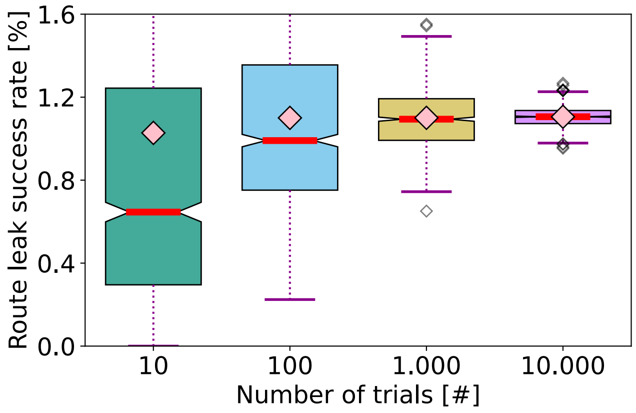


Figure 6.1: *Route leak success rate* for different number of trials.

each trial 1,000 times. The results are shown in Figure 6.1 as the mean *route leak success rate*. We observe a large spread for 10 and 100 trials because these sets are pretty small, and therefore show more impact on outliers, including ASes with very rich or very poor connectivity. The spread is much smaller for the sets of 1,000 and 10,000 trials, and the mean route leak success rate is 1.1%. The higher the number of trials, the less impact outliers have on the overall results. Table 6.1 compares the number of trials versus the required computational power in time. We choose 1,000 trials for our simulations as the computational time and precision are reasonable. A mean route leak success rate of 1.1% might appear to be relatively small at first. However, we should remember that the absolute amount of ASes is 74,110. Therefore, on average, 815 ASes are affected by each route leak. Considering the amount of traffic that this number of ASes would forward on average highlights quite a high impact of every single route leak without any security system in place. To make all simulations comparable, we choose the same set of 1,000 trials for all simulations. Only the share of ASPA and AS-Cones object and policy deployment varies across different simulations.

For simplicity, we do not consider route server peerings in our simulations. Moreover, we do not consider mutual transits, also known as sibling ASes.

**Object creation.** For both algorithms, we must create objects containing the required relationship information. The objects are simple in their design for both algorithms. In ASPA the following object would be created by `AS1` to authorize `AS2` and `AS3` as its providers:

```
ASPA{Origin, [Provider, Provider, ...]}

For example:
ASPA{AS1, [AS2, AS3]}
```

AS-Cones objects are created similarly, but the information within holds the customers of the issuing AS. The following example shows `AS2` publishing an AS-Cones object that declares `AS1` and `AS5` as its customers:

```
AS-Cones{Origin, [Customer, Customer, ...]}

For example:
AS-Cones{AS2, [AS1, AS5]}
```

Therefore, the information within the ASPA and AS-Cones objects is inverted.

The AS-Cones IETF draft specifies an additional policy object that we do not implement and use. Moreover, we do not include additional parameters specified in the IETF draft in our objects as they are not required for its rudimentary workings. Our objects are not cryptographically signed nor validated by RP-software upon retrieval. In fact, we do not use any RP-software in our simulations as the whole cryptographic procedure would only be required in an untrusted environment. We create and control all objects in a safe setting and are therefore confident about their integrity. The NIST BGP-SRx software suite employs similar simplification steps [292].

**Algorithm implementation.** The algorithm implementation of the policy component is more complex than the object creation. Each AS receiving a BGP announcement and installing the respective filtering policy executes the path plausibility algorithm. Both algorithms need access to the complete set of their objects in order to work correctly. For implementing the ASPA algorithm, we follow the exact steps outlined in the IETF draft. The only difference is that we adjust the indices within the algorithm within the simulation environment for technical reasons. This is important to consider once our implementation is used in a production environment instead of the simulation testbed.

The AS-Cones draft is very early-stage and could, therefore, not be used for the verification procedure. Instead, we rely on the ASPA validation algorithm within the AS-Cones setting to perform the validation. The relationship information contained within the AS-Cones objects has to be inverted to make the algorithm work.

We verified our implementations for both algorithms via means of unit tests. The authors of the IETF draft [293] provide valid, invalid, and unknown trajectories that we used within our unit tests.

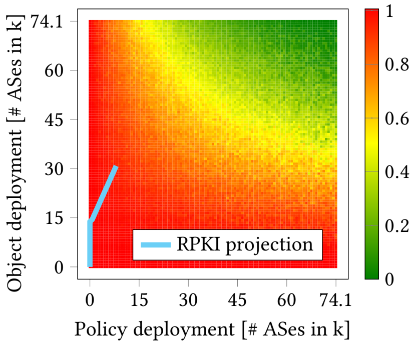
Within our simulation, we only filter invalid routes. Valid and unknown routes are accepted and forwarded by the filtering AS according to the deployed policy component. In general, it is only possible to filter unknown routes once the number of valid routes reaches a very high threshold such that unknown routes are an exception. This only holds for almost global deployment and not for a partial deployment scenario we anticipate at the beginning of the roll-out.

## 6.3 ROUTE LEAK SCENARIO

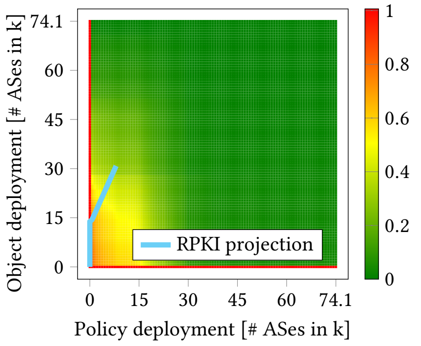
Our evaluation is designed to compare object vs. policy deployment for both ASPA and AS-Cones, with different deployment strategies. Object creation refers to the number of ASes creating and publishing objects that detail their relationship to peers, while policy deployment describes the number of ASes that deploy one of the algorithms to filter invalid routes.

**ASPA.** We present our simulation results for random ASPA object and policy deployment in Figure 6.2a. The *route leak success rate* is represented by the mean value over 1,000 trials on a color-coded scale. Green indicates that only a small number of ASes have been affected by the leak, and the protection of the algorithm worked well, while red indicates a wide spread of the route leak, compared to the baseline we established before. We show the number of ASes that deploy objects on the y-axis and the number of ASes deploying the respective policy on the x-axis. We point out that the set of ASes deploying the objects and the set of ASes deploying the policy do not necessarily need to be the same.

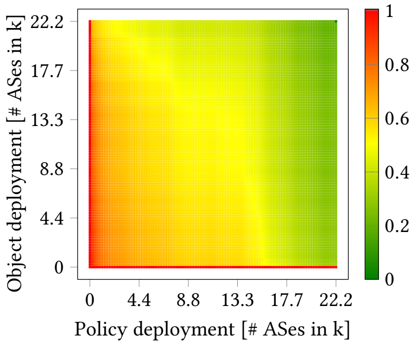
Figure 6.2a shows that the benefit of ASPA in route leak mitigation gradually increases as more objects and policies are deployed within the graph. We observe a negligible benefit below a threshold of 15k ASes for object deployment and below a



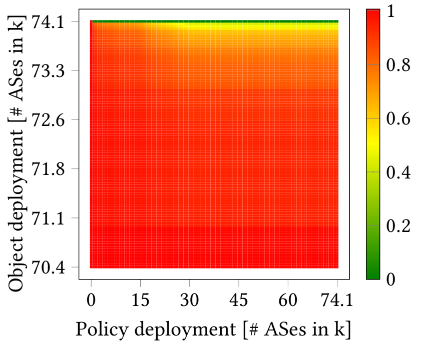
(a) Random object and policy deployment.



(b) Top-down object and policy deployment.



(c) Lower left enlarged from 6.2b.



(d) Bottom-up object and top-down policy deployment.

Figure 6.2: ASPA deployment scenarios. Objects and policies are deployed in all ASes.

threshold of 25k ASes for policy deployment. With the random deployment strategy, more than 45k ASes are required to adopt ASPA to impact the routing security level significantly. We are unlikely to witness such a high adoption in the wild.

Since random deployment offers few benefits, we focus on selective deployment with fewer objects created by ASes and fewer filtering policies required to be enforced by ASes in Figure 6.2b. However, instead of randomly selecting ASes, we strategically choose them to have the most impact. In this simulation, we implement a top-down deployment strategy, starting with the ASes with the highest out-degree. As a result, large ASes deploy the security solution first, while smaller ASes deploy the security solution at a later stage. We find that this deployment strategy yields a much higher impact on mitigating route leaks. In fact, large ASes are located rather at the core of the inter-domain routing infrastructure compared to smaller ASes being somewhat on the outside. Since the connectivity of large ASes is high and they sit in the middle of many forwarding paths, their filtering greatly impacts connected nodes as invalid routes are prevented from spreading further. It is important to note that even when tier-1 providers do not have providers themselves, they are required to issue ASPA objects containing *AS0* in them to show their adoption of the security solution. Otherwise, *no attestation* will be assumed by the algorithm, and the route would be unknown instead of invalid in the majority of cases. Unknown routes are accepted by the ASes deploying the policy whereas invalid routes would be filtered. Hence, it is essential that all tier-1 ASes that participate in ASPA publish an ASPA object containing *AS0*.

Overall, Figure 6.2b shows a significant improvement of the selective deployment strategy over the random deployment strategy. This is already achieved by a relatively small number of ASes deploying objects and policies. We take a closer look at the lower left corner of Figure 6.2b, only considering the largest 22k ASes, in Figure 6.2c. We observe significant benefits in route leak mitigation with only a few hundreds of the largest ASes deploying ASPA objects and policy (orange color). Once a deployment stage of 8k ASes (10.8%) for object deployment and 5k ASes (6.7%) for policy deployment is reached, we observe that 50% of route leaks are successfully mitigated (yellow color).

Deploying ASPA objects is much easier compared to deploying the policy, as the objects are created in a web portal at the RIR, similar to RPKI objects. Policy deployment works in an out-of-band fashion but still requires policy changes at the router to filter invalid routes. It is therefore important to point out that with increasing object deployment on the y-axis in Figure 6.2c, we already reach the yellow area in which 50% of route leaks are successfully mitigated, although only a very small fraction of ASes are required to deploy the policy, shown on the x-axis. On the other hand, deploying the policy in additional ASes with only a minimal number of ASes deploying the objects yields a quicker benefit as we reach the yellow area at about 13k ASes that deploy the policy. Deploying the policy has therefore more impact, but is more challenging to put into practice and we assume that more ASes will deploy objects rather than policy (similar to RPKI).

The previous paragraph focused on a top-down strategy with the largest ASes deploying ASPA first. ASPA is designed in the opposite fashion. An AS publishes an ASPA object that contains information about its providers. We therefore look at the opposite deployment strategy for the object creation, namely bottom-up, in Figure 6.2d. We deploy objects in this simulation from the smallest ASes towards the largest ASes, and policy deployment remains in a top-down fashion, as introduced before. Although the y-axis only shows the top 5% of ASes, we barely see the benefit of this deployment strategy at the very top of the figure. This is because the largest ASes are creating objects last. Since tier-1 providers are at the graph's core and the security solution is deployed last within them in this scenario, many route leaks are forwarded and not filtered. The deployment strategy itself is not to be preferred in any way, but it highlights the importance of the participation of tier-1 ASes. If tier-1 ASes do not deploy ASPA objects, there is no point in starting deployment elsewhere.

Overall, the selective ASPA deployment strategy in a top-down fashion is preferred over a random deployment strategy. A bottom-up deployment strategy of

ASPA objects only highlights the importance of winning tier-1 providers to implement the security solution first.

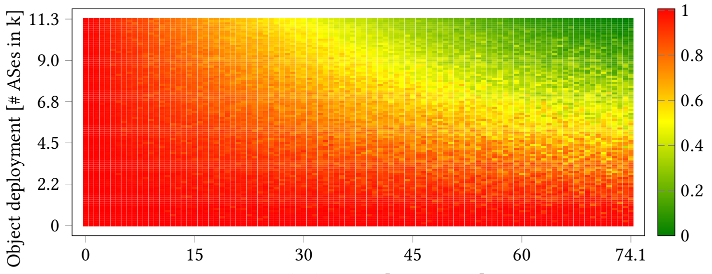
**RPKI projection.** In order to indicate how helpful ASPA deployment would be, given data from the ongoing RPKI effort, we show an RPKI projection line in the Figures 6.2a and 6.2b. It resembles the past 11 years of RPKI deployment from July 1, 2012 until September 30, 2023. We observe that in the beginning, only RPKI objects were deployed, so the RPKI projection only rises vertically. Once RPKI objects were available in the RPKI repositories, ASes started to deploy ROV, which would tilt the RPKI projection towards the right.

The underlying historical RPKI data assumes two linear functions. We obtained the data from publicly available RPKI history repositories at the RIPE NCC. While we are trying to resemble the RPKI history as closely as possible, it is evident that deployment has not been linear, and our RPKI projection remains an abstraction of historical data. From July 1, 2012, to December 31, 2019, we assume 0.007% growth per day; from January 1, 2020, to December 31, 2022, we assume 0.01768% growth per day. Our model starts on July 1, 2012 as the creation of ROAs was negligible during the first year since its inception in 2011.

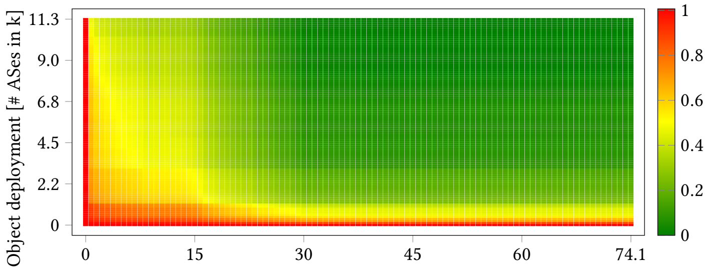
Inference of RPKI policy deployment rates, also called ROV, is much more challenging as RPKI ROV measurements are based on the inference of private router configurations. Many publications deal with different measurement methods to reliably measure ROV [14], [164]–[166], [168], [294]. See Chapter 4 for more details. In a nutshell, ROV identification methodologies are either capable of pinpointing ASes that are filtering but were only executed once at a certain point in time and, therefore, do not allow the inference of a growth rate [14], [165], or they were executed as continuous measurements, but include *collateral benefit* and therefore show higher adoption rates compared to methodologies capable of pinpointing filtering ASes [164], [168], [294]. In order to obtain a growth rate, multiple data points are needed, which is why we chose data from the latest publication [164], although presented numbers are likely higher compared to actual adoption.

Overall, we conclude that with the current state of RPKI deployment, ASPA would yield no benefit in the random deployment scenario but would yield significant benefit in route leak mitigation in the top-down approach, effectively reducing route leak propagation to below 50%.

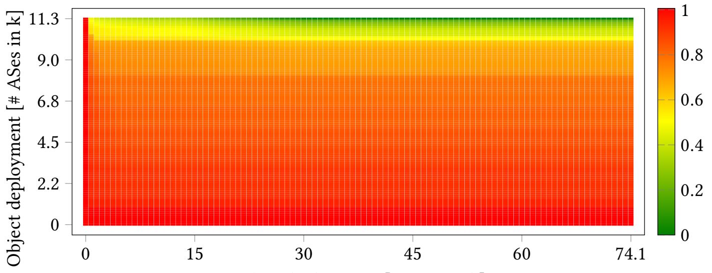
**AS-Cones.** We perform the same simulations with the AS-Cones algorithm. Our simulation results for the random AS-Cones object and deployment strategy are shown in Figure 6.3a. The y-axis scale changed compared to the ASPA figures since we only deploy AS-Cones objects for tier-1 and tier-2 ASes. Tier-3 ASes do not have a customer cone by definition and, therefore, do not require the issuance of AS-Cones objects. This is an advantage of the AS-Cones proposal as it significantly reduces the number of objects required to resemble the whole graph fully. Moreover, larger ASes are known to have more personnel and financial capabilities, making it easier for them to manage the issuance of AS-Cones objects. The AS-Cones policy, however, needs to be deployed in all ASes. It could very well be the case that a tier-3 receives a leaked route since no one else has mitigated the route leak before, and it



(a) Random object and policy deployment.



(b) Top-down object and policy deployment.



(c) Bottom-up object and top-down policy deployment.

Figure 6.3: AS-Cones deployment scenarios. Objects are only deployed in tier-1 and tier-2 ASes (11.3k). Policies are deployed in all ASes.

should be able to filter that route by implementing the security measure. Figure 6.3a shows that random object and policy deployment yields no benefit until at least 25k ASes deploy the policy or at least 5k ASes create AS-Cones objects. This is, however, an improbable scenario.

Instead, we focus again on a selective deployment strategy and start with the top-down deployment of objects and policies in Figure 6.3b. Again, objects only need to be deployed in tier-1 and tier-2 ASes, reducing the required amount tremendously. With only about 1k ASes deploying objects and policies, we observe a significant reduction in route leaks (orange color). Once the largest ASes deployed the AS-Cones objects, more objects up to 100% of object deployment will only yield about 50% of reduced *route leak success rate*. Deploying policy in many ASes is therefore crucial for the algorithm to work correctly.

The bottom-up strategy for object deployment is shown in Figure 6.3c. Again, since large ASes are deploying objects at the very end, the benefit in route leak mitigation is negligible until almost 100% of object deployment. Only with the support of large ASes will AS-Cones be able to make an impact on the overall routing security.

**Comparison.** This section provides a performance comparison of both algorithms under different deployment scenarios.

Firstly, both algorithms perform unsatisfactorily within the random object and policy deployment scenario. Suppose a random deployment strategy is nonetheless chosen. In that case, it is the AS-Cones algorithm that enables route leak mitigation with an object deployment rate of 9k ASes and a policy deployment rate of 35k ASes, see Figure 6.3a. To achieve the same benefits at ASPA deployment we would need to have 45k ASes in object and policy deployment, see Figure 6.2a. However, it is doubtful that such high adoption rates will ever be achieved. Moreover, random deployment is unlikely to be implemented, given the following deployment strategy.

Secondly, we instead plead for large ASes to deploy path plausibility algorithms in a top-down fashion. We see that both algorithms perform significantly better under such circumstances. ASPA only requires some hundred ASes creating objects and about 18k ASes deploying the policy to mitigate almost all route leaks. If 30k ASes create objects and some thousand ASes deploy the policy, ASPA achieves the same results. The RPKI projection line shows that such a scenario is much more likely to happen. AS-Cones requires only 1-2k ASes in object deployment and about 20k ASes in policy deployment to mitigate most route leaks. The increase in AS-Cones objects does not yield significant benefits once the largest ASes created their objects. Instead, significant improvements in route leak mitigation are only achieved through further policy deployment. Therefore it is ASPA that works best with many ASes deploying the objects (which is easy) and few ASes deploying the policy (which is more challenging).

Thirdly, we highlight the importance of starting with large ASes in our bottom-up simulations for object creation. With both algorithms, almost no mitigation is achieved. If large ASes do not participate in the adoption of path plausibility algorithms, smaller ASes will not be able to compensate. It is therefore crucial to incentivize large ASes to implement these new technologies as early adopters.

## 6.4 FORGED-ORIGIN PREFIX HIJACK SCENARIO

The forged-origin prefix hijack describes an attack in which the AS path attribute of a BGP announcement is manipulated. It is an intentional attack. The AS-Cones algorithm cannot protect against such an attack as the attacker would also be in control of the AS-Cones object content. The attacker could, therefore, manipulate the BGP AS path to mount a forged-origin prefix hijack and manipulate the AS-

Cones object by inserting the legitimate origin into the customer cone. In ASPA, the issuer of the object authorizes its providers. Therefore, it is impossible for a provider mounting a forged-origin prefix hijack to manipulate the ASPA object of a downstream peer. For the reason above, we only consider ASPA in this section.

Similar to the route leak scenario, we need first to establish a baseline of how many ASes would be affected by the forged-origin prefix hijack in our simulation without any security mechanism in place. Such a baseline allows for comparisons with increasing ASPA protection. The baseline is represented as the *hijacking success rate* by announcing the legitimate BGP announcement from the victim and then propagating the illegitimate BGP announcement from the attacker. It is calculated by the share of ASes that fall victim to the hijacked route versus the number of ASes which have the legitimate route installed in their RIB. We observe a mean *hijacking success rate* for 1,000 trials of 10.71%, see Figure 6.4, which translates to 7,937 ASes that are on average affected by the hijack. We point out that an ASPA object is only deployed within the victim AS if that particular AS is also chosen by the deployment strategy. This is a crucial fact, as route leak mitigation in other ASes is only possible if relationship information about the hijacked AS is readily available to the validation algorithm. If the validation algorithm has no knowledge of the relationship between hijacked AS and the attacker (as there usually is no relationship), the result would be *no attestation* instead of *invalid*. Additionally, the attacker AS has the *Default Policy* assigned to avoid filtering its own announcements.

We show results for mitigating forged-origin prefix hijacks with ASPA in the random object and policy deployment scenario in Figure 6.5a. The color-coded scale shows the *hijacking success rate* in comparison to the previously established baseline of 10.7%. A red color indicates no benefits in deploying the security solution, while the further the color moves via yellow towards green, the better the mitigation of the attack works. We observe that little benefits become visible only with about 30k ASes deploying objects and policies (orange color). It is apparent that with a random deployment strategy, only marginal benefits are created while many ASes

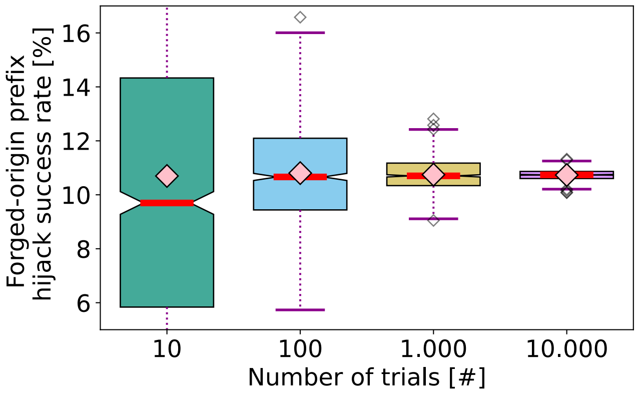
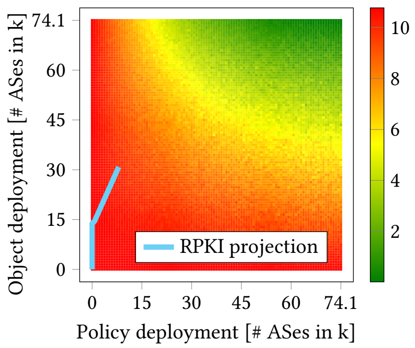
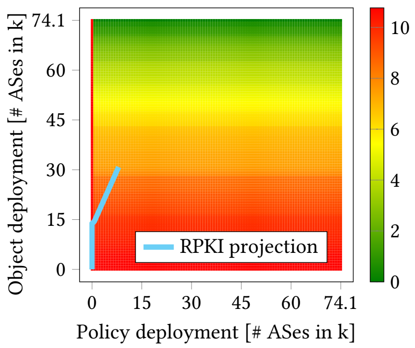


Figure 6.4: *Forged-origin prefix hijack success rate* for different number of trials.



(a) Random object and policy deployment.



(b) Top-down object and policy deployment.

Figure 6.5: Forged-origin prefix hijack scenarios with ASPA deployment.

would need to implement the solution.

We observe much better results with the top-down approach in Figure 6.5b. Once the largest ASes deploy the policy, further policy adoption only provides minor improvements. Instead, ASPA object creation is the key to successfully mitigating the forged-origin prefix hijacks. The more ASes publish information about their relationship and therefore protect their own AS from falling victim to such an attack, the lower the *hijacking success rate* becomes. Some major ASes at the core of the inter-domain infrastructure filtering the hijacked announcements are sufficient to provide collateral benefit to all ASes. Our results are similar to [287] as deploying their path-end validation algorithm already yields great benefits when deployed by only a few. In addition, for the forged-origin prefix hijacks, providing incentives to large ASes to support the security solution is of utmost importance.

## 6.5 DISCUSSION

In the following section, we discuss security considerations of ASPA and AS-Cones algorithms, how to provide incentives for deploying path plausibility algorithms, and the limitations of our simulation environment.

**Security considerations.** Both algorithms, ASPA and AS-Cones, can mitigate route leaks, but only ASPA can mitigate the forged-origin prefix hijack attack. AS-Cones can also detect the forged-origin prefix hijack attack as long as the provided AS-Cones objects contain correct information. ASPA objects carry the authorized providers, while AS-Cones objects have the customer cone as their content. In AS-Cones, the attacker is also in control of these objects. It is, therefore, easy for the attacker to manipulate the objects and insert the victim in its object, legitimizing the attack. Therefore, AS-Cones cannot detect such manipulation and can not be used to mitigate the forged-origin prefix hijack attack. Moreover, the victim of the hijack cannot withdraw or object to the AS-Cones object of the attacker in which it is listed as a customer.

To diminish the security drawback, the authors of the AS-Cones IETF draft introduced a 'verified' flag into the object. The flag is only to be set to true once the customer confirms indeed being a customer of the issuing party. Firstly, a tremendous advantage of AS-Cones is that it does not require small tier-3 ASes to create objects. This advantage would be eliminated if customers needed to do something to make the algorithm work. Secondly, the design of this addition is flawed. It does not solve the problem, as the attacker still controls the object. The 'verified' flag could be set to *true* by the attacker, and the attack can continue. Attribution of the attack would be possible by looking at historic BGP data in BGP collectors and comparing it to historic RPKI data containing the issued and altered AS-Cones objects. Hence, it is possible to attribute the attack to a particular AS that mounted it, but it cannot be prevented.

A design decision of the ASPA algorithm was to render upstream paths (routes received from a customer) valid without further checking if the contained AS path is only two hops long. For route leak detection, whether the origin is a customer, peer, or provider of the first-hop AS does not make any difference. The path would always be valid. The situation changes in case of a forged-origin prefix hijack attack. When a malicious provider of a customer AS sends that AS a hijacked route, the customer would not be able to check this route for correctness, even though the hijacked victim has an ASPA object published detailing that the provider is, in fact, not its provider. The route is valid in any case. This design flaw should be corrected in a future version of the draft. We assume that the draft's authors attempted to skip the check to save computational power and make the algorithm more efficient. In this case, at the expense of its rigorousness.

**Incentives.** In order to make sure that path plausibility algorithms are adopted, we need to think about incentives for ASes to start deploying the security solution. Our simulations show that, for a selective deployment strategy in a top-down fashion, minimal deployment already offers excellent benefits for the overall infrastructure. Nonetheless, there need to be early adopters to start deployment. An incentive for ASes to issue ASPA objects is to protect themselves against the forged-origin prefix hijack attack. In RPKI the creation of ROAs protects IP prefix space from exact and more specific prefix hijacks, which is why ASes started to create ROAs in 2012. Likewise, ASes will start the creation of ASPA objects to avoid their IP address space being hijacked by attackers. Our results show that tier-1 providers should start with creating ASPA objects. In fact, they only need to issue a *AS0* ASPA object as they do not need to authorize any upstream. Since the number of tier-1 providers is minimal and known, we propose to kick-start ASPA deployment by creating ASPA objects for the tier-1 clique by default. Once ASPA objects are available, it would make sense to start policy deployment. It is more challenging to convince the first ASes to deploy the policy as this directly relates to financial investment. One argument for deploying the policy of a path plausibility algorithm is to avoid customer traffic flowing into the direction of a route leak or a hijack. Customers might complain once traffic is routed in such a direction that the performance could be impacted or even discarded.

Another factor to consider is legislative pressure. As stated at the very beginning of the introduction of this thesis, the USA, EU, and Germany have acknowledged that problems in BGP present an imminent threat to their interests. Therefore, we might expect legislation in the future requiring ASes to implement specific security measures, such as RPKI or path plausibility algorithms.

**Limitation of simulations.** Our simulations are based on specific assumptions, such as the absence of additional security mechanisms like prefix filters, and cannot fully resemble real-world behavior. Many levels of abstraction are necessary to present the inter-domain routing infrastructure in a graph. For example, it is known that the CAIDA AS relationship dataset does not capture every AS and every relationship between ASes. These limitations lead to a biased view of the infrastructure and might also influence our results. Nonetheless, the assumptions used in our simulations are comparable to those in other research, but one should keep in mind that simulations will always take away complexity and, therefore, accuracy.

## 6.6 CONCLUDING REMARKS

In this chapter, we performed simulations to study how well the path plausibility algorithms ASPA and AS-Cones detect and mitigate route leaks and the forged-origin prefix hijack. We compared both algorithms in random and selective deployment scenarios.

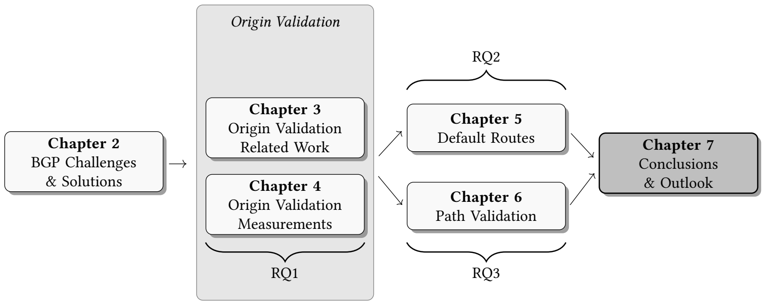
We found that both algorithms perform better in a top-down deployment strategy than in random deployment. A key factor for success is to win the largest ASes for deployment. Most route leaks can be mitigated with a minimal number of large ASes performing the actual policy deployment. Conversely, without the participation of large ASes, neither ASPA nor AS-Cones can be successful.

We also found that ASPA can mitigate forged-origin prefix hijacks. Policy deployment in large ASes shows the most impact. Only a tiny number of adopting ASes is required and then the impact is only increased by additional ASes deploying ASPA objects to cover the hijacked AS.

AS-Cones has the advantage only of requiring object deployment within tier-1 and tier-2 ASes. On the other hand, operators need to publicly specify their customer cone, which some might oppose. ASPA requires ASes to authorize their providers, which they might be more willing to do. ASPA can perform exceptionally well with more than 30k ASes deploying objects and only some hundreds deploying policies. At the same time, AS-Cones must have at least 15k ASes deploying the policies to mitigate most route leaks.

CHAPTER 7

# Conclusions and Outlook



*In the final chapter of the thesis, we revisit our research goal and questions. We summarize the answers to our research questions and draw conclusions as to whether the research goal has been achieved. Moreover, we provide an outlook and future research opportunities. Lastly, we present our contributions in the form of artifacts.*

## 7.1 MAIN CONCLUSIONS

The objective of this thesis was:

> *To assess whether we can build on top of the RPKI with algorithms securing the AS path to improve overall routing security.*

We achieved this goal by breaking the objective down into three research questions.

We started out by introducing in Chapter 2 that Border Gateway Protocol (BGP) prefix hijacking is still a threat to routing security and presented the Internet Routing Registry (IRR) and the cryptographically secure Resource Public Key Infrastructure (RPKI) as origin validation solutions. Since the RPKI only provides proof of address ownership, it does not protect against path manipulation attacks. These attacks can only be mitigated by implementing path validation algorithms. We highlighted how Border Gateway Protocol Security (BGPsec) addresses path manipulation issues but fails to consider real-world circumstances, such as partial deployment and additional computational expenses. Moreover, we introduced route leaks, a violation of BGP routing policies, and presented recently proposed path plausibility algorithms that aim to address path manipulation issues as well as route leaks. Because of that, we conclude that studies to measure the current state of RPKI deployment are required to infer whether origin validation is deployed on a wide scale. Moreover, we conclude that deploying path validation algorithms, more specifically, path plausibility algorithms, is necessary to mitigate path manipulation attacks and route leaks.

Next, we identified research gaps in the study of inter-domain routing security measurements in Chapter 3. We did so through an extensive literature survey, developed a classification scheme to assess existing RPKI measurement research, and divided the body of research into Route Origin Authorization (ROA) measurements, Route Origin Validation (ROV) measurements, and RPKI resiliency. We found that existing RPKI ROV measurements are inaccurate and falsely pinpoint ROV-filtering Autonomous Systems (ASes).

As a result, we focus our attention in RQ1 (How can we improve the identification of RPKI ROV deployment in an AS without running into the problem of wrong attribution?) on obtaining an accurate view of the current RPKI ROV deployment. We used controlled data plane measurements to answer this research question. Our research showed that existing methodologies do not consider *collateral benefit*, which leads to wrong attribution of RPKI ROV filtering. Consequently, the presented adoption rates of related work are too high. Our results show much lower adoption of RPKI ROV filtering. However, lower adoption rates do not imply that the Internet is less protected, as other ASes are still protected via collateral benefit. We also show through our results that RPKI ROV adoption is increasing, although at a lower rate than previous research anticipated. Moreover, we analyzed RPKI relying party software and found Routinator to perform best regarding its correctness, simplicity, and performance. Hence, we recommend Routinator to operators for new deployments.

We continued in RQ2 (How can we identify default routes that are present in an AS?) to identify and quantify default routes that irritated our earlier RPKI measurements. Our results show that default routes are much more common than anticipated and are even present in the Default Free Zone (DFZ). Moreover, the smaller the AS, the higher the likelihood of default routes being present. We derive from our measurements that existing Internet measurements should carefully consider the presence of default routes as they heavily influence active data plane measurements in which the absence of a control plane route is assumed, e.g., reachability checks. As an additional result, we found middleboxes to tamper with data plane packets and, therefore, influence measurement results to a great extent. Based on this, we developed a method to identify middleboxes that excludes such interferences from data plane measurements.

Based on our conclusion that only path validation algorithms are going to mitigate path manipulations and route leaks, we focus in RQ3 (What are the advantages and disadvantages for different deployment scenarios of path plausibility algorithms to improve inter-domain routing security?) on two path plausibility algorithms that could potentially solve some of the presented issues. We implemented Autonomous System Provider Authorization (ASPA) and AS-Cones in a simulation testbed. Our results show that even with little deployment, starting at the largest ASes, significant benefits in route leak and forged-origin prefix hijack attack mitigation can be achieved. We also found that path plausibility algorithms can only succeed with the participation of tier-1 providers. Based on this, we conclude that we should start path plausibility deployment at tier-1 providers, possibly incentivizing them to become early adopters.

Overall, we found that the RPKI has become a mature technology that, on January 15, 2024, protects ~47% of the IPv4 prefix space with ROAs. RPKI ROV is picking up, too, although at a slower pace than anticipated by other studies. We saw that default routes are primarily present in smaller ASes but are more common than anticipated and even found some default routes within the DFZ. We ran extensive simulations to determine whether we can build on top of the existing RPKI with path plausibility algorithms. We found that the RPKI can be extended with only a small amount of effort to support additional cryptographic objects, like ASPA and AS-Cones. Based on our results, we conclude that with a selective top-down deployment strategy of path plausibility algorithms, the impact is very high at a relatively small cost. In addition, partial deployment is supported, and an immediate benefit can be derived at a low adoption rate.

Therefore, overall routing security can be improved by deploying path validation algorithms securing the AS path on top of existing origin validation technologies like the RPKI.

## 7.2 REVISITING RESEARCH QUESTIONS

To provide a more detailed view of the approach and results contributed within this thesis for each research question defined in Chapter 1, we revisit each of them in the following paragraphs, starting with:

> **RQ 1**—*How can we improve the identification of RPKI ROV deployment in an AS without running into the problem of wrong attribution?*

Since we have shown in Chapter 3 that current methods insufficiently pinpoint ROV filtering ASes, we developed a controlled data plane measurement methodology that accurately identifies RPKI ROV filtering ASes. Our measurements were controlled by using an anchor and an experiment prefix within our control via the PEERING testbed. We swapped the prefixes regularly to confirm observations and eradicate random noise. We used data plane measurements via traceroute data obtained from RIPE Atlas probes to monitor the actual traffic flow. We made strong inferences in three categories respecting the connected assumption of previous work [165]. We also made weak inferences in three additional cases that relax the previously mentioned connected assumption to extend the measurement coverage. Furthermore, we extended the current state-of-the-art by considering Internet Exchange Point (IXP) traversals and building an include list that allowed for the differentiation between fully and partially filtering ASes. In 2021, from 5,537 vantage points in 3,694 ASes, we inferred ROV-deployment in 206 unique ASes: 10 with strong confidence, 12 with weak confidence, and 184 indirectly adopting ROV via IXP route servers.

Adoption rates presented in our study are lower compared to other studies in the field. Some studies wrongly attribute ROV-filtering to downstream ASes, resulting in higher adoption ratios. It is evident that downstream ASes and ASes peering at route servers are benefiting from RPKI ROV deployment at upstreams and IXPs, but they do not perform the filtering themselves. Nonetheless, RPKI ROV deployment is picking up.

Throughout the development of the RPKI ROV measurement methodology, we noticed false negatives from ASes that deployed RPKI ROV but were not correctly identified. Intensive discussions and debugging led to the conclusion that default routes could be the underlying issue. We therefore asked:

> **RQ 2**—*How can we identify default routes that are present in an AS?*

Default routes provide data plane connectivity although RPKI-invalid prefix ranges have been filtered on the control plane by the adopting AS. We extended two separate methodologies to identify whether a default route was present in an AS. Firstly, we considered IPv6 in addition to IPv4 and used dual-poisoning to improve the efficiency of the path-poisoning methodology. Where available, we added RIPE Atlas probes to the measurements to infer the direction of the default route. Secondly, we added NLnog vantage points to increase coverage of the not-announced prefix methodology. For evaluating both methodologies, we introduced a threshold value to decide on the ratio that would be applied before an AS is flagged as having a default route installed. In total, we found default routes to be present in 31.52% of ASes using RIPE Atlas vantage points. Splitting the ASes under test into tiers, we could measure 0%, 23.31%, and 41.14% for tier-1, tier-2, and tier-3 ASes, respectively. These results show that default routes are much more often deployed in ASes that are closer to the edge of the Internet topology than the core. They also show that ASes within the DFZ have default routes installed, although this is not allowed by definition. Our methodology is implemented as an ongoing measurement campaign. Results are available at: defaultroutes.net

Since our overall goal is to assess whether we can build on top of the RPKI with path plausibility algorithms and our findings from RQ1 sanitized by the findings from RQ2 showed that the RPKI is picking up, we asked the following question:

> **RQ 3**—*What are the advantages and disadvantages for different deployment scenarios of path plausibility algorithms to improve inter-domain routing security?*

Path validation or path plausibility is required to secure the AS path attribute of BGP and avoid path manipulations. In a simulation testbed, we implemented and evaluated the path plausibility algorithms ASPA and AS-Cones. A disadvantage was that they provide fewer security guarantees than a path validation algorithm, e.g., BGPsec, but offer significant advantages regarding partial deployment and operational cost. Our simulations showed that with only a small percentage of ASes deploying the path plausibility algorithm ASPA, significant benefits in route leak mitigation were achieved. The top-down deployment scenario yields the best outcome, starting with the largest ASes. More importantly, without the participation of tier-1 providers, path plausibility algorithms are ineffective. Since ASPA protects against the forged-origin prefix hijack attack, a significant reduction of attacker success likelihood could also be achieved in the event of such an attack. The RPKI is capable of supporting the objects of additional path plausibility algorithms, and we expect to see that these algorithms will receive significantly quicker adoption and, therefore, secure the inter-domain routing infrastructure to a greater extent compared to alternatives such as BGPsec.

## 7.3 FUTURE RESEARCH DIRECTIONS

Several directions of research were not explored during this thesis. We highlight potentially fruitful extensions and areas of research in the following subsections.

**Unified probes API.** A typical measurement problem is coverage. Coverage mostly depends on how many probes, in our case, RIPE Atlas probes, are present in the infrastructure to be measured. We relaxed the connected assumption of our RPKI ROV measurements. However, the longer the BGP path gets, the more rigorous checks will have to be performed to ensure that the attribution of filtering ASes is correct. At some point, too many factors are in play to make sound judgments. Our measurement campaign coverage could be increased by adding additional vantage points. These could be NLnog nodes, browser-based vantage points, PlanetLab nodes, or similar. The problem is the management of several platforms in a single measurement campaign, as there is no unified API to access all measurement platforms. This would be a handy extension. The same holds for measurement systems on the control plane. Looking glasses are available in many ASes but require manual login and search. There have been efforts to create a more general webpage for looking glasses, but it is still impossible via an API to run measurements in an automated fashion. This would significantly help improve measurement coverage.

**Increasing RPKI adoption.** With increasing RPKI adoption, most measurement methodologies will yield less benefit until they no longer work. This is because control plane and data plane measurements rely on RPKI-invalid test prefixes used for probing. If RPKI deployment continues to rise, RPKI-invalid prefixes will be filtered much closer to the announcing AS; therefore, testing ASes further away from the announcing AS will not be possible anymore. The same holds for statistical approaches. The idea is to rely on statistical differences between RPKI-valid and RPKI-invalid prefix announcements. An AS deploying RPKI is not expected to announce many RPKI-invalid prefixes. Suppose RPKI deployment continues to rise. In that case, statistical differences will become smaller, and the creation of clusters to differentiate between filtering and non-filtering ASes will no longer be as clear. In addition, ASes not filtering RPKI-invalid prefixes will simply receive fewer RPKI-invalid prefixes to propagate and therefore reduce their footprint. New approaches are necessary that are not influenced by the increasing deployment of RPKI itself.

**Default routes.** During our study, we extended two measurement methodologies: the path-poisoning and the not-announced prefix methodology. Both are suboptimal, either in speed or in coverage. The path-poisoning methodology requires many /24 prefix ranges to perform at an acceptable speed. Even then, measurements will take months to complete. The not-announced prefix methodology requires active probes within the ASes under test. Coverage is, therefore, extremely limited. More ideas are necessary to increase speed while not requiring the vantage point to reside within the AS during testing.

**Path Validation and Path Plausibility.** Many new ideas are being discussed within the Internet Engineering Task Force (IETF). It appears that the ASPA approach is moving forward and will be favored for deployment since BGPsec is not expected to be deployed shortly. The first ASPA object in the RPKI database was published some time ago, and a mailing list participant recently claimed that ASPA has mitigated the first route leak [295]. However, there is still a long way to go from deploying an experimental algorithm to a production system deployed worldwide. AS-Cones (or AS-Groups, for that matter) could be an alternative since only tier-1 and tier-2 ASes are required to provide relationship information. However, we saw that AS-Cones has weaker security properties. The ideas most lacking are real-world implementations to evaluate the proposed algorithms properly. We argue for more simulation and native implementations of proposed algorithms. Our future work in this context will focus on the IETF draft [296], which classifies route leak scenarios and proposes the use of a Down Only (DO) BGP Community to signal downstream ASes the direction of the route. RFC 9234 [297] uses a new BGP

Role Capability in UPDATE and OPEN messages instead to establish a relationship between two peering partners at session start and label routes accordingly. An optional, transitive BGP Path Attribute, called Only To Customer (OTC), detects and prevents route leaks. We are comparing the two newly proposed IETF drafts and ASPA and AS-Cones.

**Testbed development.** During the course of this thesis, we developed an extensive testbed for the NIST SRx framework. Our current version supports up to 55,000 containers, each running a Quagga daemon within its own docker container. We could not, however, scale the testbed to 74,000 nodes, the current amount of ASes facilitating the exchange of traffic on the Internet. Significant engineering effort is required to achieve such a goal. Technical limitations seemed to be a restraining factor, but better results might be achieved with a different choice of the underlying orchestration framework. To scale the testbed to 74,000 nodes would bring significant benefits. An experimenter could create a docker image with their algorithm and deploy it using our framework. Instead of creating an artificial implementation for a simulation testbed, the actual implementation that will run in an AS at a later stage could be tested. Theoretically, simulations are sufficient to get an understanding of the workings of an algorithm. Practically, testing the actual implementation should always be preferred because some problems cannot be simulated and will only appear once appropriately deployed.

## 7.4 ARTIFACT CONTRIBUTIONS

Throughout each chapter, supported by the respective publications, the following artifacts were created:

1. We were responsible for maintaining ongoing control plane measurements of rov.rpki.net from 2018 until the measurements were discontinued in 2020. To collect BGP dumps, BGPReader [150] was used. CAIDA modified the BGPReader API in 2018 to version 2, which required us to adapt the bgpreader_util library[1] to make BGPReader API calls. Our new version has been publicly released[2].

2. We provided our RPKI ROV measurement framework in the form of scripts as a public repository[3]. Moreover, we used the pyasn library to map IP addresses to AS names. Instead of relying on the single-threaded original version, we implemented a multithreading capability that allows us to simultaneously process many MRT/RIB BGP archives to DAT files. We also integrated the bgpstream RIPE RIS and Routeviews resources into the framework, eliminating the need for FTP downloads. Jonas Dannwolf contributed to the bgpstream

[1]https://github.com/reuteran/reuter_util
[2]https://github.com/nrodday/bgpReader_util
[3]https://github.com/nrodday/TMA-21

integration[4]. We contributed to the TraIXroute tool[5].

3. We provided the German translation for the NLNetLabs RPKI CA software Krill[6].
4. We released our default route measurement code and datasets[7]. Lukas Kaltenbach contributed to the implementation. We also present ongoing measurement results on: defaultroutes.net.
5. We released all source code and scripts necessary to set up and use our topology generator framework[8].
6. We implemented the AS-Cones algorithm into the NIST BGP-SRx software suite and released all source code[9]. Kai Hamich contributed to the initial implementation, while Nils Höger added improvements to the code.
7. We implemented and open-sourced an ASPA extension for the goBGP daemon as an add-on to the NIST BGP-SRx software suite[10]. Nils Höger contributed to the implementation.
8. We provided a reference implementation for the ASPA and AS-Cones algorithms in a Python BGP simulator. Moreover, we developed unit tests and released all related source code[11].
9. We were accepted at the IEEE/IFIP Network Operations and Management Symposium (NOMS) in 2020 to give a tutorial titled: 'Reliable measurements with BGP and RPKI'[12]. The exercises were publicly released[13]. The tutorial was joint work with Mattijs Jonker.

[4]https://github.com/nrodday/KompT-Dannwolf/tree/bgpStream
[5]https://github.com/gnomikos/traIXroute/commit/7fa1bf51f855213b77035915c11473807bdb1d2f
[6]https://github.com/NLnetLabs/lagosta/blob/master/src/locales/de.json
[7]https://github.com/nrodday/TAURIN-21
[8]https://github.com/nrodday/CNSM-23
[9]https://github.com/nrodday/CNSM-23-demo
[10]https://github.com/nrodday/CNSM-23-demo
[11]https://github.com/nrodday/NOMS-24
[12]https://noms2020.ieee-noms.org/program/tutorials.html
[13]https://github.com/nrodday/NOMS_2020_Tutorial

APPENDIX A

# BGPEval: A BGP Topology Generator

This chapter introduces a BGP topology generator framework we built to evaluate path plausibility algorithms. The idea was to evaluate these algorithms in the most realistic setting without using simulations, as they always rely on many assumptions. However, during the course of the development, we discovered that we could not scale the resulting topology for technical constraints to a size necessary to mimic a full inter-domain routing AS graph. Therefore, we resorted to simulations in Chapter 6 and did not use the BGP topology generator for its initially intended task.

Although we could not scale the testbed to a size of 74,110 ASes, we made significant improvements compared to existing work and therefore introduce our BGP topology generator framework as part of the appendix in this thesis. This work was presented as a conference paper [19] and a conference poster [20].

Our framework is based on the National Institute of Standards and Technology (NIST) BGP-Secure Routing Extension (SRx) software suite [292]. It allows for easy integration of new security algorithms and is designed for testing new features in inter-domain routing security. The NIST BGP-SRx software suite allows the spawn of ASes as containerized applications in Docker. A significant limitation is that each topology requires a definition in a configuration file. The testbed is afterward spawned according to that configuration file. It is cumbersome to create such configuration files for each topology manually. Therefore, our work extends the NIST BGP-SRx software suite by adding a topology generator that takes a directed graph in Python as an input and generates configuration files per AS within the topology. The input graph can be generated on the fly with publicly available BGP collector data [140], [146], enriched with CAIDA's AS relationship dataset (as-rel2) [141]. Hence, it becomes trivial to evaluate algorithms running within the NIST BGP-SRx software suite within an up-to-date topology that resembles the actual inter-domain routing infrastructure. Moreover, we provide an architecture to scale the existing NIST BGP-SRx software suite to up to 55,000 containers by adding several layers of abstraction.

While this framework was built for the evaluation of path plausibility algorithms within the NIST BGP-SRx software suite, it finds areas of application much beyond that. The framework runs a daemon within the container to interconnect ASes in the emulation environment. The containers can run any arbitrary software. As a result, it is easily possible to use our topology generator framework for other scenarios

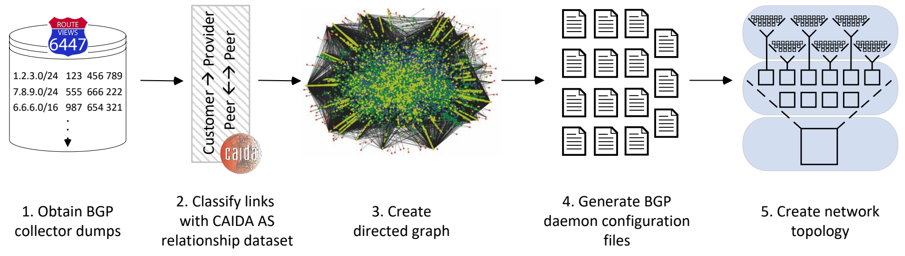

ROUTE VIEWS
6447
1.2.3.0/24 123 456 789
7.8.9.0/24 555 666 222
6.6.6.0/16 987 654 321
Customer → Provider
Peer ←→ Peer
caida
1. Obtain BGP collector dumps
2. Classify links with CAIDA AS relationship dataset
3. Create directed graph
4. Generate BGP daemon configuration files
5. Create network topology

Figure A.1: BGPEval methodology. We obtain BGP collector data and enrich the data with Center for Applied Internet Data Analysis (CAIDA) AS relationship information to obtain a directed graph. Afterward, we generate BGP router configuration files and create the testbed via multiple layers of abstraction.

beyond inter-domain routing. A possible scenario is the creation of staging environments for companies. Before new components are rolled out within a company's infrastructure, it is wise to perform testing. Tools can capture enterprise traffic and build a directed graph representing the internal topology [298], [299]. This graph can be fed into our framework and emulate the existing topology. In summary, the capabilities of our framework extend well beyond the area of BGP routing.

## A.1 BACKGROUND

There are several methods available to evaluate new algorithms. Simulation frameworks are the most prominent examples, namely NS-3 [300], Omnett++ [301], and GNS3 [302]. Simulation environments allow us to evaluate an algorithm on a theoretical level sufficiently. However, there will always be additional obstacles and things to consider in real-world deployments. Many dependencies are abstracted during simulations. Moreover, source code is usually rewritten for simulation environments. Therefore, it is not the same source code that is evaluated and running in a production environment at a later stage. Emulation is used to ease some of the pain points introduced above. MiniNet [303] is a prominent example that relies on Software Defined Networking (SDN) technology. It allows the execution of the same binaries compared to the production environment. Automated network configuration has been proposed for MiniNet [20] and GNS3 [304].

Our work differs from the previously mentioned simulation and emulation environments in that we do not require any SDN technology and yet provide an emulation environment where the same binaries can be tested compared to the production environment. In addition, BGPEval focuses on the automation of network topology generation. This is a problem that is unsolved for most simulation and emulation environments.

## A.2 METHODOLOGY

Our proposed methodology is shown in Figure A.1.

**Input graph.** To create an input graph we use publicly available BGP collector data from RouteViews [140] and Réseaux IP Européens (RIPE) Routing Information Service (RIS) [146] via CAIDA's BGP reader [150] interface from June 1, 2023, 00:00:00 until 23:59:59. The obtained file contains 678 GB of data. The graph is enriched with relationship information in the following step. We use the CAIDA AS relationship dataset (as-rel2) [141] to create directed edges between nodes. An edge from A to B implies that A is a customer of B. Peering relationships have a directed edge in each direction. The final graph consists of 74,110 nodes. Similar to the method used in Chapter 5, we classify the ASes into three tiers, following [17]. The final graph has 105 tier-1, 11,237 tier-2, and 62,768 tier-3 nodes. The created graph is an abstraction from the inter-domain routing infrastructure and cannot sustain all of its properties. Some relationships might not be contained in CAIDA's inferences; some peering relationships might not be observable in the BGP collector dumps. It is important to note that there are limits to such abstraction methods. The final graph is serialized and stored on disk.

In step three of Figure A.1, we read the graph as a serialized object in Python and create an instance in memory. We introduce the serialization step to allow for other scenarios in which the generation of the topology and the resemblance of topology are decoupled. Next, we recreate the graph using the Python networking library NetworkX [305].

**Configuration file generation.** Step four is the most important in our BGPEval framework. We provide configuration templates for Quagga and GoBGP. Based on the provided input graph, we infer the relationship and create configuration files for the chosen format. The amount of configuration files matches the amount of ASes present in the graph. Other researchers can extend our framework easily by providing additional Jinja2 [306] templates for their desired output files.

Moreover, each configuration file contains a policy component that defines how BGP announcements are forwarded. Similar to the policy component in Chapter 6, we designed each BGP policy to respect the Gao-Rexford model [44].

**Testbed creation.** Step five creates the actual network topology and spawns the testbed. We use several layers of abstraction. The larger the topology, the more crucial such a layering approach becomes. We support single or multiple hardware servers within our setup. Small topologies might only require a little computational power, while many thousands of containers (each container replicates one AS) require more resources. The limit to spawning Docker containers per machine is at ~800 such containers. Afterward, the machine becomes unresponsive. We suspect creating network interfaces per container is the underlying issue. Therefore, we only spawn 450 containers per VM to maintain a stable state. Each hardware server hosts multiple VMs.

The architecture has three total layers, as shown in Figure A.2. At the very bottom, we used three hardware servers. They are interconnected via a data link

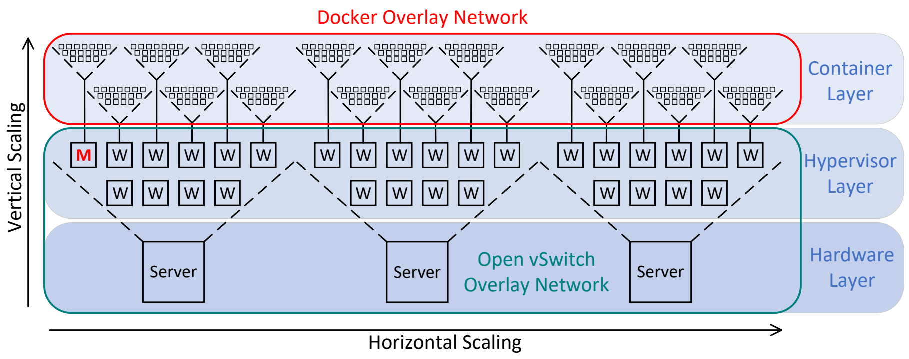


Figure A.2: BGPEval architecture. We observe three layers. Servers at the hardware layer, Virtual Machines (VMs) at the hypervisor layer, and containers at the container layer. The hypervisor layer has a single manager instance and many worker instances. Servers and VMs communicate via an Open vSwitch Layer 2 overlay network, while containers are interconnected via a Docker overlay network. Each container requires a static IP address to generate router configuration files.

overlay network. The overlay network is created by an Open vSwitch bridge and a tunnel interface for each adjacent server tunneled through an Secure Shell (SSH) connection. Hence, the hardware servers must only have SSH connectivity. We recommend enabling Spanning Tree Protocol (STP) to avoid forwarding loops if more than two hardware machines are used. As a result, the three hardware servers can exchange traffic on the data link layer.

At the middle layer, we host many VMs per hardware server. The VMs are instantiated in a Kernel-based Virtual Machine (KVM) hypervisor. All VMs are connected to the previously created Open vSwitch bridge on the host via a dedicated interface. Hence, all VMs are able to communicate with each other, regardless of their physical location. Each VM interface needs an Internet Protocol (IP) address to exchange traffic. We create a Dynamic Host Configuration Protocol (DHCP) server on a manager VM (highlighted in red in Figure A.2) that distributed IP addresses among clients. The amount of VMs per hardware server depends on its computational resources.

At the very top, we create Docker containers. This layer hosts the actual experiments, as the BGP daemons are executed within the Docker containers. To manage such a complex Docker deployment, we use Docker Swarm [307]. The manager VM hosts the Docker swarm manager and the Docker image repository. The rest of the VMs is clones at the start and automatically joins the swarm as worker nodes. With such a simple setup, it is easy to prune and rebuild the testbed as desired.

Each service running in Docker must be contained within an image. This image is uploaded to a registry and distributed to all worker nodes. The containers can be spawned once the image is available in the VM worker nodes. We deploy each container as a Docker service. Moreover, our previously generated configuration files are deployed within the Docker swarm as Docker configurations. Docker takes care of distributing the configuration files to the correct worker nodes, simply requesting their respective file and starting the container. An additional overlay network is required to make the containers talk to each other. A dedicated network interface is spawned within each container that connects to the Docker overlay network. The current version of Docker Swarm does not allow the assignment of static IP addresses. This, however, is required by our setup as the IPs need to be entered into the configuration files before spawning the testbed. Therefore, we need to know the IP and statically assign them to container interfaces beforehand. As a workaround, we log into each container via SSH in an automated fashion after the container has been spawned and disconnect and reconnect the interface from the overlay network. This action ensures the correct IP address is present at the interface. The Docker management engine, however, is unaware of that change. This is only a problem if a container crashes and is restarted by the management engine. We avoid such crashes by respecting the limits of each VM.

Once the IP address change has been performed, the NIST BGP-SRx server starts. The BGP daemon is started afterward and connects to the NIST BGP-SRx server for validation requests and to all its BGP peers.

In the end, we obtain a fully functional BGP testbed that interconnects many containers according to the provided input graph. We can log into any of the ASes to manually overwrite BGP announcements to see what happens within the network.

## A.3 NIST BGP-SRX SOFTWARE SUITE

This work extends the NIST BGP-Secure Routing Extension software suite [292] which has several components: BGP-SRx server, Quagga routing daemon, RPKI test harness, and BGPSecIO generator.

**BGP-SRx server.** The BGP-SRx server functions, contrary to the IETF drafts, as a validation instance to outsource the validation procedure from the router. The

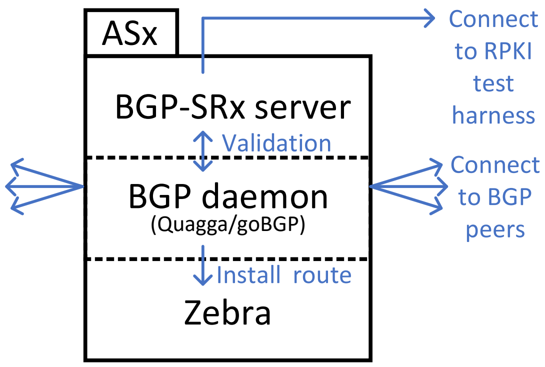


Figure A.3: Container components.

IETF draft for BGPSec [11] does not propose such an architecture, but it comes with some advantages. Firstly, the router does not need to perform the cryptographic operations itself but instead only sends a query to the SRx-server which returns the results whenever ready. Secondly, it allows to use one SRx server with multiple BGP daemons. However, it also introduces additional complexity as another module is needed. The current BGP-SRx server implements RPKI (RFC 6811), BGPsec (RFC 8205), and ASPA (Draft Version 1).

**Quagga.** The development of this routing daemon is discontinued as the FRRouting [308] daemon continued most features but it is nonetheless still used in a lot of setups. The daemon implemented an SRx-Proxy Application Programming Interface (API) that allows for the communication with the SRx server and is able to request validation for RPKI, BGPsec, and ASPA.

**RPKI test harness.** The test harness is a RP software that provides the RPKI and ASPA objects to the SRx server. Objects can be easily added via a Command Line Interface (CLI). It only provides the objects without any cryptography involved, and it does not implement RPKI delegation, signing, and publication procedures but directly exports the Validated ROA Payload (VRP) to the BGP-SRx server.

**BGPSecIO generator.** In order to quickly generate BGP traffic, the BGPSecIO generator can be used. It imitates and AS with a number and sends out predefined announcements. These announcements can be crafted by the experimenter to their liking.

## A.4 PROOF OF CONCEPT

We create two container images for our proof of concept. The architecture is shown in Figure A.3. Our first image uses Quagga, the second goBGP, as a routing daemon. The topology of 74,110 ASes that we attempt to create is based on a BGP collector dump, as described in Section A.2. Zebra [309] is used to install the routes the BGP daemon receives into the Route Information Base (RIB). With Zebra installing the routes, sending data plane packets confirming connectivity is possible. A single test harness is connected to a maximum of 1,000 SRx instances to avoid overwhelming it. As a result, several test harness instances are required that we host on different VMs to avoid overwhelming a single node.

Our physical infrastructure spans three servers:

1. 56 Cores @2.1 Ghz, 768 GB RAM
2. 128 Cores @2.45 Ghz, 2,048 GB RAM
3. 128 Cores @2.45 Ghz, 1,024 GB RAM

We managed to scale the testbeds as large as 55,000 ASes for the Quagga image and 45,000 ASes for the goBGP image. Creating a single testbed takes up to two hours, as we spawn the containers linearly for processing capability reasons.

## A.5 LIMITATIONS

**Scaling limit.** We could not create a testbed as large as 74,110 containers to support path validation algorithm evaluations within a resemblance of the inter-domain routing infrastructure. Unfortunately, latencies spiked within the created overlay networks that caused connection interruptions and reconnects once we attempted to create testbeds larger than 55,000 containers. Since the management engine of Docker needs to track the states throughout the different layers of abstraction, it seems to create significant problems beyond that size. In preparation for a large-scale testbed, our operating system and application layer limits have already been increased to a maximum [310]. In addition, instead of using the default networks, we replaced them with much larger /8 networks to provide enough address space for our many containers. We report that CPU and RAM utilization do not seem to be an issue. At the point where the infrastructure crashes, none of the indicators above show abnormalities.

Nonetheless, previous research and industry contributions were limited to ∼10,000 containers within a single testbed. Scaling a testbed to more than five times larger than what was possible before is a significant achievement.

**Gao-Rexford.** We implemented the Gao-Rexford model [44] to imitate routing policies at ASes. Real-world configurations are much more complex compared to our model. We do not consider different pricing models for two upstreams or other agreements. We also assume proper routing behavior by all participants, which does not always hold. Testing abnormal routing behavior like route leaks would require changing the presented configuration files.

**AS-level abstraction.** An AS does usually not comprise just a single router. Reality is much more complex, and ASes can comprise many thousands of routers that all might have the same but also might have different policies implemented. Our abstraction aligns with other studies, yet there remains a margin for error in such a coarse-grained implementation. Including details on intra-domain routing mechanisms would significantly increase the complexity of endeavors such as ours and are therefore regularly omitted. There is, however, a drawback to such simplifications. When different policies are deployed at other edge routers, such as partial deployment of security solutions, e.g., only for customers, we cannot represent them appropriately. Using our abstraction, either a whole AS is filtering or not.

**RPKI test harness.** Multiple RPKI test harnesses create the problem of data synchronization. Chapter 3 detailed the problem of a mirror world attack. Different knowledge at router A and B about the available RPKI objects creates a significant problem, leading to other routing decisions. We, therefore, need to ensure that the data available at the several instances of RPKI test harnesses is identical or risk an evaluation that produces unreliable results based on the drawbacks mentioned above of a mirror world attack.

## A.6 CONCLUDING REMARKS

This chapter presented our work on BGPEval: a BGP topology generator framework. We extend the NIST BGP-SRx software suite with a framework that takes a directed input graph and generates configuration files for BGP daemons. Moreover, our framework spawns the given topology and allows for evaluating security algorithms.

We could scale our testbed to a size of 55,000 containers. As the inter-domain routing infrastructure comprises of more than 74,000 ASes, we were unfortunately unable to use this framework to evaluate path plausibility algorithms. However, it is suitable for any topology smaller than that limit. The services within the containers are exchangeable. Therefore, it might see many more application areas than the one presented here.

APPENDIX B

# Ethical Considerations

Throughout this thesis, we perform many active and passive Internet measurements. It is important to consider ethical considerations when performing such measurements to avoid harm being done to people or infrastructure. We follow the guidelines outlined by Partridge and Allman [311] for network measurement papers. We are also aware of the Menlo report [312], detailing ethical principles in general for information and communication technology research and, more recently, the work by Pauley and McDaniel [313] analyzing existing ethical frameworks for Internet measurement studies.

## B.1 PASSIVE MEASUREMENTS

Data collected passively does not impact the operation of the respective infrastructure. The data contained within the datasets we use does not have personally identifiable information or any other information requiring encryption or anonymization. We use datasets passively collected by third parties, e.g., CAIDA's AS relationship dataset [141] and RIPE RIS [314] and Routeviews [140] BGP collector data. We only query the provided services within their expected load requirements.

## B.2 ACTIVE MEASUREMENTS

We also perform many active experiments on the control and data planes. Active measurements do interfere with the infrastructure and could potentially be harmful. They require careful design and execution.

**Control plane.** measurements inject BGP announcements into the inter-domain routing infrastructure to observe their propagation. We used the PEERING testbed [153] to conduct these experiments. The PEERING testbed provides an acceptable use policy on their website that we followed [315]. Moreover, our measurements were conducted with the support of the maintainers of the PEERING testbed, and with their help, we assessed the impact of each measurement before execution.

**Data plane.** measurements were used to verify connectivity to certain hosts. Since the PEERING testbed is a research platform that does not have unlimited bandwidth, we throttled our data plane bandwidth to an acceptable amount. Moreover, we used RIPE Atlas extensively during our studies. Ripe Atlas provides an ethics blog entry [316] that we followed. Since the increase of our limits to perform large-scale measurements was required to perform our studies, we created awareness among the RIPE NCC team for the measurements we planned to conduct and asked them for approval before execution started.

To allow operators to opt out of our measurements, we inserted a link to our website explaining the workings of our measurements with contact information. We did not receive any such requests. Moreover, the size of the packets was chosen as small as possible, and the frequency was set to an acceptable amount to avoid overloading any infrastructure components.

Whenever we used third-party services in a large-scale, automated fashion, e.g., for mapping the IP to Autonomous System Numbers (ASNs) [251], we used rate limiting to avoid overloading the external services.

# Lists of Acronyms

| | |
|---|---|
| **API** | Application Programming Interface. |
| **APNIC** | Asia-Pacific Network Information Centre. |
| **ARIN** | American Registry for Internet Numbers. |
| **AS** | Autonomous System. |
| **ASN** | Autonomous System Number. |
| **ASPA** | Autonomous System Provider Authorization. |
| **BGP** | Border Gateway Protocol. |
| **BGPsec** | Border Gateway Protocol Security. |
| **BSD** | Berkeley Software Distribution. |
| **BSI** | Bundesamt für Sicherheit in der Informationstechnik. |
| **CA** | Certificate Authority. |
| **CAIDA** | Center for Applied Internet Data Analysis. |
| **CAS** | Customer Autonomous System. |
| **CDN** | Content Delivery Network. |
| **CLI** | Command Line Interface. |
| **CPU** | Central Processing Unit. |
| **CRL** | Certificate Revocation List. |
| **CVD** | Coordinated Vulnerability Disclosure. |
| **DDoS** | Distributed Denial of Service. |
| **DFN** | Deutsches Forschungsnetz. |
| **DFZ** | Default Free Zone. |

**DHCP** Dynamic Host Configuration Protocol.

**DNS** Domain Name System.

**DNSSEC** DNS Security Extensions.

**DO** Down Only.

**DoS** Denial of Service.

**eBGP** external Border Gateway Protocol.

**EE** End-Entity.

**EGP** Exterior Gateway Protocol.

**FCC** Federal Communications Commission.

**FSM** Finite State Machine.

**HDD** Hard Disk Drive.

**HTTP** Hypertext Transfer Protocol.

**HTTPS** Hypertext Transfer Protocol Secure.

**IAB** Internet Architecture Board.

**IANA** Internet Assigned Numbers Authority.

**iBGP** internal Border Gateway Protocol.

**ICANN** Internet Corporation for Assigned Names and Numbers.

**ICMP** Internet Control Message Protocol.

**IETF** Internet Engineering Task Force.

**IGP** Interior Gateway Protocol.

**IGRP** Interior Gateway Routing Protocol.

**IHR** Internet Health Report.

**IIJ** Internet Initiative Japan.

**IoT** Internet of Things.

**IP** Internet Protocol.

**IPSec** Internet Protocol Security.

**IRR** Internet Routing Registry.

**IS-IS** Intermediate System to Intermediate System.

**ISC** Internet Systems Consortium.

**ISP** Internet Service Provider.

**IXP** Internet Exchange Point.

**KVM** Kernel-based Virtual Machine.

**LACNIC** Latin America and Caribbean Network Information Centre.

**LAN** Local Area Network.

**LIR** Local Internet Registry.

**LoA** Letter of Authorization.

**MANRS** Mutually Agreed Norms for Routing Security.

**MD5** Message Digest Algorithm 5.

**MIT** Massachusetts Institute of Technology.

**MPLS** Multiprotocol Label Switching.

**MRT** Multi-Threaded Routing Toolkit.

**NAT** Network Address Translation.

**NCC** Network Coordination Centre.

**NIR** National Internet Registry.

**NIST** National Institute of Standards and Technology.

**NLnog** Netherlands Network Operator Group.

**NLRI** Network Layer Reachability Information.

**NOC** Network Operation Center.

**OpenVPN** Open Virtual Private Network.

**OSPF** Open Shortest Path First.

**OTC** Only To Customer.

|  |  |
|---|---|
| **PAS** | Provider Autonomous System. |
| **PCH** | Packet Clearing House. |
| **PDU** | Protocol Data Unit. |
| **PKI** | Public Key Infrastructure. |
| **PoP** | Point of Presence. |
| **psBGP** | Pretty Secure BGP. |
| **RADB** | Routing Assets Database. |
| **RAM** | Random-Access Memory. |
| **RC** | Resource Certificate. |
| **RFC** | Request for Comments. |
| **RFD** | Route Flap Damping. |
| **RIB** | Route Information Base. |
| **RIP** | Routing Information Protocol. |
| **RIPE** | Réseaux IP Européens. |
| **RIR** | Regional Internet Registry. |
| **RIS** | Routing Information Service. |
| **ROA** | Route Origin Authorization. |
| **ROV** | Route Origin Validation. |
| **RP** | Relying Party. |
| **RPF** | Reverse Path Filtering. |
| **RPKI** | Resource Public Key Infrastructure. |
| **RRDP** | RPKI Repository Delta Protocol. |
| **RTR** | RPKI to Router. |
| **S-BGP** | Secure BGP. |
| **SD** | Secure Digital. |
| **SDN** | Software Defined Networking. |
| **SIDR** | Secure Inter-Domain Routing. |

**SIDROPS** SIDR Operations.

**SKI** Subject Key Identifier.

**soBGP** Secure Origin BGP.

**SRI** Stanford Research Institute.

**SRx** Secure Routing Extension.

**SSD** Solid State Disc.

**SSH** Secure Shell.

**SSL** Secure Sockets Layer.

**STP** Spanning Tree Protocol.

**SVGD** Stein Variational Gradient Descent.

**TA** Trust Anchor.

**TCP** Transmission Control Protocol.

**TLS** Transport Layer Security.

**TTL** Time To Live.

**UCLA** University of California, Los Angeles.

**UCSB** University of California, Santa Barbara.

**UDP** User Datagram Protocol.

**VM** Virtual Machine.

**VPN** Virtual Private Network.

**VRP** Validated ROA Payload.

**XML** Extensible Markup Language.

# List of Listings

# List of Figures

# List of Tables

# Bibliography

[1] K. Butler, T. R. Farley, P. McDaniel and J. Rexford, "A Survey of BGP Security Issues and Solutions", *Proceedings of the IEEE*, vol. 98, no. 1, pp. 100–122, 2009. DOI: 10.1109/JPROC.2009.2034031.

[2] G. Huston, M. Rossi and G. Armitage, "Securing BGP — A Literature Survey", *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 199–222, 2010. DOI: 10.1109/SURV.2011.041010.00041.

[3] RIPE NCC, *YouTube Hijacking*, 2009. [Online]. Available: http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study.

[4] A. Siddiqui, *KlaySwap – Another BGP Hijack Targeting Crypto Wallets*, 2022. [Online]. Available: https://www.manrs.org/2022/02/klayswap-another-bgp-hijack-targeting-crypto-wallets/.

[5] A. Siddiqui, *For 12 Hours, Was Part of Apple Engineering's Network Hijacked by Russia's Rostelecom?s*, 2022. [Online]. Available: https://www.manrs.org/2022/07/for-12-hours-was-part-of-apple-engineerings-network-hijacked-by-russias-rostelecom/.

[6] D. Madory, *A Brief History of the Internet's Biggest BGP Incidents*, 2023. [Online]. Available: https://www.kentik.com/blog/a-brief-history-of-the-internets-biggest-bgp-incidents/.

[7] European Commission, *The EU's Cybersecurity Strategy for the Digital Decade*, 2021. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52021IP0286.

[8] D. Goodin, *Russian-controlled telecom hijacks financial services' Internet traffic*, 2017. [Online]. Available: https://arstechnica.com/information-technology/2017/04/russian-controlled-telecom-hijacks-financial-services-internet-traffic/.

[9] Federal Communications Commission, *NOTICE OF INQUIRY: Secure Internet Routing*, 2022. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-22-18A1.pdf.

[10] J. Schlamp, T. C. Schmidt and M. Wählisch, *Zweite Internet Backbone-Studie: Auslandskabelverbindungen und CDN-Kompetenz (ZwIBACK) - Infrastruktur, Ausfallszenarien und Konsolidierung*, 2022. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/ZwiBACK/ZwiBACK-Studie.pdf?__blob=publicationFile&v=2.

[11] M. Lepinski and K. Sriram, *BGPsec Protocol Specification*, RFC 8205, Sep. 2017. DOI: 10.17487/RFC8205. [Online]. Available: https://rfc-editor.org/rfc/rfc8205.txt.

[12] N. Rodday, I. Cunha, R. Bush, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt and M. Wählisch, "The Resource Public Key Infrastructure (RPKI): A Survey on Measurements and Future Prospects", in *Transactions on Network and Service Management (TNSM)*, [Accepted for publication], IEEE, 2023.

[13] R. Bush, J. Borkenhagen, T. Bruijnzeels and J. Snijders, "Timing Parameters in the RPKI based Route Origin Validation Supply Chain", Internet Engineering Task Force, Internet-Draft draft-ietf-sidrops-rpki-rov-timing-06, Feb. 2022, Work in Progress, 10 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidrops-rpki-rov-timing/06/.

[14] N. Rodday, I. Cunha, R. Bush, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt and M. Wählisch, "Revisiting RPKI Route Origin Validation on the Data Plane", in *Proceedings of Network Traffic Measurement and Analysis Conference (TMA), IFIP*, Virtual: ACM, 2021.

[15] N. Rodday, R. Bush, I. Cunha, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt and M. Wählisch, "Poster: Extending RPKI ROV Measurement Coverage", in *Presented at ACM Internet Measurement Conference (IMC)*, Amsterdam, The Netherlands: ACM, 2019.

[16] P. H. Friedemann, N. Rodday and G. D. Rodosek, "Assessing the RPKI Validator Ecosystem", in *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, Barcelona, Spain: IEEE, 2022, pp. 295–300. DOI: 10.1109/ICUFN55119.2022.9829712.

[17] N. Rodday, L. Kaltenbach, I. Cunha, R. Bush, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt and M. Wählisch, "On the Deployment of Default Routes in Inter-domain Routing", in *Proceedings of the ACM SIGCOMM 2021 Workshop on Technologies, Applications, and Uses of a Responsible Internet*, Virtual: ACM, 2021, pp. 14–20. DOI: 10.1145/3472951.3473505.

[18] N. Rodday, G. D. Rodosek, A. Pras and R. van Rijswijk-Deij, "Exploring the Benefit of Path Plausibility Algorithms in BGP", in *IEEE/IFIP Network Operations and Management Symposium*, Acceped for publication, Seoul, South Korea: IEEE, 2024.

[19] N. Rodday and G. D. Rodosek, "BGPEval: Automating Large-Scale Testbed Creation", in *19th International Conference on Network and Service Management, CNSM 2023, Niagara Falls, ON, Canada, October 30 - Nov. 2, 2023*, IEEE, 2023, pp. 1–5. DOI: 10.23919/CNSM59352.2023.10327905.

[20] N. Rodday, R. van Baaren, L. Hendriks, R. van Rijswijk-Deij, A. Pras and G. Dreo, "Evaluating RPKI ROV Identification Methodologies in Automatically Generated Mininet Topologies", in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, Barcelona, Spain: ACM, 2020, pp. 530–531. DOI: 10.1145/3386367.3431669.

[21] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. B. Postel, L. G. Roberts and S. S. Wolff, "A Brief History of the Internet", *Compututer Communication Review*, vol. 39, no. 5, pp. 22–31, 2009. DOI: 10.1145/1629607.1629613.

[22] J. C. Licklider and R. W. Taylor, "The Computer as a Communication Device", *Science and technology*, vol. 76, no. 2, pp. 1–3, 1968.

[23] M. Townes, "The Spread of TCP/IP: How the Internet Became the Internet", *Millennium*, vol. 41, no. 1, pp. 43–64, 2012.

[24] A. S. Tanenbaum and D. Wetherall, *Computer Networks, 5th Edition.* Pearson, 2011, ISBN: 0132553171.

[25] M. K. Lottor, *Internet Growth (1981-1991)*, RFC 1296, Jan. 1992. DOI: 10.17487/RFC1296. [Online]. Available: https://www.rfc-editor.org/info/rfc1296.

[26] Statista, *Number of Internet and Social Media Users Worldwide as of January 2023*, 2023. [Online]. Available: https://www.statista.com/statistics/617136/digital-population-worldwide/.

[27] J. A. Hawkinson and T. J. Bates, *Guidelines for creation, selection, and registration of an Autonomous System (AS)*, RFC 1930, Mar. 1996. DOI: 10.17487/RFC1930. [Online]. Available: https://www.rfc-editor.org/info/rfc1930.

[28] Y. Rekhter, T. Li and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC Editor, RFC 4271, Jan. 2006. DOI: 10.17487/RFC4271. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4271.txt.

[29] D. Mills, "Exterior Gateway Protocol Formal Specification", RFC Editor, RFC 904, Apr. 1984. DOI: 10.17487/RFC0904. [Online]. Available: https://www.rfc-editor.org/info/rfc904.

[30] International Organization for Standardization, *ISO/IEC 10747: Protocol for Exchange of Inter-domain Routing Information among Intermediate Systems to Support Forwarding of ISO 8473 PDUs*, 1994.

[31] E. Chen and J. Yuan, "Autonomous-System-Wide Unique BGP Identifier for BGP-4", RFC Editor, RFC 6286, Jun. 2011. DOI: 10.17487/RFC6286. [Online]. Available: https://www.rfc-editor.org/info/rfc6286.

[32] J. Dong, M. Chen and A. Suryanarayana, "Subcodes for BGP Finite State Machine Error", RFC Editor, RFC 6608, May 2012. DOI: 10.17487/RFC6608. [Online]. Available: https://www.rfc-editor.org/info/rfc6608.

[33] Q. Vohra and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC Editor, RFC 6793, Dec. 2012. DOI: 10.17487/RFC6793. [Online]. Available: https://www.rfc-editor.org/info/rfc6793.

[34] E. Chen, J. Scudder, P. Mohapatra and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC Editor, RFC 7606, Aug. 2015. DOI: 10.17487/RFC7606. [Online]. Available: https://www.rfc-editor.org/info/rfc7606.

[35] W. Kumari, R. Bush, H. Schiller and K. Patel, "Codification of AS 0 Processing", RFC Editor, RFC 7607, Aug. 2015. DOI: 10.17487/RFC7607. [Online]. Available: https://www.rfc-editor.org/info/rfc7607.

[36] W. George and S. Amante, "Autonomous System Migration Mechanisms and Their Effects on the BGP AS_PATH Attribute", RFC Editor, RFC 7705, Nov. 2015. DOI: 10.17487/RFC7705. [Online]. Available: https://www.rfc-editor.org/info/rfc7705.

[37] J. Mauch, J. Snijders and G. Hankins, "Default External BGP (EBGP) Route Propagation Behavior without Policies", RFC Editor, RFC 8212, Jul. 2017. DOI: 10.17487/RFC8212. [Online]. Available: https://www.rfc-editor.org/info/rfc8212.

[38] R. Bush, K. Patel and D. Ward, "Extended Message Support for BGP", RFC Editor, RFC 8654, Oct. 2019. DOI: 10.17487/RFC8654. [Online]. Available: https://www.rfc-editor.org/info/rfc8654.

[39] E. Chen and J. Scudder, "Extended Optional Parameters Length for BGP OPEN Message", RFC Editor, RFC 9072, Jul. 2021. DOI: 10.17487/RFC9072. [Online]. Available: https://www.rfc-editor.org/info/rfc9072.

[40] R. Chandra, P. Traina and T. Li, "BGP Communities Attribute", RFC Editor, RFC 1997, Aug. 1996. DOI: 10.17487/RFC1997. [Online]. Available: https://www.rfc-editor.org/info/rfc1997.

[41] J. Heitz, J. Snijders, K. Patel, I. Bagdonas and N. Hilliard, "BGP Large Communities Attribute", RFC Editor, RFC 8092, Feb. 2017. DOI: 10.17487/RFC8092. [Online]. Available: https://www.rfc-editor.org/info/rfc8092.

[42] T. King, C. Dietzel, J. Snijders, G. Doering and G. Hankins, "BLACKHOLE Community", RFC Editor, RFC 7999, Oct. 2016. DOI: 10.17487/RFC7999. [Online]. Available: https://www.rfc-editor.org/info/rfc7999.

[43] Internet Assigned Numbers Authority, *BGP Communities Assignment.* [Online]. Available: https://www.iana.org/assignments/bgp-well-known-communities/bgp-well-known-communities.xhtml.

[44] L. Gao and J. Rexford, "Stable Internet Routing Without Global Coordination", in *Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, Santa Clara, California, USA: ACM, 2000, pp. 307–317. DOI: 10.1145/339331.339426.

[45] K. Varadhan, R. Govindan and D. Estrin, "Persistent Route Oscillations in Inter-Domain Routing", *Computer networks*, vol. 32, no. 1, pp. 1–16, 2000, ISSN: 1389-1286. DOI: 10.1016/S1389-1286(99)00108-5.

[46] T. G. Griffin and G. Wilfong, "An Analysis of BGP Convergence Properties", *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 277–288, 1999. DOI: 10.1145/316188.316231.

[47] Center for Applied Internet Data Analysis (CAIDA), *Geographic breakdown of the Regional Internet Registries (RIR) and National Internet Registries (NIR)*, 2020. [Online]. Available: https://www.caida.org/archive/as2org/#H2630.

[48] Y. Wang and K. Zhang, "Quantifying the Flattening of Internet Topology", in *Proceedings of the 11th International Conference on Future Internet Technologies, CFI 2016, Nanjing, China, June 15-17, 2016*, ACM, 2016, pp. 113–117. DOI: 10.1145/2935663.2935682.

[49] P. Gill, M. F. Arlitt, Z. Li and A. Mahanti, "The Flattening Internet Topology: Natural Evolution, Unsightly Barnacles or Contrived Collapse?", in *Passive and Active Network Measurement, 9th International Conference, PAM 2008, Cleveland, OH, USA, April 29-30, 2008. Proceedings*, M. Claypool and S. Uhlig, Eds., ser. Lecture Notes in Computer Science, vol. 4979, Springer, 2008, pp. 1–10. DOI: 10.1007/978-3-540-79232-1_1.

[50] R. Bush, O. Maennel, M. Roughan and S. Uhlig, "Internet Optometry: Assessing the Broken Glasses in Internet Reachability", in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, ACM, New York, NY, USA: ACM, 2009, pp. 242–253. DOI: 10.1145/1644893.1644923.

[51] S. Frankel and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC Editor, RFC 6071, Feb. 2011. DOI: 10.17487/RFC6071. [Online]. Available: https://www.rfc-editor.org/info/rfc6071.

[52] A. Heffernan, "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC Editor, RFC 2385, Aug. 1998. DOI: 10.17487/RFC2385. [Online]. Available: https://www.rfc-editor.org/info/rfc2385.

[53] MANRS, *Mutually Agreed Norms for Routing Security*, 2022. [Online]. Available: https://www.manrs.org/.

[54] Center for Applied Internet Data Analysis (CAIDA). "The CAIDA AS Relationships Dataset, 20210401.as-rel2.txt". (2021), [Online]. Available: https://publicdata.caida.org/datasets/as-relationships/serial-2/.

[55] NLNOG Stichting, *BGP Filter Guide*, 2023. [Online]. Available: https://bgpfilterguide.nlnog.net/guides/bogon_prefixes/#filter-bogon-prefixes.

[56] M. Kühne, *Interesting Graph - AS Path Lengths Over Time*, 2012. [Online]. Available: https://labs.ripe.net/author/mirjam/update-on-as-path-lengths-over-time/.

[57] P. Smith, R. Evans and M. Hughes, *RIPE-399 - RIPE Routing Working Group Recommendations on Route Aggregation*, 2006. [Online]. Available: https://www.ripe.net/publications/docs/ripe-399.

[58] P. Smith and R. Evans, *RIPE-532 - RIPE Routing Working Group Recommendations on IPv6 Route Aggregation*, 2011. [Online]. Available: https://www.ripe.net/publications/docs/ripe-532.

[59] M. Nawrocki, J. Blendin, C. Dietzel, T. C. Schmidt and M. Wählisch, "Down the Black Hole: Dismantling Operational Practices of BGP Blackholing at IXPs", in *Proceedings of the Internet Measurement Conference*, ser. IMC '19, Amsterdam, Netherlands: Association for Computing Machinery, 2019, pp. 435–448, ISBN: 9781450369480. DOI: 10.1145/3355369.3355593.

[60] P.-A. Vervier, O. Thonnard and M. Dacier, "Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks.", in *Network and Distributed System Security Symposium (NDSS)*, San Diego, California, USA: The Internet Society, 2015.

[61] Merit Network Inc., *Overview of the IRR*, 2018. [Online]. Available: https://www.irr.net/docs/overview.html.

[62] G. D. Battista, T. Refice and M. Rimondini, "How to Extract BGP Peering Information from the Internet Routing Registry", in *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, Pisa, Italy: ACM, 2006, pp. 317–322. DOI: 10.1145/1162678.1162685.

[63] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle and E. W. Biersack, "HEAP: Reliable Assessment of BGP Hijacking Attacks", *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1849–1861, 2016. DOI: 10.1109/JSAC.2016.2558978.

[64] X. Shi, Y. Xiang, Z. Wang, X. Yin and J. Wu, "Detecting Prefix Hijackings in the Internet with Argus", in *Proceedings of the 2012 Internet Measurement Conference*, Boston, Massachusetts, USA: ACM, 2012, pp. 15–28. DOI: 10.1145/2398776.2398779.

[65] F. Wang and L. Gao, "On Inferring and Characterizing Internet Routing Policies", *Journal of Communications and Networks*, vol. 9, no. 4, pp. 350–355, 2007. DOI: 10.1145/948205.948208.

[66] B. Du, G. Akiwate, T. Krenc, C. Testart, A. Marder, B. Huffaker, A. C. Snoeren and K. Claffy, "IRR Hygiene in the RPKI Era", in *International Conference on Passive and Active Network Measurement*, Virtual: Springer, 2022, pp. 321–337. DOI: 10.1007/978-3-030-98785-5_14.

[67] A. Khan, H.-c. Kim, T. Kwon and Y. Choi, "A Comparative Study on IP Prefixes and their Origin ASes in BGP and the IRR", *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 16–24, 2013. DOI: 10.1145/2500098.2500101.

[68] W. Aiello, J. Ioannidis and P. McDaniel, "Origin Authentication in Interdomain Routing", in *Proceedings of the 10th ACM conference on Computer and Communications Security*, Washington D.C., USA: ACM, 2003, pp. 165–178. DOI: 10.1145/948109.948133.

[69] E. Osterweil, S. Amante, D. Massey and D. McPherson, "The Great IPv4 Land Grab: Resource Certification for the IPv4 Grey Market", in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, Cambridge, MA, USA: ACM, 2011, pp. 1–6. DOI: 10.1145/2070562.2070574.

[70] M. Lepinski and S. Kent, *An Infrastructure to Support Secure Internet Routing*, RFC 6480, Feb. 2012. DOI: 10.17487/RFC6480. [Online]. Available: https://rfc-editor.org/rfc/rfc6480.txt.

[71] The Number Resource Organization Executive Council, *Meeting of the NRO Executive Council - 101221*, 2021. [Online]. Available: https://www.nro.net/meeting-of-the-nro-executive-council-101221/.

[72] T. Bruijnzeels, O. Muravskiy and B. Weber, "RPKI Repository Analysis and Requirements", IETF Secretariat, Internet-Draft draft-tbruijnzeels-sidr-repo-analysis-00, Feb. 2013. [Online]. Available: https://www.ietf.org/archive/id/draft-tbruijnzeels-sidr-repo-analysis-00.txt.

[73] E. Osterweil, T. Manderson, R. White and D. McPherson, "Sizing Estimates for a Fully Deployed RPKI", *Verisign Labs, TR*, vol. 1120005, 2012.

[74] Internet Architecture Board, *IAB statement on the RPKI*, 2010. [Online]. Available: https://www.iab.org/documents/correspondence-reports-documents/docs2010/iab-statement-on-the-rpki/.

[75] Internet Architecture Board, *IAB statement on the RPKI*, 2018. [Online]. Available: https://www.iab.org/documents/correspondence-reports-documents/2018-2/iab-statement-on-the-rpki/.

[76] A. Dalskov, C. Orlandi, M. Keller, K. Shrishak and H. Shulman, "Securing DNSSEC Keys via Threshold ECDSA From Generic MPC", in *European Symposium on Research in Computer Security*, Guildford, United Kingdom: Springer, 2020, pp. 654–673. DOI: 10.1007/978-3-030-59013-0_32.

[77] G. Huston and G. Michaelson, "Validation of Route Origination using the Resource Certificate PKI and ROAs", RFC Editor, RFC 6483, Feb. 2012. DOI: 10.17487/RFC6483. [Online]. Available: https://www.rfc-editor.org/info/rfc6483.

[78] G. Huston, G. Michaelson and R. Loomans, "A Profile for X.509 PKIX Resource Certificates, howpublished = Internet Requests for Comments", RFC Editor, RFC 6487, Feb. 2012. DOI: 10.17487/RFC6487. [Online]. Available: https://www.rfc-editor.org/info/rfc6487.

[79] Y. Gilad, T. Hlavacek, A. Herzberg, M. Schapira and H. Shulman, "Perfect is the Enemy of Good: Setting Realistic Goals for BGP Security", in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, Redmond, WA, USA: ACM, 2018, pp. 57–63. DOI: 10.1145/3286062.3286071.

[80] T. Hlavacek, I. Cunha, Y. Gilad, A. Herzberg, E. Katz-Bassett, M. Schapira and H. Shulman, "DISCO: Sidestepping RPKI's Deployment Barriers", in *Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA: The Internet Society, 2020. DOI: 10.14722/ndss.2020.24355.

[81] F. V. Silveira, *Routing WG Mailinglist - Publish in Parent - input requested*, 2022. [Online]. Available: https://www.ripe.net/ripe/mail/archives/routing-wg/2022-September/004613.html.

[82] NLNetLabs, *Krill 0.11.0*, 2022. [Online]. Available: https://krill.docs.nlnetlabs.nl/en/stable/.

[83] Dragon Research Labs, *RPKI Toolkit*, 2022. [Online]. Available: https://github.com/dragonresearch/rpki.net.

[84] S. Qing and D. Ma, *RPSTIR2*, bgpsecurity, 2020. [Online]. Available: https://github.com/bgpsecurity/rpstir2.

[85] L. Poinsignon, M. Chris and J. Bampton, *OctoRPKI*, Cloudflare, 2019. [Online]. Available: https://github.com/cloudflare/cfrpki.

[86] NLnet Labs, *Routinator Manual*, 2021. [Online]. Available: https://routinator.docs.nlnetlabs.nl/en/stable/.

[87] LACNIC and NIC.MX, *FORT Validator - Github Repository*, 2021. [Online]. Available: https://nicmx.github.io/FORT-validator/.

[88] K. Dzonsons, C. Jeker, J. Snijders, T. de Raadt, S. Benoit and T. Buehler, *rpki-client*, OpenBSD, 2021. [Online]. Available: https://www.rpki-client.org/.

[89] M. Puzanov, *rpki-prover*, 2020. [Online]. Available: https://github.com/lolepezy/rpki-prover.

[90] N. Trenaman, *Lifecycle of the RIPE NCC RPKI Validator*, RIPE NCC, Oct. 2020. [Online]. Available: https://labs.ripe.net/author/nathalie_nathalie/lifecycle-of-the-ripe-ncc-rpki-validator/.

[91] R. Austein, R. Bush *et al.*, *Dragon Research Labs RPKI Toolkit*, 2006. [Online]. Available: https://github.com/dragonresearch/rpki.net.

[92] B. Maddison, *RPKImancer*, 2022. [Online]. Available: https://github.com/benmaddison/rpkimancer.

[93] A. Reuter, M. Wählisch and T. C. Schmidt, “RPKI MIRO: Monitoring and Inspection of RPKI Objects”, *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 107–108, 2015. DOI: 10.1145/2785956.2790026.

[94] ZDNS, *RPKIVIZ*, 2022. [Online]. Available: http://rpkiviz.zdns.cn/.

[95] T. Bruijnzeels, R. Bush and G. Michaelson, “Resource Public Key Infrastructure (RPKI) Repository Requirements”, IETF Secretariat, Internet-Draft draft-ietf-sidrops-deprecate-rsync-00, Aug. 2020. [Online]. Available: https://www.ietf.org/archive/id/draft-ietf-sidrops-deprecate-rsync-00.txt.

[96] T. Bruijnzeels, O. Muravskiy, B. Weber and R. Austein, “The RPKI Repository Delta Protocol (RRDP)”, RFC Editor, RFC 8182, Jul. 2017. DOI: 10.17487/RFC8182. [Online]. Available: https://www.rfc-editor.org/info/rfc8182.

[97] R. Bush and R. Austein, “The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1”, RFC Editor, RFC 8210, Sep. 2017. DOI: 10.17487/RFC8210. [Online]. Available: https://www.rfc-editor.org/info/rfc8210.

[98] M. Wählisch, F. Holler, T. C. Schmidt and J. H. Schiller, “RTRlib: An Open-Source Library in C for RPKI-based Prefix Origin Validation”, in *6th Workshop on Cyber Security Experimentation and Test (CSET 13)*, Washington, D.C., USA: Usenix Association, 2013.

[99] M. Wählisch and others, *rtrlib - An open-source C implementation of the RPKI/Router Protocol client*, 2022. [Online]. Available: https://github.com/rtrlib/rtrlib.

[100] Multiple authors, *StayRTR*, 2018. [Online]. Available: https://github.com/bgp/stayrtr.

[101] M. Hoffmann, A. Band *et al.*, *RTRTR – An RPKI data proxy*, GitHub, NLnetLabs, 2020. [Online]. Available: https://github.com/NLnetLabs/rtrtr.

[102] N. Akiya, G. Swallow, S. Litkowski, B. Decraene, J. Drake and M. Chen, “Label Switched Path (LSP) Ping and Traceroute Multipath Support for Link Aggregation Group (LAG) Interfaces”, RFC Editor, RFC 8611, Jun. 2019. DOI: 10.17487/RFC8611. [Online]. Available: https://www.rfc-editor.org/info/rfc8611.

[103] A. Malhotra and S. Goldberg, “RPKI vs ROVER: Comparing the Risks of BGP Security Solutions”, 4, vol. 44, ACM New York, NY, USA, 2014, pp. 113–114. DOI: 10.1145/2619239.2631435.

[104] J. Gersch and D. Massey, “ROVER: Route Origin Verification Using DNS”, in *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, Nassau, Bahamas: IEEE, 2013, pp. 1–9. DOI: 10.1109/ICCCN.2013.6614187.

[105] J. Gersch, D. Massey, C. Olschanowsky and L. Zhang, “DNS Resource Records for Authorized Routing Information”, IETF Secretariat, Internet-Draft draft-gersch-grow-revdns-bgp-02, Feb. 2013. [Online]. Available: http://www.ietf.org/internet-drafts/draft-gersch-grow-revdns-bgp-02.txt.

[106] G. Huston, *A Reappraisal of Validation in the RPKI*, 2014. [Online]. Available: https://www.dotnxdomain.net/ispcol/2014-04/rpkiv.pdf.

[107] B. Du, C. Testart, R. Fontugne, G. Akiwate, A. C. Snoeren *et al.*, “Mind Your MANRS: Measuring the MANRS Ecosystem”, in *Proceedings of the 22nd ACM Internet Measurement Conference (IMC’22)*, Nice, France: ACM, 2022, pp. 716–729. DOI: 10.1145/3517745.3561419.

[108] Internet Initiative Japan (IIJ) Research Labs, *Internet Health Report*, 2022. [Online]. Available: https://ihr.iijlab.net/ihr/en-us/rov.

[109] R. K. C. Chang and M. Lo, “Inbound Traffic Engineering for Multihomed ASs Using AS Path Prepending”, *IEEE Network*, vol. 19, no. 2, pp. 18–25, 2005. DOI: 10.1109/MNET.2005.1407694. [Online]. Available: https://doi.org/10.1109/MNET.2005.1407694.

[110] P. de B. Marcos, L. Prehn, L. Leal, A. Dainotti, A. Feldmann and M. P. Barcellos, “AS-Path Prepending: There is no Rose Without a Thorn”, in *IMC ’20: ACM Internet Measurement Conference, Virtual Event, USA, October 27-29, 2020*, ACM, 2020, pp. 506–520. DOI: 10.1145/3419394.3423642.

[111] E. Zmijewski, *Longer is not always better*, 2009. [Online]. Available: https://web.archive.org/web/20181227050443/https://dyn.com/blog/longer-is-not-better/.

[112] M. McBride, D. Madory, J. Tantsura, R. Raszuk, H. Li, J. Heitz and G. Mishra, “AS Path Prepending”, Internet Engineering Task Force, Internet-Draft draft-ietf-grow-as-path-prepending-06, Feb. 2022, Work in Progress, 12 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-grow-as-path-prepending/06/.

[113] S. Miao, *Yet another BGP hijacking towards AS16509*, 2022. [Online]. Available: https://mailman.nanog.org/pipermail/nanog/2022-August/220320.html.

[114] D. Madory, *Tweet on AS16509 hijack*, 2022. [Online]. Available: https://twitter.com/DougMadory/status/1562089866321698819.

[115] S. Kent, C. Lynn and K. Seo, “Secure Border Gateway Protocol (S-BGP)”, *IEEE Journal on Selected areas in Communications*, vol. 18, no. 4, pp. 582–592, 2000. DOI: 10.1109/49.839934.

[116] K. Seo, C. Lynn and S. Kent, “Public-key Infrastructure for the Secure Border Gateway Protocol (S-BGP)”, in *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX’01*, IEEE, vol. 1, 2001, pp. 239–253. DOI: 10.1109/DISCEX.2001.932219.

[117] T. Wan, E. Kranakis and P. C. van Oorschot, “Pretty Secure BGP, psBGP.”, in *Network and Distributed System Security Symposium (NDSS)*, San Diego, California, USA: The Internet Society, 2005.

[118] S. T. Kent, C. Lynn, J. Mikkelson and K. Seo, “Secure Border Gateway Protocol (S-BGP)-Real World Performance and Deployment Issues.”, in *NDSS*, 2000.

[119] D. M. Nicol, S. W. Smith and M. Zhao, “Evaluation of efficient security for bgp route announcements using parallel simulation”, *Simulation Modelling Practice and Theory*, vol. 12, no. 3-4, pp. 187–216, 2004. DOI: 10.1016/J.SIMPAT.2003.10.003.

[120] M. Lepinski, “BGPSEC Protocol Specification”, Internet Engineering Task Force, Internet-Draft draft-ietf-sidr-bgpsec-protocol-01, 2011, Work in Progress, 28 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidr-bgpsec-protocol/01/.

[121] W. George and S. L. Murphy, *BGPsec Considerations for Autonomous System (AS) Migration*, RFC 8206, Sep. 2017. DOI: 10.17487/RFC8206. [Online]. Available: https://www.rfc-editor.org/info/rfc8206.

[122] R. Bush, *BGPsec Operational Considerations*, RFC 8207, Sep. 2017. DOI: 10.17487/RFC8207. [Online]. Available: https://www.rfc-editor.org/info/rfc8207.

[123] S. Turner and O. Borchert, *BGPsec Algorithms, Key Formats, and Signature Formats*, RFC 8208, Sep. 2017. DOI: 10.17487/RFC8208. [Online]. Available: https://www.rfc-editor.org/info/rfc8208.

[124] M. Reynolds, S. Turner and S. Kent, *A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests*, RFC 8209, Sep. 2017. DOI: 10.17487/RFC8209. [Online]. Available: https://www.rfc-editor.org/info/rfc8209.

[125] R. White, "Architecture and Deployment Considerations for Secure Origin BGP (soBGP)", IETF Secretariat, Internet-Draft draft-white-sobgp-architecture-02, Jun. 2006. [Online]. Available: https://www.ietf.org/archive/id/draft-white-sobgp-architecture-02.txt.

[126] P. v. Oorschot, T. Wan and E. Kranakis, "On Inter-domain Routing Security and Pretty Secure BGP (psBGP)", *ACM Transactions on Information and System Security (TISSEC)*, vol. 10, no. 3, 11–es, 2007. DOI: 10.1145/1266977.1266980.

[127] S. Kent and A. Chi, *Threat Model for BGP Path Security*, RFC 7132, Feb. 2014. DOI: 10.17487/RFC7132. [Online]. Available: https://www.rfc-editor.org/info/rfc7132.

[128] V. K. Sriram and D. Montgomery, "Design and Analysis of Optimization Algorithms to Minimize Cryptographic Processing in BGP Security Protocols", *Computer communications*, vol. 106, pp. 75–85, 2017. DOI: 10.1016/J.COMCOM.2017.03.007.

[129] E. Barker, *Digital signature standard (dss)*, en, Jul. 2013. DOI: https://doi.org/10.6028/NIST.FIPS.186-4.

[130] I. Bagdonas, *A Look at BGPsec Performance*, 2022. [Online]. Available: https://ripe84.ripe.net/archives/video/819/.

[131] R. Lychev, S. Goldberg and M. Schapira, "BGP Security in Partial Deployment: Is the Juice Worth the Squeeze?", in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, Hong Kong, China: ACM, 2013, pp. 171–182. DOI: 10.1145/2534169.2486010.

[132] I. Bagdonas, *BGPsec in the context of routing system security*, 2022. [Online]. Available: https://indico.csnog.eu/event/11/contributions/94/.

[133] K. Sriram, D. Montgomery, D. McPherson, E. Osterweil and B. Dickson, "Problem Definition and Classification of BGP Route Leaks", RFC Editor, RFC 7908, Jun. 2016.

[134] R. White, “Deployment Considerations for Secure Origin BGP (soBGP)”, Internet Engineering Task Force, Internet-Draft draft-white-sobgp-bgp-deployment-01, Jun. 2003, Work in Progress, 12 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-white-sobgp-bgp-deployment/01/.

[135] A. Azimov, E. Uskov, R. Bush, J. Snijders, R. Housley and B. Maddison, “A Profile for Autonomous System Provider Authorization”, Internet Engineering Task Force, Internet-Draft draft-ietf-sidrops-aspa-profile-16, Jul. 2023, Work in Progress, 14 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-profile/16/.

[136] A. Azimov, E. Bogomazov, R. Bush, K. Patel, J. Snijders and K. Sriram, “BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects”, Internet Engineering Task Force, Internet-Draft draft-ietf-sidrops-aspa-verification-16, Aug. 2023, Work in Progress, 23 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/16/.

[137] M. Lepinski, A. Chi and S. Kent, “Signed object template for the resource public key infrastructure (rpki)”, RFC Editor, RFC 6488, Feb. 2012. DOI: 10.17487/RFC6488. [Online]. Available: https://www.rfc-editor.org/info/rfc6488.

[138] Stichting NLnet Labs, *Routinator - Open Source RPKI Relying Party software*, 2023. [Online]. Available: https://www.nlnetlabs.nl/projects/rpki/routinator.

[139] J. Snijders, M. Stucchi and M. Aelmans, “RPKI Autonomous Systems Cones: A Profile To Define Sets of Autonomous Systems Numbers To Facilitate BGP Filtering”, Internet Engineering Task Force, Internet-Draft draft-ietf-grow-rpki-as-cones-02, Apr. 2020, Work in Progress, 10 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-grow-rpki-as-cones-02.

[140] RouteViews Project, *University of oregon routeviews project*, 2013. [Online]. Available: http://www.routeviews.org.

[141] Center for Applied Internet Data Analysis (CAIDA), *The CAIDA AS Relationships Dataset, 20221001*, 2022. [Online]. Available: https://www.caida.org/catalog/datasets/as-relationships/.

[142] J. Snijders and F. Korsbaeck, “A Profile for RPKI Signed Groupings of Autonomous System Numbers (ASGroup)”, IETF Secretariat, Internet-Draft draft-spaghetti-sidrops-rpki-asgroup-00, Nov. 2022. [Online]. Available: https://datatracker.ietf.org/api/v1/doc/document/draft-spaghetti-sidrops-rpki-asgroup/.

[143] J. Mitchell, “Autonomous System (AS) Reservation for Private Use”, RFC Editor, RFC 6996, Jul. 2013. DOI: 10.17487/RFC6996. [Online]. Available: https://www.rfc-editor.org/info/rfc6996.

[144] Unknown author, *BGP Looking Glasses*. [Online]. Available: https://www.bgplookingglass.com/.

[145] V. Giotsas, A. Dhamdhere and K. C. Claffy, "Periscope: Unifying Looking Glass Querying", in *International Conference on Passive and Active Network Measurement*, Springer, 2016, pp. 177–189. DOI: 10.1007/978-3-319-30505-9_14.

[146] RIPE NCC, *RIPE Routing Information Service (RIS)*, 2020. [Online]. Available: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

[147] Packet Clearing House, *PCH Routing Data*. [Online]. Available: https://www.pch.net/resources/BGP_Routing_Data/.

[148] P. Giardina, E. Gregori, A. Improta, A. Pischedda and L. Sani, "Isolario: a Do-ut-des Approach to Improve the Appeal of BGP Route Collecting", in *Proceedings of the workshop on Research and Applications of Internet Measurements (RAIM)*, 2015.

[149] C. Orsini, A. King, D. Giordano, V. Giotsas and A. Dainotti, "BGPStream: A Software Framework for Live and Historical BGP Data Analysis", in *Proceedings of the 2016 Internet Measurement Conference*, Santa Monica, California, USA: ACM, 2016, pp. 429–444.

[150] Center for Applied Internet Data Analysis (CAIDA), *BGP Reader*. [Online]. Available: https://bgpstream.caida.org/docs/tools/bgpreader.

[151] Center for Applied Internet Data Analysis (CAIDA), *PyBGPStream Python Wrapper*. [Online]. Available: https://bgpstream.caida.org/docs/tutorials/pybgpstream.

[152] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster and E. Katz-Bassett, "Peering: An AS for Us", in *Proc. of the 13th ACM Workshop on Hot Topics in Networks*, ACM, 2014, p. 18.

[153] The PEERING Testbed, *Peering Sessions*. [Online]. Available: https://peering.ee.columbia.edu/peers/.

[154] FriendlyARM, *NanoPI Board*. [Online]. Available: http://nanopi.io/nanopi-neo-plus2.html.

[155] RIPE NCC, *RIPE Atlas Coverage Statistics*. [Online]. Available: https://atlas.ripe.net/results/maps/network-coverage/.

[156] Y. Gilad, A. Cohen, A. Herzberg, M. Schapira and H. Shulman, "Are We There Yet? On RPKI's Deployment and Security.", in *Network and Distributed System Security Symposium (NDSS)*, San Diego, California: Internet Society, 2017.

[157] M. Wählisch, O. Maennel and T. C. Schmidt, "Towards Detecting BGP Route Hijacking Using the RPKI", *SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 103–104, Aug. 2012. DOI: 10.1145/2377677.2377702.

[158] M. Wählisch, R. Schmidt, T. C. Schmidt, O. Maennel, S. Uhlig and G. Tyson, "RiPKI: The Tragic Story of RPKI Deployment in the Web Ecosystem", in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, Philadelphia, Pennsylvania, USA: ACM, 2015, pp. 1–7. doi: 10.1145/2834050.2834102.

[159] A. Band, *Using the 'Maximum Length' Option in ROAs*, RIPE NCC, 2011. [Online]. Available: https://labs.ripe.net/author/alexband/using-the-maximum-length-option-in-roas/.

[160] T. Chung, E. Aben, T. Bruijnzeels, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, R. van Rijswijk-Deij, J. Rula and N. Sullivan, "RPKI is Coming of Age: A Longitudinal Study of RPKI Deployment and Invalid Route Origins", in *Proceedings of ACM Internet Measurement Conference (IMC)*, New York, NY, USA: ACM, 2019, pp. 406–419. doi: 10.1145/3355369.3355596.

[161] T. Hlavacek, H. Shulman and M. Waidner, "Not All Conflicts Are Created Equal: Automated Error Resolution in RPKI Deployments", in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Virtual: IEEE, 2021, pp. 1–2.

[162] Y. Li, H. Zou, Y. Chen, Y. Xu, Z. Ma, D. Ma, Y. Hu and G. Xie, "The Hanging ROA: A Secure and Scalable Encoding Scheme for Route Origin Authorization", in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, Virtual: IEEE, 2022, pp. 21–30.

[163] L. Oliver, G. Akiwate, M. Luckie, B. Du and k. claffy k, "Stop, DROP, and ROA: Effectiveness of Defenses through the lens of DROP", in *Proceedings of the 22nd ACM Internet Measurement Conference*, Nice, France: ACM, 2022.

[164] W. Li, Z. Lin, M. I. Ashiq, E. Aben, R. Fontugne, A. Phokeer and T. Chung, "RoVista: Measuring and Analyzing the Route Origin Validation (ROV) in RPKI", in *Proceedings of ACM Internet Measurement Conference (IMC)*, Montreal: ACM, 2023.

[165] A. Reuter, R. Bush, I. Cunha, E. Katz-Bassett, T. C. Schmidt and M. Wählisch, "Towards a Rigorous Methodology for Measuring Adoption of RPKI Route Validation and Filtering", *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 1, pp. 19–27, 2018.

[166] T. Hlavacek, A. Herzberg, H. Shulman and M. Waidner, "Practical Experience: Methodologies for Measuring Route Origin Validation", in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Luxembourg City, Luxembourg: IEEE, 2018, pp. 634–641.

[167] B. Cartwright-Cox, *Are BGPs security features working yet?*, 2019. [Online]. Available: https://blog.benjojo.co.uk/post/are-bgps-security-features-working-yet-rpki.

[168] G. Huston and J. Damas, *Measuring Route Origin Validation*, 2020. [Online]. Available: https://www.potaroo.net/ispcol/2020-06/rov.html.

[169] W. Chen, Z. Wang, D. Han, C. Duan, X. Yin, J. Yang and X. Shi, “ROV-MI: Large-Scale, Accurate and Efficient Measurement of ROV Deployment”, in *Network and Distributed Systems Security (NDSS) Symposium 2022*, San Diego, CA, USA: The Internet Society, 2022. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2022-214-paper.pdf.

[170] N. Künneke-Trenaman, E. Aben, J. den Hertog and J. Snijders, *RPKI Test*, 2019. [Online]. Available: https://www.ripe.net/s/rpki-test.

[171] C. Testart, P. Richter, A. King, A. Dainotti and D. Clark, “To Filter or not to Filter: Measuring the Benefits of Registering in the RPKI Today”, in *International Conference on Passive and Active Network Measurement*, Springer, 2020, pp. 71–87.

[172] T. Hlavacek, H. Shulman, N. Vogel and M. Waidner, “Keep your friends close, but your routeservers closer: Insights into rpki validation in the internet”, *arXiv preprint arXiv:2303.11772*, 2023.

[173] Cloudflare Inc., *Is BGP Safe Yet?*, 2022. [Online]. Available: https://isbgpsafe yet.com/.

[174] D. Iamartino, C. Pelsser and R. Bush, “Measuring BGP Route Origin Registration and Validation”, in *Proceedings of Passive and Active Measurement Conference (PAM)*, ser. LNCS, vol. 8995, Berlin: Springer, 2015, pp. 28–40. DOI: 10.1007/978-3-319-15509-8_3.

[175] E. Heilman, D. Cooper, L. Reyzin and S. Goldberg, “From the Consent of the Routed: Improving the Transparency of the RPKI”, in *Proceedings of the 2014 ACM conference on SIGCOMM*, Chicago, Illinois, USA: ACM, 2014, pp. 51–62. DOI: 10.1145/2740070.2626293.

[176] A. Hari and T. Lakshman, “The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet”, in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, Atlanta, GA, USA: ACM, 2016, pp. 204–210. DOI: 10.1145/3005745.3005771.

[177] D. Cooper, E. Heilman, K. Brogle, L. Reyzin and S. Goldberg, “On the Risk of Misbehaving RPKI Authorities”, in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, College Park Maryland: ACM, 2013, pp. 1–7. DOI: 10.1145/2535771.2535787.

[178] Z. Yan, G. Geng, H. Nakazato and Y.-J. Park, “Secure and Scalable Deployment of Resource Public Key Infrastructure (RPKI).”, *Journal of Internet Services and Information Security (JISIS)*, vol. 8, no. 1, pp. 31–45, 2018. DOI: 10.22667/JISIS.2018.02.28.031.

[179] K. van Hove, J. van der Ham-de Vos and R. van Rijswijk-Deij, “Rpkiller: Threat Analysis of the BGP Resource Public Key Infrastructure”, 4, vol. 4, New York, NY, USA: Association for Computing Machinery, Oct. 2023. DOI: 10.1145/3617182.

[180] J. Kristoff, R. Bush, C. Kanich, G. Michaelson, A. Phokeer, T. C. Schmidt and M. Wählisch, "On Measuring RPKI Relying Parties", in *Proceedings of the ACM Internet Measurement Conference*, New York, NY, USA: ACM, 2020, pp. 484–491. DOI: 10.1145/3419394.3423622.

[181] X. Liu, Z. Yan, G. Geng, X. Lee, S.-S. Tseng and C.-H. Ku, "RPKI Deployment: Risks and Alternative Solutions", in *Genetic and Evolutionary Computing*, Fuzhou City, Fujian Province, China: Springer, 2016, pp. 299–310. DOI: 10.1007/978-3-319-23204-1_30.

[182] K. Shrishak and H. Shulman, "Privacy Preserving and Resilient RPKI", in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, Virtual: IEEE, 2021, pp. 1–10. DOI: 10.1109/INFOCOM42981.2021.9488759.

[183] T. Hlavacek, P. Jeitner, D. Mirdita, H. Shulman and M. Waidner, "Stalloris: RPKI Downgrade Attack", in *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA: USENIX Association, Aug. 2022, pp. 4455–4471. DOI: 10.48550/ARXIV.2205.06064. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/hlavacek.

[184] R. Fontugne, A. Phokeer, C. Pelsser, K. Vermeulen and R. Bush, "RPKI Time-of-Flight: Tracking Delays in the Management, Control, and Data Planes", LNCS, 2023. DOI: 10.1007/978-3-031-28486-1_18.

[185] T. Hlavacek, P. Jeitner, D. Mirdita, H. Shulman and M. Waidner, "Behind the Scenes of RPKI", in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1413–1426. DOI: 10.1145/3548606.3560645.

[186] T. Hlavacek, P. Jeitner, D. Mirdita, H. Shulman and M. Waidner, "Beyond Limits: How to Disable Validators in Secure Networks", in *Proceedings of the ACM SIGCOMM 2023 Conference*, ser. ACM SIGCOMM '23, New York, NY, USA: Association for Computing Machinery, 2023, pp. 950–966. DOI: 10.1145/3603269.3604861.

[187] M. Fincham, *RPKI, NZNOG 2014*, 2014. [Online]. Available: https://hotplate.co.nz/archive/nznog/2014/rpki/.

[188] J. Kloots, *RPKI Routing Policy Decision-Making - a SURFnet Perspective*, 2014. [Online]. Available: https://labs.ripe.net/author/jac_kloots/rpki-routing-policy-decision-making-a-surfnet-perspective/.

[189] M. Wählisch, R. Schmidt, T. C. Schmidt, O. Maennel and S. Uhlig, "When BGP Security Meets Content Deployment: Measuring and Analysing RPKI-Protection of Websites", Technical Report, Tech. Rep., 2014.

[190] National Institute of Standards and Technology, *NIST RPKI Monitor*, 2020. [Online]. Available: https://rpki-monitor.antd.nist.gov/.

[191] The Internet Society, *MANRS ROA Stats Tool*, 2022. [Online]. Available: https://roa-stats.manrs.org/.

[192] Cloudflare Inc., *RPKI Monitor*, 2022. [Online]. Available: https://rpki.cloudflare.com/.

[193] T. Chung, E. Aben, T. Bruijnzeels, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, R. van Rijswijk-Deij, J. Rula and N. Sullivan, *Coming of Age - Website*, 2019. [Online]. Available: https://rpki-study.github.io/.

[194] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang and L. Zhang, "In Search of the Elusive Ground Truth: The Internet's AS-level Connectivity Structure", *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 1, pp. 217–228, 2008. DOI: 10.1145/1375457.1375482.

[195] Y. Gilad, O. Sagga and S. Goldberg, "MaxLength Considered Harmful to the RPKI", in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, Incheon, Republic of Korea: ACM, 2017, pp. 101–107. DOI: 10.1145/3143361.3143363.

[196] R. Bush, "Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI)", RFC Editor, RFC 7115, Jan. 2014. DOI: 10.17487/RFC7115. [Online]. Available: https://www.rfc-editor.org/info/rfc7115.

[197] T. Manderson, K. Sriram and R. White, "Use Cases and Interpretations of Resource Public Key Infrastructure (RPKI) Objects for Issuers and Relying Parties", RFC Editor, RFC 6907, Mar. 2013. DOI: 10.17487/RFC6907. [Online]. Available: https://www.rfc-editor.org/info/rfc6907.

[198] Y. Gilad, *compress roas*, 2017. [Online]. Available: https://github.com/yossigi/compress_roas.

[199] Y. Gilad, S. Goldberg, K. Sriram, J. Snijders and B. Maddison, "The Use of maxLength in the RPKI", IETF Secretariat, Internet-Draft draft-ietf-sidrops-rpkimaxlen-10, May 2022. [Online]. Available: https://www.ietf.org/archive/id/draft-ietf-sidrops-rpkimaxlen-10.txt.

[200] T. Hlavacek, H. Shulman and M. Waidner, "Smart RPKI Validation: Avoiding Errors and Preventing Hijacks", in *European Symposium on Research in Computer Security*, Copenhagen, Denmark: Springer, 2022, pp. 509–530. DOI: 10.1007/978-3-031-17140-6_25.

[201] F. SIT. "Smart Validator". (2021), [Online]. Available: https://rose.smart-validator.net.

[202] V. G. Li, G. Akiwate, K. Levchenko, G. M. Voelker and S. Savage, "Clairvoyance: Inferring Blocklist Use on the Internet", in *International Conference on Passive and Active Network Measurement*, Virtual: Springer, 2021, pp. 57–75. DOI: 10.1007/978-3-030-72582-2_4.

[203] W. Li and T. Chung, *RoVista*, 2023. [Online]. Available: https://rovista.netsecurelab.org.

[204] A. Reuter, R. Bush, I. Cunha, E. Katz-Bassett, T. C. Schmidt and M. Wählisch, *Measuring RPKI Route Origin Validation Deployment.* [Online]. Available: https://rov.rpki.net.

[205] C. Gray, C. Mosig, R. Bush, C. Pelsser, M. Roughan, T. C. Schmidt and M. Wählisch, "BGP Beacons, Network Tomography, and Bayesian Computation to Locate Route Flap Damping", in *Proceedings of ACM Internet Measurement Conference (IMC)*, New York: ACM, 2020, pp. 492–505. doi: 10.1145/3419394.3423624.

[206] B. Cartwright-Cox, *The year of RPKI on the control plane*, 2020. [Online]. Available: https://ripe80.ripe.net/presentations/36-Ben_Cox.pdf.

[207] H. Shulman, N. Vogel and M. Waidner, "Poster: Insights into Global Deployment of RPKI Validation", in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, Los Angeles, USA: ACM, 2022, pp. 3467–3469.

[208] K. van Hove, *Where Did My Packet Go? Measuring the Impact of RPKI ROV*, 2022. [Online]. Available: https://labs.ripe.net/author/koen-van-hove/where-did-my-packet-go-measuring-the-impact-of-rpki-rov/.

[209] O. Gasser, Q. Scheitle, S. Gebhard and G. Carle. "ZMap v6 Fork". (2021), [Online]. Available: https://github.com/tumi8/zmap.

[210] RIPE NCC Staff, "RIPE Atlas: A Global Internet Measurement Network", *Internet Protocol Journal*, vol. 18, no. 3, 2015.

[211] B. Tierney, J. Metzger, J. Boote, E. Boyd, A. Brown, R. Carlson, M. Zekauskas, J. Zurawski, M. Swany and M. Grigoriev, "perfSONAR: Instantiating a Global Network Measurement Framework", *SOSP Wksp. Real Overlays and Distrib. Sys*, 2009.

[212] Multiple authors, *The perfSONAR Project*, 2022. [Online]. Available: https://www.perfsonar.net/.

[213] Z. Durumeric, E. Wustrow and J. A. Halderman, "ZMap: Fast Internet-wide scanning and its security applications", in *22nd {USENIX} Security Symposium ({USENIX} Security 13)*, 2013, pp. 605–620, isbn: 9781931971034.

[214] X. Lee, X. Liu, Z. Yan, G. Geng and Y. Fu, "RPKI Deployment Considerations: Problem Analysis and Alternative Solutions", IETF Secretariat, Internet-Draft draft-lee-sidr-rpki-deployment-02, Jul. 2016. [Online]. Available: https://www.ietf.org/archive/id/draft-lee-sidr-rpki-deployment-02.txt.

[215] B. Kuerbis, *Regional Address Registries, Governance and Internet Freedom.* Internet Governance Project, 2008. [Online]. Available: https://www.internetgovernance.org/wp-content/uploads/RIRs-IGP-hyderabad.pdf.

[216] M. Mueller, M. van Eeten, and B. Kuerbis, *In Important Case, RIPE-NCC seeks legal clarity on how it responds to foreign court orders*, 2011. [Online]. Available: https://www.internetgovernance.org/2011/11/23/in-important-case-ripe-ncc-seeks-legal-clarity-on-how-it-responds-to-foreign-court-orders/.

[217] P. Bright, *How the Comodo certificate fraud calls CA trust into question*, 2011. [Online]. Available: https://arstechnica.com/information-technology/2011/03/how-the-comodo-certificate-fraud-calls-ca-trust-into-question/.

[218] E. Galperin, S. Schoen and P. Eckersley, *A Post Mortem on the Iranian DigiNotar Attack*, 2011. [Online]. Available: https://www.eff.org/deeplinks/2011/09/post-mortem-iranian-diginotar-attack.

[219] M. Marquis-Boire, *A Brief History of DNS Hijackings*, 2012. [Online]. Available: http://archive.icann.org/en/meetings/costarica2012/bitcache/A%20brief%20History%20of%20DNS%20Hijacking%20%E2%80%93%20Morgan%20Marquis-Boire,%20Google-vid=33555&disposition=attachment&op=download.pdf.

[220] C. Wisniewski, *Turkish Certificate Authority Screwup Leads to Attempted Google Impersonation*, 2013. [Online]. Available: https://nakedsecurity.sophos.com/2013/01/04/turkish-certificate-authority-screwup-leads-to-attempted-google-impersonation/.

[221] D. Piscitello, *Guidance for Preparing Domain Name Orders, Seizures & Takedowns*, 2012. [Online]. Available: https://www.icann.org/en/system/files/files/guidance-domain-seizures-07mar12-en.pdf.

[222] C. Soghoian and S. Stamm, "Certified Lies: Detecting and Defeating Government Interception Attacks against SSL", in *International Conference on Financial Cryptography and Data Security*, Rodney Bay, St. Lucia: Springer, 2011, pp. 250–259. DOI: 10.1007/978-3-642-27576-0_20.

[223] The Communications Security, Reliability and Interoperability Council III, *Secure BGP Deployment - Final Report*, 2013. [Online]. Available: https://transition.fcc.gov/bureaus/pshs/advisory/csric3/CSRIC_III_WG6_Report_March_%202013.pdf.

[224] B. Kuerbis and M. Mueller, "Negotiating a New Governance Hierarchy: An Analysis of the Conflicting Incentives to Secure Internet Routing", *Communications and Strategies*, no. 81, pp. 125–142, 2011.

[225] M. Mueller, A. Schmidt and B. Kuerbis, "Internet Security and Networked Governance in International Relations", *International Studies Review*, vol. 15, no. 1, pp. 86–104, 2013.

[226] RIPE NCC, *RIPE NCC Audit Activity*, 2018. [Online]. Available: https://www.ripe.net/publications/docs/ripe-694.

[227] A. Haeberlen, I. C. Avramopoulos, J. Rexford and P. Druschel, "NetReview: Detecting When Interdomain Routing Goes Wrong.", in *6th USENIX Symposium on Networked Systems Design and Implementation*, vol. 2009, Boston, MA, USA: USENIX Association, 2009, pp. 437–452.

[228] J. Paillisse, M. Ferriol, E. Garcia, H. Latif, C. Piris, A. Lopez, B. Kuerbis, A. Rodriguez-Natal, V. Ermagan, F. Maino *et al.*, "IPchain: Securing IP Prefix Allocation and Delegation with Blockchain", in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Halifax, NS, Canada: IEEE, 2018, pp. 1236–1243.

[229] L. Mastilak, P. Helebrandt, M. Galinski and I. Kotuliak, “Secure Inter-Domain Routing Based on Blockchain: A Comprehensive Survey”, *Sensors*, vol. 22, no. 4, 2022, ISSN: 1424-8220. DOI: 10.3390/s22041437. [Online]. Available: https://www.mdpi.com/1424-8220/22/4/1437.

[230] K. Shrishak and H. Shulman, “Limiting the Power of RPKI Authorities”, in *Proceedings of the Applied Networking Research Workshop*, Madrid, Spain: ACM, 2020, pp. 12–18. DOI: 10.1145/3404868.3406674.

[231] E. Heilman, *Mirror Worlds, Eclipse Attacks and the Security of Bitcoin and the RPKI*, Dissertation, 2022. [Online]. Available: https://open.bu.edu/handle/2144/44796.

[232] S. Kent and D. Mandelberg, “Suspenders: A Fail-safe Mechanism for the RPKI”, IETF Secretariat, Internet-Draft draft-kent-sidr-suspenders-04, Oct. 2015. [Online]. Available: https://www.ietf.org/archive/id/draft-kent-sidr-suspenders-04.txt.

[233] N. Trenaman, *RPKI Outage Post-Mortem - Inconsistent Certificates*, 2021. [Online]. Available: https://mailarchive.ietf.org/arch/msg/sidrops/mlFkEcI0DCLv0ZXLY3uZmM1x2do/.

[234] N. Trenaman, *RPKI ROA Deletion: Post-mortem*, 2020. [Online]. Available: https://www.ripe.net/ripe/mail/archives/routing-wg/2020-April/004072.html.

[235] N. Trenaman, *RPKI Outage Post-Mortem - Disk Quota*, 2020. [Online]. Available: https://www.ripe.net/ripe/mail/archives/routing-wg/2020-February/004015.html.

[236] N. Trenaman, *Lessons Learned on Improving RPKI*, 2020. [Online]. Available: https://labs.ripe.net/author/nathalie_nathalie/lessons-learned-on-improving-rpki/.

[237] M. Hoffmann, *Why Routinator Doesn’t Fall Back to Rsync*, 2020. [Online]. Available: https://blog.nlnetlabs.nl/why-routinator-doesnt-fall-back-to-rsync/.

[238] Multiple authors, *RIPE NCC Routing Working Group Mail Archive*, 2022. [Online]. Available: https://www.ripe.net/ripe/mail/archives/routing-wg/2022-February/004528.html.

[239] E. Cambiaso, G. Papaleo, G. Chiola and M. Aiello, “Slow dos attacks: Definition and categorisation”, *International Journal of Trust Management in Computing and Communications*, vol. 1, no. 3-4, pp. 300–319, 2013. DOI: 10.1504/IJTMCC.2013.056440.

[240] D. Mirdita, H. Shulman and M. Waidner, “Poster: RPKI Kill Switch”, in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 3423–3425.

[241] K. van Hove, "Tree Hints for the Resource Public Key Infrastructure (RPKI)", IETF Secretariat, Internet-Draft draft-kwvanhove-sidrops-rpki-tree-hints-01, Dec. 2021. [Online]. Available: https://www.ietf.org/archive/id/draft-kwvanhove-sidrops-rpki-tree-hints-01.txt.

[242] T. Bruijnzeels, R. Bush and G. Michaelson, "Resource Public Key Infrastructure (RPKI) Repository Requirements", IETF Secretariat, Internet-Draft draft-ietf-sidrops-prefer-rrdp-01, Oct. 2021.

[243] Open Web Application Security Project (OWASP), *XML Security Cheat Sheet*, 2022. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/XML_Security_Cheat_Sheet.html.

[244] Open Web Application Security Project (OWASP), *REST Security Cheat Sheet*, 2022. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html.

[245] M. Canet, A. Kumar, C. Lauradoux, M.-A. Rakotomanga and R. Safavi-Naini, "Decompression Quines and Anti-Viruses", in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, Scottsdale, Arizona, USA: ACM, 2017, pp. 23–34. DOI: 10.1145/3029806.3029818.

[246] North American Network Operators' Group (NANOG), *Possible rsync validation dos vuln*, 2021. [Online]. Available: https://mailman.nanog.org/pipermail/nanog/2021-October/216309.html.

[247] B. Schlinker, T. Arnold, I. Cunha and E. Katz-Bassett, "PEERING: Virtualizing BGP at the Edge for Research", in *Proceedings of ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, New York, NY, USA: ACM, Dec. 2019. DOI: 10.1145/3359989.3365414.

[248] RIPE NCC, *RIPE Stat*, 2020. [Online]. Available: https://stat.ripe.net/.

[249] RIPE NCC, *RIPE Atlas Cousteau*, 2019. [Online]. Available: https://github.com/RIPE-NCC/ripe-atlas-cousteau.

[250] R. Moskowitz, D. Karrenberg, Y. Rekhter, E. Lear and G. J. de Groot, *Address Allocation for Private Internets*, RFC 1918, Feb. 1996. DOI: 10.17487/RFC1918. [Online]. Available: https://www.rfc-editor.org/info/rfc1918.

[251] Team Cymru, *IP to ASN mapping*, 2017. [Online]. Available: https://team-cymru.com/community-services/ip-asn-mapping/.

[252] G. Nomikos and X. Dimitropoulos, "traIXroute: Detecting IXPs in Traceroute Paths", in *International Conference on Passive and Active Network Measurement*, Heraklion, Crete, Greece: Springer, 2016, pp. 346–358. DOI: 10.1007/978-3-319-30505-9_26.

[253] PeeringDB, *The Interconnection Database*, 2020. [Online]. Available: https://www.peeringdb.com/.

[254] Packet Clearing House, *Internet Exchange Directory*, 2020. [Online]. Available: https://www.pch.net/ixp/dir.

[255] G. Nomikos, V. Kotronis, P. Sermpezis, P. Gigis, L. Manassakis, C. Dietzel, S. Konstantaras, X. Dimitropoulos and V. Giotsas, “O Peer, Where Art Thou? Uncovering Remote Peering Interconnections at IXPs”, in *Proceedings of the Internet Measurement Conference 2018*, New York, NY, USA: ACM, 2018, pp. 265–278.

[256] K. Edeline and B. Donnet, “A First Look at the Prevalence and Persistence of Middleboxes in the Wild”, in *2017 29th International Teletraffic Congress (ITC 29)*, vol. 1, Genoa, Italy: IEEE, 2017, pp. 161–168. DOI: 10.23919/ITC.2017.8064352.

[257] U. Javed, I. Cunha, D. Choffnes, E. Katz-Bassett, T. Anderson and A. Krishnamurthy, “PoiRoot: Investigating the Root Cause of Interdomain Path Changes”, *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 183–194, 2013. DOI: 10.1145/2486001.2486036.

[258] J. Snijders, *Deep Dive on Manifest Handling*, RPKI mailinglist, 2020. [Online]. Available: https://lists.nlnetlabs.nl/pipermail/rpki/2020-December/000245.html.

[259] D. Mandelberg and R. Hansen, *RPSTIR*, C, GitHub, version v0.13, bgpsecurity, 2021. [Online]. Available: https://github.com/bgpsecurity/rpstir.

[260] L. Poinsignon *et al.*, *GoRTR*, Go, GitHub, version v0.14.7, Cloudflare, 2021. [Online]. Available: https://github.com/cloudflare/gortr.

[261] NLnet Labs, *Homepage*, 2021. [Online]. Available: https://nlnetlabs.nl/.

[262] M. Hoffmann and A. Band, *Krill*, GitHub, 2021. [Online]. Available: https://github.com/NLnetLabs/krill.

[263] RPKI Community, *RPKI Discord Channel*, 9th Mar. 2021. [Online]. Available: https://discord.gg/8dvKB5Ykhy.

[264] NLnet Labs, *RPKI Mailinglist – Discussion on RPKI deployment and tools developed by NLnet Labs*, NLnet Labs, 2021. [Online]. Available: https://lists.nlnetlabs.nl/mailman/listinfo/rpki.

[265] Alex Band and the RPKI Community, *RPKI Documentation*, NLnetLabs. [Online]. Available: https://rpki.readthedocs.io/en/latest/index.html.

[266] *Routinator 3000 - The free, open source RPKI Validator by NLnetLabs.* Twitter, NLnetLabs, Oct. 2018. [Online]. Available: https://twitter.com/routinator3000.

[267] LACNIC and NIC.MX, *FORT project*, 2021. [Online]. Available: https://fortproject.net/en/home.

[268] K. Dzonsons, C. Jeker, S. Benoit, J. Snijders and R. Scheck, *Rpki-client-portable*, GitHub, 2020. [Online]. Available: https://github.com/rpki-client/rpki-client-portable.

[269] Raspberry Pi Foundation. “Raspberry Pi Downloads”. (2021), [Online]. Available: https://downloads.raspberrypi.org/.

[270] M. Puzanov, *Rrdp rsync fallback #57*, Haskell, GitHub, 2021. [Online]. Available: https://github.com/lolepezy/rpki-prover/pull/57.

[271] L. Poinsignon, *Status Tweet*, Twitter, 29th Jan. 2021. [Online]. Available: https://twitter.com/lpoinsig/status/1355199025100668929?s=21.

[272] P. Friedemann, *RAM usage in 0.9.0*, GitHub, 2021. [Online]. Available: https://github.com/lolepezy/rpki-prover/issues/41.

[273] M. Hoffmann, *Increased memory consumption of Routinator 0.9.0*, RPKI mailinglist, 2021. [Online]. Available: https://lists.nlnetlabs.nl/pipermail/rpki/2021-June/000289.html.

[274] J. M. Smith, K. Birkeland, T. McDaniel and M. Schuchard, "Withdrawing the BGP Re-Routing Curtain", in *Network and Distributed System Security Symposium (NDSS)*, 2020.

[275] Internet Address Hitlist. "The ANT Lab: Analysis of Network Traffic, Internet Hitlist it93w". (2021), [Online]. Available: https://ant.isi.edu/datasets/readmes/internet_address_hitlist_it93w-20210202.README.txt.

[276] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczynski, S. D. Strowes, L. Hendriks and G. Carle, "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists", in *Proceedings of the 2018 Internet Measurement Conference*, Boston, MA, USA: ACM, 2018. DOI: 10.1145/3278532.3278564.

[277] RIPE NCC, *RIPE Atlas Sagan*, 2019. [Online]. Available: https://github.com/RIPE-NCC/ripe.atlas.sagan.

[278] N. Rodday. "TAURIN-21 workshop paper artifacts". (2021), [Online]. Available: https://github.com/nrodday/TAURIN-21.

[279] R. V. Oliveira, B. Zhang and L. Zhang, "Observing the Evolution of Internet AS Topology", in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, 2007, pp. 313–324. DOI: 10.1145/1282380.1282416.

[280] J. Yuchen. "ProbLink AS Relationships Dataset, 20210401.txt". (2021), [Online]. Available: https://yuchenjin.github.io/problink-as-relationships/.

[281] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas and K. Claffy, "AS Relationships, Customer Cones, and Validation", in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 243–256. DOI: 10.1145/2504730.2504735.

[282] Y. Jin, C. Scott, A. Dhamdhere, V. Giotsas, A. Krishnamurthy and S. Shenker, "Stable and Practical {AS} Relationship Inference with ProbLink", in *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, 2019, pp. 581–598.

[283] I. Castro, J. C. Cardona, S. Gorinsky and P. Francois, "Remote Peering: More Peering without Internet Flattening", in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 185–198. DOI: 10.1145/2674005.2675013.

[284] Z. Zhao and J. Bi, “Characterizing and Analysis of the Flattening Internet Topology”, in *2013 IEEE Symposium on Computers and Communications (ISCC)*, IEEE, 2013, pp. 000 219–000 225. DOI: 10.1109/ISCC.2013.6754949.

[285] A. Marder, M. Luckie, A. Dhamdhere, B. Huffaker, k. claffy kc and J. M. Smith, “Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale”, in *Proceedings of the 2018 Internet Measurement Conference*, 2018.

[286] RIPE NCC. “RIPE Altas probes metadata”. (2021), [Online]. Available: https://ftp.ripe.net/ripe/atlas/probes/archive/2021/05/.

[287] A. Cohen, Y. Gilad, A. Herzberg and M. Schapira, “Jumpstarting BGP Security with Path-End Validation”, in *Proceedings of the 2016 ACM SIGCOMM Conference*, Florianópolis, Brazil: ACM, 2016, pp. 342–355. DOI: 10.1145/2934872.2934883.

[288] W. Brand and J. Posen, *A Reproduction of "Jumpstarting BGP Security with Path-End Validation"*, 2020. [Online]. Available: https://reproducingnetworkresearch.files.wordpress.com/2020/06/brand_posen.pdf.

[289] W. Brand and J. Posen, *bgpsecsim Github Repository*, 2020. [Online]. Available: https://github.com/jimpo/bgpsec-sim/.

[290] J. Furuness, C. Morris, R. Morillo, A. Herzberg and B. Wang, “Bgpy: The bgp python security simulator”, in *Proceedings of the 16th Cyber Security Experimentation and Test Workshop*, ser. CSET ’23, Marina del Rey, CA, USA: Association for Computing Machinery, 2023, pp. 41–56, ISBN: 9798400707889. DOI: 10.1145/3607505.3607509.

[291] *NetworkX - Network Analysis in Python*, 2023. [Online]. Available: https://networkx.org/.

[292] O. Borchert, K. Lee, K. Sriram, D. Montgomery, P. Gleichmann and M. Adalier, “BGP Secure Routing Extension (BGP-SRx): Reference Implementation and Test Tools for Emerging BGP Security Standards”, National Institute of Standards and Technology, Tech. Rep., 2021.

[293] S. Kotikalapudi, *ASPA-based BGP AS_PATH Verification - Invited Webinar at the Internet2 Organization*, 2023. [Online]. Available: https://github.com/ksriram25/ASPA/blob/master/Internet2-webinar-ASPA-sriram.pdf.

[294] B. Cartwright-Cox, *The year of RPKI on the control plane*, 2019. [Online]. Available: https://blog.benjojo.co.uk/post/the-year-of-rpki-on-the-control-plane.

[295] A. Siddiqui, *Unpacking the First Route Leak Prevented by ASPA*, 2023. [Online]. Available: https://www.manrs.org/2023/02/unpacking-the-first-route-leak-prevented-by-aspa/.

[296] K. Sriram and A. Azimov, "Methods for Detection and Mitigation of BGP Route Leaks", Internet Engineering Task Force, Internet-Draft draft-ietf-grow-route-leak-detection-mitigation-09, Jul. 2023, Work in Progress, 10 pp. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-grow-route-leak-detection-mitigation/09/.

[297] A. Azimov, E. Bogomazov, R. Bush, K. Patel and K. Sriram, "Route Leak Prevention and Detection Using Roles in UPDATE and OPEN Messages", RFC Editor, RFC 9234, May 2022. DOI: 10.17487/RFC9234. [Online]. Available: https://www.rfc-editor.org/info/rfc9234.

[298] *Netdisco.* [Online]. Available: http://netdisco.org/.

[299] *Nessus.* [Online]. Available: http://www.nessus.org/.

[300] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator", in *Modeling and tools for network simulation*, Springer, 2010, pp. 15–34. DOI: 10.1007/978-3-642-12331-3_2.

[301] A. Varga, "OMNeT++", in *Modeling and tools for network simulation*, Springer, 2010, pp. 35–59.

[302] J. C. Neumann, *The Book of GNS3: Build Virtual Network Labs using Cisco, Juniper, and more.* No Starch Press, 2015.

[303] B. Lantz, B. Heller and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks", in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010, pp. 1–6. DOI: 10.1145/1868447.1868466.

[304] R. Emiliano and M. Antunes, "Automatic Network Configuration in Virtualized Environment using GNS3", in *2015 10th International Conference on Computer Science & Education (ICCSE)*, IEEE, 2015, pp. 25–30.

[305] A. Hagberg, P. Swart and D. S Chult, "Exploring Network Structure, Dynamics, and Function using NetworkX", Jan. 2008. [Online]. Available: https://www.osti.gov/biblio/960616.

[306] *Jinja2 - Templating Engine*, 2008. [Online]. Available: https://palletsprojects.com/p/jinja/.

[307] F. Soppelsa and C. Kaewkasi, *Native Docker Clustering with Swarm.* Packt Publishing Ltd, 2016.

[308] *FRRouting*, 2017. [Online]. Available: https://frrouting.org/.

[309] K. Ishiguro, "Gnu Zebra", 2002. [Online]. Available: https://www.gnu.org/software/zebra/.

[310] T. Petric, *Running 1,000 Containers in Docker Swarm*, 2017. [Online]. Available: https://www.cloudbees.com/blog/running-1000-containers-in-docker-swarm.

[311] C. Partridge and M. Allman, "Ethical Considerations in Network Measurement Papers", *Communications of the ACM*, vol. 59, no. 10, pp. 58–64, 2016. DOI: 10.1145/2896816.

[312] E. Kenneally and D. Dittrich, "The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research", *Available at SSRN 2445102*, 2012.

[313] E. Pauley and P. McDaniel, "Understanding the Ethical Frameworks of Internet Measurement Studies", in *2nd International Workshop on Ethics in Computer Security*, 2023.

[314] *Ripe routing information service (ris) - raw data*. [Online]. Available: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data.

[315] The PEERING Testbed, *Acceptable Use Policy*. [Online]. Available: https://peering.ee.columbia.edu/aup/.

[316] R. Kisteleki, *Ethics of RIPE Atlas Measurements*, 2016. [Online]. Available: https://labs.ripe.net/author/kistel/ethics-of-ripe-atlas-measurements/.

# Curriculum Vitae of the Author

Nils Miro Rodday was born in Tönisvorst, Germany on March 12, 1989. After obtaining his highschool diploma in 2009 he carried out a voluntary year of social service with the Malteser Hilfsdienst e.V., where he completed his training as a paramedic. In 2010, he started studying business informatics at the University of Applied Sciences in Münster and finished his Bachelor's degree in 2013. After this he continued to study within the European Institute of Technology (EIT) Digital double-degree program towards his Master's degree. During the first year of the program he studied in Trento, Italy while during the second year he finished his Master's degree at the University of Twente, Netherlands with a focus on network security. His thesis was titled 'Exploring Security Vulnerabilities of Unmanned Aerial Vehicles' and received world-wide media attention. In 2015, Nils started to work as an IT-Security consultant for IBM Deutschland GmbH and was promoted to Senior IT-Security consultant after the first year. He moved on to become a PhD student at the network security group at the University of the Bundeswehr Munich in collaboration with the Design and Analysis of Communication Systems (DACS) group at the University of Twente in 2017 to work on Internet routing security. During his time as a PhD student he taught several classes at the University of the Bundeswehr Munich, including Introduction to Networking during each fall 2017–2023, Advanced Networking during each spring 2018–2022, and a Cyber-Security seminar in each spring 2019–2023. Moreover, he gave guest lectures on the topic of routing security at the University of Twente, Netherlands and the Blekinge Institute of Technology, Sweden. He was supervised by Prof. Gabi Dreo Rodosek of the University of the Bundeswehr Munich and Prof. Aiko Pras of the University of Twente, and co-supervised by Prof. Roland van Rijswijk-Deij of the University of Twente.

# PUBLICATIONS BY THE AUTHOR

This is a comprehensive list of publications by the author, sorted in reverse chronological order.

- **N. Rodday**, G.D. Rodosek, A. Pras, R. van Rijswijk-Deij, 2024. Exploring the Benefit of Path Plausibility Algorithms in BGP. In Network Operations and Management Symposium (NOMS), Seoul, South Korea. *Accepted for publication*

- **N. Rodday**, I. Cunha, R. Bush, E. Katz-Bassett, G.D. Rodosek, T.C. Schmidt, and M. Wählisch, 2023. The Resource Public Key Infrastructure (RPKI): A Survey on Measurements and Future Prospects. In Transactions on Network and Service Management (TNSM). IEEE. *Accepted for publication*

- N. Höger, **N. Rodday**, O. Borchert, G.D. Rodosek, 2023. Path Plausibility Algorithms in GoBGP. In Proceedings of the 19th International Conference on Network and Service Management (CNSM), Niagara Falls, Canada.

- **N. Rodday**, G.D. Rodosek, 2023. BGPEval: Automating Large-Scale Testbed Creation. In Proceedings of the 19th International Conference on Network and Service Management (CNSM), Niagara Falls, Canada.

- P. Friedemann, **N. Rodday**, G.D. Rodosek, 2022. Assessing the RPKI Validator Ecosystem. In Proceedings of the 13th International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain.

- **N. Rodday**, I. Cunha, R. Bush, E. Katz-Bassett, G.D. Rodosek, T.C. Schmidt, and M. Wählisch, 2021. Revisiting RPKI Route Origin Validation on the Data Plane. In Proceedings of the Network Traffic Measurement and Analysis Conference (TMA), Virtual.

- **N. Rodday**, L. Kaltenbach, I. Cunha, R. Bush, E. Katz-Bassett, G.D. Rodosek, T.C. Schmidt, and M. Wählisch, 2021. On the Deployment of Default Routes in Inter-domain Routing. In Proceedings of the ACM SIGCOMM Workshop on Technologies, Applications, and Uses of a Responsible Internet (TAURIN), Virtual.

- **N. Rodday**, R. van Baaren, L. Hendriks, R. van Rijswijk-Deij, A. Pras, and G. Dreo, 2020. Evaluating RPKI ROV identification methodologies in automatically generated mininet topologies. In Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT), Barcelona, Spain.

- R. Poschinger, **N. Rodday**, R. Labaca-Castro, and G. Dreo Rodosek, 2020. OpenMTD: A Framework for Efficient Network-Level MTD Evaluation. In Proceedings of the 7th ACM Workshop on Moving Target Defense (MTD), Virtual.

- **N. Rodday**, R.L. Castro, K. Streit, and G.D. Rodosek, 2019. Evaluating TCP Connection Healthiness. In Proceedings of the 29th International Telecommunication Networks and Applications Conference (ITNAC), Auckland, New Zealand.
- K. Streit, R.L. Castro, **N. Rodday**, and G.D. Rodosek, 2019. Topology Update Algorithm for Wireless Networks. In Proceedings of the 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), Monterey, USA.
- K. Streit, **N. Rodday**, F. Steuber, C. Schmitt, and G.D. Rodosek, 2019. Wireless SDN for Highly Utilized MANETs. In Proceedings of the International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain.
- **N. Rodday**, R. Bush, I. Cunha, E. Katz-Bassett, G. Dreo Rodosek, T.C. Schmidt, and M. Wählisch, 2019. Extending RPKI ROV Measurement Coverage, Poster @ 19th Internet Measurement Conference (IMC), Amsterdam, The Netherlands.
- **N. Rodday**, K. Streit, G. Dreo Rodosek, and A. Pras. On the Usage of DSCP and ECN Codepoints in Internet Backbone Traffic Traces for IPv4 and IPv6, 2019, In Proceedings of the International Symposium on Networks, Computers and Communications (ISNCC), Istanbul, Turkey.
- **N. Rodday**, K. Streit, G. Dreo Rodosek, A. Pras, 2018. An Empirical Study of DSCP and ECN Usage by Application in Internet Traffic Traces, Poster @ 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT), Heraklion, Greece.
- K. Streit, **N. Rodday**, G. Dreo Rodosek, 2018. AODV-CBR: Capacity-based Path Discovery Algorithm for MANETs with High Utilization, In Proceedings of the Advances in Wireless and Optical Communications (RTUWO), Riga, Latvia.
- **N. Rodday**, A. Pras and G. Dreo Rodosek. Towards European Network Sovereignty, 2018. In Proceedings of the 12th International Conference on Autonomous Infrastructure, Management and Security (AIMS), Munich, Germany.