BGP security and the future

A meta-analysis of BGP threats and security to provide a new direction for practical BGP security

T.R. van Rossum

BGP security and the future

A meta-analysis of BGP threats and security to provide a new direction for practical BGP security

by

T.R. van Rossum

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on October 15, 2020 at 13:00.

Student number:	4246306	
Project duration:	March 1, 2019 – October 15, 2020	
Thesis committee:	Dr. C. Doerr,	TU Delft, supervisor, chair
	Dr. S. Picek,	TU Delft
	Dr. K. Hildebrandt,	TU Delft

This thesis is confidential and cannot be made public until October 14, 2021.

Abstract

The Internet consists of many subnetworks, which are connected to each other. These subnetworks are the autonomous systems (ASes) that make up the Internet: each hosts a part of it. In order to successfully determine routes from one of these ASes to the other, the Border Gateway Protocol (BGP) is used. This protocol has several security flaws however, and exploitation of them has lead to parts of the Internet being temporarily unreachable.

In order to combat these flaws, several security solutions have been developed already. However, none of these have been deployed on a wide scale yet. As such, this thesis focuses on the question: why not, and what can be done to protect BGP in the future? This thesis includes an analysis of the BGP threat landscape, to find which threats are most relevant, and to find out whether or not solutions have adapted to the threat landscape. It also includes a comparison of solutions on different practical security aspects. From this comparison, I found that no solution is able to prevent attacks if only one autonomous system deploys it. Due to this, I suggest to shift attention to detective security. This thesis also includes an analysis of some detective security schemes, to see which properties of these schemes can be used for a new scheme. This new scheme is designed to comply with a list of requirements, and it uses properties from three other schemes. Development of this scheme is left as future work. Altogether, this thesis should provide a new direction for the future of BGP security.

Preface

This thesis is the final work of my Masters degree in Computer Science at the Delft University of Technology. It is the documentation of the research that I have done for my final research project. It presents the literature research done in the field of BGP security, and provides an in-depth look into what has already been done, where the research field should focus on next, and why it should focus on these areas next.

I have a lot of people to thank for finishing this project. First of all, I would like to thank Christian Doerr, my thesis supervisor, who gave feedback when necessary and pushed me to work harder when necessary. I would also like to thank the other members of my thesis committee, Stjepan Picek and Klaus Hildebrandt, for taking the time to review my thesis and be part of my thesis committee. I would also like to thank my family and friends for their support. Finishing this research has been stressful, and the support I got made all the difference. Finally, I would like to thank the Cyber Security group of the Delft University of Technology. We had a lot of great conversations, lunch breaks and coffee breaks, which also helped me a lot when I needed it.

T.R. van Rossum
Delft, October 2020

Contents

List of Figures

List of Tables

1 Introduction

The Internet, in essence, is a collection of autonomous systems, which are subnetworks consisting of servers, routers, cables, et cetera. Each autonomous system hosts a piece of the Internet for users of the Internet to access. Of course, every part of the Internet has an address, the Internet Protocol address, or IP address for short. But without a route to that address, routers would not know how to connect to that address. Finding the part of the Internet that users are trying to connect to would be similar to finding a mail address in a city when one only has the house number and street, and no navigational aid whatsoever. Of course, routers could simply send data to every other router that they are connected to in the hope of delivering the data to the right address. But a more efficient solution would be to define a route that the data should be sent along. This route creation is performed by a protocol called (e)BGP.

1.1. BGP

The (external) Border Gateway Protocol, or (e)BGP for short, is an interdomain routing protocol that runs over TCP. The primary use is to exchange network reachability information between autonomous systems (or ASes for short) over links that connect one AS to another [125]. These links between ASes are called external links. Aside from eBGP, there is also iBGP, which stands for internal Border Gateway Protocol, which is an intra-AS protocol designed to allow for the exchange of reachability information within an AS, logically over links within the AS, also called internal links. Generally, however, in the context of the Internet, and as such, in this thesis, when BGP is mentioned, the subject is eBGP rather than iBGP.

1.1.1. BGP route exchange and selection

The way the exchange works on a high level is as follows: each AS that hosts part of the Internet has a number (for identification purposes) and claims ownership of one or multiple subnetworks, which are ranges of IP addresses. These ranges are called IP prefixes and are denoted as an IP address with a forward slash and then an integer after it, such as 192.168.0.0/16 or 156.0.0.0/8. The number after the IP address denotes the number of first bits that are constant for that prefix and is called the length of the prefix. In the case of 156.0.0.0/8, the /8 means that the first eight bits of the IP address are constant, and the other 24 bits are variable. So the prefix 156.0.0.0/8 represents the set of IP addresses which goes from 156.0.0.0 to 156.255.255.255. While, in theory, the length of a prefix can be any number from 0 to 32, in practice, only the numbers between 8 and 24 are used. The ASes exchange which IP addresses they own with other ASes, as well as routes that they have to certain IP prefixes. If an AS broadcasts prefix ownership, then it does so by announcing that it owns the prefix and creating a route towards the prefix consisting of just their AS number. If the AS propagates a route, then it prepends its AS number to the route that they have to the prefix. Messages that carry this information are BGP messages and are transferred over TCP/IP from AS to AS. An example of this exchange is shown in figure 1.1. In this image, AS 6 owns prefix 12.34.0.0/16. Ownership of this prefix is broadcasted by sending the prefix along with AS number 6 to all the ASes that are connected to AS 6. In this case, prefix ownership is sent only to AS 5, which now knows that AS 6 owns the mentioned prefix. AS 5 then prepends its number to the route and sends to neighbouring ASes that the route to 12.34.0.0/16 is along AS 5 to AS 6.

Every router has its policies for selecting the routes that they use for forwarding data, which are called lo-

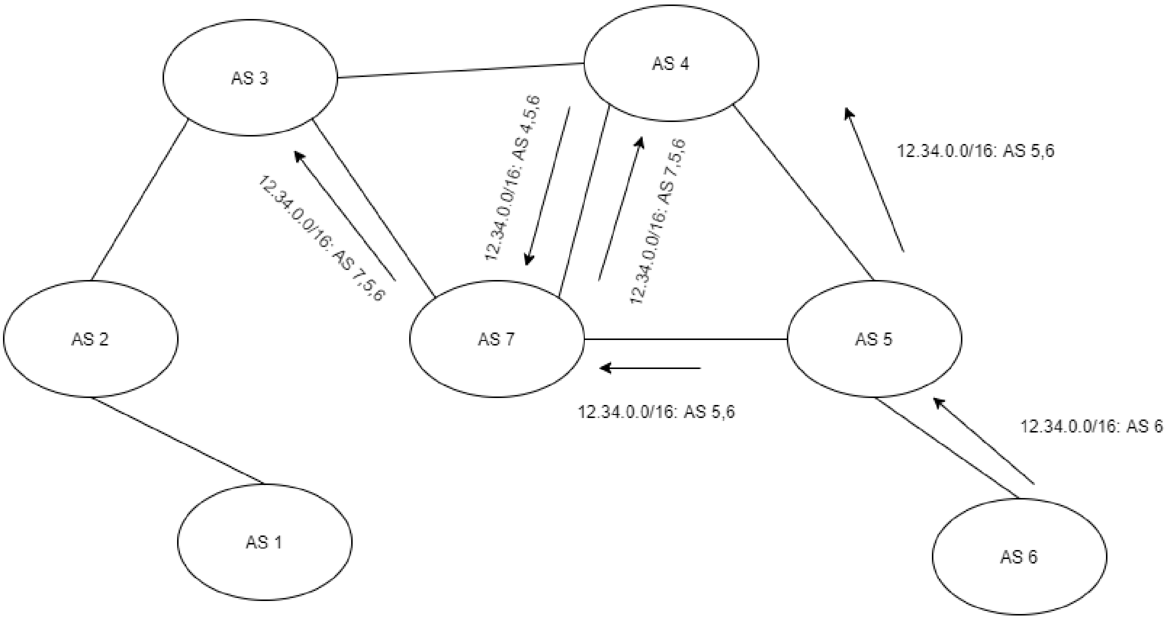


Figure 1.1: An example of the propagation of prefix ownership as well as routes towards a certain prefix. AS 6 owns prefix 12.34.0.0/16, broadcasts this ownership, and ASes that receive the message prepend their number and propagate the route.

cal policies. The rules in these policies that are most important for this thesis are as follows [4]:

- When routers have the choice between two routes to a certain prefix, they will store the shorter one and discard the longer one. In general, the shorter path is the path with the least hops. It is also possible to measure the length of a path by how long the links between ASes are in the real world, but that is used far less frequently in interdomain routing.
- If routers receive a route that leads to a prefix that is a more specific version of a prefix that they already have, it will store and forward both routes. When the destination IP address for data matches both prefixes, it will be sent along the path to the more specific prefix. Example: the router has a route to prefix 12.34.0.0/16 and receives a route to 12.34.168.0/24.

1.1.2. BGP export policies

Aside from these selection policies, there are also export policies. These export policies generally reflect the relationship between ASes in the real world. In the real world, AS-pairs represent a customer-provider relationship (such as an ISP providing paid Internet service to an AS) or peer-peer relationship (where neither AS serves the other) [72]. Logically, providers will only forward data coming from ASes that are their customers due to financial motives. Gao and Rexford have formulated three rules for BGP export policies which can be used in modeling BGP, which deal with the fact that there are providers, peers or customers in BGP in the real world [71]:

1. When a customer AS exchanges routing information with a provider AS, the customer AS can export both its routes as well as routes of its customers. The AS does not export routes learned from other providers or peers. Because of this, an AS doesn't provide transit services to its provider. This practice makes financial sense as provider ASes do not pay customer ASes for transit services.
2. When an AS exchanges routing info with a customer AS, an AS can export all of its routes. This set of routes includes those learned from providers and peers. This means that an AS does provide transit service for its customers, which makes sense because that is what the customers pay an AS to do.
3. When an AS exchanges routing information with a peer, the same export rules apply as in the case of exchanging routing information with a provider. It can thus be inferred that an AS does not provide transit services to its peers, but only to its customers.

From these export rules, one can logically deduce the following rules regarding importing and exporting routes:

1. When an AS receives routes, it will receive all the routes if the sending AS is a provider. If the sending AS is not a provider, then the receiving AS only receives routes of the sending AS or of the customers of the sending AS.

2. An AS only exports all of its routes to a customer AS, and exports only its own routes as well as those of its customers to provider or peer ASes.

1.2. BGP weaknesses

Back when BGP was developed in the 1990s, the main concern was to develop an interdomain routing protocol that would do what such a protocol is supposed to do: link subnetworks to one another to create a global network. BGP was also designed during a time where the Internet was more homogenous, a lot smaller than it is nowadays, and when Internet usage was not commonplace, nor was it expected to become commonplace. The risk of redirecting traffic by sending false BGP messages was never considered as a serious threat to the availability of a critical infrastructure because the Internet was not a critical infrastructure back then.

Because BGP was developed in a time when the Internet was less of a necessity, the threat of data being redirected to (for example) prevent it from reaching its intended destination was not considered to be important enough to prevent it from happening. As such, the protocol itself does not require nor perform any validation of data that has been propagated using the protocol. Consequently, malicious ASes could alter route data and propagate fake routes through the network, claim that they own a prefix that they do not own, and individual BGP speakers can claim to represent an AS that they are not part of. BGP is vulnerable to attacks that alter the sent data in any way.

1.3. Attacks against BGP

As mentioned before, because BGP requires no validity checks for propagated routes or ownership claims of certain prefixes, the protocol has several vulnerabilities. These vulnerabilities can be exploited, either maliciously or by negligence (such as accidentally misconfiguring a router). The following attacks all make use of the vulnerabilities in BGP.

1.3.1. Prefix hijacks

A prefix hijack happens when one AS advertises owning a prefix that it does not own, intending to redirect traffic from other ASes that was intended to go to the hijacked prefix to the AS that hijacked the prefix. This attack is possible due to BGP requiring no proof of ownership of a prefix for an AS to advertise that it owns a certain prefix. Recall from the previous section that an AS will generally select the route to a prefix with the shortest number of hops, given that there are no financial motivations at play (no customer-provider relationships, all AS pairs are peers to one another). In this case, ASes will select the invalid route if the AS that performs the hijack is closer to them in terms of hops. An example of a prefix hijack is shown in figure 1.2. AS 2 and AS 3 in this figure will select a false route toward the prefix 12.34.0.0/16, while the other ASes in this example will select the correct route. Even if AS 4 or AS 7 propagates the legitimate route to AS 3, the number of hops in the correct route will be greater than the number of hops in the already selected incorrect route. This, combined with no way of authenticating which route is correct or which AS actually owns the mentioned prefix, means that some ASes will select a wrong route towards a prefix while others will select the correct route.

When financial motivations are considered, and some AS pairs are customer-provider pairs, route selection in the case of a prefix hijack is a little more complicated as another factor is at play [56]. Table 1.1 shows what happens when an AS already has a route to a prefix and receives a different one. All valid/invalid route type combinations are considered. Logically, the attack is most likely to succeed in general if the original, valid route is a route towards a provider (as using that route would cost money). In contrast, the least likely attack to succeed is a route towards a customer (because using that route would earn money).

A variant of the prefix hijack is the subprefix hijack, which can be even more effective than a regular prefix hijack if a series of subprefixes is announced that combined makes up the original prefix. Recall that traffic for a certain prefix is forwarded along the route with the most specific prefix that matches the destination IP address of the traffic. A subprefix hijack involves an AS advertising ownership of a prefix that is more specific

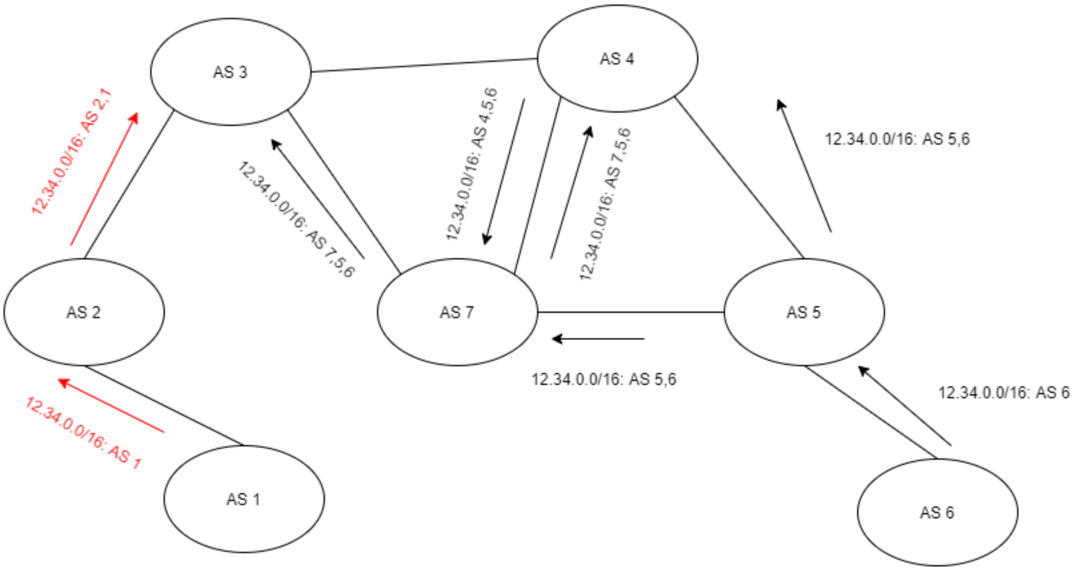


Figure 1.2: An example of a prefix hijack. Here, AS 6 is the legitimate owner of 12.34.0.0/16 while AS 1 only claims to own 12.34.0.0/16.

Valid route	Invalid route	Result
Customer	Customer	Decision based on the length of the routes, potential hijack
	Peer	Receiving AS prefers valid route, no hijack
	Provider	Receiving AS prefers valid route, no hijack
Peer	Customer	Receiving AS prefers invalid route, hijack successful
	Peer	Decision based on the length of the routes, potential hijack
	Provider	Receiving AS prefers valid route, no hijack
Provider	Customer	Receiving AS prefers invalid route, hijack successful
	Peer	Receiving AS prefers invalid route, hijack successful
	Provider	Decision based on the length of the routes, potential hijack

Table 1.1: how different AS relationships influence the success rate of a prefix hijack. Logically, customer routes have a higher preference because these routes generate revenue.

than another prefix. For example, if one AS owns the prefix 12.34.0.0/16, and has advertised ownership of this prefix, then another AS executing a subprefix hijack can advertise ownership of the prefix 12.34.0.0/17. This announcement causes the other AS to hijack traffic that was intended to go to 12.34.0.0/16 if the destination IP address also matches 12.34.0.0/17. This announcement would not hijack all the traffic that would go towards 12.34.0.0/16. To combat this, instead of only advertising one more specific prefix, the AS can advertise multiple more-specific prefixes until the set of IP addresses represented by the prefix 12.34.0.0/16 is covered by all the false prefix ownerships that have been advertised. For example, a malicious AS would only have to advertise 12.34.0.0/17 and 12.34.128.0/17 to effectively advertise ownership of 12.34.0.0/16, and due to how BGP routing works, data could be routed along either of the routes to the malicious AS. A receiving router could in theory aggregate the prefixes if it detects that both subprefixes have the same origin, which renders the attack useless if the false route of the aggregated prefix is longer than the legitimate route, but this could be circumvented by two ASes working together by executing subprefix hijacks, one advertising one more specific prefix and the other advertising the second one.

One of the most well-known examples of a (sub)prefix hijack in the real world is the Pakistan Telecom incident of 2008 [50]. In February 2008, in response to an order of the government of Pakistan to block access to YouTube in the country, Pakistan Telecom started advertising that it owns the prefix 208.65.153.0/24 to its provider, PCCW. For the record, the prefix that YouTube was using is 208.65.152.0/22. It should be noted that YouTube no longer uses this prefix. It now uses the prefix 196.49.26.0/24 [45]. The apparent reason for this hijack was to prevent the citizens of Pakistan from seeing a trailer to an anti-Islamic film that was made by the Dutch politician Geert Wilders [36]. The intent was to hijack (a portion of, as it is a subprefix hijack) the traffic from Pakistan to YouTube. However, PCCW propagated the announcement not only to Pakistan but to the rest of the world. Traffic around the world that was meant to go to YouTube was now being sent to Pakistan. This caused the hijack to have a global impact. The hijack did not take very long, as YouTube reacted after about 80 minutes, and the false routes were withdrawn after slightly over two hours [51]. Still, it is one of the most frequently mentioned incidents of prefix hijacking today. The fact that sometimes governments of countries execute these kinds of attacks indicates that nation-state actors are to be considered when looking at potential attackers in the realm of attacking BGP.

As an example of another BGP hijack most likely executed by a nation-state actor, the state-owned Telecommunications Company of Iran hijacked Telegram traffic in July 2018 [26]. The Iranian government had banned using Telegram in the country, but people kept using the messenger app as it allowed them to send encrypted messages. One reason to believe that the Iranian government had ordered this hijack is that the hijack happened a day before proposed protests were to happen over the economic crisis of the country [27]. This is not the first time that the Iranian telecommunications company tried to block a part of the Internet being available for Iranian citizens. In January 2017, to comply with the strict Internet censorship laws of the country, the company redirected traffic intended to go to adult websites to its AS in order to block access to those websites in the country [25]. The telecom company did this by announcing that it owned the prefix of the AS that hosted these websites. The intent was to block access within the country, but just like with the Pakistan Telecom incident, the hijack leaked, and Internet users from Russia to Hong Kong were affected. Iran, similar to China with their Great Firewall [23], has very strict Internet censorship laws, and the government has even built its own state-controlled Internet [28], which is faster for Iranian citizens.

Another well-known incident, which in contrast with the Pakistan Telecom incident was completely accidental and not caused by nation-state actors, is the AS 7007 incident [49]. This incident was caused by AS 7007 accidentally leaking a large part of its routing table to the Internet, creating a black hole for traffic. This particular AS most likely had a bug in the affected router, and the leaked routes were deaggregated to prefixes of length /24 (which are, generally, the most specific prefixes that are used for most routing operations). They had also replaced the path to just 7007. This incident caused a lot of traffic to be redirected to AS 7007, as routers got routing information that indicated that AS 7007 was the source of a lot of very specific prefixes.

1.3.2. Path altering

Path altering is the act of, either deliberately or by negligence, altering a broadcasted path. BGP itself provides no integrity checks for altered paths, and as such, ASes can alter paths without much interference. In a way, prefix hijacking can be seen as a form of path altering, as it broadcasts an altered path towards a certain prefix. Altering a path can result in blackholed traffic because the traffic is routed along a path that misses a link and so the data cannot go further. But it can also allow malicious ASes to snoop traffic by routing it through their routers (which can lead to a confidentiality problem if there is no traffic encryption). It can also prevent a certain legitimate path being taken by adding ASes to the path, making the pathway too long for most routers to consider taking, allowing for a different, illegitimate path to be selected. This last one can be done very easily by having an AS prepend its own AS number many times over to make the path infeasibly long. An example of a path altering attack is shown in figure 1.3. In this example, AS 3 receives updates for the prefix 12.34.0.0/16, with the path (7,5,6) and (4,5,6). According to BGP propagation rules, it should propagate the message to its neighbours with its own AS number prepended, so it should propagate either (3,7,5,6) or (3,4,5,6), depending on which route AS 3 selected. However, in this case, it decides to alter the path and propagates the path (3,6) to its neighbours. In this case, all of its neighbours receive and store the false path as that path is shorter than the one they have, or it is the only one they have. AS 3 can do one of two things: it can decide to set up a virtual connection between itself and AS 6 and route traffic to AS 6 along itself (gathering information about the traffic to AS 6 in the process), or it can decide to drop all the traffic that was intended to go to AS 6.

There is significantly less media coverage about path altering and traffic rerouting attacks. This lack of media coverage could be because these attacks are less frequent or that we simply don't know whether or not these attacks are happening as traffic can still arrive at the intended location if traffic was only rerouted. As an example of such an incident happening, in June 2019, a large part of European telecommunications was rerouted along China Telecom. This happened because Swiss co-location company Safe Host (that hosted data centres) leaked routes to China Telecom. Then China Telecom proceeded to announce these routes onto the Internet, redirecting a lot of traffic through the AS of China Telecom [47]. Among the most impacted networks were the Swiss Swisscom, the Dutch KPN, and the French Bouygues Telecom and Numericable-

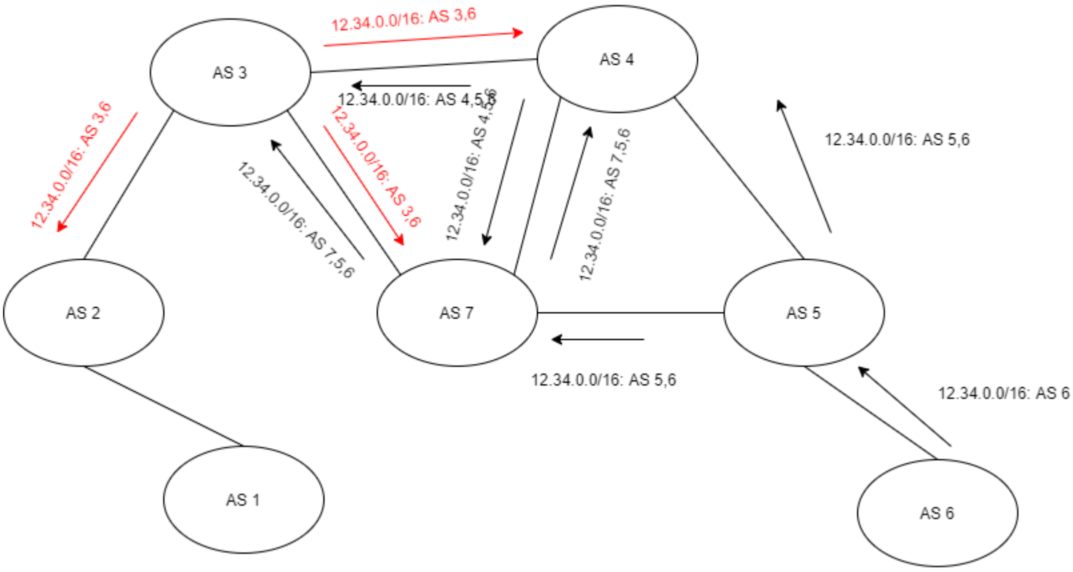


Figure 1.3: an example of a path altering attack. In this example, AS4 is the culprit, as it propagates a different path than intended.

SFR. What is interesting is that the leaked routes had the AS number of Safe Host prepended many times over, probably with the intent of preventing leaks from causing reroutes. It is unclear why these reroutes still happened even with this security measure in place.

The routes stayed in circulation for around two hours, causing massive disruption and allowing a lot of traffic to be rerouted through routers that were never intended to receive that traffic. This incident is not so much an attack, as it happened by accident, but China Telecom has rerouted traffic of Western countries through its servers before, as it has caused redirections through its servers several times already. Most attempts of redirecting have focussed on redirecting traffic that was intended to go from one part of the United States to another [46]. These previous hijack attempts have caused harm to the reputation of China Telecom and the fact that the rerouting in 2019 happened along their AS, and it went on for a lot longer than necessary has caused extra media backlash.

1.3.3. Speaker impersonation

Speaker impersonation is the act of a router claiming to speak BGP while it is not authorized or configured to speak BGP, or the router being configured to speak BGP, but claiming to belong to a different AS than it does. The latter case can be seen as a form of path altering as well, as a different AS number is injected into a path than was intended, resulting in a false path. The former case could result in BGP traffic going along potentially wiretapped routers, while the latter can cause traffic to be redirected to a different AS than originally intended. In both cases, the consequence is that traffic can be snooped, which can lead to serious problems as this compromises the confidentiality of the network traffic.

Despite this being considered as a problem of BGP security, especially in early works on the topic of potential security problems in BGP as shown by the work of Smith et al. [133] in 1998, there have been no media reports on this security flaw significantly impacting routing across the entire Internet. Because it was a concern in early works regarding the security of BGP, several early proposals for securing the protocol have included measures against the possibility of an attack abusing this vulnerability. For example, S-BGP [92], the first security solution for BGP, does require BGP speaking routers to be authorized to speak BGP by way of binding BGP speaking routers to AS numbers through the usage of a public key infrastructure (PKI). However, there is next to no concern about the possibility of speaker impersonation occurring in later BGP security solutions. This is most likely due to a combination of these attacks rarely ever actually happening in the real world and the fact that most of the goals of these attacks can also be reached by altering a path, which is far easier. As such later solutions simply have no defence against this attack.

1.3.4. Protocol manipulation attacks

The category protocol manipulation attacks is a broad category of attacks, and are about attacks that involve altering parts of the BGP messages, mainly attributes. This is a relatively new type of attack, with many sources mentioning this kind of attack being published during and after the 2000s, such as Butler et al. mentioning it in a paper published in 2005 [62]. Zeb et al. also mention it in a publication from 2011 [151]. The first time it has been called a protocol manipulation attack was in 2018 [113]. Specifically mentioned attributes that can be manipulated to carry out an attack are the MED and the RFD/MRAI. The MED is the multi-exit discriminator and is used as a tie-breaker in deciding to which BGP speakers the message should be forwarded if all other factors such as local preference are the same. Altering this attribute could lead to paths being propagated to ASes that were not supposed to receive them, which leads to a route leak. RFD stands for Route Flap Damping, and is a mechanism built into BGP to prevent routes from being withdrawn and reannounced constantly, a practice that is known as "route flapping." How it does this is discussed in the Background chapter. MRAI stands for Minimum Route Advertisement Interval and is a timer specified to limit the number of updates on a per-destination basis. Changing any of these will impact the convergence time of BGP updates, and in turn, will disrupt routing.

Similar to speaker impersonation, this is a considered problem in the field of BGP security, and there are conceivable problems related to this attack. Still, there have not been any reports of attacks with significant impact that made use of this weakness. That does not mean that there have been no attacks making use of this weakness, however. Compared to speaker impersonation, there are even fewer security solutions that have any measures to prevent this from happening. That might have to do with this attack only being considered as a threat to BGP after a lot of research on secure BGP solutions was already completed. Incidentally, the only security solution that provides some defence against these attacks was developed before many of the articles considered protocol manipulation a serious threat to BGP security. It might also have to do with the fact that extending BGP convergence time to disrupt connectivity is more convoluted than simply hiring a botnet to execute a DDoS attack. Especially because commercial botnet lending was already a working service around the time that this attack was seriously considered as a threat to BGP security [8].

1.3.5. Weaknesses inherited from TCP/IP

As mentioned before, BGP messages are transported using TCP/IP. This protocol itself is not without its flaws, though, as Bellovin pointed out in 1989 [58]. It is vulnerable to other attacks, such as SYN flooding [41] and IP spoofing [79]. These vulnerabilities can be used to disrupt the correct functioning of BGP. However, these vulnerabilities have more to do with TCP/IP than with BGP, and as such, there are no BGP security solutions that counter these weaknesses.

1.3.6. DDoS attacks

Another weakness to consider that is mentioned many times by papers discussing threats to the correct functioning of BGP is the possibility of DDoS attacks being executed against BGP speakers. This can be done by flooding the BGP speaker with lots of data, and one method more specific to BGP is to flood the BGP speaker with BGP messages. However, this is also not a weakness that is exclusive to the correct functioning of BGP, and there are DDoS countermeasures already available on the market, such as CloudFlare [11]. These countermeasures tend to not specifically deal with DDoS attacks that are caused by sending too many BGP messages, however. There are no BGP security solutions that solve DDoS attacks specifically against BGP speakers.

1.4. Research question

Briefly mentioned in the previous section is that there have been security solutions proposed to tackle these vulnerabilities in BGP. BGP security solutions have been in active development since 2000, with each new one attempting to improve on previous work. However, none of these security solutions have been deployed over the entire Internet, or on large parts of the Internet, even though attacks on the protocol persist to the current day, as can be seen with the China Telecom incident of July 2019. This begs the following question: why not? Why hasn't interdomain routing been secured yet against the possible attacks if so many security solutions have been developed and proposed, with one proposal being as recent as 2018? As such, the research question of the thesis is as follows:

Why has no BGP security solution been deployed to protect the entire Internet yet, and what can be done to protect ASes against BGP attacks in the future?

There are several aspects to this research question. As with many other fields in cybersecurity, the threat landscape changes over time. Also, there must be something wrong with the proposals to secure BGP if none of them have been deployed on a wide scale. To address these aspects, the question can be subdivided into several subquestions, each of which can be answered individually. These subquestions each deal with an aspect of BGP security, and are as follows:

1. How have threats to BGP changed over time, and have proposed security solutions adopted to possible changes?

2. What are the security solutions that have been proposed to BGP, what kind of benefits do they provide, and what can we learn from comparing them?

3. What can be done in the future of BGP security?

The subsequent chapters of this thesis answer these subquestions. Chapter 2 discusses earlier academic work regarding the deployment of BGP security. Chapter 3 is intended to give the reader some more background knowledge about both BGP and data structures that are often used in BGP security solutions, such as Public Key Infrastructures or PKIs, for short. Chapter 4 discusses and explains the different security solutions that exist in great detail. Chapter 5 analyzes the evolution of the threat landscape, and how threats to BGP have changed over time. Chapter 6 analyzes the different security solutions that exist by building taxonomies to compare them on their features, cost, centralization, and the benefits gained from deploying each one in different deployment settings, to see why none of them have been deployed on a large scale. Chapter 7 discusses and explains several different BGP detection algorithms. Chapter 8 then analyzes these algorithms by also building taxonomies to compare them on what kinds of data they use, what techniques they use, etc., to see what would be ideal for a detection scheme. Chapter 9 concludes the research and discusses potential future work.

2
Related work

There is a body of research literature on the vulnerabilities in interdomain networking and BGP. That research also includes security proposals to address some or all of the security problems in BGP. However, this thesis is to going shift away from proposing a new security solution and more towards the analysis of the BGP security ecosystem, such as the current and former threats and the existing proposals, to see what we can learn from comparing them. There has been prior academic work analysing the BGP security ecosystem. These publications can be subdivided into several categories.

2.1. Literature surveys in favour of BGP security

One of the first studies published on the topic of routing protocol security is a study performed by Perlman [121]. This study concluded that, while networking protocols were generally robust against small failures, they could not do anything against failures involving a router that intentionally modifies routing messages, also known as Byzantine failures. Given the fact that most attacks on BGP happen by using modified messages, often intentionally modified messages, it goes without saying that network routing is not protected and as such needs measures to protect it against these Byzantine failures.

Butler et al. published a literature survey discussing the weaknesses in the routing protocol as well as several security solutions that were intended to solve them [63]. This research even includes a simple taxonomy of several early security solutions. However, back when it was written, there were more security solutions, and the taxonomy only serves to explain the security features that each solution can provide. It also concludes with the fact that cryptography and the introduction of centralizing measures would help BGP security greatly. Still, the costs associated with cryptography would be too high, and the introduction of centralizing measures would centralize at least part of the Internet, which is not exactly feasible given the fact that the Internet is decentralized by construction. In contrast to many other surveys however, this survey does remain positive on the question of whether BGP security can be achieved.

Another survey, published by Farley et al. [68], concludes that the inclusion of TCP MD5 signatures have improved BGP security, but it is still lacking in a lot of areas, as explained in the survey. Mitseva et al. have published a more up to date literature survey, including more solutions that have been proposed since [113]. This literature survey also included a taxonomy, which is more in-depth than the one published in the literature survey of Butler et al., but it still lacks on aspects such as whether or not a solution provides incremental benefit if the solution is incrementally deployable. The literature survey concludes that most of the BGP security solutions proposed either solve most of the security issues that exist with BGP at the cost of a lot of computational overhead or drop some security to achieve better performance.

Finally, Lychev et al. proposed a different way to secure Internet routing [109]. The authors simulated naive attacks on ASes given full deployment of any security solution, both with and without cooperation of prefix filtering. They found that prefix filtering, so whitelisting announcements made by ASes that have no customers, helps significantly in protecting ASes from hijacks. Especially when combined with a cryptographic security solution. Prefix filtering can also be set up without requiring multiple ASes to do so at the same time.

The authors also found out that, when fully deployed, robust security solutions would provide more protection against attacks than prefix filtering, but in a partial deployment scenario prefix filtering was about as useful as a security solution. The authors conclude the article by stating that prefix filtering should be used in addition to a secure BGP solution due to the benefits that filtering can provide and the ease of setting it up. Making comparisons in different deployment scenarios is useful for measuring the security benefit that they provide, and I have taken a similar approach in this thesis.

2.2. Challenges in deploying secure BGP

There is also academic work that takes a closer look at some specific security solutions and the challenges in deploying them. Khan et al. take a closer look at S-BGP, soBGP, and psBGP for example [95]. Similar to Butler et al., it also has a simple taxonomy to compare the three proposals, but the conclusion is nothing more than that the problems with deploying the solutions, especially those with S-BGP, are still unsolved. These problems mainly have to do with the increased computational overhead that a solution to BGP security would introduce. The authors also mentioned that the most obvious negligence today is that there should be a PKI for address attestation.

Zhao et al. also mention the additional computational complexity of using S-BGP in their analysis of several preventative and one detective BGP security solution [156]. In their analysis, they compare the solutions on the strength of the security that they provide in both origin authentication and path authentication, as well as the additional estimated temporal and spatial overhead that these security features introduce. The authors then use a simulation model to simulate the temporal and spatial overhead of standard S-BGP, and S-BGP using different signature schemes. Their overhead comparison and the comparison of the strength of the security that these solutions provide is decent, but it lacks justification. For example, the authors mention that, generally, centralized verification by using a PKI provides stronger security than decentralized verification by using peer verification, but not the reason why. One reason why would be because in case of a central authority being used, only the central authority has to be trusted, whereas in a decentralized setting more parties have to be trusted.

There are also several works discussing the difficulties and the potential extra security vulnerabilities in deploying the RPKI, due to the solution being so popular as it was and still is considered the basis of BGP security, albeit with several flaws that will be exposed later on in this thesis. Liu et al. discuss several risks in deploying the RPKI [107], such as issues with the data synchronization, risks of incomplete deployment, and economic/political risks in deploying such a hierarchical solution. Wählisch et al. take a closer look at the deployment statistics regarding RPKI and conclude that some big ASes, especially those containing big content distribution networks, are lagging on deployment of the RPKI [141]. Considering ASes that host big CDNs tend to host more important parts of the Internet and therefore tend to be higher on the target list for hijackers, the fact that RPKI deployment is lagging for these ASes specifically is troubling.

2.3. Modelling BGP adoption

Another category of academic literature argues that there should be more incentive for ASes to adopt security solutions, and even discuss strategies for deploying BGP security, based on simulations or on how the market works. Networking protocols in general need incentives for individual operators to adopt them, because of generally high costs for an individual to adopt them and relatively little benefit for said individual, especially short-term [115].

Wählisch et al. also argue that there should be more incentive, especially for ASes that contain often-visited parts of the Internet, to deploy the RPKI [141]. Gill et al. propose a strategy that governments and other institutions could use to drive the deployment of secure BGP solutions [74], while Lee et al. propose an availability device to bridge adopters from their current routing solution to a new one [101]. They create a model where secure BGP solutions are deployed by given ASes, not because they are concerned with regards to the safety of their traffic, but the economic benefit of deploying them outweighs the cost. Their main insight is that, if the costs of deployment are low enough, the majority of ISPs will have an incentive to deploy BGP, because it will differentiate them from the rest of the ISPs on the market. However, one should also take into account that in this model, ISPs have incentives to disable secure BGP solutions, as it is possible that when an ISP becomes secure, some of its original incoming traffic might change the path that they take and doesn't enter through customer links anymore. The importance of market forces as a driver for the adoption of secure networking protocols has also been recognized by Alderson et al., who argue that different stakeholders have different roles and responsibilities in securing information architecture [54].

Finally, Chan et al. published a paper that proposed a new metric for protocol design, called adoptability [64]. The model abstracts a lot about the costs of adopting a certain solution. It also abstracts the overhead that such a solution introduces, to be able to get this metric. In addition, it also introduces the concept of security benefit, which is a value in the interval $[0,1]$, which represents the estimated increase in the probability that traffic passing through an AS cannot be diverted anymore after the given AS adopts a security protocol. The main result is that there is a critical threshold of cost, where if the cost of transitioning from BGP to a more secure version of BGP is lower than the utility benefit gained from doing so, almost all ASes would adopt the more secure version of BGP, whereas if the opposite was true, almost no ASes would adopt the more secure version. This has been simulated under different assumptions for network traffic and different assumptions for the skillset of the attacker. In general, though, this research attempts to abstract too many cost aspects of deploying a security solution to be useful.

2.4. Literature surveys with a less than favourable view of BGP security

One final category of academic work has a more pessimistic view on the topic of BGP security and suggests moving away from the topic entirely, seeing it as a lost cause. One such example is a paper written by Wendlandt et al. [145], which suggests dropping the idea of securing routing protocols altogether in favour of focusing more on secure data delivery. This paper suggests that confidentiality and integrity of traffic, two of the three factors of security, can be achieved using end-to-end measures, such as end-to-end encryption. The remaining factor is availability, which can be improved by simply allowing a router to store multiple routes towards a given destination, thereby reducing the odds that one cannot reach a certain part of the Internet in case of an attack occurring.

Suchara et al. performed research using an idealized version of a secure BGP protocol. They have found that, even if the best security protocol were deployed by only the ASes that make up the core of the Internet, there would be a significant impact on performance [138]. The actual impact on performance is estimated to be even worse than what is shown in the paper, as the paper uses an idealized version of a secure BGP solution.

In a similar vein, there have been calls for the creation of a Cyber Security Council. This council would be a subdivision of the UN [14]. This consideration is in part because nation-state attackers are executing these attacks (as seen with the Pakistan Telecom hijack). As such, attackers with the skillset and resources on the level of nation-state actors should be considered [30]. Also, because attacks such as prefix hijacking can affect the world on an international level, punishments such as sanctions against attacks from which a lot of attacks on BGP occur are not out of scope.

3 Background

This chapter goes into more detail on the message contents of BGP messages or data structures that are often used in BGP security solutions. It is meant to give a much deeper explanation of relevant BGP messages and data structures, and what kind of manipulation of messages is required for certain types of attacks to be executed.

3.1. The structure of a BGP message

Similar to IP, BGP messages have a fixed-size header, which serves to provide information about the data that may or may not follow the header. A schematic overview of the header is shown in figure 3.1. As can be seen, there is a 128-bit marker, which is a value that can be used for the detection of the loss of synchronization between BGP speakers as well as authentication of incoming messages. The 16-bit long length field specifies the total length of the message, including the header, in octets. Finally, the 8-bit long type indicates the type of the message: 1 for OPEN, 2 for UPDATE, 3 for NOTIFICATION, and 4 for KEEPALIVE. The purpose of these different messages is as follows:

- OPEN: this is the first message sent by each of the sides of a connection once the connection is opened. These are used to set up a BGP connection between two BGP speaking routers.

- UPDATE: this message type is used to transfer routing information between BGP speaking routers, and the information included in the message can be used to create a graph of relationships between different ASes.

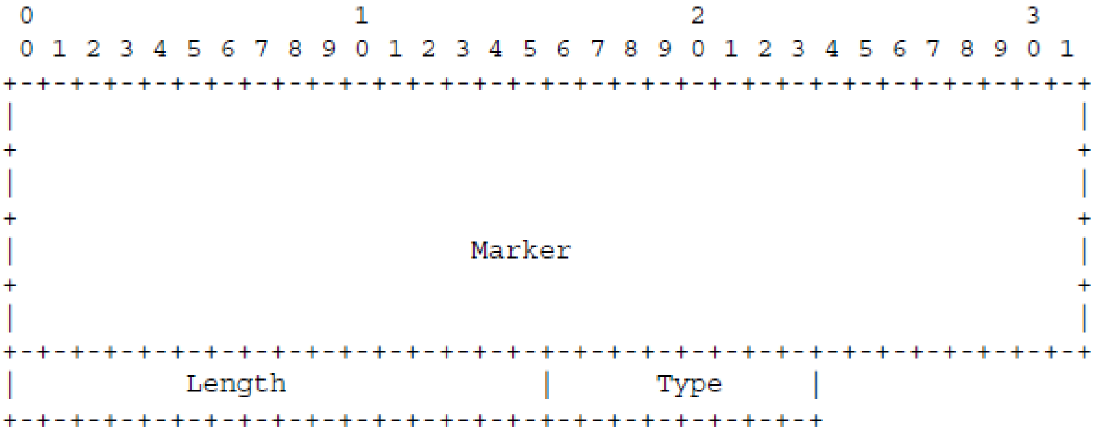
- NOTIFICATION: these messages are sent when an error is detected. A BGP connection is closed once such a message is sent.

- KEEPALIVE: messages of this type are exchanged between peers to verify that certain peers are still reachable. These should not be sent too infrequently, but also not too often. The RFC suggests that a reasonable maximum time between messages of this type is about one-third of the Hold Time interval, which is specified in the OPEN message.

As can be seen from the list, the message that is used for the exchange of router information is the UPDATE message, and as such, will be the focal point for this thesis. A more detailed explanation of the other message types is out of scope for this thesis, and interested readers should refer to RFC 1771 for a more detailed explanation [125].

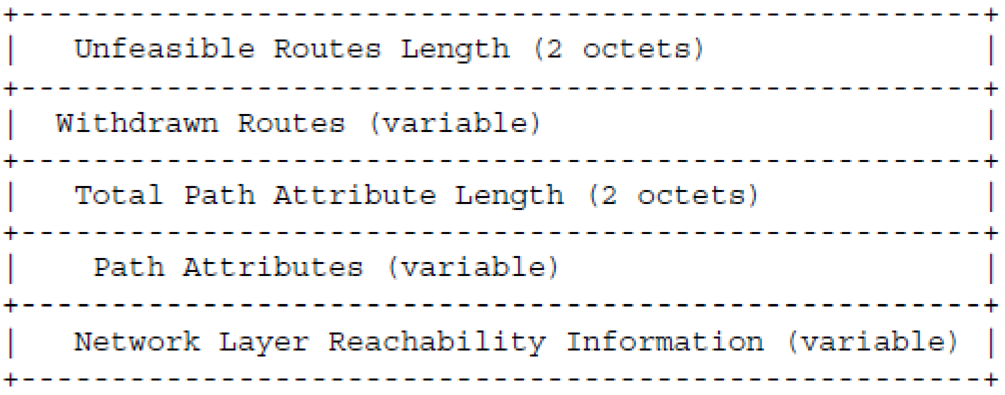
3.2. The structure of an UPDATE message

As mentioned before, the exchange of network reachability information is done by exchanging BGP messages that are of the UPDATE type. This type of message is used for advertising a single feasible route or for withdrawing multiple unfeasible routes to a certain prefix. The UPDATE message header comes after the main BGP message header, and in contrast to the BGP message header, contains mostly optional fields. A schematic overview of the message can be seen in figure 3.2.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                                                               +
|                           Marker                              |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Length               |      Type     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3.1: A schematic overview of the header of a BGP message. Source: Rekhter, 1995.



```
+-----------------------------------------------------+
|   Unfeasible Routes Length (2 octets)               |
+-----------------------------------------------------+
|  Withdrawn Routes (variable)                        |
+-----------------------------------------------------+
|   Total Path Attribute Length (2 octets)            |
+-----------------------------------------------------+
|    Path Attributes (variable)                       |
+-----------------------------------------------------+
|   Network Layer Reachability Information (variable) |
+-----------------------------------------------------+
```

Figure 3.2: A schematic overview of the different fields in the header of an UPDATE message. As mentioned before, all these fields are optional. Source: Rekhter, 1995.

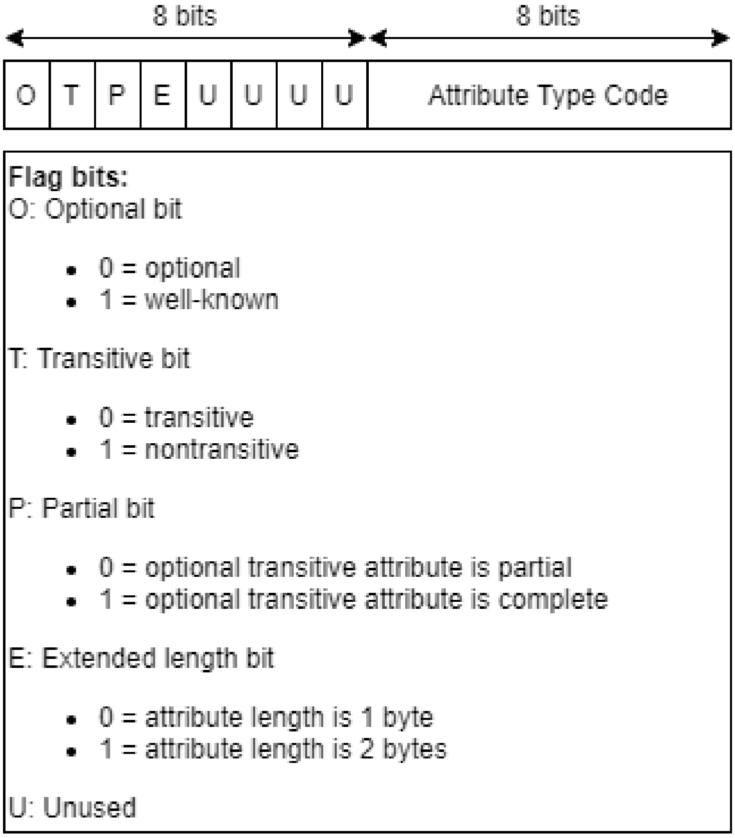
3.2.1. The routes part of the UPDATE message

The Unfeasible routes length field indicates the total size of the Withdrawn routes field in octets. The Withdrawn routes field contains a list of all the prefixes that are being withdrawn. This list consists of tuples of the form (Length, Prefix), where Length is a one-octet long field which indicates the length in bits of the IP address prefix, and the Prefix field is a field of variable length containing the IP address. These combined form a prefix: if the encoded length is decoded as 24 and the encoded IP address is decoded as 187.96.13.0, the route that needs to be withdrawn is the route that leads to prefix 187.96.13.0/24. Altering the Prefix field to broadcast a prefix that the AS does not own, leads to (sub)prefix hijacking.

3.2.2. The path attribute part of the UPDATE message

The Total Path Attribute Length indicates the length of the Path Attributes variable in octets. The Path Attributes value contains a list of all the path attributes. Each of these attributes is encoded as a triple, which is structured as follows: (attribute type, attribute length, attribute value) and have a variable length, which is indicated by the attribute length field. The attribute type field is two octets long, of which the first octet serves as the Attribute Flags octet, and the second one serves as the Attribute Type Code octet [77]. A schematic overview can be seen in figure 3.3. As can be seen, the first four flag bits in the Attribute Flags octet each mean something different, and each of them serves a different purpose. These purposes, however, are beyond the scope of this thesis, and interested readers should refer to RFC 1771 for a more detailed explanation [125].

Aside from the flag bits in Attribute Flags, there is also the Attribute Type Code. This code, along with the



Flag bits:
O: Optional bit

- 0 = optional
- 1 = well-known

T: Transitive bit

- 0 = transitive
- 1 = nontransitive

P: Partial bit

- 0 = optional transitive attribute is partial
- 1 = optional transitive attribute is complete

E: Extended length bit

- 0 = attribute length is 1 byte
- 1 = attribute length is 2 bytes

U: Unused

Figure 3.3: a schematic overview of the BGP Attribute Type. Only the first four bits of the Attribute Type Flag field serve a purpose. The other four bits are not used, and their value should be zero. Source: Goralski, 2017

flag bits of Attribute Flags, indicates how the data in attribute value should be interpreted. The codes that are relevant for BGP security are:

- AS-PATH (Type Code 2): indicates that the data is a sequence of AS path segments. Every segment is a triple, consisting of the values (path segment type, path segment length, path segment value). These variables are used as follows:

  – Path segment type: 1-byte long field that can have either the value 1 (indicating that the list of ASes in the route is an unordered set) or 2 (indicating that the list is an ordered sequence)

  – Path segment length: 1-byte long field defining the amount of ASes n in the path segment value field

  – Path segment value: list of n 2-byte long AS numbers

  When a BGP speaking router propagates a route that it has learned from incoming UPDATE messages, it modifies the AS_PATH attribute if the message is sent to a neighbouring AS. The different kinds of modification are as follows:

  – If the first path segment of the AS_PATH is a sequence, then the AS number of the AS that the speaker belongs to shall be prepended. This is essentially how prepending one's own AS number to the route is done.

  – If it is a set instead, then the BGP speaker will prepend a new segment to the path that is meant to be a sequence.

  This attribute is also used to originate paths. In that case, the BGP speaker will send the AS number of the AS it belongs to BGP speaking routers in a different AS. It will send nothing to routers within the same AS. Altering this variable can lead to either prefix hijacking (if the origin of the path is changed) or path altering (if part of the path is altered to redirect traffic from its original route to a new route).

- NEXT_HOP (Type Code 3): defines the IP address of the router that should be used as the next destination of the message. This can be both an internal (within the AS) or an external border router. Any router can be specified as long as it shares a subnet with the originating router. This variable can be altered to send BGP messages to a different router than intended, causing route leaks and traffic redirections.
- MULTI_EXIT_DISC (Type Code 4): an attribute that can be used by a BGP speaker to choose which of the multiple exit points to use to a neighbour AS. Given that all other factors, such as local preference, are equal, the exit or entry point with the lowest value for MULTI_EXIT_DISC is preferred. This variable can also be altered to send BGP messages to the wrong router, causing route leaks and traffic redirections. Tampering with this variable is a form of a protocol manipulation attack, as has been discussed in section 1.3.4.
- LOCAL_PREF (Type Code 5): a four-byte long attribute that encodes a non-negative integer to inform other BGP speakers within the AS about this BGP speakers' preference for an advertised route. BGP speakers are required to calculate the preference of each external route and include this degree of preference when advertising a route to peers. Tampering with this attribute in the message does not cause any changes to the route forwarding behaviour as this attribute is merely intended to inform other ASes about the local preference of a BGP speaker for a certain route. But tampering with this attribute in the routing database can cause a BGP speaker to adopt and forward routes it would not normally forward.

The remaining type codes are ORIGIN (type code 1), ATOMIC_AGGREGATE (Type Code 6), and AGGREGATOR (type code 7). These are not relevant to the thesis as manipulation of these attribute types can not be used in a protocol manipulation attack. Interested readers should refer to RFC 1771 for further explanation [125].

3.2.3. The NLRI part of the UPDATE message

Finally, the Network Layer Reachability Information (NLRI) field is a field of variable length that contains a list of IP address prefixes. There is no specified length of this field, but it can be inferred from the total length of the message, the length of the header, and the two lengths encoded in the message. The field contains a list of one or more tuples of the form (length, prefix), similar to the list of withdrawn routes. The path attributes that are specified in the UPDATE message can be applied to each of the routes in the list specified in the NLRI. Specifying a list of tuples this way cuts down on the number of messages that have to be sent [77]. It does, however, lead to a security problem, as the NLRI can be expanded with more IP address prefixes, which for a receiving AS means that the path attributes that are specified apply to more prefixes than they were supposed to apply to. This could cause route leaks because routes could be forwarded to the wrong ASes.

3.3. Setting up a BGP connection

The way BGP functions is that a BGP speaker can be seen as a finite state machine, with these message types being used for transitions from one state to another [5]. The state machine is rather similar to the TCP finite state machine [42], which is logical because BGP runs over TCP/IP.

Initially, the BGP speaker is in an idle state, until it connects to another BGP speaker, in which case it initializes all the resources for the peer connection. It can connect to other BGP speakers, and by doing so, it transitions to the "connect" state. The BGP speaker will remain in this state until the three-way handshake of TCP is complete. After completion of the handshake, the BGP speaking router is either in the "active" state or the "opensent" state. Which of these states the BGP speaker is in depends on whether or not the OPEN message has been successfully transmitted from one speaker to another. If successful, then the speaker will be in the "opensent" state. If not, then it will be in the "active" state. In this state, the BGP speaker will send another OPEN message, to negotiate a BGP session and to transition to the "opensent" state. If this second attempt also fails, then the state goes back to "connect". If the speaker is in the "opensent" state, then the OPEN message was successfully sent, and once the message is confirmed to be received, the speaker goes into the "openconfirm" state. In this state, a KEEPALIVE message is transmitted and both speakers transition to the established state, which is the final state and allows two BGP speakers to exchange routing data with one another.

A state machine diagram is shown in figure 3.4. This state machine diagram also contains transitions for when the TCP connection ends between two speakers. In that case, as can be seen, they go back to the "idle" state.

The way BGP connections are set up allows for a malicious actor to disrupt normal BGP connections. A malicious actor can keep sending OPEN messages to prevent transition to the "established" state, similar to TCP SYN flooding [41]. An attacker could also choose to open and close TCP connections continuously. This problem, however, is more specific to TCP.

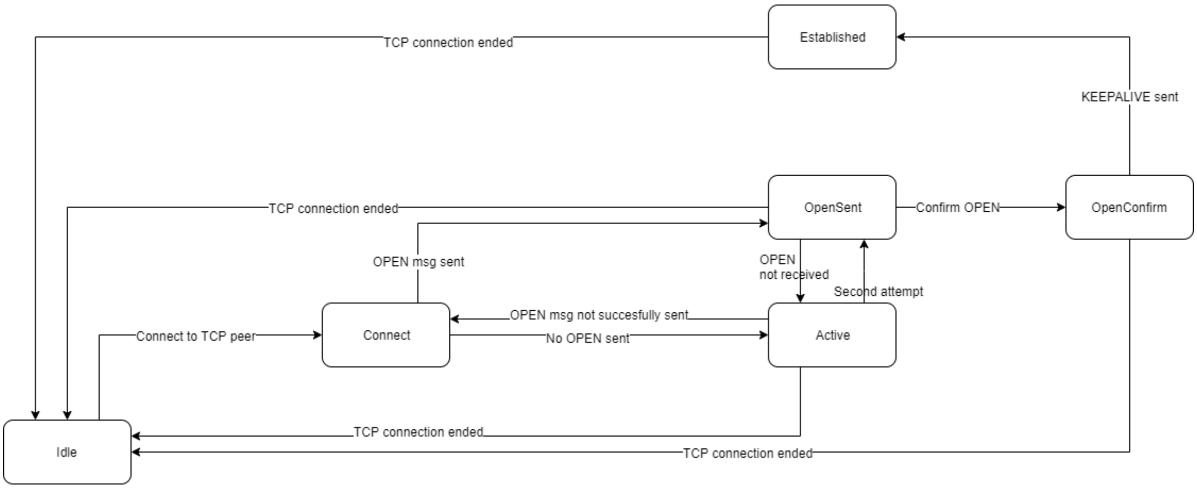


Figure 3.4: A finite state machine diagram of how a BGP connection is set up. The Established state is the final state for allowing the exchange of BGP routing information.

3.4. The routing table

By exchanging one or several UPDATE messages between each other once a BGP peering connection has been set up, two BGP-speakers (they need not necessarily be routers) belonging to different ASes that are connected by an external link effectively exchange their entire routing tables with one another. These routing tables are called Route Information Bases (RIB's) and consist of three separate parts:

- The Adj-RIBs-In
- The Loc-RIB
- The Adj-RIBs-Out

The Adj-RIBs-In store routing information that has been gathered from incoming UPDATE messages, and represent routes that can be selected in the decision process. These routes can be filtered using an incoming routes filter, to preselect which routes are going to be forwarded to the Loc-RIB for usage in the selection procedure. The filter is often manually configured and is useful for filtering out messages that come from ASes that are faulty. The Loc-RIB contains the routing information that the BGP has selected to use by applying its local policies to the routing information that was in the Adj-RIBs-In. Routes in the Loc-RIB are also written to the local routing table. These have been selected as routes that can be used to forward traffic along. The Adj-RIBs-Out contains routing information that the BGP speaker has selected as fit for advertisement to its peers. This selection also happens using a set of policies that belong to the BGP speaker. The selected routing information is the information that will be carried by the UPDATE messages coming from this BGP speaker. Figure 3.5 gives a schematic overview. It should be noted that Adj-RIBs-Out, similar to Adj-RIBs-In, also has a filter that most of the time is manually configured. Often for filtering which of the routes are forwarded to which of the peers. This is useful if the router belongs to an ISP and only forwards routes (and as such provides service) to customers.

3.4.1. Routing table decision process

The decision process in the figure has three phases [140]. In the first phase, each route that has been received from a speaker is analysed and assigned a certain preference. The routes are sorted according to preference.

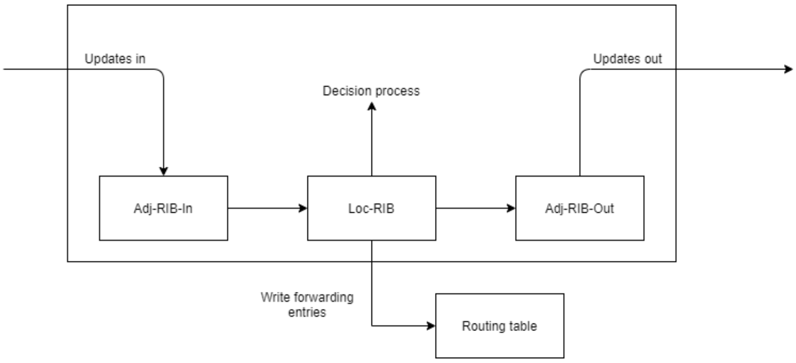


Figure 3.5: A schematic overview of the process of how incoming routes are selected as fit for routing and broadcasting. Source: Vieira, 2010

In the second phase, the best route to each destination is selected. "The best route" is defined as the route with the highest preference. The routes are then used to update the Loc-RIB. In the third phase, routes in the Loc-RIB are selected and sent to other BGP speakers. The process is illustrated in figure 3.6.

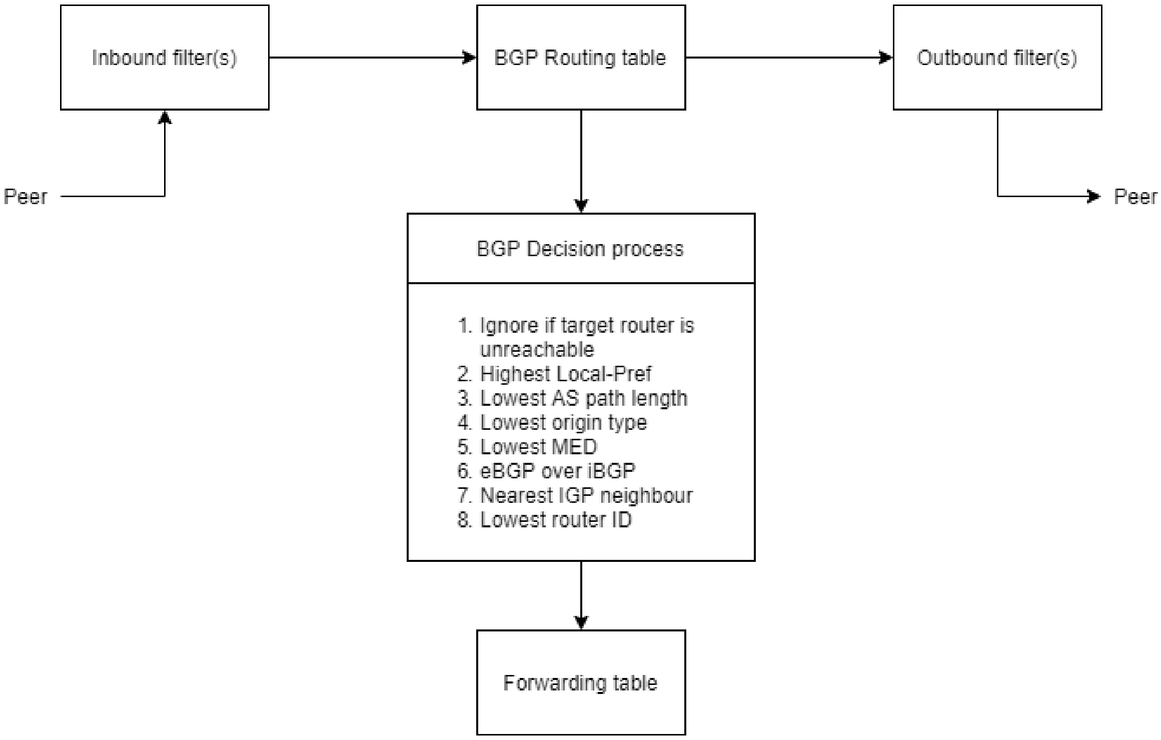


Figure 3.6: The process of selecting routes in BGP. The decision process pictured below the BGP routing table is very general and will be explained in more detail. The figure also shows that there are route filters on both incoming and outgoing routes. Source: Vieira, 2010.

The decision process shown in the picture is a simplified version of the decision process most routers use and contains the more frequently used decision rules. In reality, the routers of different manufacturers often use different decision processes. For example, this is the decision process for Junos OS routers [29]:

1. Route preference in the table is compared, with the one with a lower preference value being selected. This means the following: the routes are compared by type, and their default preferences are used. The specification explicitly mentions choosing OSPF routes over BGP routes as the route preference for OSPF routes is lower.

2. Local preference is considered, with the route with the highest local preference winning.

3. The route with a shorter AS path is preferred. In general, this means the path with the shortest amount of hops, but there can be other distance metrics as well, such as estimated geographic distance between the start AS and the end AS.

4. The route with the lowest origin code wins. This means that routes from within the AS win this as they have an origin code of 0 (representing IGP). After that, routes from outside of the AS are selected, and after that, incomplete routes are selected.

5. The route with the lowest MULTI_EXIT_DISC (MED) value is preferred. If this value is absent for a broadcasted route, then the MED value for that route is assumed to be 0.

6. Routes learned from eBGP have a preference over routes learned from iBGP.

7. For routes that are learned over iBGP, those with lower IGP costs are preferred. Because this deals with routes within the AS as it concerns iBGP, the physical next hop to within the AS is determined as follows:

   (a) BGP examines both the inet.0 and inet.3 routing tables, and the physical hop with the lowest preference is used. If there is a tie, then the physical next hop of the latter table is used. These routing tables seem to be inherent to Junos OS routers.

   (b) If there is a preference tie in the same routing table, then the physical next hop of the route with more paths available to them is used. This selection rule makes sense as this route goes to a router with paths to more destinations.

8. The route reflection cluster list attribute is considered, and the route with the shortest cluster list is preferred over other routes. Routes without this list are considered to have a cluster list of length 0.

9. The router ID is considered, and the route from the peer with the lowest router ID is preferred.

10. The peer address value is considered, and the peer with the lowest IP address is preferred. This is the final tie-breaking rule, and there are no ties possible at this level, as every peer has a different IP address.

Do note that these rules specifically apply to Junos OS routing tables and do not apply to all routers. These rules contain several tie-breaking rules that are based on properties that only Junos OS peers have, such as inet.0 and inet.3, which is specifically only for Junos OS routing. Cisco has a slightly different ruleset, with the first rule being, for example, to prefer the path with the highest WEIGHT parameter, which is a parameter specific to Cisco routers [10]. Generally speaking, the rules that are most often used as tie-breakers in routing are as follows:

1. Prefer the path with the highest LOCAL_PREF

2. Prefer the path with the shortest AS_PATH

3. Prefer the path with the lowest origin type

4. Prefer the path with the lowest MED

5. Prefer eBGP over iBGP

6. Prefer the path originating from the peer with the lowest ID

7. Prefer the path from the lowest neighbour IP address

Because most of these rules depend on variables in BGP UPDATE messages that can be altered by a malicious AS, one can deduce these rules lead to security issues in the form of (for example) route leaks.

3.5. The PKI

One data structure that comes up often in BGP security solutions is the Public Key Infrastructure, or PKI for short. One of the security solutions proposed, the Resource Public Key Infrastructure (RPKI), is nothing more than a modified PKI, as the name already implies. The PKI is a security structure that consists of software and hardware elements that a trusted third party, be it an organization or an individual, can use to establish verifiable ownership between a public key and its user by way of issuing certificates that couple a user to its public key [33].

Establishing verifiable ownership between a user and a public key is useful in many cases. Take, for example, a simple case of Bob sending an encrypted email to Alice using public-key encryption. In this case, Bob and Alice have their public-private key pairs, of which only their public key is known. Bob then wants to use Alice's public key to encrypt the message before he sends it to her. Alice sends the key, Bob uses it and sends the encrypted message to Alice, after which Alice can use the private key to decrypt the message. The problem for Bob is that he has no way of verifying that the public key that Alice sent him is Alice's public key. Because of this, someone else can try to impersonate Alice to get Bob to send the email to them. Using a PKI allows anyone to verify that a public key belongs to somebody that claims to have that public key as their public key.

As mentioned before, PKI's consist of software and hardware elements. These elements are typically as follows:

- Certification Authority (CA): this is a trusted third party that acts as a root of trust in the PKI hierarchy. The CA provides services that authenticate the identity of individuals and other entities in the network.
- Registration authority (RA): this is an entity that is certified by a root CA to issue certificates on behalf of the root CA to certain users. As such, an RA can be seen as a subordinate entity of the CA.
- Certification Database: a database that saves requests for certificates, as well as issued and revoked certificates.
- Certificate Store: a database that saves the contents of issued certificates, as well as that of pending or rejected certificate requests, on a local computer.
- Key Archival Server: the server that saves encrypted private keys in the certificate database for recovery in the case of a potential loss.

Figure 3.7 shows how certificates are issued and used to verify user identities [34]. The user has a public key and a certificate binding that public key to them. Another person, called the verifier, can then verify that the public key belongs to the user by asking the CA for verification of the certificate.

A real-life example of public keys whose ownership is verifiable by a PKI is a government of a country handing out ID cards to its citizens. In this case, the government is the root CA, local city halls that hand out ID cards on behalf of the government can be seen as RA's, your identity can be seen as the public key, and the issued ID cards can be seen as certificates that bind your public key to you. If you are asked to then prove your identity to, for example, an authority figure, who will act as a verifier, in this case, you can simply show them your ID. Then the verifier can verify whether it is true or false who you claim to be, as the certificate (the ID card) acts as valid proof that you are who you claim to be, given that you are speaking the truth.

3.5.1. PKI certificates

From the description of how a PKI fulfills its intended job, it has been made clear that it issues certificates. These certificates have the X.509 format, and since May 2008, use version 3 of this format, according to RFC 5280 [66]. Each of these certificates is a sequence of three required fields, these being the TBS (an acronym for "To Be Signed") Certificate, the signature algorithm, and the signature value. The first two of these fields consist of several other elements as well, while the last one is a simple bit string.

The TBS certificate is the main certificate and contains several fields. The most important fields are as follows:

- The version number of the certificate.
- An identifier for the algorithm used by the certificate authority to sign the certificate.

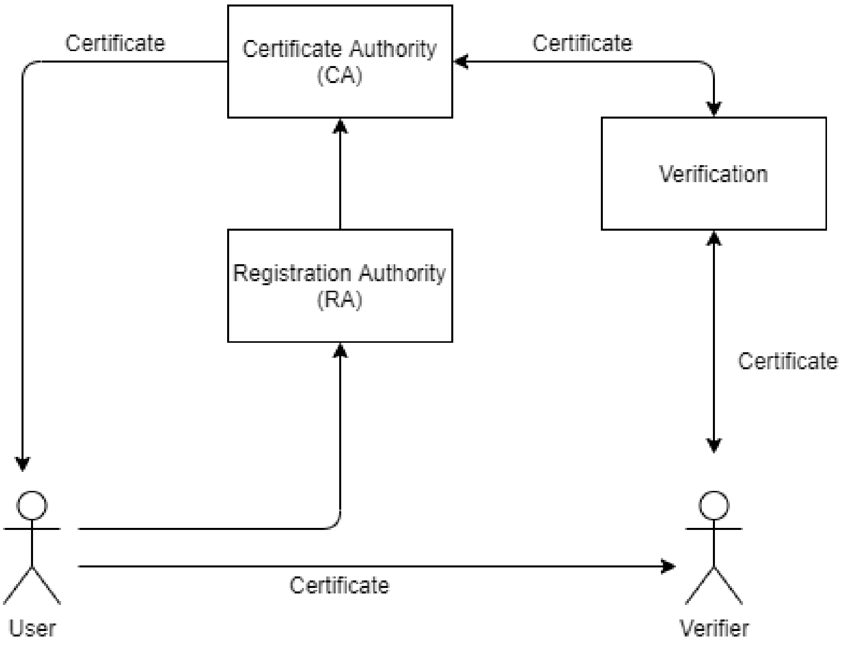


Figure 3.7: a simplified graphical overview of how issuing and verifying certificates works. The user and the verifier are both people making use of the PKI. Source: Network Encyclopedia.

- The identity of the entity that has signed and issued the certificate.
- The time interval during which the CA states that it will maintain information about the certificate.
- The identity of the entity associated with the public key.

The identity of the issuing entity and the entity associated with the public key are defined as distinguished names of the X.501 type Name, as defined in RFC 2459 [122].

The signature algorithm field is similarly defined as the signature field in the TBS certificate field. It contains an algorithm identifier as well as any additional parameters that the algorithm requires.

3.5.2. Certificate Revocation List (CRL)

Every CA maintains a list of issued certificates that have been revoked by that particular CA. This list is called the Certificate Revocation List, or CRL for short [9]. It should be noted that this list only contains certificates that have been revoked, not those that have expired. It is far easier to verify that a certificate has expired than to verify that a certificate has been revoked without using any kind of outside help, as one needs only to check the expiration date of a certificate. The CRL is instrumental in making a reusable PKI, as having no record of what certificates have been revoked would quickly lead to the PKI becoming bloated with a lot of expired certificates.

CRL issuers issue CRL's. These issuers can be the same institution as the CA, but if not, then, similar to RA's, it has to be authorized by the central CA to issue these lists. CRL issuers can generate complete CRL's, which list all unexpired certificates that have been issued and have been revoked for any reason, or a delta CRL, which only lists the certificates that have been revoked since a referenced complete CRL has been generated.

CRL's, like certificates, consist of several fields. The three main fields of a CRL are somewhat similar to the main fields of a certificate, those being the TBS certificate list, the signature algorithm, and the signature value. The signature algorithm is an algorithm identifier, and the signature value is once again a simple bit string. The main fields in the TBS certificate list are as follows:

- The version of the encoded CRL.
- The algorithm identifier for the algorithm used to sign the CRL.

- The entity that signed and issued this CRL. Must also be a distinguished name, similar to the issuer field in the certificate.
- The issue date of this CRL.
- The date by which the next CRL will be issued. The next CRL has to be issued before or on the date specified in this field, but no later. CRL issuers have to use this field in either none or all of their CRL's.
- A list of revoked certificates, along with the date of their revocation for each of these certificates.

3.5.3. Problems with PKI's

As can be seen, PKI's and their certificates can and do provide a reliable way to bind public keys belonging to entities to those same entities. Because of this, they are very often used in securing websites using SSL certificates, which makes use of several PKI protocols. As such, one might be inclined to believe that they are flawless. But PKI's are not flawless at all. One of their biggest flaws is that they require the central CA to be perfectly secure, and most importantly, always available. As such, the usage of PKI introduces a single point of failure in a system. If the CA is compromised in some way, then all of the certificates issued by that CA are compromised as well.

To give another real-world example, let's take the example of the government of a country issuing ID cards to its citizens once again. As has been said before, the government is the CA in this case, and the ID cards are certificates that bind the identity of an entity, its public key, to said entity. If, for example, the government of that particular country is overthrown or the country ceases to exist because of annexation into another country, the ID cards that the government issued tend to become invalid because the government no longer exists, which means that everyone that only has that ID card no longer has a valid certificate that proves them to be who they claim to be.

As an example of such an attack against an actual CA happening, there is the case of DigiNotar [16]. DigiNotar is a Dutch CA that issued two types of certificates, one of which under their name, and the other under the name of the Dutch government's PKIoverheid program, which is a PKI owned by the Dutch government [35]. At sometime before August 2011, an unknown attacker was able to get in the CA infrastructure at DigiNotar, and promptly issued valid SSL certificates for themselves for several often-used domains, one of them being for *.google.com, which includes the domain for Gmail, which of course is mail.google.com. The first indication that there was any kind of attack was around August 2011, when Iranian users started to notice some odd re-routing of their Gmail traffic. People speculated that the government of Iran was trying to spy on its citizens by monitoring their emails by executing a man-in-the-middle attack. As it turns out, the attack was using a wildcard certificate, issued by DigiNotar, for the domain of Google, which was used by the attacker to impersonate Google and its services to any browser that trusted the DigiNotar certificate. Browsers quickly patched this vulnerability by removing DigiNotar from their trusted CA list. Further details about the attacker are that they were most likely an Iranian citizen and acted on behalf of the government of Iran, which indicates that nation-state actors are not out of scope for these kinds of attacks. An earlier example of an attack on a CA is the attack on Comodo. This is another CA which had its security breached in March 2011. Not only did an anonymous hacker come forward and claim responsibility for executing both attacks, but according to Fox-IT, the company that investigated the DigiNotar hack, there are also signs that the same person has performed both hacks.

3.6. Hash tree

A (Merkle) hash tree is a hierarchical data structure that reduces the problem of authenticating a sequence of values to authenticating only a single value [112]. Figure 3.8 shows how such a hash tree is built. The leaves of the tree contain the individual values v_0 to v_7. These values are then hashed once, which results in their parent values $H(v_0) = v'_0$ to $H(v_7) = v'_7$. After this, the value of each parent node v_p in the hash tree is derived from the values of the left child node v_l and the one from the right child node v_r as follows: $v_p = H[v_l || v_r]$, where || stands for concatenation. Figure 3.8 displays such a hash tree. In this tree, the value of m_{03} is $H[m_{01} || m_{23}]$, for example.

The root value of such a hash tree enables one to authenticate all of the leaf nodes. If the value v_i must be verified, then a sender can send i, v_i, and a collection of values from sibling nodes on the intermediate nodes on the path from v_i to the root of the tree. For example: say that the value v_2 needs to be authenticated. The path from v_2 to the root is $v_2 \rightarrow v'_2 \rightarrow m_{23} \rightarrow m_{03} \rightarrow m_{07}$, and the siblings of each of the intermediate nodes are (in order) v'_3, m_{01}, and m_{47}. Using these values, one needs to calculate the following:

$$x = H[H[m_{01}||H[H[v_2]||v'_3]]||m_{47}]$$

Then, if x is equal to m_{07}, the value v_2 is correct. Also, hashing all the individual values v_i to v'_i first prevents a sender from having to disclose any other original value (in the case of v_2 requiring verification, that would be v_3) for authentication of a single value. As such, it is important that the function H is a one-way function, such as a hash function, because this makes it hard to find the original input.

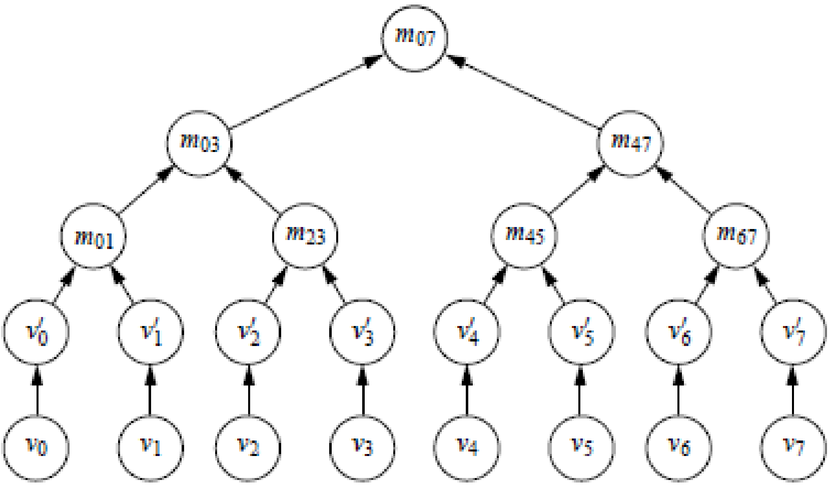


Figure 3.8: a schematic overview of a Merkle hash tree. This schematic assumes a balanced tree. Source: Hu et al., 2004.

4
BGP security solutions

Because of its security flaws allowing for many possible attacks, there have been several proposed solutions to improve the security of BGP. These security solutions range from rather basic solutions that only focus on defending against one attack (for example, prefix hijacking) to solutions that are designed to solve most of the security flaws in BGP at the same time. This chapter describes some of these proposed solutions in detail.

4.1. S-BGP

The Secure Border Gateway Protocol (S-BGP) solution is the earliest proposed solution to tackling the security flaws specific to BGP [92]. There have been earlier proposals to secure BGP in part, such as adding sequence numbers to BGP messages [134], the authentication of BGP messages [96], and adding information to UPDATE messages as they propagate through the internet [135], but this is the first proposed security scheme that provides a comprehensive solution to the security flaws in BGP.

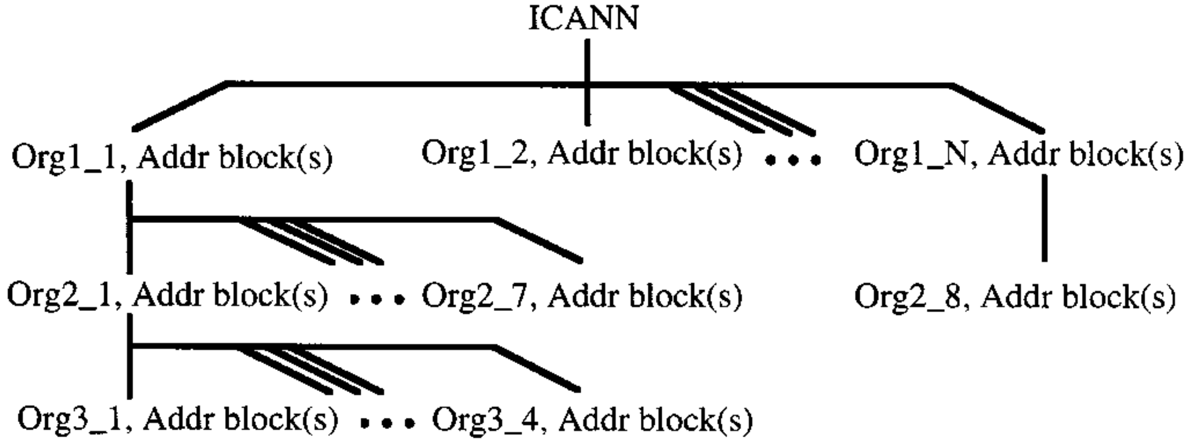
First published in 2000, it has since served as a useful basis for future research and subsequent proposals for enhancing the security of BGP. The proposal identifies several weaknesses of BGP, notably the fact that BGP runs over TCP/IP, which is a protocol that can be attacked, fictitious BGP messages can be inserted, violation of local routing policies, and accidental misconfiguration of routers leading to problems. It also outlines several requirements that S-BGP should meet, such as being able to handle the projected growth and usage of the Internet, it should be dynamic, and the architecture must be deployable, which means that the countermeasures against the weaknesses in BGP can be incrementally deployed and should work off of existing infrastructure.

As such, the proposed solution to securing BGP requires two public key infrastructures (PKI), a new path attribute containing "attestations," and the use of IPsec. The PKIs enable BGP speaking routers to validate identities and authorization of BGP speakers and autonomous system/prefix owners. These PKIs are based on the existing IP address delegation system, with the ICANN at the top and autonomous systems with only a prefix and no upstream providers at the bottom of the hierarchy. One PKI is used for address allocation, useful for verifying that a certain autonomous system owns the prefix that they claim to own, whereas the other is used for assignment of autonomous systems to BGP speaking routers, useful for verifying that a BGP speaker belongs to the autonomous system that it claims to belong to.

For the PKI that is used for address allocation, the certificates are issued through the same organizations that are responsible for address allocation. As mentioned before, the chain is rooted at ICANN, then continues down to the Regional Internet Registries (RIRs), then Local Internet Registries (LIRs) or Internet Service Providers (ISPs), then Data Storage Providers (DSPs), and so on. An example of how this PKI works can be seen in figure 4.1. The certificates issued are, in contrast to a regular PKI, used for verifying that an autonomous system owns the prefix that they claim to hold, instead of verifying the identity of an autonomous system. Certificates for organizations, which can represent RIRs, LIRs, ISPs, et cetera, are issued by the organization one level up the hierarchy. The notable exception to this is, of course, the ICANN itself, which issues its own certificate as it is the root of the PKI. For the PKI that is used for the assignment of certain BGP speakers to autonomous systems, three different certificates will be used. These three certificate types bind the following things together:

- AS numbers and the public key of an organization
- AS numbers and their public keys
- A router name, a router ID, an AS number, and the router's public key

When combined, one can see that the AS number is bound to the public key of an organization, the public key of the autonomous system, the name of the router, the ID of the router, and the public key of the router.



Subject	**Description**
Org1_x	a 1st tier organization (usually a registry)
Org2_x	a 2nd tier organization (usually an ISP or DSP)
Org3_x	a 3rd tier organization (usually a DSP or user organization)

Extension	**Description**
Addr blocks(s)	one or more IP address blocks assigned to the organization

Figure 4.1: The PKI as address allocation structure, schematically explained. Organizations denoted with Org1 are usually RIRs, while those denoted with Org2 are usually LIRs, ISPs, or DSPs. Source: Kent et al., 2000

An autonomous system claiming ownership of a certain prefix or propagating a route is also called an attestation, with the former being an address attestation and the latter being a route attestation. These attestations are signed using asymmetric key cryptography. To verify that, when an autonomous system broadcasts an attestation, the AS in question is authorized to broadcast said message to other ASes, the PKIs are used. If AS$_j$ gets a route attestation from AS$_i$, then AS$_j$ needs not only an address attestation and a certificate from the address allocation PKI, but it also needs a route attestation from each AS from the source to AS$_i$, as well as a certificate from the router assignment PKI for each AS, proving that the routers that sent this route do belong to the AS that they claim to belong to. If an address attestation is received, only the address attestation and the certificate from the corresponding PKI is necessary to verify that the autonomous system attesting to own the prefix does own the prefix.

IPsec is the final part of the S-BGP protocol. Because BGP is transported over TCP, it is protected against out-of-sequence packets. However, that is not enough. The Encapsulating Security Payload (ESP) protocol with NULL encryption from IPsec is used for encrypting BGP messages. The Internet Key Exchange handles key management for ESP. It is not clear why the authors decided to use ESP instead of the Authentication Header (AH) that is also available in the IPsec suite, but one reason for this could be that ESP has more features than AH. AH provides data integrity, origin authentication, and protection against replay attacks for IP datagrams [89], whereas ESP provides those features as well as confidentiality and limited traffic-flow confidentiality [90]. AH tends to be the better option when the main concern is authentication [1], but the main concern for S-BGP is the encryption of messages.

If the protocol would be implemented correctly, then S-BGP speakers would be able to verify that:

- Autonomous systems that claim to own a certain prefix do own that prefix
- Routes broadcasted by ASes are actual routes towards the IP prefix and have not been altered in any way
- Every AS along a route has propagated said route
- A router claiming to belong to a certain AS does belong to the mentioned AS

However, the amount of processing power necessary to do so makes this protocol unwieldy. Kent et al. analyzed the computational overhead that S-BGP would introduce and concluded that it would require the computational power of a desktop PC for the protocol to function [94]. However, this was back in 2000, and computational capacities of modern routers surpass those of a desktop PC from when the analysis was made. A high-end PC from 2000 would have a processor with a processing power of 1 GHz [15], whereas modern routers have CPUs with higher clock speeds than 1 GHz [3]. Also, while the protocol is incrementally deployable as it would be easy for an S-BGP speaking router to switch back to speaking BGP to routers that have not adopted it, doing so would logically mean that the route can no longer be secured from that point on. Aside from this, Goldberg et al. have modelled full adoption of certain security solutions, one of which being S-BGP, and have shown that attack strategies that involve manipulating the AS's export policies would be effective enough to be still able to reroute traffic [75]. This shows that S-BGP does not protect against protocol manipulation attacks. The fact that there are effective attack strategies already being thought of before the solution has seen any kind of wide-spread adoption is detrimental to the benefit of adopting the solution. Finally, due to the hierarchical structure of PKIs, using them would introduce a single point of failure for the Internet, and the correct functioning of the Internet would be reliant on the PKIs being available. This single point of failure would mean that any attacker that can successfully attack it would cause severe disruptions to the accessibility of the Internet. These factors combined make the protocol impractical, and as such, it has never seen widespread adoption.

4.2. soBGP

Secure Origin BGP (soBGP) was developed to combat BGP vulnerabilities without requiring full centralization. The main goal of soBGP is to validate that an AS is authorized to originate a prefix that it claims to own [126]. It also verifies whether or not a peer that advertises a prefix has at least one valid path towards the destination. The design requirements of soBGP show that its design is a response to the centralized nature of S-BGP, and some design requirements that show this are as follows:

- Minimize impact on current implementations of BGP
- Must use decentralized processing and trust; any kind of reliance on a central authority is prohibited as this would introduce a single point of failure
- It should not require any data downloads
- The solution must provide some level of security without every other AS participating

soBGP uses three types of certificates to fulfill these requirements. These certificates are used for advertising and correlating the identity of an AS, prefix ownership, and route policy. Each of these is validated by using a web-of-trust model. These three types of certificates and what they do are:

- Entity certificates used to establish and validate the identity of an AS, and also to establish that a BGP speaker belongs to an AS
- Authorization certificates used to assign prefixes to ASes and verify that ownership claims are correct
- Policy certificates used to define policies on a per-AS basis

These certificates are transported in a new BGP message type, the SECURITY message. BGP can support more types than just the four types that already exist due to the Type field in a BGP header having more than two bits, as seen in figure 3.1. Also, doing so would similarly allow for propagation to how current UPDATE messages are already propagated.

Figure 4.2 shows how the different certificates would be used to secure BGP. The certificates are used to build databases for the BGP speaker to refer to, to verify data. The policy certificates are also used for the creation of a topology graph, which can be used to verify the feasibility of an advertised path to a prefix. The ability to verify this path would be dependent on how complete the topology graph is. One thing to note is that, for both entity and authorization certificates, the signer AS and the subject AS do not need to be the same AS. Any AS can sign an entity certificate for another AS containing the number and the public key of the other AS. This creates a web-of-trust model where ASes need to trust on other ASes that the information that they get is correct. A local administrator seeds known public keys.

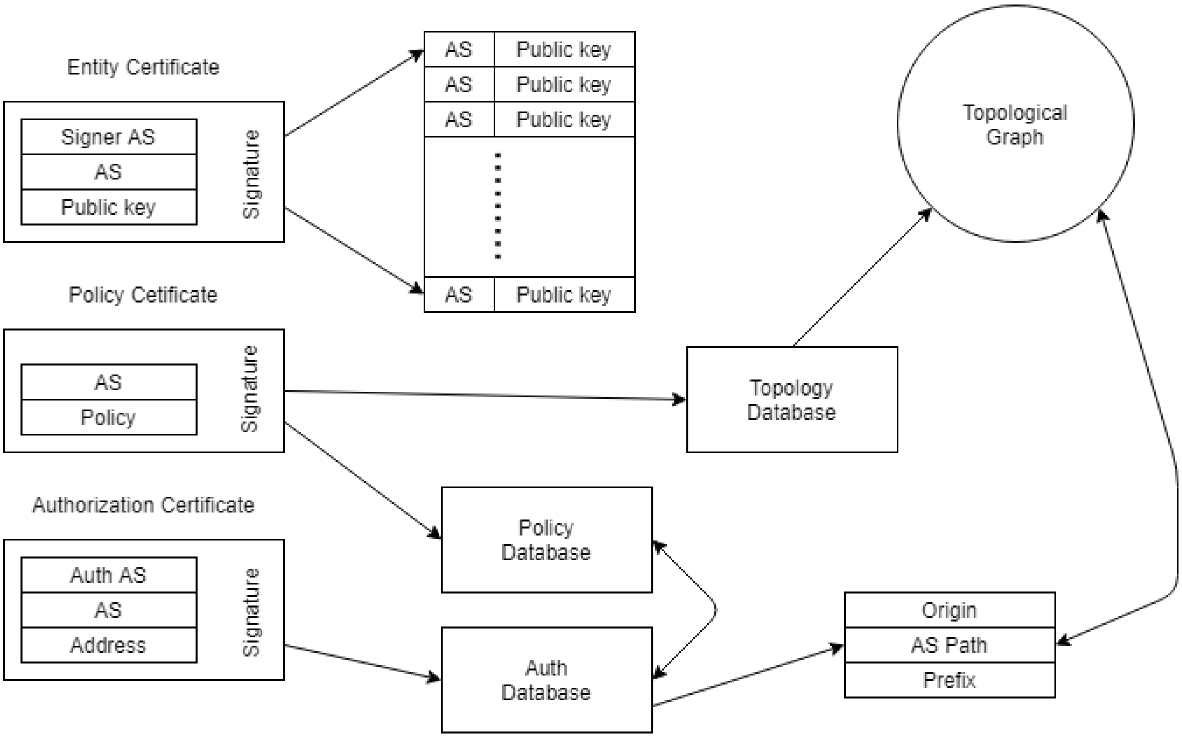


Figure 4.2: How the different certificates are used in securing BGP through soBGP. The information in them is used to create databases that are then used to verify incoming information. Source: Retana, 2003.

The amount of memory required for storing all the certificates is, logically, dependent on the amount of ASes and the number of authorized blocks. However, the certificates can be stored in a data centre (if necessary) in the AS that is connected in one way or another to the BGP speakers at the edge of the AS. In doing so, the certificate processing can also be offloaded to the servers, thus preventing the need to upgrade routers or router firmware to make it possible. These servers can also exchange their certificates to build an even more complete database.

In terms of security through incremental deployment, soBGP provides some security through incremental deployment (which is also in contrast with S-BGP). Still, the amount of security provided is proportional to how many ASes deploy soBGP. Using the topology that was created using the policy certificates, soBGP capable ASes can always verify the next hop in the AS path. Next-hop verification would not help a lot in checking the feasibility of a path however, as paths tend to be longer than one hop. ASes that run soBGP can also exchange their certificates with one another (and they do not need to be directly connected to do this), allowing for more complete overviews of the topology as well as more complete overviews of what AS owns what prefix. Because of this, the more ASes that deploy soBGP, the more secure their routing will be.

As can be seen, soBGP allows for benefits through incremental deployment, as well as security without relying on a central authority. However, due to the correlation between the benefits of deployment and the amount of ASes deploying the solution, the security benefits of a single AS deploying soBGP are still minimal. It also requires either upgrades to existing routers (hardware or software) or extra servers and infrastructure to process and transfer the certificates. It also only verifies that a certain AS owns the prefix that it claims to own; path authentication is only based on the feasibility of a path, not the integrity. The minimal gain in security from only a single AS deploying it is probably why it has not seen widespread deployment as well.

4.3. psBGP

In 2004, a new security solution was proposed, called Pretty Secure BGP (psBGP), and it was based on analysis of both security and practicality of S-BGP as well as soBGP [142]. The design goal of psBGP is to consider the best features of both solutions and combine them while leaving the worst out. This is most clear in the fact that it uses both centralized trust (a property inherent to S-BGP) as well as decentralized trust (a property that soBGP capitalizes on because it intended to completely rid BGP security from centralization and all the issues that come with it).

To develop a secure solution for BGP, security goals for the solution should be developed first. The security goals for psBGP are as follows:

1. AS number authentication: it must be verifiable that an entity using a certain AS number AS_i does represent the AS that it claims to be part of.

2. BGP speaker authentication: it must be verifiable that a BGP speaker that claims to speak on behalf of a certain AS has been authorized by the AS to speak on their behalf.

3. Data integrity: it must be verifiable that a third party has not modified a message.

4. Prefix Origin Verification: if a certain AS originates a given prefix, then it must be verifiable that that AS does have the claimed prefix or a set of prefixes that, when aggregated, make up the prefix the AS claims to own.

5. AS path verification: given a path of autonomous systems, it must be verifiable that the message has been propagated from the first AS in the path to the final AS in the path.

In psBGP, a centralized trust model is used for AS number authentication, whereas a decentralized trust model is used for verifying IP prefix ownership. For authenticating AS numbers and public keys, PKIs are used, which in contrast to those used in S-BGP, are not centralized at the highest possible level, the IANA. Instead, the root certificate authorities, in this case, are the RIRs. So instead of there being a single point of failure for the whole Internet, there is now a single point of failure per continent. Each AS is issued a public key certificate called the ASNumCert, and an AS with this certificate creates and then signs two data structures. These are a SpeakerCert for a public key and an AS number, binding the two together, and a prefix assertion list (PAL), to list what prefixes this AS and its neighbours assert to own. Figure 4.3 gives a small overview. This certificate binds an AS number to both a public key and a list of prefixes that both the AS and its neighbours own.

The SpeakerCert is also used to verify that a certain BGP speaker can speak on behalf of a certain AS. The proposal considers three design choices for BGP speaker authentication:

1. Each BGP speaker is issued a unique public key certificate to verify that it belongs to the AS it claims to belong to.

2. Each BGP speaker has a unique private key but shares a public key certificate with other speakers in the same AS. This scheme would work similarly to group signatures. The authors refer to the work of Boneh et al. for a more detailed explanation [60].

3. All BGP speakers in an AS share a public-private key pair. This scheme would also work similar to a group signature scheme, but the private key would be AS-specific instead of BGP speaker-specific as it would be shared between all BGP speakers in an AS.

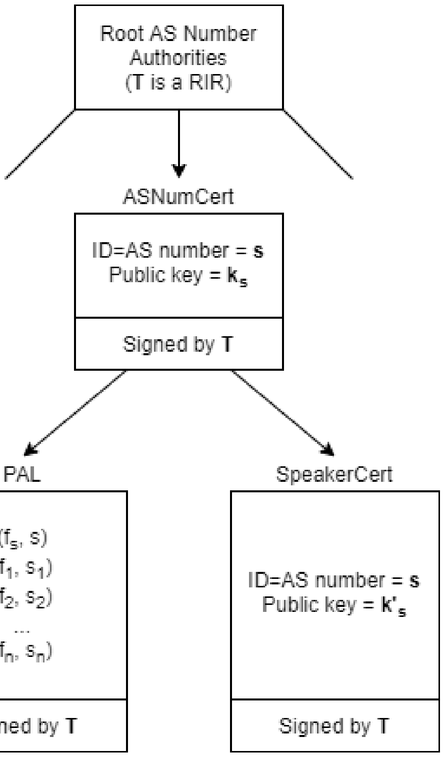


Figure 4.3: how the certificate structure of psBGP works. ASNumCerts are signed by the trust anchor T, which in this case, is a RIR. The RIR binds one public key k_s to AS s, and then s uses k_s to sign both the SpeakerCert, binding another public key k'_s to s, as well as the PAL. The PAL is an ordered list, with s and its prefix f_s at the top, and then other AS numbers and their prefixes following it, in order of AS number. Source: Wan et al, 2005

For the proposal, the third choice is made because it is the best trade-off between security and complexity. The first choice also discloses the identity of the BGP speakers, which could cause more security problems. Recall from figure 4.3 that a SpeakerCert binds a second public key to an AS. The second public key has not been used yet in the figure. This public key is the public key shared between all the BGP speakers in a single AS. The private key mentioned in the third choice is the second part of the public-private key pair. It is also shared among all the BGP speakers in an AS. It is used for signing BGP messages and also for establishing secure connections between BGP peers.

Data integrity in psBGP is facilitated using the same methods as S-BGP and soBGP. The proposal uses IPsec ESP with null encryption for the protection of BGP sessions. There is no reason stated by the authors why they use ESP over a different protocol from the IPsec suite such as AH. But the reason is probably similar to the reason for using ESP over AH in S-BGP. The main concern is encryption, not authentication, so ESP is a better choice.

Verification of prefix ownership in psBGP happens in a decentralized fashion and is done by checking the consistency of assertions between the PALs of two different ASes. To understand this, one must first understand the comparability of two assertions: two assertions made by two different ASes are comparable if they are both about the same AS number (ASN), and the asserted prefixes for each of them is not an empty set. Then, consistency is as follows: two comparable assertions are consistent if the set of asserted prefixes is the same for both assertions. Then, given that a certain AS s has n peers, an assertion is k-proper if, among s and its peers, there are k consistent assertions of a (set of) prefix(es) being owned by a certain AS. If $k = n+1$, that means that both s and all of its peers agree that s owns a set of prefixes, providing maximum confidence in the assertion. This maximum confidence can be tarnished by a single AS that makes an inconsistent claim. Verification of AS-PATH correctness is a little more tricky because there is no consensus on what constitutes the security of a path. Guaranteeing AS-PATH integrity prevents an attacker from modifying an AS-PATH, providing the closest thing to "AS-PATH security".

As can be seen, psBGP uses both centralized and decentralized methods to verify BGP messages. It improves on both S-BGP and soBGP by, as the design goal stated, combining the best features of both of these solutions into a single solution. It provides better security than soBGP as it performs integrity-based path verification, while it is less centralized than S-BGP and uses fewer certificates, especially for BGP speaker authentication. However, it seems that this solution needs to be adopted by a large portion of the Internet for the solution to provide any security benefits. Especially due to peers needing to have PAL's to verify prefix ownership of a participating AS. ASes that do not deploy this solution do not have these PALs. As such, the assertion that an AS owns a prefix can be less trustworthy simply because other ASes have not deployed the solution. This lack of PALs makes deploying the protocol a problem as a substantial amount of ASes already need to have deployed the solution for it to have any real security benefits. This is the same Catch-22 situation with S-BGP, but with less centralization.

4.4. IRV

As a different approach to securing BGP, the Interdomain Route Validation (IRV) service is a new protocol that does not intend to replace BGP completely; rather, it intends to act as a companion to BGP and acts as a separate protocol [76]. It is used to validate BGP data and acquire additional routing information that is relevant to a given AS. As such, the protocol can be deployed as a service within an AS that it represents.

The way IRV works is as follows: IRV performs origin validation by querying ASes where the data came from. This could be done by querying the origin AS at the time an UPDATE is received. Still, a smarter way of doing so is to queue sets of UPDATE messages, group those that have the same origin AS together, and then validate these groups using a single query. This reduces the number of queries necessary and, in turn, reduces the time needed for verification of UPDATE messages. IRV can also perform path validation by querying each AS in a given path. A way to further cut down on unnecessary querying would be to cache previously gained policy and route information that has been verified already. The paper does not state a way to do so, but one way to cache route information, for example, would be to create a topological map of ASes that are close-by. Figure 4.4 shows how IRV works: IRV servers/systems of a certain AS are queried by those of other ASes upon receiving UPDATE messages and route announcements, and these queries are then verified using the original ASes.

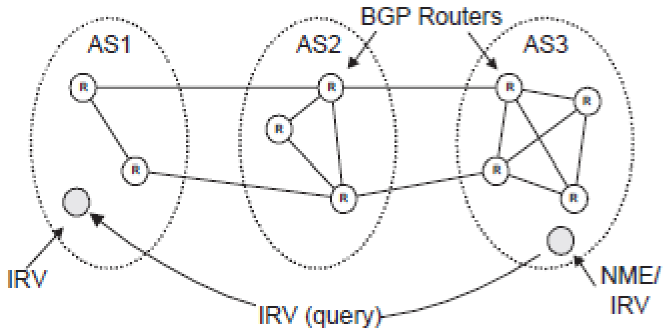


Figure 4.4: How IRV works. When AS3 receives a route from AS1, the server hosting IRV as a service in AS3 queries the one in AS1. One thing that can be derived from this image is that IRV does not have to be deployed in every single AS to secure an origin announcement, because AS2 does not have to use IRV for AS3 to verify that certain UPDATE messages come from AS1.

Because of how IRV can be implemented as a service on a server within a given AS, other ASes should know how to find the server hosting the scheme. The authors of the paper present several options for tackling this problem. The first of these is to include a hint address of each AS's IRV-hosting server within UPDATE messages. However, this would require modifying the existing BGP protocol, which is something that a standalone system is designed to prevent as much as possible. A different approach that would not involve changing BGP would be to use a well-known registry to store authoritative IRV contact information per AS. However this would probably include a centralizing factor (such as a PKI), causing problems with that as well.

Another problem with IRV is the authentication of queries to prevent unauthorized access to any sensitive data. The paper proposes to verify the authenticity of queries and their responses by using digital signatures. This approach does mean that there has to be some way to distribute public keys, but the computational overhead and the time cost of signing messages can be reduced by caching and reusing frequent requests and responses.

The IRV query system can also be extended to include not only queries but also reports. These are lists containing interdomain routing data and are sent voluntarily. Exchanging these reports can improve connectivity and general availability. Because of its voluntary nature, there should ideally be an incentive for different ASes to share this information, especially because this BGP information from outside sources can be used to infer business relationships between autonomous systems. The way this works is as follows: recall that ASes will not broadcast all of their routes to all of their neighbouring ASes. ASes only export a portion of their routes, that portion being their routes and those of their customers, to the provider or peer ASes. Because of this selective propagation of routes, different ASes will receive different routes to the same location, given that they have different provider ASes. These routes indicate business relationships but are not received by other provider ASes unless a route leak happens. When this routing information is shared however, different provider ASes can learn which ASes are customers of rival provider ASes. This can then be abused for selective advertising to get more customers. The paper suggests introducing a system of preferential treatment between ASes that share their reports.

IRV is, by design, expandable and does not suffer from the same setbacks as some earlier discussed solutions, mainly due to not relying on a central authority and also because it acts as an additional service to prevent attacks on BGP. One can see it as an improved version of soBGP because it performs actual path verification instead of just providing path feasibility checking while also being decentralized. Also, by verifying that the UPDATE messages did originate from a given AS, IRV provides some form of defence against protocol manipulation attacks, being the only solution to do so as far as I have found. However, the base version introduces a lot of overhead, requiring near-constant querying of previous ASes. As mentioned, the paper introduces several ways to reduce the amount of overhead this solution introduces, mainly by caching answers. It also still requires other ASes to adopt it for it to work well. Path verification can only verify AS pair segments of the path where both ASes have a working implementation of IRV running.

4.5. SPV

SPV is another approach to securing BGP with the intent to extend the existing protocol by adding a path attribute. The abbreviation stands for Secure Path Vector and is, just like IRV, a service that exists alongside BGP [84]. The goal of SPV is to verify the integrity of a given path using only symmetrical cryptographic primitives. This is to prevent routers from having to perform public-key cryptographic operations (which introduce a lot of computational overhead). Note that path authentication includes origin authentication because every path has an origin, and as such, the origin also needs to be authenticated. This, however, is not done using symmetrical cryptographic functions.

In SPV, there are four different types of keys, two of which are public/private key pairs. This would imply that these key pairs are used for public-key cryptography to verify the integrity of a path, but they are not. These key types are as follows:

- A single-ASN public/private key pair, of which the public one authenticates the signature of one AS in the ASPATH, and the private one is used as a seed in a pseudo-random function to derive the one-time signature and the public key.
- An epoch public key, which authenticates one ASPATH protector, and consists of a sequence of the aforementioned one-time signatures.
- A multi-epoch public key, which authenticates multiple epoch public keys.
- A prefix public/private key, of which both are used to authenticate messages from a given prefix. The public key of this key pair is to authenticate multi-epoch public keys, which produces the multi-epoch public key certificate.

SPV uses certificates for attesting prefix ownership. These certificates are equivalent to the address space PKI structure of S-BGP. At each step in the delegation of address space, the recipient of a part of the address space generates an asymmetric prefix private key. This is used to sign the prefix public key of the delegated block. This key is combined with a list of prefixes that have been delegated to the new key. This then forms the prefix public key certificate, also called the prefix certificate.

SPV does not sign UPDATE messages of BGP, and instead uses an ASPATH protector which is built using only symmetric cryptographic primitives and can be authenticated using the epoch public key. This is a data structure that has to have the following two properties: one, an attacker cannot claim a shorter route to a prefix. Two, an attacker cannot modify the AS numbers that have been inserted already in the path. In doing so, an adversarial AS is unable to alter paths in any way. Generating an ASPATH protector is done by selecting a random key X, and then a pseudo-random function F is used with the seed X to generate seeds for epoch e: $c_{i,e} = F_X(e)$. These seeds span the individual ASPATH protectors for the given epoch. The benefit of creating seeds this way is that an AS only needs the pseudo-random function (PRF) and the seed to derive all of the generated values. Securing an ASPATH of l ASes requires only l of these seeds. Each of these seeds $c_{i,e}$ is a one-time signature and they are all used to derive n values $b_{i,1,e}$ to $b_{i,n,e}$. It is also the single-ASN private key for epoch i. This is done by using the private key $c_{i,e}$ as the seed for the PRF F and j as the input: $b_{i,j,e} = F_{c_{i,e}}(j)$

These final values $b_{i,j,e}$ for all i and j are then used as the base values for the construction of a hash tree by the AS. Hash trees are explained in more detail in the background. To oversimplify their purpose, they are useful for authenticating a single value in a series of values while keeping the rest of those values a secret. To do so, the values $b_{i,j,e}$ are first blinded by using a one-way function such as a hash function to generate $b'_{i,j,e} = H[b_{i,j,e}]$. The root value of this hash tree (called $r_{i,e}$) is the single-ASN public key corresponding to the single-ASN private key $c_{i,e}$. Another hash tree is then built over all of the $r_{i,e}$ values, and the resulting root value r_e is then called the epoch public key, as it can be used to verify all of the other one-time signatures in the ASPATH protector in a single epoch. Figure 4.5 shows a diagram of such an ASPATH protector. The box marked "C" contains the four single-ASN private keys, and each of them is a hash of the previous one: $c_{i+1,e} = H[c_{i,e}]$. This creates a hash chain where each value is a hash of the previous one. These $c_{i,e}$ values are then used to generate the different values $b_{i,j,e}$ for (in this case) $i \in [1,4], j \in [1,4]$. These initial values are then used in the creation of the first hash trees, of which the root is highlighted in the box marked "B." This box highlights the single-ASN public keys. These are then used as input (without blinding them first) in the construction of a new hash tree, of which the root is r_0, which is the epoch public key of epoch 0.

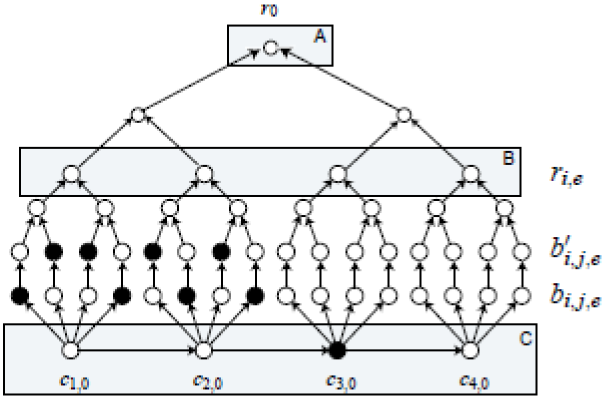


Figure 4.5: a diagram of an ASPATH protector. The epoch is 0. This ASPATH protector can secure paths with a length of four. Source: Hu et al., 2004.

The ASPATH protector is used as follows: in each epoch, the owner of a prefix uses an ASPATH to announce its prefix. As the ASPATH grows, the ASes sign their ASN into the protector by removing the current single-ASN private key $c_{i,e}$, and then each AS uses the next single ASN private key by passing $c_{i+1,e} = H[c_{i,e}]$ to the next AS. The authors mention the following example: suppose that there are three ASes with numbers A, B, and C. AS A signs $H[A]$ with the HORS (Hash to Obtain Random Subset) signature spanned by the value $c_{1,0}$. The HORS signature scheme is a signature scheme that takes inputs k and t, requires one hash function evaluation for signing and creates (as a signature) a set of values spanned by t requiring 17 hash functions for verification

[127]. This signature is sent to B along with the value $c_{2,0}$. B can then verify all the one-time signatures by recomputing all of them and verifying that the final root value of this computation matches the root value of the ASPATH protector. Based on the signature of AS A, it can compute $r_{1,0}$, because it can infer $c_{1,0}$ from the HORS signature that was sent. This can then be used to derive the values $b_{i,j,e}$ using the PRF. Computation of $r_{i,0}, i > 1$ can be done by first computing $c_{i,0}, i > 1$ (which can be done because B has $c_{2,0} = H[c_{1,0}]$) and calculating the respective values of $r_{i,0}$ using those initial values. B can then compute the root of the hash tree using the values $r_{i,0}$, and check if it matches r_0. B then signs $H[\langle A, B\rangle]$, where $\langle A, B\rangle$ denotes an ordered set. The same HORS one-time signature mechanism is used, and the spanning value used is $c_{2,0}$. B then sends A's and B's signature, combined with $c_{3,0}$ to C. C then repeats the process that B performed. This process eventually results in a lot of ASPATH protectors, which all have a root value of r_i, where i is an integer value. These values can be authenticated using another hash tree, which takes the different values of r_i as input (these don't have to be hashed first as they are publicly known, and as such, they are not confidential). This hash tree's root value is the multi-epoch public key, which is then signed by the prefix public key to form the multi-epoch certificate. Figure 4.6 shows such a hash tree constructed over 16 ASPATH protectors.

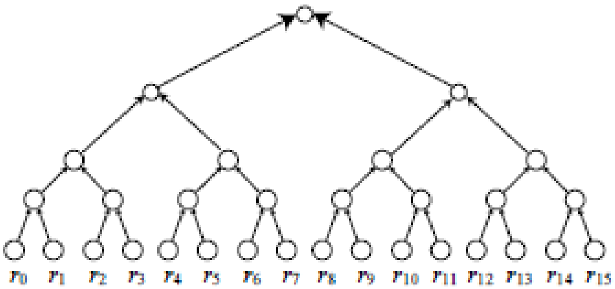


Figure 4.6: A hash tree constructed over the epoch public keys of 16 ASPATH protectors. Similar to the figure of the ASPATH protector, it assumes that the amount of epoch public keys is equal to 2^n for an integer value n, but this does not have to be the case for hash trees. Source: Hu et al., 2004

The paper also mentions some of the shortcomings of the basic ASPATH protector that has been explained here, as well as how to solve them. These weaknesses are repeatable and predictable fraud, single malicious AS fraud, and multi-path truncation attacks being possible. Repeatable and predictable fraud occurs through hash collisions (this is, two different values M and M' have the same hash: $H[M] = H[M']$). If an AS C receives a path from B with the path being A → B, It can modify the path to become A → C if $H[\langle A, B\rangle] = H[\langle A, C\rangle]$. One way to counteract this is to prepend the epoch number to each hash operation: instead of computing $H[\langle A, B\rangle]$, $H[e||\langle A, B\rangle]$ is to be computed. This makes it vastly less likely that the attacker can change paths. Single malicious AS fraud is possible because an AS is under no obligation to sign its ASN into the path. Querying the previous AS to check if the correct ASN was appended is possible, but a better solution is to require the AS to sign the next ASN into the ASPATH protector: if a path A → B is sent to C, then B signs $H[e||\langle A, B, C\rangle]$. C would then check if the one-time signature already encodes itself. The multi-path truncation attack can happen when an AS receives two paths to the same destination with different lengths, in which case the attacker can use the single-ASN private key of the first (shorter) path to modify the other path. The ideal way to counteract this is a way described in the paper as *postmodification*: the quality of a single-ASN signature "degrades" as it travels farther. This is done by using the existing values $b'_{i,j,e}$ as "semi-private" values and adding another layer of values $b''_{i,j,e} = H[b'_{i,j,e}]$ to be used as input for the ASPATH protector. Then, as the amount of hops increases, the number of private values revealed to a BGP speaker decreases, and the amount of semi-private values revealed increases. Limiting the number of old private values that are revealed at each step along the way would mean that an attacker does not get these values if the path is long enough, preventing path modification from being possible on longer paths.

SPV is by design an incrementally deployable solution. The one-way hashing of all important variables allows for ASes that deploy it to construct path protectors not only for themselves but for all the ASes between them and the previous AS that deployed SPV, acting as if all the ASes deployed SPV. For this, however, the SPV relevant data should continue to be forwarded between ASes. It also cuts out BGP speaker verification in favour of more effective origin and path verification, most likely because BGP speaker impersonation happens far less frequently than BGP origin hijacks or path altering attacks. However, there is also a lot of computational and network overhead: the paper mentioned that, in simulations for evaluating the overhead incurred by deploying SPV relative to that incurred by deploying S-BGP, the network overhead of SPV was almost three times as high as that of S-BGP. This result takes into account the advanced version of SPV being used with all the countermeasures against attacks being implemented. On the flip side, though, from the simulations performed by the authors of the paper, it is computationally far faster, with the possibility of it being up to 22 times as fast compared to S-BGP. The highest speeds relative to S-BGP are achieved in more interconnected network areas. Hu et al. do still suggest hardware acceleration for more densely connected networks, which makes it slightly less practical as this would involve replacing the hardware of BGP speakers. Still, a software implementation of SPV should be sufficient for more sparely connected networks.

4.6. HC-BGP

Hash-chain BGP (HC-BGP) is a solution developed after solutions such as SPV that aim to provide an efficient way of verifying paths. In contrast, this solution aims to provide an efficient way of doing the same for origin attestations [154]. The goal is to provide prefix ownership security by fulfilling the following requirements:

1. Ensuring origin attestation: it should prevent (sub)prefix hijacks from having an impact on the routing of the Internet.

2. Flexible: it should allow for multiple ASes to own the same prefix if these ASes do both own the prefix. While a prefix being owned by more than one AS is indicative of a possible prefix hijack, this needs not to be the case [157].

3. Incrementally deployable: it should not require all of the ASes of the Internet to adopt the protocol at the same time for it to have any impact.

4. Light-weight: it should not incur too much overhead, whether that be computational overhead, network overhead, or storage overhead.

It makes significant use of the one-way hash chain mechanism to achieve this goal, which is where it gets its name from (HC stands for Hash Chain). Furthermore, the solution exploits two key characteristics of prefix announcement over the Internet: for each announced prefix, the set of origin ASes is stable and does not change very often, and the (de-)aggregation for each prefix is infrequent.

As mentioned before, the solution makes a lot of use of one-way hash chains. This is a method that was first proposed back in 1981 by Lamport to secure passwords [100]. Of course, nowadays, there are far better ways to secure passwords, but the technique remains a lightweight cryptographic method for providing security. The method works as follows: using a hash function h, a client first notifies a server of an initial value $h^n(s)$, which is the value obtained by repeatedly hashing the outcome of $h(s)$ n times. This value needs to be sent to the server via secure communication channels. When a client wants to interact with the server, and it needs to verify that the client is who they claim to be, the client needs only to provide $k = h^{n-1}(s)$ to the server, after which the server can check whether or not $h^n(s)$ is equal to $h(k)$. One requirement for the hash functions that are used for this protocol is that they must be *second pre-image collision-resistant*: given the hash of a message $h^{i+1}(s)$ for any number of hashes i, it must be computationally hard to find another s' such that $h(s') = h^{i+1}(s)$. Furthermore, instead of relying on centralized trust by using a global PKI, HC-BGP relies on neighbouring ASes to trust one another, which should be easy to implement in practice because of currently existing commercial agreements, the authors argue.

The protocol itself only deals with origin verification, and as such, consists of only three elements: initialization, prefix announcement, and prefix withdrawal. An overview of each of these parts is as follows:

1. Initialization: each prefix p has an initial value s_p, and the prefix owner R_i sends to each of its neighbours R_{i+1} this value. These values are then propagated throughout the network. For each prefix p that R_i owns, the initial hash chain value $h^n(s_p)$ is calculated, and then that value encrypted using the private key of R_i k_i^- is used: $(h^n(s_p)_{k_i^-})$ is sent to all R_{i+1}. Each R_{i+1} then decrypts the message using R_i's public key, verifies and stores the value as belonging to that prefix, and then propagates the value encrypted with its own private key: $(h^n(s_p))_{k_{i+1}^-}$ is sent to other neighbours.

2. Prefix announcement: because of the many cases of this part, it has been split up into what the sender does and what the receiver does to make it more clear:

   - Sender: if the prefix that is going to be announced p is not a subprefix of another announced prefix $\hat{p}$, and the origin AS has not changed, then R_i announces $(p, h^{n-1}(s_p))$. Otherwise, if it is a subprefix of $\hat{p}$, send $(p, h^{n-1}(s_p), h^{c_{\hat{p}}-1}(s_{\hat{p}}))$, where $h^c(s)$ stands for the current value of the hash chain for message s. Finally, if the origin AS has changed, R_i announces p with $(p, h^{c-1}(s_p))$ and the new origin AS.
   - Receiver: the receiver R_{i+1} receives an announcement for p with hash value $h^c(s_p)$. The following can happen:
     - p exists in the table, and the origin AS has not changed: accept.
     - p exists but the origin AS has changed and $h(h^c(s_p))$ is equal to the stored value for that prefix: accept and store $h^c(s_p)$.
     - p is a new subprefix of an existing overarching prefix $\hat{p}$ and the origin AS has not changed: accept and store $h^{n-1}(s_p)$.
     - p is a new subprefix of an existing overarching prefix $\hat{p}$ and the origin AS has changed and $h(h^c(s_p))$ is equal to the stored value for that prefix: accept and store both $h^c(s_{\hat{p}})$ as well as $h^{n-1}(s_p)$.
     - In all other cases: reject.

3. Withdrawal: R_i sends a withdrawal with $(p, h^{c-1}(s_p))$ to its neighbours to withdraw prefix p and its latest hash chain value $h^c(s_p)$. Once a withdrawal for p has been received by R_{i+1} with value $h^c(s_p)$, then if $h(h^c(s_p))$ is equal to the stored value for p and the withdrawal is announced with the updated $h^c(s_p)$, it is accepted and $h^c(s_p)$ is stored. Otherwise, it is rejected. If $h^c(s_p)$ has not been updated, then the withdrawal is with an old hash value and accepted automatically. Once withdrawn, the receiver does not accept an announcement for p with an old hash chain value.

How the protocol prevents hijacking is shown in figure 4.7. In the first case, with the full prefix hijack, AS2 will notice that the origin has changed. However, the values $h^{1000}(s1)$ and $h(h^{999}(s2))$ are not the same, because the hash function chosen is second pre-image collision-resistant, as explained earlier. Because of this, it does not match the second case in the list of cases for the reception of a prefix announcement, nor any other case for that matter. Because of that, it is rejected. The same happens in the case of the sub-prefix hijack.

As can be seen, HC-BGP is built with reliance solely on light-weight cryptographic primitives in mind. Not only does it only require light-weight cryptography, but it also only requires it sparingly, because as seen before, the prefixes that each AS announces remain constant. The authors compare the computational overhead to S-BGP. They claim that, even if HC-BGP were at the same level of computational complexity as S-BGP, it would introduce orders of magnitude less computational overhead because of the rate of origin changes. However, HC-BGP provides less security than S-BGP, as S-BGP is developed to secure more than just origin attestations.

4.7. BGPcoin

BGPcoin is a security solution, focusing on securing against prefix hijacking, which makes use of the blockchain and, as the name suggests, has its roots in how cryptocurrencies operate [148]. It is the most recent solution that will be considered in this thesis. The authors argue that a security solution for origin attestation verification should meet several requirements:

1. Allocations and updates should be consistent globally, and all parties on the Internet should be able to see it

2. The solution needs to have acceptable performance and scalability to keep up with updates happening across the Internet (scalability is the main problem here; from HC-BGP, we can see that the set of prefixes owned by ASes does not update very frequently)

3. Audit requests must be resilient to tampering

4. When a resource is revoked, the resource owner must consent to the revocation

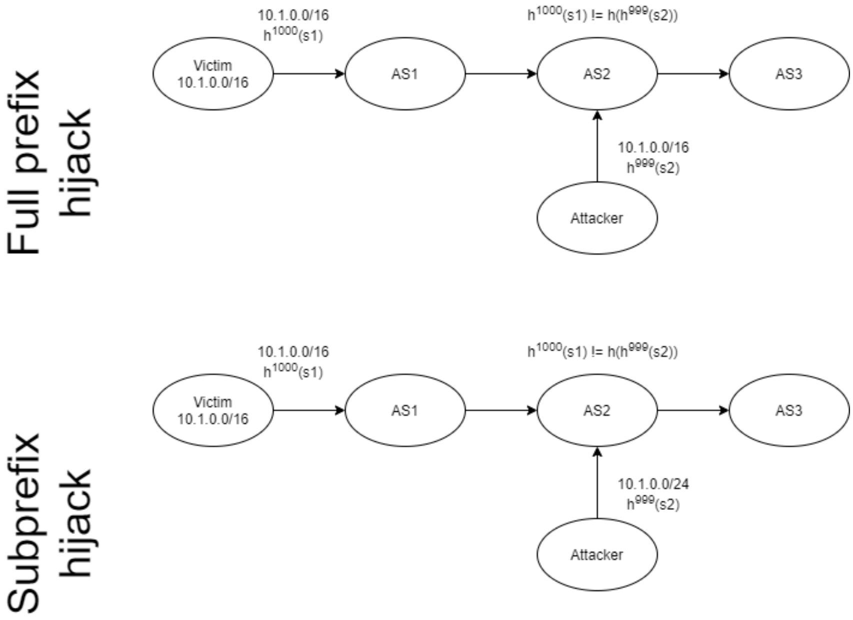


Figure 4.7: how HC-BGP prevents both full prefix hijacks as well as sub-prefix hijacks. In both cases, the victim AS owns prefix 10.1.0.0/16. Source: Zhang et al., 2009

5. The solution must be cost-effective and easy to deploy, as well as provide substantial benefit even when only a limited number of entities deploy the solution, i.e., it should be incrementally deployable

BGPcoin itself is controlled by smart contracts that allow entities to manage their Internet number resources, those being their prefix(es) and the ASN assigned to them. As the Ethereum blockchain allows users to program these smart contracts [19], the system is hosted in the Ethereum blockchain. In BGPcoin, there are five types of participants, following the hierarchy of parties delegating prefixes across the Internet: the IANA, the RIRs, the NIRs/LIRs, the ISPs, and finally all of the other entities. The smart contracts of BGPcoin also contain three basic functions that are used in securing prefix ownership: resource trading, aggregated Internet address repositing/updating, and resource sharing. To achieve these functions, BGPcoin has four different types of trading operations for prefixes, three types of operations for ASNs, and two types of operations for route origin advertisements (ROAs). These are shown in table 4.1 and their usage is shown in figure 4.8. One thing to note is that an earlier version of the protocol did not have the operations for ROAs, but had one more operation for prefixes, that being the "update prefix" operation [147].

Operation	Semantics
IP register	IANA → RIR: <IPB,∅>
IP allocate	xIR → xIR:<IPB,∅>
IP assign	xIR → xIR/ISP:<IPB,∅>
IP revoke	xIR/ISP → xIR:<IPB,∅>
ROA add	xIR/ISP:<IPB, ASN>
ROA delete	xIR:<IPB, ASN> → <IPB,∅>
ASN register	xIR:<ASN,-,->
ASN allocate	xIR → ISP: <ASN, stime, period>
ASN update	ISP:<ASN, stime′, period>

Table 4.1: The smart contract operations for prefixes, ROA's, and ASN's in BGPcoin. From this table, one can infer that the IP and ASN operations deal with the allocation of prefixes, respectively AS numbers, while the ROA operations bind the two together. xIR can be any one of the following: RIR, NIR, LIR. Source: Xing et al., 2018

In the BGPcoin system, prefixes and ASNs become resources that are delegated to ASes via a hierarchy and

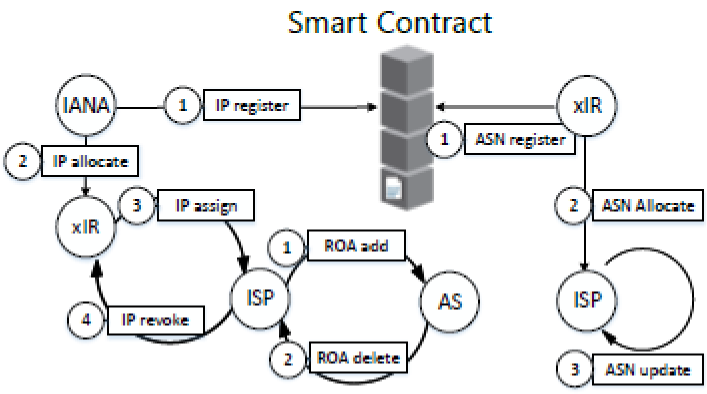


Figure 4.8: How the different operations are used. In this image, we can see the hierarchy of IANA - xIR - ISP - other ASes more clearly, especially when looking at the left part of the image containing the process for IP/ROA registering. One thing that is also more clearly shown than in the table is the IP operations delegating the prefixes and the ROA operations binding prefixes to ASes. Source: Xing et al., 2018

can be traded among participating ASes. ASes as resources are modelled as <ASN, RIR, owner, stime, vperiod>, where ASN is the AS number in question, RIR is the RIR that registered it, the owner is the owner of the ASN, the stime is the time at which an AS further down the hierarchy (NIR or lower) obtained the ASN and the vperiod states the amount of time that this ownership stays valid for. Prefixes, on the other hand, are modelled as a record in the form of <prefix, state, RIR, NIR, owner, leasee>. The prefix contains the IP prefix, and the state is the state of the prefix, which is one of the following states, which can be inferred from the different operations IP and subsequent ROA operations: unregistered, registered, allocated, assigned and binded.

In the case of prefix ownership, figure 4.9 shows how the different operations cause transitions from the starting state of unregistered to the final state of bound. As the prefix transitions from unregistered to bound, the final owner will be an LIR/ISP, and the leasee will be an end-user who has an AS and wants to use the prefix. Once that has been achieved, a ROA specifying the binding between a prefix and an ASN is added to the storage of BGPcoin. In the case of AS ownership, one thing that figure 4.8 implies is that only ISP's are the eventual owners of an ASN, while the authors state that a RIR can delegate the AS to other entities higher up in the hierarchy, like NIRs. Also, once the resource has expired, the owner could pay and request the RIR to send an ASN update transaction to the contract to keep up the period of validity granted to them by the RIR.

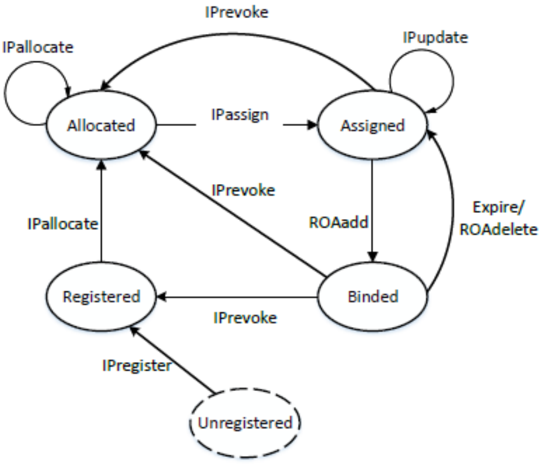


Figure 4.9: the state diagram of IP address allocation. Source: Xing et al., 2018

The big difference between prefix revocation and ASN revocation is that ASN revocation happens over time if no requests to extend ownership of an ASN are made, while owners of IP prefixes should consent to their resource being taken away from them. This issue is also an important factor in the RPKI and why basic implementations of it are not very viable, which will be discussed soon. As such, the authors propose an addition to how the prefix is modelled by adding a consent vector variable at the end of it, where the owner and delegating entities of a prefix can specify that they consent to their resource being taken away. A prefix can only be revoked by an entity if all entities lower in the hierarchy consent to the prefix being revoked.

As BGPcoin records resource assignments in the form of transactions in the Ethereum blockchain, the operations that are performed in the system cost a certain amount of ether or gas, where one gas is $1.8 * 10^{-8}$ ether. This can be a serious problem because ether might see price hikes in the future. The authors do include costs of BGPcoin trading operations in their paper in both gas and the equivalent amount of USD and show that the most expensive operation, that being the IP register operation costing 155,448 gas, costs less than 0.5 USD (0.449 USD to be exact). Still, these are based on September 2018 prices of ether, when one ether was almost 300 USD. As of December 29, 2019, ether has become a lot cheaper, as it is currently around 135 USD [18]. While this can be considered beneficial to the practicality of BGPcoin as operations cost less to execute, the reality is that cryptocurrencies such as Ethereum are volatile in price. For example, Bitcoin had many price peaks and price falls, with price peaks happening because more and more online retailers accepting Bitcoin and price falls happening because of online Bitcoin exchange websites going under [7]. There is no telling right now if Ethereum will keep decreasing in price, which would benefit the viability of BGPcoin. Still, considering that the peak price of the cryptocurrency was over 1300 USD for one ether, it isn't unlikely that the price could rise again, making the solution less viable.

4.8. The RPKI

The Resource Public Key Infrastructure (RPKI) is a solution that was first proposed by Bush and Randy in 2009 [61], and then standardized in 2012 by the IETF in RFC 6480 [102]. The proposed solution is essentially a modified global PKI, and the main purpose of it is to provide a trusted mapping from prefix sets to ASes [65]. As such, the solution itself is capable of protecting against only prefix hijacking. The main difference with a standard PKI is that certificates are not meant to be used to attest identities but rather attest ownership of a certain resource. This means that the RPKI provides authorization but not authentication. Because there is no authentication, costs and liabilities that are inherent with issuers are avoided, and more entities can take on the role of the certification authority (CA).

The RPKI has two different kinds of certificates: the CA certificates and the end-entity (EE) certificates. CA certificates in the RPKI are used to attest IP address space and AS number holdings. They are also required for an entity for that entity to be able to issue ROAs, and as such, will often be associated with the IANA, with RIRs, NIRs or ISPs (in the order of the standard hierarchy), as these institutions delegate more and more specific prefixes downwards. EE certificates, on the other hand, are used to validate Route Origination Attestations or ROAs. ROAs are signed objects that provide proof that an AS has the prefix space that it claims to own. The structure of a ROA is defined in RFC 6482 [103], and contains the following elements:

- A version number. This number is always 0, does not serve any purpose as of right now, and is most likely included to future proof the design.
- The AS ID, which contains the AS number that owns the claimed prefixes.
- The set of IP addresses that the AS in question claims to own. This is a sequence of IP prefixes as well as an optional integer value called maxLength, which specifies the maximum length that the AS can use in broadcasting the prefix. Because of this, it can have any value between the actual length of the prefix and the maximum possible length of the IP address (for IPv4, this is 32). If the maximum length is not specified, the AS can only broadcast prefixes with their original length.

When ASes receive new route information, the only thing that they would need to check for is whether or not the AS at the beginning of the route is authorized in the RPKI to originate that prefix. A schematic overview of a model of RPKI can be seen in figure 4.10. Here, ARIN is a RIR and has a CA certificate, which has been called a resource certificate or RC in this image. It then uses that certificate to suballocate the prefix to Sprint, which suballocates part of its prefix to Continental Broadband, which then signs ROAs for its subordinate end-entities.

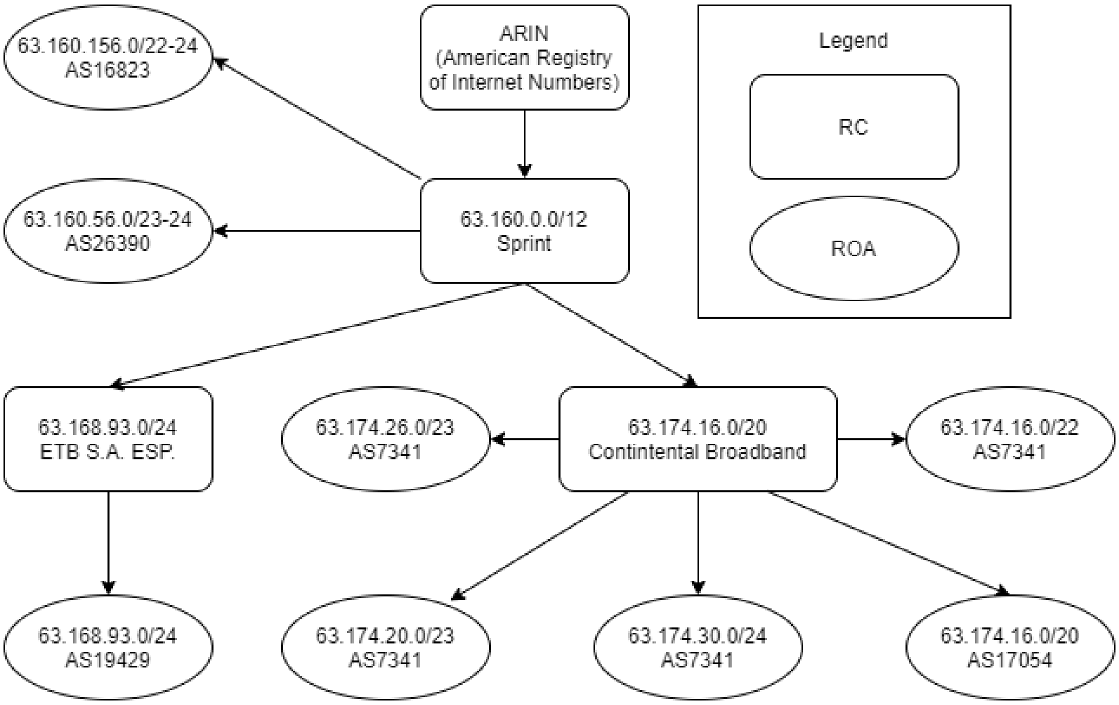


Figure 4.10: a part of a model RPKI. ARIN delegates its prefix to other parties, which then either act as further delegators or as end-entities. Source: Cooper et al., 2013

4.8.1. Why is the RPKI so popular?

The RPKI is a popular BGP security solution, probably because it is considered to be necessary for a lot of other solutions [65], and it is the only solution that has seen some adoption. For example, in June 2019, the Dutch Forum of Standardization has had a recommendation from the Dutch government to either apply or explain the RPKI [39]. This means that the RPKI should be applied in the Netherlands, or there should be a clear reason why it is not applied. Also, Liu et al. stated in 2016 that the five RIRs have finished the deployment of the RPKI, and several countries including but not limited to Japan, Ecuador, and Bangladesh, have deployed the RPKI [107]. At the same time, the RPKI is flawed and inherits many of its flaws from the flaws of normal PKIs. Critics of the RPKI have exposed many of its flaws, especially Liu et al. [107], and these risks include but are not limited to:

- More than one trust anchor accidentally or maliciously issuing certificates for the same IP prefix(es) [65]. There is no standardized set of trust anchors, but the solution allows for multiple trust anchors to exist, and potential candidates include the IANA and the five RIRs.

- The maxLength attribute can be harmful to the RPKI, as misconfiguration and allowing for longer prefix lengths than the RPKI owns can easily allow for subprefix hijacks, because this would allow ASes to announce more specific versions of an already-announced prefix. There are legitimate reasons why network operators might want to have a maxLength that is longer than the length of the prefix, such as allowing the AS to broadcast more specific prefixes in the future. This exacerbates the problem of security flaws related to the maxLength attribute. As such, Gilad et al. recommend removing it entirely and allowing network operators only to use minimal ROAs [73].

- Misconfiguration of certificate authorities being a major threat to the availability of certain autonomous systems because of the hierarchical structure of the solution.

- The potential for certificate authorities to unilaterally revoke their certificates, severing the availability of the autonomous system that originally held the certificate. This allows for organizations higher up in the hierarchy to practically hold the internet connection of an autonomous system that is under them hostage for whatever purpose they desire. To combat this, Heilman et al. have proposed introducing a .dead object to allow end-users to consent to their prefix being revoked [80]. Intentional unilateral revocation could also be discouraged by implementing laws that forbid organizations from doing so, especially considering access to the Internet is considered a human right [97].

- Downloading RPKI data would be by using rsync, and the rsync protocol has issues as well, those being that the protocol is not standardized by the IETF, that it is not efficient enough to meet current demands as many clients will connect to one server concurrently [40], and having to resync because data was added during synchronization will increase overhead in synchronization.
- Incomplete deployment of RPKI can cause some perfectly valid routes to be flagged as invalid, and going from no deployment of the RPKI to complete deployment of RPKI is practically impossible.

As such, one cannot help but wonder why the RPKI was so popular in the first place, to the point of being the only security solution designed to combat security flaws in BGP that is at least partially deployed.

On the surface, one can see that RPKI is a far simpler solution to ongoing problems with BGP than any other solution proposed up until that point. It consists of one PKI, whereas for example, S-BGP requires two PKI's to be deployed, each taking a different responsibility, and also requires a lot of cryptography to secure the routes. S-BGP does not only introduce centralization but also requires a lot of cryptographic overhead. While the RPKI does not perform any route validation whatsoever, it does simplify origin validation significantly.

One other reason could be that the RPKI was already adopted by the RIR's, which is in stark contrast with other solutions for BGP security, and initial adoption at least gives a reason for other autonomous systems also to adopt the RPKI. The problem of bootstrapping adoption has been mitigated with this solution, whereas that is a remaining problem with other solutions. Another thing that helps the RPKI is the fact that the IETF has already standardized it, which is in contrast to the other solutions that had predated it, which are not standardized.

Another possible reason, which is related to the first reason and might seem at least a little contradictory at first, is that the RPKI was never intended to solve every issue all at once completely. Rather, it is more of a base for other solutions (for example, BGPsec) to use once it has been deployed. This reason could be somewhat contradictory as one would expect a solution that would solve every problem with the BGP protocol to be the more popular one. However, combined with the fact that the RPKI is a much simpler solution to part of the problem, it can serve as an excellent base for further development regarding the future of BGP.

4.9. BGPsec

BGPsec is a security solution that was first proposed in 2011 when the RPKI was close to being standardized to complement it [85]. It was standardized in 2017 by the IETF in RFC 8205 [104]. Because the RPKI only provides origin authentication, BGP secures the routes and, as such, provides route authentication. It does so by allowing ASes to sign their BGP updates before sending them. When a prefix is first announced, the signature only covers the announced prefix, the sender AS number, the receiver AS number, and a hash of the public key from the key pair used by the router to sign updates. Then, a tuple of the public key hash, as well as the generated signature, is added to the UPDATE message. When a route is forwarded, the generated signature then covers the previous signature, both the sending and receiving AS numbers and the hash of the public key. Figure 4.11 shows how the solution operates. To validate an update, a receiver can use the interlocking chain of signatures that the sequence of ASes in the route has been traversed in order. It also consults consult the RPKI to verify that an AS owns the prefix that it claims to own.

Implementation of the BGPsec solution would be done by introducing a new attribute to BGP. This attribute would be the BGPsec-PATH attribute and is an optional non-transitive BGP path attribute. This attribute would consist of two fields: the Secure-Path field and a field which would be a sequence of one or two Signature-Blocks, although most commonly it will only contain one of these. A second one would be used when a new algorithm suite would be used.

The Secure-Path field would represent a full path, where the segments would specify the AS numbers of the segments if the AS number is to be repeated and how often, and whether or not the next (receiving) AS is part of an AS confederation. AS confederations are standardized in RFC 5065 as "A collection of autonomous systems represented and advertised as a single AS number to BGP speakers that are not members of the local

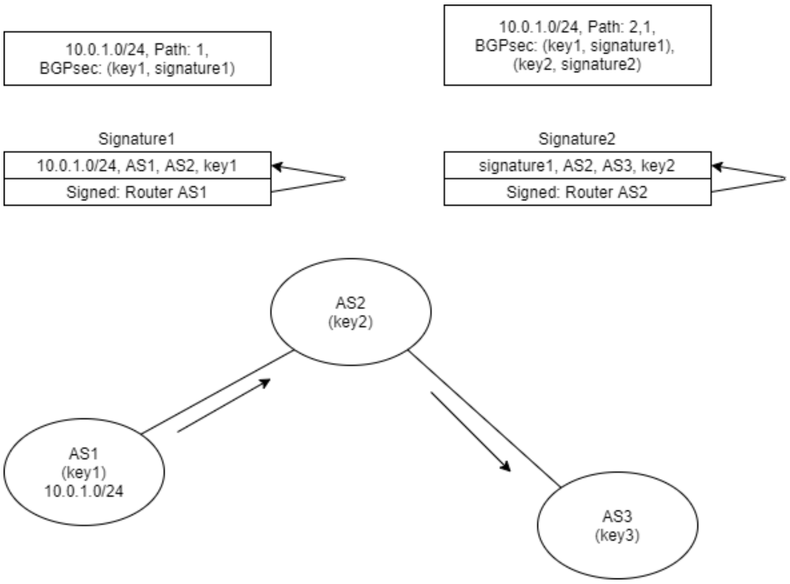


Figure 4.11: BGP message propagation using BGPsec. The signing of updates is similar to what S-BGP uses for securing route propagation. Source: Huston et al., 2011.

BGP confederation" [139]. The Signature-Block field specifies the algorithm suite used, as well as a series of signature segments, which would form the signatures that need to be passed on.

BGPsec can do origin authentication via the RPKI and path authentication in a decentralized fashion with the use of signing messages. However, the main hindrance is that it would require an update to BGP as a whole to be able to use it, as the required data would be transferred between ASes in UPDATE messages. Also, research has found that even if both the RPKI and BGPsec were to be fully implemented, BGP still would not be secure, and attacks against BGP, including those involving path altering, would still be possible. Colluding ASes that create fake paths between them can circumvent full deployment of BGPsec (and S-BGP for that matter). Only the construction of neighbour AS graphs can help detect and prevent these kinds of attacks [106]. Similar to S-BGP, the fact that attacks against a non-deployed solution are already known is detrimental to the benefit of implementing the solution. Furthermore, Li et al. proposed a series of necessary properties for securing BGP and showed that BGP armed with BGPsec could not achieve any of these properties [105]. The property "availability of routes" is a property that will be the most affected according to the authors, as BGP is known to converge slowly [98], and adding computational overhead will cause the convergence time to increase.

4.10. KC-X

Keychain-X (KC-X) is not so much a BGP security solution, but more a security scheme that makes use of keychains (which is where the KC part of the name comes from), which protects the ASPATH [150]. It is a generic signature framework that can use any digital signature algorithm. The X in KC-X is then substituted for the algorithm that is used for signing. If, for example, RSA was used, then the resulting solution would be called KC-RSA. The scheme has also been referred to as KC-BGP [154].

The design of the scheme is as follows: every BGP speaker R_i generates a temporary key pair (t_i^+ / t_i^-), where t_i^+ is the public key and t_i^- the private key. When an UPDATE message is sent from R_i to the next BGP speaker R_{i+1}, R_i first sends t_i^- to R_{i+1} in plaintext. Then, the UPDATE message is combined with the public key of that BGP speaker, and then they are signed by the private key t_{i-1}^- of the previous BGP speaker. This signing can be done using any signature scheme, such as RSA, as mentioned earlier. This passing of temporary private keys forms an authorization: the BGP speakers that pass along the UPDATE messages are granted the private key by the previous AS, instead of signing it with their private key. In the case of a route origination, where there is no previous AS to get a private key from, the BGP speaker signs the message with its private key. Figure 4.12 displays the authorization chain.

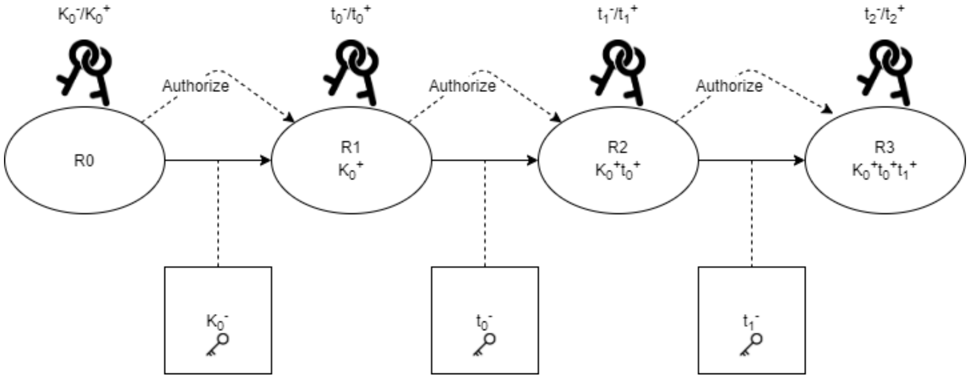


Figure 4.12: the authorization chain. Source: Yin et al., 2007.

When KC-X would be adopted by all the ASes that make up the Internet, path altering would be infeasible. However, because it is infeasible to make all the BGP speakers in the world adopt the same solution at the same time, the solution is incrementally deployable, and in the case of incremental deployment, paths would be secure up until the last AS in the path that has a KC-X adopting BGP speaker as the part of the path before then has been signed for by the last BGP speaker that received the path and passed it along.

Similar to how BGPsec is to be integrated into BGP, the scheme would be integrated into BGP by passing on the authentication information in BGP UPDATE messages as an optional transitive path attribute.

The authors also compare two implementations of KC-X, one with RSA as the signature algorithm (called KC-RSA) and the other with the Merkle Hash tree as the signature algorithm (called KC-MT), to each other as well as to other BGP security solutions. The conclusion when comparing the two different implementations with each other is that KC-MT is computationally faster due to it relying on lightweight hash functions but has larger signatures due to constructing hash trees. At the same time, KC-RSA creates smaller signatures but is computationally slower because of relying on asymmetric cryptography. KC-MT is also compared to SPV, and the authors argue that the former is simpler than the latter because of three things:

1. The epoch mechanism in SPV can make hashing operations slower over time because more values are added to the signature and as such more hashes are needed over time
2. The multi-epoch public key in SPV has to be distributed beforehand to all the speakers that need to verify an ASPATH originating from the issuer, and KC-MT does not require any extra measures for distribution
3. SPV is vulnerable to the multi-path truncation attack (although the authors proposing SPV have acknowledged this and proposed a solution in the same paper [84]). This attack is an attack where a private key obtained from a shorter ASPATH can be used to alter a longer ASPATH if both paths have the same origin.

When comparing the relative speed of individual operations by measuring them, the authors discovered that signing in KC-RSA takes over 70 times as long to complete compared to KC-MT, almost twice as long as S-BGP and around ten times as long as SPV. Also, the time it takes for S-BGP to sign a message can be reduced significantly by allowing for the precomputation of signatures.

As for verifying, S-BGP took the longest, with KC-RSA being about 12 times as fast, KC-MT being about 45 times as fast, and SPV about 23 times as fast. When comparing the introduced delay under normal workloads, KC-MT introduced the least delay, followed by SPV, then KC-RSA, then S-BGP. From these results, it would seem that KC-MT is better than KC-RSA and some other proposed security solutions. However, recall that the memory footprint of using KC-MT is higher than that of KC-RSA, and routers have limited memory. The amount of memory necessary for KC-MT is about six times as much as is used for KC-RSA. Because of this, the authors suggest a hybrid approach for practical deployment.

KC-X is a security scheme that attempts to secure BGP without requiring centralization, all BGP speakers adopting the solution at once or excessive processing power. However, the benefits to only one AS in the entire Internet adopting the solution are next to none as data needs to be exchanged between ASes, falling into the same pitfall that the incremental deployability of soBGP fell into. Also, because it requires an update to BGP for all the security information to be passed on effectively makes it even less feasible.

5 Threat landscape analysis

One aspect of BGP security is the different threats to the correct functioning of the protocol, and it is important to take a closer look at the threats to BGP security. Or more specifically, how the threat landscape evolved, to see whether or not the considered threats changed. The goal of this chapter is to answer the first subquestion: how have threats to BGP changed over time, and have proposed security solutions adapted to possible changes?

5.1. Approach

To get a picture of how the threat landscape has evolved, information on the threats to BGP security during different periods in time needs to be gathered. I have not been able to find any previous academic work that has done this analysis on BGP attacks specifically. However, there is work in other fields that presents a methodology that I have adopted for this thesis. For example, Grabosky wrote an overview of the evolution of cybercrime in "The Evolution of Cybercrime, 2006-2016" [78]. In this paper, he cited a number of papers that have been published over the years. These papers mentioned developments in cybercrime. The author used the papers to create an overview of how cybercrime has changed over the years. Mansfield wrote an article on the evolution of DDoS attacks as well [110], citing several DDoS reports from Verizon, Verisign, and Akamai. As such, I decided to follow a similar approach: gathering papers on BGP attacks and problems with BGP, and recording the year the paper was published in and the attacks that are mentioned. This literature analysis serves as a snapshot-based overview of when different threats were considered at different points in time. To augment this point-in-time data source, I decided to opt for an additional data source that showed interest in different kinds of BGP-related problems over a continuous timescale. Google Trends proved to be useful for this, as not only does it show search interest over time, but it has also been used in previous research to discover trends in other fields. Rech used it to discover trends in software engineering, for example, [124]. The second data source that will be used is Google Trends data on several BGP security related search terms and then comparing it to Google Search data to see if there is a pattern between the two. The former gives insight into how often the term has been entered into Google Search. The latter accurately depicts how much people have been writing about the subject over time. Both give a picture of public interest in (and to a certain degree, knowledge of) the topic.

5.2. Literature analysis

There is a body of academic literature discussing the threats to BGP, as has already been seen with the various papers proposing solutions to secure BGP. This section will use that work from various points in time to see if there are any trends. More specifically, three different categories of literature are defined: work from before 2000, work from between 2000 and 2009 and work from 2010 onward. These three categories represent three distinct periods in BGP security. The first category deals with considered threats against BGP before any security solution was proposed. The second category deals with considered threats after the initial security solutions were proposed. The final category deals with considered threats during and after the first security solution was standardized by the IETF, namely the RPKI. This analysis is not going to be another literature overview, such as Chapter 4. It is going to list papers that mention threats to the security of BGP, but it only gives an overview of the mentioned threats instead of an overview of the publication.

5.2.1. Pre-2000s work on the topic of threats to BGP

The earliest papers mentioning threats to the security of BGP are from before the first security solutions to counter these threats were developed, which started with S-BGP in 2000. Smith and Garcia-Luna-Aceves published a paper in 1998 discussing several different classes of attacks [133]. The classes are intruders, threats to routing information, and threats to data traffic. Each of these classes is further subdivided into several different subclasses.

In the intruder category, the authors discuss subverted, unauthorized, and masquerading BGP speakers, as well as subverted links. A subverted BGP speaker occurs when an authorized BGP speaker is caused to violate BGP, or when a speaker inappropriately claims authority for network resources. An unauthorized BGP speaker exists when a node that was not allowed to be a BGP speaker somehow circumvents this and establishes BGP links. A masquerading BGP speaker occurs when a node forges the identity of an authorized BGP speaker. A subverted link happens when either access is gained to the physical medium or a lower level protocol, which allows control of the channel.

In the category of threats to routing information, the authors mention the possibility of an intruder being able to "fabricate, modify, replay or delete" routing traffic. These can be condensed into (sub)prefix hijacking, path altering and truncating, and deliberate route flapping.

Finally, in the category about threats to data traffic, the authors discuss the potential for an intruder to drop or snoop data traffic. These threats go beyond the scope of BGP security though, and countermeasures should be implemented in transport layer data protocols, as this assumes that data is sent without any encryption whatsoever.

Another paper written by the same authors from 1996 discusses many of the same threats [134], which leads me to believe that the earlier mentioned paper from 1998 reiterated these threats. The categories of attacks in this paper are intruders, deception or disruption of routing messages, and disclosure of routing messages. In the category of intruders, aside from the threats mentioned in the paper from 1998, they also include subverted links, which can happen by compromising lower-level protocols or gaining access to the physical medium. However, these threats are not specific to BGP and, as such, fall out of the scope of this thesis. That might also be the reason why the paper from 1998 does not include them. The category of deception/disruption of routing messages includes masquerading BGP speakers taking the role of authorized BGP speakers in computing routes. Disclosure of routing messages is also similar to the paper from 1998. A paper written by Atkinson in 1997 also briefly mentions a threat against BGP [55]. Atkinson mentions that, because BGP runs over TCP, it is vulnerable to all the attacks that TCP is also vulnerable to. Aside from these works, there is not much work from this time with regards to the field of threats to BGP that provides any new threats. For example, Wang et al. published a paper in 1997, but they reiterate in their work the threats that Smith and Garcia-Luna-Aceves have already mentioned [144].

5.2.2. Work on BGP threats from the 2000s

From 2000 onwards, there is more work on the topic of potential threats to BGP security. This increase in the number of academic works probably has to do with the fact that BGP security was being taken more seriously around this time because the first proposals to secure BGP are from this time. These first proposals addressed the earlier discovered weaknesses in the existing protocol. One paper from Butler et al. from 2005 discusses several possible threats to BGP [62]. This paper also discusses several different categories of attacks, those being attacks between peers, larger-scale attacks, denial of service, and misconfiguration of BGP speakers.

The category of attacks between peers deals with attacks that happen between BGP-speaking routers that are peered to each other and have no large-scale impact. Attacks belonging to this category are:

- Attacks against confidentiality, where third parties can read messages that are sent between peers.
- Attacks against message integrity, which is a step up from the previous attack, and involves a third party not only receiving (a copy of) the messages that have been sent but also altering them, in whole or in part.
- Session termination, which is more of a consequence of the previous attack, as the previous attack can not only be used to inject false routing data but also to terminate a BGP session between routers.

The next category is the larger scale category. This category of attacks involves attacks that do not happen between two peers, but rather target specific autonomous systems (or their resources). The attacks that fall under this category are fraudulent origin attacks, which is another name for (sub)prefix hijacking, and subversion of path information: this attack involves tampering with the path attributes of an update message sent by BGP speakers. This attack mentions both MED alteration and path alteration. The former can influence which paths are chosen over which and falls into the category of protocol manipulation, whereas the latter changes the path altogether.

The DoS category deals with DoS attacks. DoS attacks can happen by blackholing a route, or just redirecting traffic continuously until the TTL of the packet has been exceeded. Route flapping is also included in this category. Of these two attacks, only route flapping is specific to BGP, as DoS attacks are also often performed without targeting BGP or Internet number resources.

The final of the four categories is the category dealing with the misconfiguration of BGP routers. This category deals with "attacks" (if you can even call them that) that originate from simple misconfigurations in BGP speaking routers. This category is important as the authors acknowledge the fact that some of the "attacks" can happen due to simple misconfigurations, without any malicious intent.

Butler et al. already provided an overview of the considered attacks during this time. It is also one of the earliest written papers from this specific period, and other papers from this period also mention many of the same threats to BGP security. For example, Murphy published in 2007 a report on threats to BGP security [114], which mentions three categories of vulnerabilities: transport of BGP over TCP, route origination, and AS-PATH construction. The latter two of these fall under (sub)prefix hijacking and path altering, while the first one is not specific to BGP and relates more to vulnerabilities in TCP.

Around the same time as Butler et al. published their work, Nordström et al. also published a report on BGP attacks [118], intended to raise awareness for attacks against BGP. The authors presented seven different types of BGP attacks, which are as follows:

1. Prefix hijacking.
2. Prefix de-aggregation.
3. Contradictory UPDATE advertisements.
4. Update modifications.
5. Link flapping.
6. Link instability.
7. Congestion-induced failures.

These attacks roughly fall under the categories of (sub)prefix hijacking, path altering, protocol manipulation, and DoS attacks, if the six categories of BGP security problems in the introduction are used to categorize these attacks. The authors have focused only on the modification of the paths in an UPDATE message. Nicholes et al. surveyed security techniques to secure BGP and, as such, discussed possible attacks [117], and the categories are generally the same as other work from this time, except for one. The categories discussed in this paper are: TCP related attacks, modification of the AS_PATH in an UPDATE message, deliberate misconfiguration of a router due to hacking the router and causing false prefix announcements and wrong route exports, and physically attacking the router. The last one is out of scope for this thesis as it is not BGP related. The other attacks roughly fall in the categories of prefix hijacking, path altering, protocol manipulation, and attacks on TCP.

In the extended version of the paper that proposed psBGP, Van Oorschot et al. mentioned in 2007 attacks targeting BGP UPDATE messages [119], specifically "modification, insertion, deletion, exposure, and replaying of messages." These fall into the categories of path altering attacks and protocol manipulation attacks. The categories mentioned in the publication are expanded upon in another article written by the same authors in the same year [143]. In this article, the authors also mention that the network layer reachability information (NLRI) can be falsified in the UPDATE message. This manipulation allows for attributes that were not intended to be applied to the prefixes stated in the NLRI to be applied to them, which can cause prefix hijacking. This article also briefly presents the three security solutions that have been proposed by that point in time, those being S-BGP, soBGP, and psBGP. Finally, Hepner et al. presented man-in-the-middle attacks in BGP being possible using prefix de-aggregation [81]. Prefix de-aggregation falls under the category of (sub)prefix hijacks.

As can be seen, there is more academic work in this period compared to the previous period, probably because BGP security started getting taken seriously, as mentioned before. One threat that is notably absent from considerations in this period while it was present in the previous one is the potential for malicious non-authorized BGP speakers to gain the ability to speak BGP on behalf of different BGP speakers, thereby subverting another BGP speaker. Also, DoS attacks are considered in this period as a threat to BGP security. This is in contrast to the period before this. Protocol manipulation is also considered at this time.

5.2.3. Work on BGP threats from 2010 onwards

There is also more academic work on the topic of threats against the security of BGP published from 2010 onwards. This is after the time when several high-profile attacks against BGP security were already launched, such as the infamous Pakistan Telecom attack which as discussed earlier happened in 2008.

As an early example of work in this period, Huston et al. published a paper that contained a full threat model back in 2011 [85]. The threat model is as follows:

- The BGP session between routers can be insecure and prone to attacks.
- Identity of the other party has to be identified as it is not identified yet by the current protocol.
- Authenticity and completeness of routing information are not guaranteed by BGP as it is.
- Forwarded routing information does not have to represent the current routing system and can easily be either false or simply outdated.

This model leads to the authors considering four different attacks: the ability to eavesdrop by redirecting traffic through a malicious autonomous system, denying service by flooding the data links, the potential for autonomous systems to masquerade as different autonomous systems, and the ability to steal addresses and obscure their identity. This is comparable to the considered threats from the previous time period. Also, one could argue that the threat of BGP speakers masquerading as other BGP speakers makes a return here, but the authors discuss the inability to verify identity more with regards to the potential of prefix hijacking instead of the ability of BGP speakers to masquerade.

Another threat model for BGP security has been proposed by Kent et al. in 2014 and is standardized by RFC 7132 [91]. This threat model does not only discuss general threats against BGP but also threats against the RPKI. This is most likely because the RPKI has been standardized and is therefore assumed to be implemented in the future. It also assumes BGPsec implementation as it assumes that PATHSEC is used in BGP UPDATE messages, probably because that solution has also been standardized, as we have seen before. The threats are described in the section of the paper regarding attacks on a BGP router, and they are:

- Inserting one or more ASes into an UPDATE message, such as in the AS_PATH or the NLRI.
- Falsely advertising a prefix origin (this is addressed by using the RPKI).
- An MITM attack being possible between two ASes if the cryptographic keys encrypting the messages are stolen.
- Downgrading usage of the secure path by opting for the less secure AS_PATH instead of PATHSEC.
- Advertising stale paths that have expired (PATHSEC includes expiration dates for path announcements), which can be considered replay attacks.
- Advertising paths with PATHSEC of which the expiration date is too short for the path to become usable. This is only an attack if there is a minimum expiration time mandated by the AS.

- Failing, whether or not on purpose, to propagate the withdrawal of a path in PATHSEC.
- Attacks on a repository publication point, which contains RPKI data.
- Attacks on a certificate authority in the RPKI.

The first two attacks are not specific to the standardized BGP security solutions, and they fall into the same categories as the ones that have been seen in other work: path alteration, protocol manipulation, and prefix hijacking. However, the last one can be solved with RPKI. The other attacks are specific to using RPKI or BGPsec. They show that these solutions are not flawless, as evidenced by the RFC discussing residual threats, which primarily focuses on threats that are specific to the deployment of the RPKI and BGPsec. RFC 7454 proposed by Durand et al. uses RFC 7132 to come up with some best practices for BGP security [67].

There is more work from this period on the topic of BGP security. For example, Schuchard et al. published a possible attack in 2010 [130], which is called the CXPST attack, and in their background, they mentioned route flapping and DoS attacks against routers. Aside from that, the proposed CXPST attack involves a botnet that selectively disrupts BGP sessions to generate a large number of BGP updates in a short amount of time. Another paper from the same year as the one of Huston et al. is the report published by Zeb et al. [151]. This paper discusses several earlier-discussed security flaws in BGP, such as (sub)prefix hijacking being possible, route truncations and eavesdropping on BGP traffic, and attacks that involve manipulation of path attributes. These attacks involve manipulation of the various attributes in BGP messages, with emphasis on the MED attribute (which has been mentioned in the work of the 2000s) as well as the NEXT-HOP attribute. Farooq et al. wrote a report back in 2011 [69], which mentioned the following threats against BGP:

- Incorrect routing updates, whether they are because the route has been falsified or because the route leads to an AS which does not own the prefix that it claims to own.
- Prefix de-aggregation.
- Manipulation of path attributes such as MED and NEXT_HOP.
- Blackholing traffic.
- Eavesdropping traffic.
- Congesting traffic and introducing traffic loops.

The first three of these threats fall under the earlier seen categories of (sub)prefix hijacking, path altering, and protocol manipulation. The others can be seen as consequences of these kinds of attacks being executed successfully.

Mitseva et al. also mentioned this type of attack in a 2018 publication [113]. The authors consider several kinds of attacks involving manipulation of protocol values. One example is the MED modification attack (tampering with the multi-exit discriminator to affect other ASes decisions). Another example is the attack that exploits the RFD/MRAI timer (artificially withdrawing and re-announcing a route to cause ASes that use the RFD timer to consider the route unstable and not adopt it in turn). This attack is an attack that has not been mentioned in academic work from earlier eras as far as I have been able to find. Aside from this, Mitseva et al. also mention a threat based on maliciously using correct data. Old examples are the DoS attack and deliberate link flapping, which could happen as a result of the DoS attack, but a new category is route leak attacks. These are attacks where an AS deliberately broadcasts routes to other ASes that are not intended to receive these routes because of business agreements. These route leaks could cause ASes to become transit ASes, as was the case with Google in 2017, which led to a lot of users experiencing delay [21].

In short, most of the threats considered in this period are the same as the threats considered in the period prior. The main difference is that attacks involving manipulation of variables in BGP messages are considered more frequently now where they were barely considered before. There is also a trend of threat models being developed and standardized. This standardization was not present in the periods before this one.

5.3. Timeline of considered threats

Given the different periods and the different threats considered in every period, a timeline of considered threats can now be made. This was already partially done by the short conclusions at the end of each subsection of the previous section, and a more detailed timeline can be seen in table 5.1.

	1996	1997	2005	2007	2009	2011	2014	2018
Prefix hijacking	*		*		*	*	*	*
Normal prefix hijacking	*		*			*	*	*
Subprefix hijacking	*		*		*	*	*	*
Path altering	*			*		*	*	*
Speaker impersonation	*							
Protocol manipulation			*	*		*		*
Misconfiguration of router			*	*				
Manipulation of variables				*		*		*
Attacks to TCP		*						
DDoS	*		*			*		*
Route flapping	*		*					
Congestion			*			*		*
Blackholing traffic						*		*

Table 5.1: the threats to BGP security that have been discussed in the various papers, when put on a timeline with the different threats per paper in the year that the paper was published in.

This timeline, like the small summaries of each of the periods of BGP security, shows that there is a trend in the threats that are considered. Pre-2000 work mentions non-BGP speakers gaining authority to speak BGP while not talking about protocol manipulation, while post-2000 work is the opposite. One question one can ask is: is this shift reflected in the proposed security solutions? Do different proposals cater to the threats of the time? Logically speaking, they should, but to see if they do, this timeline needs to be compared to the timeline of BGP security solutions.

5.3.1. Comparing the timeline of threats to the timeline of BGP security solutions

To compare the timeline of threats to the timeline of BGP security solutions, a timeline of security solutions is required. This timeline is presented in table 5.2. Recall from Chapter 4 that S-BGP, soBGP, and psBGP all in one way or another provided a way for verifying that BGP speakers belonged to the autonomous system that they claim to belong to. This is reflected in earlier works on the topic of threats to BGP, where BGP speaker authorization/authentication has been considered as a possible threat. Also, after the proposals of the early 2000s, such as S-BGP and psBGP, the proposals were more focused on preventing one kind of attack against BGP. As an example: HC-BGP and the RPKI only focus on origin authentication and BGPsec focuses primarily on route authentication and performs origin authentication using the RPKI. It is however unlikely whether or not this has to do with changes in the threats that are considered or simply with the fact that prefix hijacks have been more prevalent than path altering. Finally, IRV is the only security solution presented which deals with attackers influencing the protocol and executing a protocol manipulation attack, and that solution was proposed right before the first papers discussing protocol manipulation as a problem started being published.

5.4. Conclusions from the literature analysis

To conclude the literature analysis, we can see that not only has the landscape of considered threats has changed, but threat models have also been developed and even standardized by an RFC. Earlier threats fo-

	Year proposed	Origin auth.	Path auth.	BGP speaker auth.	Attribute verification
S-BGP	2000	Yes	Yes	Yes	No
soBGP	2003	Yes	No	Yes	No
IRV	2003	Yes	Yes	No	Yes
psBGP	2004	Yes	Yes	Yes	No
SPV	2004	Yes	Yes	No	No
KC-X	2007	Yes	Yes	No	No
HC-BGP	2009	Yes	No	No	No
RPKI	2009 (proposed) 2012 (standardized)	Yes	No	No	No
BGPsec	2011 (proposed) 2017 (standardized)	Yes	Yes	No	No
BGPcoin	2018	Yes	No	No	No

Table 5.2: the timeline of when different BGP security solutions were first proposed. Note that in the case of the RPKI and BGPsec, which have been standardized by the IETF, both the year of ther porposal and of their standardization has been used.

cused more on impersonation, which stems from the fact that routers (that may or may not be authorized to speak BGP on behalf of the AS that they belong to) that speak BGP can claim to belong to an AS that they do not belong to, while later threats also focus on abusing the intricacies of BGP as well as DoS attacks. These differences in threats that are deemed important enough are partially reflected in the security solutions that the research community has proposed over the years, as has been mentioned before with only the earliest few BGP security solutions having measures in place to check whether or not a BGP speaker is allowed to speak on behalf of a given AS.

5.5. Google Search and Google Trends data

Because the various works serve only as snapshots of certain times and what kind of threats have been considered at the time, the insight that they give into how the threat landscape has evolved over the years is useful but it does not give a full picture. There can also be a lot of time between academic works on the subject. For instance, when comparing work from after the 2000s, there were a few papers from 2010 and 2011, then two RFC's in 2014 and 2015, and then another paper from 2018. As such, data that gives a more continuous insight into how the threat landscape has evolved is useful as an additional source. Also, while the papers give a nice overview of the threats that have been considered over time by the scientific community, it is not a good indication of public interest regarding BGP security, or how much the public has written about it. This is why the secondary data source of comparing Google Search hits per year and Google Trends data is used. Google Trends is a search trend feature that allows users to see how frequently a search term has been entered into Google over a certain period of time [22]. This data source allows us to compare how often a topic within the scope of BGP security has been searched for on the Internet over time (through using Google Trends data) versus how much people have been writing about the subject over time (through using the number of hits on Google Search during different years).

5.5.1. How to compare Google Search and Google Trends data

There are several problems when using this approach to compare how often a topic has been entered into a search machine and how much has been written about the same topic. The problems are as follows:

- Google Trends displays search statistics relative to the month when the topic was entered the most often in Google, and Google Search hits will result in absolute numbers.
- Google Trends displays trends on a month-to-month basis, and while it is possible to do the same with Google Search by specifying date ranges in search queries, doing so cannot be automated and thus makes the comparison process require significantly more manual labour.
- Google Trends only records as far back as 2004, whereas the topic of BGP security was discussed on the Internet before then as evidenced by literary works from before 2000 discussing threats to BGP.

As such, the data gathered needs to be transformed and trade-offs between accuracy and the amount of manual labour required need to be made. For one, to solve the second problem, Google Search hits will be recorded on a year-by-year basis. Secondly, to solve the first problem, the amount of Google Search hits per year will be divided by the highest amount of hits in a given year and then multiplied by 100 to allow for direct comparison with Google Trends data. Recording Google Search hits on a year-by-year basis also helps in this case, as the odds are that the variance is higher in the number of hits between years rather than months. If I opted for recording hits on a month-to-month basis, most months would probably have a similar amount of hits, and when transformed using the method described, it would look similar as well. The odds of most years having a similar amount of hits are lower because the timespan is greater. Finally, to solve the third and final problem, only hits from 2004 and after are recorded.

5.5.2. Terms entered

The terms entered into Google Trends are meant to represent a wide spectrum of topics in the field of BGP security. They also fall into four general categories: solutions (only the solutions that were standardized by the IETF were considered, as those are the most popular and the solutions that are at least considered feasible enough to be standardized), general terms related to BGP security, attack vectors, and consequences of attacks. They are as follows:

- BGP solutions:
    - **BGPsec**
    - **RPKI**
- Terms related to BGP security:
    - **BGP security**
    - BGP verification
    - BGP integrity
    - BGP confidentiality
    - BGP validation
    - BGP signature
    - BGP certificate
    - BGP route validation
    - BGP origin authentication
- Attack vectors:
    - **BGP attack**
    - IP hijack
    - BGP mismanagement
- Consequences of attacks:
    - **BGP blackhole**
    - **BGP hijacking**
    - BGP path change

All the search terms are entered in quotation marks for Google Search as well as Google Trends. This selection of terms on the topic of BGP security represents several aspects of BGP security, as exemplified by using the categories to subdivide the terms. Confidentiality and integrity are mentioned, and availability is represented through searching for BGP/IP hijacking. Methods to provide said security (origin/route validation) are also in the selection, as well as some of the more popular BGP security solutions such as RPKI.

5.5.3. Results

The results when comparing the Google Trends data with Google Search hits per year are shown in figures 5.1a to 5.1f. Unfortunately, very few terms had sufficient data to generate Google Trends data. Only the terms highlighted by **bold** text had Google Trends data, and these tend to be the most general terms of their category. This lack of data might have to do with the fact that BGP security is an often overlooked part of Internet security; people are most likely more concerned about possible weaknesses in encryption methods that are used daily than about BGP security.

5.6. Conclusions from the Google Search and Google Trends data

From the comparisons that can be made, several conclusions can be drawn. These are:

- In general, there seems to be a negative relationship between the number of searches on a topic versus the amount that is written regarding that topic. This probably has to do with more people knowing about the subject after the first few websites have covered it, which leads to less searching for it. The subject that seems to have the highest positive relationship is the RPKI. This, in turn, probably has to do with the solution being rather popular, as has been covered in Chapter 4.
- The is also a trend in the amount of Google Search results in that they increase over time. Of course, there are exceptions to this, such as the Google Search results for the term "BGP attack" sharply declining in 2017, but the general trend is still noticeable.
- Regarding BGPsec specifically, there seems to be interest in the topic as early as late 2008 to early 2009, as evidenced by the Google Trends spikes around that time, which is interesting as the solution was not even proposed back then as far as I can see. This might have to do with the RPKI being proposed back then and people thinking about BGPsec to augment the RPKI.
- There are some peaks in search interest when looking at the trends on BGP blackhole. An example is from the end of 2005 to the beginning of 2006 (which is the highest peak). Another example is at some time in 2008 (which is after a time of very little to no interest). These two peaks correlate with BGP outages: the former is after both the Google outage in May 2005 [20] and the Con-Edison accidental hijacking [12], and the latter is right after the infamous Pakistan Telecom incident [50]. It would be logical to assume that these hijacks cause these spikes in interest.
- In addition to the previous point: peaks in search interest right after an attack happened can be seen in more Google Trends results than just the results of BGP blackhole. We see similar peaks when looking at the results for BGP hijacking, where there are peaks after not only the Google incident of May 2005 but also after the TTNet incident of December 2004 [43]. When looking at the results for BGP security, the same peak can be seen right after the time of the Google outage.
- The trends for RPKI show an interesting peak around 2005. The earliest proposals for RPKI were from 2009, so it is not clear to me why the interest was that high before the solution was even proposed.

Unfortunately, these conclusions do not show any changes in the threat landscape. They mainly show that there is a correlation between BGP hijacks happening and search interest peaking right after them happening, but that is somewhat obvious, especially if the impact is large, as in the case of the Pakistan Telecom accident. If there were Google Trends results for terms such as "BGP route validation" and "BGP origin authentication," the results would probably say more about how the threat landscape has evolved. These are terms that have more to do with possible attacks against BGP, but unfortunately, this is not the case.

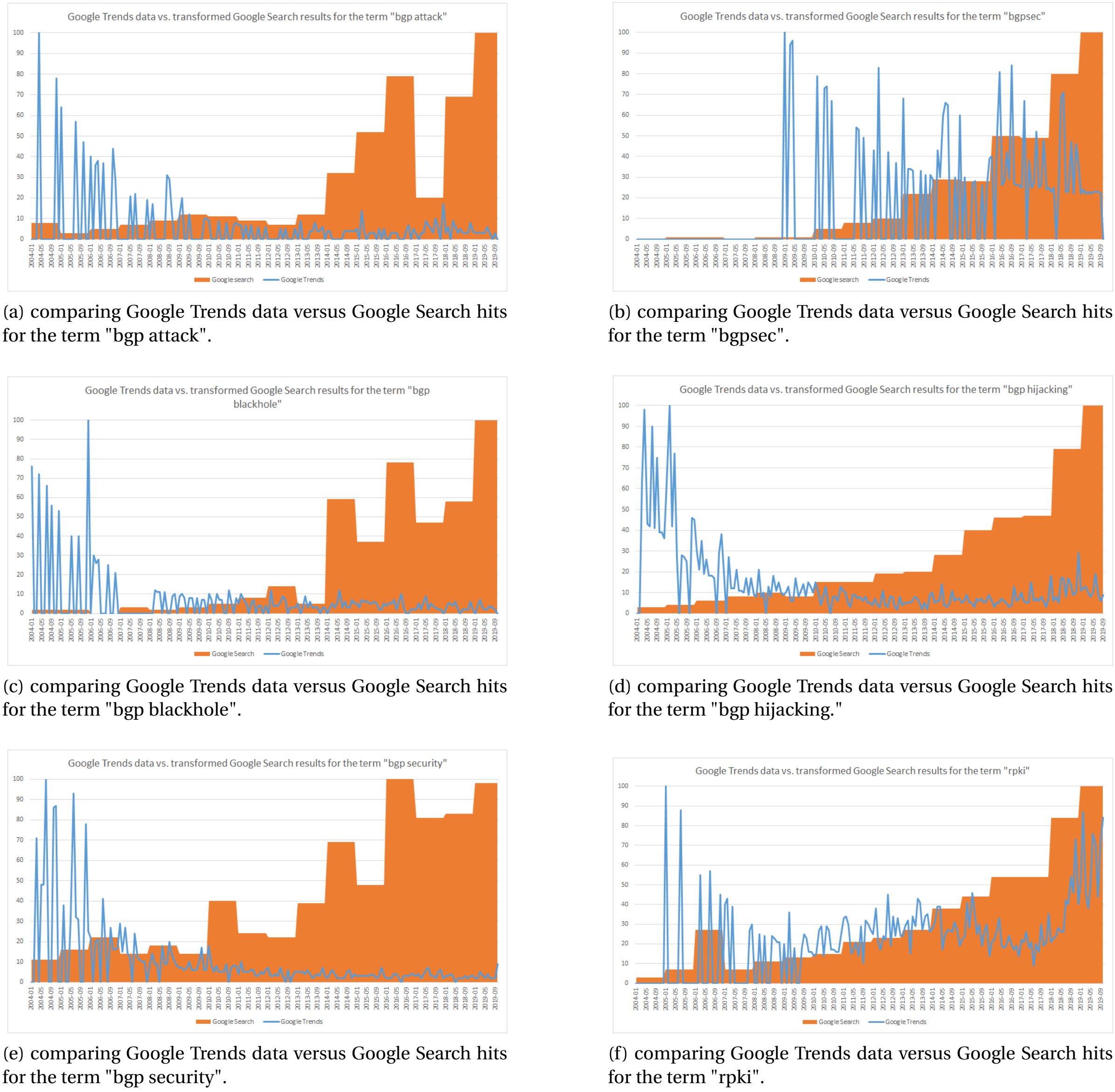
5.7. Conclusions on how the threat landscape has evolved

As a reminder, the purpose of the threat landscape analysis is to answer the first subquestion of the research question, that subquestion being as follows:

How have threats to BGP changed over time, and have proposed security solutions adopted to possible changes?

As can be seen from the literature analysis, the threat landscape is perceived to have changed over time. Attacks such as BGP speaker impersonation used to be considered as threats to BGP security while they are not anymore now, and more recently, academic works also mention the possibility of parameter manipulation to disrupt BGP. One could say that attacks used to focus more on impersonation, and now focus more on the manipulation of the protocol. When looking at the history of BGP security solutions, we can see that these solutions adapt to the threat landscape to a degree, removing BGP speaker authentication to protect with less overhead against the other threats that are still considered important enough. As for comparing Google Trends data with Google Search hits over time, there are no threat landscape relevant conclusions to be drawn from there. As such, this question has the following answer:

The focus of threats used to be on impersonation, hijacking, and path altering. This focus shifted towards manipulation of the protocol, hijacking, and path altering. Hijacking and path altering have thus always been considered a threat to BGP security, whereas there used to be focus on impersonation which changed to focus on manipulation. Over time, standardized threat models appeared as well. Aside from this, BGP security solutions have adapted to the landscape of considered threats, to a degree.



(a) comparing Google Trends data versus Google Search hits for the term "bgp attack".

(b) comparing Google Trends data versus Google Search hits for the term "bgpsec".

(c) comparing Google Trends data versus Google Search hits for the term "bgp blackhole".

(d) comparing Google Trends data versus Google Search hits for the term "bgp hijacking."

(e) comparing Google Trends data versus Google Search hits for the term "bgp security".

(f) comparing Google Trends data versus Google Search hits for the term "rpki".

Figure 5.1: The Google Trends and Google Search results. The blue line shows the Google Trends data over time, indicating search interest in the topics over time. The orange histograms show the transformed Google Search hits, indicating the number of websites discussing this topic over time, which in turn reflects how much has been written about the topic over time. In general, there seems to be a lot of search interest, after which the search interest dies down and more websites discussing the matter appear.

6
BGP security solution analysis

The various BGP security solutions have already been presented, and as has been seen in Chapter 5, they have adapted over time to the change in the threat landscape of BGP. This adaptation should have helped in reducing computational overhead, but because no solution is used on a large scale yet, this is not enough. To see why they are not being used to protect the entire Internet, we need to take a closer look at them. The goal of this chapter is to answer the second part of the second subquestion: what are the security solutions that have been proposed to BGP, what kind of benefits do they provide, and what can we learn from comparing them? The first part has been answered in Chapter 4.

6.1. Approach

The various solutions to secure BGP and how they do what they do have already been proposed in Chapter 4. Now it is time to compare them. To do so, a taxonomy of BGP security solutions will be created that compares them on several criteria and get a better insight into why none of the solutions are deployed on a large scale yet.

In contrast to the approach used to gather information on how the BGP threat landscape has changed over time, there is previous academic work that has included taxonomies of BGP security solutions. Mitseva et al. for example presented a taxonomy in their work [113], which compares security solutions on what kinds of attacks they protect against, the different types of performance delay that they introduce, whether or not the solution is incrementally deployable and even the degree of standardization of the solutions, i.e., have the solutions been (partially) standardized or not. This taxonomy is a good starting point, but it has some shortcomings:

- It does not include some preventative solutions that were already developed by the time it was published, such as HC-BGP and KC-X. It does, on the other hand, include a security solution that is intended to be preventative but has elements that are often attributed to detective security: Listen and Whisper.
- It does not specify if a certain solution introduces centralization. Even though centralization is a negative feature for a BGP security solution as it creates a single point of failure for the Internet or large parts of it. Centralization has also been shown in earlier academic work to be detrimental to the deployment of BGP solutions, such as the RPKI [107].
- It investigates what kinds of performance delays are introduced by the different security solutions that are analysed, but it does not look into which methods the security solutions use to achieve their proposed security. There is also no distinction in severity between the different categories of performance delay, whereas convergence delay would introduce more performance delay in BGP overall than additional bandwidth overhead would. This is because the former impacts all traffic along one AS and the latter only impacts BGP communication between ASes.
- It defines a set of criteria for deployability/adoptability that abstracts a lot of other factors. According to the taxonomy, a solution is deployable if it is incrementally deployable, i.e., if it supports data transfer

from ASes that adopt the solution, through ASes that do not, to ASes that also adopt the solution. This incremental deployability is a reason to use a certain solution, but it would not necessarily incentivize an AS to adopt it. Adoptability is defined in the taxonomy as "the quantity of volunteer ASes willing to adopt the new protocol over time," but the taxonomy does not specify a quantity. It only specifies if the solution is not adoptable at all, partially adoptable, or fully adoptable. The authors of the paper state that the adoptability of a solution depends on the set of initial adopters and their routing policies, but do not estimate the adoptability of different protocols in different deployment scenarios. Not estimating the adoptability of solutions in different scenarios feels like a mistake on behalf of the authors.

Farley et al. also presented a taxonomy in their work, but this only compares solutions on the different types of attacks that they defend against [68]. It also does not include many of the more recent BGP security solutions, as the paper is from 2004. Butler et al. included a similar taxonomy in their work [63], including the different types of security that the solutions provide as well as a degree of how good the security is that the solutions provide. This taxonomy is also outdated as the taxonomy does not include security solutions such as the RPKI and BGPsec, because these solutions were not proposed back when the survey was published.

6.2. Creating the taxonomies

As mentioned, the taxonomies presented in earlier work focused on the different types of weaknesses in BGP that they protected against, or the different kinds of authentication that they provided. The one by Mitseva el al. also included the deployability and adoptability of solutions. This part checked if solutions were incrementally deployable or if they incentivized other ASes to adopt the solution, respectively. As mentioned earlier, however, these criteria are narrowly defined and leave out some important factors.

None of the taxonomies presented in earlier work compared BGP security solutions on the different techniques that they use to achieve this security and the cost associated with using these techniques. The costs are an important factor in providing practical security as a perfect security solution that requires a significant upgrade in hardware or introduces a lot of computational overhead is still impractical. As such, one of the main new features of the taxonomy presented in this thesis will be the inclusion of comparing BGP security solutions on the techniques that they use to provide security as well as an estimate how costly it is.

Comparing preventive BGP security solutions is done by comparing the solutions in four different categories, and these categories are what I based my taxonomies on. Each of these taxonomies compares BGP security solutions on different criteria. The taxonomies are:

- Features: solutions are compared with one another based on the different kinds of authentication/verification that they provide. This is similar to most taxonomies found in earlier academic work but updated to include all of the more recent BGP security solutions. This taxonomy is useful to compare security solutions on what kind of security they provide, as ideally, we would want a security solution that protects against most attacks.

- Cost: solutions are compared on the different techniques that they use to achieve said security to see how costly it is to implement the solution. Note that "costly" does not mean monetary cost only; this can also mean that there is a significant amount of computational overhead introduced when the solution is deployed. As said before, this has not been done before in a taxonomy presented in earlier works. Previous work did investigate what kinds of performance delay was introduced by the different solutions, but did not attempt to quantify the impact. This taxonomy does attempt to quantify the impact. This taxonomy is useful because a security solution for BGP ideally introduces minimal extra cost, as stakeholders (network operators, ISP's) would ideally like a solution that provides protection against most if not all of the threats for a low cost.

- Centralization: how centralizing the solution is. Is the solution completely decentralized, or does it have a single authority that is ultimately responsible for verifying that attestations are correct? Comparing BGP security solutions this way has never been done in previous work as far as I have been able to find. Though the flaws of having a single point of failure for the entire Internet have been known and documented since the flaws in S-BGP were pointed out [94]. This taxonomy is included because the ideal solution is not fully centralized, as introducing a single point of failure for the entire Internet would mean that an attacker has to target only one institution to attack large parts of the Internet. The fact that attackers are able to attack and disrupt availability of large parts of the Internet by attacking one institution is bad for benign Internet users.

- Benefit: the estimated benefit that deployment of the solution provides in different scenarios regarding the level of prior deployment of the security solution. This is the main thing that the earlier work by for example Mitseva et al. lacks, even though they mention that it is important to consider different scenario's of previous adoption. This taxonomy is useful for estimating the security benefit gained in different scenarios. If the estimated security benefit for adopting a security solution is very low because no other ASes have deployed it, then that does not provide an incentive for the average network operator to deploy the solution.

The main purpose of this approach is to compare BGP security solutions not only on the security that they can provide but also on how practical it is to deploy them. Also, in every taxonomy, the base versions of the proposed security solutions will be compared to one another without any other enhancements. There have been proposed enhancements to several solutions, as has been shown with the RPKI, but also other solutions have proposed enhancement, such as one to S-BGP by Kent et al. [93]. This enhancement uses the existing route attestations not only to cover the route parameter but also to cover other parameters such as the MED. These, however, are left out because this might make the comparison rather unfair, as more popular solutions such as the RPKI have more possible enhancements than less popular solutions.

6.2.1. Features taxonomy

This taxonomy is not much different from other taxonomies seen in previous work as it includes many of the same categories as taxonomies seen in previous work included. The categories for this taxonomy are based on the security features found in the various solutions and also based on the protection necessary for securing against the four main threats discussed in the introduction of the thesis. These four main threats also represent the categories that the most commonly discussed threats against BGP fall into, as can be seen in Chapter 5. To recap, the threats are:

- Prefix hijacking: an AS advertises owning a prefix that it does not actually own, intending to redirect traffic from other ASes that was intended to go to the AS that actually owns the prefix, to the AS that hijacked the prefix. This is one of the most common attacks against BGP, with many examples of it having been executed over the years, as can be seen in Chapter 1.

- Path altering: an AS alters a broadcasted path, with the intent to either drop the traffic and prevent it from reaching its destination, or to redirect traffic through itself to gather data about the traffic flow. This is also a relatively common attack, and examples of it have been documented by media reports, and these reports have been mentioned in Chapter 1.

- BGP speaker impersonation: a BGP speaker within an AS claims to belong to a different AS than the AS it actually belongs to. This can cause traffic to be redirected to a certain BGP speaker instead of the whole AS, which could allow an attacker to gather data from the traffic if the BGP speaker is being wiretapped. There have been no media reports on this kind of attack actually happening, but that does not mean that attacks of this kind have not happened, just that they have not been noticed yet. As can be seen in Chapter 5, this kind of attack is often mentioned as a possible attack, and these kinds of attacks could become more common once prefix hijacking and path altering become harder due to increased security. As such, it is useful to have this kind of security in a solution, to pre-empt changes in the threat landscape.

- Protocol manipulation: an AS manipulating the various non-path attributes in the BGP UPDATE messages, to cause other ASes to behave differently than usual. As with BGP speaker impersonation, there have been no media reports of this kind of attack happening, but that does not mean that it is not happening at all. Attacks that are noticeable enough to be picked up by the media could happen once protection against more common attacks is incorporated, and network operators would want this to pre-empt threat landscape changes.

Protection against these threats is offered by the following authentication/verification measures, that form the categories:

- Origin authentication: whether or not the solution provides the AS that deploys it a way to authenticate its ownership of a certain prefix.

	Origin authentication	Path authentication	BGP speaker authentication	Attribute verification
S-BGP	Yes	Yes	Yes	No
soBGP	Yes	No	Yes	No
psBGP	Yes	Yes	Yes	No
IRV	Yes	Yes	No	Yes
SPV	Yes	Yes	No	No
HC-BGP	Yes	No	No	No
BGPcoin	Yes	No	No	No
RPKI	Yes	No	No	No
BGPsec	Yes	Yes	No	No
KC-X	Yes	Yes	No	No

Table 6.1: The features taxonomy.

- Path authentication: whether or not a solution protects against the adoption of paths that have been altered to be false.

- BGP speaker authentication: whether or not a solution provides a way to be able to verify that a certain BGP speaker belongs to the AS that it claims to belong to.

- Attribute verification: whether or not the solution provides a way for the included attributes in the UPDATE message to be verified that they are correct and have not been tampered with.

The taxonomy is presented in table 6.1. From this taxonomy, we can see that every security solution offers origin authentication, but some offer this because of path authentication (as in, origin authentication is not done on its own but rather through path authentication, as every path has an origin), or a different solution is used for origin authentication, such as BGPsec, which uses the RPKI to do this. Also, only IRV provides attribute verification, and only the older solutions provide BGP speaker authentication, which has been discussed already in Chapter 5.

6.2.2. Cost taxonomy

The next taxonomy is the cost taxonomy. This part deals with the different techniques used and hardware upgrades required to deploy a BGP security solution. This taxonomy contains categories that have not been seen in previous work and the main purpose of this taxonomy, as has been said before, is to provide a rough estimate on the cost of deploying and continuing to use a BGP security solution. As such, each category has a low, medium or high cost attached to it, except for one, which has a variable cost. The categories and the estimated costs for each of them are based on what has been used in the various different security solutions, and they are as follows:

- Extra hardware required: does deploying the solution require extra hardware to be in place or not, or does deploying the solution require updated hardware because of processing power requirements. The amount of extra hardware required for a solution to work could have a big impact on the costs of deploying the solution, so it is useful to look into how much extra hardware would be required. The cost of extra hardware can vary wildly for the solutions; because of this, there is no set cost estimate to this category, and the cost estimate for each solution is determined on a case-by-case basis. This category also assumes that all the new hardware needs to be bought and that AS operators do not have spare servers/cables. This assumption is made because the supply of these can vary wildly between ASes. An organization hosting an AS of a major ISP with lots of customers will most likely have more spare servers/cables than an organization hosting an AS that has no customers.

- Uses signatures or certificates: whether or not the solution makes use of signatures or certificates to protect against attacks to BGP. Signatures and certificates have been used in many of the security solutions as they provide a security framework to attest ownership of a resource, preventing the need to develop a new way of attesting ownership which could have a higher cost or be less secure. Not all signatures and certificates are created equal though: the paper detailing KC-X shows that there is a difference of about two orders of magnitude between the time taken to sign a message using cryptographic protocols versus using hash functions [150], so this category has been split up into three different categories:

  - Cryptography-based: whether or not the signatures or certificates are generated using cryptographic methods. Because using cryptographic methods to sign BGP messages introduces a lot of computational overhead as we have seen several times in Chapter 4, the cost estimate for this category is high.
  - Hash-based: whether or not the signatures or certificates are generated using hash functions. Because using hash function-based methods to sign BGP messages introduces two orders of magnitude less computational overhead compared to using cryptography-based methods, the cost estimate for this category is low.
  - Certificates in a repository: whether or not the solution makes use of certificates in a repository, generally a CA in a PKI-system. This approach would involve downloading certificates from a CA. This procedure is more of a setup procedure and not a continuous endeavour, so gets the same cost estimate as "requires data downloads" gets: low.

- Encrypts BGP message content: whether or not the BGP messages are encrypted. Given that an encryption/decryption key is shared between ASes, encrypting BGP message contents would be a good way to prevent hijacks from happening as malicious ASes could only understand the message if they have the key to decrypt it. So it is useful to see whether or not a security solution uses this to provide security. Encryption and decryption is also a computationally inexpensive process as gigabytes can be encrypted per second easily [17], and as such, does not introduce as much computational overhead in sending and receiving BGP messages as using signatures or certificates would. It does introduce more computational overhead than using hash functions would as using hash functions does not require "de-hashing" the hashed message (because this is impossible) or key exchange (as there is no key). Hash functions are also generally designed to create a fast result instead of creating ciphertext that is near-impossible to crack [24]. As such, the cost estimate for this category is medium.

- Uses existing monitoring solutions: whether or not the solution not only makes use of its own BGP feed of incoming BGP messages but also, additionally, makes use of existing BGP monitoring services or solutions to do its job. There are several BGP monitoring services online, such as BGPmon [149], RouteViews [128], or BGPstream [120], which can provide solutions with *additional* data that could be handy for preventing BGP attacks. For example, an attack can be pre-empted because BGP monitoring services provide data that indicates a prefix hijack ahead of time, which can be passed to a solution, which then uses this information to prevent the AS it is operating in from being polluted with false data. As such, it is a good idea to see whether or not a security solution makes use of these services. Using such service or running a solution within the AS does not require as much additional overhead as, for example, encrypting messages, because it would just require a server being connected to the data stream. Also, the amount of data coming in is not that much. The paper on BGPmon, for example, mentions that the system sampled about 26 megabytes of uncompressed data in two hours [149]. This equates to a data stream of 26MB/7200 ≈ 3.6 kilobytes per second. Real-time downloading of this data stream can be achieved even with 56kbps dial-up Internet speeds. So the cost estimate for this category is low.

- Uses hash functions: whether or not the solution makes use of hash functions to provide security. Hash functions can be used in a variety of ways, such as in creating a data structure to secure the path (as seen in SPV) or sending hashes of secret values between one another to verify authenticity (as seen in HC-BGP). Just as with signatures and/or certificates, making use of existing hash functions can provide additional security without having to develop something new, so it is worthwhile to look into which of the presented security solutions make use of it. Hashing is a cryptographic process with low computational overhead, lower than, for example, encrypting or decrypting messages. This is because, as mentioned before, hash functions are designed with speed in mind [24]. So the cost estimate for this category is low.

- Verifies data by querying other ASes: whether or not the solution performs message integrity checking by querying other ASes for additional information that can be used to check if the messages are correct.

Querying previous ASes to check whether the received message is correct would provide a way to not only check for the correctness of the path, but also to check that the other attributes have not been tampered with. This is a security benefit that is not provided by most of the other categories, making it worthwhile to investigate which solutions provide this. Querying ASes for verification would increase the processing time needed for a single message up to about three times the normal amount of time needed though, as a query would have to be sent back to the originating AS. Then, the result would have to be sent to the inquiring AS. This up to threefold increase in processing time is estimated to introduce more computational overhead than using hash functions or using monitoring solutions would (because those introduce very little computational overhead), but less than cryptographically signing messages would (because that method can introduce orders of magnitude more computational overhead). As such, the cost estimate for this category is medium.

- Requires data downloads: whether or not the solution requires prior data downloads to work effectively. This is useful to look into as "requiring data downloads" indicates that the AS deploying the solution is not immediately protected when the solution has been deployed, as it still needs to download data from somewhere. This would be something to keep in mind for network operators. It does not introduce significant continuous overhead as it is more part of a setup procedure, and sometimes needs syncing. Also, in general, the certificates tend not to be that large. For example, the X.509 certificate format is less than a kilobyte in size when compressed [111]. As such, the cost estimate for this category is low.

With these cost estimates in mind, every BGP security solution also gets a total estimated cost, which is based on the estimated costs per category added up. In this sum of costs, if a solution satisfies three categories that have a low-cost estimate, then it will be treated as a medium-cost estimate, and if it satisfies three medium-cost categories, it will be treated as a high-cost estimate.

The taxonomy is presented in table 6.2. The final cost estimate for KC-X is estimated to be medium, as KC-RSA is estimated to be high-cost, and KC-MT is estimated to be low-cost. The paper detailing the scheme recommends using a combination of both, which makes a cost estimate of "medium" a good middle ground. The cost estimates for the category "extra hardware required" are justified as follows: IRV can be implemented on a server within an AS, which in general is a low extra hardware cost considering an AS can consist of many servers, routers, and other parts of a subnetwork. soBGP might require new infrastructure to be in place for the generation and transmission of certificates, which can be anywhere between a lot of extra infrastructure or just a few cables here and there. Because of this, a middle ground has been chosen, and the cost estimate is medium. The RPKI might require some extra data centres over the world to store data for ASes to access, which is also a low cost considering the scope; a well-placed data centre that can cater to the demands of its own AS as well as several nearby ASes could reduce access latency for a lot of ASes, resulting in a low cost for each individual AS.

One thing that stands out from the taxonomy is that no solution makes use of existing BGP monitoring services or solutions. These could help in securing BGP, as they can provide additional information augmenting the information that can be gathered from incoming BGP UPDATE messages. The estimated cost of using one is also low. The main reason for the fact that monitoring services are not used is probably that monitoring services have been developed and deployed after the first few security solutions were presented. BGPmon for example was proposed in 2009 [149] and BGPstream in 2016 [120]. This is one area for potential improvement, though. Something else that stands out from the taxonomy is that the only solution with a final cost result of "High" is S-BGP. This might be another factor (the primary factor being the fact that S-BGP was the first proposed solution) in explaining why so many papers detailing new security solutions cite S-BGP and the prohibitively high computational complexity of using it as inspiration for their new and improved solutions.

	Extra hardware required	Uses signatures or certificates			Encrypts BGP message content	Uses existing monitoring solutions	Uses hashing	Verifies data by querying other ASes	Requires data downloads	Final cost result
		Cryptography-based	Hash-based	Certificates in repository						
S-BGP	No	Yes	No	No	Yes	No	No	No	Yes	High
soBGP	Medium	No	No	Yes	No	No	No	No	Yes	Medium
psBGP	No	No	No	Yes	Yes	No	No	Yes	Yes	Medium
IRV	Low	No	No	No	No	No	No	Yes	No	Medium
SPV	Low	No	Yes	No	No	No	Yes	No	Yes	Medium
HC-BGP	Low	No	No	No	No	No	Yes	No	No	Low
BGPCoin	No	No	No	No	No	No	No	No	Yes	Low
RPKI	No	No	No	Yes	No	No	No	No	Yes	Low
BGPSec	No	No	No	Yes	No	No	No	No	Yes	Low
KC-X	No	Yes, KC-RSA	Yes, KC-MT	No	No	No	Yes, KC-MT	No	No	Medium
Estimated additional cost	Variable	High	Low	Low	Medium	Low	Low	Medium	Low	

Table 6.2: The cost taxonomy.

	Level of centralization	Final result
S-BGP	IANA level	High
soBGP	None	None
psBGP	RIR level	Medium
IRV	None	None
SPV	IANA level	High
HC-BGP	None	None
BGPCoin	None	None
RPKI	IANA or RIR level	Medium/High
BGPSec	IANA or RIR level (makes use of RPKI)	Medium/High
KC-X	None	None

Table 6.3: The centralization taxonomy.

6.2.3. Centralization taxonomy

The next taxonomy that will be presented is the centralization taxonomy. This taxonomy presents how centralizing the solutions are. As can be seen from the different BGP security solutions, there can be different levels of centralization; S-BGP for example uses two PKIs centralized at the highest possible level, the IANA level, and as such is a completely centralized security solution, while one of the possibilities for the RPKI is that the RIRs can act as CAs, which would result in a not completely centralized solution but rather one that is centralized at a lower level. We have already seen that a higher level of centralization is not beneficial for BGP security solutions as that introduces a single point of failure. But that does not mean that centralization of trust itself is so bad that security solutions that make use of it cannot be used to secure BGP messages. For example: VPN services like NordVPN [32] provide their services by connecting clients to their servers. More often than not, a server will serve more than one client, creating a centralizing factor. An attacker could disrupt service for many users by attacking one of these servers. However, VPN services often offer more than one server (NordVPN offers over 5000 as of May 2020), and users can choose which server they want to connect to, which can be used to get around this problem. And the popularity of VPNs [44] indicates that the introduced centralization does not mean that they are unusable. Because of this, it is worthwhile to at least look at the degree of centralization that some of these solutions introduce.

This taxonomy only has one category, the "level of centralization" category: to what level the solution makes use of centralized trust. This can be the highest possible level, with the trust anchor being the IANA (which is IANA level), a lower level with trust anchors being RIRs or LIRs, or of course no level of centralization at all.

Each solution also gets a final level of centralization, which can be one of the following:

- High: the solution is centralized at the IANA level.
- Medium: the solution is centralized at the RIR level.
- Low: the solution is centralized at a lower level than the RIR level.
- None: the solution is not centralized at all.

The taxonomy is presented in table 6.3. At first glance, one can see that most solutions are either completely centralized with a trust anchor at the IANA or completely decentralized. Only the RPKI, BGPsec (because it uses the RPKI) and psBGP are centralized at a level lower than the highest possible level. In the case of the RPKI and BGPsec, that is only because the implementation allows for both IANA and RIR level centralization. Also, if a solution is centralized, it uses a PKI. One solution that seemed to have some degree of centralization without using a PKI was SPV, but upon closer inspection of the specification, the solution is at least heavily implied to use one, and as such, I assume that it does use a PKI.

6.2.4. Benefits taxonomy

The final taxonomy that will be presented is the benefits taxonomy. This taxonomy estimates the added benefit for an AS to deploy the security solution in several different scenarios of prior deployment. The goal of the taxonomy is to estimate how well a solution performs in different scenarios of prior deployment. This is useful to investigate as we would ideally like a solution to provide substantial security benefits without relying on other ASes having deployed the same solution already.

The different scenarios are meant to represent a sliding scale of prior deployment, and they are as follows:

- All other ASes have already deployed the solution. This represents the ideal scenario for deployment: all the other ASes that make up the Internet have already deployed it. So, the expected benefit from your AS deploying it would logically be the highest possible as the security-related data being sent between ASes is received and processed properly. The solutions that would provide security against the highest amount of threats would be the best in this scenario if cost is not taken into account.

- A uniformly random selection of 50% of all the other ASes have deployed the solution. This represents a less-than-ideal deployment scenario and is intended to show which solutions offer incremental security and which do not. Incremental deployment is useful in security solutions as it is nigh impossible to force every AS to adopt the same security solution at the same time.

- No other AS has deployed the solution. This scenario is the least ideal (and, at the same time, the scenario closest to reality). It is intended to show which solutions offer any kind of security if only one AS adopts them. Ideally, a security solution offers at least some security benefits to the deploying AS if no other AS deploys it, as benefits of deploying a security solution would entirely depend on other ASes also deploying it otherwise. That would be less than ideal as it would create a catch-22 situation, where no AS deploys the security solution because no other AS has deployed it.

Similar to the cost estimate, the benefit estimate for each of the scenarios follows a discrete scale, with the levels going from the highest possible level called Maximum, to High, to Medium, to Low, to None. As for the levels, recall that there are four security features in the features taxonomy, those being origin authentication, path authentication, BGP speaker authentication, and attribute verification. With these in mind, the benefit per solution per scenario is estimated according to the following rules:

- Maximum: all four of these security features are present and work in all cases for this solution in this deployment scenario.

- High: two to three of these security features are present and work in all cases for this solution in this deployment scenario.

- Medium: one of these security features is present and works in all cases for this solution in this deployment scenario, or multiple security features are present but only partially work, and the parts sum up to be at least one and less than two. For example: if both origin authentication and path authentication are present, but they both only work at an estimated 50% effectiveness in the case of a random 50% of all ASes deploying the solution, then these two parts sum up to be one.

- Low: one or more security features are present for this solution in this deployment scenario, but the features only partially work, and the parts sum up to be less than one.

- None: no security features are present for this solution in this deployment scenario.

The taxonomy is presented in table 6.4. In general, if the solution offers incremental benefit to deployment (soBGP does this for example and was specifically designed to do so, as well as a few others), then the security benefit is expected to work 50% of the time if a uniformly random selection of 50% of all the ASes in the Internet deploy it. The selection of ASes can impact the estimated benefit a lot, as one paper that criticised S-BGP for requiring too much computational power to be feasible stated that only deploying it in twelve ASes could protect half the Internet [94]. Furthermore, 85% of all the autonomous systems that make up the Internet are stubs [74], which means that they are only connected to a provider. If a significant portion of these is included, the estimated partial security benefit is logically lower. However, if we were to select a uniformly random subset of 50% of all the ASes and average out the estimated security benefits over all the selections, the average security benefit would be around 50% of the amount of benefit that full deployment would provide.

One important thing that stands out from this taxonomy is that no security solution offers any kind of protection if no other AS has adopted the security solution yet. This is a huge problem when considering practical

	All other ASes adopted solution	50% of ASes adopted solution	No other ASes adopted solution
S-BGP	High	None	None
soBGP	High	Medium	None
psBGP	High	None	None
IRV	High	Medium	None
SPV	High	Low	None
HC-BGP	Medium	None	None
BGPCoin	Medium	Low	None
RPKI	Medium	Low	None
BGPSec	High	None	None
KC-X	High	Low	None

Table 6.4: The benefits taxonomy.

BGP security as mentioned before: why would an AS want to be an early adopter if there is no benefit for the AS itself to be gained from doing so? Security solutions can be made to be incrementally deployable and secure an AS against most if not all possible attacks, which is a *reason* to deploy a security solution. But the fact that there is no security benefit for an AS to deploy the solution if it is the only AS to deploy it means that there is no *incentive* for ASes to deploy security solutions on their own. It should be noted that the RPKI has already alleviated this issue somewhat because it is already deployed on a small scale, but it also introduces either a single point of failure for the entire Internet (if rooted at the IANA) or a single point of failure per continent (if rooted at the RIRs), which is less than ideal.

6.3. What can we learn from the taxonomies?

There are two main things that can be learned from the taxonomies:

1. There is no BGP security solution that makes use of existing monitoring solutions, which is an area that could be worthwhile to look into. These monitoring services could provide data from different vantage points, which can aid in preventing attacks from affecting routing by detecting anomalies in routes ahead of time and preventing the AS from taking those suspicious routes. This is not so much prevention-based security but more detection-based security.

2. There is no BGP security solution that provides security benefits if only one AS deploys it. This is a huge incentive barrier to the deployment of secure BGP, as ASes need incentives to protect their routing, and only solutions that would provide security benefits if a single AS would adopt them would provide this incentive, as the AS in question can adopt the solution to secure its traffic, staying ahead of its competitors.

The second point can be extended to a general rule: there is not going to be a preventative BGP security solution that provides a benefit if no other ASes adopt the solution because to verify prefix ownership, path integrity, et cetera. BGP speakers need to exchange extra data with either one another directly (if the solution is decentralized) or via a central authority (if the solution uses one). This extra data is not specified in the existing BGP specification, even though there is room for doing so by expanding the attributes field.

This is an aspect of the problem in deploying preventative BGP security that has often been overlooked. Prior work done on the subject has argued that the proposed solutions to secure BGP introduced too much computational overhead, introduced a single point of failure, or that there were attacks possible that completely circumvented the added security benefits. As seen in the related work, there have been publications that argue for BGP security deployment strategies, such as the one proposed by Gill et al. [74]. Still, these also do not mention that security solutions need deployment incentives. Economic incentives are a big part of incentivizing users to deploy security solutions, as outlined by Alderson et al. [54]. Also, as has been mentioned before, around 85% of the ASes in the Internet are stub ASes. These kinds of ASes are rarely attacked because attackers are more incentivized to go after bigger ASes to make more impact, so the stub ASes have next to no economic incentive to deploy secure BGP. Aside from economic incentives being important, governments can also mandate the implementation of security solutions, as suggested by Murray et al. [115]. However, some governments benefit from BGP being as vulnerable as it is now. This is evidenced by the Pakistan Telecom incident of 2008. This means that government-mandated deployment cannot be trusted to work.

The fact that no security solution provides benefits if no other AS adopts the same solution is also somewhat backed up by the work of Chan et al. [64], which (as mentioned before) has shown that there is a cost versus benefit threshold, where if the estimated benefit is higher than the estimated cost, almost all ASes in the Internet will deploy the solution. In contrast, almost none of the ASes will deploy the solution if that is not the case. The work of Chan et al., however, never mentioned this catch-22 situation being a problem.

Because preventative security lacks economic incentives to deploy it and governments cannot always be trusted to order AS operators to deploy it in their country as they might have a stake in the protocol being so vulnerable to these kinds of attacks, a different approach needs to be taken to secure BGP against attacks. In information security, there are three types of security controls: preventative security controls, detective security controls, and corrective (or, more commonly called, reactive) security controls [13]. The logical next step would be to look at detective security controls, of which several have been proposed over the years. There are pros and cons to moving away from preventative security to detective security, however:

Pros:

- ASes that deploy detective security would not have to rely on other ASes deploying detective security as well, as detective security controls most likely do not need direct or indirect communication with another AS to perform attack detection.
- Depending on the computational requirements for detective security, the deployment costs could be quite low, with likely the only hardware necessity being a small server that reads BGP messages and raises alarms when an attack is detected.
- Not all ASes would have to deploy the same detective security solution if the given solution does not have to communicate with other ASes to work.

Cons:

- Perfect security would be a near impossibility when moving from preventative to detective security controls. Detective security generally becomes better over time, due to more data being gathered and, as such, more accurate detections being made. Still, there will always be false positives and false negatives.
- Only detecting attacks would not be sufficient, as there still needs to be a way from preventing false routing data from entering the routing database of a BGP speaker. Reactive security is, therefore, ideally built into detective security.

As can be seen, there are more pros than cons. Also, while the cons mentioned are bad, the fact is that we have the choice between no security if only one AS adopts a solution in the case of choosing preventative measures, versus imperfect security if only one AS adopts a detection method. As such, it makes sense to move away from preventative security and look at detective security from now on.

6.4. Conclusions on the analysis of BGP security solutions

As a small recap, the purpose of the taxonomy was to answer the second part of the second subquestion of the research question, that subquestion being:

What are the security solutions that have been proposed to BGP, what kind of benefits do they provide, and what can we learn from comparing them?

The first part has been answered by Chapter 4 already, which presented the various prevention-based BGP security solutions. There have been several papers discussing reasons why none of the BGP security solutions have been deployed on a large scale yet. These have arguments ranging from a lot of computational overhead being introduced for little benefit, to attacks still being possible even with full deployment of the solutions, to requiring a central authority for the whole Internet. One argument that has not been brought up that the taxonomy shows us is that there is no security incentive for ASes to be the first to deploy a security solution. As such, the question of "what kind of benefits do they provide, and what can we learn from comparing them?" can be answered as follows:

Aside from protection against threats that are not considered as important enough anymore, additional computational overhead or reliance on a central authority that some of these solutions introduce, there is also the lack of a security incentive for ASes to deploy security solutions as there is no preventative solution that provides security benefits without other ASes also deploying the same solution.

7 BGP detection algorithms

Because preventative BGP security creates a catch-22 for deployment, it is wise to transition away from considering preventative security to considering detective security. As such, it is important to take a look at the detection algorithms that have already been developed. There have been many detection algorithms proposed over the years, as summarized by, for example, Al-Musawi et al. [53] and Nicholes et al. [117]. These kinds of detection algorithms range from anomaly detection techniques that also detect general link failures, to detection algorithms specialized in finding BGP hijacks. This thesis will focus on the latter category. This chapter shows several detection schemes that were developed to find BGP hijacks.

7.1. PGBGP

PGBGP is a protocol-preserving enhancement to BGP proposed by Karlin et al. in 2006 [88]. The algorithm that the enhancement introduces focuses on detecting (sub)prefix hijacks and the idea behind it is that unfamiliar routes are to be treated with caution before being adopted: if a route is never seen before, then the BGP speaker should wait some time before choosing it as it could be a malicious route.

The way the algorithm works is as follows: because the algorithm needs to know which routes can be considered normal, it first creates a history of known origins for each prefix from both the router's RIB and history of updates. This continues for h amount of days, where h can be specified by the user. This means that during initialization, all updates in the first h days are accepted. After the initialization is over, suspicious routes are quarantined for s amount of days, where s is also specified by the user. A route is suspicious if the prefix advertised has a different origin, and if none of the (recently seen) origins can be found on the route. Finally, the algorithm removes stale data from its database, and as such, PGBGP removes known origins for a prefix if these origins have not appeared in the router's RIB in the last h days. If no mention of the prefix is found in the RIB in the last h days, the entire prefix is removed. Quarantining suspicious updates for any amount of time does limit the dynamic update capability of Internet routes, however, and AS operators that want to use this scheme need to take this into account.

From the description of how the algorithm works, it is clear that PGBGP only requires a user to set two different parameters, h and s. Both parameters should not be too short or too long. The authors suggest that s should be set to one day, while h should be set to ten days.

Preventing prefix hijacks is done by giving suspicious routes the lowest possible preference when forwarding them to the selection procedure. Preventing subprefix hijacks is more complicated as the router has no normal routes to the subprefix to compare the abnormal route to. The algorithm approaches this problem by forwarding packets using the route for the larger prefix. When possible, the superprefix route selected should lead to a neighbouring AS that has not announced the subprefix, as this AS can redirect data packets along the suspicious path anyway if there is no control mechanism in place to stop it.

The authors have tested the algorithm using a BGP simulator called BSIM [86]. Unfortunately, it seems that BSIM is no longer online as of 2020. The simulations show that, for exact prefix hijacks, PGBGP can protect most ASes from them if only a small fraction of ASes, mainly core ASes (ASes that are connected to a lot of other ASes), use the enhancement. This is because these core ASes deploying the enhancement can detect false routes and ignore them, preventing the spread of these routes to other ASes. To identify subprefix hijacks on the same level of effectiveness as preventing prefix hijacks, more ASes need to deploy the enhancement.

To save time for network operators to manually go over all suspicious routes and determine for each route if it is suspicious or not, the authors present a prototype system called the Internet Alert Registry or IAR for short. It is an opt-in service where network operators can submit their email address and the ASN's that they wish to monitor. In the case of a hijack happening where either the instigating or victim AS is identified by the ASN which is in the set of ASN's that the operator wishes to monitor, the operator gets an email informing them that the AS with the given ASN is either the victim or the perpetrator of an attack. The authors have also implemented a proof-of-concept IAR [87], but just as with BSIM, that is no longer online as well.

7.2. PHAS

The Prefix Hijack Alert System, or PHAS for short, is a system developed by Lad et al. and was first presented in 2006 [99]. The approach outlined by the authors is to examine BGP routing data collected from BGP collectors such as RouteViews or RIPE and provide real-time notifications of any potential prefix hijacks happening. Detection of prefix hijacks is done by defining an origin set for each prefix and tracking changes in this set. The origin set $O_{SET}(P,t)$ is defined as the union of all the AS numbers of ASes that originate the prefix P at time t, where the origin of P is defined as the last ASN in the route that a router M_i has announced: $O_{SET}(P,t) = \cup_{i=1}^{N} origin(M_i,P,t)$.

The system itself consists of four components: user registration, origin set monitoring, notification transmission, and the local notification filter. Figure 7.1 shows how these components are connected.

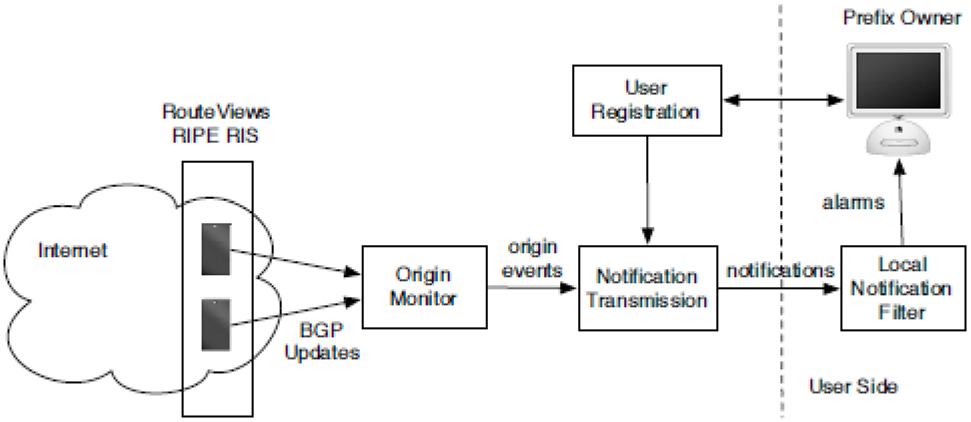


Figure 7.1: a schematic overview of PHAS. Source: Lad et al., 2006.

The functionality for each of these components is as follows:

1. User registration: prefix owners that are interested in using the system as a service need to register their prefix with the PHAS server and provide one or several contact email addresses as well as a password. Any changes to the account require the user to provide their email address as well as their password.

2. Origin set monitoring: PHAS maintains an origin set for each registered prefix. If there is a change to this set, an origin event is generated and forwarded to the notification transmission system.

3. Notification transmission: this is the system that takes origin events from the origin set monitoring system and decides whether or not the origin event warrants notification to be sent to the prefix owner. This is harder than translating the origin event into a message that can be read, as this system also needs to decide which of the email addresses the notification is going to be sent.

4. Local notification filter: checks whether or not an origin event should be sent to the network operator. Network administrators can even provide their filters not to get emails when a legitimate origin event happens, such as a transfer of prefix from one AS to another.

The authors propose two ways to detect origin set changes. The first one is instantaneous origin change. This approach involves gathering data on prefix origins from one of several BGP monitoring services, and checking if there is a change between $O_{SET}(P,t,k)$ and $O_{SET}(P,t-1,k)$. If there is a change, then an origin event occurs where additional and removed origin ASN's are included, as well as the complete current origin set. The second way is the windowed origin change. This method introduces the notion of the windowed origin set $O_{SET}(P,t,k)$, which is the set of all origins for P that were observed by at least one router M_i during the time $[t-k,t]$. Formally, $O_{SET}(P,t,k)$ can be defined as $O_{SET}(P,t,k) = \cup_{i=1}^{N} origin(M_i,P,t,k)$, and $origin(M_i,P,t,k)$ can be defined as $origin(M_i,P,t,k) = \cup_{i=t-k}^{t} origin(M_i,P,t)$. This approach provides a continuously moving window for monitoring the origins of P. Similarly, it sends origin events to the first approach, but this time around when $O_{SET}(P,t,k)$ is not equal to $O_{SET}(P,t-1,k)$. Introducing a sliding window reduces the number of repeated origin events but delays the notification of origin-loss events. Because of this, the authors propose an adaptive window size per prefix, starting at one hour, which becomes larger as more notifications come in but also decreases exponentially over time. This one-hour window size does mean that PHAS does not provide real-time detection.

The authors propose several extensions to the system. The first one is to classify prefixes as one of three cases:

1. False origin, which is the standard case.
2. False last hop, where the last hop in the path can be deemed invalid since the prefix owner's AS knows its immediate neighbours.
3. Covered prefix hijack, also known as the subprefix hijack.

For the latter two cases, other extensions need to be made to the system to be able to detect these hijacks. In the case of detecting false last hops, the system could create the last-hop set $LH_{SET}(A)$ as the set of last hops for the AS A, and send notifications in the case that a new last hop is detected. In the case of detecting covered prefix hijacks (or subprefix hijacks), the system can be adapted to check if the prefix announced is a more specific version of a different prefix owned by a different AS.

7.3. Zheng-Ji-Pei-Wang-Francis

Zheng et al. proposed a light-weight service-based scheme for real-time detection of IP prefix hijacks in 2007 [158]. This scheme has no explicit name. As such, it is referred to by the combination of the last names of the authors. In this scheme, several vantage points throughout the world monitor BGP data, and the scheme makes use of two techniques to detect prefix hijacks: network location change detection and path disagreement detection. The first of these techniques makes use of the fact that IP prefix assignment on the Internet is on a long-term basis, and as such, the network distance between two points is likely to remain constant over long periods. In the case of a prefix hijack happening, there is a high chance that the hijacking AS and the victim AS are not close by. As such, the network distance varies significantly. The second technique is intended to be used in conjunction with the first one. This technique makes use of a reference point along the path from the monitor to the monitored prefix, which should be close to the monitored prefix but still fall outside of its AS. Because of topological closeness, legitimate changes should affect the route from both the monitor to the target prefix and the route from the monitor to the reference point. If these paths are very different, then it is likely that a prefix hijack has occurred. Figure 7.2 shows the most likely difference between a legitimate route change and prefix hijacking.
The detection scheme itself consists of three basic steps:

1. For each target prefix, several monitors are selected from the set of candidate monitors.
2. Each selected monitor periodically measures the network distance to each of its target prefixes to potentially detect significant alterations in hop counts.
3. If a significant change is detected, then the monitor will measure the disagreement between the path to the target prefix and the path to the reference point. This disagreement of paths is measured by the amount of different ASN's on the same location, relative to the length of the path.

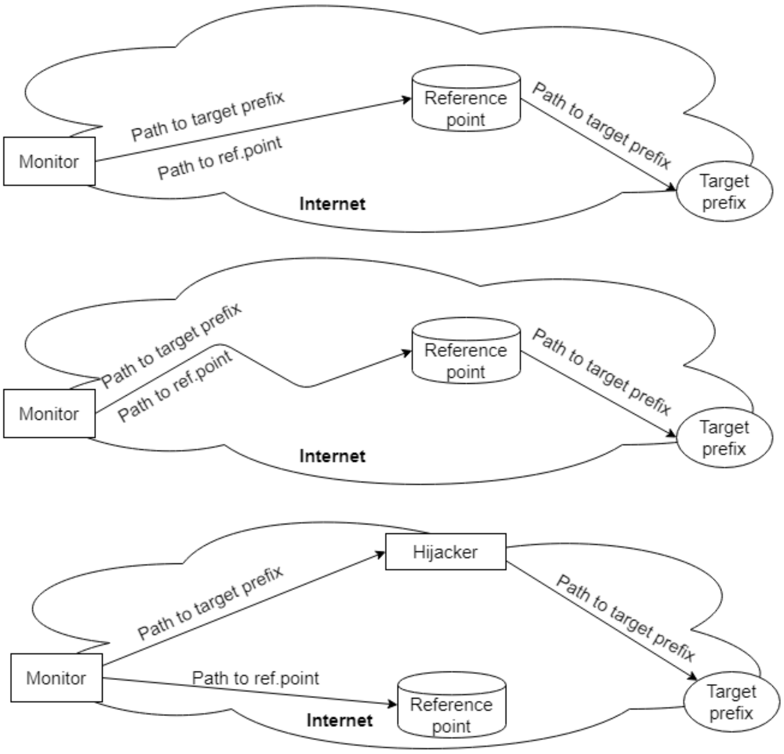


Figure 7.2: the different situations using reference points as described in the scheme. The top image shows normal behaviour. The middle image shows legitimate route changes, where the path from monitor to target prefix is still mostly the same as the path from the monitor to the reference point. The bottom image shows what happens in the case of a hijack. In this case, the paths from monitor to reference point is entirely different from the path from the monitor to the prefix. Source: Zheng et al., 2007.

For the first step, the candidate monitor set is initially the set of all the monitors that the proposed service has set up around the world. In general, the subset of selected monitors should each be in different geographical regions. The authors suggest modelling the selection of monitors per target prefix as a hierarchical clustering problem, where the initial amount of M clusters/monitors is reduced to m clusters by merging two clusters with the largest correlation between them until only m clusters remain, and then selecting a random monitor from each of the m clusters.

The second step makes use of a detection algorithm that falls into the category of change-point detection algorithms. Because the hop count generally remains stable and any change happening to it is all of a sudden, the simplest time-series change detection method is chosen: a moving average with a fixed-size sliding window. The technique makes use of two fixed-size sliding windows W_1 and W_2, with the first of them being used for the past average hop count and the second being used to smooth out significant noise. Therefore the window for the first should be greater than the second. The authors suggest using size 12 for the first window and size 10 for the second. If the relative difference between these two moving averages is big, then it is very likely that a prefix hijack has occurred.

The third step involves measuring the similarity between paths. Because the method is data plane-based and only gets traceroutes instead of BGP paths (which belong to the control plane), these traceroutes have to be converted to AS number routes first. This can be done using websites such as the by-the-authors-suggested iPlane [48]. The similarity of two paths with the same origin is defined as follows: given two paths P_1 and P_2 where P_1 is longer than or of equal length to P_2, define P_1' as the same path as P_1 but of the same length as P_2 with the other AS numbers at the end of the path cut out. Then, calculate the Hamming distance d between P_1' and P_2. This distance is the amount of AS numbers that are the same and at the same location in the path. Then, given d and the length of P_2 as l, the similarity can be calculated as follows: $s = 1 - \frac{d}{l}$. If this similarity is very low, then it is likely that a hijack has occurred.

7.4. iSPY

iSPY is a prefix hijack detection algorithm proposed by Zhang et al. in 2008 [155]. The detection system is designed with the following critical requirements in mind: it must provide real-time detection, it must be accurate, it must be light-weight and not introduce a lot of computational overhead, it must be easy to deploy, it must provide the prefix owner an incentive to deploy it, and it must be able to notify the victim of the hijacked prefix.

The design of the scheme exploits one essential characteristic of prefix hijacks, and that is that they always pollute a significant portion of the ASes that make up the Internet. As such, during a hijack, probes that are initiated from the victim are expected not to be able to get to their destination, because they will go from the victim to the probed destination to the attacker, as the attacker claims to own the prefix of the victim. This affection of a significant portion of the Internet is what can be used to determine the difference between hijacking and link failure. To do this, the reachability of the Internet for the prefix owner is recorded through the use of a set of paths called victim paths or vPaths. An example of how these vPaths are created from a topology can be seen in figure 7.3.

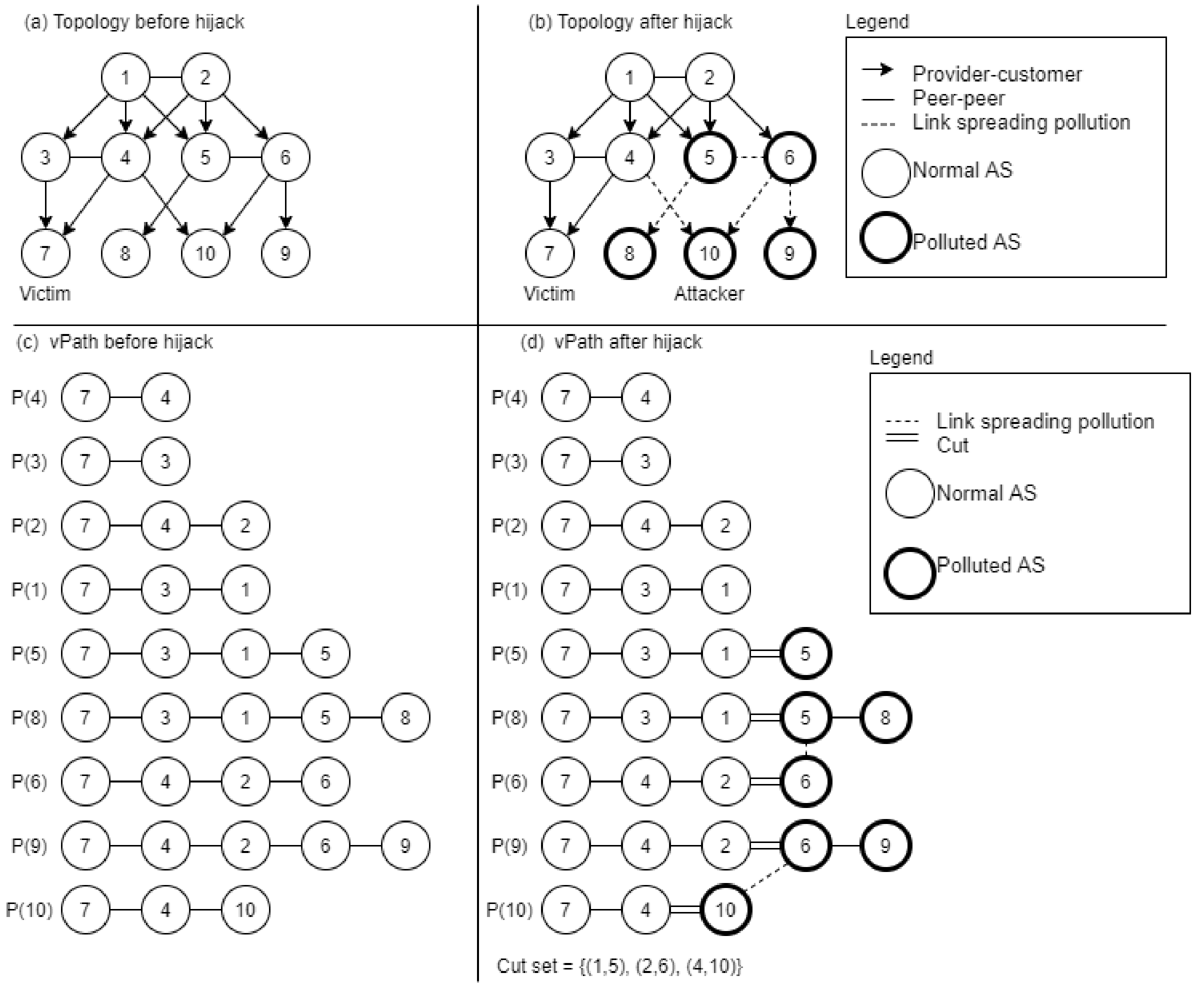


Figure 7.3: examples of the vPath before and after a prefix hijack has polluted a part of the Internet. In this image, AS 7 is the probing AS and AS 10 the attacking AS. Probes launched from AS 7 that reach an AS that is polluted by the hijack of AS 10 will receive the probe, but instead of sending a reply back to AS 7, they will send it to AS 10 as that AS now claims to own the prefix of AS 7. This image also introduces the notion of cuts in the vPaths, which will be explained shortly. Source: Zhang et al., 2008.

These victim paths are then used for hijack detection by constantly probing to all transit ASes from the current AS, creating snapshots of the reachability of the current Internet as seen from the perspective of the prefix owner, and comparing it with an old reachability snapshot. If there is a significant difference in reachability, then there might be a prefix hijack happening. To measure differences in reachability, the notion of a cut in the vPath is introduced. If not all of the ASes in the old snapshot are reachable in the new one, then the links where the begin points are the last ASes that the traceroute obtained a reply from are considered the cut. For example, using figure 7.3, we define the set of links (1,5), (2,6) and (4,10) as the cut, as 1, 2 and 4 are the last ASes that AS 7 (the probing AS) got a reply from.

Considering the authors wanted to exploit the fact that, in the case of a prefix hijack, there is always a significant portion of the Internet affected, the size of the cut (which is the number of links in the set) can be used as a distinguishing feature, or a signature of a prefix hijack happening. If the size of the cut is large, a prefix hijack is most likely the culprit. If the size is small, then it could be a hijack, but it is more likely that it is a link failure or something else, but not necessarily an attack. The authors justify this line of reasoning by simulating prefix hijacks on a realistic AS-level Internet topology and then measuring the sizes of the cuts created when an attack is executed. From these simulations, they have found that over 99% of prefix hijacks resulted in a cut size of more than 20. While this is a good result on its own to find a discriminatory feature for detecting whether or not a prefix hijack occurs, link failures are not simulated. As such, no comparison can be made between cut sizes for prefix hijacks and cut sizes for link failures.

7.5. Hu-Mao

Hu and Mao proposed a scheme for IP prefix hijacking in 2007 [83]. As this scheme also has no specified name, I have decided to call it Hu-Mao after the authors of the paper presenting it. The authors base their scheme on an attacker model for IP prefix hijacking, which consists of several possible attacks:

1. Prefix hijack: the most direct way of hijacking a prefix, by claiming false ownership of a prefix that belongs to a different AS. This can be detected by using different vantage points to see if one or several different ASes originate the same prefix, called a multiple origin AS (MOAS) conflict. There are legitimate reasons for such a conflict to exist, however, and these should be taken into account.

2. Hijacking a prefix and its AS: this attack involves attackers in a certain AS changing routes, so that part of the traffic to the victim AS is redirected through them to their legitimate destination. Filtering these routes requires the router to have an accurate view of the BGP topology.

3. Hijacking a subnet of a prefix: standard subprefix hijacking. These can be detected by extending the definition of MOAS conflicts to subMOAS conflicts, by including origin conflicts which involve subnetworks of a prefix.

4. Hijacking a subnet and its AS: this type is a combination of attack 2 and attack 3, and combines the advantages of both to avoid both types of conflicts.

5. Hijacking on a legit path: this involves violating the rule of forwarding traffic. The authors do not focus on this kind of attack.

The detection algorithm focuses on detecting the first four of these attacks by use of data-plane properties of the network. There are four different methods used in detection, those being the fingerprinting techniques, techniques based on inter-AS relationships, the reflect-scan, and customer-provider checks. Because this scheme uses several different techniques, whereas others used only one or two techniques, and because the techniques used require various data sources, I have decided to split this section up into several different subsections for clarity, describing the techniques used and how detection is finally performed.

7.5.1. Fingerprinting techniques

The consistency of the destination network is a major factor in the successful detection of hijacks, and as such, fingerprinting methods are used in the scheme. The authors discuss four different types of fingerprinting techniques in the paper. While using each of them individually cannot guarantee being able to distinguish two different machines, when one combines this data, it can significantly reduce the false positives and negatives:

1. Host OS properties: attackers are unlikely to use a similar OS, and even less likely to have their OSes set up the same way.

2. IP ID probing: the IP header requires a 16-bit identifier field, which is designed to be unique for each IP datagram. The authors use this method to verify whether two machines are the same by sending probe packets: if the IP ID is incremented, then the machines are the same, and if not, then they are not the same. This technique is based on the work of Bellovin et al. for counting hosts behind a NAT [59].

3. TCP timestamp probing: the TCP timestamp of two machines is probed to see if they are the same. If the timestamp is relatively similar, then they are most likely the same machine. If not, then they are most likely different. This is roughly the same as IP ID probing.

4. ICMP timestamp probing: sending ICMP timestamp requests to two different machines is likely going to result in two different results, whereas if they are the same machine, then the results should be similar.

7.5.2. Techniques based on inter-AS relationships

Aside from these techniques, three other techniques are used based on inter-AS relationships:

1. Edge popularity: the popularity of links between ASes is measured and used as an indicator of whether or not a link is valid. Brand new links are more likely to be fake links than links that have been seen a few times already.

2. Geographic constraint: if one observed link between two ASes connects two ASes that are geographically very far away, then that link is likely fake (note that this does not necessarily have to be the case; for example, AS1103 is a Dutch AS which is directly connected to AS6461, an American AS [2])

3. Relationship constraint: using the work of Gao et al. on inferring AS relationships [70], the scheme infers inter-AS relationships and detects obvious violations of these relationships, indicating fake AS links.

7.5.3. Customer-provider checks

This check operates on the assumption that providers will never intentionally hijack their customers' routes, as this can only hurt them financially. The check is explained in the extended version of the paper of the authors [82], and is based on the valley-free routing principle [71]: edges appearing before the tier-1 ASes in the path are edges from customer to provider, and edges appearing after tier-1 ASes are all provider-to-customer. Tier-1 ASes are easy to identify because they have no providers.

7.5.4. Reflect-scan

The final technique that the scheme uses is the reflect-scan technique. This technique is based on the TCP Idlescan technique, and similar to IP ID fingerprinting, it makes use of predictable IP ID incrementing when packets are sent to a victim AS. The process is displayed in figure 7.4, and is as follows:

1. The probing host H_0 will first send probe packets to a host H_1 of a subnet P' of a prefix P and record the IP ID received.

2. Then, H_0 will send a SYN packet with the IP address of H_1 as the source to host H_2, which owns prefix P.

3. Then, H_1 will communicate back and forth with H_2.

4. Finally, H_0 will communicate with H_1 again. If H_1 has only incremented its IP ID once, then the message that was supposedly between H_1 and H_2 has never reached H_1, indicating a subprefix hijack.

7.5.5. Detection of hijacks

Detection of the different hijacks is done by using several different techniques. Table 7.1 displays an overview of the techniques used per possible attack. The reflect-scan is used in subprefix hijack detection because it was made for that purpose, whereas normal prefix hijack detection can be done using the fingerprinting methods.

Besides this, the authors present the system architecture that they have in mind for the implementation

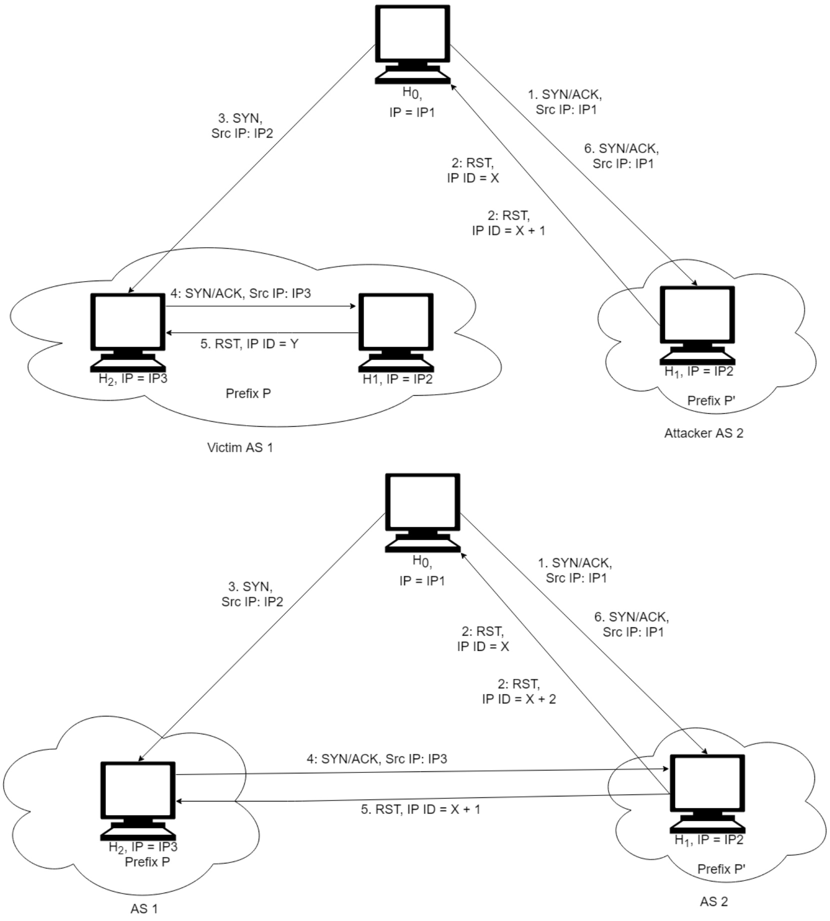


Figure 7.4: a schematic overview of the reflect-scan technique. The top image represents the situation where a hijack has occurred, while the bottom image represents normal behaviour. Source: Hu et al., 2006.

of this scheme. This scheme consists of three parts: the monitoring module (which collects and processes BGP updates in real-time and classifies updates into either valid updates or anomalous updates), the probing module (which takes processed input from the monitor module and selects appropriate probing techniques), and the detection module (which analyses and compares the results from the probing module to determine whether or not it is a hijack). Figure 7.5 gives an overview of the proposed system architecture. BGP updates enter the system and get classified using the described techniques to determine whether or not a hijack has taken place. If a subnet hijack has likely taken place, then the probing modules are used to make sure that such a hijack has taken place.

7.6. Argus

Argus is a system for prefix hijack detection proposed by Xiang et al. in 2011 [146]. The system consists of three modules: the Anomaly Monitoring Module (AMM), the Hijacking Identification Module (HIM), and the Live-IP Collection Module (LCM). An overview of the architecture is shown in figure 7.6.

	AS not hijacked	AS hijacked
Prefix	Fingerprint-based checks	Fingerprint-based checks, inter-AS relationship checks
Subprefix	Customer-provider checks, reflect-scan	Inter-AS relationship checks, reflect-scan

Table 7.1: overview of the techniques used to detect each of the different attacks.

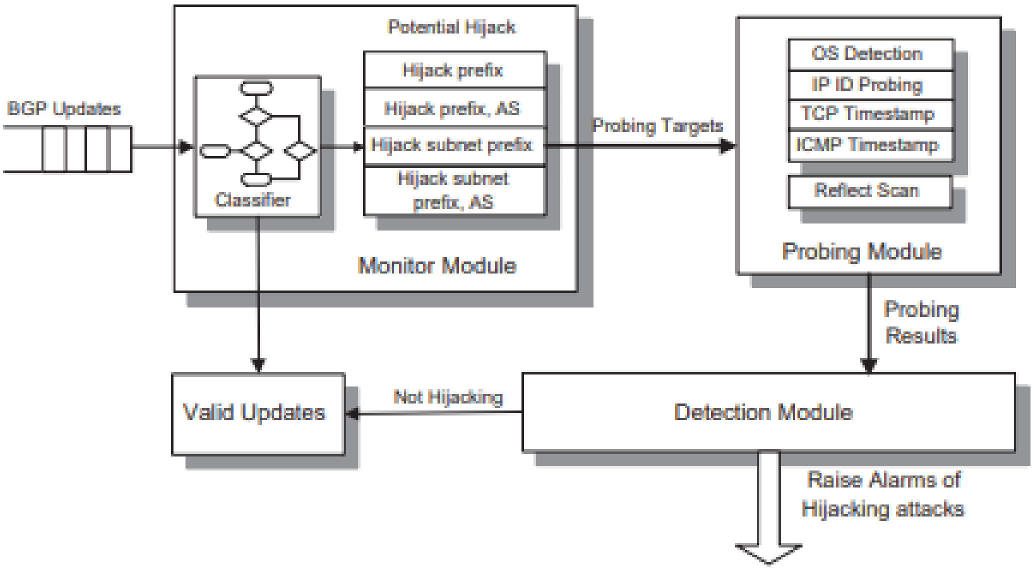


Figure 7.5: a graphical overview of how Hu-Mao is to be implemented. Source: Hu et al., 2006.

The AMM collects live updates from BGPmon [149] and uses it to check whether or not the update is anomalous. One thing to note is that BGPmon can alert network operators to prefix hijacks already, as that is a feature of BGPmon [6]. The authors of the paper detailing Argus do not seem to make use of this feature of BGPmon, and it is not completely clear why. One reason for this might be that the AMM in Argus distinguishes several anomalies, whereas BGPmon only reports prefix hijacks. The AMM discovers three types of anomalies:

1. Origin anomaly: the origin AS of a received path is anomalous, indicating a false origin.
2. Neighbour anomaly: an adjacent pair of ASes in the path is anomalous, indicating a false link.
3. Policy anomaly: an adjacent AS triple in the path is anomalous, indicating a breach of policy. The system does not care for the specific breach of policy.

An anomaly is defined as either an origin, a path AS pair, or a path AS triple not having appeared in the routing information database before. To assist the AMM, Argus also makes use of the LCM to gather recent live IPs in every routable prefix. This collects IP addresses by collecting every valid IP address x.x.x.1 in a prefix (for example: if the prefix is 123.123.128.0/23, then the set of every valid IP address x.x.x.1 in the prefix is the set containing 123.123.128.1 and 123.123.129.1), as well as the IP addresses from the traceroute paths in the iPlane daily result. If there is both a prefix abnormality and there are live IP addresses in that prefix, then there is most likely a hijack happening.

When the AMM detects an anomaly, Argus will use the HIM to activate the identification process to classify that anomaly. It will then launch both control plane and data plane probing mechanisms. Control plane based methods are real-time and contain a lot more information than data plane based methods, while data plane based methods are more accurate. This is because control plane based methods have trouble dealing with legitimate changes in network topology, while data plane based methods need to probe a large number of network continuously and as such introduce a lot more overhead.

Control plane probing is done by querying the control plane using m threads called C-threads, which continuously gather BGP routes for prefix P. In second t after an anomaly occurred, all m C-threads will construct one vector called $C_t = \{c_{t,j} | 1 \leq j \leq m\}$, with $c_{t,j}$ being 0 if the best BGP route $r_{t,j}$ for P in the j-th eye at time t contains the anomaly, and 1 if it does not. Data plane probing is done at the same time and uses m threads called D-threads. These will be used to construct a vector $D_t = \{d_{t,j} | 1 \leq j \leq m\}$, with $d_{t,j}$ being 1 if the

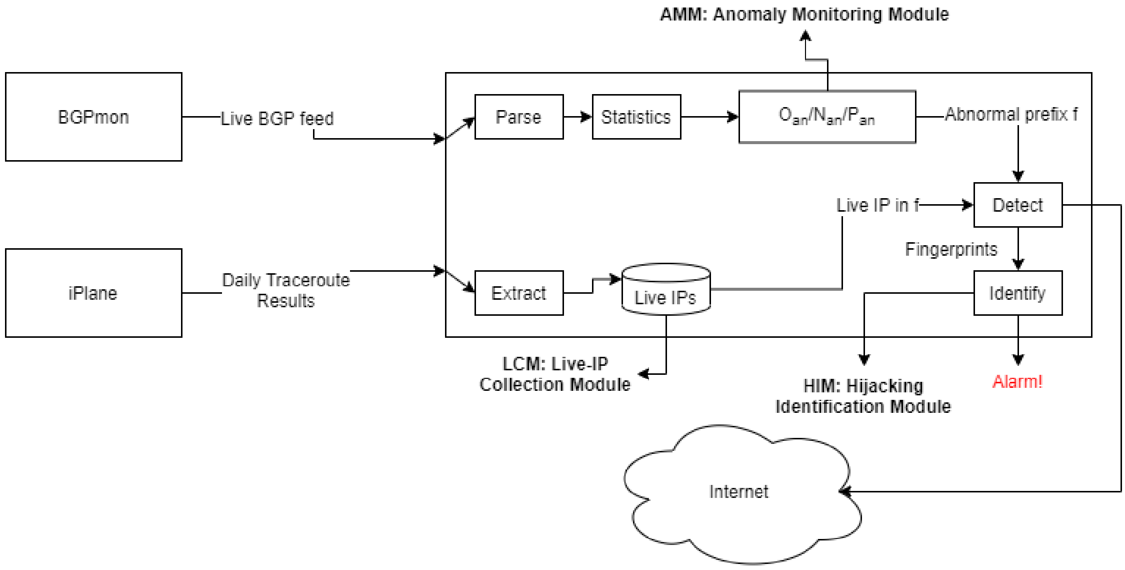


Figure 7.6: overview of the architecture of Argus. Source: Xiang et al., 2011.

probe to P got a reply, and 0 otherwise. Afterwards, the correlation between the two vectors is calculated. If both vectors are positively correlated, then that means that there is a correlation between routes containing an anomaly and no reply from the probe being received. This correlation indicates a possible prefix hijack. As such, the correlation coefficient F_t between them is calculated as follows, using $E(C_t)$ and $E(D_t)$ as the averages of C_t and D_t respectively:

$$F_t = \frac{\sum_{j=1}^{m}(c_{t,j} - E(C_t))(d_{t,j} - E(D_t))}{\sqrt{\sum_{j=1}^{m}(c_{t,j} - E(C_t))^2 \times \sum_{j=1}^{m}(d_{t,j} - E(D_t))^2}}$$

If this correlation coefficient is sufficiently high, then that is a strong indication of a prefix hijack. The authors suggest a threshold μ of 0.6, with a higher threshold achieving a lower false-positive rate but a higher false-negative rate. In general, in the field of cybersecurity, we want the false-positive rate and the false-negative rate to be balanced, as our security needs to do its job without warranting too many unnecessary investigations due to false positives.

7.7. ARTEMIS

ARTEMIS is a recently suggested detection system, proposed by Sermpezis et al. in 2018 [131]. It is a control-plane based monitoring system that makes use of two observations: the fact that BGP monitoring is far more advanced compared to the time when previous solutions for hijack detection were developed, and the system shifts from a third-party service to a first-party self-deployed system.

ARTEMIS collects BGP data from several sources. These sources are BGPmon [149], RIPE's Route Information System (RIS) [38] and RouteViews [128]. The system itself can run locally and, as such, enables the detection of hijacking events for its prefixes. Aside from the incoming BGP data, it also uses a local configuration file with information on the prefixes that are owned by the network. This information includes lists of owned ASns and prefixes, ASNs of neighbouring ASes, and local routing policies. This file can be updated automatically, which considering the changing nature of the Internet topology at the AS level, is beneficial.

To detect a subprefix hijack, the network operator stores all owned and announced prefixes in its local configuration file. In case of a subprefix hijack happening, the BGP messages used for the hijack attempt are seen by the monitoring services. Then ARTEMIS can immediately classify them as a hijack because the prefix owner did not announce this subprefix, making detection trivial as the configuration file does not contain an announcement for this subprefix. For detecting exact prefix hijacking, we first need to introduce the two different cases with regards to hijacking: type-0 prefix hijacking and type-n prefix hijacking:

- Type-0 prefix hijacking: a hijacking AS AS_x claims to own a prefix which is owned by a different AS.
- Type-n prefix hijacking: a hijacking AS AS_x alters the path to a prefix that it attempts to hack by inserting itself into the path to the prefix, creating a fake link. Here, the position of the rightmost fake link determines the number to replace n: if the normal path is (AS_4, AS_3, AS_2, AS_1) with AS_1 being the legitimate owner of the prefix, then the path $(AS_4, AS_3, AS_2, AS_x, AS_1)$ would be a type-1 prefix hijack, while $(AS_4, AS_x, AS_3, AS_2, AS_1)$ would be a type-3 prefix hijack. Being able to detect this is useful for being able to detect path altering attacks.

Type-0 and type-1 prefix hijacks are trivially easy to detect using this scheme, as the configuration file contains not only the origin ASNs per-prefix but also a list of neighbour ASNs per prefix. Detecting type-n attacks where $n \geq 2$ is harder. For this type of detection, ARTEMIS stores locally a list of previously verified AS links, a list of AS links obtained from the monitors, and another one obtained from the BGP speaking routers in the network. Due to the shifting nature of Internet topology, these need to be updated constantly because not doing so would result in them having stale information. As such, the authors suggest using 10-month sliding windows of historical data. The detection algorithm is then triggered when a path is received with an AS link in it that has not been seen before. The downside of using this approach is that checking whether or not a link has been observed previously can result in a lot of false positives. Because of this, the authors have suggested two additional rules to check for before labelling an AS link as suspicious:

- Bi-directionality: if an AS link from AS X to AS Y has never been seen before, then verify if the reverse link, from AS Y to AS X, has been observed by this AS. If that also has never been observed, then the link is suspicious.
- Left AS intersection: if a reverse link from AS Y to AS X has been observed, then another check has to be performed: define P^{old} as the set of all the paths containing this link: $P^{old} = \{P | P = (L_P, AS_Y, AS_X, R_P)\}$, with L_P and R_P being path segments at the left and the right side of the reverse link respectively. Then, define L^{old} as $L^{old} = \{L_P | P \in P^{old}\}$. Calculate the intersection of all these path segments. If the intersection is not an empty set and at least one AS in this intersection also appears on the left side of the path with the reverse link, then it is classified as a suspicious reverse link. If not, then it is classified as a legitimate link.

ARTEMIS also provides a mitigation approach. The detection module provides for each detected event the affected prefixes, the type of hijacking event, the observed impact, the ASNs of the ASes that were involved, and the reliability of the detection. The mitigation techniques proposed by the authors are self-operated mitigation with prefix deaggregation and outsourcing mitigation with MOAS announcements. The first one involves announcing deaggregating the owned prefix and announcing more specific versions of the prefix, effectively performing a counter-subprefix hijack. The second one is based on DDoS mitigation services and involves a third party announcing the hijacked prefix, attracting traffic from parts of the Internet, and tunnelling it to the original host. This approach to risk management is one that I have not seen before in BGP security; in risk management, there are four general ways to manage risk [57]:

1. Avoiding risk altogether.
2. Reducing risk.
3. Accepting risk.
4. Transferring risk.

Preventative BGP security tends to fall into the first category as it attempts to eliminate most or all sources of risk in BGP. Detective BGP security tends to fall in the second category, as detection tends to be imperfect, and misclassifications can happen. This approach falls into the final category because the risk of an attack happening has been transferred to a third party which channels information back to the AS.

The authors use BGP simulations to measure how well outsourcing performs as a mitigation measure. These simulations show that outsourcing to one or more organizations is very effective in mitigating the impact of hijacking, reducing the impact of a hijack to a prefix owner by at least a third for type-0 and type-1 prefix hijacks if it is outsourced to just one randomly selected organization. The traffic is then sent from the other organization to the original host.

7.8. aPHD

Zhang et al. proposed a prefix hijack detection method called aPHD fairly recently, in September 2019 [153]. This method takes inspiration from how the human immune system works. Researchers have simulated this immune system and created the artificial immune system (AIS) [152]. This prefix hijack detection scheme is similar to AIS in that it is made up of many independent objects and that the goal is to secure the system by detecting intrusions. The design goals of this system are to have low computational overhead, high accuracy, and a low error rate.

aPHD trains its detectors on BGP data in a process called evidence collection and uses it in an immune model for prefix hijack detection. This evidence collection gathers both UPDATE messages and events. From the UPDATE messages, it extracts information such as the prefix, the prefix length, and route attributes. These are then combined to form a binary string, and these binary strings are added to a set called the antigen set Ag. For the event gathering, the system crawls the BGPstream Twitter page for attack events. The system then gathers the same info as it does from the UPDATE messages, and also transforms it into a binary string. These binary strings are also combined into a set.

Given the definition of the antigen set, the immune model works as follows: define two sets $Self$ and $Nonself$, which is a partition of Ag into two sets: $Self \cap Nonself = \emptyset, Self \cup Nonself = Ag$. $Self$ changes over time, removing bitstrings that have not been used as often while adding new ones.

There is also a set of detectors D used as antibodies, which are randomly generated bitstrings with a period, age, and count. The set of detectors D is then split into the set of immature detectors I, mature detectors M, and memory detectors E. These sets also change over time, with immature detectors evolving into mature detectors if they are not matched to a bitstring in $Self$ during the tolerance period, screened out during self-tolerance, and new immature detectors being randomly generated. The self-tolerance function is as follows: $f_t(I) = I - \{d|d \in I \wedge \exists x \in Self \wedge f_m(d,x) = 1\}$, with $f_m(d,x)$ being the matching function which returns 1 if a string of at least r bits is shared between the two inputs, with r being a matching threshold. Removing these immature detectors that match with strings in $Self$ should decrease the false positive rate, as will be explained shortly. Mature detectors change over time by immature detectors maturing into mature ones, creating new mature detectors, while old ones die due to reaching their maximum age or change into memory detectors as a certain count is reached. E dynamically evolves by adding new strings from M and deleting strings that have been matched with $Nonself$.

The attack detection flowchart using this scheme is shown in figure 7.7. From the flowchart, we can see that, if memory or mature detectors detect prefix hijacking attacks because the memory detectors or mature detectors match with bitstrings corresponding to UPDATE messages corresponding to these types of attacks. There is no alarm triggered when a bitstring is matched with an immature detector because these detectors are randomly generated and as such would not be the best indication of a prefix hijack happening.

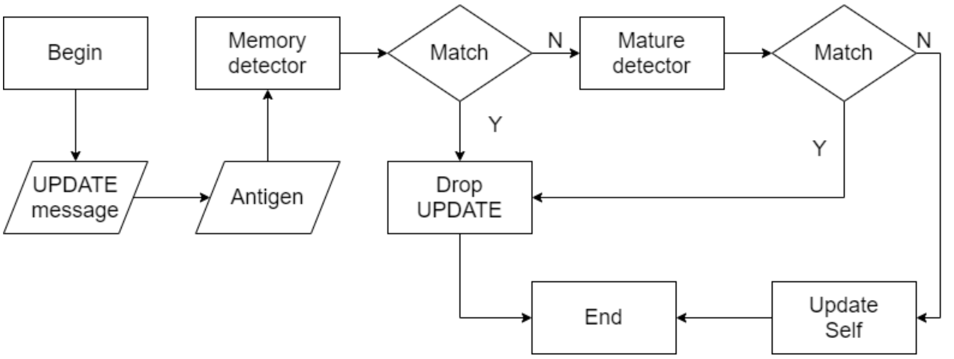


Figure 7.7: the attack detection flowchart used in aPHD. Source: Zhang et al., 2019.

8 BGP detection algorithm analysis

Several detection schemes have been presented in Chapter 7. To see what they have in common and how they could be improved to provide a new step forward for BGP security, a new taxonomy focussing on detection algorithms and their features should be made. Also, a list of ideal requirements should be made to guide future research on BGP detection techniques. The goal of this chapter is to answer the third research subquestion: what can be done in the future of BGP security?

8.1. Approach

The approach to compare BGP detection algorithms is similar to the approach used to compare security solutions: once again, I create taxonomies that compare the detection algorithms based on several criteria, and get some insight into which ones are better for deployment in the real world and how they could be improved even further.

There has been some prior academic work on creating taxonomies for BGP detection algorithms. However, there is less prior work compared to work done on the topic of comparing BGP security solutions, which means that these categories are more improvised. This improvisation means that the categories are based more on logic and what follows from the papers describing the different detection algorithms, rather than what has been done already in comparing this kind of BGP security. The lack of more prior academic work can most likely be attributed to BGP detection being relatively new. Musawi et al. published a survey in 2016 comparing several BGP anomaly detection techniques [116]. This survey also includes detection algorithms that focus not specifically on BGP attacks, but on general anomalies happening in Internet routing as well, such as link failures. This thesis focuses exclusively on the detection of BGP attacks. The classification does provide some basis for the taxonomy presented in this thesis, but it has the following shortcomings:

- It lists the different data sources that the different detection schemes use but does not compare methods based on the different data sources that they use. Using different kinds of data sources can result in more accurate classification at the cost of some additional computational overhead. The different data sources are also very specific instead of grouped into a general category. That makes comparing them even harder.
- It compares methods on which plane(s) the detection schemes use(s) but does not mention that using either one of them has benefits and drawbacks, as mentioned in the paper presenting Argus [146]. A "plane" is a conceptual model in network engineering that intends to divide different kinds of networking-related tasks [37]. The data plane is responsible for handling data that passes through your network, like packets that are sent and received. The control plane sets policies for how the packets traverse the data plane, and as such, contains routing information. So control plane-related information would be information regarding routing, such as stored routes and BGP updates. In contrast, data plane-related information would be information regarding the reachability of a certain host, such as network probe results.
- The authors state that no detection scheme has all the features that they consider ideal, and as such, no detection scheme is perfect right now. Thus they conclude that more work needs to be done on the

subject of BGP hijack detection. That is not a shortcoming on its own, but the authors do not mention the possibility of combining parts of two or more detection schemes into one detection scheme that has all or most of the ideal properties. They mention that something brand new has to be developed, effectively throwing a lot of research out of the window instead of looking at what has been done and trying to find something that can be useful.

The goal of the taxonomies in this chapter is to find out if there is a possibility of a combination detection scheme being able to achieve a set of ideal properties. Instead of concluding that no detection scheme is perfect and that a new one has to be made, I want to look at what has been done already, just like with the preventative security solutions. Sriram et al. suggested something similar in 2009 [136]. They suggested to enhance PGBGP with security enhancements that are based on registry (e.g. RIR, LIR, etc.) improvements [132]. The paper is quite old however and many new detection schemes have been developed, so it is worthwhile to see if there is an even better combination possible. The goal here is to see how existing research can be put to better use. To do so, the taxonomy is designed to compare not only detection schemes on what kind of methods they use but also the different kinds of data that they use to perform their detection.

The main difference between this approach and the approach developed for preventative BGP security is that there is no taxonomy detailing whether or not a detection scheme detects different kinds of attacks. The reason for this is as follows: the BGP detection algorithms discussed exclusively focus on the detection of (sub)prefix hijacks. Some of the schemes have elements that could potentially be used for detecting false path altering, but no scheme provides this detection. As such, it would not make sense to compare detection schemes on what kind of attacks they detect, because they are all designed to detect the same kind of attacks.

8.2. What requirements would a combination scheme ideally fulfill?

No detection algorithm currently is perfect, but considering what is beneficial to a detection algorithm for proper functioning and what is hindering, a good detection algorithm would satisfy some requirements. These "must have" requirements are based on the current best properties of the discussed detection schemes, as well as on preventing problems that have been witnessed in the proposal of prevention-based BGP security solutions. The requirements are:

- The detection scheme makes use of both data plane-based and control based methods to perform detection. The paper describing Argus states that using methods that gather information from both planes is ideal because data plane methods are more accurate while control plane methods are real-time [146].

- The detection scheme needs to be able to run on its own in the AS, instead of it being deployed as a service. If the detection scheme were to be deployed as a service that network operators could register their AS to, then it would lead to a reliance on a service being online to function. That would, in turn, create a similar single-point-of-failure problem that using a PKI would create for prevention-based security solutions.

- The detection scheme should only use a historic window of the most recent data due to the changing nature of Internet topology. Data regarding which ASes own which prefixes that was true five years ago might not be true anymore, because of (for example) prefix transfers. Recent data is defined in this context as BGP data from up to a year ago. The reasoning for this cut-off point is that the detection schemes that suggest using a sliding window of data suggest having a sliding window size of up to a year.

- The detection scheme does not solely rely on an external BGP feed to function. It can use an external feed to increase the accuracy of its detections, but it should be able to function without one as well. Relying on an external data source would create the same problems that deploying the detection scheme as a service would. It would create a single point of failure.

- The detection scheme uses as many different data sources and techniques as possible without any redundancy in the data or techniques used. Using multiple data sources and techniques tends to lead to more accurate classification at the cost of some processing power. However, having to process redundant data or executing redundant processes generally introduces unnecessary computational overhead. Delays in classification due to using more data or processes should also be taken into account, but most delays can be solved by using faster hardware.

In addition to these requirements, there are some other "should have" requirements that a combination scheme should ideally fulfill, but the taxonomies will not include these factors in comparing the presented detection algorithms because the data is not available or the property is out of scope for detective security. These requirements are:

- The detection scheme ideally provides automated countermeasures in the case of a potential prefix hijack occurring, if the hijacked prefix is the prefix that the scheme-deploying AS owns. Providing countermeasures would mean that mitigating prefix hijacks does not have to be done by a network operator. This requirement is normally out of scope for detective security as it falls more into the category of reactive security.
- The detection scheme ideally provides real-time detection. (Near) real-time detection is important as a BGP hijack can cause havoc within ten minutes, as briefly mentioned by Musawi et al. [116], who modelled such a redirection happening and the impact that such a redirect had on network traffic, and backed up by real-life attacks such as the Pakistan Telecom incident, which show that redirection happens within minutes of an attack being successful [51]. This is according to simulations of attacks that the authors have performed. (Near) real-time detection of BGP hijacks was not included as a category in any taxonomy as it is not as clear for some of the detection schemes if they provide this detection speed or not. Musawi et al. do include this feature in their taxonomy, but for some of the discussed detection schemes in that paper, the outcome for "provides (near) real-time detection" is "unknown".

8.3. Creating the taxonomies

Just as with the BGP security solutions taxonomies, several taxonomies will be created to compare the various detection algorithms outlined in this thesis. The intent is to see which schemes would be good to combine to get a combination scheme that fulfills the requirements listed in section 8.2. To do this, four different taxonomies will be used to compare the BGP detection schemes presented in this thesis, and they are:

- Plane: algorithms are compared on if they use control plane methods or data plane methods for their classification. These have their pros and cons, as Xiang et al. outlined when they proposed Argus [146]. So it is worth it to see which detection algorithms use a combination of data plane based methods and control plane based methods.
- Information: what kind of information the detection algorithms use. Instead of listing which specific information sources each detection scheme uses, the taxonomy uses categories of information sources. That makes comparisons easier. An ideal combination of schemes has little overlap in the different sources, as overlapping categories of data sources can cause processing of redundant data. So it is useful to see the different categories of information that the different detection schemes use, in order to see which (parts of) schemes could be combined without creating significant overlap in information sources. The used information sources are partially dependent on what methods the detection schemes use, and more specifically, whether the detection methods are data plane based or control plane based. They are not completely correlated however, as will be shown soon.
- Techniques: which techniques are used by the algorithms to classify whether or not a BGP UPDATE message is legitimate or part of an attack. This taxonomy is based on the classification scheme from previous academic work. An ideal combination of schemes has few overlapping detection methods as overlap in detection methods can cause extra processing overhead without providing more accurate results. So analysing which techniques each scheme uses is useful for determining which schemes should be combined.
- Features: what kind of beneficial features the detection algorithm has built-in. A combination of ideal properties is the best for a new detection scheme, so it is nice to know which beneficial features each of the schemes have.

8.3.1. Plane taxonomy

The first taxonomy that will be presented is the plane taxonomy. The descriptions of the detection algorithms sometimes already stated which plane of information the algorithms made use of, and this taxonomy reiterates these statements for comparison's sake. This one is relatively simple as there are only two planes that are used in detection schemes, and as such the taxonomy has only two categories:

- Data plane: whether or not the detection algorithm uses methods that are based on the data plane. To recap, the data plane is the network plane that deals with packets passing through the network, and as such, these methods tend to be traceroutes and pings. In the context of BGP hijack detection, these methods focus on the reachability of certain destinations instead of figuring out the routes to those destinations. Data plane based methods tend to be more accurate because they need to probe a larger portion of the network to get their results, but this comes at the cost of increased overhead due to the amount of probes needed.

- Control plane: whether or not the algorithm uses methods that are based on the control plane. As mentioned before, the control plane is the network plane that deals with routing and how to send packets from one point in the network to another. As such, these methods in the context of BGP hijack detection rely on parsing UPDATE messages to get routes from there and focus on routes over reachability. Control plane based methods tend to be faster, often real-time, and contain more information as the information is drawn directly from the BGP messages (and as such does not require probes to be sent and received before a classification can be made), but this comes at the cost of accuracy as these methods have a hard time distinguishing legitimate network changes from hijacks.

The taxonomy can be seen in table 8.1. Because using methods from either plane has benefits and drawbacks (as mentioned in section 7.6, as the authors of Argus have detailed the benefits and drawbacks of using both), an ideal solution would use a combination of both. Argus and Hu-Mao are the only ones to do so currently, and the paper presenting Argus is also the only one that has mentioned there being pros and cons to using methods from either plane exclusively.

Plane	Data	Control
PGBGP and IAR	No	Yes
PHAS	No	Yes
Zheng-Ji-Pei-Wang-Francis	Yes	No
iSPY	Yes	No
Hu-Mao	Yes	Yes
Argus	Yes	Yes
ARTEMIS	No	Yes
aPHD	No	Yes

Table 8.1: The plane taxonomy.

8.3.2. Information taxonomy

The next taxonomy to be presented is the information taxonomy. The different kinds of information that each scheme uses has been described in Chapter 7 already, and this taxonomy puts the kinds of information in three generalized categories. The categories are each designed to reflect the different types of information that could be useful for the detection of hijacking events occurring, based on both the discussed detection methods and the discussed prevention methods. They also reflect what has been used in various proposed detection schemes already. The categories are:

- Host information: whether the detection algorithm makes use of information about the host in deciding if the message is a hijack or not. This can be the prefix that the AS that the host belongs to owns, but it could also be information such as host OS, the port used, etc. Changes to the host OS or even the prefix that the host owns rarely happen legitimately, so keeping track of these is useful for detecting hijacks.

- Path information: whether the detection algorithm makes use of information inferred from paths to a prefix in deciding if the message is a hijack or not. This category includes information beyond what the paths say, and can (for example) also be inter-AS relationships such as customer-provider relationships inferred from paths. Though legitimate path changes happen more regularly than legitimate changes in a host because of link failures and power outages, big differences in paths can still indicate hijacks, and as such it is useful to look into which schemes use this information.

- Reachability information: whether or not the detection algorithm uses data regarding the reachability of certain hosts to classify UPDATE messages. If a certain AS is not reachable any more, then this can mean that the AS is down or that the reachability probe has not been returned. If a large part of previously reachable ASes are suddenly not reachable any more, then that is a good indicator of a prefix hijack (as seen with the method that iSPY uses). As such, this information can be used to accurately determine whether or not a prefix hijack has happened.

The taxonomy is presented in table 8.2. As can be seen, most of the detection schemes use more than one category of information. This is most likely to increase the detection accuracy, as using more data sources for classification tends to lead to more accurate classifications. Before making the taxonomy, I expected there to be a clear divide between the methods used and the information used. Detection algorithms that made use of control plane methods were expected to make use of path information. In contrast, detection algorithms that made use of data plane methods were expected to make use of the other information sources. This expectation is not completely true: Zheng-Ji-Pei-Wang-Francis, for example, makes use of path information without using control plane methods at all. That is because the detection algorithm uses traceroute (a data plane method) to gather paths from host to host, and then translate these paths to BGP paths using lookup methods.

In detection algorithms, it is important to have some sort of trade-off between the amount of data used and the accuracy of detections, because more data tends to mean more computational overhead. In comparison, fewer data will lead to less accurate classification. On the other hand, the problem of having too little computational power can be solved using faster processors or possibly parallel computing. In that case, the main problem with using too much data would probably be redundancy in the data. That would sacrifice computational power (because redundant data needs to be processed) while not gaining more accurate classifications (because no new data that could help classification is used).

Information	Host information	Path information	Reachability information
PGBGP and IAR	Yes	Yes	No
PHAS	Yes	No	No
Zheng-Ji-Pei-Wang-Francis	No	Yes	Yes
iSPY	No	No	Yes
Hu-Mao	Yes	Yes	No
Argus	Yes	Yes	Yes
ARTEMIS	Yes	Yes	No
aPHD	Yes	No	No

Table 8.2: The information taxonomy.

8.3.3. Techniques taxonomy

The next taxonomy is the techniques taxonomy. This taxonomy deals with generalized categories of classification techniques that are used by the algorithms to detect whether or not a hijack is occurring. These general categories are based on common trends seen in the techniques that have been used by existing detection schemes that are presented in this thesis, and they are:

- Fingerprinting: whether or not the detection algorithm makes use of fingerprinting techniques to identify ASes and classify UPDATE messages as benign or malicious. Fingerprinting techniques can be device fingerprinting techniques, which involve collecting information about the hardware and software of a certain device for the purpose of identifying said device. But also, specifically for BGP security, it includes which AS owns which prefix, and previously received BGP paths. As mentioned before, these kinds of changes rarely happen legitimately and therefore are a strong indicator of a hijack occurring. So it is useful to see which schemes use this kind of technique.

- Statistical analysis: whether or not the detection algorithm makes use of statistical analysis of information gathered from the plane(s), such as moving averages over time, to detect that a BGP hijack is happening. Using statistical analysis can be a good way to detect hijacks as sudden spikes in statistics often indicate anomalies happening. It is also not that computationally expensive to use statistical analysis most of the time, as oftentimes, only a small amount of data (a single moving average, or sev-

Techniques	Fingerprinting	Statistical analysis	Reachability probing
PGBGP and IAR	Yes	No	No
PHAS	Yes	No	No
Zheng-Ji-Pei-Wang-Francis	No	Yes	Yes
iSPY	No	No	Yes
Hu-Mao	Yes	Yes	No
Argus	Yes	No	Yes
ARTEMIS	Yes	No	No
aPHD	Yes	No	No

Table 8.3: The techniques taxonomy.

eral different averages, for example) has to be stored to perform this kind of analysis. So using these kinds of techniques can be effective and lightweight at the same time.

- Reachability probing: whether or not the detection algorithm makes use of reachability probing techniques such as traceroute and ping to detect if a BGP hijack is happening. As mentioned in the reachability information category, big changes between old results of reachability probes and new results of those probes tend to be a strong indicator of a successful hijack, so it is handy to see which of the schemes use this method.

The taxonomy is shown in table 8.3. As can be seen in this taxonomy, most of the detection schemes use only one kind of technique. This was already alluded to in Section 7.5. Also, there is no detection scheme that uses techniques from all three different categories. So, to satisfy the requirement of using as many different kinds of techniques as possible without redundancy in the kinds of techniques used, a combination of techniques from different schemes is required.

There are some relationships between this taxonomy and the data taxonomy. For example, if the detection algorithm uses statistical analysis, then the detection algorithm uses path information for detecting whether or not a prefix hijack is happening. This is because many statistics are inferred from paths, such as edge popularity and inter-AS relationships in Hu-Mao. Also, schemes that make use of fingerprinting techniques use host information and vice versa, and schemes that make use of reachability probing techniques make use of reachability data and vice versa. The former is logical because much of the information that would be collected in fingerprinting would relate to the host or the AS (for example: the fingerprinting techniques used by Hu-Mao collect data such as host OS and TCP timestamps, PHAS collects data on which prefix is owned by which AS, et cetera). The latter is logical because the most straightforward way to gather reachability information would be through reachability probing. It would in theory be possible to gather reachability information from collected paths, by analyzing differences in paths over time, but there hasn't been a scheme that is discussed in this thesis that uses this way of gathering reachability information. This way would also be a slightly less accurate way of gathering reachability info because legitimate changes in network topology can cause a false positive, as mentioned in Section 7.6. Argus does come close with its C-threads, but those are more intended to cross-validate the reachability probe results that are gathered by the D-threads.

Before making this taxonomy, I expected algorithms that made use of data plane methods to make use of fingerprinting, as gathering some of the information necessary for good fingerprinting (ports used, host OS, etc.) is done by methods that use the data plane, by using, for example, Nmap [31]. However, one can see that there are algorithms that make use of data plane methods that do not make use of fingerprinting as a detection technique, and that there are algorithms that make use of fingerprinting that do not make use of data plane methods. However, if a method uses reachability probing, then it uses data plane based methods, which is logical considering reachability probing is generally done using tools such as traceroute, whois, and ping, which are all data plane based methods.

8.3.4. Features taxonomy

The final taxonomy is the features taxonomy. This taxonomy deals with the different beneficial features that each of the taxonomies have. For determining which of the schemes are going to be combined, it is nice to know which kinds of beneficial features they have. The categories of features are as follows:

Features	Self-deployable system	Only uses recent data	Does not require external BGP feed
PGBGP and IAR	Yes	Yes	Yes
PHAS	No	No	No
Zheng-Ji-Pei-Wang-Francis	No	Yes	No
iSPY	Yes	No	Yes
Hu-Mao	Yes	No	Yes
Argus	Yes	No	No
ARTEMIS	Yes	Yes	No
aPHD	Yes	Yes	No

Table 8.4: The features taxonomy.

- Self-deployable system: whether the detection algorithm is deployable as a system within the AS or not. If not, then it exists as a separately run service that other autonomous systems can make use of. This is a beneficial feature because, ideally, a detection algorithm can function within an AS on its own because it removes the reliance on a service being online.

- Only uses recent data: whether or not the detection algorithm makes use of old data as well as new data to perform detection or only a window of more recent data. Recent data is defined in this context as BGP data from up to a year ago, to be consistent with the criterium listed in section 8.2. This is a beneficial feature because, due to the shifting nature of Internet topology, historical data tends to be irrelevant after a long time.

- Does not require external BGP feed: whether or not the detection algorithm does not have to make use of a BGP monitoring service, such as BGPmon or iPlane (to name some that we have seen in the descriptions of the detection algorithms that are in this paper). This also includes BGP monitoring services that are used specifically by that algorithm. If a detection algorithm uses an external BGP feed **in addition to** the BGP feed that it already receives from its BGP speakers to (for example) make more accurate classifications, then it does not require an external feed to function properly. This is a beneficial feature as a detection algorithm that has to make use of external BGP feeds can be crippled or completely defunct when these feeds are offline.

The features taxonomy can be seen in table 8.4. Before making the taxonomy, I assumed that detection of prefix hijacks happening outside of the AS that the system is deployed in would only be possible if the system uses an external BGP feed because using an external BGP feed would provide data about other parts of the Internet topology which is oftentimes necessary for this kind of detection. In other words, the detection of other hijacked prefixes implies that the system uses an external BGP feed. However, Hu-Mao shows that this does not have to be the case, as it detects other prefix hijacks as well without using an external BGP feed.

8.4. What can we learn from the taxonomies?

From the taxonomies and also from the descriptions of the detection algorithms themselves, we can see that there is a wide variety in how BGP hijacks are detected. These options can have beneficial or hindering elements.

From the taxonomies in section 8.3 and the requirements listed in section 8.2, an ideal solution would most likely be a combination of Hu-Mao, ARTEMIS and Argus, using the best elements of all three in a single solution. The combination would then probably consider of using the techniques that Hu-Mao and Argus use for data collection while using the detection techniques that ARTEMIS employs.

8.4.1. Why choose these solutions in particular?

There are several reasons for choosing these solutions over the others. The reasons can be summarized as follows:

For data collection, Hu-Mao and Argus are both good as they use both data plane and control plane methods. Argus unfortunately requires not one but two external BGP feeds to function properly, which means that Hu-Mao is also good to use because it uses incoming BGP UPDATE messages instead of relying on a external BGP data source. Hu-Mao uses a multitude of techniques that cover two of the main categories of techniques (fingerprinting and statistical analysis) whereas Argus provides reachability probing by using the C- and D-threads. So combined, they provide the ability to use all different categories of information and techniques. The main problem of these two methods is that they do not have a cut-off point in their specification for when old data is not meant to be used any more.

For detection, ARTEMIS is the detection algorithm that provides a method that can be used in the future to detect illegitimate path alterations through its detection of type-*n* prefix hijacks, which is an area of BGP attack detection that is still new. There are more benefits to using ARTEMIS, which are:

- It can be configured to not use data that is older than a specific period of time, and this period of time can be specified by the user. The authors presenting ARTEMIS suggest this period of time to be ten months.
- It also provides near real-time detection, being able to detect attacks within seconds of an attacker making an announcement. The importance of real-time detection is explained in Section 8.2.
- It provides reactive security in the form of providing measures to counteract a prefix hijack. It is also the only detection scheme to do so as far as I have found. Even though this is a property that is more associated with reactive security instead of detective security, it is a nice property to have.

The extra benefits that ARTEMIS provides also satisfy the two "should have" requirements outlined in section 8.2. ARTEMIS on its own does not satisfy all of the "must have" requirements, however. This is why using parts of other schemes is ideal for improving ARTEMIS. Combining ARTEMIS with some properties of Hu-Mao and Argus can give the following benefits:

- Using the BGP data that is received by the AS, like Hu-Mao does, would result in the scheme not being completely reliant on the availability of an external data source.
- Argus provides ARTEMIS reachability information, which is a information category that ARTEMIS does not use. ARTEMIS uses information that falls into the categories of host information and path information. With regards to host information, it could be worthwhile to also use the information that Hu-Mao uses, as Hu-Mao mainly collects information regarding the host system, whereas ARTEMIS mainly collects information regarding the origins that each AS owns. ARTEMIS, Hu-Mao and Argus also use path information differently: Hu-Mao uses it to infer relationships between ASes for example, whereas ARTEMIS stores known paths to compare incoming paths with stored ones, and Argus uses it to probe. So they also collect different kinds of host and path information.
- ARTEMIS only uses fingerprinting, while some of the techniques in Hu-Mao (such as edge popularity checking) can provide statistical analysis and the C- and D-threads of Argus can provide reachability probing. This would lead to the combination scheme satisfying the requirement of using multiple techniques without having redundancy in the techniques used.
- ARTEMIS only uses control plane based methods for its detection, which cannot handle legitimate network changes that well. Hu-Mao and Argus can provide additional data plane based methods for detecting anomalies.

8.4.2. Why not use any of the other presented solutions in the combination scheme?

There are several reasons for combining Hu-Mao, Argus and ARTEMIS into a detection scheme, but this alone is not enough to justify choosing these two schemes to combine over the other schemes that have been presented in this thesis. As such, here are reasons for why the other schemes are not good alternatives:

- PGBGP and IAR: although this scheme does have all of the beneficial features as seen in the features taxonomy, it can also cause BGP convergence delay by deciding to not adopt new routes until hours or even days after they have been first announced. This in turn hinders the user’s Internet access.

- PHAS: this scheme is set up as a service that one can subscribe to. This causes a reliance on the availability of the service. It also does not provide (near) real-time detection as seen in Section 7.2.

- Zheng-Ji-Pei-Wang-Francis: this scheme is also set up as a service that one can subscribe to, which causes a reliance on the availability of the service.

- iSPY: unclear whether or not the method provides real-time detection, this depends on how often probes are launched to check for any cuts in the network. Argus also already provides a similar approach for detecting hijacks using their C- and D-threads, and uses this approach whenever the AMM of Argus (which is connected to BGPmon, which in turn provides real-time information) detects an anomaly.

- aPHD: relies on BGPstream, and because of the way that it translates events to bitstrings (whereas the other methods presented in the thesis do not translate any of their gathered information to bitstrings or any other format), it is very hard to combine this with any of the other methods for more accurate detection.

It should be noted that a combination of Zheng-Ji-Pei-Wang-Francis with PHAS would satisfy the requirement of using as many different kinds of information/techniques without redundancy. This is because they use complementary information/techniques. However, both schemes are set up as services that one can subscribe to, and their detection techniques require external BGP feeds (Zheng-Ji-Pei-Wang-Francis even requires multiple BGP feeds to function), which makes such a combination less than ideal.

8.4.3. What would such a combination of two schemes look like?

Now that the ideal scheme to be developed next has been established to be a combination of properties from ARTEMIS, Hu-Mao and Argus, the next step is to come up with a list of what this combination scheme should include. This list should meet the requirements listed in section 8.2. To that end, I have come up with the following list:

- The scheme should be deployable on its own in an AS. This should not be that much of a problem considering Hu-Mao, ARTEMIS and Argus are deployable as standalone systems.

- The scheme should throw away data if the data is older than a year at most. The authors of the paper proposing ARTEMIS already suggest not using data that is older than ten months. This implies that ARTEMIS already has some mechanism built into it that allows it to ignore data that is older than a user-specified amount of time.

- The scheme should use the device fingerprinting techniques (such as host OS properties, IP ID probing), techniques based on inferring inter-AS relationships, the edge popularity data and the C- and D-threads that are presented in the papers describing Hu-Mao and Argus. The reason for this is that fingerprinting can be used to gather extra information about hosts in a certain AS to detect (sub)prefix hijacks, whereas inter-AS relationship checking, edge popularity and the usage of C- and D-threads is useful for detecting type-*n* hijacks more accurately. This would also mean that the combination scheme satisfies the requirement of using both data plane based methods and control plane based methods. It would also mean that it would use data and techniques from all different categories, satisfying the requirement of using as much data as possible without any redundancy in the data.

- The scheme should not only use BGP feeds such as BGPmon and iPlane, but also use the BGP UPDATE messages that enter the AS that hosts the scheme, similar to Hu-Mao. This is to prevent complete reliance on external data sources because reliance on external data is not beneficial. External data can, however, be used to make more accurate classifications.

- The scheme should include the reactive security measures that ARTEMIS includes, and use them whenever its prefix is hijacked. These reactive measures have been proven to be an effective countermeasure against prefix hijacks [131], and as such should be used once a hijack has been detected. Also, detection alone is not enough, as just detection does not prevent a prefix from being hijacked or an AS adopting routes towards a false location or along a malicious AS.

- The scheme should include the detection of type-0, type-n and subprefix hijacks that is present in ARTEMIS while using the extra data that device fingerprinting, inferring of inter-AS relationships and usage of C- and D-threads provide. The reason for this is the near real-time detection of hijacks present in ARTEMIS, as well as the already-high accuracy of the detection scheme. These data sources also have very little overlap, which satisfies the requirement of not using redundant data.

8.5. Conclusions on the analysis of BGP detection algorithms

To recap, the purpose of this chapter was to answer the third subquestion of the research question, that subquestion being:

What can be done in the future of BGP security?

As has been seen in Chapter 6, preventative security does not work, not only because of the additional computational overhead or the requirement of a central authority, but also because no preventative solution works when only one AS makes use of it, because they have to exchange information with one another, either directly or indirectly through a central authority. The logical next step is to have a look at detective (and reactive) security, which are relatively new fields showing a lot of promise. The requirements in this chapter have outlined some desirable properties for a good new detection scheme, and the taxonomies have helped in finding properties from already proposed detection schemes that would, when combined into a single scheme, satisfy these properties. As such, the question can be answered as follows:

Research needs to move away from preventative security to detective and reactive security. As of now, there is no detection scheme that is perfect, but Hu-Mao, Argus and ARTEMIS have properties that can be combined to make a new and better detection scheme.

9

Conclusion and future work

To recap, BGP security was and still is a big problem for the Internet, as even small attacks can pollute vast amounts of the autonomous systems that make up the Internet, redirecting large parts of the Internet to different locations than the intended destination. This thesis has investigated three different aspects of BGP security:

- The threat landscape, how it has evolved and how security solutions have evolved along with it.
- The different preventative security solutions, what they have in common, and what they all lack.
- Several detective BGP security solutions, and comparing them to see what this kind of security still lacks.

By investigating these topics, this thesis has contributed several new things to science. The contributions of this thesis are as follows:

- An up-to-date, detailed overview of the security solutions that have been developed over the years.
- A practicality-based (as in, focused on what is feasible to deploy) in-depth taxonomy of preventative security solutions. This is different from several earlier presented taxonomy, which focused mainly on the protection that these security solutions offer and whether or not they are "deployable", with "deployable" meaning that they are incrementally deployable.
- An analysis of how the BGP threat landscape evolved, using both snapshot sources (academic papers) and non-snapshot sources (Google Search hits per year and Google Trends data).
- Combining the evolution of threat landscape with the evolution of preventative security to see if there is a relationship between the security benefits offered and the changes in the threat landscape over time.
- A practicality-based taxonomy of detection algorithms. Taxonomies and surveys before this focused on the techniques used, and not so much on whether or not the techniques or the data collection processes were good or bad. The taxonomy presented in this thesis also outlines which properties of detection algorithms are better and which are worse.

From the evolution of the threat landscape, one can conclude that there is a relationship between the evolution of the threat landscape and the features of each security solution, especially with regards to the different kinds of security that the solutions offer. This has changed over time as considerations for threats have changed over time, and as such the different security solutions have removed protection against certain threats that are not considered important enough anymore, causing less extra computational overhead while still protecting against the threats that are considered important.

From the comparison between preventative BGP security solutions, one can conclude that prevention is a lost cause, but not only because of issues arising from over-centralization to routers not being able to handle the extra processing overhead, but the main reason is that ASes have to cooperate to be able to effectively use preventative security in BGP. The logical next step would then be to look at detective security because preventative security is not feasible.

From the comparison between several detective BGP security solutions, one can conclude that even though current detection schemes can already detect BGP hijacks with high accuracy, a lot more work can still be done to improve them. Some ideal properties for a new detection scheme have been proposed, and I created taxonomies to compare the detection schemes to find what combination of features from these schemes would contain these properties. This resulted in a combination of techniques and features from Hu-Mao, Argus, and ARTEMIS, using the best properties of these three while removing as many undesirable properties as possible in the process.

The main research question of this thesis was stated as follows:

Why has no BGP security solution been deployed to protect the entire Internet yet, and what can be done to protect ASes against BGP attacks in the future?

By answering the three subquestions, an answer to the main research question can be formulated. The answer to the research question is as follows:

There are various reasons why no BGP security solution has been deployed to protect the entire Internet yet even though BGP security solutions have (to a degree) adapted to cut out protection against threats that are not considered important enough any more to protect with less overhead against the threats that are still considered important. But one reason that has not been considered as of yet is the fact that preventative security on its own requires multiple ASes to deploy it because ASes need to exchange information with one another for it to work. This creates a catch-22 for preventative BGP security, and therefore it can be seen as a lost cause. As such, detective BGP security is the logical next step, and the way forward for BGP security. As of now, it seems that a combination of properties from the detection schemes Hu-Mao, Argus, and ARTEMIS is a good start for a new detection scheme.

That being said, detective BGP security has its problems as well. The main problem is that completely perfect security is most likely an impossibility, given that there will always be false positives and negatives, and new attacks against BGP are being developed all the time which could circumvent these detection mechanisms. But detective BGP security can provide substantial benefits to the AS that deploys it even if no other AS deploys this kind of BGP security (or any kind of BGP security, for that matter), and different ASes can all deploy different detection schemes and still be able to get some kind of security benefit from them; because detective security does not require ASes to exchange information with one another, they do not have to be standardized. And deploying detective BGP security to get some benefit is better than not deploying it and be vulnerable to hijacks without knowing it.

As for future work, better detection schemes can be developed. Chapter 8 already outlined some ideal properties of a new detection scheme in section 8.4, with most of these properties being drawn from Hu-Mao and ARTEMIS (with one technique from Argus) and a suggestion on how a combination of the two would look like is listed in section 8.4.3. A scheme containing these features would probably include, among others, these properties:

- The scheme would use external BGP data sources such as BGPmon and iPlane **in addition to** BGP UPDATE messages that are sent to the AS. External BGP sources are useful for gathering more data, but the detection scheme should not rely solely on external data sources as this can put the scheme out of service if these are down for whatever reason.
- The scheme uses the device fingerprinting techniques, techniques based on inferring inter-AS relationships and C- and D-threads. These are presented in Hu-Mao and Argus. Fingerprinting can be used to gather extra information about hosts in a certain AS to detect (sub)prefix hijacks (which is data that is not likely to change), while inter-AS relationship checking (because inter-AS relationships do not change that often) and C- and D-threads (for reachability checking) are useful for detecting type-n hijacks more accurately.
- The scheme would use the detection of type-0 attacks that ARTEMIS provides, as this makes it almost

trivial to detect prefix hijacks against one's own prefix.

- The scheme makes use of both data plane methods and control plane methods. Using these at the same time balances accuracy and amount of processing power required.
- The scheme only makes use of data received in a sliding window of time. The topology of the Internet is changing all the time, and as such older data is often out-of-date. The size of this sliding window should be decided in a trade-off between the amount of data gathered and the relevance of the data. More data would mean more accurate predictions whereas less data would mean faster predictions.

In addition to the detection-relevant properties, a new scheme should ideally have some mitigation options built-in as well to provide a way of reactive security. As has been seen, ARTEMIS provides two ways to counteract hijacks, those being counter-prefix-hijacking and using a third party to announce the hijacked prefix and tunnel traffic back to the AS. The second one is based on DDoS mitigation services, and instead of counteracting risk, intends to shift risk to a third party, which is a risk management approach for these kinds of attacks that I had not seen until then.

Aside from this, more research can be performed on more accurate classification methods. This can always be done by using more data, but research can also be focusing on other ways to discriminate between benign UPDATE messages and malicious UPDATE messages. With regards to detective security for BGP, this thesis has exclusively focused on detection schemes that detect actual hijacks happening, and detection schemes that detect general anomalies have not been discussed. However, there are several schemes that instead of detecting hijacks, detect general anomalies. Examples are the Listen and Whisper scheme proposed by Subramanian et al. [137] BGPlens by Prakash et al. [123], a scheme by Lutu et al. [108], and a scheme proposed by Al-Musawi et al. [52]. The techniques used by these schemes can be applied in a new detection scheme to further boost the accuracy of the detection.

Sticking to the trend of where we could focus our research from here, more research can be performed on how to detect different kinds of attacks, such as path altering attacks. ARTEMIS and Hu-Mao both have elements that could potentially be useful for a basis in detecting path altering attacks. Examples of this are the type-n prefix hijack detection from ARTEMIS and the edge popularity measuring of Hu-Mao. Both of these could in theory be used to detect false or highly improbable links between ASes.

Another possible area of potential future research is to use multiple detection schemes at the same time and then using the outputs of those detection schemes to get a more accurate result. This approach has been explored a bit in this thesis by combining type-$0/n$ detection with C- and D-threads to get a more accurate result. An example of a scheme that uses multiple algorithms is HEAP. HEAP is a scheme proposed by Schlamp et al. that already takes the classification of multiple detection algorithms, and then verifies these outcomes using other data sources such as registry inference and topology analysis to get more accurate classification results [129]. This is useful for narrowing down the number of false alarms that are raised by the basic detection algorithms.

Bibliography

[1] Why use ipsec ah vs esp? https://security.stackexchange.com/questions/90021/why-use-ipsec-ah-vs-esp. Accessed: 2020-3-6.

[2] Surfnet, the netherlands. https://bgpview.io/asn/1103#graph. Accessed: 2020-1-20.

[3] Asus rt-ac88u. https://www.asus.com/us/Networking/RT-AC88U/specifications/. Accessed: 2020-3-4.

[4] How does bgp select the best routing path. https://www.noction.com/blog/bgp-best-path-selection-algorithm, . Accessed: 2019-10-22.

[5] Bgp finite state model. https://www.inetdaemon.com/tutorials/internet/ip/routing/bgp/operation/finite_state_model.shtml, . Accessed: 2019-10-21.

[6] Bgpmon features. https://bgpmon.net/services/route-monitoring/, . Accessed: 2020-4-29.

[7] Bitcoin's price history. https://www.investopedia.com/articles/forex/121815/bitcoins-price-history.asp. Accessed: 2019-12-29.

[8] The botnet business. https://securelist.com/the-botnet-business/36209/. Accessed: 2020-2-7.

[9] Certificate revocation list (crl). https://searchsecurity.techtarget.com/definition/Certificate-Revocation-List. Accessed: 2019-12-20.

[10] Bgp best path selection algorithm. https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html. Accessed: 2019-12-18.

[11] Cloudflare. https://www.cloudflare.com/. Accessed: 2020-2-11.

[12] Con-ed steals the 'net. https://dyn.com/blog/coned-steals-the-net/, . Accessed: 2020-1-9.

[13] Learn different types of security controls in cissp. https://blog.eduonix.com/networking-and-security/learn-different-types-security-controls-cissp/, . Accessed: 2020-1-15.

[14] Why we need a un cyber security council. https://securelink.net/nb-nb/insights/why-we-need-a-un-cyber-security-council/. Accessed: 2019-12-11.

[15] How a top performance computer looked in the year 2000. https://www.topdesignmag.com/top-performance-computer-looked-the-year-2000/. Accessed: 2020-3-4.

[16] Final report on diginotar hack shows total compromise of ca servers. https://threatpost.com/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112/77170/. Accessed: 2019-12-20.

[17] Innovations in ethernet encryption (802.1ae - macsec) for securing high speed (1-100ge) wan deployments. https://www.cisco.com/c/dam/en/us/td/docs/solutions/Enterprise/Security/MACsec/WP-High-Speed-WAN-Encrypt-MACsec.pdf. Accessed: 2020-3-16.

[18] Ethereum (eth) price. https://coinmarketcap.com/currencies/ethereum/, . Accessed: 2019-12-29.

[19] What is ethereum? https://ethereum.org/what-is-ethereum/, . Accessed: 2019-12-28.

[20] Google (not) hacked? just a dns glitch says google. https://gigaom.com/2005/05/07/google-hacked/, . Accessed: 2020-1-9.

[21] Bgp leak causing internet outages in japan and beyond. https://bgpmon.net/bgp-leak-causing-internet-outages-in-japan-and-beyond/, . Accessed: 2020-1-7.

[22] Google trends: What is google trends? https://www.wordstream.com/google-trends, . Accessed: 2020-1-7.

[23] The great firewall of china: Xi jinping's internet shutdown. https://www.theguardian.com/news/2018/jun/29/the-great-firewall-of-china-xi-jinpings-internet-shutdown. Accessed: 2020-3-4.

[24] Hashing vs encryption — the big players of the cyber security world. https://sectigostore.com/blog/hashing-vs-encryption-the-big-players-of-the-cyber-security-world/. Accessed: 2020-3-17.

[25] Iran's porn censorship broke browsers as far away as hong kong. https://www.theverge.com/2017/1/7/14195118/iran-porn-block-censorship-overflow-bgp-hijack, . Accessed: 2020-3-4.

[26] Iran's telecommunications company illegally reroutes telegram app traffic. https://iranhumanrights.org/2018/08/irans-telecommunications-company-illegally-reroutes-telegram-app
. Accessed: 2020-3-2.

[27] Telegram traffic from around the world took a detour through iran. https://www.cyberscoop.com/telegram-iran-bgp-hijacking/, . Accessed: 2020-3-2.

[28] World wide [redacted]: inside iran's private internet. https://www.theverge.com/2013/4/24/4259672/world-wide-redacted-inside-irans-private-internet, . Accessed: 2020-3-4.

[29] Examine bgp routes and route selection. https://www.juniper.net/documentation/en_US/junos/topics/task/verification/bgp-routes-and-selection-introduction.html. Accessed: 2019-12-17.

[30] What stops a nation-state bgp hijack? https://www.senki.org/what-stops-a-nation-state-bgp-hijack/. Accessed: 2019-12-11.

[31] Nmap. https://nmap.org/. Accessed: 2020-1-27.

[32] Nordvpn. https://nordvpn.com/nl/. Accessed: 2020-5-30.

[33] Public key infrastructure. https://docs.microsoft.com/nl-nl/windows/win32/seccertenroll/public-key-infrastructure?redirectedfrom=MSDN, . Accessed: 2019-12-18.

[34] Public key infrastructure (pki). https://networkencyclopedia.com/public-key-infrastructure-pki/, . Accessed: 2019-12-18.

[35] Pkioverheid certificaten. https://cert.pkioverheid.nl/cert-pkioverheid-nl.htm, . Accessed: 2019-12-20.

[36] Pakistan's accidental youtube re-routing exposes trust flaw in net. https://www.wired.com/2008/02/pakistans-accid/. Accessed: 2020-9-30.

[37] Plane (in networking). https://whatis.techtarget.com/definition/plane-in-networking. Accessed: 2020-5-12.

[38] Ripe ncc ris tools and web interfaces. https://www.ripe.net/analyse/archived-projects/ris-tools-web-interfaces. Accessed: 2020-1-21.

[39] Intakeadvies rpki. https://www.forumstandaardisatie.nl/sites/bfs/files/proceedings/FS-190612.4B_Intakeadvies_RPKI.pdf. Accessed: 2020-1-2.

[40] Rsync considered inefficient and harmful. https://www.ietf.org/proceedings/89/slides/slides-89-sidr-6.pdf. Accessed: 2020-3-11.

[41] Syn flood (half open attack). https://searchsecurity.techtarget.com/definition/SYN-flooding. Accessed: 2020-2-11.

[42] Tcp operational overview and the tcp finite state machine (fsm). http://www.tcpipguide.com/free/t_TCPOperationalOverviewandtheTCPFiniteStateMachineF-2.htm. Accessed: 2020-2-7.

[43] Internet-wide catastrophe—last year. https://web.archive.org/web/20080228131639/http://www.renesys.com/blog/2005/12/internetwide_nearcatastrophela.shtml. Accessed: 2020-1-9.

[44] Do you need a vpn? quite possibly. here's why. https://mashable.com/article/why-you-need-vpn/?europe=true. Accessed: 2020-5-30.

[45] Youtube, llc. https://bgpview.io/asn/11344#prefixes-v4. Accessed: 2020-3-2.

[46] Oracle confirms china telecom internet traffic 'misdirections'. https://www.zdnet.com/article/oracle-confirms-china-telecom-internet-traffic-misdirections/, . Accessed: 2019-11-13.

[47] Large european routing leak sends traffic through china telecom. https://blog.apnic.net/2019/06/07/large-european-routing-leak-sends-traffic-through-china-telecom/, . Accessed: 2019-11-13.

[48] iplane: An information plane for distributed services. https://web.eecs.umich.edu/~harshavm/iplane/. Accessed: 2020-1-17.

[49] Internet routing black hole. http://catless.ncl.ac.uk/Risks/19.12.html#subj1. Accessed: 2019-10-23.

[50] Pakistan hijacks youtube. https://dyn.com/blog/pakistan-hijacks-youtube-1/, . Accessed: 2019-10-23.

[51] Youtube hijacking: A ripe ncc ris case study. https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study, . Accessed: 2019-10-23.

[52] Bahaa Al-Musawi, Philip Branch, and Grenville Armitage. Detecting bgp instability using recurrence quantification analysis (rqa). In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2015.

[53] Bahaa Al-Musawi, Philip Branch, and Grenville Armitage. Bgp anomaly detection techniques: A survey. *IEEE Communications Surveys & Tutorials*, 19(1):377–396, 2016.

[54] David Alderson and Kevin Soo Hoo. The role of economic incentives in securing cyberspace. *Center for International Security and Cooperation, Stanford [Online]. Available at: http://cisac. fsi. stanford. edu/publications/role_of_economic_incentives_in_securing_cyberspace_the*, 2004.

[55] Randall J Atkinson. Toward a more secure internet. *Computer*, 30(1):57–61, 1997.

[56] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the internet. *ACM SIGCOMM Computer Communication Review*, 37(4):265–276, 2007.

[57] Esther Zippora Baranoff and Esther Zippora Baranoff. *Risk management and insurance*. Wiley Danvers, 2004.

[58] Steven M Bellovin. Security problems in the tcp/ip protocol suite. *ACM SIGCOMM Computer Communication Review*, 19(2):32–48, 1989.

[59] Steven M Bellovin. A technique for counting natted hosts. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 267–272. ACM, 2002.

[60] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Annual international cryptology conference*, pages 41–55. Springer, 2004.

[61] Randy Bush, Rob Austein, Steve Bellovin, and Michael Elkins. The rpki & origin validation, 2009.

[62] Kevin Butler, Toni Farley, Patrick McDaniel, and Jennifer Rexford. A survey of bgp security. *ACM, draft version*, 5:1–35, 2004.

[63] Kevin Butler, Toni R Farley, Patrick McDaniel, and Jennifer Rexford. A survey of bgp security issues and solutions. *Proceedings of the IEEE*, 98(1):100–122, 2009.

[64] Haowen Chan, Debabrata Dash, Adrian Perrig, and Hui Zhang. Modeling adoptability of secure bgp protocol. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 279–290. ACM, 2006.

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

[66] David Cooper, Stefan Santesson, S Farrell, Sharon Boeyen, Rusell Housley, and W Polk. Rfc 5280: Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile. *IETF, May*, 2008.

[67] J Durand, I Pepelnjak, and G Doering. Rfc 7454-bgp operations and security, 2015.

[68] Toni Farley, Patrick Mcdaniel, and Kevin Butler. A survey of bgp security issues and solutions. *AT&T Labs-Research, Florham Park, NJ*, 2004.

[69] Muhammad Farooq and Akhtar Zeb. Bgp threats and practical security. Master's thesis, 2011.

[70] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking (ToN)*, 9(6):733–745, 2001.

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

[72] Lixin Gao, Timothy G Griffin, and Jennifer Rexford. Inherently safe backup routing with bgp. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, volume 1, pages 547–556. IEEE, 2001.

[73] Yossi Gilad, Omar Sagga, and Sharon Goldberg. Maxlength considered harmful to the rpki. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 101–107. ACM, 2017.

[74] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Let the market drive deployment: A strategy for transitioning to bgp security. *ACM SIGCOMM computer communication review*, 41(4):14–25, 2011.

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

[76] Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick D McDaniel, and Aviel D Rubin. Working around bgp: An incremental approach to improving security and accuracy in interdomain routing. In *NDSS*, volume 23, page 156, 2003.

[77] Walter Goralski. *The illustrated network: how TCP/IP works in a modern network.* Morgan Kaufmann, 2017.

[78] Peter Grabosky. The evolution of cybercrime, 2006–2016. In *Cybercrime through an interdisciplinary lens*, pages 29–50. Routledge, 2016.

[79] Nelson E Hastings and Paul A McLean. Tcp/ip spoofing fundamentals. In *Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pages 218–224. IEEE, 1996.

[80] Ethan Heilman, Danny Cooper, Leonid Reyzin, and Sharon Goldberg. From the consent of the routed: Improving the transparency of the rpki. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 51–62. ACM, 2014.

[81] Clint Hepner and Earl Zmijewski. Defending against bgp man-in-the-middle attacks. *Talk at BlackHat*, 2009, 2009.

[82] Xin Hu and Z Morley Mao. Accurate real-time identification of ip hijacking.

[83] Xin Hu and Z Morley Mao. Accurate real-time identification of ip prefix hijacking. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 3–17. IEEE, 2007.

[84] Yih-Chun Hu, Adrian Perrig, and Marvin Sirbu. Spv: Secure path vector routing for securing bgp. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 179–192. ACM, 2004.

[85] Geoff Huston and Randy Bush. Securing bgp with bgpsec. In *The Internet Protocol Forum*, volume 14, 2011.

[86] S. Forrest J. Karlin and J. Rexford. Pgbgp simulator. `http://cs.unm.edu/karlinjf/pgbgp/`, . Accessed: 2020-1-16.

[87] S. Forrest J. Karlin and J. Rexford. Internet alert registry. `http://cs.unm.edu/karlinjf/IAR/`, . Accessed: 2020-1-16.

[88] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Pretty good bgp: Improving bgp by cautiously adopting routes. In *Proceedings of the 2006 IEEE International Conference on Network Protocols*, pages 290–299. IEEE, 2006.

[89] Stephen Kent and Randall Atkinson. Ip authentication header, 1998.

[90] Stephen Kent and Randall Atkinson. Rfc2406: Ip encapsulating security payload (esp), 1998.

[91] Stephen Kent and Andrew Chi. Threat model for bgp path security. Technical report, RFC 7132, February, 2014.

[92] Stephen Kent, Charles Lynn, and Karen Seo. Secure border gateway protocol (s-bgp). *IEEE Journal on Selected areas in Communications*, 18(4):582–592, 2000.

[93] Stephen T Kent. Securing the border gateway protocol: A status update. In *IFIP International Conference on Communications and Multimedia Security*, pages 40–53. Springer, 2003.

[94] Stephen T Kent, Charles Lynn, Joanne Mikkelson, and Karen Seo. Secure border gateway protocol (s-bgp)-real world performance and deployment issues. In *NDSS*, 2000.

[95] Naasir Kamaal Khan, Gulabchand K Gupta, and ZA Usmani. Deployment issues of sbgp, sobgp and psbgp: A comparative analysis. *International Journal of Advances in Engineering & Technology*, 1(4): 236, 2011.

[96] Brijesh Kumar. Integration of security in network routing protocols. *ACM SIGSAC Review*, 11(2):18–25, 1993.

[97] Frank La Rue. Report of the special rapporteur on the promotion and protection of the right to freedom of opinion and expression. 2011.

[98] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. *IEEE/ACM transactions on networking*, 9(3):293–306, 2001.

[99] Mohit Lad, Daniel Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. Phas: A prefix hijack alert system.

[100] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[101] Taeho Lee, Pawel Szalachowski, David Barrera, Adrian Perrig, Heejo Lee, and David Watrin. Bootstrapping real-world deployment of future internet architectures. *arXiv preprint arXiv:1508.02240*, 2015.

[102] Matt Lepinski and Stephen Kent. An infrastructure to support secure internet routing. 2012.

[103] Matt Lepinski, Derrick Kong, and Stephen Kent. A profile for route origin authorizations (roas). 2012.

[104] Matthew Lepinski. Bgpsec protocol specification. 2017.

[105] Qi Li, Yih-Chun Hu, and Xinwen Zhang. Even rockets cannot make pigs fly sustainably: Can bgp be secured with bgpsec. In *Workshop SENT'14, 23 February 2014, San Diego, USA, Copyright 2014 Internet Society: Proceedings*. Internet Society, 2014.

[106] Qi Li, Xinwen Zhang, Xin Zhang, and Purui Su. Invalidating idealized bgp security proposals and countermeasures. *IEEE Transactions on Dependable and Secure Computing*, 12(3):298–311, 2014.

[107] Xiaowei Liu, Zhiwei Yan, Guanggang Geng, Xiaodong Lee, Shian-Shyong Tseng, and Ching-Heng Ku. Rpki deployment: Risks and alternative solutions. In *Genetic and Evolutionary Computing*, pages 299–310. Springer, 2016.

[108] Andra Lutu, Marcelo Bagnulo, Jesus Cid-Sueiro, and Olaf Maennel. Separating wheat from chaff: Winnowing unintended prefixes using machine learning. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 943–951. IEEE, 2014.

[109] Robert Lychev, Michael Schapira, and Sharon Goldberg. Rethinking security for internet routing. *Communications of the ACM*, 59(10):48–57, 2016.

[110] Steve Mansfield-Devine. The growth and evolution of ddos. *Network Security*, 2015(10):13–20, 2015.

[111] D McGrew and M Pritikin. The compressed x. 509 certificate format. *draft-pritikin-comp-x509-00, May*, 2010.

[112] Ralph C Merkle. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*, pages 369–378. Springer, 1987.

[113] Asya Mitseva, Andriy Panchenko, and Thomas Engel. The state of affairs in bgp security: A survey of attacks and defenses. *Computer Communications*, 124:45–60, 2018.

[114] Sandra L Murphy. Secure inter-domain routing standards evolution and role in the future gig. In *MILCOM 2007-IEEE Military Communications Conference*, pages 1–7. IEEE, 2007.

[115] David Murray. Why internet protocols need incentives. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pages 261–266. IEEE, 2015.

[116] Bahaa Musawi, Philip Branch, and Grenville Armitage. Bgp anomaly detection techniques: A survey. *IEEE Communications Surveys and Tutorials*, PP:1–1, 10 2016. doi: 10.1109/COMST.2016.2622240.

[117] Martin O Nicholes and Biswanath Mukherjee. A survey of security techniques for the border gateway protocol (bgp). *IEEE communications surveys & tutorials*, 11(1):52–65, 2009.

[118] Ola Nordström and Constantinos Dovrolis. Beware of bgp attacks. *ACM SIGCOMM Computer Communication Review*, 34(2):1–8, 2004.

[119] PC van Oorschot, Tao Wan, and Evangelos Kranakis. On interdomain routing security and pretty secure bgp (psbgp). *ACM Transactions on Information and System Security (TISSEC)*, 10(3):11–es, 2007.

[120] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. Bgpstream: a software framework for live and historical bgp data analysis. In *Proceedings of the 2016 Internet Measurement Conference*, pages 429–444. ACM, 2016.

[121] Radia Joy Perlman. *Network layer protocols with byzantine robustness*. PhD thesis, Massachusetts Institute of Technology, 1988.

[122] W Polk and D Solo. Internet x. 509 public key infrastructure certificate and crl profile. 1999.

[123] B Aditya Prakash, Nicholas Valler, David Andersen, Michalis Faloutsos, and Christos Faloutsos. Bgplens: Patterns and anomalies in internet routing updates. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1315–1324, 2009.

[124] Jörg Rech. Discovering trends in software engineering with google trend. *ACM SIGSOFT software engineering notes*, 32(2):1–2, 2007.

[125] Yakov Rekhter and Tony Li. A border gateway protocol 4 (bgp-4). 1995.

[126] Alvaro Retana. Secure origin bgp (sobgp). In *NANOG28 Meeting*, 2003.

[127] Leonid Reyzin and Natan Reyzin. Better than biba: Short one-time signatures with fast signing and verifying. In *Australasian Conference on Information Security and Privacy*, pages 144–153. Springer, 2002.

[128] Oregon RouteViews. University of oregon routeviews project. *Eugene, OR.[Online]. Available: http://www. routeviews. org.*

[129] Johann Schlamp, Ralph Holz, Quentin Jacquemart, Georg Carle, and Ernst W Biersack. Heap: reliable assessment of bgp hijacking attacks. *IEEE Journal on Selected Areas in Communications*, 34(6):1849–1861, 2016.

[130] Max Schuchard, Abedelaziz Mohaisen, Denis Foo Kune, Nicholas Hopper, Yongdae Kim, and Eugene Y Vasserman. Losing control of the internet: using the data plane to attack the control plane. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 726–728. ACM, 2010.

[131] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. Artemis: Neutralizing bgp hijacking within a minute. *IEEE/ACM Transactions on Networking*, 26(6):2471–2486, 2018.

[132] Georgos Siganos and Michalis Faloutsos. A blueprint for improving the robustness of internet routing, 2005.

[133] Bradley R Smith and JJ Garcia-Luna-Aceves. Efficient security mechanisms for the border gateway routing protocol. *Computer Communications*, 21(3):203–210, 1998.

[134] Bradley R Smith and JJ Garcia-Lunes-Aceves. Securing the border gateway routing protocol. Technical report, CALIFORNIA UNIV SANTA CRUZ DEPT OF COMPUTER ENGINEERING, 1996.

[135] Bradley R Smith, Shree Murthy, and Jose Joaquin Garcia-Luna-Aceves. Securing distance-vector routing protocols. In *Proceedings of SNDSS'97: Internet Society 1997 Symposium on Network and Distributed System Security*, pages 85–92. IEEE, 1997.

[136] Kotikapaludi Sriram, Oliver Borchert, Okhee Kim, Patrick Gleichmann, and Doug Montgomery. A comparative analysis of bgp anomaly detection and robustness algorithms. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 25–38. IEEE, 2009.

[137] Lakshminarayanan Subramanian, Volker Roth, Ion Stoica, Scott Shenker, and Randy Katz. Listen and whisper: Security mechanisms for bgp. In *Proc. NSDI*, volume 4. Citeseer, 2004.

[138] Martin Suchara, Ioannis Avramopoulos, and Jennifer Rexford. Securing bgp incrementally. In *Proceedings of the 2007 ACM CoNEXT conference*, page 52. ACM, 2007.

[139] P Traina, D McPherson, and J Scudder. Rfc 5065: Autonomous system confederations for bgp, 2007.

[140] Dario Vieira. A survey of bgp session maintenance issues and solutions. *Network Protocols & Algorithms*, 2(1):132–157, 2010.

[141] Matthias Wählisch, Robert Schmidt, Thomas C Schmidt, Olaf Maennel, Steve Uhlig, and Gareth Tyson. Ripki: The tragic story of rpki deployment in the web ecosystem. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, page 11. ACM, 2015.

[142] Tao Wan, Evangelos Kranakis, and Paul C van Oorschot. Pretty secure bgp, psbgp. In *NDSS*. Citeseer, 2005.

[143] Tao Wan, P van Oorschot, and Evangelos Kranakis. A selective introduction to border gateway protocol (bgp) security issues. *Proc. of NATO Advanced Studies Institute on Network Security and Intrusion Detection*, 2007.

[144] Feiyi Wang, Brian Vetter, and Shyhtsun Felix Wu. Secure routing protocols: Theory and practice. Technical report, Technical report, North Carolina State University, 1997.

[145] Dan Wendlandt, Ioannis Avramopoulos, David G Andersen, and Jennifer Rexford. Don't secure routing protocols, secure data delivery. *Proc. of ACM HotNets, Irvine, CA, USA*, pages 7–12, 2006.

[146] Yang Xiang, Zhiliang Wang, Xia Yin, and Jianping Wu. Argus: An accurate and agile system to detecting ip prefix hijacking. In *2011 19th IEEE International Conference on Network Protocols*, pages 43–48. IEEE, 2011.

[147] Qianqian Xing, Baosheng Wang, and Xiaofeng Wang. Poster: Bgpcoin: A trustworthy blockchain-based resource management solution for bgp security. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2591–2593. ACM, 2017.

[148] Qianqian Xing, Baosheng Wang, and Xiaofeng Wang. Bgpcoin: Blockchain-based internet number resource authority and bgp security solution. *Symmetry*, 10(9):408, 2018.

[149] He Yan, Ricardo Oliveira, Kevin Burnett, Dave Matthews, Lixia Zhang, and Dan Massey. Bgpmon: A real-time, scalable, extensible monitoring system. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 212–223. IEEE, 2009.

[150] Heng Yin, Bo Sheng, Haining Wang, and Jianping Pan. Securing bgp through keychain-based signatures. In *2007 Fifteenth IEEE International Workshop on Quality of Service*, pages 154–163. IEEE, 2007.

[151] Akhtar Zeb and Muhammad Farooq. Bgp threats and practical security. 2011.

[152] Fuyong Zhang and Ying Ma. Using irp with a novel artificial immune algorithm for windows malicious executables detection. In *2016 International conference on progress in informatics and computing (PIC)*, pages 610–616. IEEE, 2016.

[153] Jian Zhang, Daofeng Li, and Bowen Zhao. A prefix hijacking detection model based on the immune network theory. *IEEE Access*, 7:132384–132394, 2019.

[154] Ying Zhang, Zheng Zhang, Z Morley Mao, and Y Charlie Hu. Hc-bgp: A light-weight and flexible scheme for securing prefix ownership. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, pages 23–32. IEEE, 2009.

[155] Zheng Zhang, Ying Zhang, Y Charlie Hu, Z Morley Mao, and Randy Bush. Ispy: detecting ip prefix hijacking on my own. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 327–338. ACM, 2008.

[156] Meiyuan Zhao, Sean W Smith, and David M Nicol. The performance impact of bgp security. *IEEE network*, 19(6):42–48, 2005.

[157] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S Felix Wu, and Lixia Zhang. An analysis of bgp multiple origin as (moas) conflicts. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 31–35. ACM, 2001.

[158] Changxi Zheng, Lusheng Ji, Dan Pei, Jia Wang, and Paul Francis. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 277–288. ACM, 2007.