Autonomous System Relationship Authorization

Presenter: Nan Geng (Huawei)

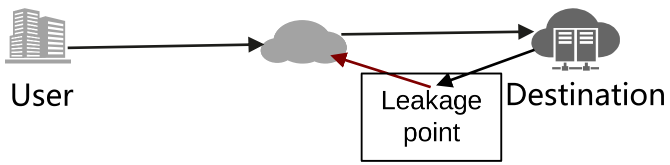
2023.9

Contents

Contents

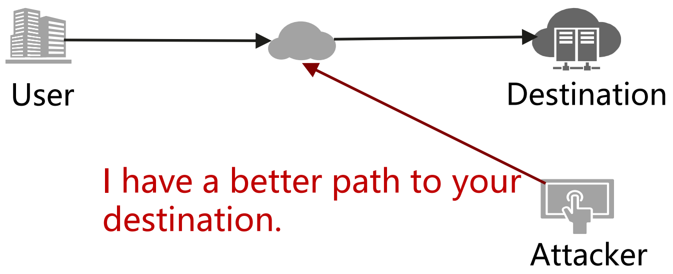
Background

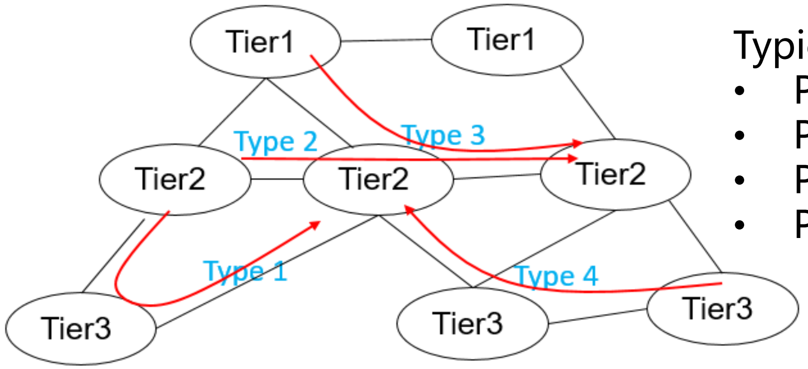
Route Leakage



Oh, I shouldn't advertise these routes, this traffic shouldn't go this path.

Path Hijack

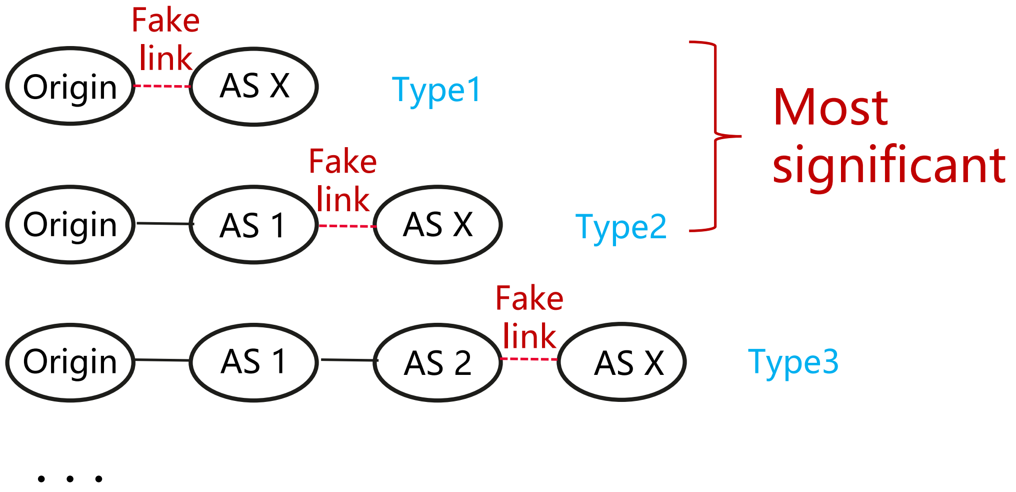




Typical leakage cases

- P2C-C2P
- P2P-P2P
- P2C-P2P
- P2P-C2P

Valley-path

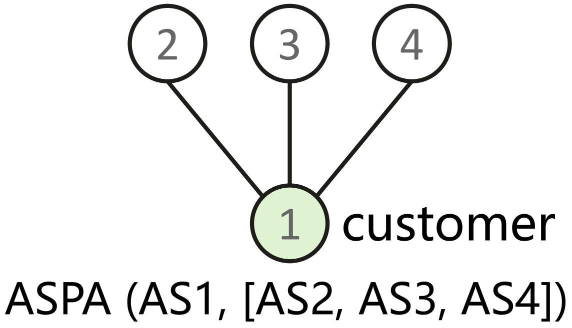


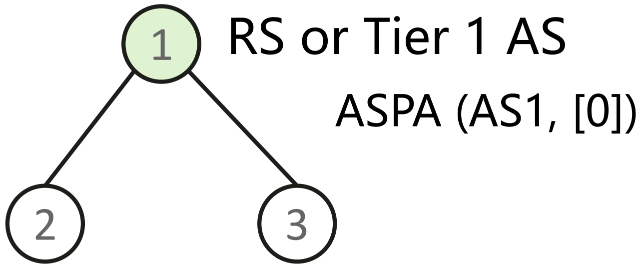
RFC7908 : Problem Definition and Classification of BGP Route Leaks

Contents

ASPA

- **ASPA (Autonomous System Provider Authorization):** Detect route leakage and part of origin-forged attack based on business relationship
- **RPKI ASPA record:** ASPA (ASx, [ASy1, ASy2, ...]) , where Asx is the registration AS and ASy is the provider AS. Store data in RPKI repo.

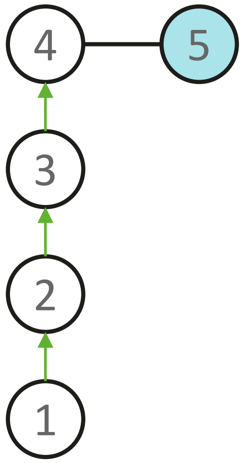




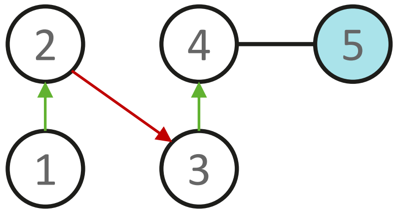
- **Hop-check:** Given an AS pair (ASx, ASy), Hop-check() function checks whether ASy is the provider of ASx based the ASPA data and returns Provider+, Not provider+, or No attestation.

ASPA

- **Upstream check:** verify the route received from customer, lateral peer, RS, or RS-client. Simply,
  - > Valid: Each pair should be Provider+
  - > Invalid: Not provider+ exists
  - > Unknown: Others



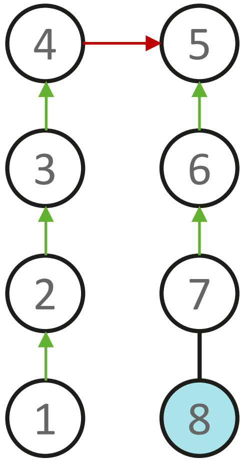
Example 1: AS5 verifies path [AS1, AS2, AS3, AS4] and gets valid/unknown result



Example 2: AS5 verifies path [AS1, AS2, AS3, AS4] and gets invalid result

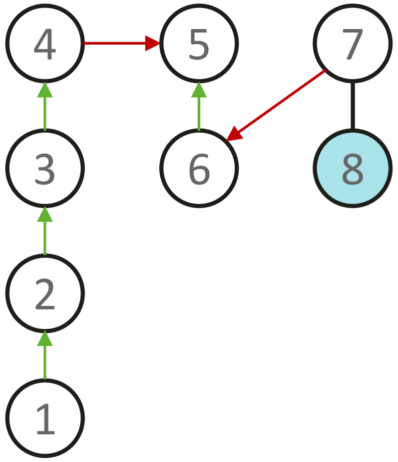
ASPA

- **Downstream check:** verify the route received from transit or mutual transit provider. Simply,
  - > Valid: The first valid up ramp and the last valid down ramp composite the complete path
  - > Invalid: More than two invalid hops between the first valid up ramp and the last valid down ramp
  - > Unknown: Others



Example 3: AS8 verifies the path and gets valid/unknown result

The first valid up ramp: [1, 2, 3, 4]
The last valid down ramp: [5, 6, 7]



Example 4: AS8 verifies the path and gets invalid result

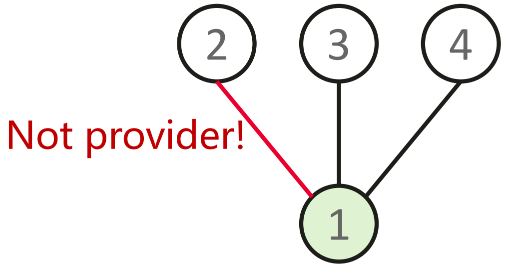
The first valid up ramp: [1, 2, 3, 4]
The last valid down ramp: [7]

ASPA Analysis

- ASPA can **effectively detect invalid paths resulted by route leakage and part of origin-forged attacks**, if the ASPA data are correctly and adequately registered
- ASPA is a promising mechanism

- There are also some **remaining challenges of ASPA (cover more scenarios?)**:
  - How to guarantee **the correctness of ASPA data registration**
  - How to improve the ability of **detecting path hijacks (fake links)**
  - How to improve the validation benefits **under partial registration**
  - How to cope with complex scenarios such as **legitimate valley-paths, hybrid/partial transit relationships**

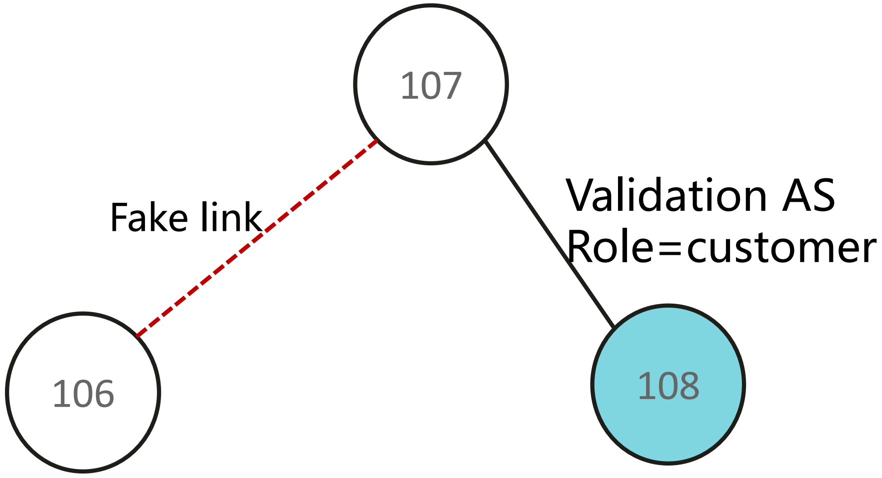
Case 1: Cannot cross-check the correctness of registration



- AS1 registers: AS2, AS3, and AS4 are my providers
- But, AS2 is not its provider
- The incorrectness cannot be detected by the ASPA data registered by AS2

Case 2: Cannot detect hijack introduced by provider

- Route received by AS108: {Origin ASN=106, AS path = [107, 106]}



ASPA records:

	AS106	**AS107**
ASPA	provider-set = {105}	provider-set = {0}

- **ASPA result: valid**
- Analysis: In the current ASPA design, any downstream paths with length no longer than 2 will be considered valid

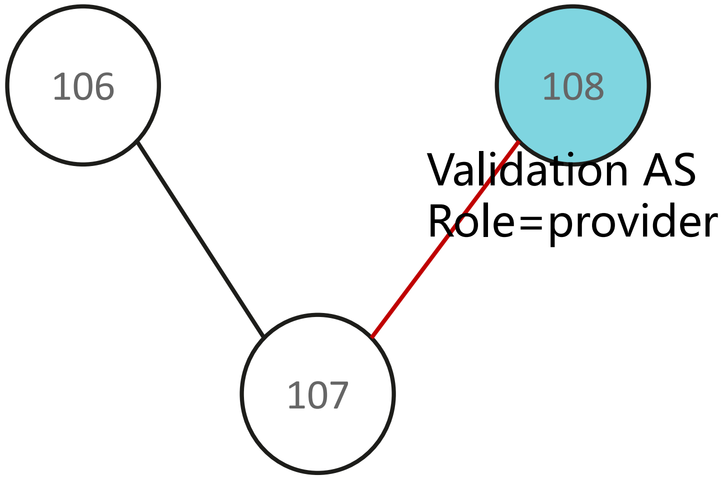
```
root@ip-s8:~/gengnan/bgp/openbgpd-portable # bgpctl s rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced,
       S = Stale, E = Error
origin validation state: N = not-found, V = valid, ! = invalid
aspa validation state: ? = unknown, V = valid, ! = invalid
origin: i = IGP, e = EGP, ? = Incomplete

flags  vs destination          gateway          lpref   med aspath origin
*>    V-V 192.106.0.0/24       10.107.10.154     100      0 107 106 i
*>    V-V 192.107.0.0/24       10.107.10.155     100      0 107 i
AI*>  V-? 192.108.0.0/24       0.0.0.0           100      0 i
```

- **Expected result: invalid-hijack**

* Experiment are based on OpenBgpd

Case 3: Cannot detect leakage with partial registration

- Route received by AS108: {Origin ASN=106, AS path = [107, 106]}



ASPA records:

	AS106	AS107
ASPA	No data	provider-set = {108}

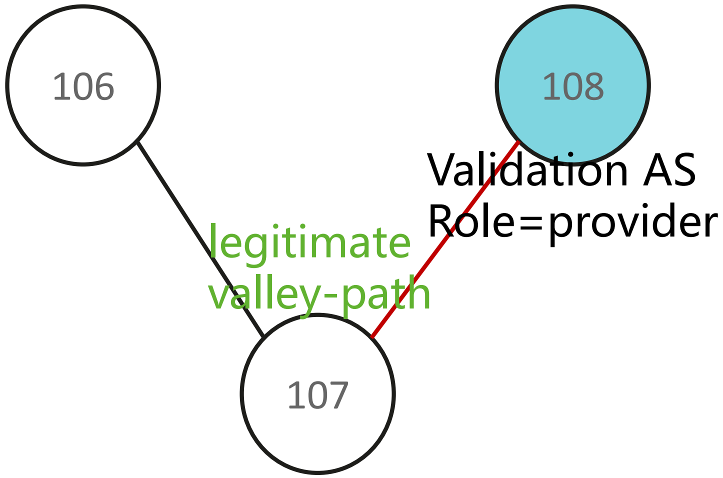
- **ASPA result: unknown**
- Analysis: When do upstream check, hop-check(106, 107) returns No Attestation because AS106 has no record

```
root@ip-s8:~/gengnan/bgp/openbgpd-portable # bgpctl s rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced,
       S = Stale, E = Error
origin validation state: N = not-found, V = valid, ! = invalid
aspa validation state: ? = unknown, V = valid, ! = invalid
origin: i = IGP, e = EGP, ? = Incomplete

flags  vs destination          gateway          lpref   med aspath origin
*>     V-? 192.106.0.0/24      10.107.10.154     100      0 107 106 i
*>     V-V 192.107.0.0/24      10.107.10.155     100      0 107 i
AI*>   V-? 192.108.0.0/24      0.0.0.0           100      0 i
```

- **Expected result: invalid**

Case 4: Cannot cope with complex scenarios

- Route received by AS108: {Origin ASN=106， AS path = [107, 106]}



ASPA records:

	AS106	**AS107**
ASPA	provider-set = {105}	——

- **ASPA result: invalid**
- Analysis: In the current ASPA design, legitimate valley-paths, hybrid/partial transit relationships cannot be coped with

```
root@ip-s8:~/gengnan/bgp/openbgpd-portable # bgpctl s rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced,
       S = Stale, E = Error
origin validation state: N = not-found, V = valid, ! = invalid
aspa validation state: ? = unknown, V = valid, ! = invalid
origin: i = IGP, e = EGP, ? = Incomplete

flags  vs destination          gateway          lpref   med aspath origin
*>     V-! 192.106.0.0/24      10.107.10.154     100      0 107 106 i
*>     V-V 192.107.0.0/24      10.107.10.155     100      0 107 i
AI*>   V-? 192.108.0.0/24      0.0.0.0           100      0 i
```

- **Expected result: valid**

Contents

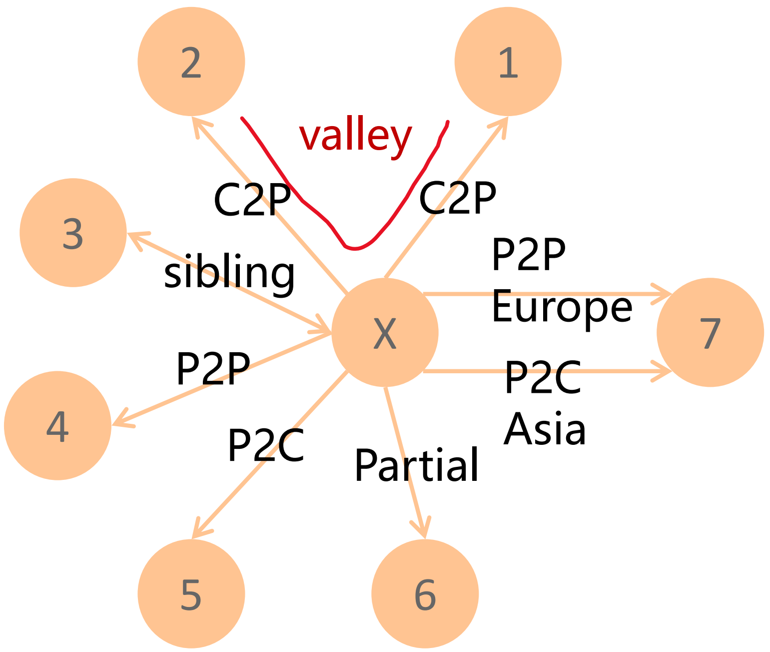
Autonomous System Relationship Authorization (ASRA)

- Preliminary idea: Register more relationship information

Level	Feature	Content	Description
1	Mandatory	Providers	Same as ASPA
2	Optional	Neighbors	All AS adjacencies, similar to [1]
3	Optional	All normal and complex relationships	Customer, lateral peer, hybrid, partial, valley-path, etc.

- [1] draft-huston-sidr-aao-profile-03. A Profile for AS Adjacency Attestation Objects. Geoff Huston , George G. Michaelson

Illustration of ASRA Record



An example with many kinds of relationships

Data type	Requirement	Record data
provider set	Mandatory	[1, 2, 3]
other neighbor set	Optional	[4, 5, 6, 7]
customer set	Optional	[3, 5]
lateral peer set		[4]
partial transit set		[6]
hybrid set		[(7,tag1:P, tag2:C)], tag means geographic location
valley-path set		[[1, 2]], means 1, X, and 2 can form a valley-path

What Can be Done using ASRA Data

- Usage 1: Cross-check the correctness of registration
  > X registers Y as the provider, but Y registers X as the peer. A conflict occurs.
  > X states Y is the provider, but X does not appear in the neighbor set of Y. A conflict occurs.

- Usage 2: Identify fake links
  > An AS can register all neighboring ASes. Fake links in the AS path can be detected and considered as hijacking.

- Usage 3: Hop-check(Y, X) for Hop-check(X, Y) under partial registration
  > In Hop-check(X, Y), if X does not register data, you can use the ASRA data registered by Y to verify the X-to-Y business relationship.

- Usage 4: Cope with complex scenarios
  > Legitimate valley-path, hybrid/partial relationship

ASRA Implementation based on OpenBgpd

- Modify the open source project OpenBgpd for implementing the ASRA prototype
- The config file is extended. Three new kinds of records are supported:
  - > Neighbor set
  - > Customer set
  - > Peer set

```
aspa-set {
    customer-as 107 provider-as {106, 108} customer-set {0} peer-set {0}
    customer-as 106 provider-as {105} neighbor-set {111, 222}
}
```

- The verification is modified, and the ASRA records can be fully used.
- The complex relationships and legitimate valley-path are not supported. RTR protocol is not extended.

- Experiment: 3 servers, Ubuntu 20.04.5 LTS， named as AS106, AS107, and AS108, respectively

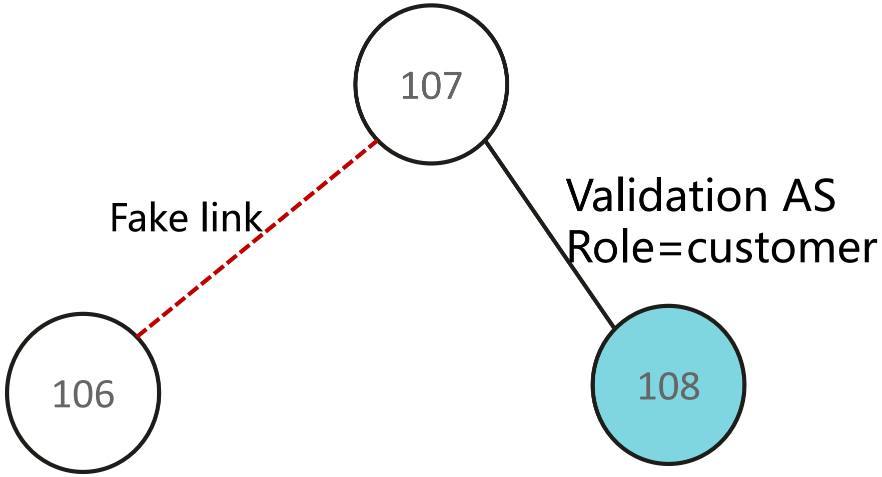
AS 106
Subnet: 192.106.0.0/24

AS 107
Subnet: 192.107.0.0/24

AS 108
Subnet: 192.108.0.0/24

Case 1: Can detect hijack introduced by provider

- Route received by AS108: {Origin ASN=106，AS path = [107, 106]}



ASPA & ASRA records:

	AS106	AS107
ASPA	provider-set = {105}	——
ASRA	provider-set = {105} neighbor-set = {111, 222}	——
ASRA	provider-set = {105} customer-set = {111} peer-set = {222}	——

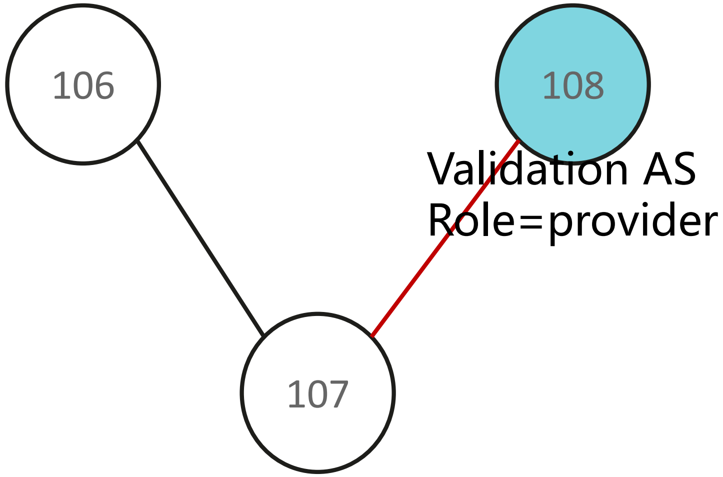
- **ASPA result: valid**
- Analysis: In the current ASPA design, any downstream paths with length no longer than 2 will be considered valid

- **NEW result：invalid-hijack**
- Analysis: Since all neighbor links are known, the fake link can be easily detected

```
root@ip-s8:~/gengnan/bgp/openbgpd-portable # bgpctl show rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced,
       S = Stale, E = Error
origin validation state: N = not-found, V = valid, ! = invalid
aspa validation state: ? = unknown, V = valid, ! = invalid, !! = invalid-hijack
origin: i = IGP, e = EGP, ? = Incomplete

flags  vs destination          gateway          lpref   med aspath origin
*>     V-!! 192.106.0.0/24      10.107.10.154      100      0 107 106 i
*>     V-V 192.107.0.0/24      10.107.10.155      100      0 107 i
AI*>   V-? 192.108.0.0/24      0.0.0.0            100      0 i
```

Case 2: Can detect leakage with partial registration

- Route received by AS108: {Origin ASN=106， AS path = [107, 106]}



ASPA & ASRA records:

	AS106	AS107
ASPA	无	provider-set = {108}
ASRA	无	provider-set = {108} customer-set = {0} peer-set={0}

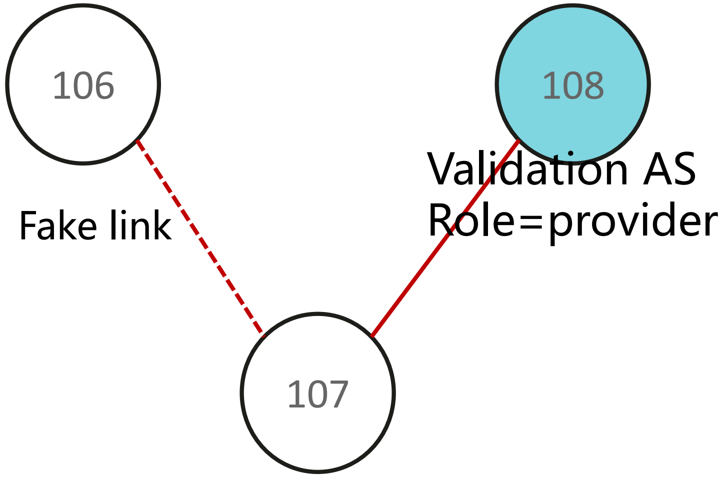
- **ASPA result: unknown**
- Analysis: When do upstream check, hop-check(106, 107) returns No Attestation because AS106 has no record

- **NEW result：invalid**
- Analysis: Backward verification is used. When AS106 does not have registration data, AS107 can be used to register customer and peer data for verification. Only provider data (sibling cannot be identified) or provider and neighbor data cannot be used for backward verification.

```
root@ip-s8:~/gengnan/bgp/openbgpd-portable # bgpctl show rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced,
       S = Stale, E = Error
origin validation state: N = not-found, V = valid, ! = invalid
aspa validation state: ? = unknown, V = valid, ! = invalid, !! = invalid-hijack
origin: i = IGP, e = EGP, ? = Incomplete

flags  vs destination          gateway          lpref   med aspath origin
*>     V-!! 192.106.0.0/24      10.107.10.154     100     0 107 106 i
*>     V-V 192.107.0.0/24      10.107.10.155     100     0 107 i
AI*>   V-? 192.108.0.0/24      0.0.0.0           100     0 i
```

Case 3: Can detect hijack with partial registration

- Route received by AS108: {Origin ASN=106，AS path = [107, 106]}



ASPA & ASRA records:

	AS106	AS107
ASPA	无	provider-set = {108}
ASRA	无	provider-set = {108} neighbor-set = {0}
ASRA	无	provider-set = {108} customer-set = {0} peer-set={0}

- **ASPA result: unknown**
- Analysis: When do upstream check, hop-check(106, 107) returns No Attestation because AS106 has no record

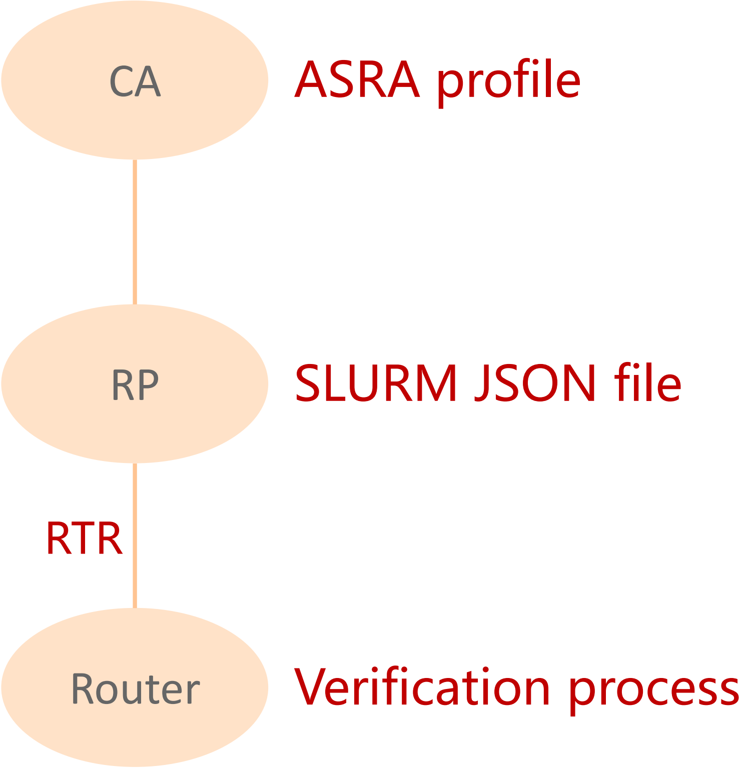
- **NEW result：invalid-hijack**
- Analysis: Backward verification is used. When AS 106 has no registration data, the data registered by AS 107 can be used for verification to identify hijacking

```
root@ip-s8:~/gengnan/bgp/openbgpd-portable # bgpctl show rib
flags: * = Valid, > = Selected, I = via IBGP, A = Announced,
       S = Stale, E = Error
origin validation state: N = not-found, V = valid, ! = invalid
aspa validation state: ? = unknown, V = valid, ! = invalid, !! = invalid-hijack
origin: i = IGP, e = EGP, ? = Incomplete

flags  vs destination          gateway          lpref   med aspath origin
*>     V-!! 192.106.0.0/24      10.107.10.154     100      0 107 106 i
*>     V-V 192.107.0.0/24      10.107.10.155     100      0 107 i
AI*>   V-? 192.108.0.0/24      0.0.0.0           100      0 i
```

Possible Extensions for ASRA

- Possible extensions for ASRA deployment
  - > ASRA Profile
  - > SLURM for local data provision
  - > RTR protocol for ASRA record synchronization
  - > ASRA verification (internal implementation)

Registration is challenging



Deployment Consideration

- Globally public data registration is challenging. Some ASes may be concerned about **privacy**.
  - There are some analysis on why privacy problem is not much important in many cases [1]
  - **Selective registration**. ASes can choose to register neighbor set, detailed relationships, etc. Neighbor set induces relatively less privacy concerns.
  - **Regional registration**. ASRA data can be registered and used with a region, which can efficiently prevent Type1 and Type2 origin-forged attacks.

- ASRA can **improve deployment benefits under partial registration**.
  - Generally, ASPA has less deployment benefits than ROA under the same registration ratio

- [1] Cohen, Avichai, et al. "Jumpstarting BGP security with path-end validation." Proceedings of the 2016 ACM SIGCOMM Conference. 2016.

Conclusion

- ASRA: AS can optionally register more detailed AS relationships, and ASPA can be enhanced.
  - > Usage 1: Cross-check the correctness of registration
  - > Usage 2: Identify fake links
  - > Usage 3: Improve deployment benefits under partial registration
  - > Usage 4: Cope with complex scenarios

- Limitation: Can detect route leakage and fake link hijacking, but cannot prevent and identify AS path tampering (BGPsec for path protection)

Thank you!