Route Leaks Identification by Detecting Routing Loops

Song Li[1], Haixin Duan[2], Zhiliang Wang[2], and Xing Li[1]

[1] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China,
lisong10@mails.tsinghua.edu.cn,Xing@cernet.edu.cn
[2] Institute of Network Science and Cyberspace, Tsinghua University,
Beijing 100084, China,
duanhx@tsinghua.edu.cn,wzl@csnet1.cs.tsinghua.edu.cn

Abstract. Route leaks have become an important security problem of inter-domain routing. Operators increasingly suffer from large-scale or small-scale route leak incidents in recent years. Route leaks can redirect traffic to unintended networks, which puts the traffic at risk of Man-in-the-Middle attack. Unlike other security threats such as prefix hijacking that advertises bogus BGP route, route leaks announce routes which are true but in violation of routing policies to BGP neighbors. Since the routing policies are usually kept confidential, detecting route leaks in the Internet is a challenging problem. In this paper, we reveal a link between routing loops and route leaks. We find that some route leaks may cause routing loops. Hence detecting routing loops is expected to be able to identify route leaks. We provide theoretical analysis to confirm the expectation, and further propose a detection mechanism which can identify the leaked route as well as the perpetrator AS. Our mechanism does not require information about routing policies. It passively monitors BGP routes to detect route leaks and hence it is lightweight and easy to deploy. The evaluation results show that our mechanism can detect a lot of route leaks that occur in the Internet per day.

Key words: AS relationship, Routing policies, Route leaks, Routing loops, Identification

1 Introduction

Border Gateway Protocol (BGP) is a path-vector routing protocol which undertakes the exchange of reachability information between Autonomous Systems (ASes). While BGP is crucial to the Internet, it is often under threats of attack and misconfiguration due to lack of built-in security mechanism. Among the threats, prefix hijacking has been considered the main security problem. Prefix hijacking can take over the victim's IP prefix by advertising bogus BGP routes. In order to prevent prefix hijacking, a number of solutions [18, 24, 26, 20] have been proposed to ensure the correctness of BGP routing messages.

In this paper, we discuss another important BGP security problem: route leaks, which draw the attention of many researchers recently [28, 15]. Different

from prefix hijacking, route leaks do not advertise bogus BGP routes, but leak routes in violation of routing policies to BGP neighbors. In other words, in a route leak, the content of the leaked route is true, but the propagation of the route is erroneous.

Routing policies are usually used to control the chosen and propagation of BGP routes. They are created based on the business relationships between ASes. In general, the business relationships are divided into three categories [12]: provider-to-customer (p2c), peer-to-peer (p2p), and sibling-to-sibling (s2s). In a provider-to-customer relationship, the customer AS pays the provider AS for traffic destined for the rest of the Internet. In a peer-to-peer relationship, the peering ASes have a settlement-free agreement which means neither AS pays the other for the traffic destined to each other and their customers. In a sibling-to-sibling relationship, the two ASes are administrated by the same organization and they can freely exchange traffic without any expenses.

Previous research [12, 13] shows that an AS commonly adopts the following import and export routing policies according to the business relationships:

- Import policy: A customer-learned route is preferred over peer-learned route over provider-learned route.
- Export policy: A customer-learned route can be exported to all neighbors; a provider-learned route or peer-learned route can only be exported to customers.

The export policy is also known as the *valley-free rule*. When an AS advertises a route that violates the valley-free rule, it can be considered a route leak. According to the neighbor's import policy, the leaked route may be selected as the new best BGP route, which will result in the relevant traffic being redirected to the leaking AS. For instance, on February 23rd, 2012, the Australian route leak incidents [16] misrouted large amount of traffic to AS38285, and led to the interruption of Internet service in the country.

As more and more route leak incidents and their serious impacts are being reported [3, 7], it becomes necessary to detect or prevent route leaks in the Internet. There are numerous BGP security proposals [18, 21, 31] so far. They have focused on detection or prevention of bogus BGP routes. However, because route leaks announce valid routes rather than bogus routes to BGP neighbor, those solutions cannot defend against route leaks [16]. Another common way to prevent route leaks is using route filter to reject the leaked routes. But as mentioned in [16], it is difficult to maintain an accurate and timely route filter in practice, especially for the larger providers.

In this paper, we reveal a link between routing loops and route leaks. According to BGP rules for route selection, when an AS receives a loop route with its own ASN in the AS-Path, the route will be ignored. However, we find that the ignored loop routes received from a peer or customer may imply that there are route leaks which have occurred in the route. We further present a mechanism which identifies route leaks by detecting routing loops. Our mechanism can monitor the route leaks that occur in the Internet without having to know

routing policies. Moreover, it can identify the leaking AS (i.e., the perpetrator AS), which is beneficial to mitigate the impact of route leaks in time.

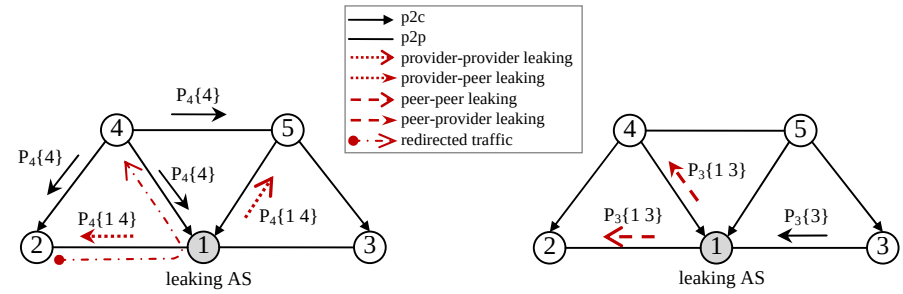
The rest of the paper is organized as follows. Section 2 provides a brief background on route leaks. In Section 3 we discuss the link between routing loops and route leaks, and present the theorems and approaches for route leaks identification. Section 4 provides the detection results of our approach. Some discussions about the detection are given in Section 5. We describe the related work in Section 6. Finally, Section 7 concludes the paper.

2 Route Leaks

Route leaks have often been discussed in the Internet community. However, there has been no exact definition of route leaks until recently [9, 28]. A route leak involves three parties: the sending AS, the leaking AS and the receiving AS. It occurs when the leaking AS mistakenly propagates the route learned from the sending AS into the receiving AS in violation of the valley-free rule. In this sense, a route leak can be expressed as an anomalous AS triple (u_{i-1}, u_i, u_{i+1}) which we call *leaking triple*, where u_i is the *leaking AS*.

According to different methods of violating the valley-free rule, route leaks can be grouped into the following four categories:

- **Provider-Provider leaking**: A provider route is mistakenly announced to another provider. The leaking pattern is p2c-c2p.
- **Provider-Peer leaking**: A provider route is mistakenly announced to a peer. The leaking pattern is p2c-p2p.
- **Peer-Peer leaking**: A peer route is mistakenly announced to another peer. The leaking pattern is p2p-p2p.
- **Peer-Provider leaking**: A peer route is mistakenly announced to a provider. The leaking pattern is p2p-c2p.



(a) Provider-Provider and Provider-Peer leaking (b) Peer-Peer and Peer-Provider leaking

Fig. 1: Four types of route leaks

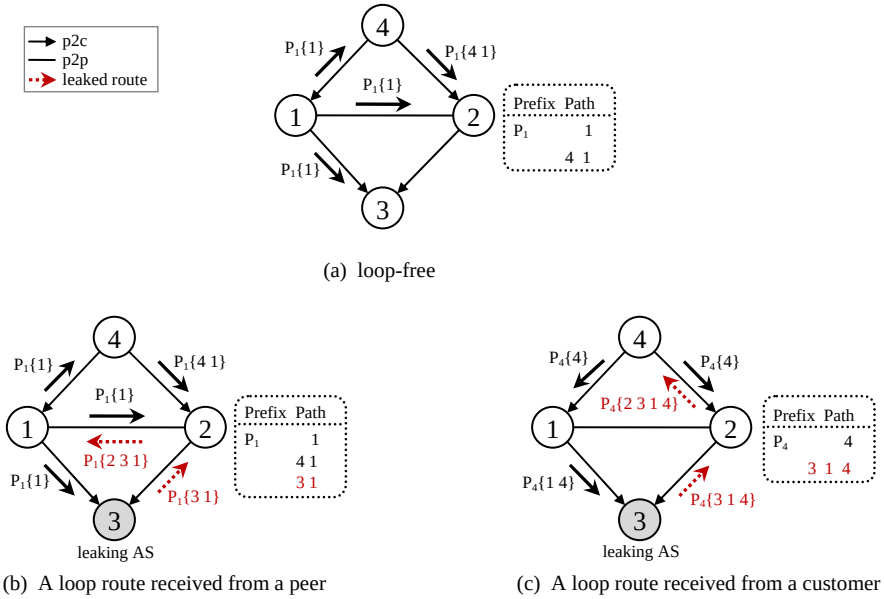
Figure 1 shows the four types of route leaks. The impact of route leaks on the receiving AS is it can lead to traffic redirection. For instance, in Figure 1(a), AS1 leaks the route learned from AS4 into AS2. According to the import policies of AS2, the leaked route (peer-learned route) is preferred over the existing BGP route (provider-learned route) in its routing table. Therefore, its traffic destined for AS4 will be redirected to AS1, which gives AS1 a chance to perform a Man-in-the-Middle (MITM) attack [23, 17].

3 Routing Loops and Route Leak Detection

Intuitively, we need to know about the AS relationships between ASes in order to identify route leaks. However, the business relationships and routing policies are often kept confidential, which makes the identification of route leaks hard. In this section, we present a novel method to detect route leaks without having to know the relationships.

3.1 Routing Loops Caused by Route Leaks

As a path vector routing protocol, BGP eliminates routing loops by checking if its own AS number (ASN) is contained in the AS-Path of received route. In general, an AS is less likely to receive a route containing its ASN in the AS-Path from its neighbors. This is because its neighbor will usually select the direct link



(a) loop-free

(b) A loop route received from a peer

(c) A loop route received from a customer

Fig. 2: Routing loops caused by route leaks

between them as the best path to it. For example, in Figure 2(a), AS1 has three neighbors and it announces prefix P_1 to them. There are two routes for prefix P_1 in the routing table of AS2. One is {1}, and the other is {4 1}. Certainly, AS2 will select {1} as the best path to its neighbor AS1 rather than {4 1}. Therefore, AS2 will not propagate {4 1} into AS1 and AS1 will not receive a route {2 4 1} that contains its own ASN.

However, that could change in a route leak case. For instance, in Figure 2(b), AS3 violates the valley-free rule and leaks the route learned from AS1 into AS2. And hence there are three routes for prefix P_1 in the routing table of AS2. Since AS3 is the customer of AS2, AS2 will select the leaked route {3 1} as the new best path to AS1 according to the common import policy. In the next step, AS2 will announce a new route {2 3 1} to AS1. And as a result, AS1 will receive a route that contains its own ASN from its neighbor AS2, i.e., it receives a route with routing loop from a peer neighbor (AS2). Similarly, in Figure 2(c), the route leak will also make AS4 receive a loop route with its own ASN in the AS-Path from a customer neighbor (AS2).

Since the above examples illustrate route leaks may cause routing loops, it is intuitively expected that detecting routing loops in the Internet may identify route leaks. We confirm this expectation below.

3.2 Route Leak Identification

First, it is important to note that the following conclusions do not consider the complex relationships such as *sibling* and *mutual transit* [12, 22]. Because the sibling ASes belong to the same organization, they can exchange routes of each other's customers, peers and providers. Therefore, as Figure 3(a) shows, the sibling relationship can result in routing loops like route leaks do. Similarly, as shown in Figure 3(b), the mutual transit AS pair provide transit service mutually, which can also lead to routing loops. We will discuss the method of distinguishing the routing loops caused by route leaks, sibling and mutual transit relationships in the next section.

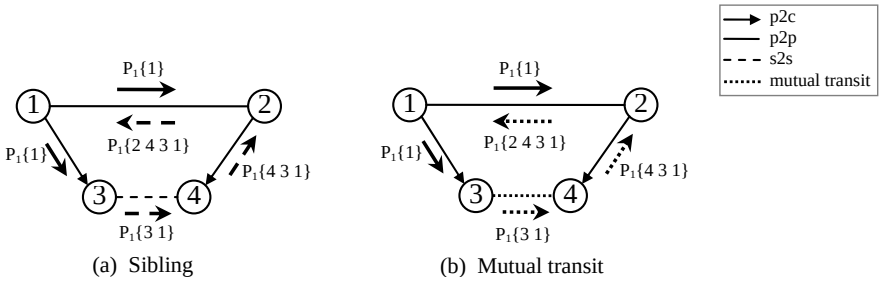


Fig. 3: Routing loops caused by complex relationships of sibling and mutual transit

Second, we introduce the definition of downhill AS-Path [12]. A downhill AS-Path $(u_1, ..., u_n)$ means that for $1 \leq i < n$, the relationship of (u_i, u_{i+1}) is p2c or s2s.

Hypothesis 1 *An AS does not have a p2p or c2p relationship with any AS behind it in a downhill path.*

This hypothesis is based on the valley-free rule and the acyclic type-of-relationship [19]. It means that if $(u_1, ..., u_n)$ is a downhill path, for $1 < i \leq n$, the relationship between u_1 and u_i cannot be p2p or c2p. Here we introduce this hypothesis to assume that the Internet AS topology is a directed acyclic graph [19]. Given the hypothesis, we present the following theorem.

Theorem 1. *Under the hypothesis 1, if an AS receives a route that is originated by itself from its peer or customer, then it can identify the route is a leaked route.*

Proof. We prove by contradiction. Suppose X and Y are BGP neighbors, and the relationship between them is p2p/p2c. If X receives a route originated by itself from Y, let us suppose the route is $\{Y, ..., X\}$. And then we get a full route propagation AS-Path $\{X, Y, ..., X\}$, which includes a routing loop originated from X.

Let us assume that the AS-Path $\{X, Y, ..., X\}$ conforms to the valley-free rule. Because $\{X, Y\}$ is a p2p/p2c link, the path $\{Y, ..., X\}$ can then only be a downhill path according to the valley-free rule. However, given that the relationship between Y and X is p2p/c2p, this means that Y has a p2p/c2p relationship with an AS behind it (i.e., X) in the downhill path $\{Y, ..., X\}$. Clearly it contradicts the hypothesis 1. Therefore, the preceding assumption that the AS-Path $\{X, Y, ..., X\}$ is valley-free is not true, i.e., the route $\{Y, ..., X\}$ is a leaked route.

Corollary 1. *Under the hypothesis 1, if an AS receives a route that contains its own ASN from its peer or customer, then it can identify the route is a leaked route.*

Proof. Similarly, suppose X receives the route $\{Y, ..., X, ...\}$, where Y is its peer or customer. According to Theorem 1, the route propagation path $\{X, Y, ..., X\}$ is not valley-free. Therefore, the propagation path $\{X, Y, ..., X, ...\}$ is also not valley-free, i.e., $\{Y, ..., X, ...\}$ is a leaked route.

Corollary 2. *Under the hypotheses 1, if a tier-1 AS receives a route that contains its own ASN, then we conclude that*

(1) The route is a leaked route.

(2) If there is only one route leak in the route, then the leaking AS is located in the loop and the route leak is a Provider-Provider leaking.

Proof. Since the route received by a tier-1 AS must come from a peer or customer, it is easy to draw the first conclusion based on Corollary 1. For the second conclusion, we suppose that X is the tier-1 AS, and $\{u_1, ..., u_n, X, ...\}$ is the route it receives. Then we have a propagation path $\{X, u_1, ..., u_n, X, ...\}$. According to

Theorem 1, the sub-path $\{X, u_1, ..., u_n, X\}$ is not valley-free, i.e., there must be route leaks occur in the loop. Consequently, if there is only one route leak in the path $\{X, u_1, ..., u_n, X, ...\}$, the leaking AS should be located in the loop path $\{X, u_1, ..., u_n, X\}$.

Next, we prove that the only one route leak is Provider-Provider leaking by contradiction. Suppose the route leak is a Provider-Peer leaking, i.e., the leaking pattern is p2c-p2p. Given that X is a tier-1 AS, the sequence of relationships in the loop path $\{X, u_1, ..., u_n, X\}$ will be $\{p2p/p2c, ..., p2c-p2p, ..., c2p/p2p\}$. According to the valley-free rule, there are at least two route leaks in the path. One is p2c-p2p, and the other occurs in $\{p2p, ..., c2p/p2p\}$. Therefore, it contradicts the precondition that there is only one route leak in the route. In the case of Peer-Peer or Peer-Provider leaking, a similar argument applies. Therefore, the route leak can only be a Provider-Provider leaking.

Hypothesis 2 *The relationship between a tier-1 AS and its non-tier-1 neighbor is p2c.*

This hypothesis is based on the fact that the tier-1 ASes are at the top of the hierarchy of the Internet. And hence in the vast majority of cases, it is reasonable that they provide transit services for their non-tier-1 neighbors.

Corollary 3. *Under the hypotheses 1, 2, if a route contains two non-adjacent tier-1 ASes, then we conclude that*

(1) The route is a leaked route.

(2) If there is only one route leak in the route, then the leaking AS is located between the two non-adjacent tier-1 ASes and the route leak is a Provider-Provider leaking.

Proof. We begin with the proof of (1). Suppose the route that contains two non-adjacent tier-1 ASes is $\{..., Y, u_1, ..., u_n, X, ...\}$, where Y and X are tier-1 ASes and u_i is non-tier-1 AS. This implies that there is a best BGP route $\{u_1, ..., u_n, X, ...\}$ in the routing table of Y.

Because tier-1 ASes peer with each other and form a full mesh topology [11], Y and X must be neighbors and their relationship is p2p. Given that u_1 is a non-tier-1 AS (i.e., it should be a customer of Y according to hypothesis 2), Y will advertise the customer route $\{u_1, ..., u_n, X, ...\}$ to X. Therefore, X will receive a route $\{Y, u_1, ..., u_n, X, ...\}$ that contains its own ASN from its peer (Y). Hence, according to Corollary 2, the route $\{Y, u_1, ..., u_n, X, ...\}$ must be a leaked route. And consequently, the route $\{..., Y, u_1, ..., u_n, X, ...\}$ is also a leaked route.

Next, we prove (2). First, let's consider the route propagation path - $\{X, Y, u_1, ..., u_n, X, ...\}$. According to Corollary 2, the leaking AS must be located in the loop $\{X, Y, u_1, ..., u_n, X\}$. Given that $\{Y, u_1\}$ is a p2c link, we can conclude that Y is not a leaking AS. As a result, the leaking AS should be located in $\{Y, u_1, ..., u_n, X\}$, i.e., between X and Y. Second, if there is only one route leak in the route $\{Y, u_1, ..., u_n, X\}$, it can be proved as in Corollary 2 that the route leak must be a Provider-Provider leaking.

3.3 Leaking AS Identification

Once a leaked route is detected, the most important thing is to identify the leaking AS to mitigate and eliminate the impact of route leaks. The Corollary 2 and Corollary 3 give the general location of the route leak. We now discuss a way to further determine the specific position of the leaking AS.

First, in Corollary 2 the route leak has been proved to be a Provider-Provider leaking, i.e., the pattern of the loop path $\{X, u_1, ..., u_n, X\}$ is: $\{p2p/p2c, ..., p2c-c2p, ..., c2p/p2p\}$. This means that the leaking AS is at the bottom of the valley path.

Second, according to [12], it is reasonable that a provider network is typically larger than its customer network and hence it is common that a provider AS has a higher degree than its customer does. To verify this point, we counted the degrees of ASes in the Internet topology derived from BGP data in Routeviews [8], and validated that 98.34% of the p2c links in the largest ground-truth data of AS relationships [22] conform to the assumption that the provider's degree is higher than the customer's degree.

Hence, on the basis of the above analysis, it is extremely likely that in Corollary 2 the leaking AS should be the AS with the lowest degree in the loop path. Similarly, in Corollary 3, the leaking AS is supposed to be the AS with the lowest degree located between the two non-adjacent tier-1 ASes.

3.4 Detection Mechanism

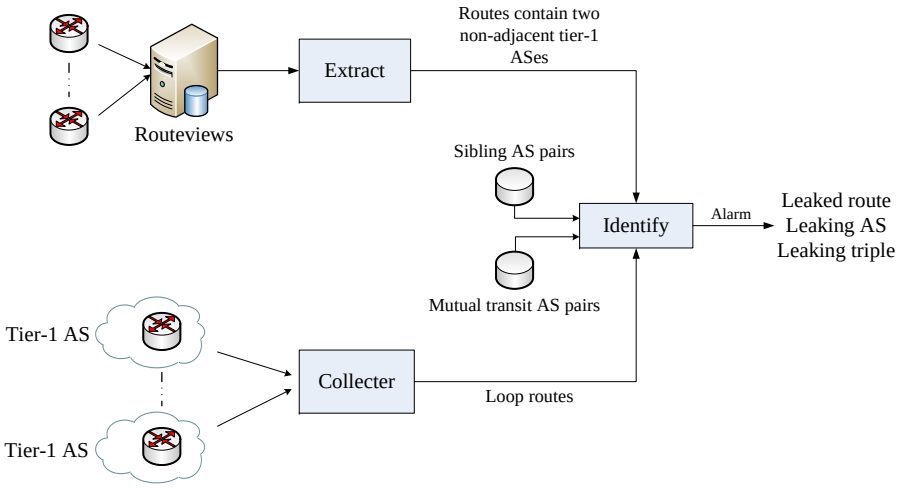


Fig. 4: Architecture of route leaks identification system

The Theorem 1 and Corollary 1 can be used to detect route leaks in an AS. The Corollary 2 and Corollary 3 can be exploited to build a distributed system

to detect route leaks that occur in the Internet. Figure 4 shows the architecture of our route leaks identification system. Our system consists of three modules: routes collection module, sibling and mutual transit inference module and leak identification module.

Algorithm 1 Route leaks detection algorithm

Input: Routes collected by Routeviews that contain two non-adjacent tier-1 ASes
Loop Routes received by tier-1 ASes
P_s: Set of sibling AS pairs
P_m: Set of mutual transit AS pairs

Output: L_r: The leaked route
L_{AS}: The leaking AS
L_{tp}: The leaking triple

if route contains two non-adjacent tier-1 ASes: $\{..., Y, u_1, ..., u_n, X, ...\}$ **then**
  extract sub-path $l : \{Y, u_1, ..., u_n, X\}$
  for $1 \leq i < n$ **do**
    if $\{u_i, u_{i+1}\} \in P_s$ or $\{u_i, u_{i+1}\} \in P_m$ **then**
      return
    end if
  end for
  $L_r \leftarrow \{..., Y, u_1, ..., u_n, X, ...\}$
  find that u_j such that $degree[u_j] = \min_{1 \; i \; n} degree[u_i]$
  $L_{AS} \leftarrow u_j$
  $L_{tp} \leftarrow \{u_{j\ \ 1}, u_j, u_{j+1}\}$
end if

if route contains routing loop: $\{X, u_1, ..., u_n, X, ...\}$ **then**
  extract the loop path $l : \{X, u_1, ..., u_n, X\}$
  for $1 \leq i < n$ **do**
    if $\{u_i, u_{i+1}\} \in P_s$ or $\{u_i, u_{i+1}\} \in P_m$ **then**
      return
    end if
  end for
  $L_r \leftarrow \{X, u_1, ..., u_n, X, ...\}$
  find that u_j such that $degree[u_j] = \min_{1 \; i \; n} degree[u_i]$
  $L_{AS} \leftarrow u_j$
  $L_{tp} \leftarrow \{u_{j\ \ 1}, u_j, u_{j+1}\}$
end if
return

1. **Routes collection module**: This module collects anomalous routes from Routeviews and tier-1 ASes. According to Corollary 3, we extract those routes that contain two non-adjacent tier-1 ASes from Routeviews. And

based on Corollary 2, we also collect loop routes received by tier-1 ASes for detecting route leaks.

2. **Sibling and mutual transit inference module**: This module is an assistant module. It infers the AS relationships of sibling and mutual transit. The inference methods are described in the next section. It should be mentioned that the inferred database will be updated periodically (one month).
3. **Leak identification module**: This module detects route leaks from the collected routes. The detection algorithm is summarized in Algorithm 1. The route will first be checked if it contains sibling or mutual transit AS pairs. If not, then it will be identified as a leaked route and the leaking AS will be further identified using the method presented above. Once the leaking AS is determined, the leaking triple is also figured out, i.e., the route leak incident is identified.

As we can see, the route leaks identification system does not need information about routing policies. It only performs passive monitoring of BGP routes to detect route leaks that occur in the Internet, and hence it is lightweight and easy to deploy.

4 Detection Results

In this section, we present the detection results of route leaks. Our route leaks identification system has been deployed since 01/01/2015. At present, the system only collects BGP routes from Routeviews. Collecting loop routes from tier-1 ASes needs to contact with their operators one by one and is a part of our future work. Nonetheless, it does not affect the evaluation of the effectiveness of our mechanism, because the detection algorithms for the two types of input data (i.e., loop routes and routes containing two non-adjacent tier-1 ASes) are nearly identical, as illustrated in Algorithm 1.

For illustrative purposes, we provide detection results of one month from 01/01/2015 to 01/31/2015. It should be mentioned that we selected the ASes in the clique inferred by [22] as tier-1 ASes. There were 471458 routes that contain two non-adjacent tier-1 ASes (we call them *T1-T1* routes) in the month. As mentioned in the above section, those routes can be caused by route leaks or complex relationships of sibling and mutual transit.

4.1 T1-T1 Routes Caused by Complex Relationships of Sibling and Mutual Transit

Our detection system used the AS-to-organization data [1] derived from WHOIS database to infer the sibling ASes. Those ASes belong to the same organization were inferred to be siblings. There were 631995 sibling AS pairs in total.

Next, the set of mutual transit ASes is inferred as follows. Suppose a route containing a tier-1 AS is $\{..., T1, u_1, ..., u_i, u_{i+1}, ..., u_n\}$, where $T1$ is the tier-1 AS. According to the heuristic algorithm described in [25, 22] (i.e., the links

seen by a tier-1 AS are p2c), the link (u_i, u_{i+1}) should be p2c. Hence, if the reverse-link (u_{i+1}, u_i) is also seen by a tier-1 AS, i.e., there exists another route $\{..., T1, u_1, ..., u_{i+1}, u_i, ..., u_m\}$ in the global routing table, the relationship of (u_i, u_{i+1}) is probably mutual transit. Note that a route leak can also results in the reverse-link (u_{i+1}, u_i) being seen by a tier-1 AS. To distinguish between them, our system picked out all the AS pairs that both their forward-link and reverse-link were seen by tier-1 ASes every day during the last month. We believe that those AS pairs are mutual transit because they lasted for one month, whereas a route leak would generally last much shorter.

Table 1: Results for the detected T1-T1 routes

T1-T1 routes	Number	Percent
Routes containing sibling ASes	31236	6.60%
Routes containing mutual transit ASes	98635	20.90%
Leaked routes	341587	72.50%

With the sibling and mutual transit data, our system filtered out the T1-T1 routes that contain sibling ASes and mutual transit ASes. Table 1 shows the results for the detected T1-T1 routes. There are less than one-third of T1-T1 routes that are caused by sibling and mutual transit relationships, and the rest of routes are identified as leaked routes. It should be mentioned that because the inferred siblings and mutual transit ASes may be incomplete, the identified leaked routes are *probable* leaked routes. Below we analyze those probable leaked routes in detail.

4.2 Analysis of Leaked Routes

As illustrated in Section 3.4, our detection system identifies the leaked route as well as the leaking AS and leaking triple. There were 268 leaking ASes and 447 leaking triples that were extracted from the 341587 leaked T1-T1 routes. As mentioned in Section 2, a route leak incident can be represented as a leaking triple. Figure 5 shows the number of leaking triples (i.e., route leak incidents) per day.

Note that not just the detected T1-T1 routes, any route containing those leaking triples in the routing table were leaked routes. To gain insight into the leaking triples, we also filtered out all routes that contain them in the month. Figure 6 shows the number of leaked routes per day. By comparing Figure 5 with Figure 6, it can be seen that there was no positive correlation between the number of route leak incidents and the number of leaked routes. This is because there are big differences in the impact of route leak incidents (i.e., how many ASes adopted the leaked routes). Some route leaks polluted quite a number of ASes in the Internet, and other route leaks only impact a few ASes.

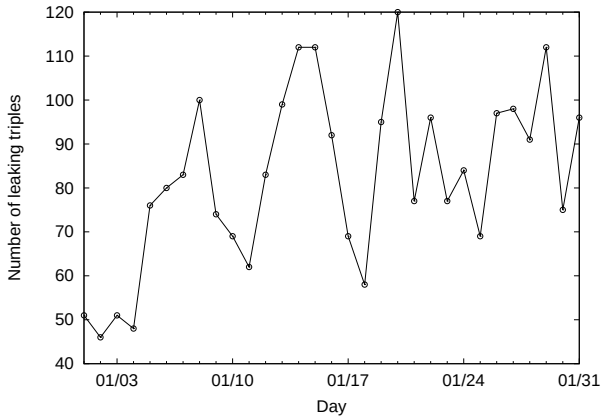


Fig. 5: Number of leaking triples (route leak incidents) per day

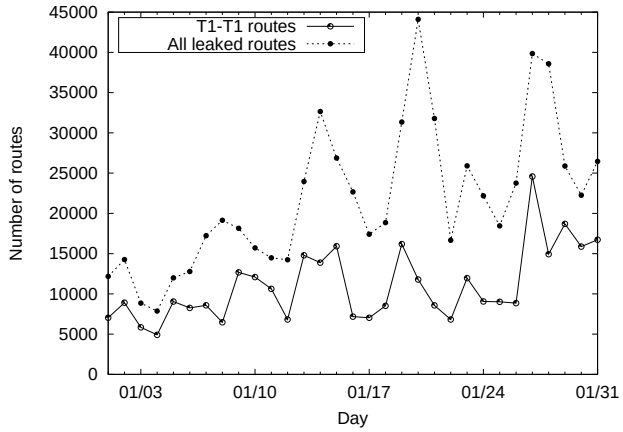


Fig. 6: Number of leaked routes per day

Since the route leak incident usually results from misconfigurations [30], the leaking triple should be an anomaly and hence it should not appear or seldom appeared in the global routing table before. To verify this, we studied the days of appearance of the leaking triples in the Routeviews last month (i.e., December 2014). As Figure 7 shows, although 61.5% of the leaking triples appeared less than 2 days, it is surprising that 28.4% of them appeared more than 5 days and 9.6% of them appeared every day last month.

We further investigated the long-term leaking triples that appeared for more than 5 days. The prefix list based filtering was found to be the major cause of the route leaks with a long persistency. Figure 8 shows an instance of long-term route leaks caused by the prefix list based configuration.

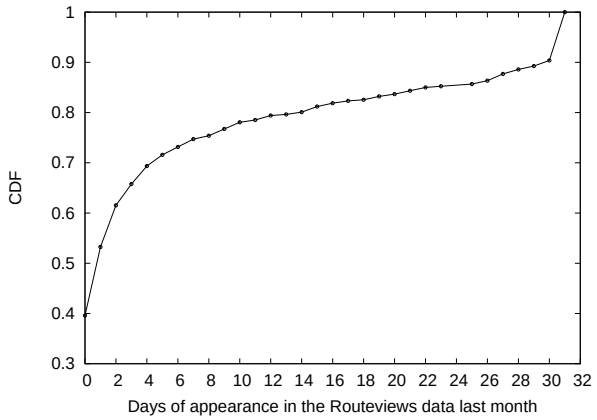


Fig. 7: Distribution of the leaking triples as a function of their days of appearance

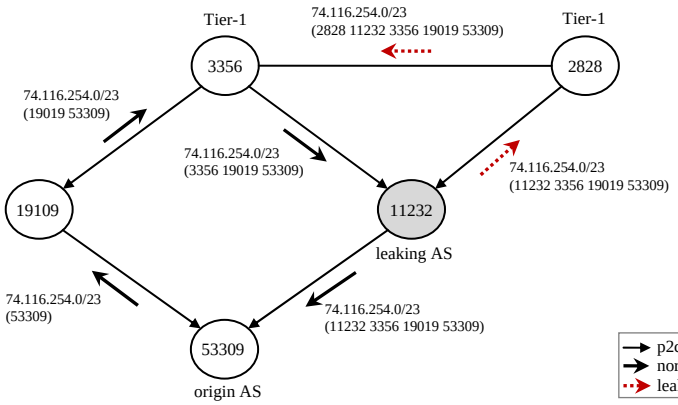


Fig. 8: An instance of long-term route leaks caused by prefix list based filtering

In Figure 8, AS53309 is multi-homed to AS19109 and AS11232, and it owns two prefixes: 74.116.252.0/23 and 74.116.254.0/23. As a provider of AS53309, AS11232 provides transit to the prefixes of AS53309. Usually AS11232 will use the prefix list of AS53309 to maintain a route filter. That is, the route with the prefix included in the list can be exported to its own providers (i.e., AS3356 and AS2828).

Most of the time, the above prefix list based filtering works properly. However, we found that in a special case of traffic engineering, the prefix list based configuration can lead to route leaks. For instance, due to possible traffic engineering policy, AS53309 did not announce the prefix 74.116.254.0/23 to AS11232

and only announced it to AS19019 during December 2014. Consequently, there was only one route destined for 74.116.254.0/23 in the routing table of AS11232, which was {3356 19019 53309}. Since the prefix of the route can pass through the route filter, AS11232 propagated the route to its upstream provider AS2828. As a result, a typical Provider-Provider leaking occurred.

Note that besides the traffic engineering, when the link between AS11232 and AS53309 fails, the prefix list based configuration can also result in the route leak above, as illustrated in [30].

It is reasonable that once the prefix list based filtering was configured, it would not be changed unless the customer updates their prefix list. Therefore, the route leaks caused by the prefix list based configuration would last a long time.

From Figure 8, we can see that the key feature of route leaks caused by prefix list based configuration is the leaking AS and origin AS are BGP neighbors. By checking if the long-term route leaks meet this condition, we found that there are about 62.2% of the long-term route leaks that can be attributed to the prefix based export configuration. The causes behind the rest long-term route leaks were hard to identify because of the confidentiality of the routing policies. And as a future work, we will do a survey of the ISP operators involved in those long-term route leaks to learn the possible causes.

5 Discussion

5.1 Loop Routes Received From a Provider

We have proved that a loop route received from a peer or customer should be a leaked route. However, some operators in the NANOG [6] mailing list provided us with several loop routes received from their providers [5]. We studied those loop routes and found that they were not leaked routes and also caused by the traffic engineering illustrated in Figure 8.

As we can see in Figure 8, there are two ASes that receive loop routes. One is AS3356, and the loop route it receives is from a peer. The other is AS53309, and the loop route is received from a provider. According to the valley-free rule, the former loop route is a leaked route and the latter is not. Hence, it can be seen from this example that when an AS receives a loop route from its provider, it cannot identify the route is a leaked route.

5.2 Complex Routing Policies

The term "route leaks" in our discussion refer to route advertisements that violate valley-free rule. However, routing policies between ASes in the Internet are sometimes complex than the valley-free rule. For example, one of the long-term leaking triples we detected is {2914 17676 209}. Since AS2914 and AS209 are tier-1 ASes and AS17676 is a non-tier-1 AS, this triple is typically in violation of the valley-free rule. But the results queried from IRR database [4] show

that AS17676 has complex routing policies which announce routes learned-from AS209 to AS2914 and its other providers. This means that although the triple {2914 17676 209} is not valley-free according to our definition, it is in a special arrangement and not a real route leak.

Therefore, it should be emphasized again that the route leaks identified by our system are advertisements in the sense of valley-free violation.

5.3 Limitations

Our detection system also has a few limitations. First, as mentioned in Section 4.1, the inferred siblings and mutual transit ASes may be incomplete, which might lead to false positives in detection of route leaks. Second, our system identifies the leaking AS by comparing the degrees of ASes. But as described in Section 3.3, it cannot be 100 percent certain that a provider AS has a higher degree than its customer does. Hence the identified leaking AS might be false in a few rare cases.

6 Related Work

While most existing work on BGP security has focused on the *correctness of routing information*, some studies have been concerned with the *correct application of routing policies*. More than a decade ago, Mahajan et. al. [30] studied the export misconfiguration (i.e., route leak) which violates the export routing policy. Then in [27, 14], the valley-free violation in inter-domain routing has been characterized and investigated. And recently, researchers formally define the advertisement of BGP routes in violation of the valley-free rule as "route leaks" [9, 28].

There are a few proposals on prevention or detection of route leaks. Qiu et. al. [27] proposed a prevention mechanism that carries pattern information of path in a transitive attribute. The new transitive attribute can be used by the receiver to determine if the advertisement is a leaked route. Although their mechanism can prevent propagating the leaked routes without revealing AS relationships, but it would fail when the attached pattern information is tampered by attackers. Another two similar approaches [29, 10] also insert a flag in the BGP route to mark the target (i.e., customer, peer or provider) of the advertisements, and they further protect the integrity of flags by using cryptographic techniques such as S-BGP [18] and BGPSEC [21]. However, they may face challenges because those cryptographic techniques will cause high resource overhead and it is far from the full deployment of them. In [28], three detection approaches are presented to identify route leaks. Although they can address different types of route leaks, but some of them require advertisements of false prefixes, which may be unacceptable for operators.

Compared to the prevention mechanisms [27, 29, 10], our approach does not require modification of BGP protocol. Moreover, unlike the detection mechanism

in [28] that can only be used by an AS to detect the leaked routes for its sake, our approach can monitor the route leaks that occur around the Internet and further identify the leaking AS and leaking triple.

7 Conclusions and Future Works

Route leaks detection is a challenging problem due to the confidential nature of business relationships and routing policies between ASes. In this paper, we studied the routing loops caused by route leaks and presented a novel mechanism that identifies route leaks by monitoring routing loops. We provided a theoretical analysis of the link between routing loops and route leaks. The theoretical analysis shows that when an AS receives a route with loop from its peer or customer, there should be route leaks that occur in the route. We further extended the theorem to the case of tier-1 ASes, and proposed a system to detect the leaked routes in the Internet. In addition to the leaked route, our system can identify the leaking AS and the leaking triple which can be helpful for mitigating and eliminating the impacts of route leak incidents in time. The detection results show that our system can discover a lot of route leak incidents that occur in the Internet per day.

As part of our future work, we will continue building the submodule of gathering loop routes from tier-1 ASes. We plan to start with those tier-1 ASes that peered with our campus network (The China Education and Research Network, CERNET [2]). We believe that once such a submodule is completed, we can detect more route leaks by exploiting those routing loops.

8 Acknowledgment

The authors would like to thank Randy Bush for his helpful comments. This work was supported by National Natural Science Foundation of China (Grant Nos. 61472215).

References

[1] The caida as organizations dataset - 20150101. http://data.caida.org/datasets/as-organizations
[2] Cernet homepage. http://www.edu.cn/english_1369/index.shtml
[3] Chinese routing errors redirect russian traffic. http://research.dyn.com/2014/11/chinese-routing-errors-redirect-russian-traffic
[4] Irr - internet routing registry. http://www.irr.net
[5] look for bgp routes containing local as#. http://mailman.nanog.org/pipermail/nanog/2015-January/072922.html
[6] Nanog. http://www.nanog.org
[7] Routing leak briefly takes down google. http://research.dyn.com/2015/03/routing-leak-briefly-takes-google

[8] University of oregon route views project. http://www.routeviews.org
[9] Dickson, B.: Route leaks – definitions. http://tools.ietf.org/html/draft-dickson-sidr-route-leak-def-03 (2012)
[10] Dickson, B.: Route leaks – requirements for detection and prevention thereof. http://tools.ietf.org/html/draft-dickson-sidr-route-leak-reqts-02 (2012)
[11] Faratin, P., Clark, D.D., Bauer, S., Lehr, W.: Complexity of internet interconnections: Technology, incentives and implications for policy (2007)
[12] Gao, L.: On inferring autonomous system relationships in the internet. IEEE/ACM Trans. Netw. 9(6), 733–745 (2001)
[13] Gill, P., Schapira, M., Goldberg, S.: A survey of interdomain routing policies. Computer Communication Review 44(1), 28–34 (2014)
[14] Giotsas, V., Zhou, S.: Valley-free violation in internet routinganalysis based on bgp community data. In: Communications (ICC), 2012 IEEE International Conference on. pp. 1193–1197. IEEE (2012)
[15] Goldberg, S.: Why is it taking so long to secure internet routing? Communications of the ACM 57(10), 56–63 (2014)
[16] Huston, G.: Leaking routes. http://labs.apnic.net/?p=139 (2012)
[17] Huston, G.: Mitm and routing security. http://labs.apnic.net/?p=447 (2013)
[18] Kent, S., Lynn, C., Seo, K.: Secure border gateway protocol (s-bgp). IEEE Journal on Selected Areas in Communications 18(4), 582–592 (2000)
[19] Kosub, S., Maaß, M.G., Täubig, H.: Acyclic type-of-relationship problems on the internet. In: Combinatorial and Algorithmic Aspects of Networking. pp. 98–111. Springer (2006)
[20] Lepinski, M., Kent, S.: An Infrastructure to Support Secure Internet Routing. RFC 6480 (Feb 2012)
[21] Lepinski, M., Turner, S.: An overview of bgpsec. http://tools.ietf.org/html/draft-ietf-sidr-bgpsec-overview-07 (2015)
[22] Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., et al.: As relationships, customer cones, and validation. In: Proceedings of the 2013 conference on Internet measurement conference. pp. 243–256. ACM (2013)
[23] McPherson, D., Amante, S., Osterweil, E., Mitchell, D.: Route-leaks & mitm attacks against bgpsec. http://tools.ietf.org/html/draft-ietf-grow-simple-leak-attack-bgpsec-no-help-04 (April 2014)
[24] Ng, J.: Extensions to bgp to support secure origin bgp (sobgp). http://tools.ietf.org/html/draft-ng-sobgp-bgp-extensions-02 (April 2004)
[25] Oliveira, R., Willinger, W., Zhang, B., et al.: Quantifying the completeness of the observed internet as-level structure (2008)
[26] van Oorschot, P.C., Wan, T., Kranakis, E.: On interdomain routing security and pretty secure bgp (psbgp). ACM Transactions on Information and System Security (TISSEC) 10(3), 11 (2007)
[27] Qiu, S.Y., McDaniel, P.D., Monrose, F.: Toward valley-free inter-domain routing. In: Communications, 2007. ICC'07. IEEE International Conference on. pp. 2009–2016. IEEE (2007)
[28] Siddiqui, M., Montero, D., Serral-Gracià, R., Yannuzzi, M.: Self-reliant detection of route leaks in inter-domain routing. Computer Networks 82, 135–155 (2015)
[29] Sundaresan, S., Lychev, R., Valancius, V.: Preventing attacks on bgp policies: One bit is enough (2011)
[30] Wetherall, D., Mahajan, R., Anderson, T.: Understanding bgp misconfigurations. In: Proc. ACM SIGCOMM (2002)

[31] Zhang, Z., Zhang, Y., Hu, Y.C., Mao, Z.M., Bush, R.: Ispy: detecting ip prefix hijacking on my own. ACM SIGCOMM Computer Communication Review 38(4), 327–338 (2008)