

ASPA-08+

With special thanks to Ben Maddison and Sriram Kotikalapudi

# The ASPA Profile-07

```
ASProviderAttestation ::= SEQUENCE {  
    version [0] ASPAVersion DEFAULT v0,  
    aFI AddressFamilyIdentifier,  
    customerASID ASID,  
    providerASSET SEQUENCE (SIZE(1..MAX)) OF ASID }
```

# The ASPA Profile-08

```
ASProviderAttestation ::= SEQUENCE {  
    version [0] ASPAVersion DEFAULT v0,  
    customerASID ASID,  
    providers ProviderASSet }
```

```
ProviderASSet ::= SEQUENCE (SIZE(1..MAX)) OF ProviderAS
```

```
ProviderAS ::= SEQUENCE {  
    providerASID ASID,  
    afiLimit AddressFamilyIdentifier OPTIONAL }
```

# Yet Another ASPA Object

```
ASProviderAttestation ::= SEQUENCE {  
    version [0] ASPAVersion DEFAULT v0,  
    aFI AddressFamilyIdentifier OPTIONAL,  
    customerASID ASID,  
    providerASSET SEQUENCE (SIZE(1..MAX)) OF ASID }
```

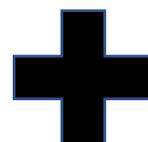
# Considerations

We can't drop AFI, can we?

- ASPA records in different AFIs will be separate at the level of router;
- ASPA records in different AFIs will be separate at the level of RTR;
- **ASPA records in different AFIs will be separate at the level of RPKI?**

# Voting

<b>ASPA-07</b>	<b>ASPA-08</b>
Alexander Azimov	Ben Maddison
Randy Bush	Claudio Jeker
Ties de Kock	Tim Bruijnzeels
Russ Housley	Job Snijders*



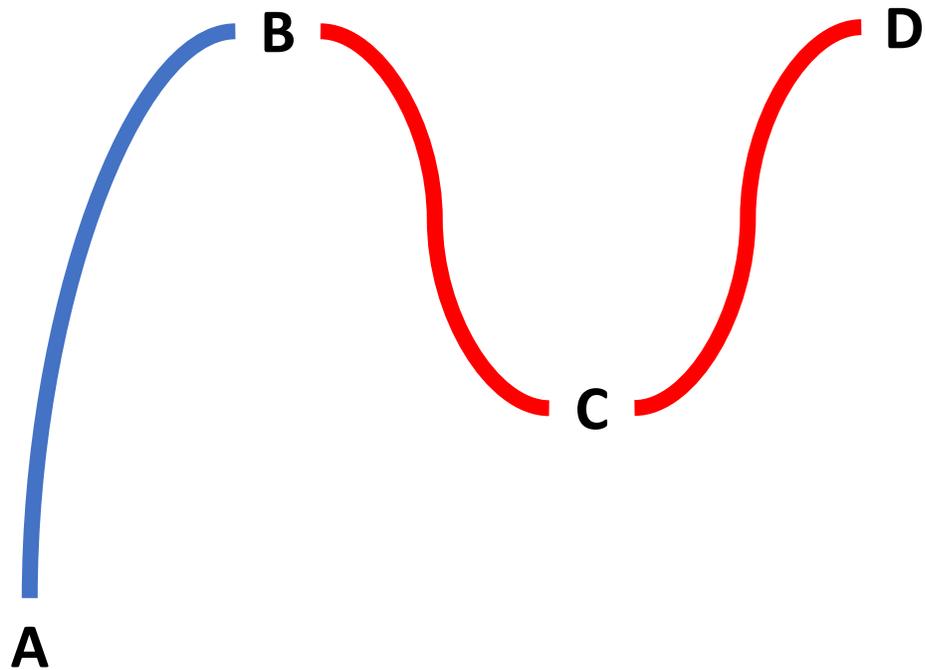
**Intercop**

# Invalid Indexes

**Invalid Index** is as a minimal  $I$  for which  $(AS(I), AS(I+1), AFI)$  returns Invalid. If  $I$  index doesn't exist, we put the length of `AS_PATH` in its value.

**Reverse Invalid Index** is Invalid Index defined for reverse `AS_PATH`.

# Route Leak Detection at Peer, Provider, IX



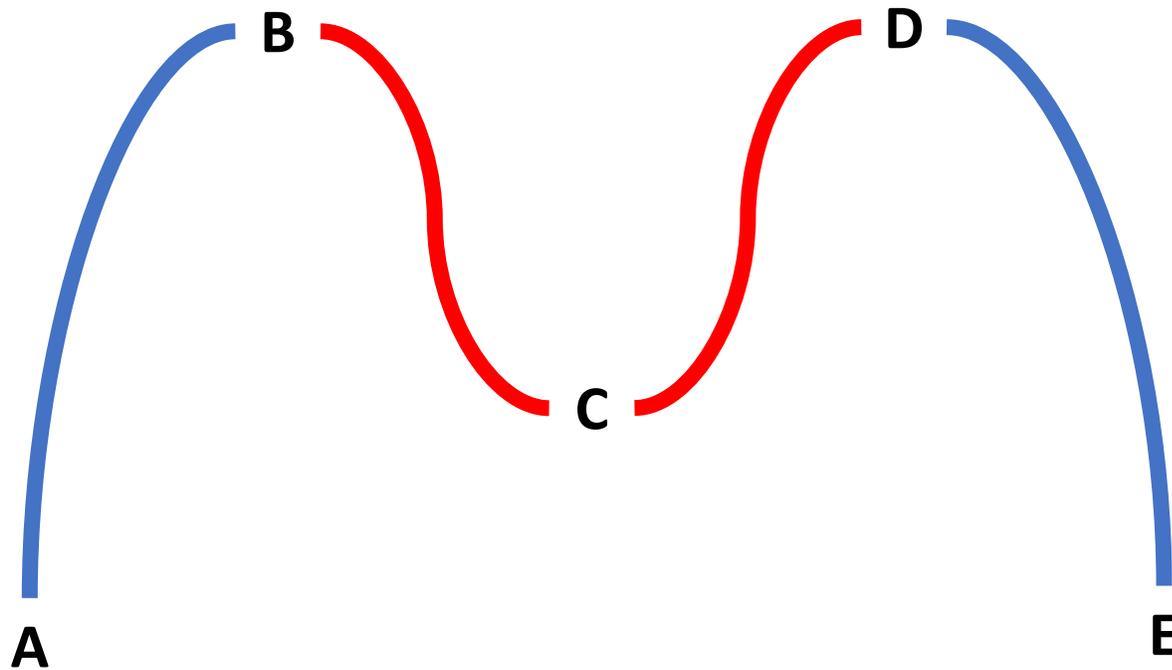
$\text{len}(AB) = \text{Invalid Index}$

Leak detection:  $\text{Invalid Index} < \text{len}(\text{AS\_PATH})$

# Route Leak Detection at RS, RS-Client

- If a non-transparent IX – register the RS AS in ASPA;
- RS uses Provider/Peer procedure;
- RS-client uses Provider/Peer procedure too!

# Route Leak Detection at Customer



len(AB) = Invalid Index  
len(DE) = Reverse Invalid Index

Leak detection: Invalid Index + Reverse Invalid Index < len(AS\_PATH)

## The Unknown Path

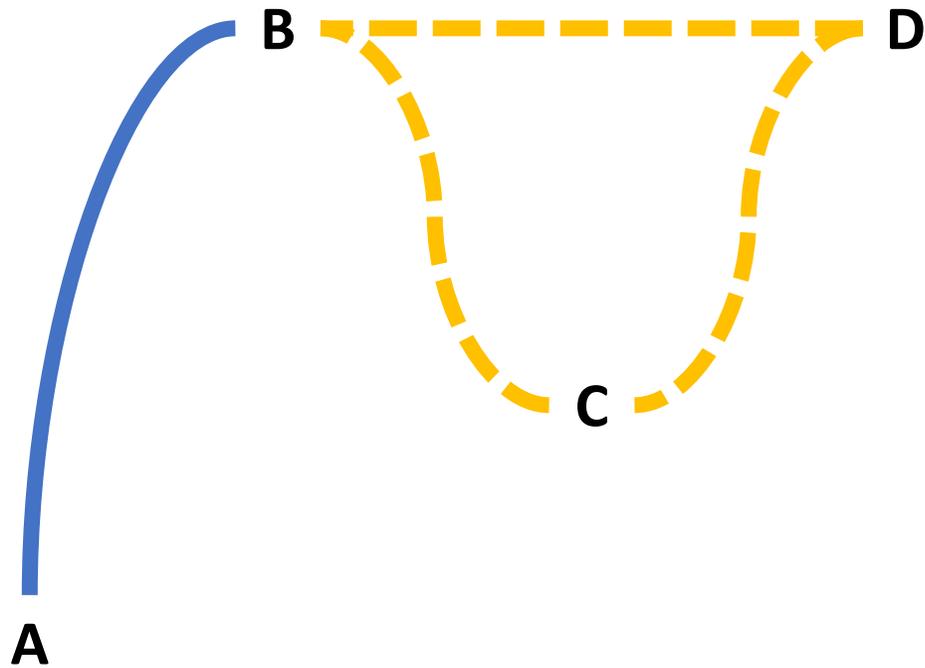
The path that *MAY* have been leaked

# Unknown Indexes

**Unknown Index** is a minimal  $I$  for which  $(AS(I), AS(I+1), AFI)$  returns Unknown. If  $I$  is greater than Invalid Index or  $I$  doesn't exist we equate its value to the value of Invalid Index.

**Reverse Unknown Index** is Invalid Index defined for reverse `AS_PATH`.

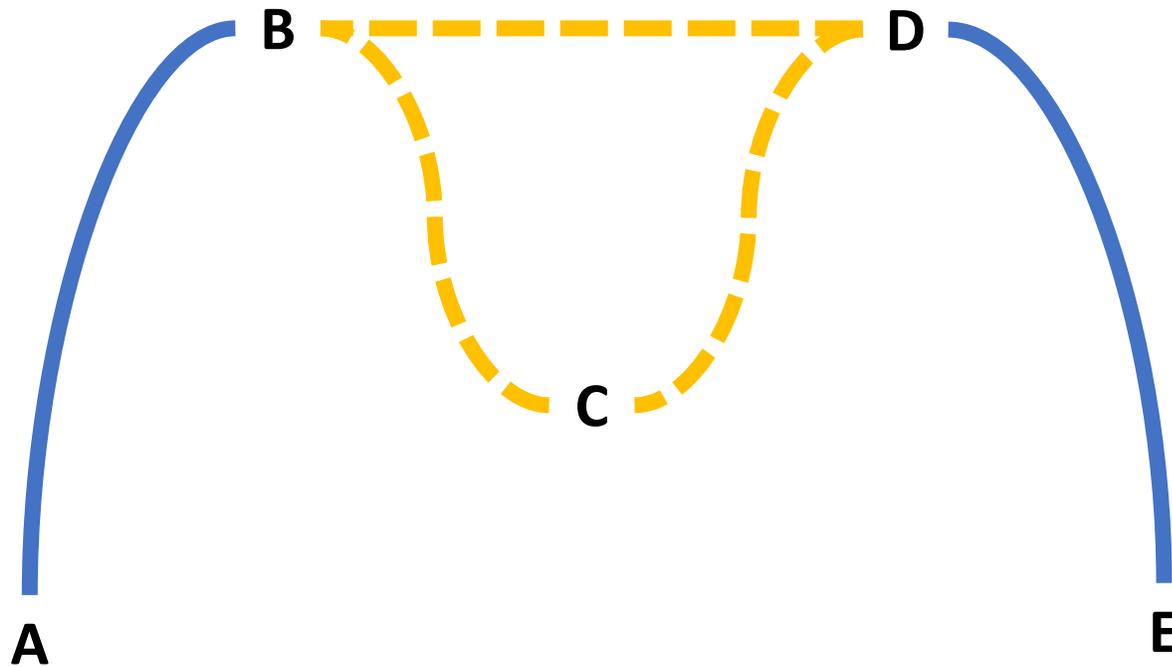
# Unknown Detection at Peer, Provider, IX



$\text{len}(AB) = \text{Unknown Index}$

Unknown detection:  $\text{Unknown Index} < \text{len}(\text{AS\_PATH})$

# Unknown Detection at Customer



$\text{len}(AB) = \text{Unknown Index}$   
 $\text{len}(DE) = \text{Reverse Unknown Index}$

Unknown detection:  $\text{Unknown Index} + \text{Reverse Unknown Index} < \text{len}(\text{AS\_PATH})$

# All Together at Peer, Provider, IX

- Invalid  $\text{Index} < \text{len}(\text{AS\_PATH})$  – Invalid;
- Unknown  $\text{Index} < \text{len}(\text{AS\_PATH})$  – Unknown;
- Otherwise, Valid.

# All Together at Customer

- Invalid Index + Reverse Invalid Index < len(AS\_PATH) – Invalid;
- Unknown Index + Reverse Unknown Index < len(AS\_PATH) – Unknown;
- Otherwise, Valid.

# Considerations

- What is the fate of deprecate-as-set-confed-set?
- Should routes with AS\_SET in the middle be marked as Invalid?
- Should routes with AS\_SET in the beginning be marked as Invalid?
- Volunteers to read?
- Volunteers to code?

<https://github.com/QratorLabs/ASPA/>