CS558 Network Security

Lecture 7: BGPSec

hello world

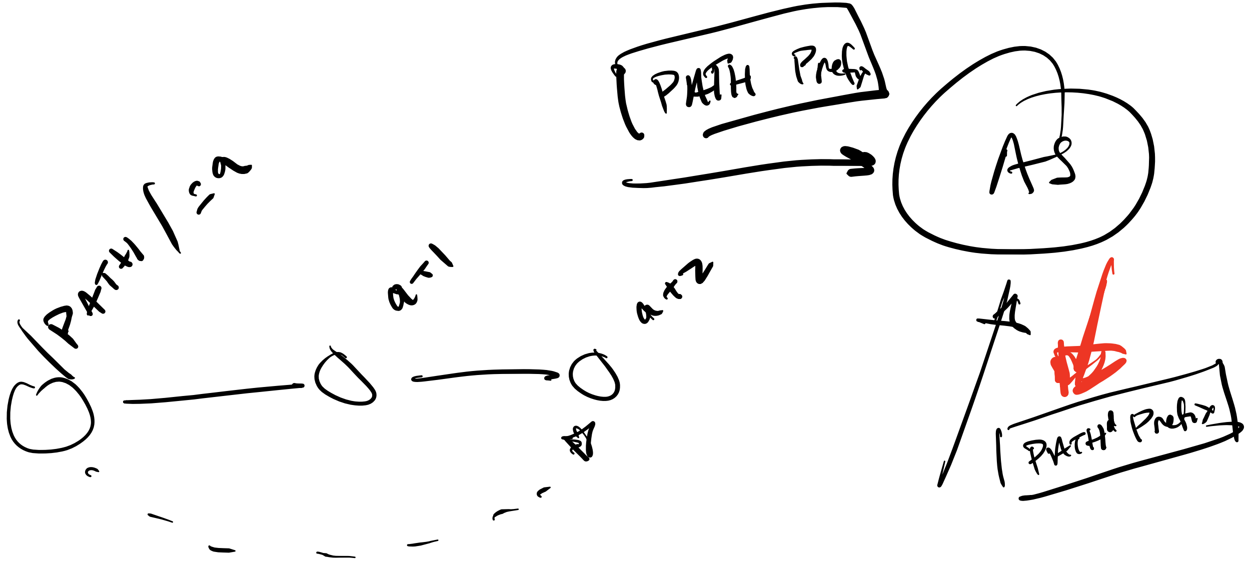
BOSTON UNIVERSITY

BGP Hijacking

PATH 128.220/16

PATH' 128.220.1/24

- Sub-prefix Hijack → Scenario
- Hijack with Shorter AS_PATH (One Hop Attack)



$|path'| < |path|$

Defense: RPKI (Resource Public Key Infrastructure)

Eliminate Subprefix hijack

RIR → pk_{RIR}, sk_{RIR}

Big ISP → pk_{BISP} sk_{BISP} $Sign_{sk_{RIR}}(\text{"}pk_{BISP}\ /8\text{"}) \rightarrow \sigma_{RIR}$

meaningful

ISP → pk_{ISP} sk_{ISP} $Sign_{sk_{BISP}}(\text{"}pk_{ISP}\ /16\text{"}) \rightarrow \sigma_{BISP}$

AS → pk_{AS} sk_{AS} $Sign_{sk_{ISP}}(\text{"}pk_{AS}\ /24\text{"}) \rightarrow \sigma_{ISP}$

ROA

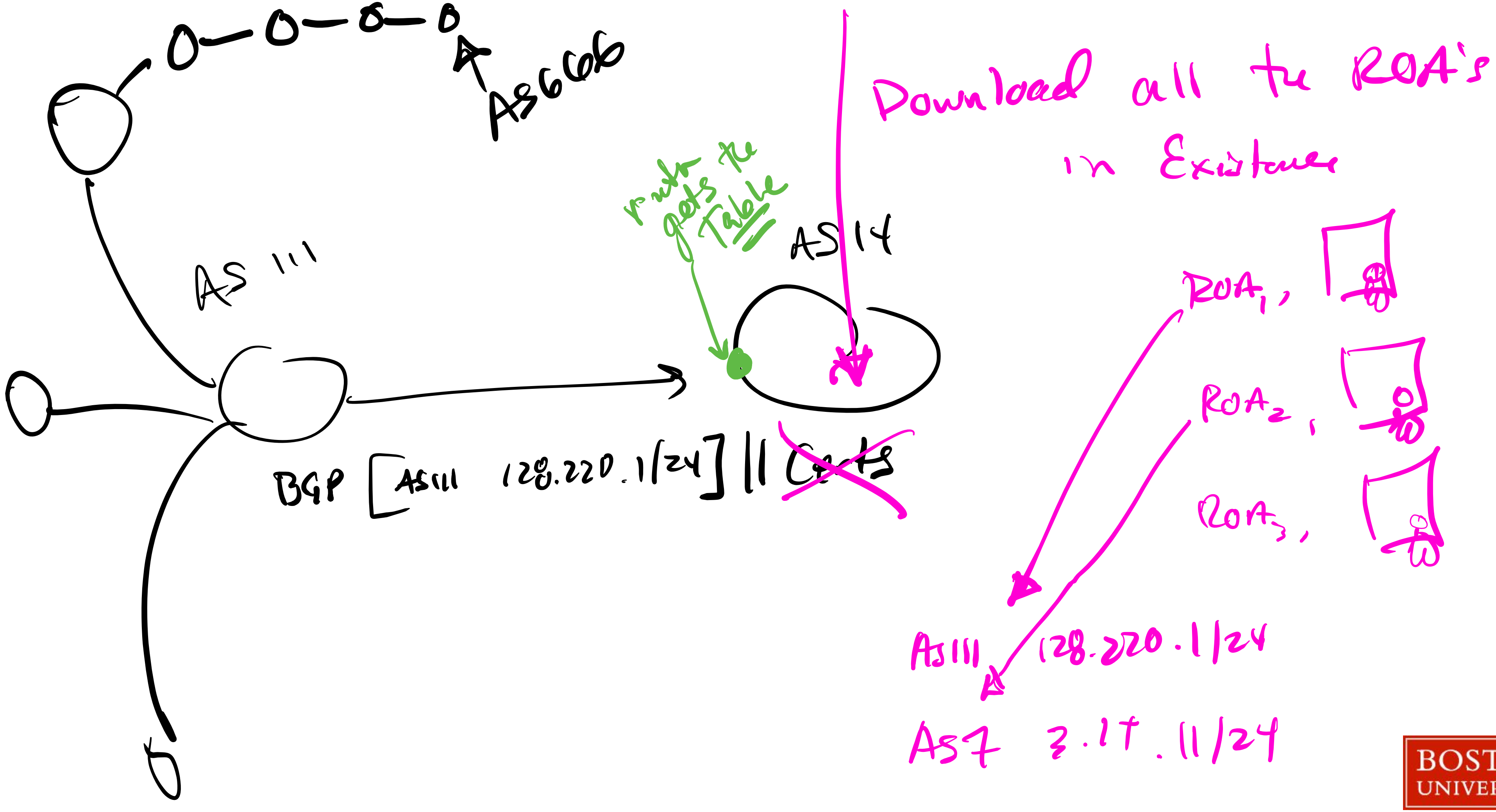
ROA $Sign_{sk_{AS}}(\text{"[AS111 128.220.1/24]"}) \rightarrow \sigma_{ROA}$

ROA

[AS111   128.220.1/24]

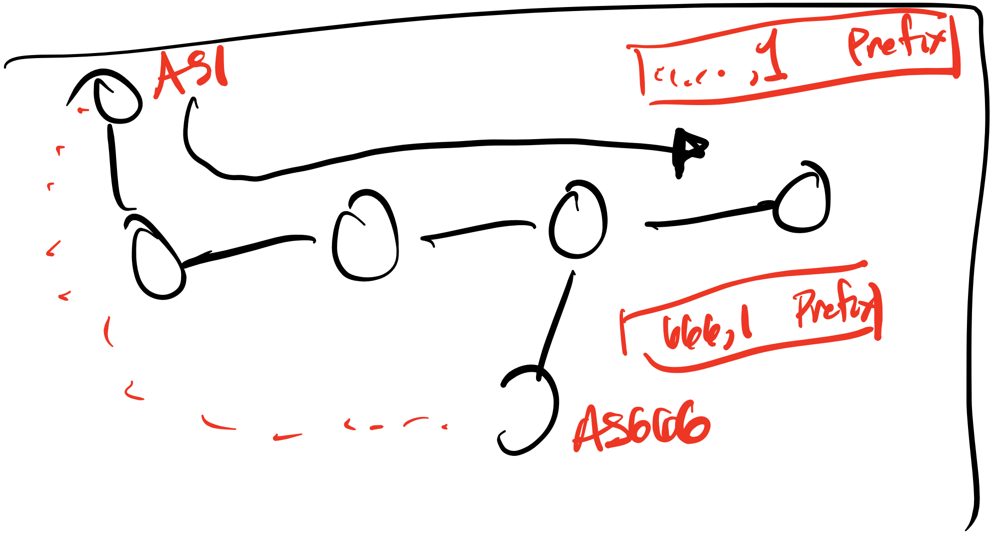
Digitally Signed

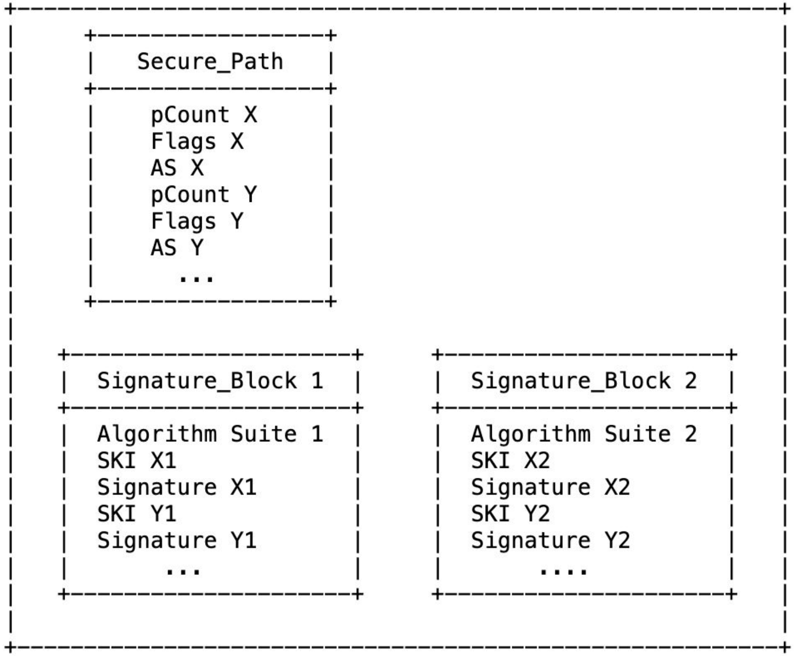
BOSTON UNIVERSITY


AS6606
AS 111
router gets the Table
AS 14
BGP [AS111 128.220.1/24] || ~~Certs~~
Download all the ROA's in Existence
ROA_1,
ROA_2,
ROA_3,
AS111 128.220.1/24
AS7 3.17.11/24

Defense: BGPSec – An Extension of The BGP Protocol

- BGPSec_Path
  - Secure_Path
  - Signature_Block





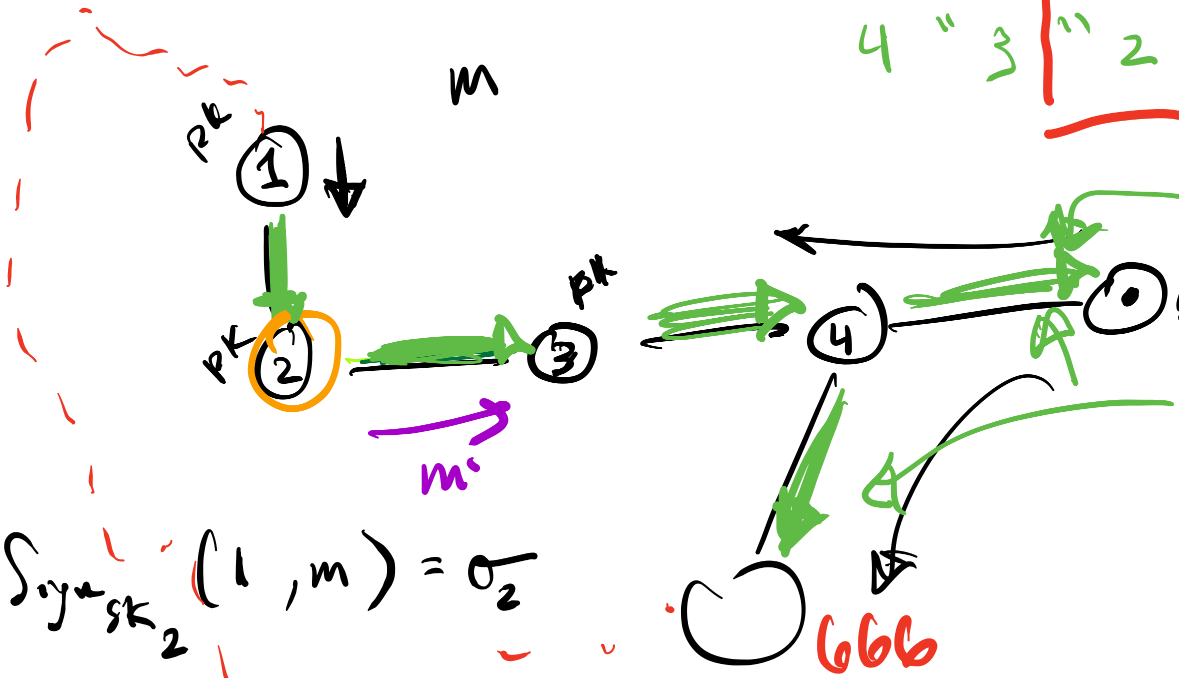
```
+-----------------------------------------------------------------+
|     +-----------------+                                          |
|     |   Secure_Path   |                                          |
|     +-----------------+                                          |
|     |    pCount X     |                                          |
|     |    Flags X      |                                          |
|     |    AS X         |                                          |
|     |    pCount Y     |                                          |
|     |    Flags Y      |                                          |
|     |    AS Y         |                                          |
|     |      ...        |                                          |
|     +-----------------+                                          |
|                                                                  |
|   +---------------------+    +---------------------+             |
|   |  Signature_Block 1  |    |  Signature_Block 2  |             |
|   +---------------------+    +---------------------+             |
|   |  Algorithm Suite 1  |    |  Algorithm Suite 2  |             |
|   |  SKI X1             |    |  SKI X2             |             |
|   |  Signature X1       |    |  Signature X2       |             |
|   |  SKI Y1             |    |  SKI Y2             |             |
|   |  Signature Y1       |    |  Signature Y2       |             |
|   |   ...               |    |   ....              |             |
|   +---------------------+    +---------------------+             |
|                                                                  |
+-----------------------------------------------------------------+
```

Figure 2: High-Level Diagram of the BGPsec_PATH Attribute

High Level Idea + Options

$3 \text{ " } 2 \text{ " } 1 \;\; \sigma_2 \text{ " } \sigma_3 \text{ " } \sigma_4$

$4 \text{ " } 3 \text{ " } 2 \;\; \sigma_1 \text{ " } \sigma_2 \text{ " } \sigma_3$



$5, 4, 3, 2, \sigma_1, \sigma_2, \sigma_3, \sigma_4$

$666, 4, 3, 2, \sigma_1, \sigma_2, \sigma_3, \sigma_4$

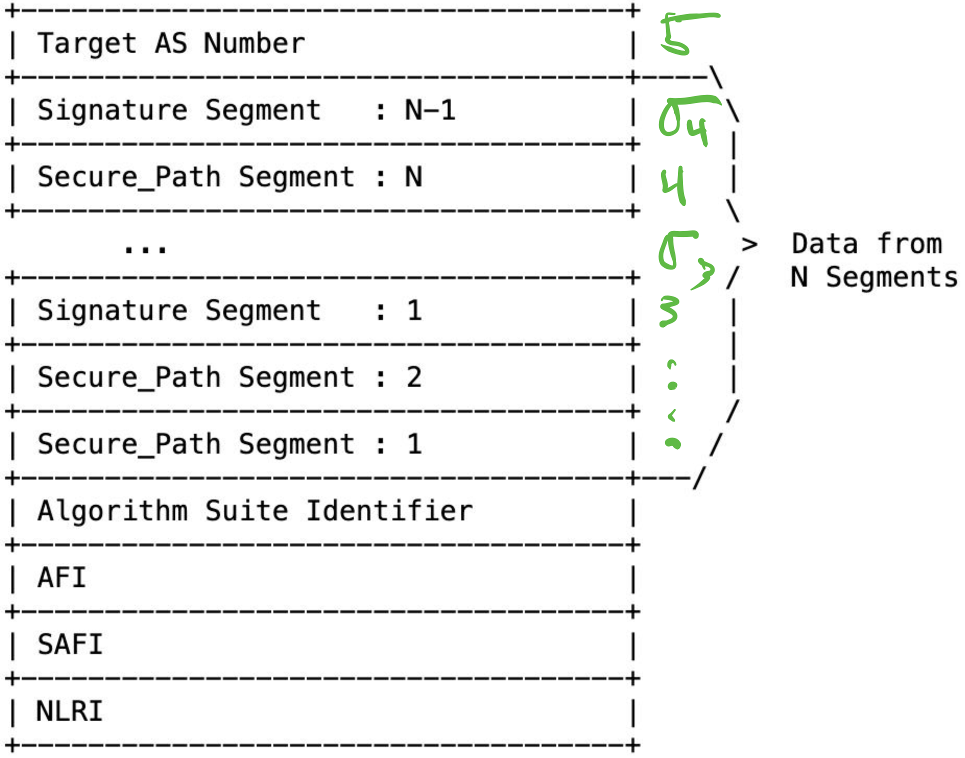
$\mathrm{Sign}_{sk_2}(1, m) = \sigma_2$

$\mathrm{Sign}_{sk_1}(2, m) = \sigma_1$

$\text{" } 1 \;\; \sigma_{666} \text{ "}$

- As #
- where it came from
- Sign when its going
- prefix

BGPSec Signing

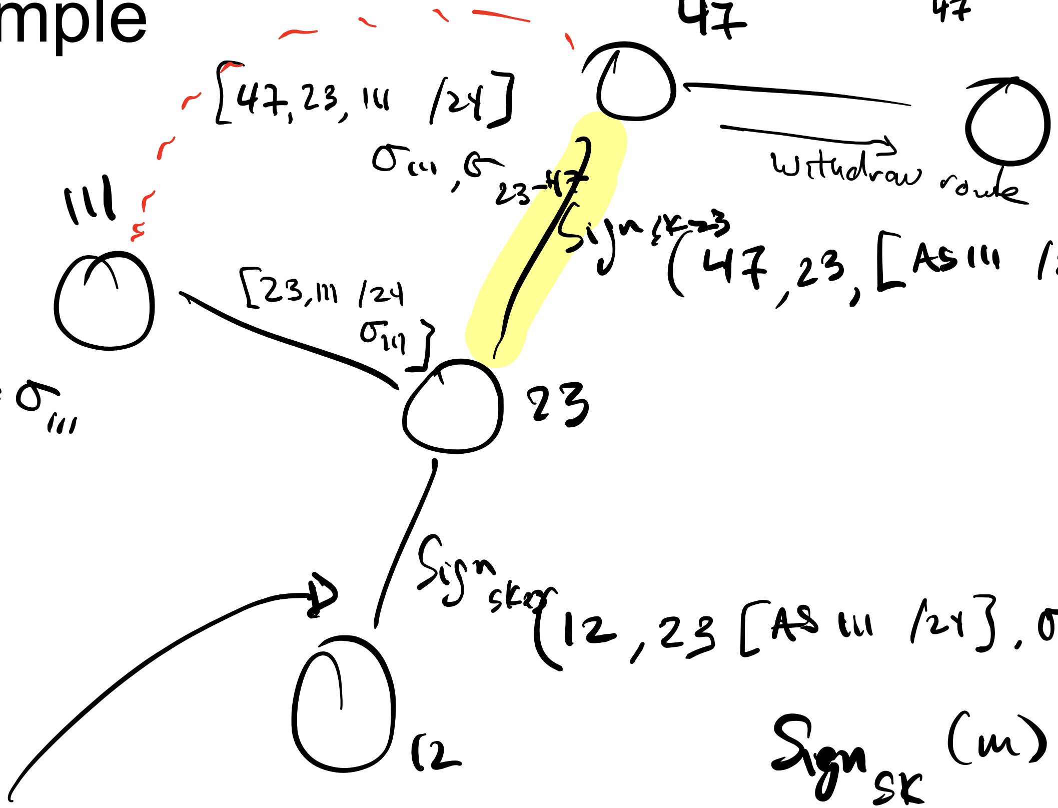


```
+------------------------------------+
| Target AS Number                   |
+------------------------------------+----\
| Signature Segment   : N-1          |     \
+------------------------------------+     |
| Secure_Path Segment : N            |     |
+------------------------------------+     \
       ...                                  >  Data from
+------------------------------------+     /   N Segments
| Signature Segment   : 1            |     |
+------------------------------------+     |
| Secure_Path Segment : 2            |     |
+------------------------------------+     /
| Secure_Path Segment : 1            |    /
+------------------------------------+---/
| Algorithm Suite Identifier         |
+------------------------------------+
| AFI                                |
+------------------------------------+
| SAFI                               |
+------------------------------------+
| NLRI                               |
+------------------------------------+
```

Figure 8: Sequence of Octets to Be Hashed

BGPSec Example

$\text{Sign}_{sk_{47}}(13, 47, 23 [AS111 \;/24], \sigma_{111}, \sigma_{23-47}) = \sigma_{47}$



$[47, 23, 111 \;/24]$ $\sigma_{111}, \sigma_{23-47}$

$\text{Sign}_{sk_{23}}(47, 23, [AS\,111 \;/24], \sigma_{111}) = \sigma_{23-47}$

$\text{Sign}_{sk_{111}}(23, [AS\,111 \;/24]) = \sigma_{111}$

$[23, 111 \;/24 \;\sigma_{111}]$

$\text{Sign}_{sk_{23}}(12, 23 [AS\,111 \;/24], \sigma_{111}) = \sigma_{23-12}$

$[12, 23, 111 \;/24] \;\sigma_{111}, \sigma_{23-12}$

$\text{Sign}_{sk}(m) = \sigma$

$\text{Verify}_{pk}(m, \sigma) = 1$

BGP Hijacking

RPKI ← almost no computational ~~[illegible]~~ overhead

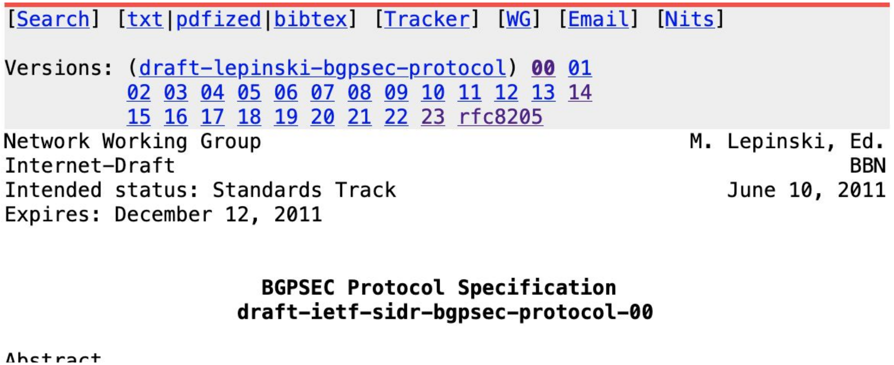
- Sub-prefix Hijack
- Hijack with Shorter AS_PATH (One Hop Attack)
- + Replay Attacks

BGPSec ← A lot of overhead ~~[illegible]~~

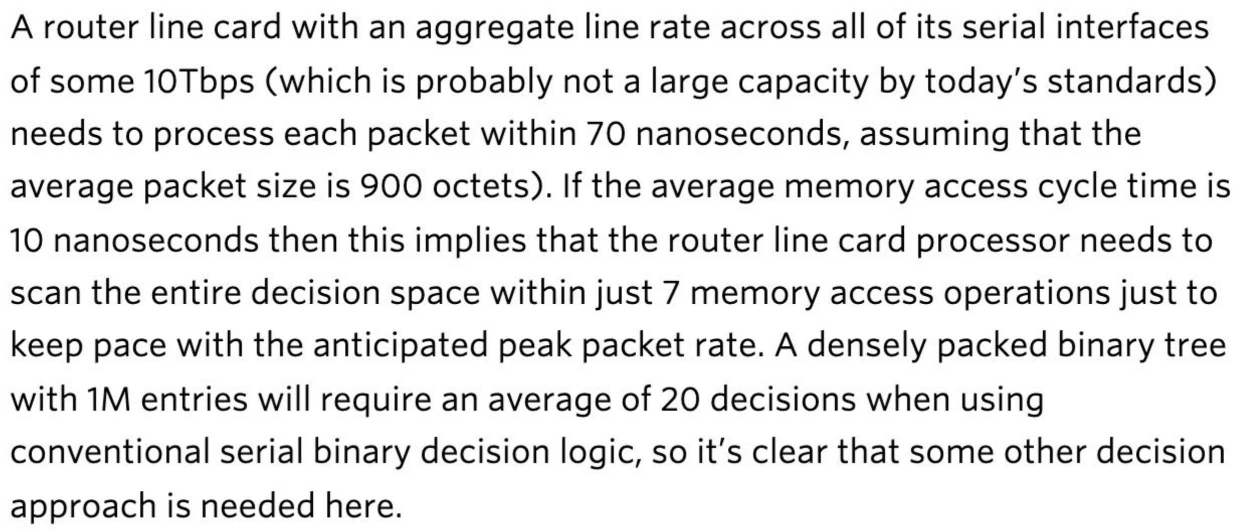
BOSTON UNIVERSITY

BGPSec “downsides”

- Need to validate Signatures on router (RAM and CPU problems)
- Change the maximum BGP message size
- Reduced Dynamism
- Getting the Cryptography Right
- “Future proofing” is hard

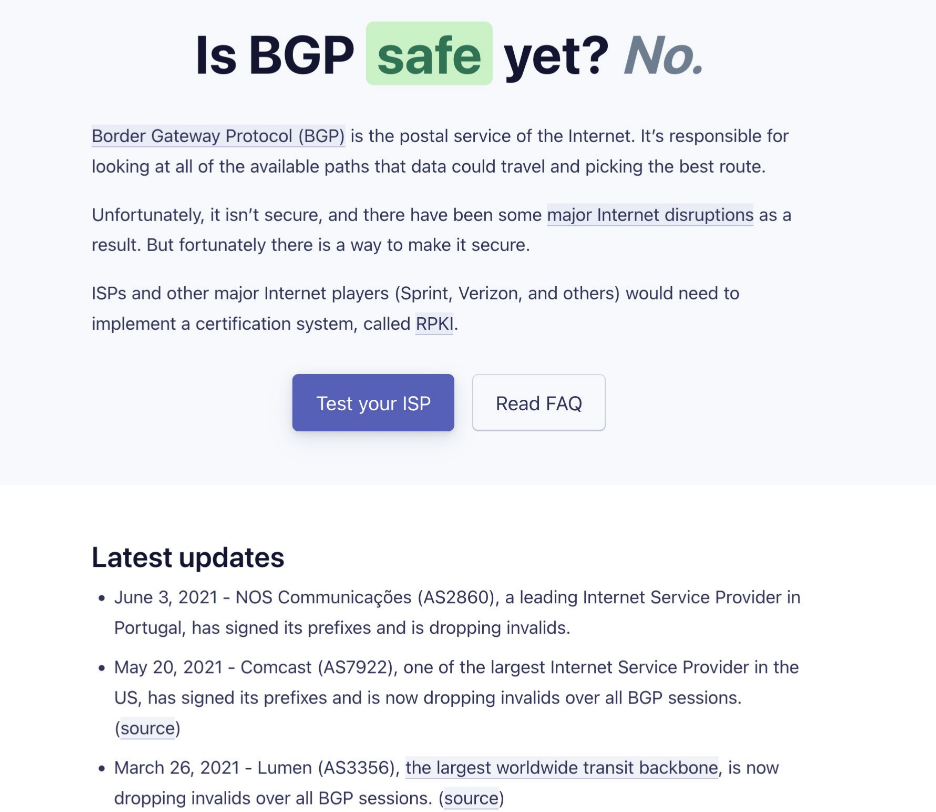


```
[Search] [txt|pdfized|bibtex] [Tracker] [WG] [Email] [Nits]

Versions: (draft-lepinski-bgpsec-protocol) 00 01
          02 03 04 05 06 07 08 09 10 11 12 13 14
          15 16 17 18 19 20 21 22 23 rfc8205
Network Working Group                                    M. Lepinski, Ed.
Internet-Draft                                                        BBN
Intended status: Standards Track                            June 10, 2011
Expires: December 12, 2011


                     BGPSEC Protocol Specification
                   draft-ietf-sidr-bgpsec-protocol-00

Abstract
```



A router line card with an aggregate line rate across all of its serial interfaces of some 10Tbps (which is probably not a large capacity by today's standards) needs to process each packet within 70 nanoseconds, assuming that the average packet size is 900 octets). If the average memory access cycle time is 10 nanoseconds then this implies that the router line card processor needs to scan the entire decision space within just 7 memory access operations just to keep pace with the anticipated peak packet rate. A densely packed binary tree with 1M entries will require an average of 20 decisions when using conventional serial binary decision logic, so it's clear that some other decision approach is needed here.

https://blog.apnic.net/2022/01/06/bgp-in-2021-the-bgp-table/



Is BGP safe yet? *No.*

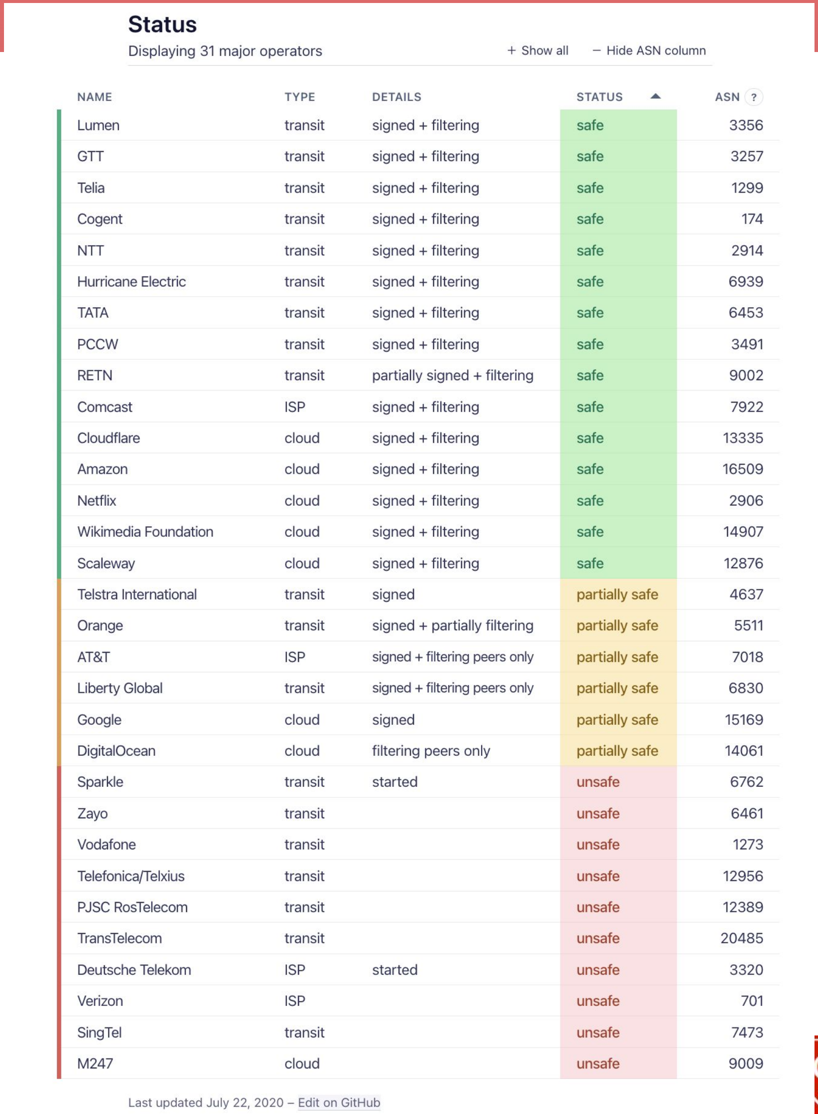
Border Gateway Protocol (BGP) is the postal service of the Internet. It's responsible for looking at all of the available paths that data could travel and picking the best route.

Unfortunately, it isn't secure, and there have been some major Internet disruptions as a result. But fortunately there is a way to make it secure.

ISPs and other major Internet players (Sprint, Verizon, and others) would need to implement a certification system, called RPKI.

Latest updates

- June 3, 2021 - NOS Communicações (AS2860), a leading Internet Service Provider in Portugal, has signed its prefixes and is dropping invalids.
- May 20, 2021 - Comcast (AS7922), one of the largest Internet Service Provider in the US, has signed its prefixes and is now dropping invalids over all BGP sessions. (source)
- March 26, 2021 - Lumen (AS3356), the largest worldwide transit backbone, is now dropping invalids over all BGP sessions. (source)

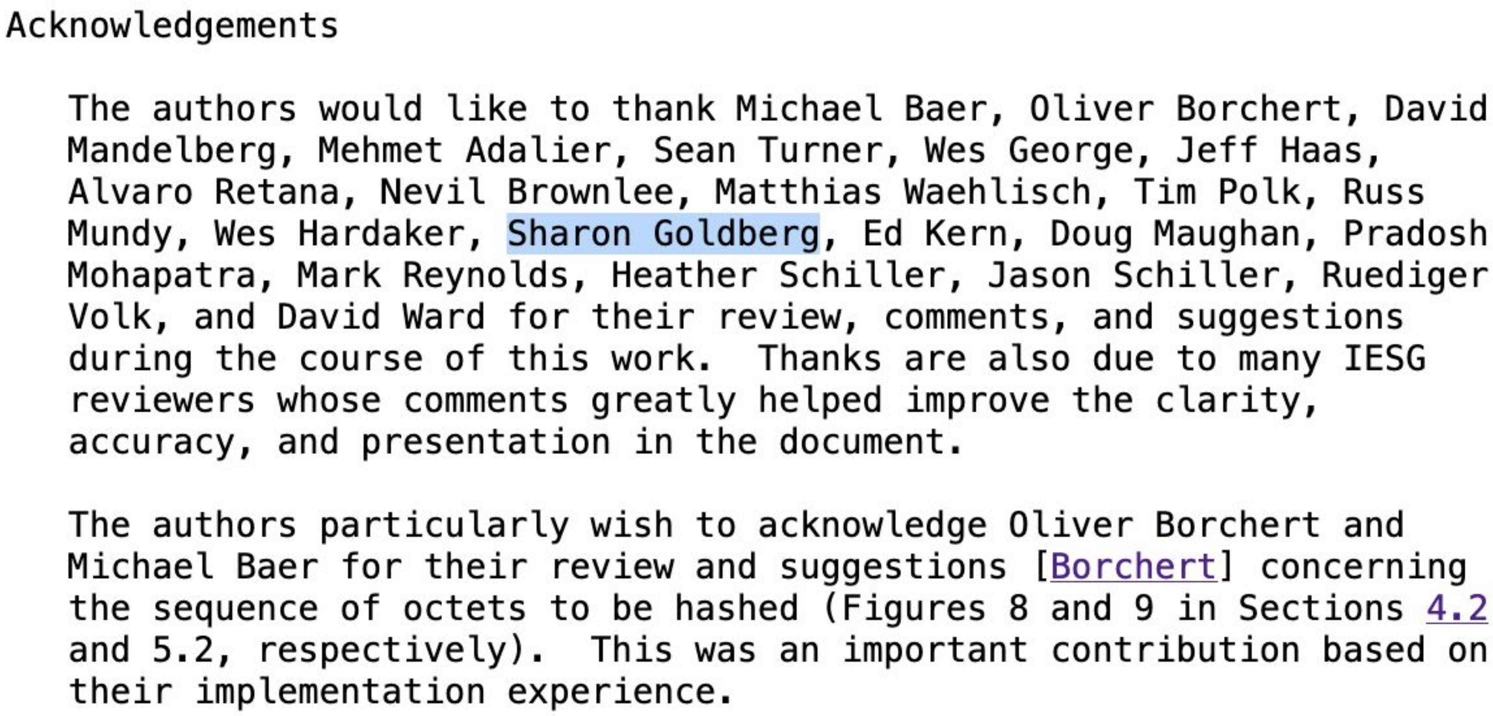


Status

Displaying 31 major operators

NAME	TYPE	DETAILS	STATUS ▲	ASN ?
Lumen	transit	signed + filtering	safe	3356
GTT	transit	signed + filtering	safe	3257
Telia	transit	signed + filtering	safe	1299
Cogent	transit	signed + filtering	safe	174
NTT	transit	signed + filtering	safe	2914
Hurricane Electric	transit	signed + filtering	safe	6939
TATA	transit	signed + filtering	safe	6453
PCCW	transit	signed + filtering	safe	3491
RETN	transit	partially signed + filtering	safe	9002
Comcast	ISP	signed + filtering	safe	7922
Cloudflare	cloud	signed + filtering	safe	13335
Amazon	cloud	signed + filtering	safe	16509
Netflix	cloud	signed + filtering	safe	2906
Wikimedia Foundation	cloud	signed + filtering	safe	14907
Scaleway	cloud	signed + filtering	safe	12876
Telstra International	transit	signed	partially safe	4637
Orange	transit	signed + partially filtering	partially safe	5511
AT&T	ISP	signed + filtering peers only	partially safe	7018
Liberty Global	transit	signed + filtering peers only	partially safe	6830
Google	cloud	signed	partially safe	15169
DigitalOcean	cloud	filtering peers only	partially safe	14061
Sparkle	transit	started	unsafe	6762
Zayo	transit		unsafe	6461
Vodafone	transit		unsafe	1273
Telefonica/Telxius	transit		unsafe	12956
PJSC RosTelecom	transit		unsafe	12389
TransTelecom	transit		unsafe	20485
Deutsche Telekom	ISP	started	unsafe	3320
Verizon	ISP		unsafe	701
SingTel	transit		unsafe	7473
M247	cloud		unsafe	9009

Last updated July 22, 2020 – Edit on GitHub



Acknowledgements

The authors would like to thank Michael Baer, Oliver Borchert, David Mandelberg, Mehmet Adalier, Sean Turner, Wes George, Jeff Haas, Alvaro Retana, Nevil Brownlee, Matthias Waehlisch, Tim Polk, Russ Mundy, Wes Hardaker, Sharon Goldberg, Ed Kern, Doug Maughan, Pradosh Mohapatra, Mark Reynolds, Heather Schiller, Jason Schiller, Ruediger Volk, and David Ward for their review, comments, and suggestions during the course of this work. Thanks are also due to many IESG reviewers whose comments greatly helped improve the clarity, accuracy, and presentation in the document.

The authors particularly wish to acknowledge Oliver Borchert and Michael Baer for their review and suggestions [Borchert] concerning the sequence of octets to be hashed (Figures 8 and 9 in Sections 4.2 and 5.2, respectively). This was an important contribution based on their implementation experience.