Improving BGP-iSec against attacks and accidents

Keishi Hashiba[1,a)] Shingo Okamura[1,2,b)]

Abstract: In recent years, BGP-iSec, a security mechanism as an extension of BGP security utilizing RPKI, has been proposed. This protocol is designed to prevent AS_PATH manipulations and route leaks, which are difficult to prevent with ROA/ROV. However, even with BGP-iSec, there are still attack and accidents that are difficult to prevent. In this study, we analyze the causes of these issues, comparing it with ASPA. We also present an improvement proposal that take into account existing issues and discuss its effectiveness.

Keywords: BGP, BGP-iSec, RPKI, Network Security

1. Introduction

The Internet consists of Autonomous Systems (ASes) interconnected through the Border Gateway Protocol (BGP), which, despite being the de facto standard, suffers from well-known security weaknesses. Incorrect or malicious routing information, such as route leaks, hijacks, or path manipulations, can lead to serious threats, including disconnection, traffic interception, or eavesdropping.

To address these problems, several RPKI-based security mechanisms have been proposed, including ROA/ROV [1], [2], BGPsec [3], and ASPA [4]. While these mechanisms strengthen BGP security, they still leave important vulnerabilities unaddressed [5], [6], [7].

To deal with this situation, Morris et al. proposed BGP-iSec [8]. Its primary goals are to provide security benefits even to early adopters—unlike BGPsec, where early adopters have few benefits—and to mitigate post-ROV attacks, which ROA/ROV cannot prevent. Moreover, BGP-iSec provides advantages in certain scenarios; for instance, it can mitigate attacks that remain unpreventable under ASPA. Although BGP-iSec can prevent certain types of route leaks and attacks, some vulnerabilities and errors remain, as also pointed out in our recent study [9]. These observations motivate us to revisit and extend the design of BGP-iSec.

In this paper, we extend the design of BGP-iSec by proposing three improvements:

- **iSec-provider**: extends validation to routes learned from provider ASes; which are not covered by the original BGP-iSec, thereby preventing additional classes of route leaks and attacks.
- **iSec-apex**: introduces validation of peer relations at the apex of a route; which strengthens iSec-provider, but its applicability may be limited in real-world peering operations.
- **iSec-dist**: validates the distance between adopting ASes to mitigate path-shortening attacks.

We further analyze how these three extensions perform in specific attack and failure scenarios that remain unpreventable under the original BGP-iSec, using a simulator we developed.

[1] Graduate School of Information Science and Technology, The University of Osaka
[2] National Institute of Technology, Nara College
[a)] u625214h@ecs.osaka-u.ac.jp
[b)] okamura@info.nara-k.ac.jp

2. Preliminaries

2.1 Border Gateway Protocol (BGP)

BGP, Border Gateway Protocol, is an inter-domain routing protocol standardized by IETF, and its most recent version is defined in `RFC 4271` [10]. The Internet consists of Autonomous Systems (ASes), and BGP is used for exchanging reachability information between ASes.

The connections between ASes are typically categorized into two types. In a *provider-customer* relation, a provider AS offers connectivity to a customer AS, and the customer AS pays fees to the provider AS. In contrast, a *peer* relation is established between two ASes of similar scale or role, in which each AS exchanges routes with the other without payments. Every AS is expected not to advertise any route learned from non-customer ASes to non-customer ASes. To simplify, we assume this model as the *Valley Free Model* [11], where the Internet consists of inter-AS connections whose relation is either a provider-customer or a peer-peer relation.

Each AS has its routing policy. BGP [10] defines several attributes for route selection, among which the `LOCAL_PREF` (Local Preference) attribute plays an important role. Operators assign higher `LOCAL_PREF` values to more preferred routes. In practice, ASes typically prioritize routes in the following order: customer, peer, and provider. This order reflects economic incentives: customer routes generate revenue, peer routes are settlement-free, and provider routes incur costs.

2.2 Security threats on BGP

Here, we outline the major security threats associated with BGP that are relevant to this paper. We do not consider attacks that can be prevented with ROA/ROV in this paper because BGP-iSec,

which we describe later, has been designed to tackle post-ROV attacks.

2.2.1 Prefix hijacking

Prefix hijacking is an attack where an AS originates IP prefixes that it does not legitimately own, thereby attracting traffic destined to those prefixes. This attack is well-known and has caused several large-scale incidents on the Internet. However, it can be mitigated by Route Origin Validation (ROV) in the Resource Public Key Infrastructure (RPKI). Since our focus is on threats that remain after deploying ROV, we do not consider prefix hijacking in this paper.

2.2.2 Path manipulation

Path manipulation occurs when a malicious AS forges the `AS_PATH` attribute and advertises it to its neighbors. Examples include announcing an `AS_PATH` that does not reflect the actual sequence of ASes traversed by the route, or shortening the `AS_PATH` by omitting intermediate ASes. Such manipulations can influence routing decisions because ASes often prefer shorter paths when other attributes are equal.

In this paper, we focus in particular on the omission of intermediate ASes from the `AS_PATH`, which can mislead route selection. We analyze this attack in detail in Sections 3 and 4.

2.2.3 Route leaks

`RFC 7908` [12] categorizes six types of *route leaks*. In this paper, we consider the following two cases: One is called *Hairpin Turn with Full Prefix*, in which an AS sends routes learned from a provider AS to another provider AS; the other is called *Lateral ISP-ISP-ISP Leak*, in which an AS leaks routes learned from a peer AS to another peer AS. Both cases violate the valley-free principle and may result in suboptimal routing or unintended traffic attraction.

2.3 Security Mechanisms for BGP

2.3.1 RPKI

Resource Public Key Infrastructure (*RPKI*) is a framework defined in `RFC 6480` [13]. It is conceptually similar to a traditional PKI, but its trust anchors are the five Regional Internet Registries (RIRs) instead of conventional certificate authorities. RPKI allows holders of Internet number resources (IP prefixes and AS numbers) to prove their ownership and authorization. Security mechanisms such as ROA/ROV and ASPA are built on top of this infrastructure. In this paper, we also consider a security proposal called BGP-iSec, introduced later, which is likewise based on RPKI.

2.3.2 BGPsec

BGPsec, defined in `RFC 8205` [3], extends BGP by adding cryptographic signatures to protect the `AS_PATH`. Although this design can prevent certain types of path manipulation, BGPsec requires high computational resources for signature generation and verification. Moreover, its effectiveness depends on a high adoption rate, which makes incremental deployment difficult. As a result, BGPsec has seen little adoption in practice.

2.3.3 ROA/ROV

A *Route Origin Authorization* (*ROA*) is an RPKI object that binds an IP prefix to an authorized origin AS [14]. Routers can perform *Route Origin Validation* (ROV) [15] based on these objects. ROV can mitigate origin hijacks—attacks in which an AS falsely claims to originate a prefix. However, it does not address threats such as path manipulation or route leaks. As of 2025, roughly half of IPv4 prefixes are covered by valid ROAs [16].

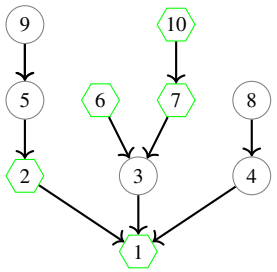


Fig. 1: An example of ProConID. Here, the ProConID-list of AS 1 is $\{2, 6, 7\}$.

2.3.4 Autonomous System Provider Authorization (ASPA)

ASPA is a proposal currently under discussion in the IETF SIDR Operations (sidrops) working group [17]. It enables each AS to publish its set of provider ASes as an RPKI object, allowing verification of the validity of provider–customer relationships along an `AS_PATH`. ASPA primarily aims to detect certain types of path manipulation and route leaks, though it cannot prevent all such cases, and some issues have been reported in practice [5], [6].

2.4 BGP-iSec [8]

Morris et al. proposed a new BGP security mechanism called *BGP-iSec*. BGP-iSec is designed to address two major problems. First, it aims to prevent post-ROV attacks, such as path manipulations, which cannot be mitigated solely with ROA/ROV. Second, it attempts to provide benefits even for early adopters, in contrast to BGPsec, whose limited incremental deployability has been noted [7].

BGP-iSec consists of three variants: Protected OTC, UP Attributes, and the Providers-Cone Identification (ProConID) list. Among these, the ProConID-list provides stronger security than the other variants. Therefore, in this paper, we focus exclusively on the ProConID-list and refer to it simply as BGP-iSec.

The ProConID-list is the list of BGP-iSec-adopting ASes within the provider cone, i.e., the set of ASes that can be first reached by traversing provider–customer links upward from a given AS. Figure 1 illustrates an example: the ProConID-list of AS 1 is $\{2, 6, 7\}$. BGP-iSec assumes that BGP-iSec-adopting ASes will know all other adopting ASes and their public keys. This assumption is referred to as the *Known Adoption and Public Keys* (KAPK) assumption.

Now, we describe the validation procedure of BGP-iSec. Assume that AS R received a route, which has the AS path $\{X_l, \ldots, X_1, X_0\}$, where X_0 is the origin AS and X_l is the neighboring AS of the AS R. BGP-iSec considers only the ASes that adopt BGP-iSec and indexes them as $i_0 < \cdots < i_p \leq l$, where $i_0 = 0$ since the origin must adopt BGP-iSec. The last index i_p equals l if the neighbor X_l adopts BGP-iSec, and $i_p < l$ otherwise.

The validation decision is made as follows:

(1) **Provider case:** If X_l is a provider of R, the route is treated as `Valid`.

(2) **Peer case:** If X_l is a peer of R, the route is `Valid` if and only

if for every BGP-iSec-adopting AS X_{i_j} with $i_0 \leq i_j < i_p$, it holds that $X_{i_{j+1}}$ is included in the ProConID-list of X_{i_j}.

(3) **Customer case:** If X_l is a customer of R, the condition is the same as in the peer case, but in addition, AS R must also be included in the ProConID-list of X_{i_p}.

It is worth noting that the routes learned from provider ASes are not validated at all in the process. This omission contributes to certain vulnerabilities in BGP-iSec.

3. Vulnerabilities of BGP-iSec

In this section, we discuss the vulnerabilities of BGP-iSec. Although BGP-iSec has been proposed to address path manipulations and route leaks, several types of these attacks still cannot be prevented. Building partially on our earlier findings [9], we examine some of the previously identified weaknesses while also exploring additional vulnerabilities that were not covered in the original study but arise in broader deployment scenarios.

In the figures throughout the rest of this paper, each node represents an AS. Hexagonal nodes indicate ASes that adopt a security mechanism, while circular nodes indicate non-adopting ASes. A red node denotes a malicious AS, i.e., an AS that may launch an attack or leak its routes. Lines indicate connections between ASes: arrows represent provider-to-customer links, while non-arrowed lines denote peer connections.

3.1 Malicious Routes from Provider ASes

As noted earlier, the original BGP-iSec design performs no validation on routes learned from provider ASes. Consequently, any malicious or improperly leaked announcements from provider ASes remain undetectable.

Figure 2 shows an example. AS 3 leaks its route, originally received from its provider AS 4, to another AS, AS 2, either intentionally or accidentally. AS 2 then propagates it to its customer, AS 1. AS 1 receives and validates the route using BGP-iSec, but cannot detect the improper path. Therefore, AS 1 accepts this inappropriate route to AS 5 as a `Valid` route.

3.2 Path Manipulations from Provider ASes

Figure 3 shows an example of such an attack. This vulnerability was discussed in our previous study [9]. In this case, AS 9 does not have any legitimate route to AS 5, but it sends AS 2 a malicious advertisement with `AS_PATH` {9, 5} and origin AS 5. However, AS 1 cannot detect this manipulation because the route is received from its provider.

Similar vulnerabilities have been reported for ASPA and ASPV [5], [6], where manipulations from providers also evade detection.

3.3 Path Manipulations from Customer ASes

Even when routes are received from customer or peer ASes, certain attacks remain undetectable under BGP-iSec.

Figure 4 shows an example attack from a customer AS. AS 4, acting as an attacker, propagates a manipulated route to its provider by omitting non-adopting ASes (AS 5 and AS 6) from the middle of the `AS_PATH` to make the path appear shorter. AS 1 receives this route but cannot detect the manipulation. Because BGP path selection generally prioritizes shorter paths when `LocPrf` values are equal, the maliciously shortened route is likely to be preferred.

This manipulation is possible because BGP-iSec only checks relationships among adopting ASes and cannot validate the presence of non-adopting ASes between two adopters, as non-adopters do not publish any security objects or security status via RPKI.

3.4 Other Security Mechanisms

We have discussed attacks and route leaks that cannot be prevented solely with BGP-iSec. However, some attacks can be mitigated by other security mechanisms. For example, in the attack shown in Figure 4, if BGP-iSec-adopting ASes publish ASPA objects and use ASPV instead of BGP-iSec, the malicious route would be marked `Invalid`, since AS 4 is not in the SPAS (Set of Provider ASes) of AS 7.

4. Improvements to BGP-iSec

We propose several improvements to BGP-iSec to address the vulnerabilities identified in the previous section.

4.1 Proposal 1: Validation of Routes Learned from Provider ASes

We refer to this extension as *iSec-provider*. The original BGP-iSec does not validate routes learned from provider ASes. We extend the validation process to also cover such routes.

Recall that the routes learned from customer/peer ASes and provider ASes differ. Routes learned from customer or peer ASes must consist solely of up-ramp paths (i.e., from customer to provider). In contrast, routes learned from provider ASes may include both up-ramp and down-ramp segments, as well as a peer connection at the apex of the route.

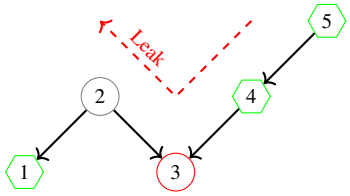
In iSec-provider, the validation process for routes learned from peer and customer ASes remains identical to that of the original BGP-iSec. Here, we define the additional process for routes learned from provider ASes, which may contain both up-ramp and down-ramp segments and potentially a peer relationship at the apex.

Let the `AS_PATH` of a route be $\{X_l, \ldots, X_p, X_q, \ldots, X_1, X_0\}$, where X_0 is the origin AS (which must adopt iSec-provider), and X_l is the neighboring AS of the receiving/validating AS. Following the original BGP-iSec, we only consider the ASes that adopt iSec-provider, indexed as $i_k, \ldots, p, q, \ldots, 0$. Here, $i_k = l$ if the neighboring AS X_l also adopts iSec-provider; otherwise $i_k \neq l$.

The validation process of iSec-provider classifies routes into three categories: `Valid`, `Invalid`, and `UnknownApex`.

For routes learned from customer and peer ASes, the validation process remains identical to the original BGP-iSec. A route is valid if and only if, for every iSec-provider-adopting AS X_{i_j} with $q > i_j \geq 0$, the next AS $X_{i_{j+1}}$ is contained in the ProConID-list of AS X_{i_j}. Otherwise, the route is `Invalid`. Importantly, routes learned from a peer or customer AS will never be labeled as `UnknownApex`.

For routes learned from provider ASes, however, this process will generally fail, since such routes typically contain a down-ramp segment. Specifically, if AS X_p is not in the ProConID-list of AS X_q, the validation fails. In this case, the apex of the `AS_PATH`

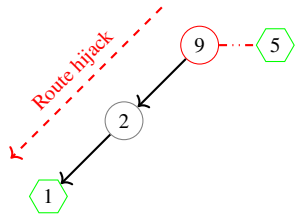


(a) Experiment

```
== AS information ==========================================================================
AS Number   : 1
Prefix      : 192.168.1.0/24
Connections : Providers : {2}
Routes      :  Destination         | LocPrf | Isec     | AS_PATH (adj > ... > origin)
               --------------------+--------+----------+-----------------------------
               * 192.168.2.0/24    | 90     | ----     | 2
               --------------------+--------+----------+-----------------------------
               * 192.168.3.0/24    | 90     | ----     | 2 > 3
               --------------------+--------+----------+-----------------------------
               * 192.168.5.0/24    | 90     | Valid    | 2 > 3 > 4 > 5
============================================================================================
```

(b) Simulation result

Fig. 2: Leaked route propagated via a provider AS



(a) Experiment

```
== AS information ==========================================================================
AS Number   : 1
Prefix      : 192.168.1.0/24
Connections : Providers : {2}
Routes      :  Destination         | LocPrf | Isec     | AS_PATH (adj > ... > origin)
               --------------------+--------+----------+-----------------------------
               * 192.168.2.0/24    | 90     | ----     | 2
               --------------------+--------+----------+-----------------------------
               * 192.168.5.0/24    | 90     | Valid    | 2 > 9 > 5
               --------------------+--------+----------+-----------------------------
               * 192.168.9.0/24    | 90     | ----     | 2 > 9
============================================================================================
```

(b) Simulation result

Fig. 3: Undetectable path manipulation from a provider AS

must lie between AS X_p and AS X_q, and may coincide with either of them.

Hence, the validation process is reversed as follows: After AS X_p, the route is valid if, for every iSec-provider-adopting AS X_{i_j} with $i_k \geq i_j > p$, AS $X_{i_{j-1}}$ is contained in the ProConID-list of AS X_{i_j}. Even if no ProConID-list validation fails, the route may still be labeled `UnknownApex`, depending on the validation result between AS X_p and X_q. We describe this later.

Figure 5 shows an example of a route leak that can be prevented with iSec-provider. Assume that AS 1, 4, and 5 adopt iSec-provider and publish RPKI objects correctly. AS 3 leaks a route originated by AS 5, and since AS 2 cannot detect this, it propagates the route to AS 1. AS 1 validates the route with iSec-provider, but the first ProConID-list validation failure occurs at AS 5, because the ProConID-list of AS 5 is ϕ. Thus, the apex of this route is expected to lie between AS 5 and the next AS, AS 4. However, ProConID-list validation also fails at AS 4, since the ProConID-list of AS 1 does not include AS 4. Therefore, the route is labeled `Invalid`.

A limitation of this process is that non-adopting ASes between AS X_q and AS X_p, including both endpoints, are not subject to validation. Consequently, a malicious AS could exploit this unvalidated portion of the `AS_PATH` to insert illegitimate AS relationships, meaning that the proposed method cannot fully prevent such attacks.

Apex of `AS_PATH` and `UnknownApex`

If a legitimate route has only one AS at its apex (i.e., the apex is not a peer connection), then AS X_q—the first AS where ProConID validation fails—must correspond to the apex AS. Therefore, AS X_q should be included in the ProConID-list of AS X_p. Recall that even if X_q is not included, it does not necessarily mean that the route is invalid; the route may be `UnknownApex` even if the route does not have any invalid AS pairs based on ProConID-list except for the apex.

If the apex is a peer connection consisting of two ASes, neither AS X_p nor X_q appears in the other's ProConID-list, making the original ProConID-list insufficient for validation in this case. Hence, the original ProConID-list is insufficient for validation in this case.

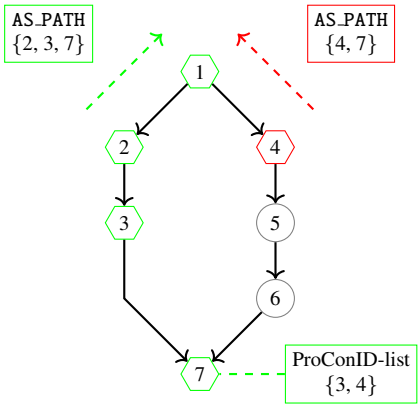
Figure 6 shows an example of such a case. This type of path manipulation is also discussed in an Internet Draft [18]. Assume all ASes publish RPKI objects correctly, and AS 1 propagates a malicious route with `AS_PATH` $\{1, 4, 5\}$ to AS 7. AS 7 validates this route, and the apex is estimated to lie between AS 4 and AS 1, but no fatal errors are found. Thus, the route is labeled `UnknownApex`.

The category of `UnknownApex` has a limitation similar to the `Unknown` state in ASPA verification. If all `UnknownApex` routes are strictly rejected, some legitimate paths may also be discarded, leading to unnecessary connectivity loss. On the other hand, treating them the same as `Valid` routes may also be inappropriate, since attackers could deliberately exploit this state to bypass validation. Therefore, operators must carefully balance security and reachability, for example, by accepting `UnknownApex` routes with a lower `LOCAL_PREF` than `Valid` routes, rather than unconditionally accepting or rejecting them.

4.2 Proposal 2: Validation of Peer Relations at the Apex of the Route

As shown in Figure 6, if the apex of a route consists of two ASes, it cannot be validated with iSec-provider because the RPKI object lacks sufficient information. Our second proposal is therefore to extend RPKI objects with peer information. We refer to this extension as *iSec-apex*.

In iSec-apex, ASes—especially transit or higher-tier ASes that are likely to form the apex of some routes—are expected to publish the list of their peer ASes. The validation method is essentially the same as that of iSec-provider but with additional peer-validation



(a) Experiment

```
== AS information ==========================================================================
AS Number   : 1
Prefix      : 192.168.1.0/24
Connections : Customers : {2, 4}
Routes      :  Destination         | LocPrf | Isec     | AS_PATH (adj > ... > origin)
               ---------------------+--------+----------+-----------------------------
               * 192.168.2.0/24     | 200    | Valid    | 2
               ---------------------+--------+----------+-----------------------------
               * 192.168.3.0/24     | 200    | Valid    | 2 > 3
               ---------------------+--------+----------+-----------------------------
               * 192.168.4.0/24     | 200    | Valid    | 4
               ---------------------+--------+----------+-----------------------------
               * 192.168.5.0/24     | 200    | ----     | 4 > 5
               ---------------------+--------+----------+-----------------------------
               * 192.168.6.0/24     | 200    | ----     | 4 > 5 > 6
               ---------------------+--------+----------+-----------------------------
                 192.168.7.0/24     | 200    | Valid    | 2 > 3 > 7
                 192.168.7.0/24     | 200    | Valid    | 4 > 5 > 6 > 7
               * 192.168.7.0/24     | 200    | Valid    | 4 > 7
============================================================================================
```

(b) Simulation result

Fig. 4: Undetectable path manipulation

```
== AS information ==========================================================================
AS Number   : 1
Prefix      : 192.168.1.0/24
Connections : Providers : {2}
Routes      :  Destination         | LocPrf | Isec     | AS_PATH (adj > ... > origin)
               ---------------------+--------+----------+-----------------------------
               * 192.168.2.0/24     | 90     | ----     | 2
               ---------------------+--------+----------+-----------------------------
               * 192.168.3.0/24     | 90     | ----     | 2 > 3
               ---------------------+--------+----------+-----------------------------
                 192.168.5.0/24     | 90     | Invalid  | 2 > 3 > 4 > 5
============================================================================================
```

Fig. 5: Preventable inappropriate routes with iSec-provider

for the apex of a path.

However, iSec-apex can validate an apex consisting of a peer connection. If ProConID-list validation fails at AS X_q, we check whether the next AS, X_p, is in the peer list of AS X_q. If X_p and X_q form a peer connection, they are considered the apex ASes even if subsequent adopting ASes do not satisfy ProConID-list validation. If they are not peers, the route is not necessarily invalid. As in the previous proposal, the AS may instead assign the path a lower priority using the `LOCAL_PREF` attribute.

Figure 7 shows the result of validation with iSec-apex. The settings are the same as those in Figure 6. With iSec-apex, the malicious route is labeled `UnknownApex`, as in iSec-provider. In contrast, the legitimate route is labeled `Valid`, and AS 7 prefers it.

On the Publication of Peer Relations

This proposal requires the publication of peer relations. The publication of all peer relations is not always desirable from an operational or business perspective. Although publishing already well-known peerings may be feasible, in general, the benefit of disclosing additional peer information is limited, which constrains the practical effectiveness of this proposal.

4.3 Proposal 3: Validation of the Distance to the Next Adopting AS

This proposal aims to prevent path-shortening attacks as shown in Figure 4. We propose extending RPKI objects with the number of non-adopting ASes between adopting ASes. We refer to this proposal as *iSec-dist*.

In iSec-dist, each AS publishes its ProConID-list, but each element of the list is associated with a distance value, representing the number of ASes between the publishing AS and the adopting AS.

For example, in Figure 1, AS 1 publishes its ProConID-list as $\{(2,0),(6,1),(7,1)\}$, where $(2,0)$ indicates that AS 2 is in the ProConID-list of AS 1 and that no AS exists between AS 1 and 2. Similarly, $(6,1)$ indicates that one AS exists between AS 1 and AS 6.

Figure 8 shows an example of how iSec-dist works. Compared to Figure 4, the route with `AS_PATH` $\{4,7\}$ is labeled `Invalid`. In this case, AS 7 publishes its ProConID-list with distance values $\{(3,0),(4,2)\}$. When AS 4 announces the route $\{4,7\}$ to AS 1, the route has no ASes between AS 4 and 7, which contradicts the RPKI object published by AS 7. Therefore, the route is labeled `Invalid`.

However, this mechanism has limitations when an AS has multiple, different-length paths to an adopting AS within its provider cone.

4.4 Unpreventable Attacks

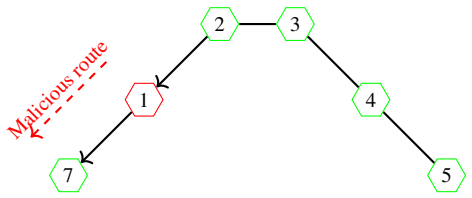
There are still attacks or inappropriate routes that cannot be prevented by our three proposals. As shown in Figure 3, a simple attack cannot be prevented when only a small number of ASes adopt security mechanisms (e.g., only AS 5 is subject to ProConID-list validation).

For iSec-provider and iSec-apex, the two ASes at the apex must adopt the security mechanism for the route to be labeled `Valid`. otherwise, the route may be labeled `UnknownApex`.

5. Conclusion

We revisited the design of BGP-iSec and identified several problems in its validation process. To address these vulnerabilities, we proposed three extensions: iSec-provider, iSec-apex, and iSec-dist. These mechanisms extend validation to cover routes learned from provider ASes and enable the prevention of additional classes of route leaks and hijacks that cannot be addressed by the original BGP-iSec.

Nevertheless, vulnerabilities remain. Our proposals offer limited additional security for routes that traverse only a few adopting ASes. In addition, the requirement to publish peer relations raises concerns about business confidentiality and operational risks, which may limit the practical effectiveness of iSec-apex.



(a) Experiment

```
== AS information ==========================================================================
AS Number   : 7
Prefix      : 192.168.7.0/24
Connections : Providers : {1}
Routes      :  Destination         | LocPrf | Isec     | AS_PATH (adj > ... > origin)
               ---------------------+--------+----------+-------------------------------
               * 192.168.1.0/24     | 90     | Valid    | 1
               ---------------------+--------+----------+-------------------------------
               * 192.168.2.0/24     | 90     | Valid    | 1 > 2
               ---------------------+--------+----------+-------------------------------
               * 192.168.3.0/24     | 90     | ? Apex   | 1 > 2 > 3
               ---------------------+--------+----------+-------------------------------
               * 192.168.4.0/24     | 90     | ? Apex   | 1 > 2 > 3 > 4
               ---------------------+--------+----------+-------------------------------
                 192.168.5.0/24     | 90     | ? Apex   | 1 > 2 > 3 > 4 > 5
               * 192.168.5.0/24     | 90     | ? Apex   | 1 > 4 > 5
============================================================================================
```

(b) Simulation result

Fig. 6: Example of the `UnknownApex` result

```
== AS information ==========================================================================
AS Number   : 7
Prefix      : 192.168.7.0/24
Connections : Providers : {1}
Routes      :  Destination         | LocPrf | Isec     | AS_PATH (adj > ... > origin)
               ---------------------+--------+----------+-------------------------------
               * 192.168.1.0/24     | 90     | Valid    | 1
               ---------------------+--------+----------+-------------------------------
               * 192.168.2.0/24     | 90     | Valid    | 1 > 2
               ---------------------+--------+----------+-------------------------------
               * 192.168.3.0/24     | 90     | Valid    | 1 > 2 > 3
               ---------------------+--------+----------+-------------------------------
               * 192.168.4.0/24     | 90     | Valid    | 1 > 2 > 3 > 4
               ---------------------+--------+----------+-------------------------------
               * 192.168.5.0/24     | 90     | Valid    | 1 > 2 > 3 > 4 > 5
                 192.168.5.0/24     | 90     | ? Apex   | 1 > 4 > 5
============================================================================================
```

Fig. 7: Preventable route hijack with iSec-apex

```
== AS information ==========================================================================
AS Number   : 1
Prefix      : 192.168.1.0/24
Connections : Customers : {2, 4}
Routes      :  Destination         | LocPrf | Isec     | AS_PATH (adj > ... > origin)
               ---------------------+--------+----------+-------------------------------
               * 192.168.2.0/24     | 200    | Valid    | 2
               ---------------------+--------+----------+-------------------------------
               * 192.168.3.0/24     | 200    | Valid    | 2 > 3
               ---------------------+--------+----------+-------------------------------
               * 192.168.4.0/24     | 200    | Valid    | 4
               ---------------------+--------+----------+-------------------------------
               * 192.168.5.0/24     | 200    | ----     | 4 > 5
               ---------------------+--------+----------+-------------------------------
               * 192.168.6.0/24     | 200    | ----     | 4 > 5 > 6
               ---------------------+--------+----------+-------------------------------
               * 192.168.7.0/24     | 200    | Valid    | 2 > 3 > 7
                 192.168.7.0/24     | 200    | Valid    | 4 > 5 > 6 > 7
                 192.168.7.0/24     | 200    | Invalid  | 4 > 7
============================================================================================
```

Fig. 8: Preventable route hijack with iSec-dist

Further work includes developing methods to prevent attacks that remain unpreventable by our current proposals, designing validation mechanisms that are more practical for real-world operations while avoiding additional complexity in RPKI objects, and evaluating the practical usefulness of our proposals in the context of the current Internet.

Acknowledgments The authors would like to thank Professor Kyosuke Yamashita and Professor Takanori Isobe for insightful discussions and continuous guidance throughout this work. This work was supported by JSPS KAKENHI Grant Number JP24H00696 and JST AIP Grant Number JPMJCR24U1.

References

[1] Lepinski, M., Kong, D. and Kent, S.: A Profile for Route Origin Authorizations (ROAs), RFC 6482 (2012).
[2] Bush, R.: Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI), RFC 7115 (2014).
[3] Lepinski, M. and Sriram, K.: BGPsec Protocol Specification, RFC 8205 (2017).
[4] Azimov, A., Bogomazov, E., Bush, R., Patel, K., Snijders, J. and Sriram, K.: BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects, Internet-Draft draft-ietf-sidrops-aspa-verification-22, Internet Engineering Task Force (2025). Work in Progress.
[5] Yamaguchi, Y.: A Study on BGP Routing Failures in the Deployment of the Path Validation Mechanism ASPA, *In the master thesis* (2024).
[6] Yamaguchi, Y., Kimura, T., Yanai, N. and Inomata, A.: A Study on BGP Routing Operations in the Deployment of ASPA as Path Validation, *IA2023-40, IEICE* (2023).
[7] Lychev, R., Goldberg, S. and Schapira, M.: BGP security in partial deployment: is the juice worth the squeeze?, *SIGCOMM Comput. Commun. Rev.*, Vol. 43, No. 4, p. 171–182 (online), DOI: 10.1145/2534169.2486010 (2013).
[8] Morris, C., Herzberg, A. and Secondo, S.: BGP-iSec: Improved Security of Internet Routing Against Post-ROV Attacks, (online), DOI: 10.14722/ndss.2024.241035 (2023).
[9] Hashiba, K. and Okamura, S.: An Analysis of Issues Encountered in the Deployment of Novel Security Technologies Based on RPKI, *IA2025* (2025).
[10] Rekhter, Y., Hares, S. and Li, T.: A Border Gateway Protocol 4 (BGP-4), RFC 4271 (2006).
[11] Gao, L. and Rexford, J.: Stable internet routing without global coordination, *IEEE/ACM Trans. Netw.*, Vol. 9, No. 6, p. 681–692 (online), DOI: 10.1109/90.974523 (2001).
[12] Sriram, K., Montgomery, D., McPherson, D. R., Osterweil, E. and Dickson, B.: Problem Definition and Classification of BGP Route Leaks, RFC 7908 (2016).
[13] Lepinski, M. and Kent, S.: An Infrastructure to Support Secure Internet Routing, RFC 6480 (2012).
[14] Snijders, J., Maddison, B., Lepinski, M., Kong, D. and Kent, S.: A Profile for Route Origin Authorizations (ROAs), RFC 9582 (2024).
[15] Mohapatra, P., Scudder, J., Ward, D., Bush, R. and Austein, R.: BGP Prefix Origin Validation, RFC 6811 (2013).
[16] APNIC Labs: ROA Use World Map, `https://stats.labs.apnic.net/roa` (2025). Accessed: 2025-08-18.
[17] Azimov, A., Uskov, E., Bush, R., Snijders, J., Housley, R. and Maddison, B.: A Profile for Autonomous System Provider Authorization, Internet-Draft draft-ietf-sidrops-aspa-profile-19, Internet Engineering Task Force (2025). Work in Progress.
[18] Azimov, A., Bogomazov, E., Bush, R., Patel, K., Snijders, J. and Sriram, K.: BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects, Internet-Draft draft-ietf-sidrops-aspa-verification-17, Internet Engineering Task Force (2024). Work in Progress.