

# Draft-ietf-sidr-bgpsec-protocol

Matt Lepinski

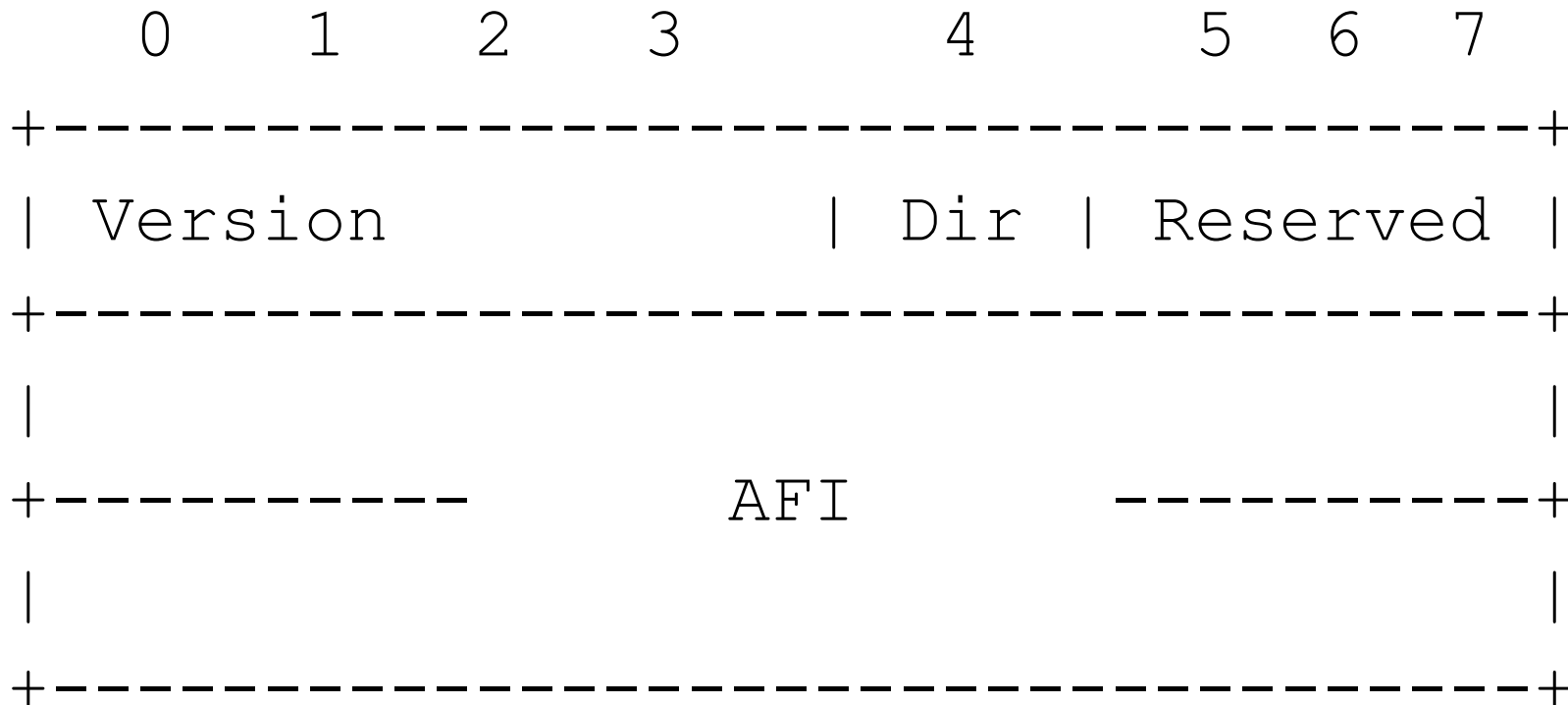
# This Document Contains

- An optional, non-transitive path attribute
  - BGPSEC\_Path attribute
- A capability [RFC5492] for negotiating support for this attribute
- Processing instructions for creating, modifying (adding signatures) and validating this attribute

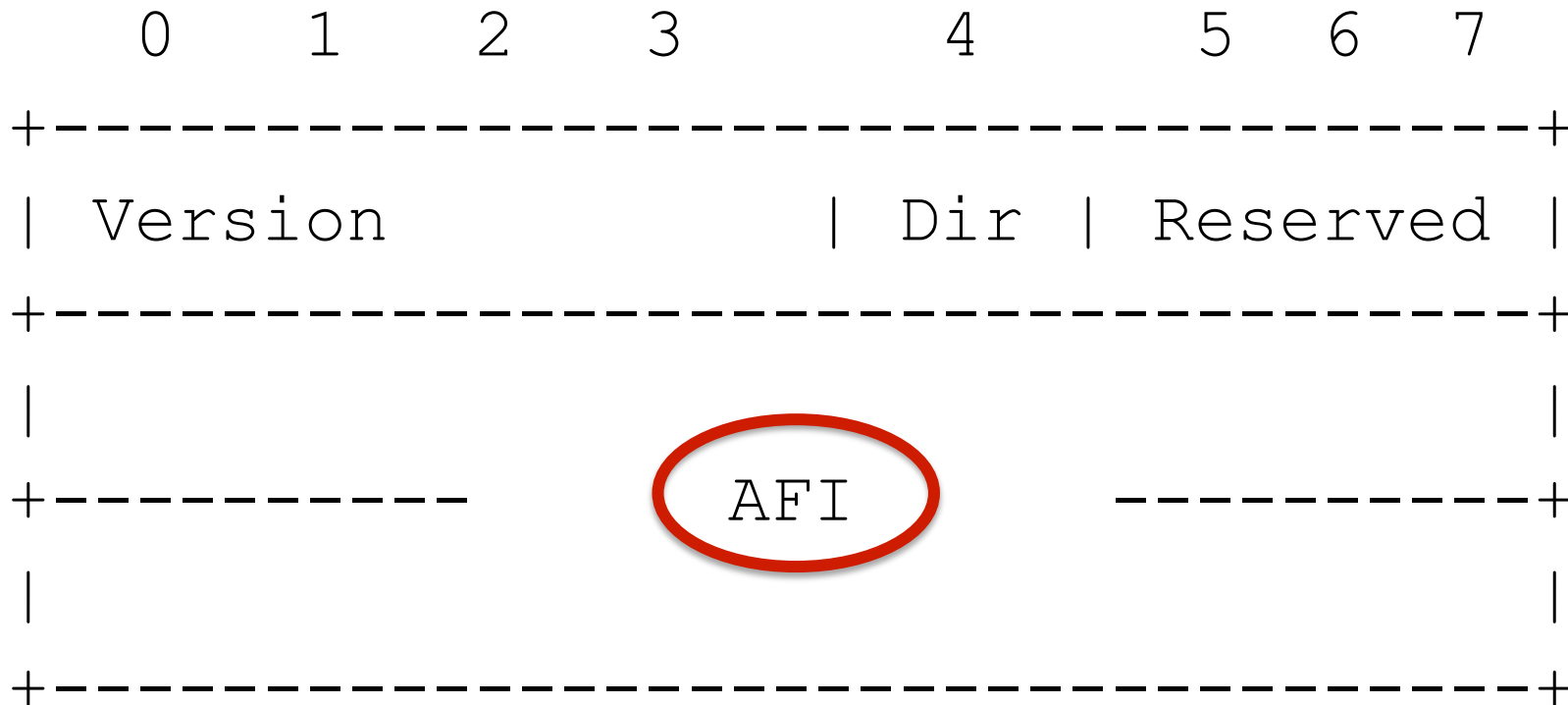
# Capability Negotiation

- Design Decision:
  - Don't send signatures unless you know your peer understands them
- If you support BGPSEC
  - You should support draft-ietf-idr-bgp-extended-messages
  - You must support 4-byte AS Numbers (RFC 4893)

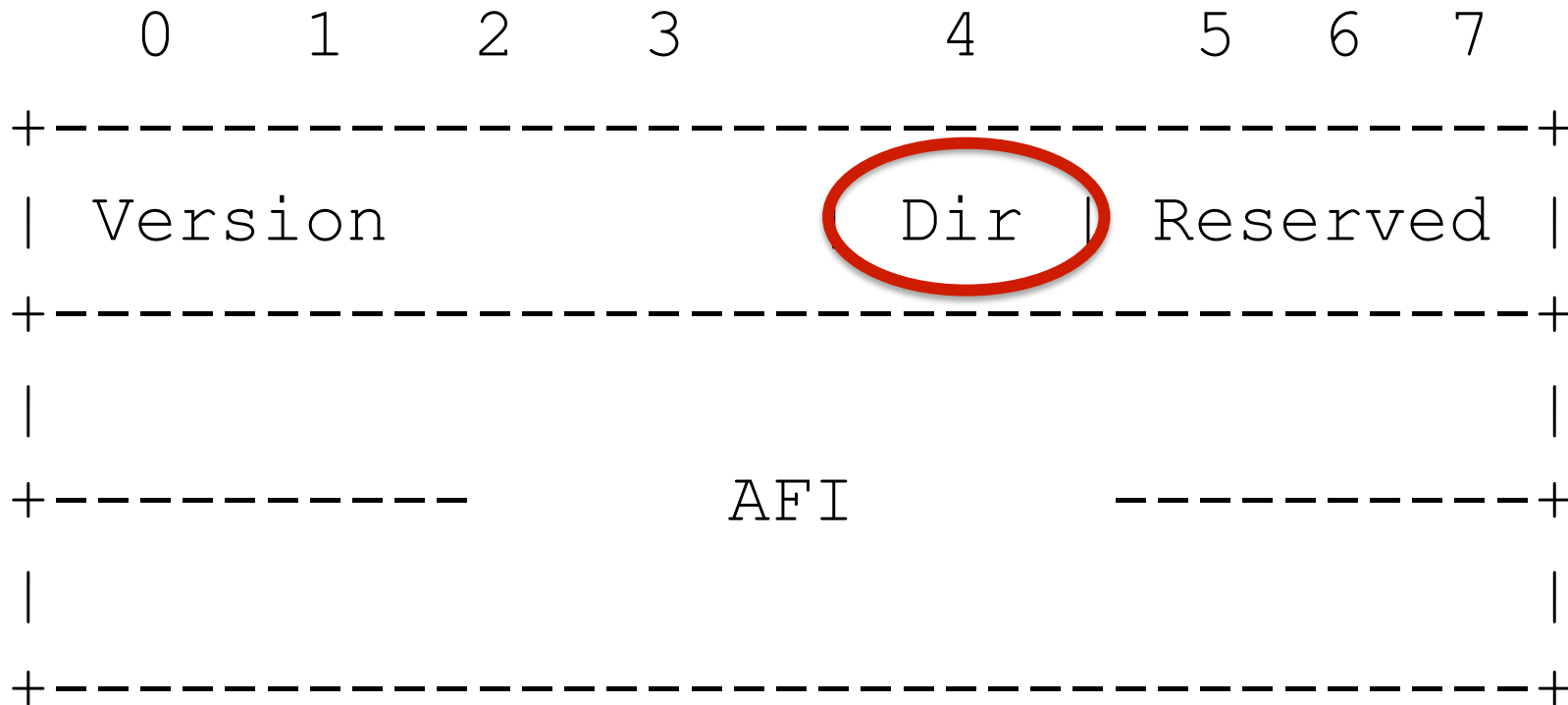
# Capability Negotiation



# Capability Negotiation

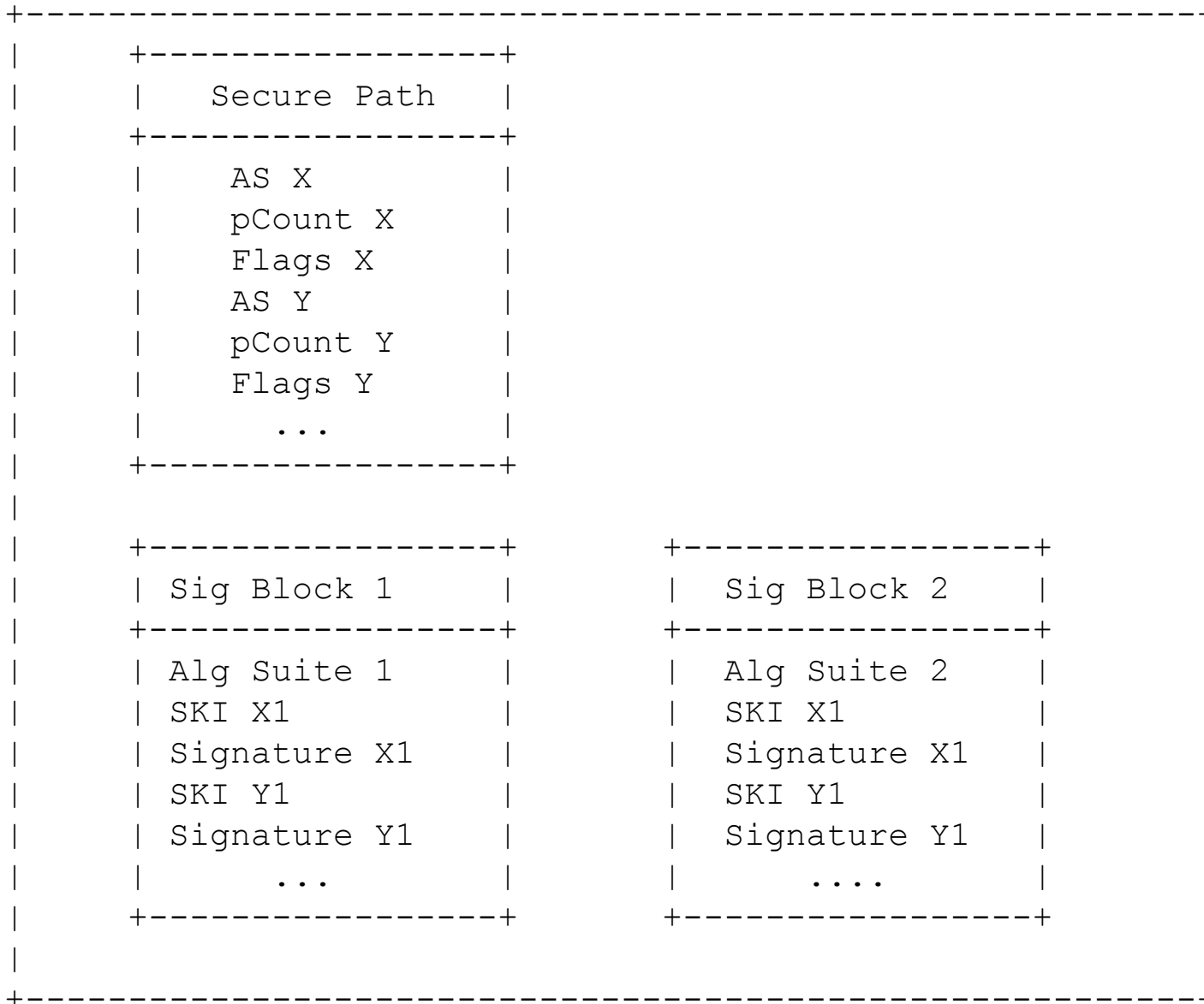


# Capability Negotiation

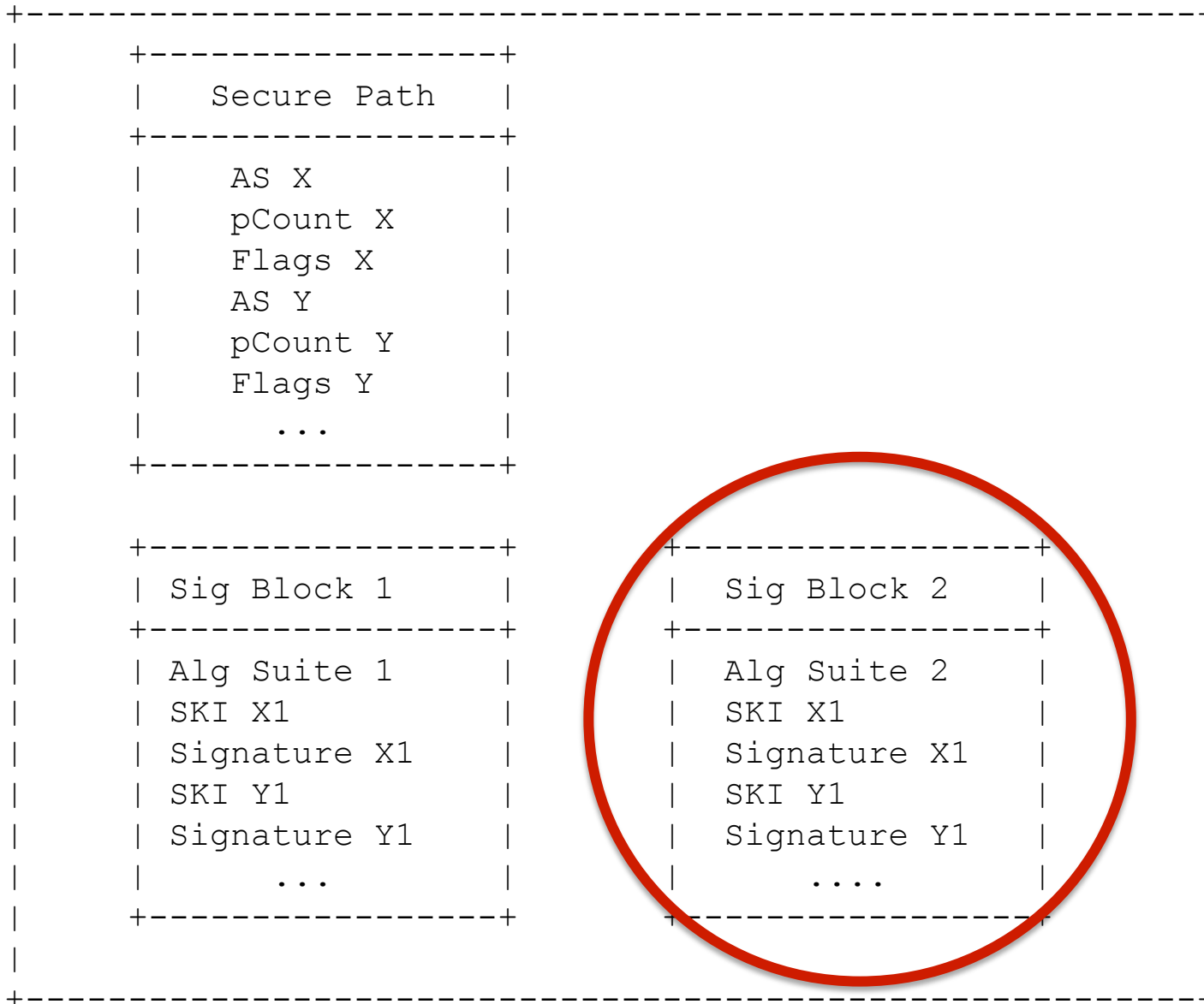


# Capability Negotiation

- Negotiation is done separately for each address family
  - Current specification supports only IPv4 and IPv6
- Send and Receive are negotiated separately
  - Sending signatures is easier than validating signatures
  - We anticipate that “stub” ASes may wish to send BGPSEC signatures but not receive







# The BGPSEC\_Path Attribute

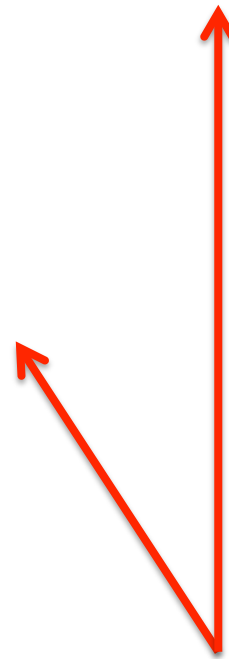
- Contains the AS Path information and a signature attesting to each hop of the AS Path
- An update will either contain BGPSEC\_Path (path security) or AS\_Path (no path security) but not both
- Routers will pull path information from BGPSEC\_Path into their internal AS path format and use this for everything that AS\_Path is used for

## Secure\_Path

Secure_Path Length	(2 octets)
One or More Secure_Path Segments	(variable)

## Secure\_Path Segment

AS Number	(4 octets)
pCount	(1 octet)
Flags	(1 octet)



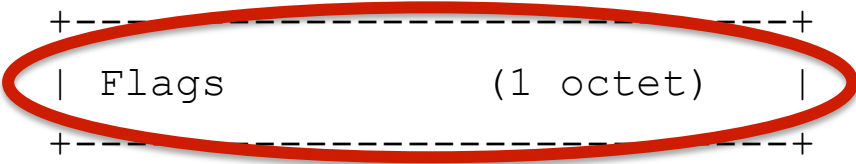
**This is the data that is being signed!**

## Secure\_Path

```
+-----+
| Secure_Path Length          (2 octets) |
+-----+
| One or More Secure_Path Segments (variable) |
+-----+
```

## Secure\_Path Segment

```
+-----+
| AS Number          (4 octets) |
+-----+
| pCount            (1 octet)  |
+-----+
| Flags              (1 octet)  |
+-----+
```



## Secure\_Path

Secure_Path Length	(2 octets)
One or More Secure_Path Segments	(variable)

## Secure\_Path Segment

AS Number	(4 octets)
pCount	(1 octet)
Flags	(1 octet)

# BGPSEC\_Path

- The Flags field allows us to have a couple extra bits per hop that are protected by the signature
  - Currently only one flag bit defined
  - Bit is set whenever you would otherwise be in an AS\_Confed\_Sequence [RFC 5065]
- pCount allows for multiple copies of an AS number without multiple signatures

## **Signature\_Block**

+-----+		
Signature_Block Length	(2 octets)	
+-----+		
Algorithm Suite Identifier	(1 octet)	
+-----+		
Sequence of Signature Segments	(variable)	
+-----+		

## **Signature Segments**

+-----+		
Subject Key Identifier	(20 octets)	
+-----+		
Signature Length	(2 octets)	
+-----+		
Signature	(variable)	
+-----+		

### Signature\_Block

+-----+		
Signature_Block Length	(2 octets)	
+-----+		
Algorithm Suite Identifier	(1 octet)	
+-----+		
Sequence of Signature Segments	(variable)	
+-----+		

### Signature Segments

+-----+		
Subject Key Identifier	(20 octets)	
+-----+		
Signature Length	(2 octets)	
+-----+		
Signature	(variable)	
+-----+		



# BGPSEC\_Path

- It is very important that we all use the same signing algorithm
  - See draft-ietf-sidr-bgpsec-algs
  - The signature that I create needs be verifiable by everyone else (not just my neighbor)!
- Subject Key Identifier just helps us find the right certificate to use in verifying the signature

# Validation

- BGPSEC path security is intended as a complement to (RPKI) origin security
- Two validation states:
  - Either a path has a valid signature chain
  - ... or else it doesn't
- What one does with the validation result (i.e., policy) is up to them

# Validation

- We anticipate that an AS will validate the signatures once on the edge of the AS
  - Then use whatever mechanism they choose to signal validation state within their AS.  
(E.g., maybe they set a community)
  - We don't need to standardize how one signals validation state within an AS
  - It is also perfectly fine for each router to do its own validation

# Partial Deployment

- If your peer doesn't support BGPSEC, then you send them unsigned messages
- We can't provide useful security guarantees unless the entire path supports BGPSEC
  - This means if you get an unsigned message, you propagate that route unsigned
  - ... although maybe some of your customers let you sign on their behalf

# Partial Deployment

- An AS doesn't need to upgrade to BGPSEC all at once
  - Signatures get stripped off when moving from a BGPSEC speaker to a non-BGPSEC peer regardless of whether the peer is internal or external
- The document contains an procedure for converting a BGPSEC-signed update to an unsigned update

# Final Slide

- The document contains a lot more details. Please read it if you are interested
- This is a great time to provide feedback
  - The specification is stable (no longer a rapidly moving target)
  - ... but it is not too late to make changes based on your feedback