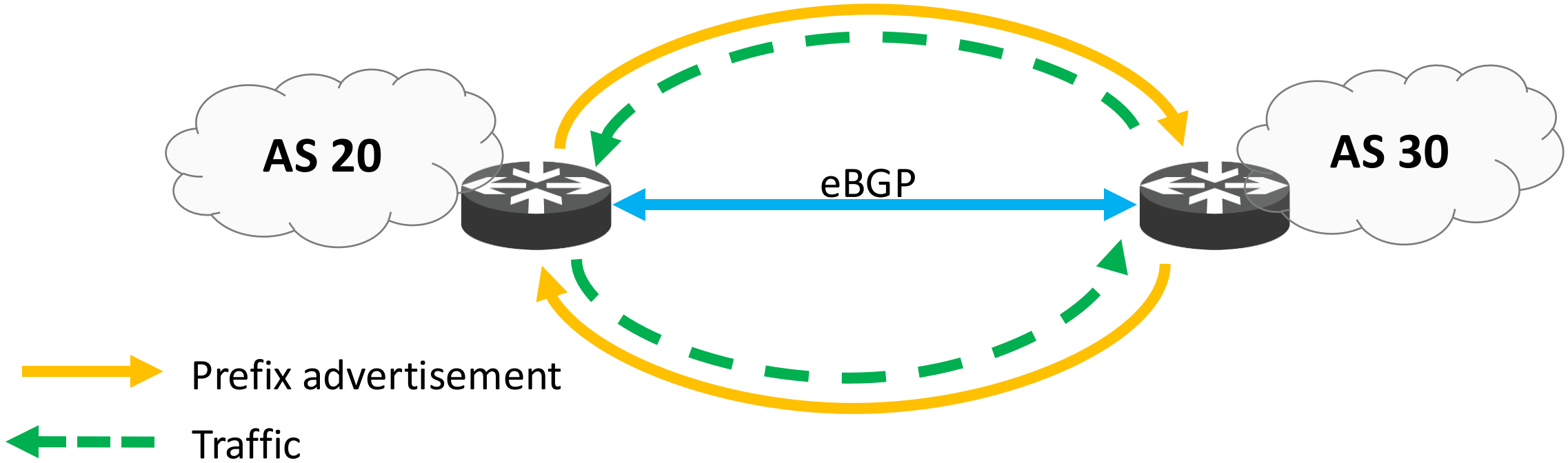
BGP Best Practices

Agenda

- Intro. to BGP ?
- BGP vs IGP
- BGP Security Vulnerabilities Analysis
- BCP to address Vulnerabilities

BGP ?

- **ASN :** An Autonomous System (AS) is a group of IP networks with a unified routing policy, under one admin.
- Exchange routing information between different **AS**



BGP Vs IGP

- **IGP**
- Only Exchange routing **within** AS
- Only carry **infrastructure** prefixes
- Do not carry **customer/ Internet** prefixes

✓ Less no. of prefixes, better **scalability** and faster **convergence**

BGP Vs IGP

Two types of **BGP** sessions

- **iBGP** to exchange routing information within AS
  - **Internet** routes
  - **Customer** routes

- **eBGP** to exchange routing information with other ASes
  - Advertise **your own** prefixes
  - Advertise **Customer** prefixes
  - **Apply policies to enforce control**

BGP built-in problem

- Routing works by RUMOUR
  - It is only based on trust, no built-in security

- Assume everyone is correct (and *honest*)
  - No verification of the correctness of prefixes or AS paths

BGP Security Vulnerabilities Analysis

Lack of Built-in Security Measures

- No protection for message integrity or authenticity
- No validation of AS authority to announce routes
- No verification of path attribute authenticity

BCP to address Vulnerabilities

According to RFC 7454 and supporting best practices:

- Protect BGP Speakers
  - Apply data-plane filters (e.g., uRPF) to block spoofed packets
  - Use control-plane ACLs to block unauthorized access to TCP port 179
- Secure BGP Sessions
  - Use MD5 (RFC 2385) or TCP-AO (RFC 5925) for peer authentication
  - Prefer TCP-AO over MD5 for stronger algorithms (e.g., HMAC-SHA1) and key rotation without session disruption

BCP to address Vulnerabilities

- Route Filtering
  - Implement prefix/AS-path filters (deny bogons routes)
  - Set max-prefix limits to prevent route floods
- IRR Registration
  - Maintain accurate routing data in IRR databases to support automated filtering (aut-num)
- RPKI Validation
  - Register routes in RPKI (ROA) to validate origin authenticity
  - Deploy ROV (Route Origin Validation) to reject invalid routes

Config. Example - Block Spoofing (uRPF)

- **Strict Mode** (for single-homed interfaces)

Cisco

```
interface GigabitEthernet0/0
 ip verify unicast source reachable-via rx
```

Juniper

```
set interfaces ge-0/0/0 unit 0 family inet rpf-check
```

- **Loose Mode** (for multi-homed interfaces):

Cisco

```
interface GigabitEthernet0/0
 ip verify unicast source reachable-via any
```

Juniper

```
set interfaces ge-0/0/0 unit 0 family inet rpf-check mode loose
```

Config. Example – Protect TCP-179

- Cisco

```
ip access-list extended CoPP
 permit tcp host 10.10.10.2 host 192.168.1.1 eq 179
# Allow trusted peer
 deny tcp any any eq 179
# Block others


# Apply ACL to Control-Plane
!
control-plane
 service-policy input CoPP
```

- Juniper

```
firewall {
  family inet {
    filter BGP-CONTROL-PLANE {
      term ALLOW-BGP-PEERS {
        from {
          source-address {
            10.10.10.2/32;   # Trusted peer
          }
          destination-port bgp;
        }
        then accept;
      }
      term BLOCK-OTHERS {
        then discard;
      }
    }
  }
}
```

apply:

```
set interfaces lo0 unit 0 family inet filter input BGP-CONTROL-PLANE
```

Config. Example – TCP-AO

- Cisco

```
key chain BGP-TCPAO tcp
 key 1
  send-id 10
  recv-id 20
  cryptographic-algorithm aes-128-cmac
  key-string encrypted 066A0D020D1C470B1E
  send-lifetime 12:00:00 Mar 18 2025 infinite
 !
!
router bgp 65001
 neighbor 203.0.113.2
  remote-as 65002
  ao BGP-TCPAO include-tcp-options
  address-family ipv4 unicast
 !
!
```

** If TCP-AO is not supported, use MD5 **

- Juniper

```
security {
    authentication-key-chains {
        key-chain BGP-TCPAO {
            key 1 {
                secret "$9$Hsd4QbvfhT6m";  # AES-128-CMAC-96 encrypted
                start-time "2025-03-18.12:00:00 +0000";
                algorithm ao;
                ao-attribute {
                    send-id 20;    # Matches Cisco's recv-id
                    recv-id 10;    # Matches Cisco's send-id
                    tcp-ao-option enabled;
                    cryptographic-algorithm aes-128-cmac-96;
                }

protocols {
    bgp {
        group EBGP {
            neighbor 203.0.113.1 {
                authentication-algorithm ao;
                authentication-key-chain BGP-TCPAO;
            }
        }
```

Config. Example – Filtering

- Drop bogons when receiving and advertising **IPv4** & **IPv6**

Bogon IPv4 Prefixes:

```
0.0.0.0/8 (This network)
10.0.0.0/8 (Private-use networks)
100.64.0.0/10 (Carrier-grade NAT)
127.0.0.0/8 (Loopback)
169.254.0.0/16 (Link-local)
172.16.0.0/12 (Private-use networks)
192.0.0.0/24 (IETF Protocol Assignments)
192.0.2.0/24 (TEST-NET-1)
192.168.0.0/16 (Private-use networks)
198.18.0.0/15 (Benchmarking)
198.51.100.0/24 (TEST-NET-2)
203.0.113.0/24 (TEST-NET-3)
224.0.0.0/4 (Multicast)
240.0.0.0/4 (Reserved for future use)
255.255.255.255/32 (Limited broadcast)
```

Additional considerations:

- Filter prefixes longer than /24 (0.0.0.0/0 ge 25)
- Include your own network ranges
- Regularly update this list as allocations change

Tools & Techniques

- IRR
  - *Helps auto generate network (prefix/as-path) filters using RPSL tools*
    - Filter out route advertisements not described in the registry

```
~ bgpq3 -bl PEERv4-IN AS17660
PEERv4-IN = [
    45.64.248.0/22,
    103.7.252.0/22,
    103.7.254.0/23,
    103.245.240.0/22,
    103.245.242.0/23,
    119.2.96.0/19,
    119.2.96.0/20,
    202.89.24.0/21,
    202.144.128.0/19,
    202.144.128.0/23,
    202.144.144.0/20,
    202.144.148.0/22
];
~ bgpq3 -6bl PEERv4-IN AS17660
PEERv4-IN = [
    2405:d000::/32,
    2405:d000:7000::/36
];
```

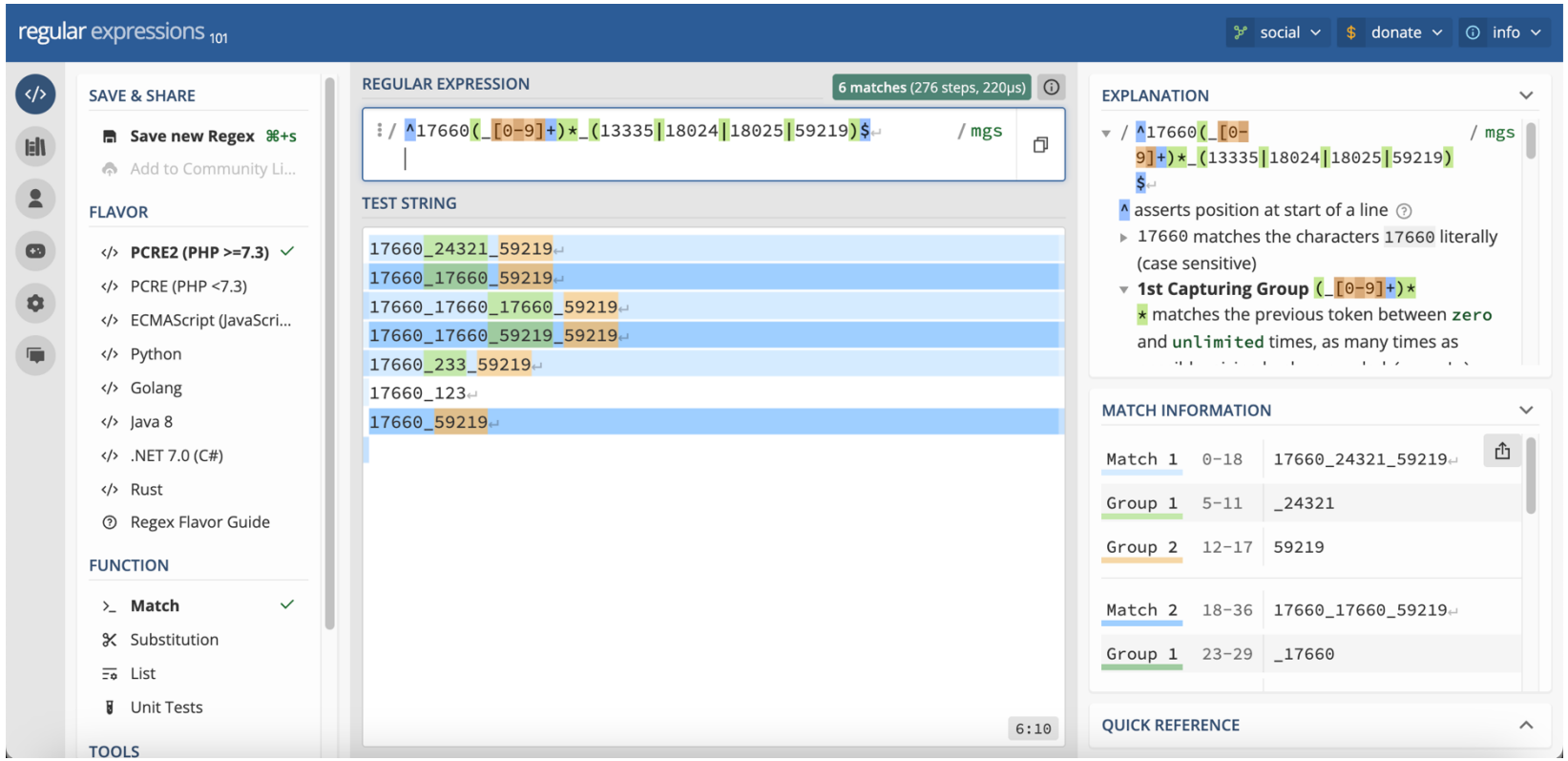
```
~ bgpq3 -S APNIC -bl PEERv4-IN AS17660
PEERv4-IN = [
    45.64.248.0/22,
    103.245.240.0/22,
    103.245.242.0/23,
    119.2.96.0/19
];
~ bgpq3 -S APNIC -Jl PEERv4-IN AS17660
policy-options {
replace:
 prefix-list PEERv4-IN {
    45.64.248.0/22;
    103.245.240.0/22;
    103.245.242.0/23;
    119.2.96.0/19;
 }
}
```

```
~ bgpq3 -3f 17660 -l BT-IN AS-DRUKNET-TRANSIT
no ip as-path access-list BT-IN
ip as-path access-list BT-IN permit ^17660(_17660)*$
ip as-path access-list BT-IN permit ^17660(_[0-9]+)*_(18024|18025|38004|59219)$
ip as-path access-list BT-IN permit ^17660(_[0-9]+)*_(132232|134715|135666|137925)$
ip as-path access-list BT-IN permit ^17660(_[0-9]+)*_(137994)$
```

```
~ bgpq3 -3f 38195 -l SUPERLOOP-IN AS-SUPERLOOP
no ip as-path access-list SUPERLOOP-IN
ip as-path access-list SUPERLOOP-IN permit ^38195(_38195)*$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(681|4647|4749|4785)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(4841|4858|5091|5740)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(6404|6461|7280|7469)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(7477|7490|7578|7585)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(7604|7628|7631|7699)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(8360|8444|9249|9290)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(9313|9438|9463|9479)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(9499|9544|9549|9661)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(9795|9797|10143|10145)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(10310|11031|11054|12041)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(12189|13331|13414|13720)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(14148|15133|15562|15967)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(16164|17158|17457|17462)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17477|17498|17732|17766)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17812|17819|17829|17889)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17906|17907|17983|17985)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(17991|18000|18110|18201)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(18231|18291|18292|18349)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(18385|18407|18549|18701)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(19385|19397|20473|21534)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(21859|22097|22363|23156)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(23197|23352|23667|23677)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(23686|23747|23858|23913)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(23935|24007|24008|24033)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24065|24093|24098|24129)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24231|24233|24238|24242)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24322|24341|24380|24459)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(24570|25605|25665|27232)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(29457|30081|30103|30109)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(30215|30762|31732|32771)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(36351|37993|38068|38172)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38220|38263|38269|38298)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38451|38534|38541|38570)$
ip as-path access-list SUPERLOOP-IN permit ^38195(_[0-9]+)*_(38716|38719|38726|38809)$
```

Tools & Techniques

Regex101 - The Ultimate Regular Expression **Tester**



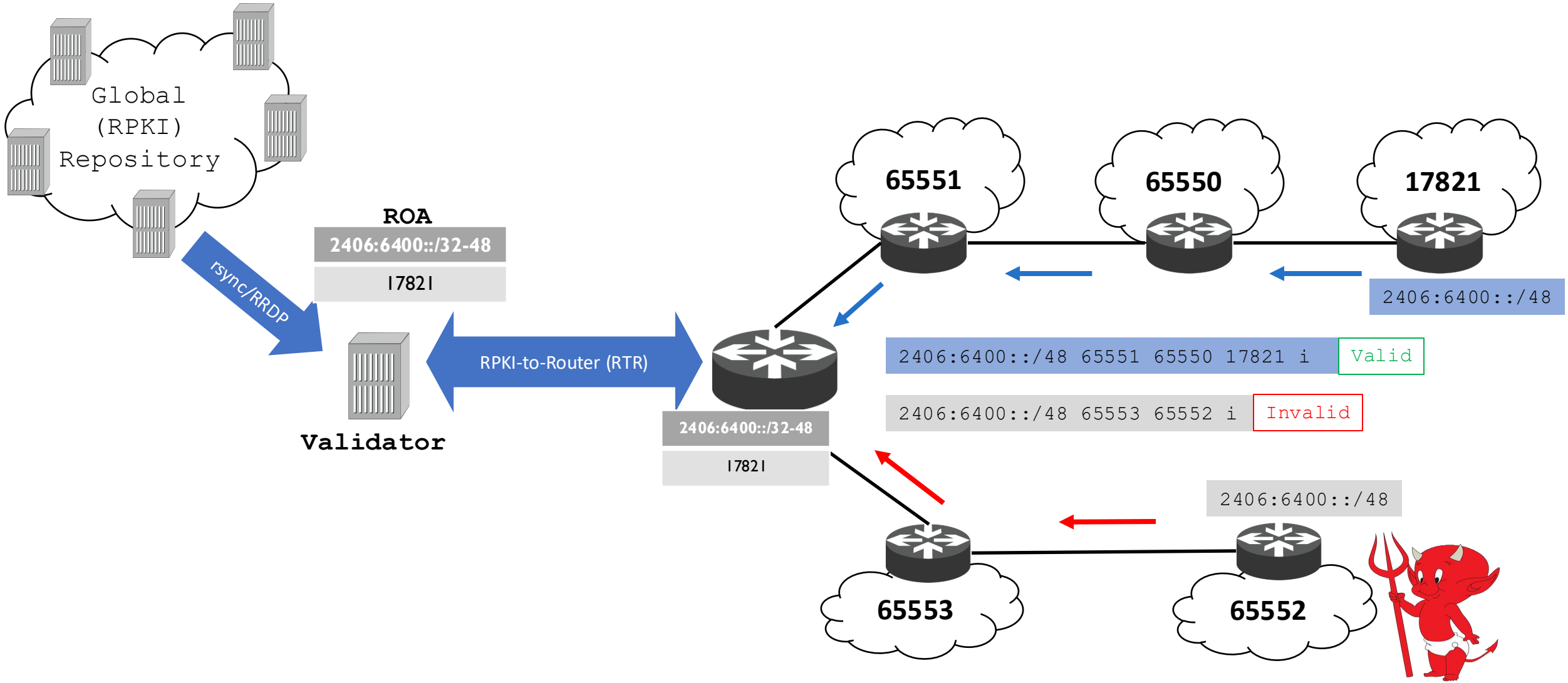
Route Origin Authorization (ROA)

- Digitally signed object
  - Binds list of prefixes and the nominated ASN
  - *can be verified cryptographically*

Prefix	203.176.32.0/19
Max-length	/24
Origin ASN	**AS17821**

- ** *Multiple ROAs can exist for the same prefix*

Route Origin Validation (ROV)




Questions

Thank You!