Bachelor Informatica

UvA Universiteit van Amsterdam

# BGP Routing Security and Deployment Strategies

Bryan Eikema

June 17, 2015

**Supervisor(s):** Benno Overeinder (NLnet Labs), Stavros Konstantaras (NLnet Labs)

**Signed:**

### Abstract

The Resource Public Key Infrastructure is an infrastructure that can be used to secure BGP. One application of this infrastructure is origin validation, in which a BGP speaker can verify that another BGP speaker has the right to originate certain IP prefixes that they advertise. Use of this infrastructure is not yet widespread. In this research we study and analyse the effects of various strategies to deploy origin validation. We use a BGP simulator to simulate the effects on the security and performance of the network using several deployment strategies and security policies. We find that deploying origin validation to a smaller groups of large Autonomous Systems give the best results for securing the network. We see that some security policies have negative effects on the performance of the network, but have a positive effect on the security of the network. Finally, we give insights in the current status of BGP security.

# Contents

CHAPTER 1

# Introduction

The Border Gateway Protocol (BGP) is arguably one of the most important protocol in the Internet. BGP forms the routing infrastructure that routes packets between entire networks. BGP routers exchange routing and reachability information with their BGP neighbors to acquire global knowledge of the network topology. However, the protocol itself does not include any mechanisms to confirm the correctness of this information. It fully trusts the information received from peers. The correct functioning of BGP is key for the correct functioning of the Internet. This makes BGP an attractive target for attacks. Mistakes made by a BGP speaker can also cause wrong information to spread throughout the network, possibly rendering parts of the network unreachable.

In recents years researchers have come up with an infrastructure that allows BGP routers to confirm the correctness of routing information they receive, the Resource Public Key Infrastructure (RPKI). The RPKI is a Public Key Infrastructure (PKI) that provides cryptographic proof that networks are allowed to send certain routing information, thus introducing cryptographic trust to information received. An application of this infrastructure is origin validation. Origin validation allows BGP speakers to confirm that other BGP speakers are allowed to originate the IP prefixes that they advertise. By using origin validation the negative effects of mistakes made by BGP speakers can be mitigated.

What a BGP speaker does with the information gathered from doing origin validation is up to the BGP speakers its local policies. It is not specified how to use this information, yet suggestions have been made by major routing vendors. In this research we will name the policies that decide how to deal with this information *security policies*. A BGP speaker could choose to use a security policy that prefers routes with a valid origin over routes with an invalid origin, independent of other routing properties. Security policies can also decide whether routes to certain prefixes, with for example an invalid origin, should be dropped or not.

The use of the RPKI and origin validation is unfortunately not yet widespread. An analysis by RIPE shows that out of 590,077 unique ”IPv4 prefix / origin network” pairs collected, more than 90% could not be validated using origin validation [1]. We expect that the amount of networks doing origin validation is even less than the amount of networks using the RPKI to allow the validation of their own prefixes.

## 1.1 Research Questions

In this research we will look at how we can best deploy origin validation to the Internet. We will experiment with various deployment strategies and security policies and measure the effects on security and performance of the network. We will also look which prefixes can currently be validated that are reachable on the Internet and attempt to gain insights into the current status of deployment. The research questions that we will try to answer are stated below.

- *What is the impact on routing security for different origin validation deployment strategies?*
- *What is the impact on routing security for different origin validation security policies?*
- *What is the current status of routing security given the current publication and potential usage of RPKI data?*

## 1.2 Related Work

In a similar research by Gill et al. in 2011 a deployment strategy is proposed based on the economic incentives of businesses [2]. The researchers appeal to economic incentives for adoption instead of security incentives, trying to side step issues that occurred in the slow adoption of IPv6. They suggest a model where market pressure is created by financing a small set of early adopters to deploy BGP security. They use a security policy where BGP is only secured when other routing considerations are equally preferrable between two routes. This way, they claim, incentives are created to deploy BGP security. They also suggest that stub networks, networks that do not offer transit to other networks, only deploy a unidirectional version of BGP security. When all the transit networks validate whether the routes they have are secure, stub networks should not have to do the validation part themselves. This way stub networks are more likely to deploy BGP security. They show that under the correct circumstances this can lead to the successful deployment of BGP security.

## 1.3 Thesis Outline

yhIn chapter two we will explain the workings of BGP, the RPKI and applications of the RPKI, such as origin validation. Chapter three will discuss our approach to answer our research questions. We explain what experiments we are going to do, how we are going to simulate those experiments and what deployment strategies and security policies we use for those experiments. In chapter four we shortly discuss how we implement origin validation into the simulator and how we generate our experiments. In chapters five and six we present and discuss the results of our experiments. We draw conclusions from these results in chapter seven and discuss what future work could be done in chapter eight.

CHAPTER 2

# Background

## 2.1 The Border Gateway Protocol

The Border Gateway Protocol (BGP) is the inter-domain routing protocol of the Internet. It is the protocol that connects tens of thousands of networks in the Internet to form one big interconnected network. It is the de facto standard inter-domain routing protocol in the Internet and is therefore very important for the correct functioning of the Internet.

### 2.1.1 Routing

BGP exchanges network reachability information between Autonomous Systems. An Autonomous System (AS) is a set of routers under a single technical administration, a unique AS number, that uses intra-domain routing protocols to route packets within the the Autonomous System [3]. Examples of Autonomous Systems are corporate networks, like Google's network, and networks of Internet Service Providers, such as XS4ALL its network. An Autonomous System can use several intra-domain routing protocols within, but will present a single consistent picture of the destinations reachable through the AS to the outside world. BGP speakers announce and withdraw routes to IP prefixes to their BGP speaking neighbors. When a BGP speaker receives multiple routes to the same prefix, it goes through a complex decision process in which it decides the best route using local policies and BGP path attributes. Local policies are not defined by the BGP standard, but can differ for every BGP speaker. Local policies usually adhere to business policies.

One of the main path attributes in a BGP announcement is the AS-path. Every AS has got a unique AS number that it prepends to the AS-path before advertising it to its neighbors. One of the uses of the AS-path is to prevent routing loops from occurring. When an AS receives a BGP announcement it checks whether the AS-path contains its own AS number, if so it will drop the route. The AS-path also plays a role in the route decision process. BGP speakers can choose to prefer shorter routes over longer routes, if local policies allow to do so.

A non-negative integer value called *local preference* is used to enforce local policies in BGP its route decision proces. Local policies will adjust this value to give routes a higher or lower preference than other routes. The local preference value may not depend on the existence or non-existence of other routes. The BGP decision process will always pick a route with a higher local preference value over one with a lower local preference value. If local preference values are not conclusive about selecting the best route, a tie breaking process will start that will decide the best route based on path attributes, such as the length of the AS-path. We will not go into the details of this complex tie breaking process.

### 2.1.2 Security

The Border Gateway Protocol does not provide any mechanisms to confirm the correctness of routing information received from other BGP speakers. Being such an important protocol in the Internet, this presents a vulnerability. If BGP speakers accidentally announce wrong routing information, it could potentially lead to parts of the network becoming unreachable. An example of this is the worldwide blocking of YouTube in February 2008. The Pakistani government ordered to have YouTube blocked. However, they made a mistake in configuring their BGP messages and caused most of YouTube's traffic worldwide to lead to Pakistani YouTube servers, which crashed and caused the YouTube website to be disrupted for multiple hours [4].

## 2.2 Resource Public Key Infrastructure

The Resource Public Key Infrastructure (RPKI) is a Public Key Infrastructure (PKI) that forms the backbone of BGP security. The RPKI exploits existing technologies, standards and processes to create a simple and robust framework that can be used to secure BGP [5]. It enables BGP speakers to verify the correctness of routing information they receive. BGP speakers using the RPKI for this are called Relying Parties (RPs). The RPKI allows Relying Parties to validate assertions about the legitimate holdership of Internet number resources: IP addresses and AS numbers. This is achieved through the use of a Public Key Infrastructure that uses resource certificates.

### 2.2.1 Resource certificates

Resource certificates are X.509 certificates conforming to the PKIX profile [6] with a critical extension for listing IP addresses and AS numbers [7][8]. An X.509 certificate consists of a serial number that is unique within the issuing authority, information about the issuer, validity information that states the starting date and ending date between which the certificate is valid, the subject's public key and a digital signature made with the private key of the issuer. Resource certificates add an extension to this that also state the Internet number resources allocated to the subject, as well as information about the public repositories where the certificate and possible member certificates can be found.

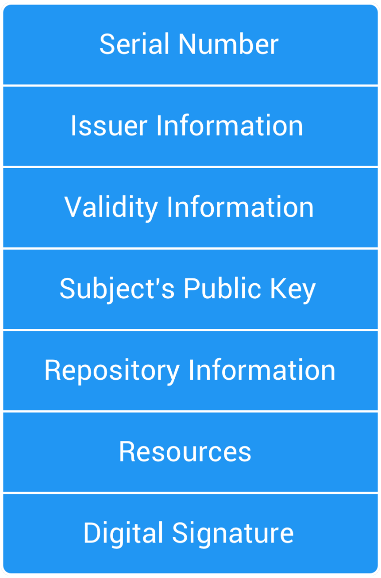

Figure 2.1: A simplified resource certificate

### 2.2.2 Infrastructure

The RPKI must enable Relying Parties to validate resource certificates. Not all issuers of resource certificates can blindly be trusted. An exception is the trust anchor of the PKI, which is a central entity that everyone trusts and owns a self-signed root certificate. The trust anchor issues certificates to its members. These members can also issue certificates to their own members. An authority that issues certificates is called a Certificate Authority (CA). To validate a certificate, the RP must construct a chain of valid certificates, starting at the trust anchor and going down to the CA.

The resources stated in the certificates should reflect the actual current state of allocations of these resources. Hence, the infrastructure needs a good knowledge of these allocations. The RPKI therefore mirrors the existing resource allocation infrastructure [7], the Internet Number Registry System [9]. The root of the Internet Number Registry System is the IANA, the Internet Assigned Numbers Authority. The IANA manages a complete pool of all possible IP addresses and AS numbers. It allocates blocks of these resources to the five Regional Internet Registries (RIRs): RIPE NCC, ARIN, APNIC, AfriNIC and LACNIC[1]. RIRs suballocate those resources to Local Internet Registries (LIRs), Internet Service Providers (ISPs) and end users. LIRs can continue to make suballocations to other LIRs, ISPs and end users.

Every Internet registry in the Internet Number Registry System should thus also take the role of a Certificate Authority in the RPKI. Accompanying an allocation of Internet number resources, should come a resource certificate that acts as a public attestation of that allocation. Ideally, the IANA holds a self-signed root certificate and issues certificates to all the RIRs. The RIRs issues certificates confirming Internet number resource allocations to their members. If that member is a LIR, they can use their obtained resource certificate to issue member certificates to their members. Currently the operational practice is that the RIRs act as a trust anchor.

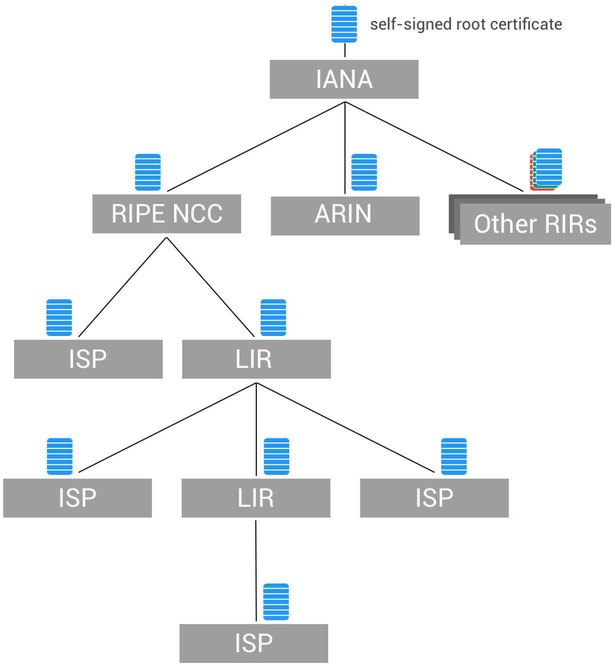

Figure 2.2: The Resource Public Key Infrastructure

[1] Respectively for the regions Europe & the Middle-East, North America, Asia Pacific, Africa and Latin America.

### 2.2.3 Repositories

Resource certificates should be publicly available information in order for RPs to be able to find these certificates. The RPKI does not use a single central repository containing all of the certificates. Instead, it uses a distributed system where each Certificate Authority is required to maintain its own repository [10]. Every CA thus maintains a repository containing all the certificates of their members and a digitally signed manifest file listing all of the repository its contents. The main purpose of this manifest file is that RPs can confirm that they have retrieved a complete copy of the repository [11]. In order to be able to find these repositories every resource certificate states the location of the corresponding repository, if one exists, and the location of the repository where the resource certificate itself is stored. For example, the root certificate states the location of the root repository. In the root repository the member certificates for the RIRs can be found. The RIPE NCC resource certificate would contain location of the root repository, where RIPE NCC its resource certificate is stored, and the location of the RIPE NCC repository, where member certificates of RIPE NCC are stored.

### 2.2.4 Certificate Revocation Lists

In the framework discussed so far, every resource certificate contains validity information stating the starting date and the expiration date between which the certificate is valid. However, a certificate may need to be revoked before the expiration date of the certificate. Therefore the RPKI makes conventional use of Certificate Revocation Lists (CRLs) [6]. A CRL states all the serial numbers of certificates that have been revoked. Examples of reasons for a CA to revoke a certificate are key rollover, reduction in the set of resources allocated or termination of the resource allocation [5]. Every CA maintains and regularly updates a CRL file in its public repository. The CRL is also stated in the manifest file for that repository.

## 2.3 Securing BGP using the RPKI

The goal of the RPKI is to provide a reliable infrastructure that can be used to secure BGP. There are two applications of the RPKI that we will discuss: origin validation and path validation. Origin validation focuses on verifying that the origin AS in the BGP announcement is allowed to originate the prefix stated in that BGP announcement. Path validation focuses on verifying that the AS path in the BGP announcement is correct [5]. In this research we will focus on origin validation.

### 2.3.1 Origin Validation

Origin validation, also called Route Origin Validation (ROV), uses the RPKI to verify that an AS is allowed to originate an IP prefix. Using origin validation, some mistakes made by BGP speakers can be detected. For example, when a BGP speaker announces a prefix that it does not have legitimate holdership of, ideally origin validation would detect it. An important object for origin validation is the Route Origin Authorization (ROA). Origin validation can be realized without making changes to the BGP protocol.

#### Route Origin Authorizations

BGP does not have a notion of resource holders and resource allocations. It only knows of AS numbers originating sets of prefixes. Therefore, the RPKI resource certificates are not sufficient to guide routing decisions [11]. A Route Origin Authorization is a signed object in the RPKI created by a holder of Internet number resources that explicitly authorizes an AS to originate routes to a given set of prefixes [11]. Such a ROA contains a single AS number, validity information that states the start and expiration date of the ROA and a sequence of prefixes with each sequence having an optional `maxLength` field [12]. These ROAs are created by resource holders and signed with their private key. This private key corresponds to the public key stated in their resource certificate. The ROAs are published in the RPKI repository system and are listed in the repository's manifest file.

### The ROA maxLength Field

The `maxLength` field of a ROA object is an optional field that can be set by the creator of the ROA. When the field is not set, the only prefix the AS is allowed to originate is the exact prefix specified in the ROA. All more specific prefixes advertised are considered invalid. When the `maxLength` field is set, it must be at least as big as the prefix length and at most the size of the IP address (32 or 128 for IPv4 and IPv6 respectively) [12]. The `maxLength` field defines the maximum length of more specific prefixes the AS is allowed to advertise [11]. For example, if the prefix `84.16.0.0/12` is advertised with a `maxLength` of 14, the AS is allowed to advertise `84.16.0.0/12`, `84.16.0.0/13`, `84.16.0.0/14`, but not `84.16.0.0/15`.

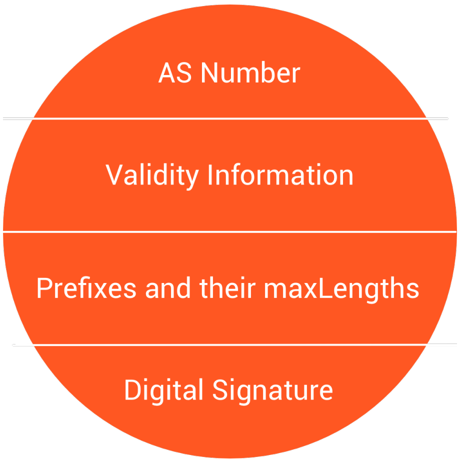

Figure 2.3: A Route Origination Authorization object

### Validation Process

ROAs can be used to verify that the AS originating a prefix is allowed to do so. If a ROA exists for a certain prefix, the origin validation process can be performed for that prefix. In order for the prefix to be valid, there must exist a valid ROA to prove this. A ROA is only valid when a number of conditions are met. Those conditions are stated below.

- A ROA or resource certificate is only valid if the chain of resource certificates leading up to a trust anchor are all valid.
- A ROA or resource certificate is only valid after the starting date and before the expiration date.
- ROAs and resource certificates are only valid if the digital signature can be verified using the public key stated in the corresponding (parent) certificate [13].
- The prefixes and AS number stated in a ROA must be encompassed in the prefixes stated in the corresponding resource certificate [12].
- All the resources stated in a resource certificate must be encompassed in the resources stated in the parent certificate [14].
- A resource certificate is not valid when it is revoked through the use of a Certificate Revocation List.

When one or multiple ROAs are valid for a given prefix, it can be verified that an advertised prefix is allowed or disallowed to originate from an AS. That is, keeping in mind the `maxLength` field. Advertising a more specific prefix than allowed by the `maxLength` field causes the prefix its origin to become invalid. A more detailed description of the ROA validation process can be found in RFCs 6488, 6487 and 6482.

### Policies

The BGP protocol itself is not altered by doing origin validation. The results of origin validation can be used in BGP policies, the regular method for selecting and rejecting routes. We will name BGP policies that deal with the results of doing origin validation *security policies*. Using security policies, routers could, for example, give a higher local preference value to routes that have a valid origin than to routes that have an invalid origin. A security policy could also decide to completely drop the invalid routes. However, this is not advised until RPKI has been deployed at a large scale.

### Validators

Origin validation is not commonly done by BGP routers themselves. Instead, a dedicated validator server goes through the computationally-expensive cryptographic validation process. Such a validator fetches a complete copy of the entire RPKI at regular intervals using rsync. It goes through the validation process and validates and invalidates prefixes. Using the RPKI to Router protocol (rtr) a router can fetch the results of the validation process and decide whether a BGP announcement has an `UNKNOWN`, `VALID` or `INVALID` origin. Origins are `VALID` or `INVALID` if ROAs exist for that prefix. The validation process described earlier will decide whether an origin is `VALID` or `INVALID`. If no ROA exists for the advertised prefix, the origin AS has the validity status `UNKNOWN` for this prefix. There is no standard defining what BGP speakers should do with this information, so local policies will deal with this.

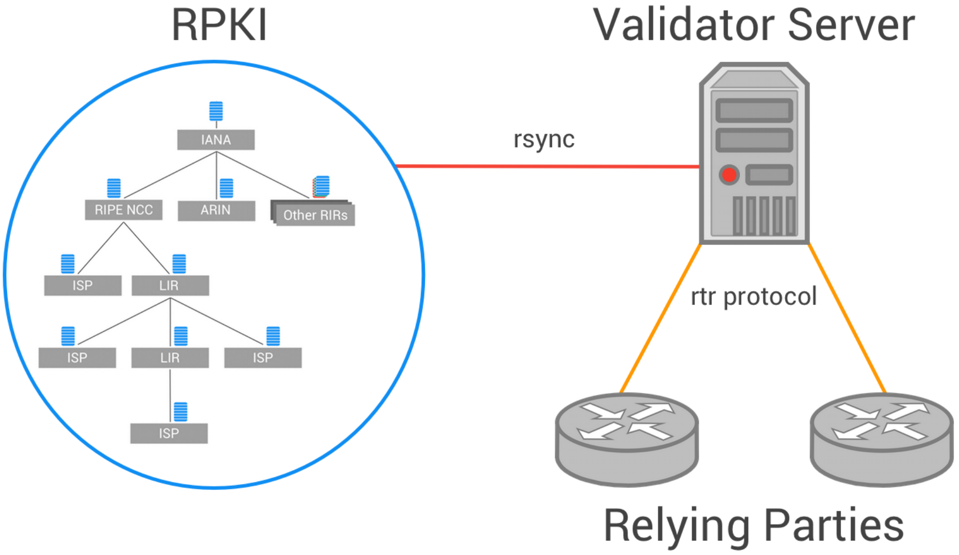

Figure 2.4: Acquiring origin validity information as a Relying Party

## 2.3.2 Path Validation

Origin validation can assure that the origin AS of a BGP announcement is authorized to originate the prefix advertised. However, origin validation has no way of validating that the AS path in the BGP announcement has not been altered. In path validation, or BGPsec, BGP announcements are digitally signed to enable this. The signature covers the BGP announcement as received, the local AS number, the AS number to which the update is being sent and a hash of the public key of the keypair used to sign the BGP announcement [5]. This signature can be used to validate that the AS path of the advertised BGP announcement is the actual path that the BGP announcement has traversed.

Path validation requires that every AS in the AS path digitally signs BGP announcements before advertising them. When a single AS in the AS path has not deployed path validation, the entire path cannot be validated. Whereas origin validation allows piecemeal deployment spread over the network, path validation requires chains of ASes to deploy path validation simultaneously in order to be effective. Path validation is a lot more compute-intensive than origin validation as as well and no values can be pre-computed. For every advertisement a BGP speaker advertises, it needs to go through a cryptographic signing process as well as doing the actual path validation. In this research we will only focus on the deployment of origin validation.

CHAPTER 3

# Approach

In order to answer our research questions we are going to simulate the Internet's BGP speakers using a BGP simulator called BGPsim. We are going to extend the simulator to allow groups of BGP speakers to do origin validation using pre-defined security policies. For our experiments we will deploy origin validation within selected groups of ASes to allow us to get insights in the effects of deploying origin validation.

## 3.1 BGPsim

For our experiments we will use the BGP simulator BGPsim. This simulator was developed to allow BGP simulations of large-scale networks, such as the Internet [15]. BGPsim abstracts from the behavior of intra-domain routing protocols by only simulating the network on the Autonomous System level. It models the BGP decision state machine by asking questions whether a certain route is better than another route, whether a route should be accepted and whether a route should be advertised to a certain neighbor. The routing policy that we will use for our experiments will accept shorter routes over longer routes, breaks ties by preferring existing routes over new routes and advertises prefixes to neighbor such that the network remains valley-free, a concept explained in the topology section.

One of the inputs that the simulator requires is a list of events. The events that we will use for our simulation are BGP announcement events. In such an announcement event, it is specified what prefix is announced by what AS and at what time the event should be launched. The simulator will schedule these events and will try to adhere to the specified launch times. After all events have been launched, the simulator will stop its execution and output any results to a specified results directory. In order to let prefixes properly propagate throughout the network, a sleep event can be used to let the simulator continue running for a specified amount of time.

The simulator is a highly parallel program that runs on multiple computing nodes connected with TCP connections. Every computing node will simulate an equal amount of ASes. A special node is the coordinator node. The coordinator node takes care of synchronizing all computing nodes and deploying events to the computing nodes. The coordinator node does not simulate any ASes himself and is connected to all computing nodes using a TCP connection.

## 3.2 Network Topology

BGPsim requires a topology of the network as its input. Therefore, we require a topology of all the BGP speakers in the Internet. The Center for Applied Internet Data Analysis (CAIDA) is an independent analysis and research group based at the University of California's San Diego Supercomputer Center [16]. CAIDA provides Internet topology data in the form of AS relationships [17]. They use routing table snapshots from the Route Views project[1] and RIPE's Route Information Service (RIS)[2] to construct BGP paths. These BGP paths can be used in combination with the IANA's public list of AS assignments and RIPE's WHOIS database to infer relationships between ASes [18].

The relationships in the CAIDA topology file can be `CUSTOMER` to `PROVIDER` relationships and `PEER` to `PEER` relationships. The simulator uses these relationships to construct a graph of the network that it can use in its simulation. The simulator's routing policy will also use these relationships to decide whether a prefix should be advertised to a specific neighbor. The routing policy will adhere to the valley-free routing rule. This rule is based on the assumption that customer-to-provider links cost the customer money, whereas peer-to-peer links do not [19]. The rule prevents policy violations that could cause ASes to act as a transit for non-customer ASes.

## 3.3 Origin Validation

BGPsim has no means of doing origin validation built in. Unfortunately, the simulator has no API or framework that we can easily extend with our own origin validation plugins. Therefore, we have to edit the existing simulator codebase in order to implement origin validation. We will be drawing inspiration from a previous attempt on this by a MSc student who did not finish his work.

Every AS simulated in BGPsim uses a routing policy as explained in the BGPsim section. We are going to extend this by having every AS that does origin validation use a security policy as well. This security policy will override the functionality of the routing policy. However, we still allow the use of the routing policy in the case of a tie, for example. The security policy's inputs are the prefix, the current and the new route, the current and the new neighbor AS that advertised the route to us, the routing policy that would be used for this AS and the origin validity statuses of the current and the new route. The output of the security policy is whether the new route should be used over the current route, if we have a current route. Otherwise, the security policy will decide whether we accept the new route or drop it. All security policies will still always use the routing policy as explained in the BGPsim and topology sections for advertising their prefixes, to adhere to the valley-free rule.

The origin validity statuses of the two routes can be `VALID`, `UNKNOWN` or `INVALID`, as defined in chapter 2. Instead of going through the computationally-expensive cryptographic process to validate the origins of the routes, the origin validity status of each announcement is pre-defined in the events list given to the simulator as input. For each announcement event defined in the events list a field with the origin validity status tells the simulator what the results of doing origin validation on this BGP prefix announcement would be if we would go through the validation process. This way a lot of computations can be avoided without affecting the results of our experiments.

Two approaches to implement the retrieval of origin validity statuses were considered. The first of which is to have one or multiple validator units that BGP speakers can approach to get the origin validity status of a BGP announcement. The validators know the pre-defined validity statuses of prefixes and can return that information to querying BGP speakers. The second approach is to send along the validity status with the BGP announcement. Though, debatably the centralized approach using validators is closer to reality, the second approach does not require extra messages to be sent. Because the simulator is heavily parallelized already we felt adding central units with the added communication would be a bad idea. Therefore we went with the second approach and thus every BGP announcement carries the origin validity status with it. We can trust the origin of the BGP announcement to send the correct validity status, since we define this ourselves in the events list.

[1] `http://www.routeviews.org`

[2] `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris`

## 3.4 Security Policies

Every BGP speaker doing origin validation in the simulation is applying some security policy to decide what to do with an advertised route. We define four security policies based on policies used in practice and policies suggested in research.

### 3.4.1 The Hesitant Policy

The *hesitant* policy is based on the security policy used in the research to a market-driven deployment of BGP security [2]. We named it as so as it refers to BGP speakers that want to use the RPKI, but are hesitant to make big decisions using it. Therefore the results of doing origin validation are only used when the routing policy decides the two routes are of equal preference. In reality this could be two routes that have equal local preference values according business policies. In our simulations this means that two routes are of equal length. When routes are equally-preferred according to our routing policy, the hesitant security policy will prefer `VALID` routes over `UNKNOWN` routes over `INVALID` routes. In this policy routes are never dropped because of their origin validity status.

### 3.4.2 The Prefer Policy

The *prefer* policy is based on suggestions for policies by Juniper and Cisco [20][21]. The prefer policy prefers `VALID` routes over `UNKNOWN` routes over `INVALID` routes. Only when two routes have an equal origin validity status other routing properties will come into play. In reality this can be achieved by giving a higher local preference value to a more preferable origin validity status. Routes will never be dropped because of their origin validity status using this policy.

### 3.4.3 The Secure Policy

The *secure* policy is a lot like the prefer policy. It will prefer `VALID` routes over `UNKNOWN` routes and will only at a tie look at other routing properties. The difference is that the secure policy will drop all `INVALID` routes. Because routes are dropped there is a chance that parts of the network become disconnected. The secure policy is mentioned by Juniper [20] and thus might be used in practice. Therefore, looking at the effect on connectivity within the network is interesting for this policy.

### 3.4.4 The Strict Policy

The *strict* policy is a policy unlikely to be deployed in the Internet today. The policy accepts only `VALID` routes and thus drops any `UNKNOWN` or `INVALID` routes. Most routes to prefixes in the Internet today have an `UNKNOWN` origin validity status, since there exist no ROAs for those prefixes. Therefore, dropping all `UNKNOWN` prefixes can have a large negative effect on the connectivity of the network.

## 3.5 Deployment Strategies

In this research we look at the effect of simple deployment strategies on the security and performance of the network. When an AS does origin validation it will check the origin validity of incoming BGP announcements and use this information in its routing decisions. We will only deploy origin validation for transit networks, that is, networks providing transit to other networks. Stub networks, networks that only send and receive traffic destined for or originating from them, will never do origin validation. This is equal to the unidirectional version of origin validation mentioned in the research by Gill et al. [2]

The general idea behind this is that if all transit networks would do origin validation, all stub networks would automatically be safe since their provider does the origin validation for them. This dramatically reduces the amount of ASes that need to do origin validation for 100%

deployment. In our CAIDA network data from February this year we count 7,778 (15.5%) transit networks out of a total of 50,100 networks [22]. We define a network as a transit network when its transit degree is greater than 0. The transit degree of an AS is defined as the number of unique neighbors that an AS is observed to provide transit to in advertised BGP paths [17]. This data is provided by CAIDA. ASes with a transit degree of 0 are stub networks.

### 3.5.1 AS Rank and AS Customer Cone Size

We base our deployment strategies on CAIDA's AS rankings data. CAIDA ranks all ASes in the network based on their AS customer cone size [22]. The AS customer cone size is a metric of the influence that an AS has on the rest of the network. In "AS Relationships, Customer Cones, and Validation" by Luckie et al. the AS customer cone size of an AS is defined as the set of ASes that can be reached following only provider to customer links [18]. For an example AS $X$, the AS customer cone size includes $X$ its customers, its customers customers and so on, with some limitations. BGP paths are collected from public data sources and used to limit the amount of ASes within an AS its customer cone. An AS $Y$ is only included in $X$ its customer cone if a provider or peer of AS $X$ has a BGP path in which $X$ provides transit to $Y$. This is done because $X$ does not necessarily provide transit to all its indirect customers. An AS does not have to advertise all its BGP paths to its provider, since data over the provider link will cost the AS money. Furthermore, only BGP paths that AS $X$ advertises to its providers and peers are used to infer the AS customer cone size. This is because AS $X$ does not advertise all its BGP paths to providers and peers either, and the actual influence of AS $X$ on the network is thus better measured at peers and providers of $X$. A limitation of this method of inferring the AS customer cone size for ASes is that CAIDA only has access to best paths, rather than all BGP paths. Therefore the AS customer cone size of an AS might be underestimated.

### 3.5.2 AS Groups

Using the AS customer cone size we divide the group of all transit networks into five subgroups over which we will gradually deploy origin validation. These groups are roughly chosen to increasingly grow in size. We will name the groups tier 1, large tier 2, middle-sized tier 2, small tier 2 and tier 3 networks. The groups are summed up below. Lower end boundaries are inclusive, upper end boundaries are exclusive.

1. Tier 1: AS customer cone size of 5000+ (12 ASes)

2. Large Tier 2: AS customer cone size of 1000-5000 (26 ASes)

3. Middle-sized Tier 2: AS customer cone size of 100-1000 (207 ASes)

4. Small Tier 2: AS customer cone size of 10-100 (1,425 ASes)

5. Tier 3: AS customer cone size of 0-10 (6,108 ASes)

We will deploy origin validation isolated within these groups. For example, we will look at the effect of deploying origin validation only within the small tier 2 AS group. That is, transit networks with an AS customer cone size larger than or equal to 10 and smaller than 100. Within each AS group we will deploy origin validation by picking random samples from these AS groups. As a baseline we will also randomly deploy origin validation over the entire group of all transit networks. The effects of deploying origin validation within these AS groups can be compared to completely random deployment to assess the effectiveness of deploying origin validation in a structured way.

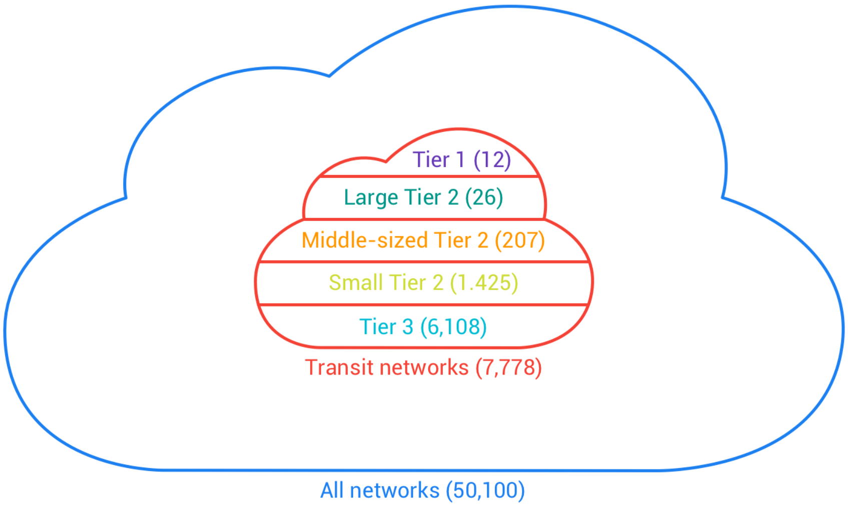

Figure 3.1: Chosen AS groups

### 3.5.3 Modeling the Current Status of BGP Security

Our final deployment strategy resolves around the current status of BGP security. We try to find the set of ASes that currently employ origin validation. Factual data about this is not available, since local policies used in BGP are mostly private to businesses. Therefore we use publicly available data from RPKI repositories to approximate the set of ASes that could do origin validation. Using the RIPE NCC RPKI Validator [23] we fetched a total of 15,731 ROAs that have currently been published in public RPKI repositories. From these ROAs we construct the set of all ASes that have published an ROA. We make the highly speculative assumptions that all ASes employing origin validation also publish ROAs and that the set of ASes doing origin validation is a subset of the set of ASes that publish ROAs. These assumptions are based on the idea that publishing a ROA is a small effort for a business, that cannot do much harm, while employing origin validation requires good care in order to prevent parts of the network becoming unreachable through the business its ASes. We use this set of ASes as a 7th AS group, deploying origin validation randomly in fractions in the same way as the other AS groups. The results of this experiment are interesting and relevant in the context of “what if“ current RPKI-aware ASes enable origin validation.

## 3.6 Experiments

Using the security policies and deployment strategies we will conduct experiments to measure the effects on the security and performance of the network. Both security and performance have their own experiments and measures. We will repeat all experiments for every deployment strategy and every security policy. All ASes doing origin validation will use the same security policy in a single experiment.

### 3.6.1 Security

To test the security of the network we will let two ASes announce the same prefix, one originating from network $A$ and one originating from network $B$. We assume that there exists a ROA for the prefix allowing network $A$ to originate the prefix, but there does not exist a ROA to allow network $B$ to originate the prefix. Network $B$ thus plays the role of the hijacker in this scenario where doing origin validation on network $B$ its BGP announcement would result in an `INVALID` origin validity status, whereas applying origin validation on network $A$ would give a `VALID` origin validity status. To measure the security of the network we will simply count the number of ASes in the entire network that have a route to the VALID origin as their best path versus the number of ASes in the network that have a route to the INVALID origin as their best path.s Also see figure 3.2.

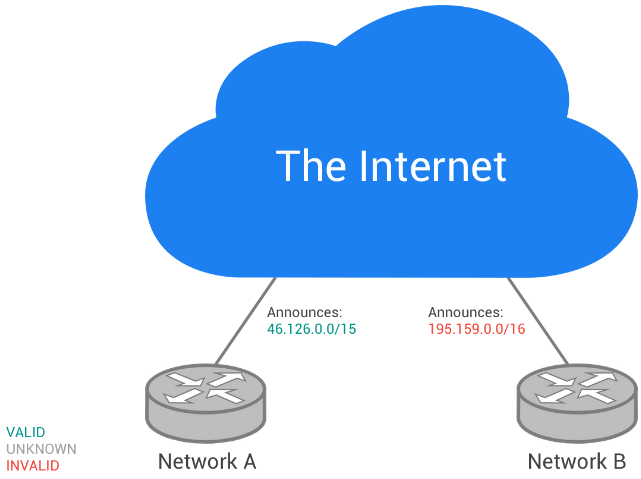

Figure 3.2: The security experiment using example prefixes.

### 3.6.2 Performance

For the performance experiment we will measure connectivity and path length. We will have a single AS announce three different prefixes. One of these prefixes will have a VALID origin, one will have an UNKNOWN origin and one will have an INVALID origin. For each of these prefixes we will count the amount of ASes that have an entry for that prefix. This gives us a measure of connectivity. We will also calculate the average AS path length for all AS paths within the network. If the average path length increases for a certain security policy or deployment strategy this will mean a downgrade in performance. Similarly, if the connectivity of certain prefixes decrease, it will mean a downgrade in performance. Also see figure 3.3.

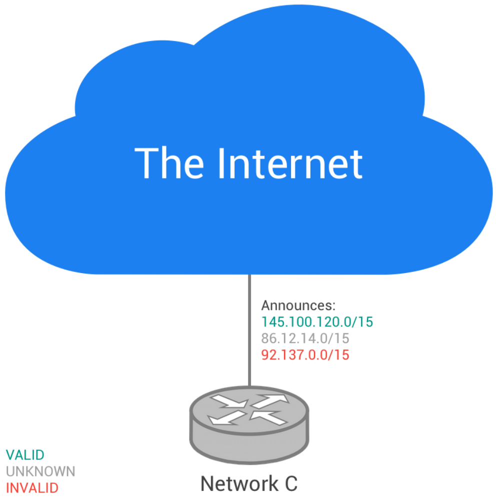

Figure 3.3: The performance experiment using example prefixes.

CHAPTER 4

# Implementation

BGPsim needs to be extended to support origin validation. We need to adjust the actual simulator code, instead of using some API or framework. The simulator is written in Java. We will adjust and extend the simulator code and create scripts to generate the experiments and parse the results.

## 4.1 Origin Validation

### 4.1.1 Validity Status

When an AS doing origin validation receives a BGP announcement, the AS needs to be able to retrieve the origin validity status of that announcement. As we have said before, the simulator does not need to actually do the cryptographic validation process. The origin validity status will be sent along with the BGP announcement. We have adjusted the simulator to support this. In the events file given as input to the simulator, a field is added called `validity`, which contains either the value `INVALID`, `UNKNOWN` or `VALID`. The BGP announcement carries a `Route` object, containing attributes such as the origin AS and the AS path. We have added a `validityState` field to this as well, so that the origin validity state is passed around from AS to AS along with the BGP announcement.

```xml
<ann showOnScreen="true">
    <asId>AS12593</asId>
    <announcements>
        <prefix>0</prefix>
    </announcements>
    <validity>INVALID</validity>
    <schedule class="explicit" launchTime="20000"/>
</ann>
```

Figure 4.1: An announcement event in the events.xml file

In the figure above an example is shown of an announcement event in the `events.xml` file with the `validity` field included. It represents a BGP announcement that should launch 20 seconds into the simulation. The BGP announcement is announced by AS 12593 and contains the prefix 0. In the simulator prefixes are represented as integers. Prefixes are equal when the two integer values representing them are equal. The `validity` field in this BGP announcement is `INVALID`. The simulator will interpret this as the enumeration value `INVALID` and pass it along with the BGP announcement.

### 4.1.2 Security Policies

The simulator uses the RoutingPolicy Java interface to decide whether routes should be accepted, are better than existing routes and whether routes should be advertised to certain neighbors. The RoutingPolicy interface contains two methods: isBetter(...) and isAdvertisable (...). The isBetter(...) method is used to decide whether a new route $B$ to a prefix is better than the currently existing route $A$. In the case that there is no existing route to the prefix, route $A$ has the value null. If the new route is not accepted by the isBetter(...) method while there is no existing current route to the prefix, the AS will remain having no route to that prefix. isAdvertisable(...) is used to decide whether a route $A$ existant in the AS its routing table should be advertised to a given neighbor $N$. This way an AS has full control of which neighbors receive which routes.

In order to be able to have ASes do origin validation using security policies we replace this RoutingPolicy Java interface with the similar SecurityPolicy Java interface. The SecurityPolicy interface contains the same methods as the RoutingPolicy interface, except for one extra parameter that each method receives, which is the original RoutingPolicy interface. See the figure below.

```java
public interface SecurityPolicy {

    public boolean isBetter(RoutingPolicy routingPolicy, ASIdentifier asId,
            Prefix prefix, Route newRoute, Neighbor newNeighbor,
            Route currentRoute, Neighbor currentNeighbor);

    public boolean isAdvertisable(RoutingPolicy routingPolicy,
            ASIdentifier asId, Prefix prefix, Neighbor neighbor,
            Neighbors neighbors, Route route);

}
```

Figure 4.2: The SecurityPolicy interface

The ASIdentifier parameter in both methods refer to the Autonomous System that makes the decision. The Neighbor object passed to isAdvertisable(...) is the neighbor to which the prefix would be advertised, if allowed so, and the Neighbors object is a list of all neighbors of the AS making the decision. The original RoutingPolicy object is passed along with the methods and is used in the security policies when necessary.

The four security policies mentioned in chapter 3 are implemented by implementing the SecurityPolicy Java interface. They follow the behavior as defined in chapter 3. The security policies for all ASes are managed by the SecurityPolicyManager. It knows which security policy is used for each AS in the network. For ASes that do not do origin validation there is the *ignore* security policy. The ignore policy simply forwards all decision to the original routing policy. The SecurityPolicyManager reads the security policies used by each AS from a security configuration file at the start of the simulation. In this security configuration file a list of ASes is given together with the security policies those ASes use. If an AS is not mentioned in this list, the simulator will not do origin validation and thus use the ignore security policy as a default.

### 4.1.3 Simulator Output

At the end of a simulation the simulator outputs its results to a specified results directory. For our experiments we need to know the best paths each AS has to the prefixes stated in the events list. Therefore, every AS writes its routing table containing only best paths to a file. There is one file for each node in the simulator. An example routing table in the simulator's output is shown in the figure below.

```xml
<routingTable as="AS9640" secureOrigin="false">
    <prefixes>
        <prefix num="0">
            <origin>AS4766</origin>
            <pathLength>3</pathLength>
            <path>
                <as>AS6539</as>
                <as>AS577</as>
                <as>AS4766</as>
            </path>
            <validity>VALID</validity>
        </prefix>
    </prefixes>
</routingTable>
```

Figure 4.3: A routing table in the simulator output

The routing table denotes the concerning AS in the `as` attribute and whether that AS did origin validation in the `secureOrigin` attribute. All prefixes for which the AS has an entry in its routing table are listed. The prefix number is denoted by the `num` attribute. The origin of the prefix, the AS path and the validity status are listed for each prefix. The length of the AS path is stated for each prefix as well, for convenience when parsing the results.

## 4.2 Generating Experiments

A simulator run requires a topology file, an events file, a security configuration file, which the simulator finds through the properties file. The properties file defines the working space, results directory, the topology, the events and security configuration filenames and several simulation parameters. We have two experiments to run, which we repeat for seven deployment strategies and four security policies. For each security policy and deployment strategy we run the simulator for several fractions of deployment: 0%, 10%, etc. Furthermore, we repeat every experiment 5 times with different randomly chosen samples. Given that the two experiments can be run simultaneously in one simulator run, we need to run the simulator a total of $7 * 4 * 11 * 5 = 1,540$ times and need to create the appropriate input files for it.

Our experiments are defined in the events file. Since we can run the two experiments in a single simulator run, we only need a single events file for all experiments. We also only need a single topology file, since all experiments will use the same topology. We thus only need to generate the security configuration files, in which we define what ASes do origin validation, and the property files in which we connect all input files together. The directory structure for our experiments is shown in figure 4.4.

For each deployment strategy we deploy origin validation randomly over a set of ASes. For example, for the tier 1 deployment strategy we deploy origin validation over the set of ASes with a customer cone size equal to or larger than 5,000. The order in which ASes are deployed for a deployment strategy is random within the set of ASes. Since this is repeated five times, there are 5 random orders in which origin validation is deployed for a single deployment strategy. However, for a single deployment order, every security policy uses that same deployment order.

```
experiments/
    <deployment_strategy>/
        <order_number>/
            <percentage>/
                <security_policy>/
                    properties.xml
                    security.xml

    topology.xml
    events.xml

example: experiments/100-1000/order-3/40/strict/properties.xml
```

Figure 4.4: Directory structure for our experiments

CHAPTER 5

# Results

We have done two experiments to measure the effects of several deployment strategies and security policies on the security and performance of the network. In the security experiment we let two ASes advertise the same prefix, one announces a `VALID` prefix and the other an `INVALID` prefix. We have randomly chosen two ASes out of the small tier 2 AS group to announce the prefixes: AS 6539 from Bell Canada announce the `VALID` prefix and AS 12593 from Ukrcom Ltd. announces the `INVALID` prefix. AS 12593 has a customer cone size of 21 and AS 6539 a customer cone size of 19, both ASes have a transit degree of 19. For the performance experiment a single AS will announce three different prefixes, one of which is `VALID`, one of which is `UNKNOWN` and of which is `INVALID`. The prefixes will be announced by AS 59524 from KPN B.V., which is a stub network.

We measure the results only at certain fractions of deployment for a single deployment strategy: 0%, 10%, . . . , 90%, 100%. To approximate values between those points we do linear interpolation. As explained in chapter three, every measurement point in our results is averaged over five random deployment orders. For the security experiments, we measure what fraction of all the ASes in the entire network have an entry to the `VALID` prefix as their best path. For the performance experiment we do the same, but for all three prefixes. We also measure the average path length for the performance experiment.

## 5.1 Random deployment

The first strategy to take a look at is the random deployment strategy. Here we randomly deploy origin validation over all 7,778 transit networks. The results of both experiments are shown in the figures below.

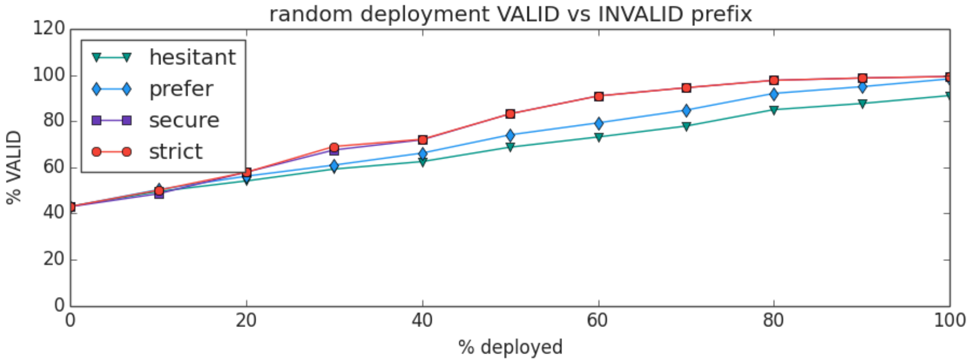

Figure 5.1: Fraction of ASes that use the VALID prefix as their best route for the random deployment strategy

The fraction of ASes receiving the VALID prefix at 0% deployment is 43% of the network. This is the baseline from which all deployment strategies shall start. The other 57% of the network will accept the `INVALID` prefix advertised by the other AS when no origin validation is used. Because origin validation is deployed over the entire network, we can see that the number of ASes that have an entry for the `VALID` prefix as the best path goes up to 100%. An exception is the hesitant policy, which can be explained with the fact that the hesitant policy does not have the origin validity status as its primary criteria for deciding best paths. It will only look at the origin validity status if the routes are of equal length. An interesting observation is that the secure and strict policies grow more secure more quickly than the prefer and hesitant policies.

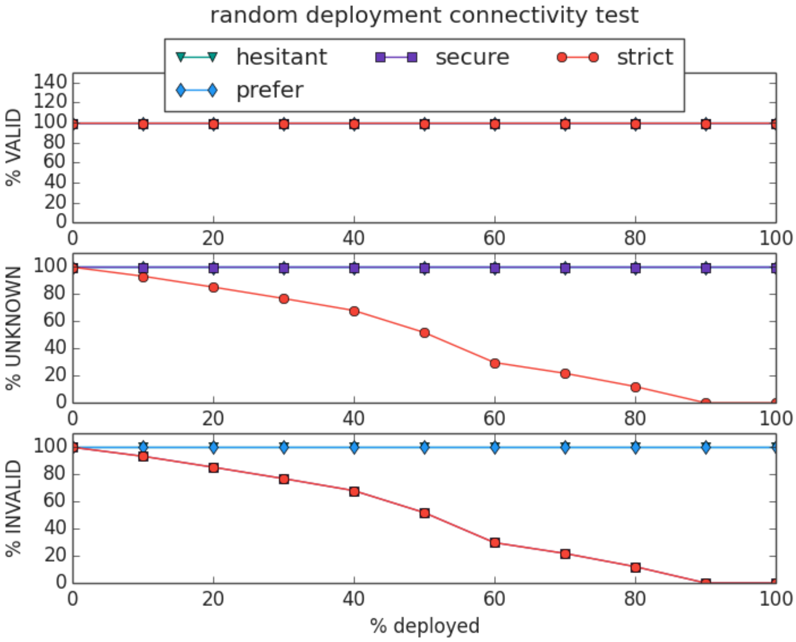

Figure 5.2: Fraction of ASes that have an entry for each prefix advertised in the performance experiment for the random deployment strategy

The connectivity of the `VALID` prefix will obviously stay at 100% for all security policies. The `UNKNOWN` prefix connectivity is only affected by the use of the strict policy. Using the strict policy will cause `UNKNOWN` and `INVALID` prefixes to completely disappear from ASes their routing tables at approximately 90% deployment. The secure policy has the exact same effect, yet only for `INVALID` prefixes.

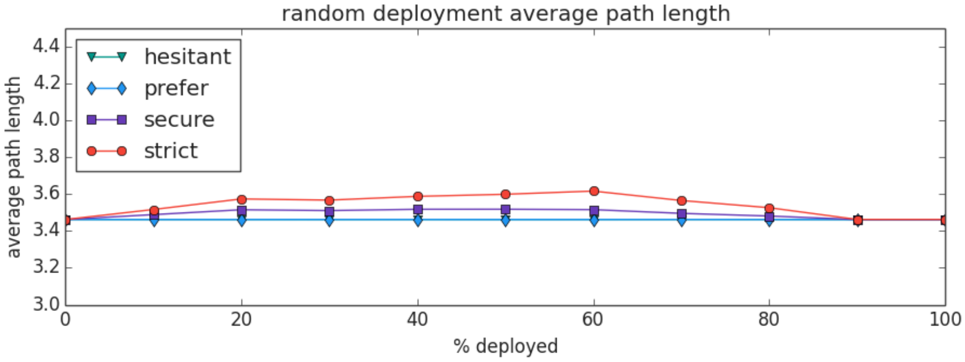

Figure 5.3: The average path length for the random deployment strategy

The average path length to the three prefixes advertised by AS 59524 will slightly increase and decrease again for certain policies as origin validation is deployed. Because the secure and strict policies drop prefixes, the average path length is affected. This can be explained with the fact that ASes have to use longer routes, since some routes might not be existent anymore. The average path length for all security policies converge to the average path length at 0% again at 90% deployment. As can be seen in figure 5.2, `INVALID` and `UNKNOWN` prefixes disappear completely from the network for the strict policy. Between 0% and 90% deployment, some paths to the `INVALID` prefix will not be available anymore. ASes that do not do origin validation will have to find other, possibly longer, routes to the `INVALID` prefix. Therefore, all paths after 90% deployment will be those to the `VALID` prefix, which causes the average path length to be the same as at 0% deployment. In the same way this can be explained for the secure policy.

We will use the random deployment strategy as a reference for future deployment strategies. Therefore, we present the table below containing interpolated and averaged values for the amount of ASes deploying origin validation in the upcoming deployment strategies. We average the values over all four security policies and use linear interpolation between measurement points.

| **#ASes doing origin validation** | **Security: VALID fraction** | **Performance: VALID fraction** | **Performance: UNKNOWN fraction** | **Performance: INVALID fraction** | **Average path length** |
|---|---|---|---|---|---|
| 0 | 43.0 | 99.4 | 99.4 | 99.4 | 3.46 |
| 12 | 43.1 | 99.4 | 99.4 | 99.4 | 3.46 |
| 26 | 43.2 | 99.4 | 99.4 | 99.3 | 3.46 |
| 207 | 44.7 | 99.4 | 99.0 | 98.5 | 3.47 |
| 1,425 | 55.2 | 99.4 | 96.1 | 92.7 | 3.50 |
| 2,561 | 60.2 | 99.4 | 93.0 | 86.7 | 3.50 |
| 6,108 | 92.3 | 99.4 | 77.9 | 56.4 | 3.49 |

Figure 5.4: Interpolated and averaged values for certain amounts of ASes deploying origin validation

## 5.2 Tier 3

We are now going to look at the effects of deploying origin validation within specified groups of ASes. The first group we are going to look at is the group of ASes that have a customer cone size bigger than or equal to 0 and smaller than 10. The results are shown in the figures below

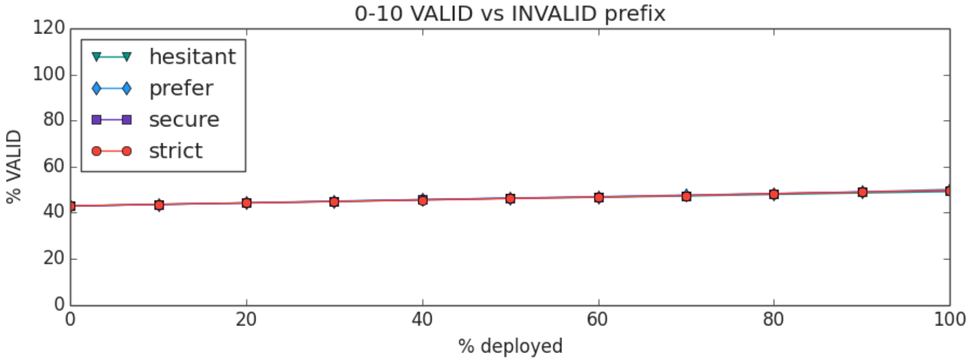

Figure 5.5: Fraction of ASes that use the VALID prefix as their best route for the tier 3 deployment strategy

We can observe that for all security policies the fraction of ASes using a path to the `VALID` prefix increases from approximately 43% at 0% deployment to 50% at 100% deployment. We deploy origin validation over 6,108 ASes. This is considerably worse than random deployment. After deploying origin validation over 6,108 ASes using the random deployment strategy, an average of 92.3% of the network used a path to the `VALID` prefix. We can also observe that for this deployment strategy it does not seem to matter what security policy we use for the network to be secure.

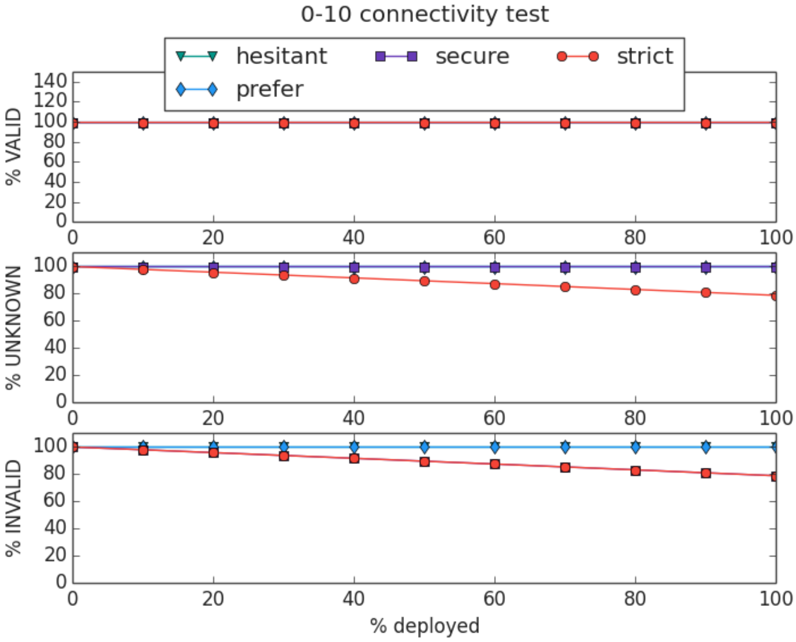

Figure 5.6: Fraction of ASes that have an entry for each prefix advertised in the performance experiment for the tier 3 deployment strategy

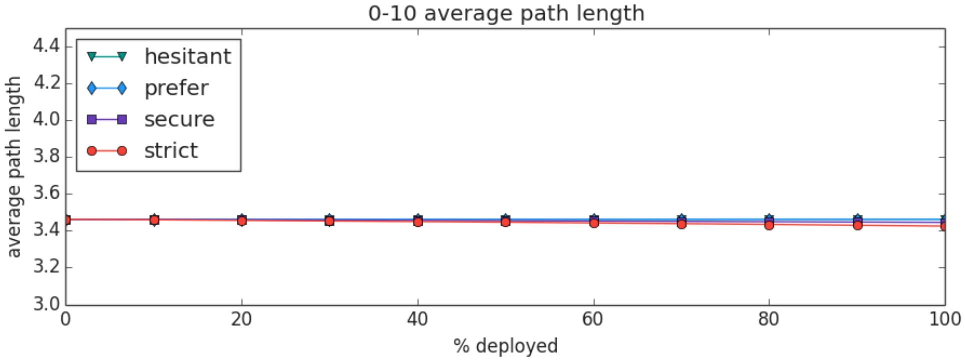

Figure 5.7: The average path length for the tier 3 deployment strategy

As expected we see a slight decrease in the connectivity of `UNKNOWN` and `INVALID` prefixes for the secure and strict security policies. The average path length remains approximately the same for all fractions of deployment and for all security policies.

## 5.3 Small Tier 2

The small tier 2 AS group includes all ASes with a customer cone size equal to or greater than 10 and less than 100. This group consists of 1,425 ASes.

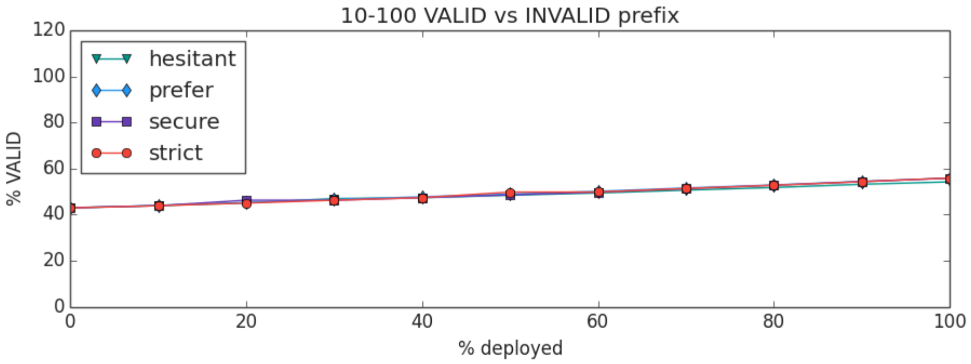

Figure 5.8: Fraction of ASes that use the VALID prefix as their best route for the small tier 2 deployment strategy

Deploying origin validation to the small tier 2 AS group causes an average increase of 12.4% in ASes using a path to the `VALID` prefix, starting at 43%, increasing up to 55.4%. This performs approximately the same as randomly deploying origin validation, which caused an average of 55.2% of the network to have a path to the `VALID` prefix after deploying origin validation to 1,425 ASes, as can be seen in figure 5.4. And again, it does not seem to matter what security policy is used. The results for the performance experiment are as expected, as can be seen in the figures below.

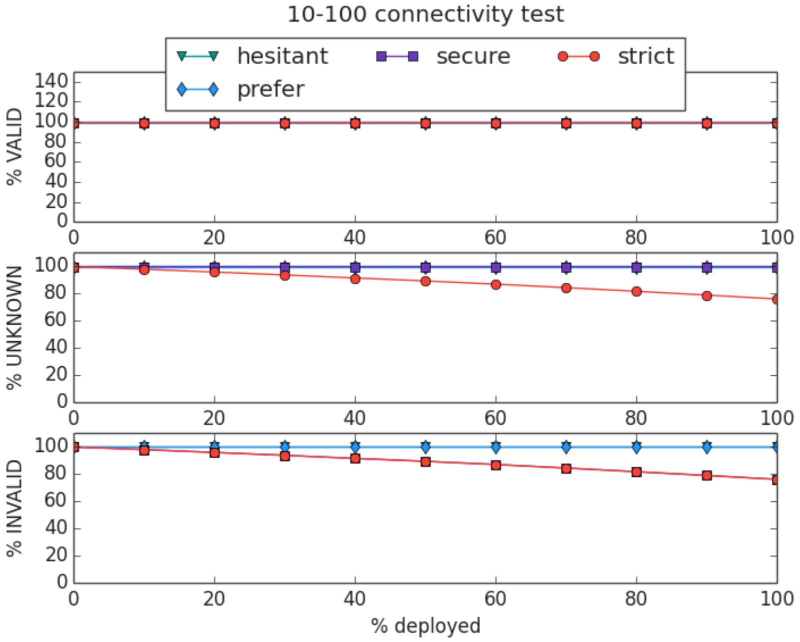

Figure 5.9: Fraction of ASes that have an entry for each prefix advertised in the performance experiment for the small tier 2 deployment strategy

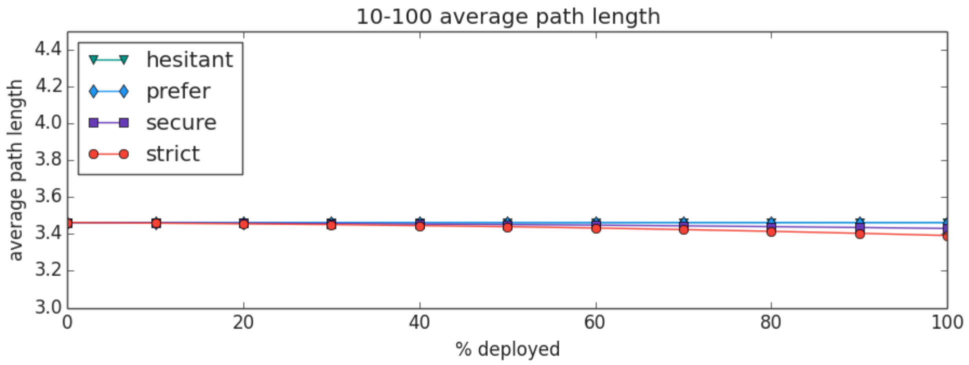

Figure 5.10: The average path length for the small tier 2 deployment strategy

## 5.4 Middle-sized Tier 2

The middle-sized tier 2 AS group contains all ASes that have a customer cone size bigger than or equal to 100 and smaller than 1,000. This AS group contains 207 ASes.

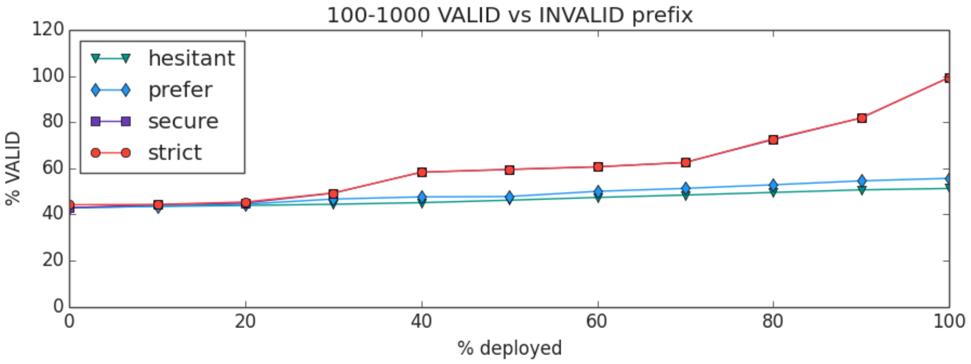

Figure 5.11: Fraction of ASes that use the VALID prefix as their best route for the middle-sized tier 2 deployment strategy

Deploying origin validation in the middle-sized tier 2 AS group has a large impact on the security of the network. Especially, for the secure and strict security policies. 100% deployment within this AS group using a secure or a strict policy leads to approximately 100% of the network having a route to the `VALID` prefix. Randomly deploying origin validation to the same amount of ASes only causes 44.7% of the network to have a route to the `VALID` prefix. The prefer policy reaches 55.6% of the network to have a path to the `VALID` prefix and the hesitant policy 51.2%. All are better than random deployment, but using the secure or the strict policy causes the entire network to use a path to the `VALID` prefix.

These results are unexpected, we did expect an improvement over the random deployment. However, we did not expect that the secure and strict policies could have such a large effect on the security of the network. It might have to do with the ASes we chose to advertise the prefixes, they are in the small tier 2 AS group. If they reach most of their connectivity through ASes in the middle-sized tier 2 AS group, doing origin validation in that layer can have a big effect.

For example, say that there are two ASes within the middle-sized tier 2 AS group, each of them only receives one of the two prefixes advertised. They advertise the routes to their neighbors, that do not necessarily do origin validation. If the prefer policy is used, the AS in the middle-sized tier 2 AS group that receives the `INVALID` prefix will still advertise it to its neighbors, since it never receives the VALID prefix. Neighbors not doing origin validation receiving both prefixes will simply pick the shortest route, which does not necessarily have to be that one to VALID prefix. However, if the secure or strict policy is used, the AS in the middle-sized tier 2 AS group receiving INVALID prefix will not accept it, and thus not advertise it to its neighbors. ASes that do not do origin validation therefore might only receive the path to the VALID prefix and thus accept that path in its routing table.

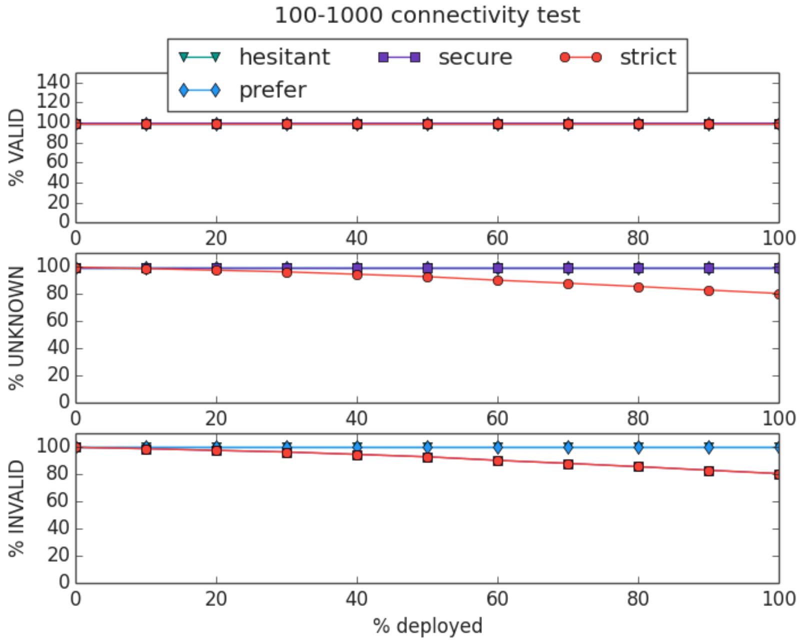

Figure 5.12: Fraction of ASes that have an entry for each prefix advertised in the performance experiment for the middle-sized tier 2 deployment strategy

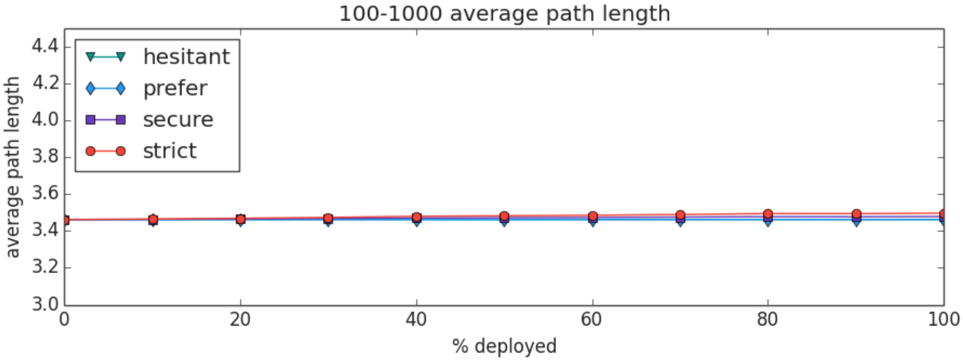

Figure 5.13: The average path length for the middle-sized tier 2 deployment strategy

Surpisingly, connectivity is not affected that much, despite causing a massive increase in the number of ASes choosing the VALID prefix in the security experiment. INVALID and UNKNOWN prefixes will still reach most of the network, but probably through a different, possibly longer, route. Even though we do see a very slight increase in the average path length for secure and strict policies, the increase is lower than expected.

## 5.5 Large Tier 2

The larger tier 2 AS group contains 26 ASes that have a customer cone size bigger than or equal to 1,000 and smaller than 5,000. The results of the two experiments are shown below.

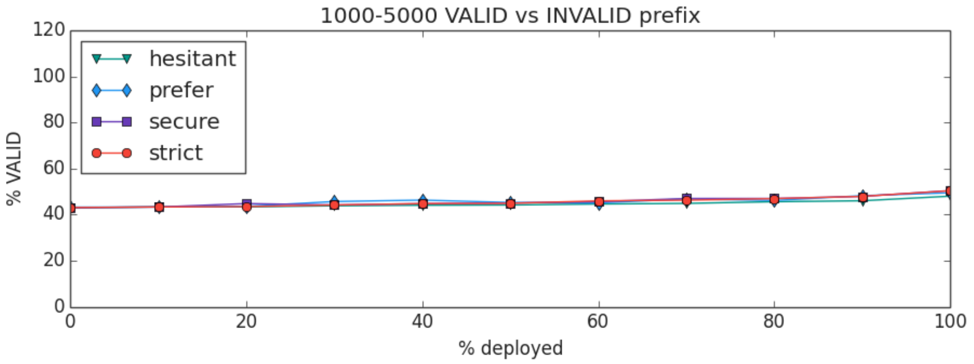

Figure 5.14: Fraction of ASes that use the VALID prefix as their best route for the large tier 2 deployment strategy

In contrast to the middle-sized tier 2 AS group this AS group has a significantly smaller effect on the security of the network. At 100% deployment an average of 49.5% of the network has a route to the `VALID` prefix, which is an increase of approximately 6.5%. This is better than random deployment, where only an average of 43.2% of the network chose the route to the `VALID` prefix when deploying origin validation over 26 random ASes, as can be seen in figure 5.4.

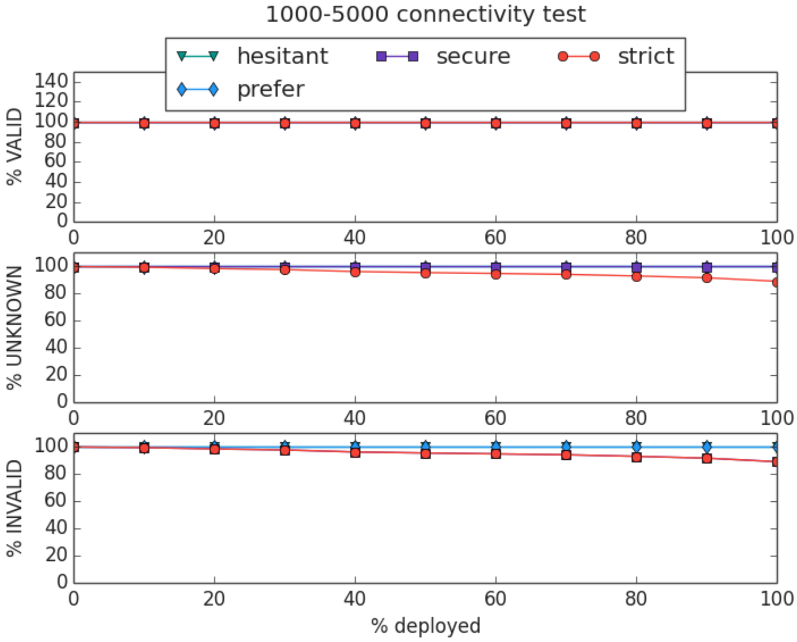

Figure 5.15: Fraction of ASes that have an entry for each prefix advertised in the performance experiment for the large tier 2 deployment strategy

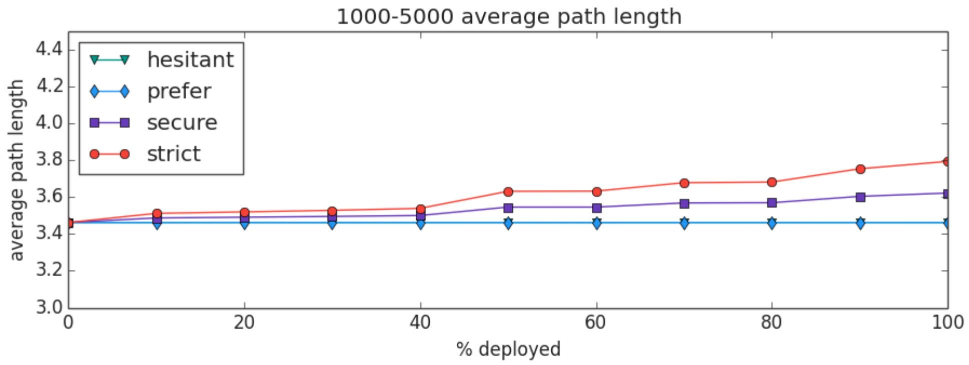

Figure 5.16: The average path length for the large tier 2 deployment strategy

Connectivity remains fairly high for the secure and strict security policies using this deployment strategy. We can see, though, that the average path length does increase rather significantly for these policies. Because certain prefixes are dropped, other longer routes have to be taken, increasing the average path length. This effect is bigger than we have seen in other AS groups. The network seems to depend on these ASes as a transit AS to reach their destinations using a short path.

## 5.6 Tier 1

The tier 1 AS group consists of ASes with the largest customer cone size in the network. These ASes have a customer cone size bigger than or equal to 5,000. There are only 12 ASes in this group.

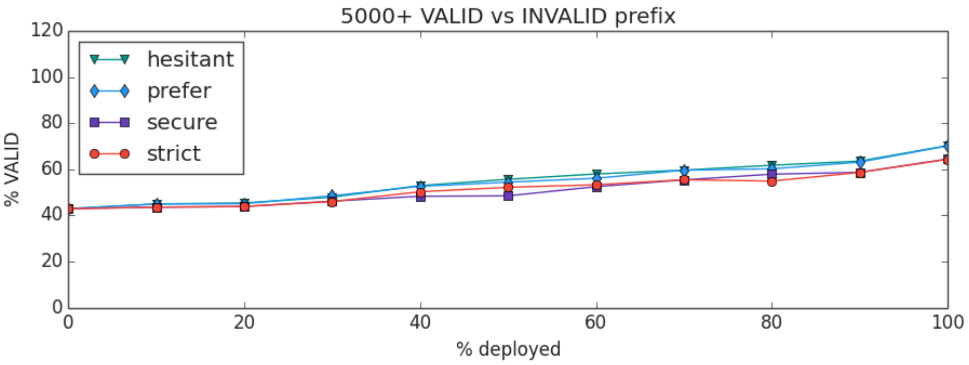

Figure 5.17: Fraction of ASes that use the VALID prefix as their best route for the tier 1 deployment strategy

Deploying origin validation within this AS group increases the fraction of ASes having a path to the `VALID` prefix from an average of 42.9% to an average of 67.3%. This is significantly better than random deployment, as expected. We did, however, expect to see a bigger effect on the network. We inspected how much ASes have a route through these top ASes and how many of them are to prefixes with a `VALID` origin and how many of them are to prefixes with an `INVALID` origin. We averaged the results over all security policies and all random deployment orders. The results are shown in the figure below.

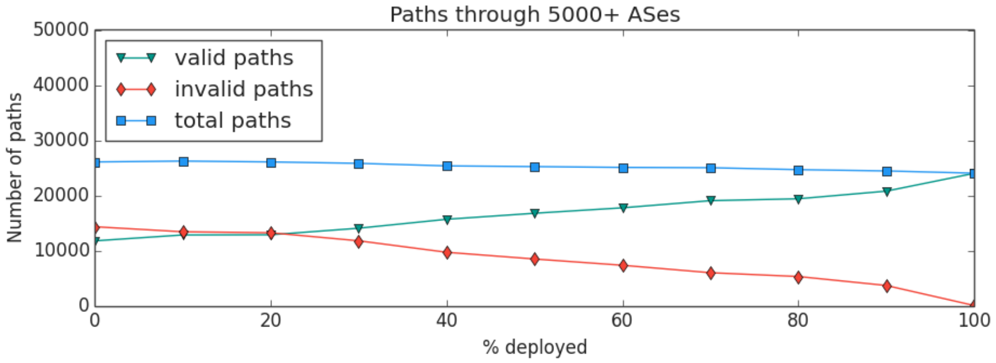

Figure 5.18: Average amount of paths through 5.000+ ASes

We can see that the number of paths going through ASes with a customer cone size equal to or larger than 5,000 decreases by approximately 2,000 out of a total of approximately 26,000 paths. Those paths are probably replaced by shorter paths not going through ASes in the tier 1 AS group. We start out with approximately 11,000 paths to the `VALID` origin and approximately 14,000 paths to the `INVALID` origin. Our maximum gain in ASes choosing a path to the `VALID` origin is thus approximately 14,000 paths. This is an increase of approximately 25% to 30% of ASes, which confirms our results in the security experiment.

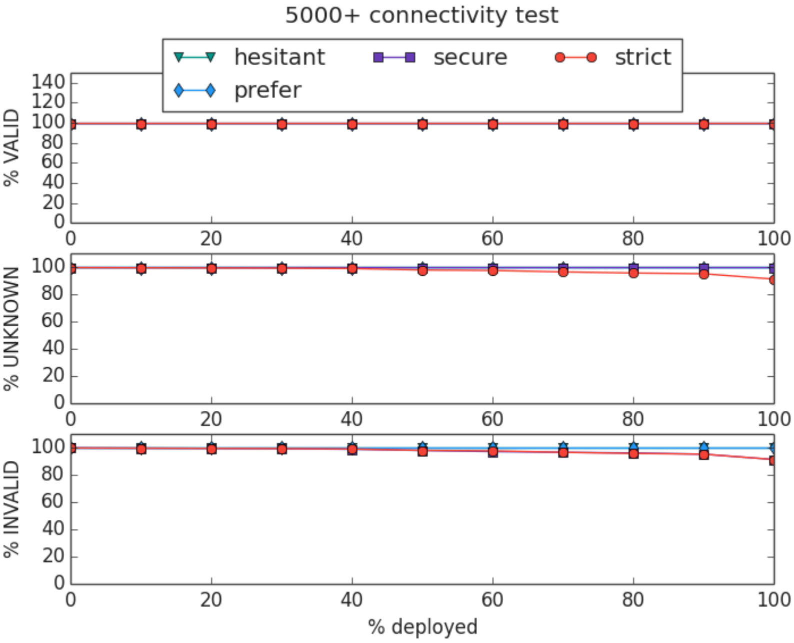

Figure 5.19: Fraction of ASes that have an entry for each prefix advertised in the performance experiment for the tier 1 deployment strategy

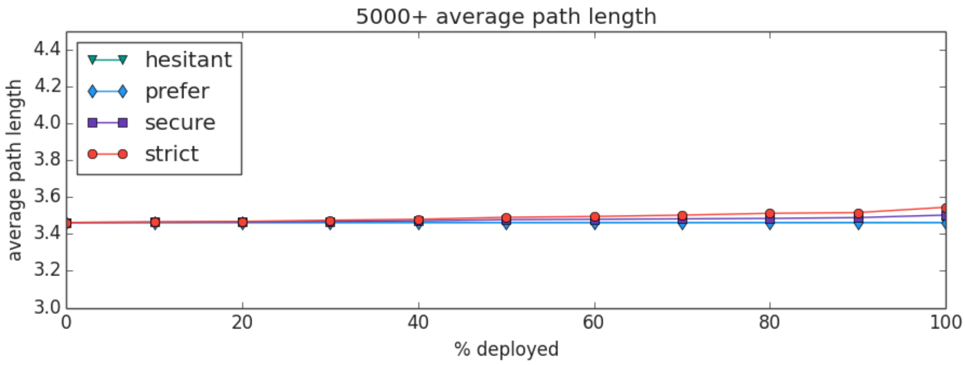

Figure 5.20: The average path length for the tier 1 deployment strategy

We only lose a small bit of connectivity for the `UNKNOWN` and `INVALID` using the secure and strict policies. Most paths will be replaced with one not going through the ASes in the tier 1 AS group. This does increase the average path length slightly.

## 5.7 Current status of BGP security

In the current status deployment strategy we only deploy origin validation to ASes if they have published a ROA in the RPKI. There are 2,561 ASes in this group.

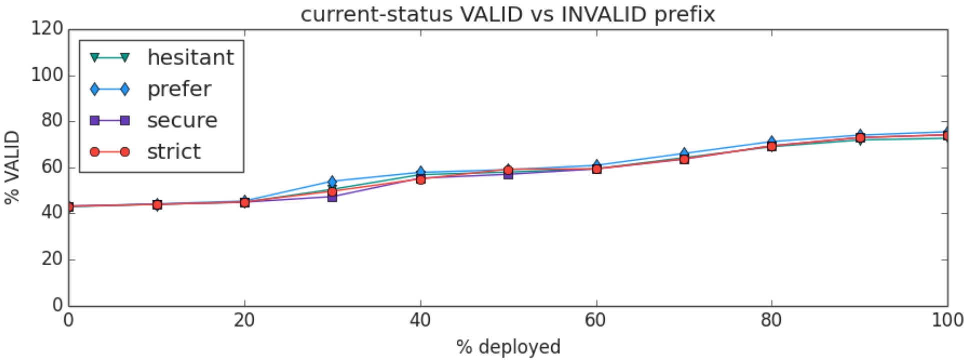

Figure 5.21: Fraction of ASes that use the VALID prefix as their best route for the random deployment strategy

The fraction of ASes choosing a path to the `VALID` prefix increases from an average of 43.0% to an average of 74.0%. This is better than when deploying origin validation over 2,561 random ASes, in which case on average 60.2% of the network would chose a path to the `VALID` prefix. We have looked at to which of the five customer cone size based AS groups these ASes belong to. We present the results of this in figure 5.21. It turns out 9 of these ASes have a customer cone size greater than 5,000, which there are only 12 of in the network. This explains why it performs so much better than the random deployment strategy.

| **AS group** | **Amount** |
|---|---|
| stub networks | 1,672 |
| tier 3 | 555 |
| small tier 2 | 268 |
| middle-sized tier 2 | 49 |
| large tier 2 | 8 |
| tier 1 | 9 |

Figure 5.22: AS group division for the current status networks

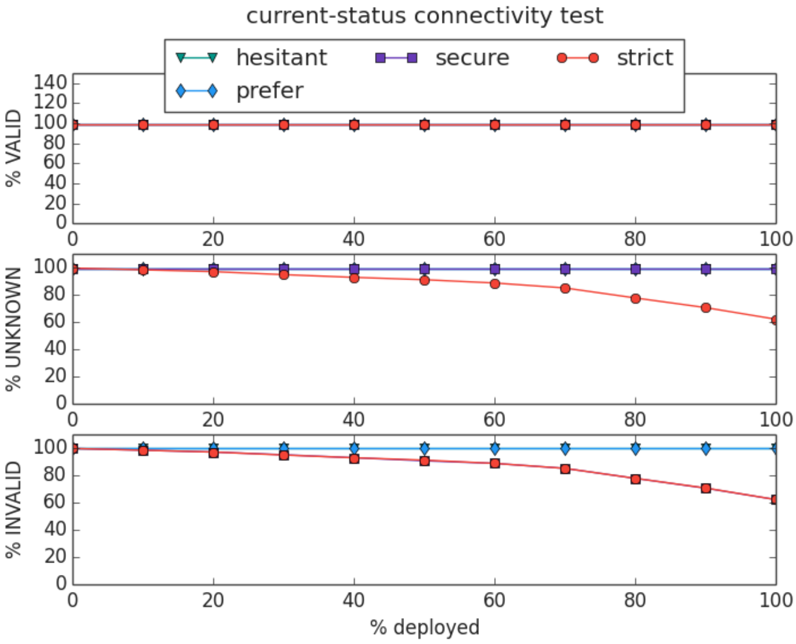

Figure 5.23: Fraction of ASes that have an entry for each prefix advertised in the performance experiment for the current status deployment strategy

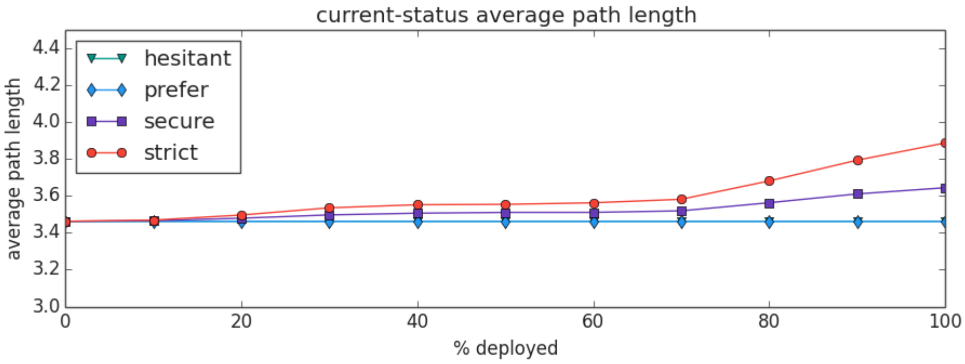

Figure 5.24: The average path length for the current status deployment strategy

# CHAPTER 6

# Discussion

In our results we have seen that deploying origin validation in a structural way can be better for the security of the network than random deployment. Deploying origin validation to ASes with a bigger customer cone size tends to have a larger impact on the network, as expected. Deploying origin validation to the tier 3 or small tier 2 AS groups only give slight improvements to security and perform no better or even worse than random deployment. The middle-sized tier 2 AS group has a significant effect on the security of the network. It is possible that this is caused by the choice of ASes that advertise the prefixes. It would require further research to support this claim. Our results suggest that a good deployment strategy should focus on deploying origin validation to the bigger ASes in the network, since larger groups of ASes with a small customer cone size have a smaller effect on the network's security than smaller groups of ASes with a large customer cone size have. Of course, all ASes in the network should publish ROAs for the prefixes that they advertise for any deployment strategy to be truly effective.

We have seen that the choice of security policy can affect the security of the network. In random deployment of origin validation on the entire network we see that the prefer policy acts slightly better than the hesitant policy, the secure policy acts slightly better than the prefer policy and the strict policy acts the same as the secure policy. Though, when only deploying origin validation within AS groups this effect does not seem to be visible, except for in the middle-sized tier 2 AS group. In the middle-sized tier 2 AS group the secure and strict policies have a significantly larger effect on the network's security than the other policies have. We argued that this can be caused by the advertising ASes depending on the middle-sized tier 2 ASes to spread their prefixes through the network. The connectivity test does not support this claim, but the connectivity test is done by a smaller AS than the ASes that the security experiment are done with. Therefore the effect can still be caused by specific choice of ASes that advertise the prefixes in the security experiment. Further research would be necessary to support this.

The secure and strict security policies have a negative effect on the connectivity and average path length of the network. This is caused by the secure policy dropping `INVALID` prefixes and the strict policy dropping both `INVALID` and `UNKNOWN` prefixes. However, the negative impact on the network's performance is rather small when deploying secure and strict policies within the customer cone size based AS groups. We see a larger, negative, impact on the network's connectivity and average path length for the large tier 2 AS group. We think that the network is relying on the bigger ASes in the network to allow them to traverse the network using a short route. We expect the effect to be bigger if we would combine the AS groups with larger customer cone sizes, yet further research would have to prove this.

According to our experiments the fraction of ASes chosing a path to the `VALID` prefix would nearly double if all ASes in the current status AS group would do origin validation. That is, all ASes that have published a ROA to the RPKI. The results suggest that using the hesitant or prefer security policies would give the best result, having approximately the same amount of security as the secure and strict policies without the drawbacks on performance that the secure and strict policies have.

The hesitant security policy performs better than we expected it to do. It performs only slightly worse than the prefer policy. We think that businesses will find it easier to adopt the hesitant policy than the other policies, since their current policies will still be the primary decisor in the route decision process. However, the routing policy we have used only looks at the path length of routes, and is not necessarily representative for policies that are used in the real world.

CHAPTER 7

# Conclusion

In this research we have tried to answer three research questions.

- *What is the impact on routing security for different origin validation deployment strategies?*
- *What is the impact on routing security for different origin validation security policies?*
- *What is the current status of routing security given the current publication and potential usage of RPKI data?*

We can conclude that doing structural deployment can have a larger effect on the network's security with fewer ASes doing origin validation. Our results show that deploying origin validation to small groups of ASes with large customer cone sizes have more effect than deploying origin validation to large groups of ASes with small customer cone sizes. Therefore, we suggest a top-down deployment strategy where the largest ASes deploy origin validation first. We base this conclusion on empirical data and provide no mathematical proof that this is the optimal deployment strategy.

The security policy used does have an effect on the security of the network. However, our results show that in general the differences are small. The use of the secure or the strict security policies can have a negative effect on the connectivity and average path length of the network. Results suggest that this effect is greater when deploying origin validation to the larger ASes in the network. Therefore, we would suggest the hesitant or the prefer security policies to be used along with the top-down deployment strategy. We base this purely on empirical evidence and do not provide mathematical proof that this is the optimal security policy to use. The use of the hesitant policy shows good results in our research. We must, however, keep in mind that our routing policy is not representative for the private business policies used in the real world. Therefore, we suggest the use of the prefer policy over the hesitant policy in the top-down deployment strategy.

Our experiments have shown that when the entire set of ASes currently publishing ROAs would deploy origin validation, the number of ASes choosing a path to the `VALID` prefix will nearly double to almost 80% of the network. This is probably caused by the large amount of large ASes within this set. We argue that it probably is not the case that all networks publishing ROAs are doing origin validation. We think it will be a subset of these networks, because we think that publishing a ROA is a small effort to a business compared to the effort of deploying origin validation. Public data about the policies used by businesses is not available to confirm this.

CHAPTER 8

# Future Work

The results of this research raises some questions that are left open. Future work answering these questions could provide more insights in how origin validation is best deployed. Deploying origin validation using a secure or strict policy in the middle-sized tier 2 AS group shows a great improvement in security in our experiments. This increase could be caused by the specific choice of ASes that advertise the prefixes in our experiments. Future work could show whether this claim is true or that deployment of origin validation within this AS group proves highly effective for securing the network. Future work could also look at the negative effects of using the secure or strict security policies when deploying origin validation using the top-down deployment strategy. Based on our results we expect that the negative effects of these policies on the connectivity and average path length of the network are bigger when deploying origin validation top-down than when doing random deployment.

We focused on the deployment of origin validation in this research. Another application of the RPKI is path validation, or BGPsec. In path validation it is important that chains of connected ASes all deploy path validation in order for the path to be secure. In our deployment strategies ASes that deploy origin validation do not have to be neighbors and can be scattered throughout the network. Path validation would most likely require different deployment strategies in order to be effective. Future work could research what deployment strategies would prove most effective for path validation and see how they relate to the deployment strategy that we have suggested for origin validation.

# Bibliography

[1] *RIPE - RPKI Deployment Monitor.* URL: http://rpki-monitor.antd.nist.gov/?p=2&s=0 (visited on 15/06/2015).

[2] Phillipa Gill, Michael Schapira, and Sharon Goldberg. "Let the market drive deployment: A strategy for transitioning to BGP security". In: *ACM SIGCOMM Computer Communication Review.* Vol. 41. 4. ACM. 2011, pp. 14–25.

[3] Y Rekhter, T Li, and S Hares. *RFC 4271: Border gateway protocol 4.* 2006.

[4] *Washington Post - The long life of a 'quick' fix.* URL: http://www.washingtonpost.com/sf/business/2015/05/31/net-of-insecurity-part-2/?utm_content=buffer29ac3&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer (visited on 01/06/2015).

[5] Geoff Huston and Randy Bush. "Securing BGP with BGPsec". In: *The ISP Column* (July 2011).

[6] Russell Housley et al. *Rfc 5280: Internet X. 509 Public Key Infrastructure Certificate and CRL profile.* 2008.

[7] Geoff Huston. "Resource Certification". In: *IETF Journal* 4.3 (2009), pp. 21–26.

[8] C Lynn, S Kent, and K Seo. *RFC3779: X. 509 Extensions for IP Addresses and AS Identifiers.* 2004.

[9] R Housley, J Curran, and G Huston. *D. Conrad," The Internet Numbers Registry System.* Tech. rep. RFC 7020, August, 2013.

[10] G Huston and R Loomans. *G. Michaelson," A Profile for Resource Certificate Repository Structure.* Tech. rep. RFC 6481, February, 2012.

[11] M Lepinski and S Kent. *RFC 6480: an infrastructure to support secure Internet routing. Internet Engineering Task Force (IETF).* 2012.

[12] M Lepinski, S Kent, and D Kong. *RFC 6482: A Profile for Route Origin Authorizations (ROAs). Internet Engineering Task Force (IETF), 201 2.*

[13] M Lepinski, A Chi, and S Kent. *RFC 6488: Signed Object Template for the Resource Public Key Infrastructure (RPKI). Internet Engineering Task Force (IETF), 201 2.*

[14] G Huston, R Loomans, and G Michaelson. *RFC 6487: A Profile for X. 509 PKIX Resource Certificates. Internet Engineering Task Force (IETF), 201 2.*

[15] Maciej Wojciechowski. "Border Gateway Protocol Modeling and Simulation (MSc. thesis)". In: (2008). URL: http://www.nlnetlabs.nl/downloads/publications/thesis_bgpsim.pdf.

[16] *CAIDA 2014-2017 Program Plan.* URL: http://www.caida.org/home/about/progplan/progplan2014/ (visited on 01/06/2015).

[17] *CAIDA AS Relationships data.* URL: http://www.caida.org/data/as-relationships/ (visited on 01/06/2015).

[18] Matthew Luckie et al. "AS relationships, customer cones, and validation". In: *Proceedings of the 2013 conference on Internet measurement conference.* ACM. 2013, pp. 243–256.

[19] Lixin Gao. “On inferring autonomous system relationships in the Internet”. In: *IEEE/ACM Transactions on Networking (ToN)* 9.6 (2001), pp. 733–745.

[20] *Juniper RPKI Documentation, Example: Configuring Origin Validation for BGP.* URL: `http://www.juniper.net/techpubs/en_US/junos12.2/topics/topic-map/bgp-origin-as-validation.html` (visited on 01/06/2015).

[21] *Cisco Documentation, BGP Commands: M through N.* URL: `http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/command/irg-cr-book/bgp-m1.html` (visited on 01/06/2015).

[22] *CAIDA AS Rank data.* URL: `http://as-rank.caida.org` (visited on 01/06/2015).

[23] *RIPE NCC RPKI Validator.* URL: `https://www.ripe.net/manage-ips-and-asns/resource-management/certification/tools-and-resources` (visited on 01/06/2015).