Computer Networks 82 (2015) 135–155

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Self-reliant detection of route leaks in inter-domain routing

M.S. Siddiqui *, D. Montero, R. Serral-Gracià, M. Yannuzzi

Networking and Information Technology Lab (NetIT Lab), Technical University of Catalonia (UPC), Spain

ARTICLE INFO

Article history:
Available online 7 March 2015

Keywords:
BGP
Reliability
Security
Routing
Internet
Inter-domain

ABSTRACT

Route leaks are among the several inter-domain routing anomalies that have the potential to cause large scale service disruptions on the Internet. The reason behind the occurrence of route leaks is the violation of routing policies among Autonomous Systems (ASes). There exist a few rudimentary solutions that can be used as a first line of defense, such as the utilization of route filters, but these palliatives become unfeasible in large domains due to the administrative overhead and the cost of maintaining the filters updated. As a result, a significant part of the Internet is defenseless against route leak attacks. In this paper, we examine the different types of route leaks and propose detection methodologies for improving the reliability of the routing system. Our main contributions can be summarized as follows. We develop a relatively basic theoretical framework, which, under realistic assumptions, enables a domain to autonomously determine if a particular route advertisement received from a neighbor corresponds to a route leak. Based on this, we propose three incremental methodologies, namely Cross-Path (CP), Benign Fool Back (BFB), and Reverse Benign Fool Back (R-BFB), for autonomously detecting route leaks. Our strength resides in the fact that these detection techniques solely require the analysis of control and data plane information available within the domain. We analyze the performance of the proposed route leak identification techniques both through real-time experiments as well as simulations at large scale. Our results show that the proposed detection techniques achieve high success rates for countering route leaks in different scenarios.

1. Introduction

The security and reliability of the Border Gateway Protocol (BGP) [1] have been actively investigated since its adoption as the standardized inter-domain routing protocol among Autonomous Systems (ASes) in the Internet. The implicit trust model among ASes for exchanging reachability information using BGP, along with the lack of in-built security mechanisms in the protocol itself make the inter-domain routing system vulnerable to a number of security threats, such as false IP prefix origination and false route advertisements. As evident from the Youtube incident in 2008 [2] and alleged Chinese Telecom traffic hijacking event in 2010 [3], even non-sophisticated attacks have the potential to globally disrupt the Internet. Another inter-domain routing anomaly with the potential to produce large scale service disruptions is the "route leak" problem. Route leaks occur due to policy violations while exporting routes to a neighbor AS. The ASes typically set their policies for exporting or importing routes from a neighbor AS according to the business relationship that they have with that specific neighbor on a given inter-domain link. There are three types of business relationships between any two ASes: (1) customer–provider; (2) peer–peer; and (3) sibling–sibling relation. In a customer–provider relation, the provider AS offers transit to the customer AS. The ASes in a peer–peer relation usually

* Corresponding author. Tel.: +34 938967294.
E-mail addresses: siddiqui@ac.upc.edu (M.S. Siddiqui), dmontero@ac.upc.edu (D. Montero), rserral@ac.upc.edu (R. Serral-Gracià), yannuzzi@ac.upc.edu (M. Yannuzzi).

http://dx.doi.org/10.1016/j.comnet.2015.02.029
1389-1286/

exchange only their customers' traffic between each other up to an agreed upon threshold. A sibling–sibling relation exists between two ASes which belong to the same organization and the ASes typically offer customized transit to each other. A peer–peer relation is different from a sibling–sibling relation in the sense that the ASes, in the latter case, are owned by the same organization whereas, in the former case, the two ASes belong to two distinct organizations. This difference leads to different type of AS polices among the ASes (cf. Section 7).

A route leak occurs when an AS advertises a route toward a neighbor AS that does not respect the agreed business relationship between them. For instance, if a customer AS starts offering transit between two of its providers, then it is a route leak. Similarly, a route leak will occur if an AS advertises routes learned from one provider toward a peer AS. We will delve into these aspects later on, but in general terms, a route leak entails a violation of the business relationship that rules the interconnection of domains.

The main concern about route leaks is that they are a common occurrence, and regardless if they are due to misconfigurations or deliberate attacks, they can lead to traffic loss, sub-optimal routing, and more importantly, traffic hijacking. For instance, in 2012, a multi-homed ISP leaked routes learned from one of its providers to another provider, causing a national level disruption in Internet service in Australia [4]. Another major route leak incident occurred the same year, when one of Google's peers improperly advertised Google routes to its provider, knocking out Google services for around half an hour [5]—we shall describe these two incidents in more detail later in Section 2.

Route leaks are apparently simple but hard to solve. This is because the ASes keep the information regarding their relationships and policies with other ASes confidential, which makes the identification of policy violations a challenging problem. Although there are orthodox countermeasures for the route leak problem, including route filters, Internet Route Registries (IRRs), and several BGP monitoring tools, they become impotent or unreliable in face of scalability, due to the high cost of maintenance and dependence on third party information.

In this paper, we extend our work presented in [6] where we formally analyzed and developed the route leak problem. In [6], we described different types of route leaks and explained how, where, and why they occur with the help of example scenarios. More importantly, we showed that, under realistic assumptions and routing conditions, a single AS can detect route leaks utilizing only the standard routing information available at hand, and without needing any vantage point deployed in the internetwork. Our approach targets inference and route leak detection requiring neither changes nor extensions to the BGP protocol. Based on the theoretical framework presented in [6], in this paper we develop three incremental route leak identification techniques, namely Cross-Path (CP), Benign Fool Back (BFB) and Reverse Benign Fool Back (R-BFB). The first two techniques are based on the analysis of BGP's control-plane information, i.e., our mechanisms are able to counter a considerable fraction of route leaks utilizing only the information available from the Routing Information Base (RIB) of the BGP routers in the AS—and obviously the knowledge of the AS relationships with direct neighbors as well. The third technique, R-BFB, also takes advantage of data-plane traffic to provide additional information to the analytics performed to the BGP RIBs. The CP, BFB and R-BFB techniques are described in detail in Sections 4–6, respectively. Furthermore, we evaluate the proposed techniques both experimentally as well as through event-driven simulations at large scale. For the latter, we utilized a sub-graph of the Internet graph extracted from ARK [7], and we performed simulations using NS2 [8] and BGP++ [9] on a topology composed of more than 1600 ASes. For the experimental part, we deployed an inter-domain network topology consisting of almost 1000 ASes using Linux Containers (Docker [10]), with the aim of testing our route leak identification techniques in a scenario that can realistically support the data-plane part. The results from our tests, which include more than 20,000 event driven simulations and 1930 real-time experiments, show that an AS is able to autonomously detect route leaks in different scenarios with a high success rate using the CP, BFB and R-BFB, especially, when the three techniques are combined and used together. As far as our knowledge goes, our work introduces the first theoretical and experimental analysis for autonomously detecting route leaks in the Internet.

The rest of the paper is organized as follows. Section 2 describes two real world examples of route leaks. The theoretical framework for detecting route leaks including, definition and description of different types, hypotheses and formalization for their detection, is explained in Section 3. Sections 4–6, introduce the three Route Leak Detection (RLD) techniques, CP, BFB and R-BFB, respectively. The simulations and experimental tests and their results are covered in their respective sections. Section 7 discusses the route leak problem and its detection in sibling–sibling relationship and Section 8 highlights open issues. The related work along with the comparison with our proposed solution is provided in Section 9, and finally, Section 10 concludes the paper.

2. Route leaks in real world

Internet service outages by virtue of the BGP shortcomings are frequent [11], but only a few succeed to get mass attention—in practice this typically depends on the scale of the service disruption and the profile of the victims. In this section, we illustrate two major Internet disruption incidents, that we refer to as Telstra-Dodo [4] and Google-Moratel [5]. The apparent causes behind the disruptions point out to incidents that involuntary produced route leaks. More specifically, these incidents were thoroughly analyzed, and the collected evidence boils down to the violation of routing policies between ASes. However, what could not be clarified, is if they were due to intentional (e.g., a traffic hijack attack) or unintentional misconfiguration (e.g., a fat-finger problem) over the export policies of an AS. Despite the traces and evidence left, we found that some service providers involved in these cases claimed that the issues were due to hardware failures, thereby avoiding to mention the possible case of route leaks [12].

Let us describe these two incidents, which we consider clear examples of what route leaks are and their repercussions. A country-level Internet service disruption occurred in Australia on February 23, 2012 [4], which was attributed to malfunctioning of a router. Apparently, one of Dodo's network (AS38285) edge routers exported all its internal routes to one of its providers, namely Telstra (AS1221) (see Fig. 1). The internal routes that Dodo advertised or leaked to Telstra included all routes learned from its other providers. These provider-learned routes enclosed all the exported routes of Optus (AS7474), PIPE Internet Exchange (AS23745, AS18398) and the Equinex Exchange (AS24115). Besides, Optus had a peer link with Telstra and, as the latter learned the route to Optus (it's peer) through Dodo (it's customer), it preferred the customer path as "the best path" (i.e., all traffic coming from Telstra toward Optus was routed via Dodo). The reason behind preferring a customer path over a direct peer link is purely economical. We shall explain AS-Path preferences based on the type of AS relationships in detail later in Section 3. As shown in Fig. 1, this route leak incident turned into a snowball effect when Telstra advertised the new set of Dodo-learned routes to its provider, Telstra International (AS4637), which further advertised them to its peers and customers. Eventually, the disruption on the Internet service became visible once Telstra started forwarding large amounts of traffic toward Dodo, which was not equipped to handle the traffic volume. Therefore, the peers and customers of Telstra International also started to experience the Internet service disruption. This entire event occurred in less than an hour, causing large scale connectivity problems across Australia.

Another widely noticed Internet outage due to route leaks that directly affected Google's services over some portions of the Internet took place on November of 2012, and lasted for about 27 min [5]. In this case, Google (AS15169) experienced routing issues with its peer Moratel (AS23947). Fig. 2 illustrates the scenario in terms

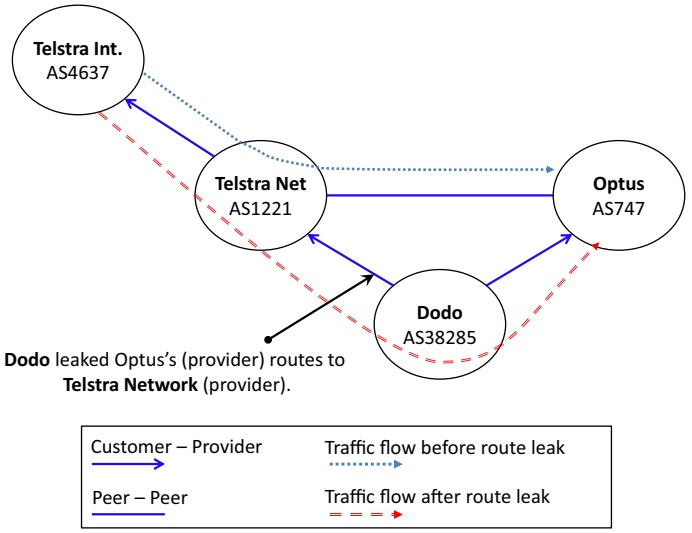


Fig. 1. Change of traffic flow in case of the Dodo route leak in April 2012.

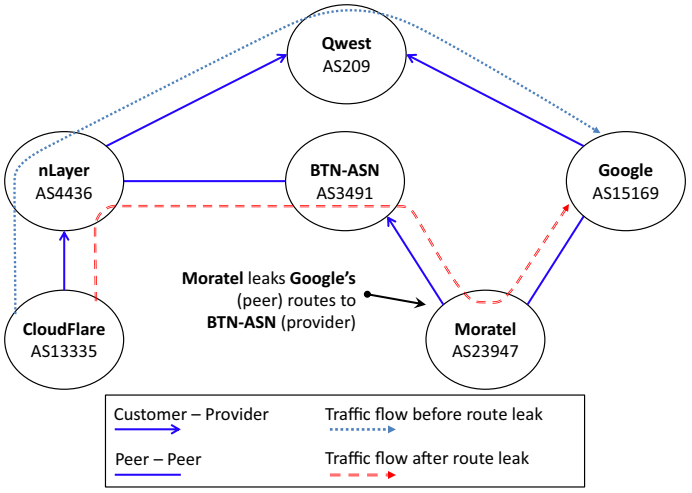


Fig. 2. Change of traffic flow in case of the Google route leak in November 2012.

of the traffic path change from the perspective of one of the affected users, CloudFlare (AS13335). They received a route toward Google through an Indonesian service provider Moratel (AS23947). This happened because Moratel exported the routes learned from its peer (Google) toward its provider (BTN-ASN), and Moratel's provider selected the leaked routes and exported them further. CloudFlare's provider, nLayer (AS4436), preferred the route received from its peer (BTN-ASN) over the old route it had toward Google through its provider, Qwest (AS209). Again, the reason behind preferring a peer route over a provider route is economical (cf. Section 3). The leaked routes from Moratel propagated and attracted a huge amount of traffic for Google through itself. The Moratel Network could not cope with such huge traffic load and eventually started dropping traffic. Whilst this problem was figured out and solved, Google's outage was seen from different segments of the Internet. These incidents clearly expose the inefficacy of the techniques and tools available today for countering route leaks—the main ones will be outlined in next section. In summary, route leaks represent a high risk and challenging problem that requires new approaches and research efforts. This is precisely the motivation for this paper.

3. Formalizing route leaks

In this section, we formally describe the route leak problem and lay out the theoretical framework for the identification of route leaks, but first we define the terminology and the set of policies that rule the routing among ASes.

3.1. Preliminaries

A "provider link" of an AS is a link that connects it to its provider AS. Similarly, the terms "customer link", "peer link" or "sibling link" refer to a link that connects an AS with a customer AS, a peer AS or a sibling AS, respectively. In this paper, we focus on the two dominant AS relationships in the Internet, which are the customer–provider and peer–peer relationships, since the percentage of sibling relations in the Internet is comparatively negligible [13].

Whilst the relationship between two ASes is business oriented, pragmatically it is implemented through the BGP protocol. BGP provides complete flexibility for implementing route export or import policies according to the defined relationship, by means of several attributes associated with each advertised route. For example, a provider AS will export all its routes toward its customer ASes in order to attract traffic through its customer links. We are more interested in the export policies, as route leaks occur due to violation of business policies through these exports. The guidelines used for exporting routes (i.e., how to advertise routes depending on the type of relationship with the neighbor AS) are referred to as valley-free rules [13], and they can be summarized as follows:

Rule $\mathcal{R}$.1. "Routes learned from Customers can be further advertised to other Customers, Peers and Providers."

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

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

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

3.2. Defining route leaks

At present, there is no standard definition of the route leak problem in the Internet community. The working group in charge of securing inter-domain routing, namely, the SIDR WG [14], has delegated the task of defining the route leak problem to the GROW WG [15]. The reason for this is that SIDR not only considers route leaks out of their scope but also because their proposals, including RPKI [16], ROA [17] and BGPSEC [18], fail to counter route leaks. There exist some attempts in the literature from where we can extract the initial understanding of the route leak problem. In [19], the author defines route leaks as *the advertisement of a non-customer route over a peer or a provider link*.

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

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

That is, any route advertisement by an AS which infringes the valley-free rules $\mathcal{R}$.2 or $\mathcal{R}$.3 is a route leak. Note that rule $\mathcal{R}$.1 cannot be infringed, since an AS can always export customer routes independently of the

business relationship with the neighbor to which it is exporting the route to. Also note that the valley-free rules are not necessarily upheld while exchanging routes under complex AS relationships, e.g., under hybrid relationships—these will be discussed later in Section 4. However, such complex relationships are quite uncommon in practice, so the above definition provides a realistic and quite general basis for our initial modeling of route leaks. Using the above definition, we identify two possible types of route leaks from the perspective of an AS which wants to detect route leaks corresponding to the type of the link they occur on, namely, *Customer Route Leaks* and *Peer Route Leaks*. We proceed to describe them through examples.

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

Peer route leak: Let us consider now the scenario shown in Fig. 4(a). The AS c is multi-homed with provider ASes a and b. AS d has a peer relation with AS e, and AS d and AS e have a customer–provider relationship with ASes a and b, respectively. AS a and AS b also have a peer link between them. Let us consider the propagation of a route for prefix $\mathcal{P}_1$ owned by AS c, i.e., c advertises the route $\mathcal{P}_1 : [c]$ to its providers. Following $\mathcal{R}.1$, a forwards $\mathcal{P}_1 : [a, c]$ to its customer d and b forwards $\mathcal{P}_1 : [b, c]$ to its customer e. By $\mathcal{R}.3$, d does not advertise the route to its peer e, and reciprocally. The traffic aimed for $\mathcal{P}_1$ originated in AS d would follow the path $[a, c]$ as shown in Fig. 4(a). Now, if AS e advertises the route for prefix $\mathcal{P}_1$ to its peer d, the latter would receive two different routes for prefix $\mathcal{P}_1$, i.e., $\mathcal{P}_1 : [a, c]$ via a, and $\mathcal{P}_1 : [e, b, c]$ via e, as shown in Fig. 4(b). Since d will prefer routes learned from peers over routes learned from providers, the traffic between d and c will now follow the path $[e, b, c]$. Note that similarly to the case shown in Fig. 3, the path length via a is shorter than the path length via e, but still d will select the peer route, since d will prioritize it by setting higher the *local-pref* value. In this example, the route $\mathcal{P}_1 : [e, b, c]$ exported by AS e toward AS d results in a route leak, given that it violates the valley-free rule $\mathcal{R}.3$. Observe that, the route leak examples shown in Figs. 3 and 4 infringe rule $\mathcal{R}.3$, but other examples can be easily elaborated infringing rule $\mathcal{R}.2$.

3.3. Route leak identification

The identification of route leaks is the first step toward solving the route leak problem. Thus, we systematically analyze the various environments where route leaks are possible, and then propose a mechanism for their identification using the definition of valley-free rules stated in the previous section.

In our study, we assume that the route leak identification analysis only uses readily available data, e.g., information obtained directly from the routing tables—that is, from the Route Information Base (RIB) of the routers. We particularly exclude from our analysis data obtained from external sources, such as route information imported from vantage points. In this sense, our identification analysis focuses on what can actually be inferred in a domain under realistic routing conditions, by solely examining the routes received from its neighbors.

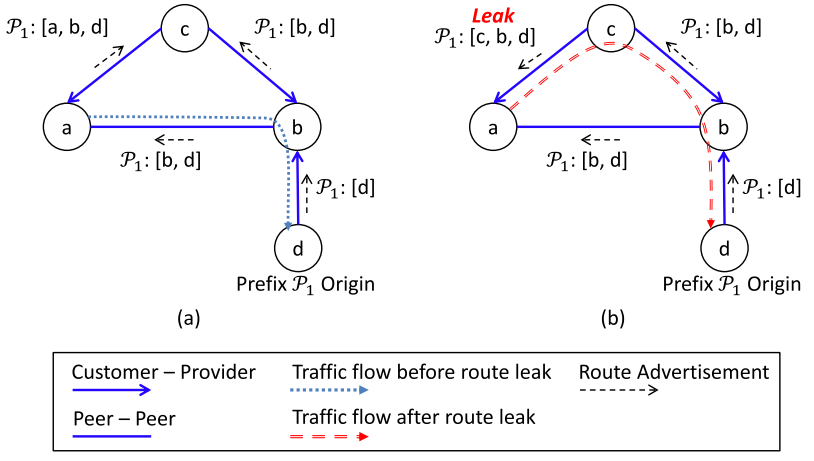


Fig. 3. Customer route leak scenario: (a) Before the route leak. (b) After the route leak (AS c leaks a route toward its provider AS a).

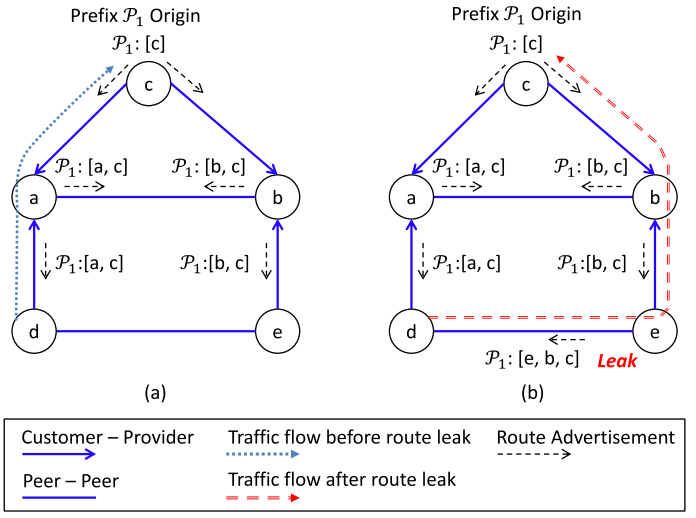


Fig. 4. Peer route leak scenario: (a) Before the route leak. (b) After the route leak (AS *e* leaks a route toward its peer AS *d*).

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

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

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

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

Let us assume a reference AS *a* in charge of identifying route leaks, as shown in Fig. 5(a). Then for domain *a*, route leaks can only occur as a result of routes exported by its customer AS *c* or peer AS *p*. In the case that *c* exports routes owned by itself, then such route advertisements can never produce a route leak, since *c*, being customer of *a*, can export its own routes to its provider. Similarly, *p* is allowed to export its own routes to its peer *a*. Hence, it should be clear that the advertisement of routes owned by a customer or a peer ASes can never cause a route leak on AS *a*. In other words, a route leak could only occur when a customer or a peer AS exports routes that they imported from their respective neighbors. Observe that, according to $\mathcal{R}$.1, an AS can export the routes it imported from its customers toward its providers or peers, i.e., both *c* and *p* are allowed to export the routes that they imported from their customer cones toward AS *a*. Then, by using the facts $\mathcal{F}$.1 and $\mathcal{F}$.2 together, it is obvious that the possible network topologies for the occurrence of a route leak for AS *a* are the ones shown in Fig. 5(b). For the customer route leak case, *c* could leak either its peer or its provider routes to *a*. Similarly, for peer route leak scenario, *p* could leak either

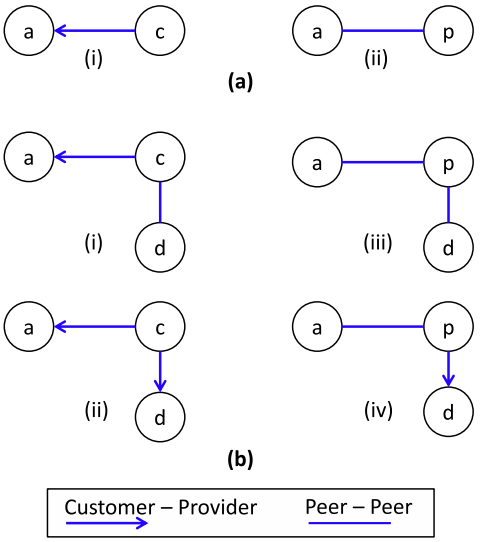


Fig. 5. (a) Possible cases for the occurrence of a route leak on AS *a*; (b) possible neighbor links of AS *a*'s customers and peers that can produce a route leak on AS *a*.

its peer or provider routes to *a*. In any route leak scenario, there are at least three ASes involved; the victim AS *V* which receives the leaked routes, the route leaker AS *L* which leaks the route, and the owner AS *O* which owns or forwarded the routes that are leaked. For example, in Fig. 5(b) (i), *a* is the victim *V*, *c* is the route leaker *L*, and *d* is the owner *O* of the routes that can be leaked.

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

is thus to independently detect route leaks despite the lack of information of its neighbors' neighbors relationships.

Let us then consider a network topology scenario for generalizing the local identification of a route leak. Fig. 7 depicts the case where our reference AS a is the victim (V) receiving new route advertisements from its neighbors. The goal is to examine under which conditions AS a can locally validate these advertisements prior to inserting them in the RIB and FIB tables of its routers. Domain b represents a neighbor that is directly connected to AS a by a peer–peer or customer–provider link, and it is the one that the victim a suspects that is responsible for leaking the routes (the leaker L). Furthermore, c (the owner O) is a direct neighbor of b, which advertises valid routes to AS b of the form $[c, \ldots]$ (where "…" refers to zero or more ASes in the AS-path). These routes can be potentially announced by b to a, e.g., through routes of the form $[b, c, \ldots]$. These announcements can be identified as leaks by the victim if they are against the valley-free rules. However, from a's perspective, the announcements cannot be validated due to the lack of information about the type of relationship between the suspect b and its direct neighbor c.

We already stated that the minimum scenario required for a route leak occurrence contemplates three actors: the victim, the leaker, and the owner of a route. However, for the sake of generality, we consider the case when the suspect b leaks a route imported from c, but that was originated by another AS, e.g., d. Thus, the potential route that AS b would leak to the victim a would be one learned from c toward d, of the form $[c, \ldots, d]$. Considering that the Internet is a connected graph, it is sound to assume that before the leak occurrence, the victim has a valid route to d, of the form $[\ldots, d]$. When the suspect AS b leaks the route to AS d to attract its traffic (i.e., AS b advertises to AS a a route of the form $[b, c, \ldots, d]$), the victim will be in a position to observe a new route advertisement for the same destination AS. This reference topology and the general assumptions that we will make next shall be used in the remainder of this Section, while formalizing the identification of route leaks in Theorems 1 and 2.

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

Remark. The purpose of our theoretical study is to capture what the victim AS can infer upon a route leak. Therefore, our analysis is focused on the transition from a valley-free valid routing state to the routing state right after the leak. Later, in Section 8, we will discuss the engineering aspects about how the victim can actually start the identification analysis from a valley-free valid state. In summary, $\mathcal{H}.1$ indicates that any route contained in the initial state of the RIBs at AS a is compliant with $\mathcal{R}.1$, $\mathcal{R}.2$ and $\mathcal{R}.3$.

Hypothesis $\mathcal{H}.2$. "An AS does not have a peer relationship with the providers of its provider."

Remark. This hypothesis is based on the assumption that a provider AS is much larger than the customer AS in terms of infrastructure. As shown in Fig. 6(a), it is very unlikely that AS x has a peer relationship with a provider of its providers, since a very large provider z will have no economical incentives for peering with a domain x at lower tiers of the AS hierarchy. On the contrary, the incentive will be to charge AS x for the transit traffic (cf. Fig. 6(a)).

Hypothesis $\mathcal{H}.3$. "A cyclic chain of provider relationships among ASes is non-existent."

Remark. This hypothesis means that we assume an Internet that is loop-free in terms of provider-customer relationships. As shown in Fig. 6(b), it is implausible that AS x is the provider of the provider of its providers. It is a common assumption in the literature that Internet topologies can be modeled as Directed Acyclic Graphs (DAGs) [20].

Now, given the valley-free rules (i.e., $\mathcal{R}.1$–$\mathcal{R}.3$), and the hypotheses defined above, we proceed to formalize the conditions for detecting peer route leaks (cf. Fig. 7(a)).

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

- *A direct route to a peer AS b, i.e., $[b]$.*
- *An alternative route to the peer AS b via AS b's direct neighbor AS c, i.e., a route of the form $[\ldots, c, b]$.*

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

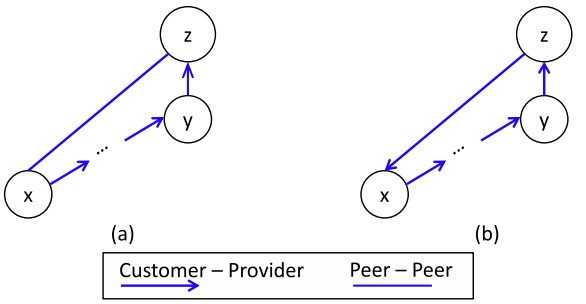


Fig. 6. Unlikely AS relationships among ASes: (a) Hypothesis 2. (b) Hypothesis 3.

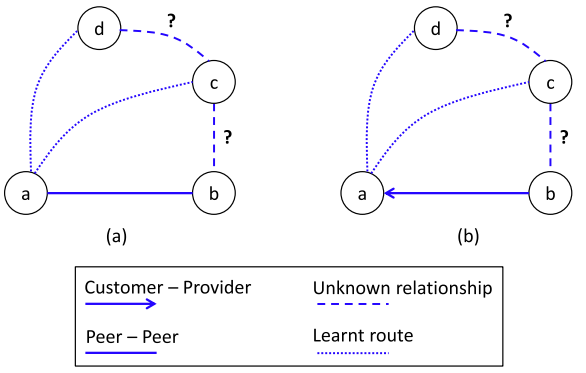


Fig. 7. Generalized topologies for route leak detection: (a) Peer route leak. (b) Customer route leak.

Proof. According to $\mathcal{R}.1$–$\mathcal{R}.3$, AS b could only advertise a route with AS-path $[b, c, \ldots]$ to AS a, iff, AS c is a customer of AS b. This is because if AS c is a peer or provider of AS b, then AS b is not allowed to advertise routes learned from AS c to its peer AS a. Let us suppose then that AS c is a customer of AS b. We know that the initial state of the routing databases at AS a contain a route to b with AS-path $[\ldots, c, b]$. Now, a could only receive the route to b with AS-path $[\ldots, c, b]$, iff, AS a belongs to the customer cone of AS c. This is because according to $\mathcal{R}.3$, c would advertise its provider routes through b only to its customers. But if a belongs to the customer cone of c, then this contradicts the hypothesis $\mathcal{H}.2$, that is, a has a peer relation with the provider of its provider. Therefore, we conclude that AS c cannot be a customer of AS b. This implies that c is either a peer or a provider of b, and therefore, the route advertised by AS b toward AS a with AS-path $[b, c, \ldots]$ is a route leak. □

To illustrate the reach and potential application of Theorem 1, let us consider again the peer route leak example given in Fig. 4(b). In practice, the route database of AS d would have a route with AS-path $[a, b, e]$ to e via b, plus the direct route $[e]$ to e in its initial state. The former is because a and b would exchange customer routes with each other. Assuming that the initial state at AS d is valley-free valid, the set up in Fig. 4(b) is under the hypotheses of Theorem 1, so AS d can autonomously conclude that the route $\mathcal{P}_1 : [e, b, c]$ received from AS e is a route leak.

We proceed now to formalize the detection of customer route leaks (cf. Fig. 7(b)).

Theorem 2. *Let the initial state of the routing databases of an AS a contain the following:*

- *A direct route to a customer AS b, i.e., $[b]$.*
- *An alternative route to the customer AS b via AS b's direct neighbor AS c, i.e., a route of the form $[\ldots, c, b]$.*

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

Proof. Just as in the proof of Theorem 1, AS b could only advertise a route with AS-path $[b, c, \ldots]$ to a, iff, c is a customer of b. This is because if c is a peer or provider of b, then b is not allowed to advertise routes learned from c to its provider AS a. Let us suppose then that c is a customer of b. We know that the initial state of the routing databases at AS a contain a route to b with AS-path $[\ldots, c, b]$. Now, a could only receive the route to b with AS-path $[\ldots, c, b]$, iff, a belongs to the customer cone of c. This is because according to $\mathcal{R}.3$, c would advertise its provider routes only to its customers. But if a belongs to the customer cone of c, then this contradicts $\mathcal{H}.3$, since there is a cyclic chain of provider relationships among a, b, and c, that is, a is a provider of b, which is a provider of c, which in turn is provider of a. We conclude that AS c cannot be a customer of AS b. This implies that c is either a peer or a provider of b. Hence, the route advertised by AS b toward AS a with AS-path $[b, c, \ldots]$ is a route leak. □

It can be shown that if the initial conditions are met, then Theorem 2 applies to the example illustrated in Fig. 3(b).

4. Cross-Path (CP) route leak identification technique

In this section, we start with one of the most straightforward approaches for detecting route leaks. In the following sections, we will incorporate additional mechanisms, which, as we shall show, will progressively improve the results in the detection. In a nutshell, the Cross-Path (CP) technique is based on the theoretical route leak countering framework described in the previous section. Algorithm 1 summarizes the step-by-step Cross-Path logic for identifying route leaks. The CP utilizes information available in the router RIBs as well as the information about the business relationships with neighbor ASes. Observe that, at the beginning of the detection process, the assumption is that the RIB tables are initially correct (i.e., they are free from entries derived by neighbor route leaks). A common solution to ensure the valley-free property of the routes is to momentarily set up route filters for all incoming BGP updates. This is only required for a short period, so as to ensure that the BGP routers only hold valley-free routes. Once the CP route leak detection technique has started, the route filters can be removed—or they can be kept though with the advantage that they neither need to be maintained nor updated. We further discuss the viability and impact of using route filters in Section 8.

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

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

```
1: if AS l ∈ N_pe ∪ N_c then
2:   for all a_i ∈ R, where 0 < i ⩽ R.length do
3:     if a_i ∈ T then
4:       R ← ∅
5:       return true
6:     end if
7:   end for
8:   R' ← [..., o, ..., l, ...]
9:   if R' ∈ RIBs then
10:    R ← ∅
11:    return true
12:  end if
13: end if
14: RIBs ← RIBs ∪ R
15: return false
```

For every incoming route advertisement from a neighbor customer or peer AS, the algorithm looks for an existing cross-path in the router RIBs considering the hypothesis and conditions outlined in the previous section. In order to make the cross-path checking more rigorous, we can generalize the cross-path check in the form $[\ldots, o, \ldots, l, \ldots]$ in the valley-free valid RIBs. This generalization follows from the proof of Theorems 1 and 2. This is because presence of any combination of the route, including $[\ldots, o, l], [\ldots, o, l, \ldots]$, $[o, l, \ldots]$, and $[o, \ldots, l]$, in the valley-free valid state of AS v while receiving a new route of the form $[l, o, \ldots]$ from AS l violates $\mathcal{H}.2$ and $\mathcal{H}.3$ for Theorems 1 and 2, respectively. In this case, a received route from a customer or a peer AS l of the form $[l, o, \ldots]$ can be declared as a route leak if the route $[\ldots, o, \ldots, l, \ldots]$ exists in the valley-free valid RIBs. If a cross-path is found, then the received route advertisement is considered a route leak and discarded, otherwise, it is included in the valley-free RIB.

Another particularity of our algorithm is that it uses the set of public Tier-1 ASes as input for detecting route leaks. Specifically, we consider the route advertisement received from a peer or customer AS a route leak if it contains a Tier-1 AS in the AS-Path. This logic is different from [11], where the author considers a route advertisement as a route leak only if it contains more Tier-1 ASes than a predefined threshold. Based on our route leak identification framework, we contend that it is highly unlikely that a AS learns a route to a Tier-1 AS or a route to any destination via Tier-1 through a neighbor customer or peer AS. In this regard, our approach is more comprehensive and encompasses the logic used in [11].

The CP technique does not incur any extra overhead, however it depends on two main procedures for its functionality. Firstly, it requires setting up the route filters for the initial period of time to ensure valley-free valid RIB. As mentioned earlier, the route filters need not to be maintained any further once the CP technique is triggered. In this manner, the maintenance cost of route filters does not become a liability when the number of routes scale up. Secondly, the AS-Path in every new incoming route advertisement, from peer and customer ASes, is checked against the valley-free valid RIB for a possible cross-path. The above two overheads of the CP technique neither pose heavy burden nor disruptive effect on the existing inter-domain routing system. Next, we describe the experimental and simulation setup used to evaluate the CP route leak detection technique along with the results obtained and their analysis.

4.1. Simulations setup and result analysis

In order to validate the CP technique, we tested it by running exhaustive simulations in a testbed with a scaled-down version of an actual Internet topology. It is worth mentioning that the cost for running event-driven simulations on a complete Internet graph is prohibitively expensive in terms of time and resources. Thus, it is a common research practice to utilize small realistic graphs, extracted from the actual Internet graph, to avoid excessive computational cost [21]. In our testing framework, an important consideration during topology reduction was to preserve some of the essential properties of the complete Internet graph, e.g., average node degree and degree distribution, so that the results obtained can be rationally extrapolated to larger topologies. Furthermore, we assured that the scaled-down topology contains all the AS relationships considered in our route leak detection framework with the potential to produce all different types of route leaks. For this purpose, we obtained a subgraph of ARK's recent Internet graph (2013) [7] containing 1650 ASes and 3744 inter-domain links using graph reduction technique presented in [21]. Observe that our approach of using a subgraph of ARK's Internet graph means that the topology we used in our simulations is actually part of the Internet. In the rest of the paper, we refer to the obtained scaled-down topology as *Topology-1650*. The simulations were setup and run using the Network Simulator NS2 [8], along with BGP++ [9] to complement NS2's lack of native BGP capabilities. We considered a single router per AS and each AS's BGP router was configured according to its policies and relationships with its neighbors according to the extracted topology. As a result, we were able to simulate and test the RLD techniques with BGP in a Internet-like topology for different route leak scenarios and evaluate their impact.

For *Topology-1650*, we identified a total of 20,747 different possible route leak scenarios, out of which 17,151 were harmful route leaks, i.e., the route leak poisoned the RIB of the victim AS successfully. That is, although the route leak occurred, the leaked routes were not chosen as they were not the best path to the corresponding destination, thus failing to poison the BGP forwarding table of the victim AS *V*. One example of such a route leak scenario is depicted in Fig. 8. Even if *L* leaks routes of *O* to *V*, these leaked routes will not affect the forwarding table of *V*. This is because on receiving routes toward *O* from *L* and directly from *O* itself, the victim *V*, following the shorter AS-Path criteria, prefers the direct shorter AS-Path route. It is important to note that the reason *V* decides the best route based on shorter AS-Path criteria is because *V* has same provider relation with both *L* and *O*.

Table 1 shows the simulation results of CP route leak detection technique for the harmful route leaks. We classify the route leaks further on the basis of the AS relationship between *L* and *O*, in order to analyze the results in depth. That is, we divide CRL cases into two subcategories, one where *O* has a provider relation with *L*, denoted by CRL (Pr), and other where *O* has a peer relation with *L*, denoted by CRL (Pe). Similarly for PRL, we classify them into PRL (Pr) and PRL (Pe) cases, where *O* has a provider and a peer relation with AS *L*, respectively. We observe that CP detects 94.11% and 93.30% of all the CRL (Pr) and PRL (Pr) route leak cases, respectively. Whereas, for the CRL (Pe) and PRL (Pe), the CP performs poorly i.e., 23.73% for CRL (Pe) and 5.90% for PRL (Pe). The reason behind better performance of CP in route leak cases where *O* is provider of *L* is that *O* being the provider of *L* advertises *L*'s route to all its providers, peers and other customers, thus increasing the chances for the possibility of cross-path observance at AS *V*. In the route leak cases where *O* is a peer of *L*, the chances of observing a cross-path involving the two

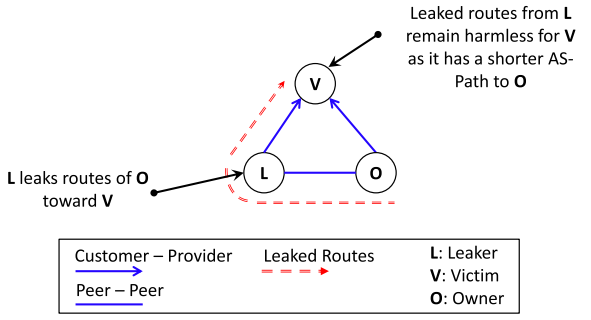


Fig. 8. General representation of one of the harmless route leak scenarios.

consecutive peers are very low in practice, since a peer does not advertise routes of another peer any further except to its customer cone, hence the poor performance of the CP technique for those cases.

4.2. Experimental setup and result analysis

Besides the simulations, we also tested the CP technique using real-time experiments. The main purpose of experimental evaluation is to verify the robustness of the technique in real environment. In order to build an experimental setup which is with in the computational and memory limits of our testbed infrastructure, we extracted a subgraph of *Topology-1650* containing 990 ASes and 2146 inter-domain links. We used similar criterion for obtaining the 990 ASes topology (referred to as *Topology-990* in the rest of the paper) as we did for *Topology-1650*, i.e., retaining certain characteristics of *Topology-1650*. Our testbed consisted of a single virtual Linux Container (Docker [10]) for each AS in *Topology-990* equipped with the well-known Quagga routing suite. The inter-domain links among the ASes were setup according to *Topology-990* as well. Initially, as determined by our hypotheses, all the nodes were connected and configured in line with the valley-free rules, hence without any route leaks. Then, for each experiment, once BGP converged, a route leak was generated, i.e., an AS (L) leaked routes of one of its neighbors (O) to another neighbor (V). Then the AS V used the CP technique to detect the route leak based on the available BGP information.

In this topology, we were able to anticipate 951 Customer Route Leaks (CRL) and 979 Peer Route Leaks (PRL) possible scenarios. Hence we ran a total of 1930 different experiments, each with one route leak occurrence. Out of the 1930 different route leak scenarios, we were able to rule out 239 leaks that were harmless. It is noteworthy that the set of route leak scenarios anticipated for *Topology-990* is a subset of route leak scenarios for *Topology-1650*. This is because *Topology-990* is a subgraph of *Topology-1650*.

In the remaining 1691 harmful route leaks, there are 810 CRL and 881 PRL route leaks. Table 1 shows the results obtained with the CP technique for the harmful route leak experiments. From the perspective of the extended classification of the route leaks, we observe a similar performance trend in the experiments as well. As shown in Table 1, the CP route leak detection performance is more than 97% in both CRL (Pr) and PRL (Pr), whereas for CRL (Pe) and PRL (Pe), it detects 21.64% and 5.71% of the route leaks, respectively. The justification of these results is similar to the one given for the experimental results of the CP technique. Furthermore, it is worth mentioning that the respective route leaks, for different cases, that are detected in our experimental study were also correctly detected in our simulation evaluation. This assures the behavior stability of our technique in both modes of evaluation. The difference in success rate percentages, for different route leak types, is due to the difference in the number of leak scenarios ran in each mode of evaluation.

5. Benign Fool Back (BFB) route leak identification technique

In the context of improving the performance of CP technique for detecting route leaks when the leaker L leaks its peer routes toward the victim V, we propose *Benign Fool Back (BFB)*. This technique exploits the commonly practiced preference of routes based on the type of relationships an AS has with its neighbors. We assume that, under normal circumstances, an AS, more specifically the leaker, follows the principle of preferring customer routes over peer and provider routes, and that it prefers a shorter AS-path route over a longer one. However, this policy might not necessarily be upheld always, such as in case of sibling AS relationships, but at least applies for a majority of them. We also assume that the ASes involved in the potential route leak incident are not using IP prefix origin verification mechanisms, such as ROA [17]. We claim that these are realistic assumptions, since most of the route leaks reported in the Internet are due to apparent misconfigurations rather than deliberate attacks, and ROA is not used by the large majority of the ASes in the Internet. To illustrate BFB, let us consider the example shown in Fig. 9(a). If an AS, the potential victim V, starts receiving

Table 1
Simple Cross-Path (CP) detection: experimental and simulation results for different route leak scenarios.

Leak scenarios	Cross path (Experiment)				Cross Path (Simulation)			
	# Leaks	# Harmful Leaks	# Leaks Detected	% Leaks Detected (%)	# Leaks	# Harmful Leaks	# Leaks Detected	% Leaks Detected (%)
CRL (Pr)[a]	725	713	701	98.31	4879	4773	4492	94.11
CRL (Pe)[b]	226	97	21	21.64	5714	3974	943	23.73
PRL (Pr)[a]	825	811	792	97.65	5724	5406	5044	93.30
PRL (Pe)[b]	154	70	4	5.71	4430	2998	177	5.90

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

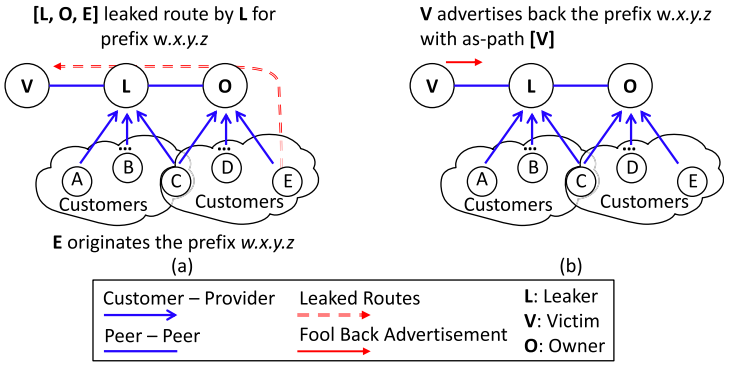


Fig. 9. Benign Fool Back: (a) L leaks O's routes to V; (b) the potential victim V sends a Fool Back advertisement to L.

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

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

The first condition can be verified by inspecting the AS-Paths in the suspicious routes received from the peer neighbor. For the second condition, V can select the destination AS, for the fool back advertisement, by making sure it does not belong to the set of ASes that L has advertised to V as its customers, i.e., the customer cone of L. This is possible because V and L has peer–peer relationship between them and by principle would exchange their respective customer routes with each other. In this framework, if V suspects this could be the result of a route leak, then V could select an IP prefix destination from the newly received suspicious routes according to the criteria defined above and advertise it back to L, that is, V could try to fool back its peer L (see Fig. 9(b)). Let us assume that V chooses IP prefix $w.x.y.z$ to fool back its peer L for identifying a route leak. Once L receives the fake advertisement for $w.x.y.z$ from V, there are two options, either L accepts this route as its best path or not. If L selects the fake advertisement from V as the best route toward IP prefix $w.x.y.z$, then it would send a withdrawal for the IP prefix $w.x.y.z$ route it sent earlier toward V. On reception of the withdrawal from L, V can infer that the route received earlier from L for $w.x.y.z$ was a leak—that is, it was a non-customer route received by V on its peering link with L. This is because if $w.x.y.z$ belongs to the customer cone of L, then L would have not selected the fake route sent by its peer V, since, according to our hypothesis, customer routes are preferred over peer routes. Also observe that, the decision of choosing candidate routes that are at least two AS hops away from L increases the chances of BFB to succeed, since thanks to the shortest-path principle, the Fool Back advertisement $[V]$ for $w.x.y.z$ will prevail over the alternative peer route $[O, E, \ldots]$ at L. The AS V can run BFB strategy for all the newly received suspicious routes by carefully selecting the fool back IP prefix to detect route leaks. Algorithm 2 shows the step-by-step working of the BFB technique.

Algorithm 2. Benign Fool Back (BFB): It allows an AS V to identify whether a new suspicious route advertisement is a leak or not.

Given: L: Leaker AS i.e., the neighbor AS from which the suspicious route advertisement is received.
$\mathcal{R}_S$: Set of suspicious route advertisements received from L
Output: **true** if the suspicious route received is a leak
false otherwise.
1: Select a route from $\mathcal{R}_S$ for a particular destination AS $\mathcal{E}$ such that:
(i) AS $\mathcal{E}$ is at least two hops away from L.
(ii) AS $\mathcal{E}$ is not a common customer of both L and its next hop AS in the suspicious route.
2: Select a prefix p belonging to AS $\mathcal{E}$
3: Advertise prefix p to L (Fool back advertisement)
4: Wait (Configurable)
5: **if** Withdrawal for prefix p is received from L **then**
6: Send prefix p withdrawal to L
7: **return true**
8: **else**
9: Send prefix p withdrawal to L
10: **return false**
11: **end if**

Let us now consider the example when the potential victim V initiates the BFB strategy on a false suspicion. For the case of PRL, even if V sends the Fool Back advertisement to the alleged leaker L, this would not prefer it over its legitimate customer route, and hence the fool back

advertisement would stay harmless in legitimate cases, as depicted in Fig. 10(b)—this is why we call this strategy "benign". In other words, the fool back advertisement would only temporarily poison the route for customers of *L* in the case that *L* had actually leaked a route to *V*. Also observe that once the withdrawal is received by the victim, it withdraws the Fool Back advertisement and it can start the remediation actions. The amount of time that a victim must wait (Step 4 in Algorithm 2) before sending the withdrawal for the fool back advertisement back to *L* needs to be set in a manner such that it allows sufficient time to receive a withdrawal from *L* as well as short enough to minimize the route poisoning of customers of *L* in case the route leak suspicion is legitimate. In our testing framework, we set the waiting time equal to the Minimum Route Advertisement Interval (MRAI) value. This allowed enough time to receive a withdrawal from *L* while minimizing the route poisoning affect on the customers of *L*. As shown in Fig. 10(a), in case of an actual leak even if *L* further forwards the poisoned route toward *O*, it will remain harmless as *E* belongs to the customer cone of *O* and it will prefer a customer route over the poisoned peer route. In the case of CRL, where *L* leaks its peer routes toward the victim *V*, the BFB is not applicable. This is because *L* is leaking peer *O* routes toward its provider *V*, thus a fool back advertisement from the provider victim will be worthless in presence of existing peer routes. Finally, in terms of the overhead, the burden caused by the BFB technique on the BGP control-plane includes suspicious route identification and advertisement and withdrawal of the fool back prefix. For BFB, the suspicious route identification requires the valley-free valid RIB and the new incoming route from a peer as input to verify if the AS ever had any route to the advertised destinations through that particular peer AS or not in order to trigger the detection. Given the inputs, the suspicious route identification procedure is self-learning in nature and hence does not pose any administrative cost, however it does add to the processing burden.

5.1. Simulations result and analysis

For evaluating the BFB technique, we use the same simulation setup and different route leaks scenarios. As shown in Table 2, the BFB detection technique proves to be very useful in PRL (Pe) case as it improves the route leak detection success rate almost sixfold compared to the CP technique, i.e., from 5.90% to 34.95%. However, we consider 34.95% yet low detection success rate and the main reason for low performance is that, when *L* leaks the routes learned from a peer, the number of routes announced are far less than when the leaked routes are from a provider. Furthermore, these routes provide reachability to a narrower stub-like block of the Internet compared to the routes received from the latter, thus, observance of cross-path is less likely. Another reason for yet low route leak detection success rate is because BFB technique is applicable to only those route leak scenarios which fulfill the conditions 1.(i) and 1.(ii) given in Algorithm 2. As expected, the BFB detection technique does not help in CRL (Pe) route leak cases as the BFB technique fails if *V* is the provider of *L*.

5.2. Experimental results and analysis

We verify the performance of the BFB technique in the same experimental testbed as described in Section 4.2. Table 2 shows the route leak detection results for CP and BFB combined. We can observe that the BFB technique improves the route leak detection success rate for PRL (Pe) to 25.71% which remains poor and the reasons are the same as discussed above in the simulation case. In

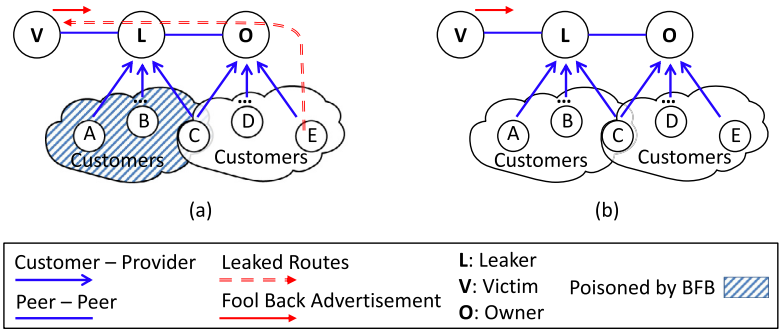


Fig. 10. Route poisoning impact of Benign Fool Back: (a) Valid suspicion; (b) false suspicion.

Table 2
Cross Path + BFB Detection: Experimental and Simulation results for different route leak scenarios.

Leak scenarios	Cross Path + BFB (Experiment)			Cross Path + BFB (Simulation)		
	# Harmful Leaks	# Leaks Detected	% Leaks Detected (%)	# Harmful Leaks	# Leaks Detected	% Leaks Detected (%)
CRL (Pr)[a]	713	701	98.31	4773	4492	94.11
CRL (Pe)[b]	97	21	21.64	3974	943	23.73
PRL (Pr)[a]	811	792	97.65	5406	5044	93.30
PRL (Pe)[b]	70	18	25.71	2998	1048	34.95

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

addition, the same 18 route leaks scenarios were successfully detected out of these 70 particular PRL (Pe) cases in our simulation tests for the BFB technique.

6. Reverse Benign Fool Back (R-BFB) route leak identification technique

In the previous sections, we presented two route leak detection techniques and showed through simulations and real-time experiments that different type of route leaks in different scenarios can be detected with a reasonable success rate by using BGP intelligence available at the control-plane level only. In order to further improve the route leak detection performance, we propose to use data-plane traffic intelligence along with the control-plane in Reverse BFB. The R-BFB targets to improve the route leak detection in PRL as well as CRL cases where L and O have a peer relation. As self-explanatory from the name, this technique is based on the BFB technique described in the previous section, however the "reverse" means that it is the O who initiates the benign fool back advertisement and tries to detect a route leak occurrence. Furthermore, R-BFB utilizes both control-plane and data-plane information to counter route leaks. If an AS observes traffic through one of its peer neighbor from sources that the neighbor has not advertised through BGP, then either it could be because of an unadvertised new customer of the peer neighbor or the AS might be a collateral victim of a route leak. The reason we say collateral victim is that the alien traffic received by the AS might be due to a route leaked by the corresponding neighbor (through which the AS is receiving the traffic) to one of its neighbors. R-BFB enables an AS to avoid the adverse impact of a route leak even if it is not the direct victim by using the available BGP information on both control-plane and data-plane. We explain the R-BFB technique with help of an example scenario shown in Fig. 11(a). If L leaks the routes learned from O to V, then the traffic from AS G to AS E would follow the path $[G, V, L, O, E]$. If L has not advertised routes learned from V to O, i.e., it leaked in one direction only, then O can take measures to verify if it is a collateral victim of a route leak by using R-BFB. For this purpose, O chooses an IP prefix from the unadvertised sources (i.e., AS G) and advertise them back to the AS from where it is receiving the traffic, i.e., L. The criteria to choose the unadvertised source (i.e., AS G) for the reverse benign fool back are the following.

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

On receiving a fake shorter AS-Path length route advertisement for AS G from O, if L decides to choose it as its best path toward AS G, then O can be assured that it is a collateral victim of a route leak. But unlike BFB, L will not send any withdrawals toward O for AS G routes as it never advertised them to O in the first place. However, O can still sense the change of best path at L for AS G, if it receives traffic destined for AS G from L, that is L accepted O's false reverse fool back advertisement. As shown in Fig. 11(b), the traffic from B to G gets diverted toward O because of the reverse benign fool back advertisement instead of taking the path B, L, V, G. This confirms that the traffic $[G, V, L, O, E]$ was indeed a consequence of a route leak because if AS G was a new unadvertised customer of L, then it would not have preferred the false reverse fool back advertisement over it. The R-BFB has similar line of reasoning for verifying if an unadvertised source traffic is a fallout of a route leak as BFB, however the former depends on data-plane traffic monitoring for triggering and concluding itself. Algorithm 3 shows the step-by-step working of the R-BFB technique.

It is important to note that the impact of the R-BFB, in terms of route poisoning of AS G, is temporarily confined to the customers of L only, in case of an actual route leak. This is because O sends the withdrawal for the reverse fool back advertisement toward L on receiving traffic destined for AS G from L (See steps 5–7 in Algorithm 3). In case of R-BFB, the waiting time, in Step 4 of Algorithm 3, is set to double the MRAI value to allow enough time to receive traffic destined for AS G in case L falls for the reverse fool back advertisement. In case of the false suspicion, the false reverse fool back advertisement of R-BFB gets discarded against a valid customer route and thus has no adverse affects. Unlike the BFB, the R-BFB is applicable to PRL as well as CRL in which L and O have a peer relation, as illustrated in Fig. 12. Furthermore, it is worth mentioning that unlike CP and BFB, which allow an AS to detect route leaks if the AS is a direct victim, R-BFB enables an AS to detect route leaks that are not directed at the AS but are affecting it one way or the other. The R-BFB technique poses

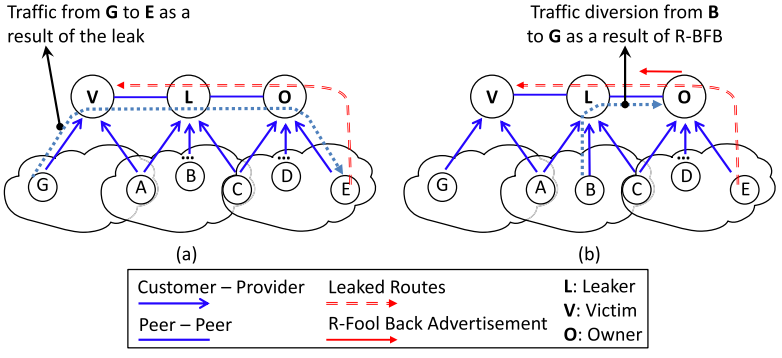


Fig. 11. Reverse BFB for PRL: (a) Traffic flow from AS G to AS E due to route leak; (b) traffic flow from AS B to AS E due to reverse fool back advertisement.

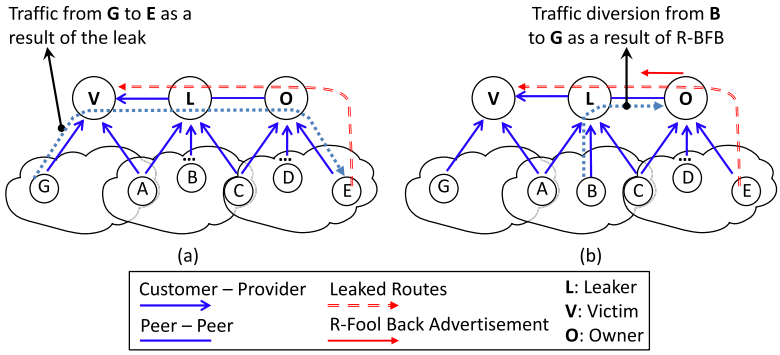


Fig. 12. Reverse BFB for CRL: (a) Traffic flow from AS G to AS E due to route leak; (b) traffic flow from AS B to AS E due to reverse fool back advertisement.

overhead on both the BGP control-plane and on the data-plane. The suspicious traffic identification requires an AS to monitor incoming traffic on the data-plane for sources that a particular neighbor customer or peer AS has not advertised through BGP. The suspicious traffic monitoring incurs heavy overhead in order to trigger the detection procedure. Furthermore, R-BFB also has the overhead of advertising and withdrawing the reverse fool back prefix on the control-plane. In comparison to the other two RLD techniques, R-BFB incurs the heaviest overhead but it enables the AS to detect route leaks in wider scenarios where CP and BFB fail. Next, we discuss the performance evaluation of R-BFB technique in our experimental testbed.

Algorithm 3. Reverse-Benign Fool Back (R-BFB): It allows an AS to identify whether it is a collateral victim of a route leak or not.

Given: L: Leaker AS i.e., neighbor AS from which the suspicious traffic is received.
$\mathcal{A}_T$: Set of ASes which are source of the alien traffic received on data-plane
Output: **true** if the suspicious traffic is due to a route leak
false otherwise.
1: Select an AS $\mathcal{E}$ from $\mathcal{A}_T$ such that:
(i) there is no route to AS $\mathcal{E}$ through L.
(ii) AS $\mathcal{E}$ is at least two hops away from L.
(iii) AS $\mathcal{E}$ is not a common customer of both L and the next hop AS.
2: Select a prefix p belonging to AS $\mathcal{E}$.
3: Advertise prefix p to L (Reverse Fool back advertisement)
4: Wait (Configurable)
5: **if** data-plane traffic is received from L destined for AS $\mathcal{E}$ **then**
6: Send prefix p withdrawal to L
7: **return true**
8: **else**
9: Send prefix p withdrawal to L
10: **return false**
11: **end if**

Table 3
Cross Path + BFB + R-BFB Detection: Experimental results.

Leak Scenarios	Cross Path + BFB + R-BFB Detection		
	# Harmful Leaks	# Leaks Detected	% Leaks Detected (%)
CRL (Pr)[a]	713	701	98.31
CRL (Pe)[b]	97	93	95.87
PRL (Pr)[a]	811	792	97.65
PRL (Pe)[b]	70	46	65.71

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

6.1. Experimental results and analysis

The inclusion of data-plane intelligence provides an extra pair of eyes for detection of route leaks in different scenarios. For the same set of route leak experiments as in Sections 4.2 and 5.2, Table 3 shows the route leak detection results of R-BFB technique on top of CP and BFB. R-BFB improves the detection success rate for both CRL (Pe) and PRL (Pe) scenarios from 21.64% to 95.87% and 25.71% to 65.71%, respectively. For the PRL (Pe) case, the R-BFB was only applicable to 46 route leak scenarios as in the rest of the scenarios the victim AS did not have customers. Similarly, for CRL (Pe), R-BFB could not be used in 4 route leak scenarios due to absence of customers at the victim AS. With these results, we contend that intelligence from both control-plane and data-plane provide enough information to detect most of the route leaks.

We did not test the R-BFB technique using our simulation environment because NS2 does not allow emulation of data-plane. Furthermore the BGP++ implementation in NS-2 only simulates the BGP control plane, but without enforcing the routing rules to the nodes, thus not allowing the generation of regular traffic through the paths as learned by BGP. Thus for R-BFB, we confine our study to the experiments performed using *Topology-990* in this paper.

7. Route leak problem in sibling–sibling relations

In this section we analyze the route leak problem in the context of sibling AS relationship. Any two different ASes

are said to have a sibling–sibling relation among themselves if they are under the administration of a single organization. For example, if a larger ISP acquires a smaller ISP with a distinct ASN or extends its network under a different ASN, then the relationship between the two ASes, now under the same administration, is called a sibling–sibling relationship, i.e., they are the children of the same 'mother' organization. In the sibling–sibling relation, the ASes typically offer transit to each other, i.e., sibling ASes can exchange their provider, peer and customer routes between themselves. The main reason for analyzing route leak problem in sibling relationship case separately from customer–provider and peer–peer relationships is because the valley-free route re-advertisement model, stated in Section 3, does not encompass the former relationship case. Although, there are no hard and fast rules governing the re-advertisement of routes learned from sibling AS, the profit optimization goal of a service provider can be used to draw out economically compelling guidelines.

7.1. Defining route leak problem for sibling routes

In the case of sibling relation, collective (i.e., for both ASes) revenue optimization has to be considered, as the two ASes are owned by the same organization. In that perspective, let us analyze different possible re-advertisement scenarios of sibling routes. Fig. 13, shows a sibling–sibling relation between AS c and AS d. As shown in the figure, d forwards its provider route for prefix $\mathcal{P}_1 : [d, b]$ toward its sibling c. Now, if c further re-advertises the route for prefix $\mathcal{P}_1 : [c, d, b]$ to its provider AS a, then a would prefer the route it learned from customer c over the route $\mathcal{P}_1 : [b]$ it learned from its peer AS b. As a consequence, the traffic between a and b will follow the path $[a, c, d, b]$, i.e., c and d would be providing a transit between a and b for zero revenue. We recall that c and d being customers of a and b are paying a and b for transit to other networks. Thus, we can consider the re-advertisement of sibling's provider routes to ones own provider as against economic convention.

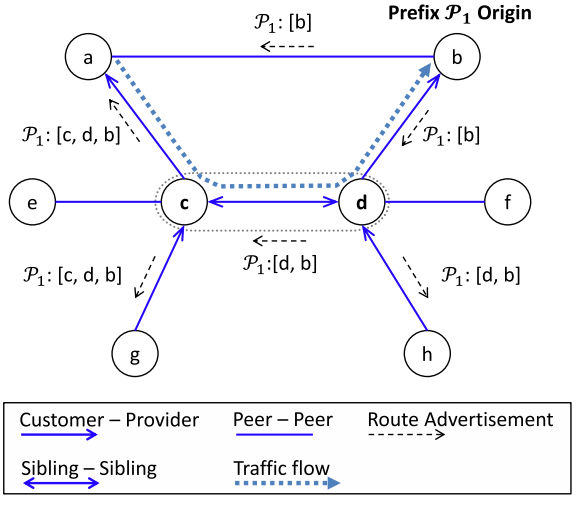


Fig. 13. Re-advertising sibling's provider routes to own provider.

Fig. 14, illustrates when if an AS forwards sibling's provider route to its peer. That is, c re-advertises its sibling's provider route for prefix $\mathcal{P}_1 : [c, d, b]$ to its peer AS e. As a result c and d will again be providing transit to traffic whose source and destination does not belong to either of them for zero revenue. Hence, the re-advertisement of sibling's provider routes to ones own peers is economically invalid as well.

Figs. 15 and 16 illustrate the re-advertisement of sibling's peer route of prefix $\mathcal{P}_2$ toward own provider and peer ASes, respectively. In both cases, the resulting traffic flows will be revenue unfriendly. Thus, the re-advertisement of sibling's peer routes to ones own provider and peer ASes is economically irrational. It is worth mentioning that re-advertisement of sibling's provider or peer routes toward own customers is economically logical as it might cause traffic between own customers and sibling's provider or peers resulting in increase of revenues.

The re-advertisement of sibling's customer routes toward its own provider, peer and customers seems economically prudent as it will cause revenue generating traffic flows, as illustrated in Fig. 17.

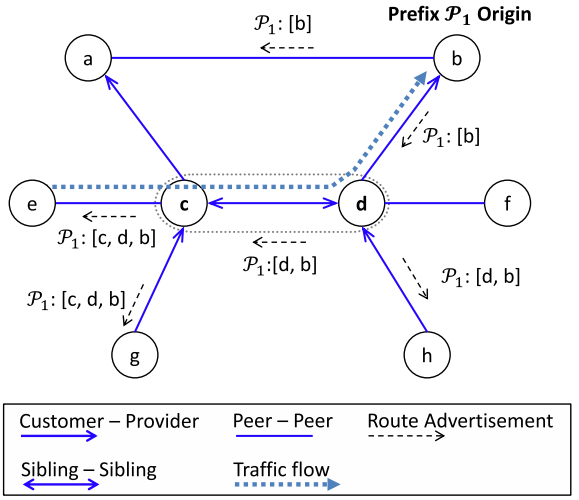


Fig. 14. Re-advertising sibling's provider routes to own peer.

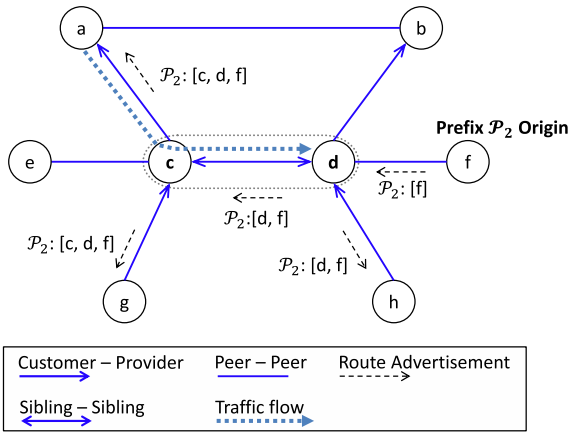


Fig. 15. Re-advertising sibling's peer routes to own provider.

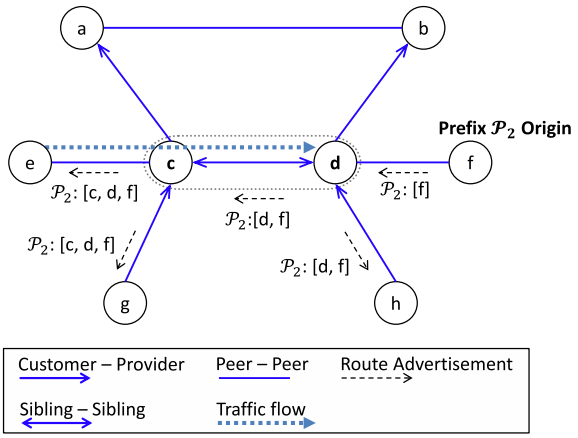


Fig. 16. Re-advertising sibling's peer routes to own peer.

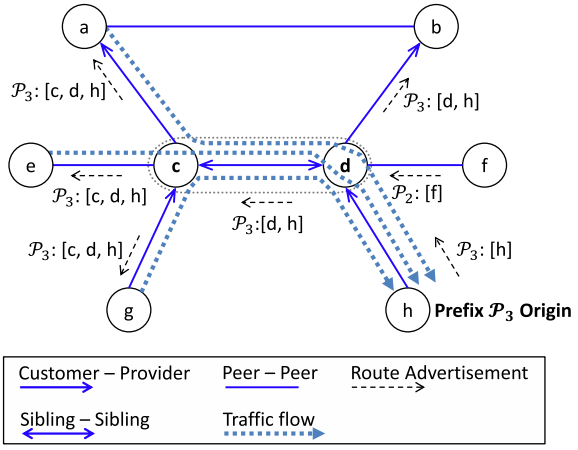


Fig. 17. Re-advertising sibling's peer routes to own customer.

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

Rule $\mathcal{R}$.4. "Sibling's customer routes can be further re-advertised to own customers, peers and providers."
Rule $\mathcal{R}$.5. "Sibling's peer routes can be further advertised to own customers only."
Rule $\mathcal{R}$.6. "Sibling's provider routes can be further advertised to own customers only."

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

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

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

7.2. Route leak detection for sibling routes

The RLD techniques described in Sections 4–6 cannot be directly applied for detecting sibling route leaks given the inherent nature of the sibling AS relationship. We explain this point with the help of an example. Let us consider the network given in Fig. 17 and assume, for traffic engineering purposes, that either c does not advertises its customer g directly to its provider a or withdraws the route $[c, g]$ from a. However, it does advertises its customer g to its sibling d which in turn advertises to its provider b. In such a situation, the following routes with corresponding AS-Paths can be observed in the RIB of a including;

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

We can observe a cross-path between $[c, d]$ and $[d, c]$ for two different set of prefixes. The cross-path technique (cf. Algorithm 1) described in Section 4 would fall prey to false positive and output route leak detected. This happens due to lack of prior sibling relation information and as a consequence the CP technique treats c and d as two separate entities. Similarly, for BFB and RBFB, lack of information of sibling relation between c and d makes the application of those route leak detection techniques doubtful. This is because, having sibling relation among any two ASes allows them to implement complex traffic engineering policies, which cannot be anticipated to any point of certainty, thus making it difficult to detect route leaks.

In this section, we discussed the route leak problem in the presence of a sibling relationship. However, we contend that it is important for an AS to have prior information of sibling relationships in order to detect sibling route leaks. For example, prior knowledge of sibling relationships in the CP technique can enable it to detect sibling route leaks. The advance information of sibling relationships will allow the CP technique to treat the two sibling members as one entity, thus avoiding any false positives.

The assumption of beforehand information of ASes which have sibling–sibling relation is not irrational. If not automated, manual efforts can be made to build up a set of sibling ASes by utilizing the information available in online databases such as IRR. Although we contend that the BGP policy information available in IRRs is unreliable and not up-to-date, it is reasonable to extract sibling information based on the owner organization as it changes less frequently compared to the BGP policies ([22]).

8. Open issues

Even though our proposals can be applied in many practical situations (e.g., the Dodo-Telstra incident could have been avoided), there are still some others that might not satisfy the hypotheses of Theorems 1 and 2 given in Section 3.3, and therefore, they need further analysis. In the remainder of this Section, we discuss the reach and limitations of the contributions in this paper.

Hybrid relationships: The valley-free rules for exporting routes serve as a reasonable stepping stone toward theoretically modeling the route leak problem. However, the

valley-free export rules are not necessarily satisfied under certain complex relationships between ASes, such as hybrid relationships. These latter refer to cases where two large ASes have different relationships between them at geographically different points of presence (PoP). For example, two ASes may have a customer–provider relation in one region and a peer–peer relation in another region. We contend that the analysis presented in this paper may even stay valid in various hybrid scenarios, since the routing information that is relevant for the detection is the one contained in the routers in proximity with the occurrence of the route leak—independently of the divergence on the routing views at geographically separated areas.

Route leak propagation: Observe that our analysis can only be used for detecting when a route leak is initiated. Detecting route leak propagation is far more difficult than detecting its initiation. The route leak propagation refers to the scenario where the *victim* AS receives a route leak and forwards it further to its neighbors. The *victim* AS may forward the route leak to its neighbors according to the relationship it has with them, which makes it more difficult for any AS receiving the propagated route to detect it as a route leak. An extended version of the customer route leak example presented in Section 3 is shown in Fig. 18. The AS a forwards the leaked route $\mathcal{P}_1[a, c, b, d]$ received from its customer AS c to its peer AS e, which is allowed according to $\mathcal{R}.1$–$\mathcal{R}.3$. The AS e further advertises this leaked route to its customers, including AS f. Note that neither AS e nor AS f can detect this route advertisement as a route leak, since they receive it in accordance with the relationship that they have with their corresponding neighbors. We leave the detection of route leak propagation for future research.

Initial valley-free state: From an engineering perspective, the hypothesis $\mathcal{H}.1$ is reasonably achievable by many transit domains, since route filters can be set to that end for a short period. This will ensure that the routes imported up to that stage are valley-free. Once this is guaranteed, the route filters need not be maintained and could be removed. Observe that the reluctance of providers for using filters does not lie on their initial configuration, but rather on keeping them updated. In any case, this method of applying and removing filters is challenging for very large providers, and without SIDR's solutions (cf. Section 9.1) in place, it can only be achieved through a chain of trust during filter configuration. Further research is needed on how to ensure that the initial state at the potential victims is valley-free.

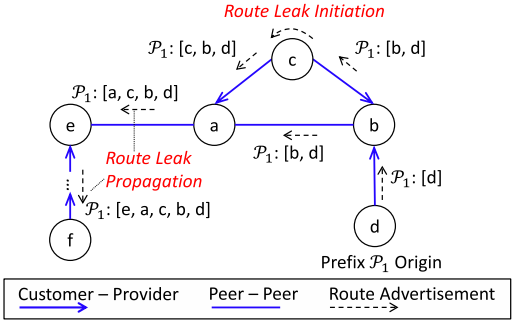


Fig. 18. Route leak initiation and route leak propagation.

9. Related work

There are very few research works which study the route leak problem in detail and propose a solution as well. Apart from the research studies, there are a few conventional methods, e.g., route filters, that can be used as a possible solution for the route leak problem. In this section, we discuss the research studies and the conventional mitigation methods that particularly target to resolve the route leak problem.

9.1. Research studies

The primary difficulty in solving the route leak problem lies in the secrecy of the AS relationships in the Internet. There are several AS relationship inference schemes proposed in the literature, including contributions such as [13,22,23]. The existing solutions typically infer the relationships between any two ASes by analyzing the BGP data collected at different points in the network, called vantage points. One fundamental critique on such inference schemes is that their knowledge base for inferring the AS relationships is partial, i.e., their view of the Internet is restricted to the data collection points. Ager et al. [24] highlight the limited nature of such AS relationship inference schemes, by detecting far higher number of peer-to-peer links within only one large Internet Exchange Point (IXP), as compared to the number of peer-to-peer links in the entire Internet discovered by well-known inference schemes.

Sundaresan et al. [25] also investigate the export policy violation attacks in inter-domain routing, calling them *traffic attraction attacks*. In essence, to detect export policy violations they exploit the valley-free path feature that a particular BGP update once traversed through a provider-customer link or a peer–peer link should not go over a customer–provider link or another peer–peer link, respectively. They propose to set a flag in the BGP advertisement when it is sent to a peer AS or a customer AS. The flag is contained in a new *ATTEST* attribute which is appended by all the ASes in the AS-Path. Furthermore, they proposed to include the *ATTEST* attribute in the signed part of the Secure BGP (S-BGP) [26] message to maintain the integrity of the flags set by each AS in the AS-Path. In this way, any AS can determine if an update received from a customer AS or a peer AS has violated the export policy rules by verifying the flags in the chain of *ATTEST* attribute. However, according to their results, their solution becomes effective when more than 60% of ASes deploy the scheme. They tested their scheme only for the stub route leak case (i.e., when a multi-home AS leaks a route learned from provider to another provider) and expected worse performance for other route leak scenarios such as peer route leak. The main shortcomings of this scheme is that it requires changes in the BGP protocol to accommodate the new *ATTEST* attribute. The scheme also depends on the Route Attestations (RA) and Address Attestations (AA) mechanisms of S-BGP [26] which incur software and hardware burden of third party security infrastructure. Furthermore, this scheme requires high deployment percentage in order to be effective for a

certain type of route leaks. And more importantly, the setting and signing of the flag in the *ATTEST* attribute discloses AS policies more than what are already revealed by the BGP protocol at present.

It is worth mentioning that the security solutions proposed by the IETF's Secure Inter-Domain Routing (SIDR) Working Group (WG) [14], namely, the Resource Public Key Infrastructure (RPKI) [16], Route Origin Authorization (ROA) [17], and Secure BGP (BGPSEC) [18] do not address the route leak problem. This is because route leaks are not covered by SIDR's solutions, since they were not included in the original agenda of the WG. Indeed, the SIDR WG has requested the Global Routing Operations WG [15] to define the route leak problem before even attempting to address it. Recently an idea of using Route Leak Protection (RLP) field inside the BGPSEC signatures to counter route leak problem is under discussion in the GROW WG [15]. The RLP field consists of two bits whose value is set by the AS sending the BGPSEC update to indicate the receiving AS if it is allowed to advertise the routes included in the update to its providers or peers. If the RLP field is set to 00 then the receiving AS can forward the update to its providers or peers and if it is set to 01 then the receiving AS is not allowed to forward the update to its providers or peers. Now, if an AS receives a update from its customer AS such that it observes 01 in the RLP field while unwinding and verifying the signature segments of all the ASes in the AS-Path, then it can consider this update as a route leak.

Let us explain the RLP working using the topology in Fig. 19. According to solution, AS_4 will put 00 while advertising its IP prefix 10.1.1.0/24 toward its provider AS_2, i.e., it allows AS_2 to further advertise the IP prefix. Now, in step (ii.b), AS_2 puts 01 while advertising the IP prefix to AS_3, i.e., disallowing AS_3 to advertise the update to its providers and peers. Now, if AS_3 leaks the route to AS_1, then AS_1 can establish it as a route leak as it will observe a 01 in the signature segment added by AS_2. The RLP solution works well for mitigating route leaks, however it suffers from similar problems as faced by the solution proposed by Sundaresan et al. [25]. In addition to the syntactical and operational changes to the BGP protocol, the RLP solution will only be effective if all ASes in the AS-Path are BGPSEC enabled. During the partial deployment tenure, the RLP solution can be deceived legitimately as the BGPSEC protocol allows BGPSEC functionality downgrade to support backward compatibility with the BGP protocol. Moreover, the RLP solution reveals AS policies more than what BGP already does. This is because in the RLP solution, an AS has to explicitly indicate and sign if the next hop is allowed or not allowed to further advertise a particular route. The BGPSEC functionality downgrade and the AS policy revelation issues raises concern on the robustness and adaptability of RLP solution for mitigating the route leak problem.

Another methodology to resolve route leaks was proposed in [19]. This method suggests to color each AS-hop in the AS-Path according to the corresponding link type, e.g., an AS-hop is "Green" if toward a provider, and is "Yellow" if toward a peer or customer. That is, a route received from a customer must have all AS-hops marked "Green" or otherwise it is a route leak. Likewise, a route received from a peer must have all AS-hops marked "Green" except the last AS-hop marked "Yellow" or else its a route leak. This coloring scheme should be used in conjunction with BGPSEC by having a signature block similar to the AS-Path signature block to avoid manipulation attacks at every AS-hop. The BGPSEC mode of implementation adds extra burden of signing and verifying the color signature block on the already resource demanding BGPSEC implementation. Consequently, this solution inherits the disadvantages and issues of BGPSEC.

9.1.1. Comparison discussion

The solutions proposed by the studies discussed above have common disadvantages including alteration of the BGP protocol, high deployment requirement to be effective, and revelation of AS policies. The first two problems put a question mark on the robustness and adaptability of the

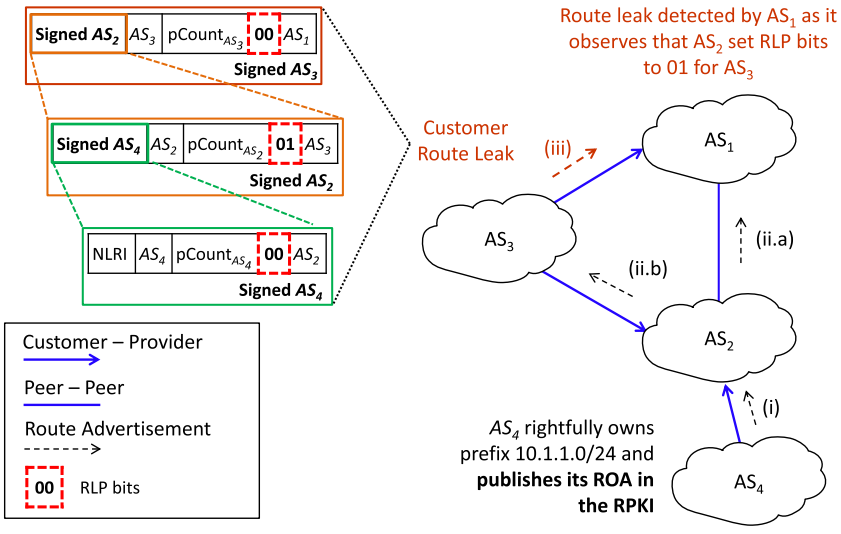


Fig. 19. Route leak on a customer link in presence of RPKI, ROA and BGPSEC.

solutions for mitigating route leaks, however it will be more difficult to convince the industrial players for the latter one, that is to earn relaxation on the confidentiality of the AS policies. In comparison, the RLD techniques, namely CP, BFB and R-BFB, do not suffer from any of these issues. One of the main reason why several past BGP security proposals did not achieve mass adoption is because they required changes to the BGP protocol which did not offer enough incentives to overcome the BGP technological inertia. The RLD techniques do not require changes to the BGP protocol instead they utilize the available BGP control and data plane information with in the AS to their advantage. Furthermore, the effectiveness of the RLD methodologies do not depend on their mass adoption, i.e., they remain potent even though if no other domain adopts them. The results from our tests show that an AS is able autonomously detect route leaks in different scenarios with a high success rate using the RLD techniques. The concealment of the AS policies is an important requirement as the service providers treat their AS policies as their business secrets. The RLD techniques also do not reveal AS policies any more than what already is revealed by the BGP protocol.

However, it is worth mentioning that the solutions discussed above target a high level of security, regardless of intentional or unintentional route leak, as they embed the security mechanism with in the BGP protocol. On the other hand, among the RLD techniques, only CP provides security against both intentional and unintentional route leaks, unlike BFB and R-BFB, which mainly target route leaks as result of misconfigurations.

9.2. Conventional methods

Overall, the conventional methods to mitigate route leaks include route filters, Internet Route Registries (IRRs), and BGP monitoring tools. The utilization of route filters on the BGP routers between two ASes aims at filtering out routes that are in violation—or are out of the scope—of the agreed policies. The timely and accurate maintenance of route filters becomes challenging as the number of allowed prefixes increase up to thousands, due to the administrative burden. As a result, the ASes prefer to rely on trust and do not maintain up-to-date prefix filters—hence saving their high maintenance cost. The YouTube incident in 2008 [2], and the Google incident in 2012 [5], could have been avoided if the route filters at the providers were effective.

The IRRs provide an online structured database of route objects that can be used to automate the maintenance of the route filters. However, IRRs also suffer from high maintenance cost because the route objects in the IRRs have to be defined first and then kept up-to-date, so the route filters can be automatically maintained. Besides, IRR records are not maintained by all ASes, and existence of duplicate, false, and incomplete records have raised questions on the sanity of the information contained in IRRs.

The BGP monitoring tools, such as Nemecis [27], Prefix Hijack Alert System (PHAS) [28], Pretty Good BGP (PGBGP) [29] and Argus [30], analyze BGP data collected at different vantage points to detect irregularities. These monitoring tools have to be trained on up-to-date policies to detect any irregularity, thus causing similar administrative burden as route filters and IRRs. Such monitoring tools are good as long as the irregularities are observed at the vantage points, so strategic attacks avoiding the vantage points can still succeed without detection. Both, BGP monitoring tools and AS relationship inference schemes depend on BGP data collected at different vantage points. However, the former utilize the data to detect irregularities against pre-defined policies, whereas the latter use the data to infer the business relationships and type of peering among ASes.

An interesting route leak detection solution was proposed by [11] by counting the number of predefined "Big Network" ASes in an AS-Path of a route under consideration. The set of "Big Network" ASes is composed of mostly Tier-1 ASes. This simple technique is based on the fact that an AS-Path should not contain more than two Tier-1 ASes in it. Thus, if an AS-Path contains more than the fixed threshold number (default threshold is 2) of allowed "Big Network" ASes, then it is flagged as a route leak. This solution is based on the same concept as the RLD techniques that is to utilize BGP knowledge to detect the route leaks.

9.2.1. Comparison discussion

The rudimentary solutions discussed above can be used as a first line of defense, however, they prove to be a stopgap solution and incur high administrative cost in face of scalability. In comparison to the route filtering solution, RLD techniques require route filters for an initial training period to ascertain the defined hypotheses, but their continuous maintenance is not required. Hence, the cost of administrating the route filter does not increase once the RLD techniques are triggered.

With regard to the BGP monitoring tools, the RLD techniques are self-reliant and do not rely on any third party security information or infrastructure. And for this reason they avoid the high administrative cost required to train and maintain the monitoring infrastructure up-to-date with the routing policies. Consequently, the RLD techniques also do not require any effort for trust establishment with the third party to avoid bogus information exchanges.

As mentioned earlier, the solution presented in [11] is similar in concept with the RLD techniques, however, the RLD techniques go much beyond than just using Tier-1 ASes information and take advantage of direct neighbor AS relationship, BGP control-plane and data-plane information for the purpose of detecting route leaks. One of the downsides of the "Big Network" technique is that it does not consider the local AS policies or AS neighbor relationship knowledge and thus not only it falls prey to generation of false positives, but also fails to detect route leaks which do not involve "Big Network" ASes in the AS-Path. On the other hand, the RLD techniques detect route leaks regardless of the size of the AS by using the available BGP information at hand.

10. Conclusion

In this paper, we studied a set of anomalies that threaten the security and reliability of the inter-domain routing system, which are referred to as route leaks. We

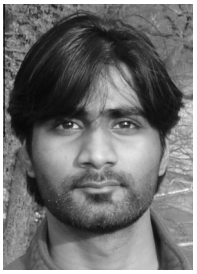
introduced a basic theoretical framework including realistic hypotheses and theorems, under which an AS is able to detect route leak initiation autonomously. The main advantages of our approach include: (a) no reliance on third party information (e.g., vantage points); (b) no changes required to control-plane protocols (e.g., to BGP); and (c) from an engineering perspective, route filters may be needed for an initial training period to ascertain the defined hypotheses, but their continuous maintenance is not required. We also provably showed with the help of real-time experiments and large scale event driven simulations that route leak detection techniques, namely CP, BFB, and R-BFB which are based on the theoretical framework described in this paper, enable an AS to autonomously detect route leak with high success rate. We also shed light on the route leak occurrences for sibling–sibling relationships. Our real-time experimental results show a high route leak detection success rate when all the three route leak detection techniques are used. However, we concede that the theoretical and pragmatic analysis presented in this paper is valid for detecting—under certain conditions—route leak initiations only. The detection of a propagated route leak requires further investigations. Further research is also needed to find ways of detecting route leaks under conditions relaxing the hypotheses of Theorems 1 and 2 (e.g., in the absence of cross paths in the RIBs for the route leaker and the route owner).

Acknowledgements

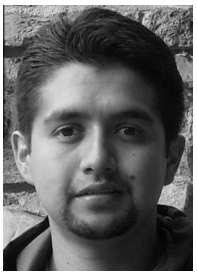
The authors would like to acknowledge the support received from the Spanish Ministry of Science and Innovation under contract TEC2012-34682, project partially funded by FEDER, the Catalan Government under contract 2009 SGR1508, the IST Open-LAB Project under contract FP7-287581, and Cisco Systems through a Cisco RFP grant.

References

[1] Y. Rekhter, T. Li, S. Hares, A Border Gateway Protocol 4 (BGP-4), RFC 4271, IETF, 2006.
[2] RIPE NCC, YouTube Hijacking: A RIPE NCC RIS case study, 2010. <http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study>.
[3] C. Labovitz, China Hijacks 15% of Internet Traffic, 2010. <http://www.arbornetworks.com/asert/2010/11/china-hijacks-15-of-internet-traffic/>.
[4] G. Huston, Leaking Routes, 2012. <http://www.potaroo.net/ispcol/2012-03/leaks.html>.
[5] T. Paseka, Why Google Went Offline Today and a Bit about How the Internet Works, 2012. <http://blog.cloudflare.com/why-google-went-offline-today-and-a-bit-about>.
[6] M.S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracia, X. Masip-Bruin, Route leak identification: a step toward making inter-domain routing more reliable, in: 10th International Conference on Design of Reliable Communication Networks (DRCN 2014), Ghent, Belgium.
[7] The CAIDA UCSD IPv4 Routed/24 Topology Dataset – 01.04.2014, 2014. <http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml>.
[8] The Network Simulator – NS-2, 2014. <http://www.isi.edu/nsnam/ns/>.
[9] BGP++, 2014. <http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/>.
[10] What is Docker?, 2014. <https://www.docker.com/whatisdocker/>.
[11] Detecting Route Leaks by Counting – NANOG 41, 2007. <https://www.nanog.org/meetings/nanog41/presentations/mauch-lightning.pdf>.
[12] R. Crozier, J. Hutchinson, Dodo Cops Blame for National Internet Outages, 2012. <http://www.itnews.com.au/News/291364,dodo-cops-blame-for-national-internet-outages.aspx>.
[13] L. Gao, On inferring autonomous system relationships in the Internet, IEEE/ACM Trans. Netw. 9 (2001) 733–745.
[14] Secure InterDomain Routing (SIDR) Working Group IETF, 2013. <http://datatracker.ietf.org/wg/sidr/>.
[15] Global Routing Operations (GROW) Working Group IETF, 2013. <http://datatracker.ietf.org/wg/grow/>.
[16] M. Lepinski, S. Kent, An Infrastructure to Support Secure Internet Routing, RFC 6480, IETF, 2012.
[17] M. Lepinski, S. Kent, D. Kong, A Profile for Route Origin Authorizations (ROAs), RFC 6482, IETF, 2012.
[18] M. Lepinski, BGPSEC Protocol Specification, draft-ietf-sidr-bgpsec-protocol, 2013.
[19] B. Dickson, Route Leaks – Requirements for Detection and Prevention thereof, draft-dickson-sidr-route-leak-reqts, 2012.
[20] B. Hummel, S. Kosub, Acyclic type-of-relationship problems on the Internet: an experimental analysis, in: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, IMC '07, ACM, New York, NY, USA, 2007, pp. 221–226.
[21] V. Krishnamurthy, M. Faloutsos, M. Chrobak, J.-H. Cui, L. Lao, A.G. Percus, Sampling large Internet topologies for simulation purposes, Comput. Netw. 51 (2007) 4284–4302.
[22] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, G. Riley, AS relationships: inference and validation, SIGCOMM Comput. Commun. Rev. 37 (2007) 29–40.
[23] L. Subramanian, S. Agarwal, J. Rexford, R. Katz, Characterizing the Internet Hierarchy from Multiple Vantage Points, Technical Report, Berkeley, CA, USA, 2001.
[24] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, W. Willinger, Anatomy of a large European IXP, in: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '12, ACM, New York, NY, USA, 2012, pp. 163–174.
[25] S. Sundaresan, R. Lychev, V. Valancius, Preventing Attacks on BGP Policies: One Bit is Enough, Technical Report GT-CS-11-07, Georgia Institute of Technology, 2013.
[26] S. Kent, C. Lynn, J. Mikkelson, K. Seo, Secure border gateway protocol (S-BGP), IEEE J. Sel. Areas Commun. 18 (2000) 103–116.
[27] G. Siganos, M. Faloutsos, Analyzing BGP policies: methodology and tool, in: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1640–1651.
[28] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, L. Zhang, Phas: a prefix hijack alert system, in: Proceedings of the 15th Conference on USENIX Security Symposium, USENIX-SS'06, vol. 15, USENIX Association, Berkeley, CA, USA, 2006.
[29] J. Karlin, S. Forrest, J. Rexford, Pretty good BGP: improving BGP by cautiously adopting routes, in: Proceedings of the 2006 14th IEEE International Conference on Network Protocols, 2006. ICNP '06, pp. 290–299.
[30] X. Shi, Y. Xiang, Z. Wang, X. Yin, J. Wu, Detecting prefix hijackings in the Internet with argus, in: Proceedings of the 2012 ACM Conference on Internet Measurement Conference, IMC '12, ACM, New York, NY, USA, 2012, pp. 15–28.



Muhammad Shuaib Siddiqui is a research associate as well as a Ph.D. candidate at Technical University of Catalonia (UPC), Spain. He received his B.Sc in Computer Engineering from King Fahd University of Petroleum & Minerals (KFUPM), Saudi Arabia, and M.Sc in Communication Systems Engineering from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Currently, he is a PhD candidate working both with the Networking and Information Technology Lab (NetITLab), and the Advanced Network Architectures Lab (CRAAX) at UPC. His research interests include Network Security, Inter-Domain Routing Protocols, and Performance Evaluation.



Diego Montero received his B.Sc. in Computer Engineering from University of Cuenca (UDC), Ecuador. He completed his M.Sc. in Computer Architecture, Networks and Systems (CANS) from Technical University of Catalonia (UPC), Spain. He is currently a Ph.D. candidate at the Networking and Information Technology Lab (NetITLab), where his research interests include Network Security, Software Defined Networking (SDN) and Cloud Computing.

Rene Serral-Gracia received his degree in computer science (2003) and a Ph.D. (2009) from the Technical University of Catalunya (UPC). He is the R&D head of the Networking and Information Technology Lab (NetITLab) at UPC, where he is leading different research initiatives, including projects under the European FP7 Research Framework as well as with industry. He is also an Associate Professor at the Department of Computer Architecture at UPC. His research interests are focused on Software Defined Networks (SDNs), overlay networks, network security, routing optimization, and QoE assessment of multimedia traffic.

Marcelo Yannuzzi received a degree in Electrical Engineering from the University of the Republic, Uruguay, and the MSc. and Ph.D. degrees in Computer Science from the Department of Computer Architecture (DAC), Technical University of Catalonia (UPC), Spain. He is the head of the Networking and Information Technology Lab (NetITLab) at UPC, as well as the head of the Advanced Network Architectures (ANA) research group at UPC. He is involved in several research initiatives and projects in close interaction with European and US companies and research centers. His research interests lie on Software Defined Networks (SDNs), Network-based Intelligence (NBI), outsourced computation and control of network functions, security, network management, smart orchestrations, and mobility.