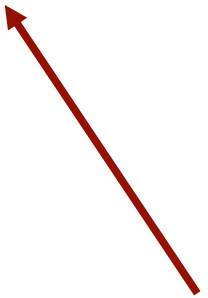
# BGPSEC Protocol

(From -01 to -02 and on to -03)

Matt Lepinski

# Agenda

- Overview
- BGPSEC Capability
- Format of BGPSEC_Path_Signatures
- Replay Protection



**Key Topic Today**

# Overview:
# What's happened since Taipei?

- Lots of feedback on attribute formats
  - Thanks to all who have provided feedback!
  - If you are thinking about implementation please let me know what you think of the -03 formats
- Feb. Interim after NANOG in San Diego
  - Thanks to everyone who showed up!
  - Replay Protection
  - Route Leaks

# Overview: Document Status

- I published an -02 version based on discussions at the February interim
- Since then I have gotten feedback that the attribute formats in the -02 version are not implementation-friendly
- An -03 version is in the drafts archive today addresses this feedback
- I will talk about the -02 and -03 versions in this presentation
- Goal: Publish a version of the document in May that has a stable attribute format

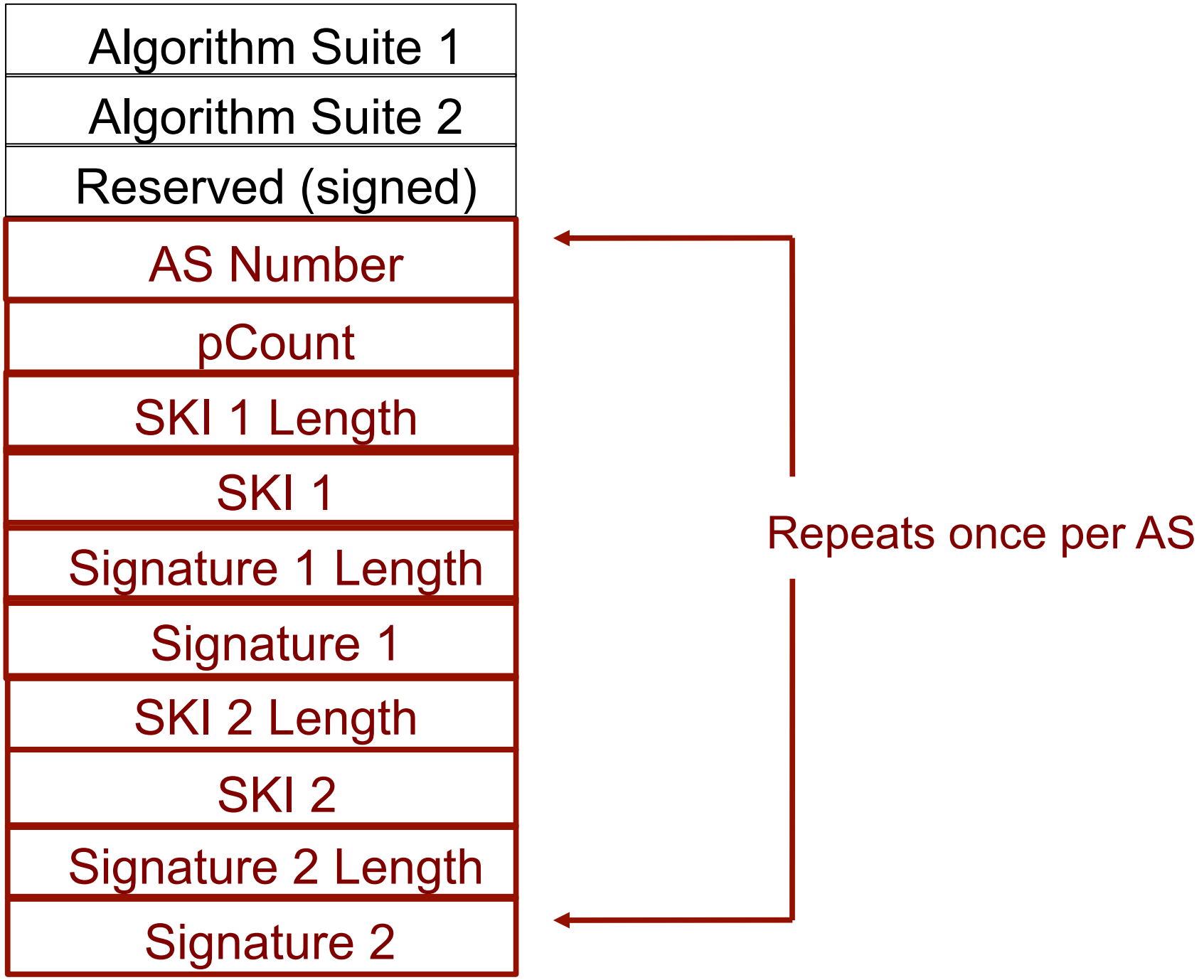
# BGPSEC Capability

- Change from -01 to -02:
  - BGPSEC Capability uses exactly the same AFI/ SAFI format as RFC 4760
  - Namely: 2-octet AFI and 1-octet SAFI
  - Note that for this version of the spec, BGPSEC is still only defined for IPv4 and Ipv6.

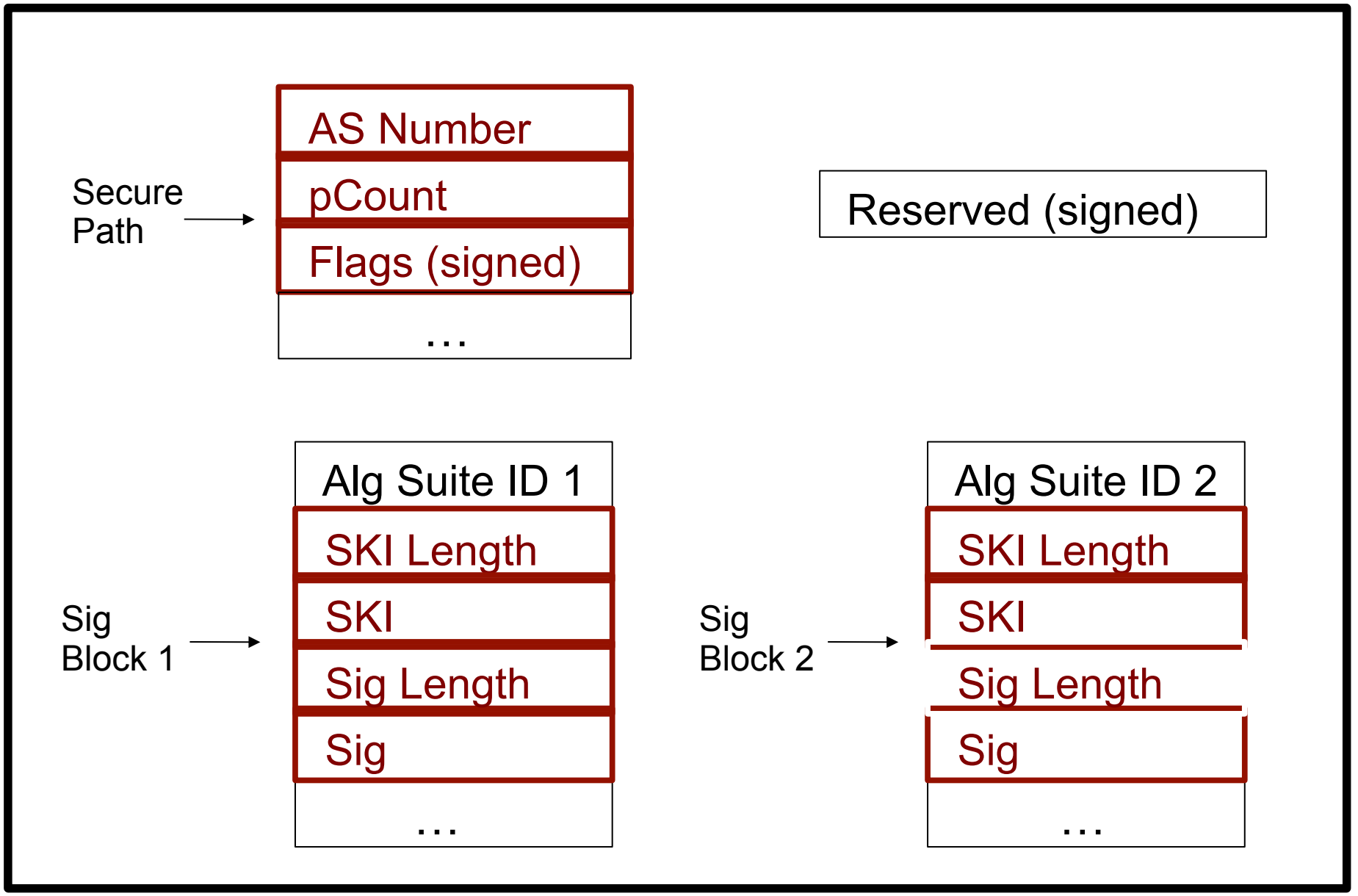
# Path_Signatures Format

- Feedback from -01:
  - Don't use AS_Path for data that is semantically different than BGP-4 AS_Path
  - Don't duplicate data
    - It just creates more error cases!
  - Bring AS number into the BGPSEC_Path_Signatures attribute
    - So the validation procedure doesn't need to hunt for it
  - Make sure all of the lengths are explicit

# First Try: The -02 Format



# Next Try : The -03 Format



# A Couple Notes on the -03 Format

- We will talk about “Reserved” later with regards to replay protection
- Note that the Secure_Path and both Signature_Blocks have their own length field (which didn't fit conveniently in the diagram)
- Since AS number is now in the BGPSEC_Path_Signatures attribute, to avoid repeating data we DO NOT include AS4_Path in signed update messages
- Note that Flags could be used in future for something like “customer”/”transit” coloring

# SKI Length ?

- Recall, that the (AS, SKI) pair is used to look up a key (from a valid certificate) in your cache
- Currently, the RPKI specs mandate a 20 byte SKI
- Within your AS you should have two different keys with the same SKI
  - And a collision occurs, just generate a new key
- Is there any reason we would ever want to change the RPKI specs to allow longer SKIs?

# Replay Protection

- Version -01 had an Expire-Time in the BGPSEC_Path_Signatures
- Used a “beaconing” mechanism in which each prefix was re-advertised periodically
  - With a new Expire-Time and a new Signature
- Goal was to protect against replay of stale signatures
  - E.g., Business relationship changes, but old business partner still has my signature saying that path from me to old partner is valid

# Replay Protection

- Disadvantages of the -01 use of Expire-Time were discussed at length at the Feb interim
- Consensus was to remove this mechanism from the -02 draft
  - Needs more discussion!
- -02 and -03 have there is a “reserved” field
  - Sender sets to zero, receiver ignores value
  - Should permit backwards-compatible introduction of an Expire-Time variant in the future, if desired

# Summary of Proposed Mechanisms

- Expire-Time mechanism from -01
- Key Issue:
  - Frequent sending of “beacons” inflicts high cost on the routing system
  - There may be an incentive to set low expire time (frequent beacons) to gain “better” protection

# Summary of Proposed Mechanisms

- Use of RPKI to invalidate old signatures
- Idea:
  - Revoke a certificate in the case of suspected vulnerability to replay attack (e.g., business relationship changes)
- Key Issue:
  - Frequent revocation/expiration of RPKI certificates would have high network cost
  - Idea needs to be made more concrete if we are to give advice to BGPSEC operators

# Summary of Proposed Mechanisms

- Expire-Time with Coarse Granularity
- Idea:
  - Set units of Expire-Time to be something like “days”
  - Provides some protection with less network load
- Key Issue:
  - More discussion is needed regarding how to set the correct units.
  - Very difficult to change units if we choose poorly

# Summary of Proposed Mechanisms

- Signing-Time with implicit validity period
- Idea:
  - Validity period not chosen by the sender, all signatures have a (relatively long) implicit validity period
- Key Issue:
  - More discussion is needed regarding how to set the implicit validity period
  - Very difficult to change implicit validity period if we choose poorly