Potential Impact of BGPSEC Mechanisms on Global BGP Dynamics

Prepared by NIST BGP Security Team
(K. Sriram, D. Montgomery, O. Borchert, O. Kim, and P. Gleichmann)
Contact: ksriram@nist.gov dougm@nist.gov

Shared with the BGPSEC Design Team
October 16, 2009

This work was supported by the Department of Homeland Security under the Secure Protocols for the Routing Infrastructure (SPRI) program and the NIST Information Technology Laboratory Robust Inter-Domain Routing program.
https://www.nist.gov/programs-projects/robust-inter-domain-routing

Current Global BGP Dynamics

- In the normal operation of BGP, updates are generated only due to changes in reachability information:
  - A prefix is withdrawn;
  - A new prefix is announced;
  - Changes to Local Pref, AS path, etc.
- When a prefix is withdrawn, often path hunting follows which results in many unnecessary updates:
  - As a result, the update churn in the current BGP is quite high as it is.
- How is the BGP churn further impacted due to BGPSEC?

Additional Churn with Deployment of BGPSEC

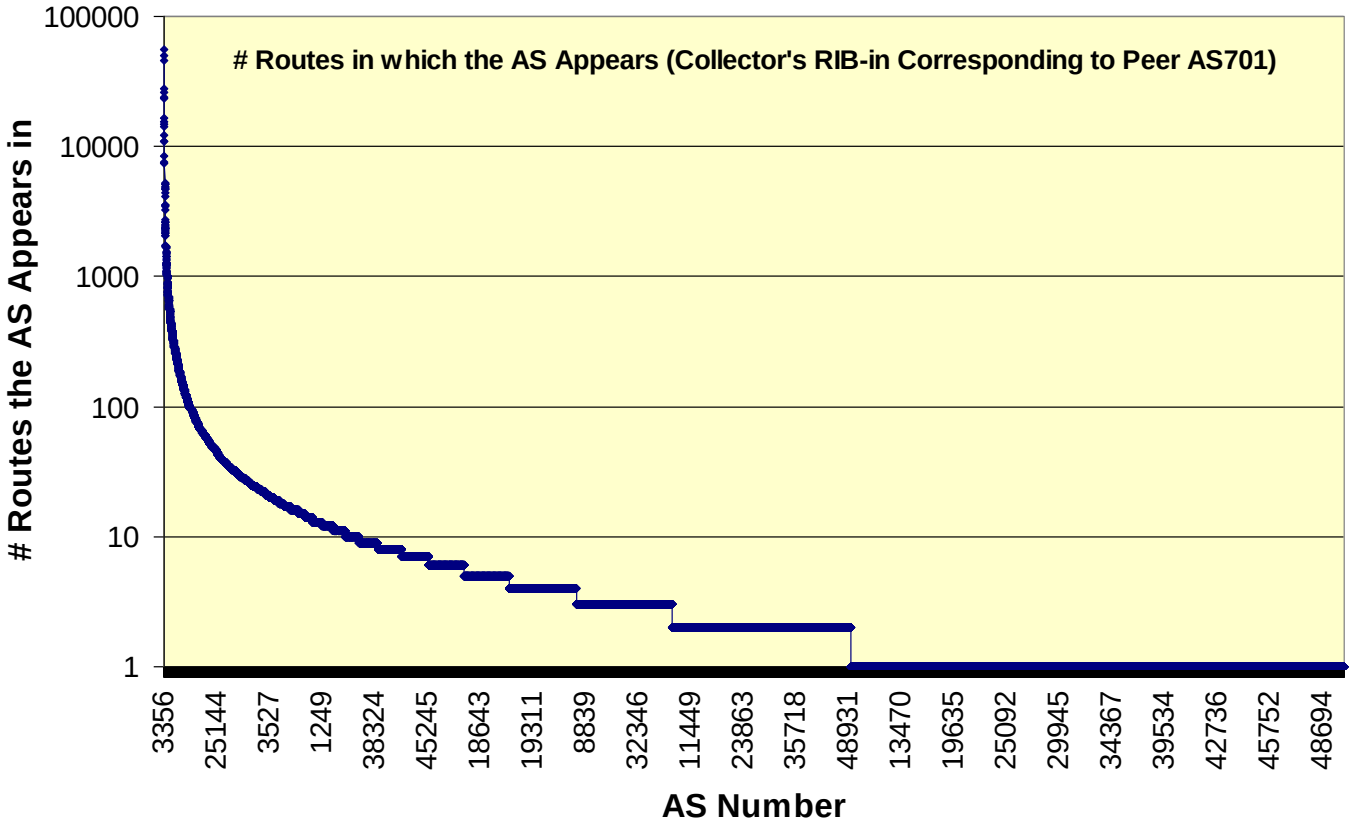
- Expiry/revocation/creation of resource certificates, path attestations, ROAs
- A new added dimension to BGP system churn:
  - The RPKI is a distributed information system entirely external to BGP, with temporal objects that effect BGP's behavior.
  - In particular the expiration or revocation of resource certificates, ROAs and attestations may lead to changes to otherwise established and stable BGP paths.
  - CRLs would also cause path hunting resulting in further BGP churn.

Modeling the Churn in BGPSEC

- Generate statistics on the total number of allocations, assignments and suballocations of address space
- Estimate the total number of ROAs, prefix certs, and AS certs
- Impact of expiry / revocation of certs, ROAs:
  - Generate statistics regarding the number of times a given AS may appear in routes in RIB
  - Estimate update churn generated at a BGP router in response to, for example, expiry / revocation of an AS cert
  - Estimate update churn generated when a prefix cert expires
  - Estimate update churn generated when a ROA is revoked, etc.

Distribution of ASes in Routes (1)

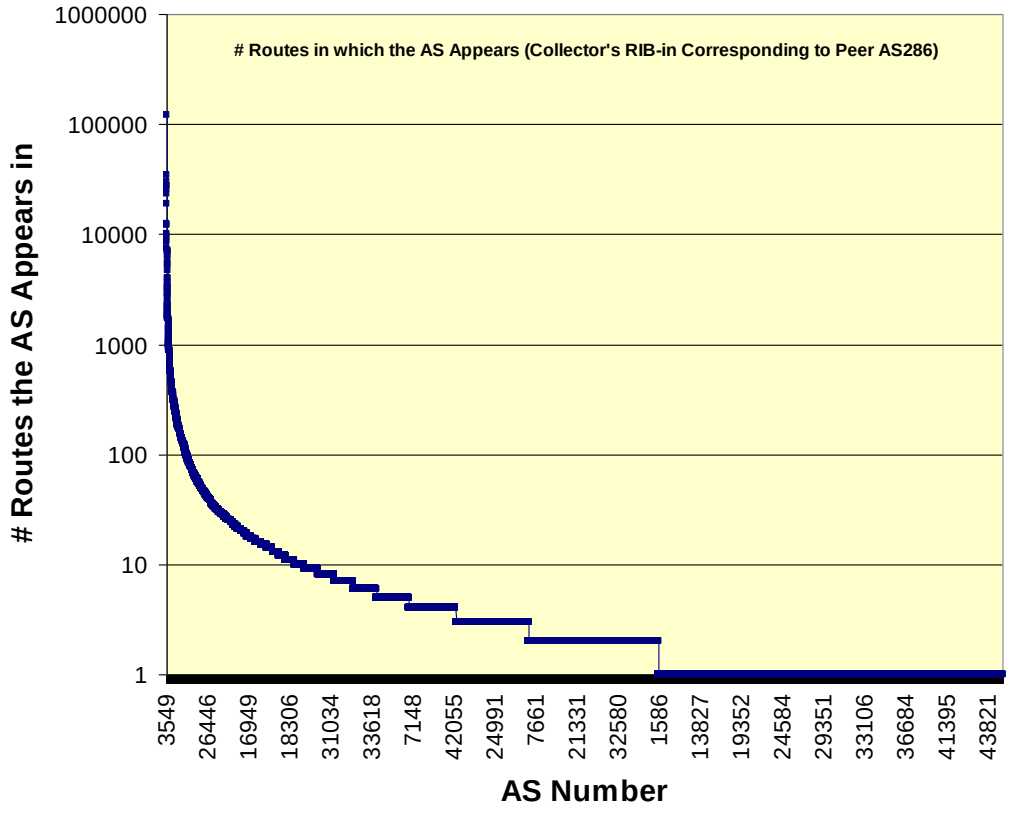
- Data from Oregon Collector
- Collector's RIB-in corresponding to Peer AS701 (UUNET)
- Total #ASes = 32002 (observed); Total # Routes = 287820
- Average # routes in which a given AS is found = 27.34



Note: Excluded AS701 (found in all 287820 routes)

Distribution of ASes in Routes (2)

- Data from rrc03 RIPE-RIS collector
- Collector's RIB-in corresponding to Peer AS286 (KPN)
- Total #ASes = 27519 (observed); Total # Routes = 244175
- Average # routes in which a given AS is found = 32.47

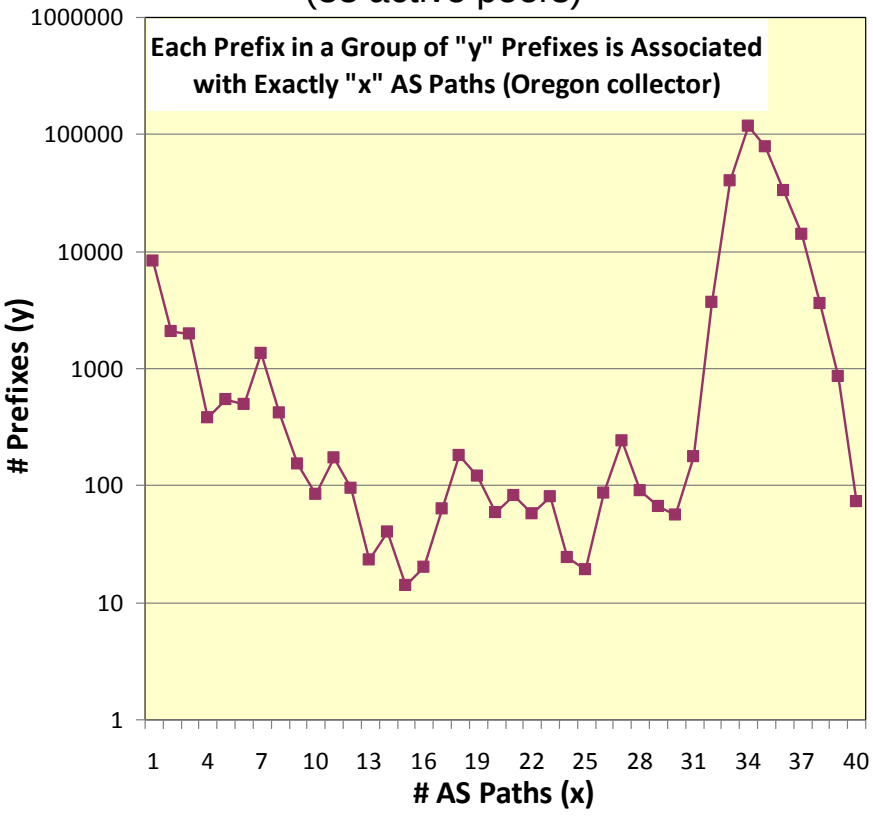


Note: Excluded AS286

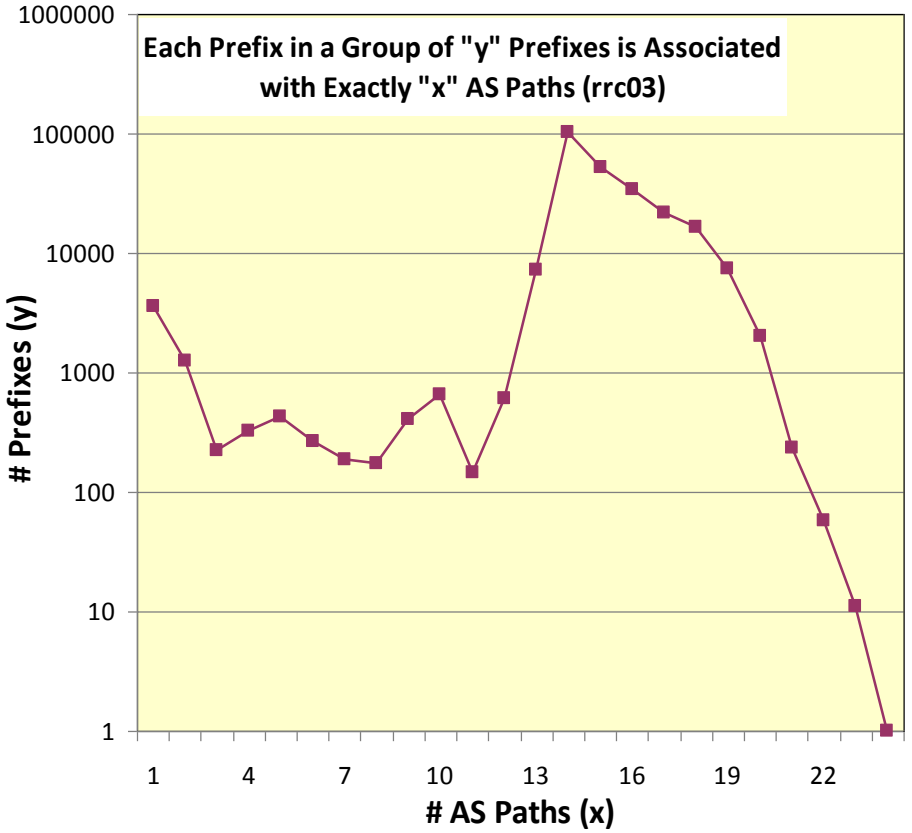
(found in all 244175 routes)

Distribution of Prefixes in Routes

Data from Oregon Collector: # Prefixes = 304722
(33 active peers)



Data from rrc03 RIPE-RIS collector: # Prefixes = 252254
(14 active peers)



- Number of RIB-in entries affected – when a prefix is affected by cert or ROA expiry/revocation – is approximately equal to the number of peers

Churn When an AS Cert is Revoked

Avg. # of Routes in Local RIB that an AS appears in (approx.)	30
Avg. # Routes invalidated in Local RIB (due to expiry/revocation of an AS cert) at each BGPSEC routers	30
# BGPSEC routers (long term estimated ballpark)	100000
# update messages in Global BGPSEC system	3000000

- Many waves of updates are generated throughout the BGPSEC enabled Internet
- In reality, the learning of AS cert revocation will be staggered in time at ASes; So many routers will pick alternate paths from other neighbor ASes who have not yet withdrawn routes containing the bad AS
- Path hunting goes on and many more than the estimated 3 million updates will propagate
- Be very proactive to prevent AS cert expiry/revocation whenever preventable

Churn When an Address Allocation Cert is Revoked

A large address space (/12) registration is expired or the cert is revoked	
# Actually routed prefixes affected (2048 suballocations that are /24 each)	2048
# ROAs that are invalidated	2048
# Prefix withdrawals per BGPSEC router	2048
# BGPSEC routers (long term)	100000
# update messages in Global BGPSEC system	204800000

- Many waves of updates (withdrawals, alternate paths) are generated throughout the BGPSEC enabled Internet
- The ripples intersect producing more ripples (of withdrawals, alternate paths)
- In reality, the learning of ROA withdrawals will be staggered in time at ASes; So many routers will pick alternate paths from other neighbor ASes who have not yet withdrawn routes containing the bad prefixes
- Be very proactive to prevent address allocation cert expiry/revocation whenever preventable

Churn When a New AS Cert or a New ROA is Created

- BGPSEC speakers need not/should not react to this except for:
  - Case A: The BGPSEC speaker which is originator of the prefix in ROA
  - Case B: BGP speaker(s) at the AS whose cert has been created
- In Case A, the originator AS of said prefix (and its suballocations) originates signed updates for the prefix (and suballocations); all others wait for the updates to propagate and act on them only when they arrive at their routers
- In Case B, said AS will propagate signed updates for prefixes it originates; all others wait for the updates to propagate and act on them only when they arrive at their routers
- Caveat: See next page

Churn When a New AS Cert or a New ROA is Created

- Caveat:

- Is this an accepted sequence of events?
- Is this what AS22 is expected to do under these circumstances?
- Feedback from 09/22/09 Meeting: Yes, this should be the correct mode of operations under these circumstances.

How to Minimize Churn in BGPSEC

- Excerpt from http://www.ietf.org/id/draft-ietf-sidr-arch-08.txt
  - If a CA certificate is reissued with the same public key, it should not be necessary to reissue (with an updated AIA URI) all certificates signed by the certificate being reissued. Therefore, a certification authority SHOULD use a persistent URI naming scheme for issued certificates. That is, reissued certificates should use the same publication point as previously issued certificates having the same subject and public key, and should overwrite such certificates.
- Was the above intended to serve the following purpose?
  - When a cert is altered to have a new expiry date, its URI remains invariant. The routers has previously received and stored the URI and public key for each resource cert. When a cert changes due to change of expiry date, no “refresh” messages need be sent to the routers.
  - There are benefits to this approach in reducing churn associated with expiry of certs in BGPSEC. Basically, RPKI server monitors ROA/cert renewals/expiries; Routers assumes ROAs/certs are fine unless notified otherwise by RPKI server

How to Minimize Churn in BGPSEC

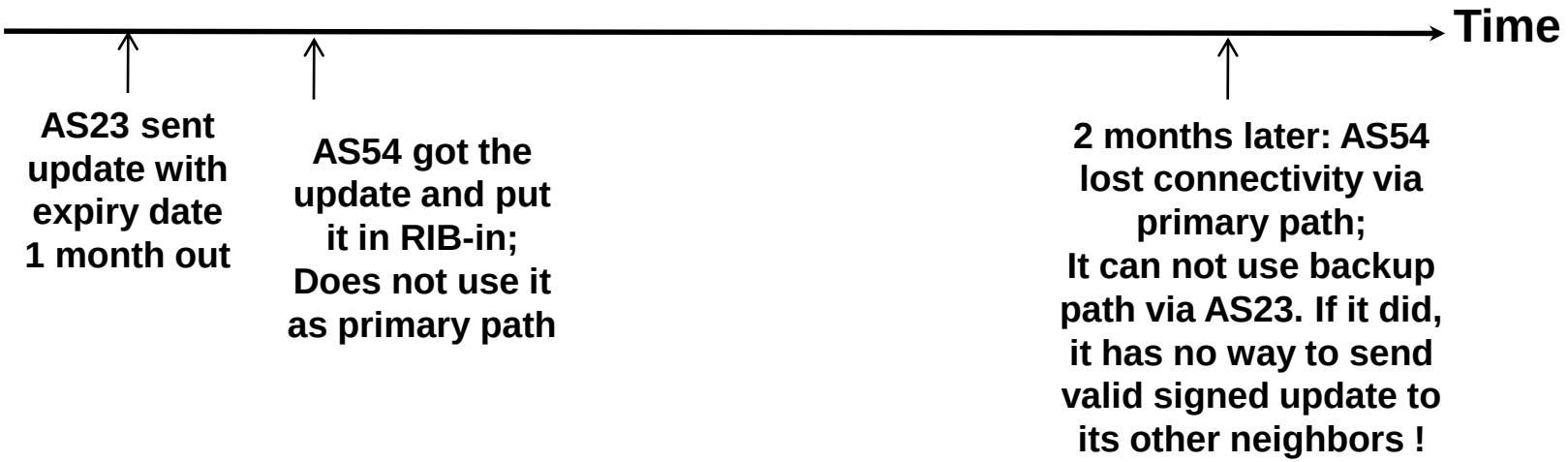
- RPKI Server performs validations of resource certs and ROAs (walking the cert chains as necessary)
- The information that is sent to the routers from RPKI-svr is:
  - For AS certs: {ASN, public key, and URI} triplet for each validated cert *
  - Based on ROAs validations, “effectively” a white list of {prefix, origin} pairs is also sent ++
  - Any of the above information can be retracted if necessary from the RPKI-svr by sending “withdrawal” messages ++

* Steve Kent’s suggestion on BGPSEC list
++ draft-ymbk-rpki-rtr-protocol-04

How to Minimize Churn in BGPSEC

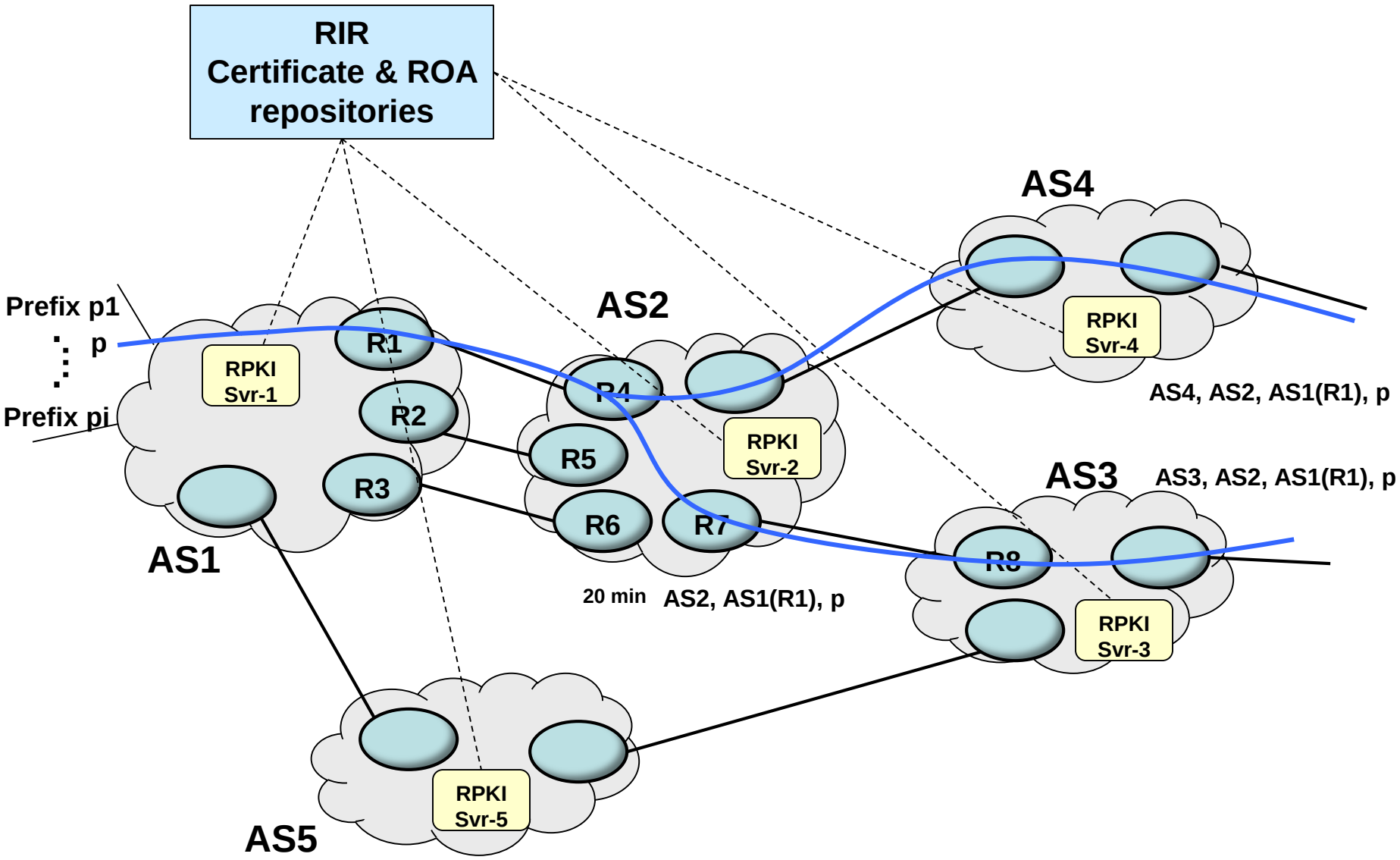
- AS cert's URI information is already included in the signer field of AS attestations in updates
- Routers never scan for expiry dates on certs, ROAs, etc; may be they do not even store certs, ROAs
- Signature validation at routers does not take into account the cert or ROA expiry date at all
- Expiry date in AS attestation in updates in independent of AS cert expiry – further clarification on its usage is needed
- Router need not even know if a cert had expired and it was reissued
- Router receives from the RPKI-svr a "withdrawal" of a previously validated cert or ROA only if the cert or ROA indeed expired and was not renewed or was otherwise revoked
- By doing the above, cert-renewal related BGPSEC churn can be reduced

Discussion on Expiry Date in Updates

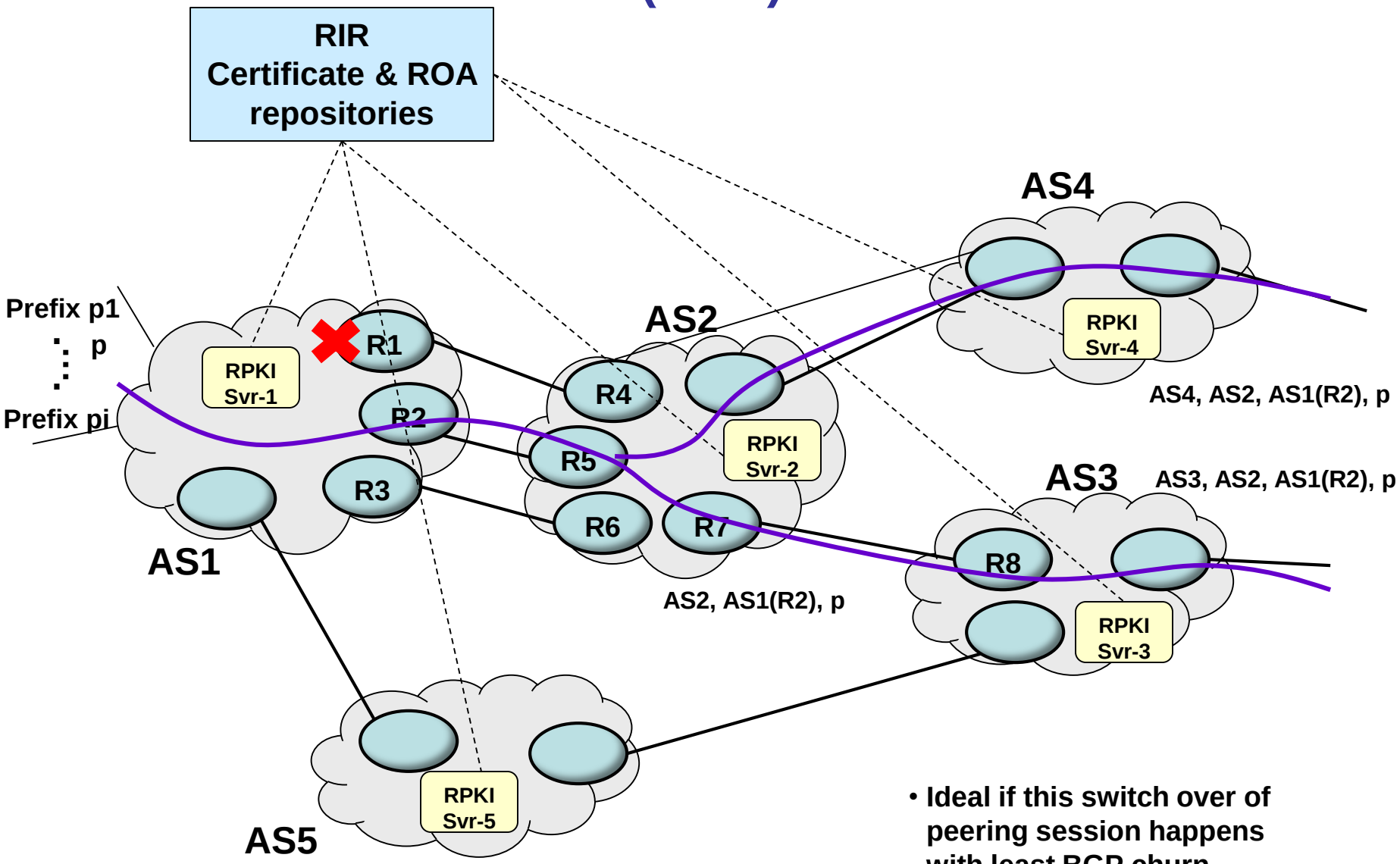


- How do want to use expiry date associated with signatures?
- Will expiry date be included in the signed information so that signature validation fails if expiry date in altered by an AS in the path?
- Will originating AS re-beacon updates just prior to expiry date?
- What are we achieving with signature expiry date in update? Is it adequately effective in preventing replay attacks?

Multiple Peerings, Revocation and BGP Churn



Multiple Peerings, Revocation and BGP Churn
(Ideal)



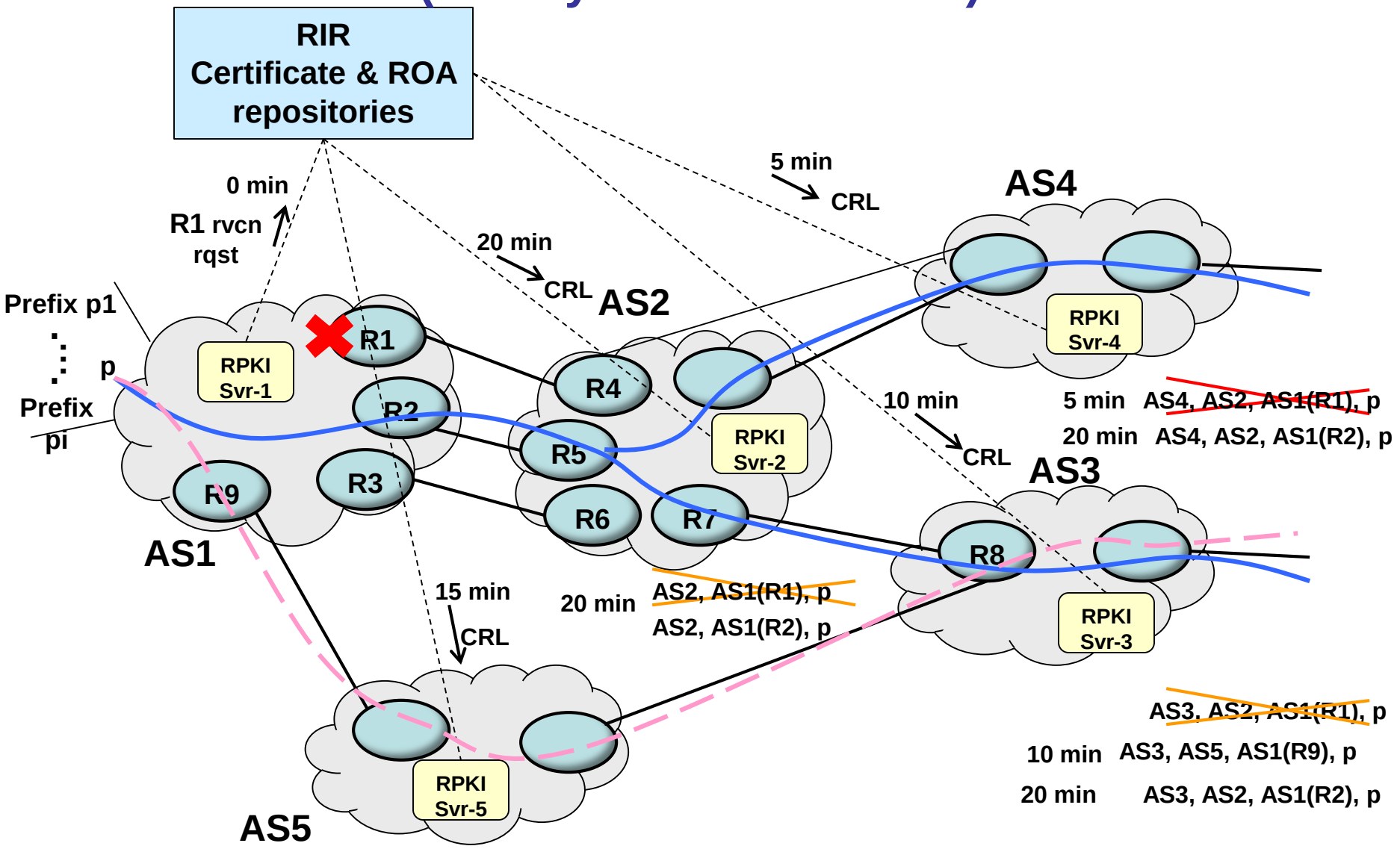
- **Ideal if this switch over of peering session happens with least BGP churn**

Multiple Peerings, Revocation and BGP Churn (Ideal)

- AS2 first to learn that the cert for R1 in AS1 is revoked; AS2 selects another peering session and propagates a new update
- AS path is unchanged; only the attestation of AS1 is changed from R1-cert based to R2-cert based
- Update with new valid attestation radiates outward from AS2 to upstream ASes
- All other AS-peers of AS1 also receive the CRL well ahead of ASes that are 2 or more hops away
- This keeps BGP churn to minimum that would be necessary

Multiple Peerings, Revocation and BGP Churn

(Reality – much messier)

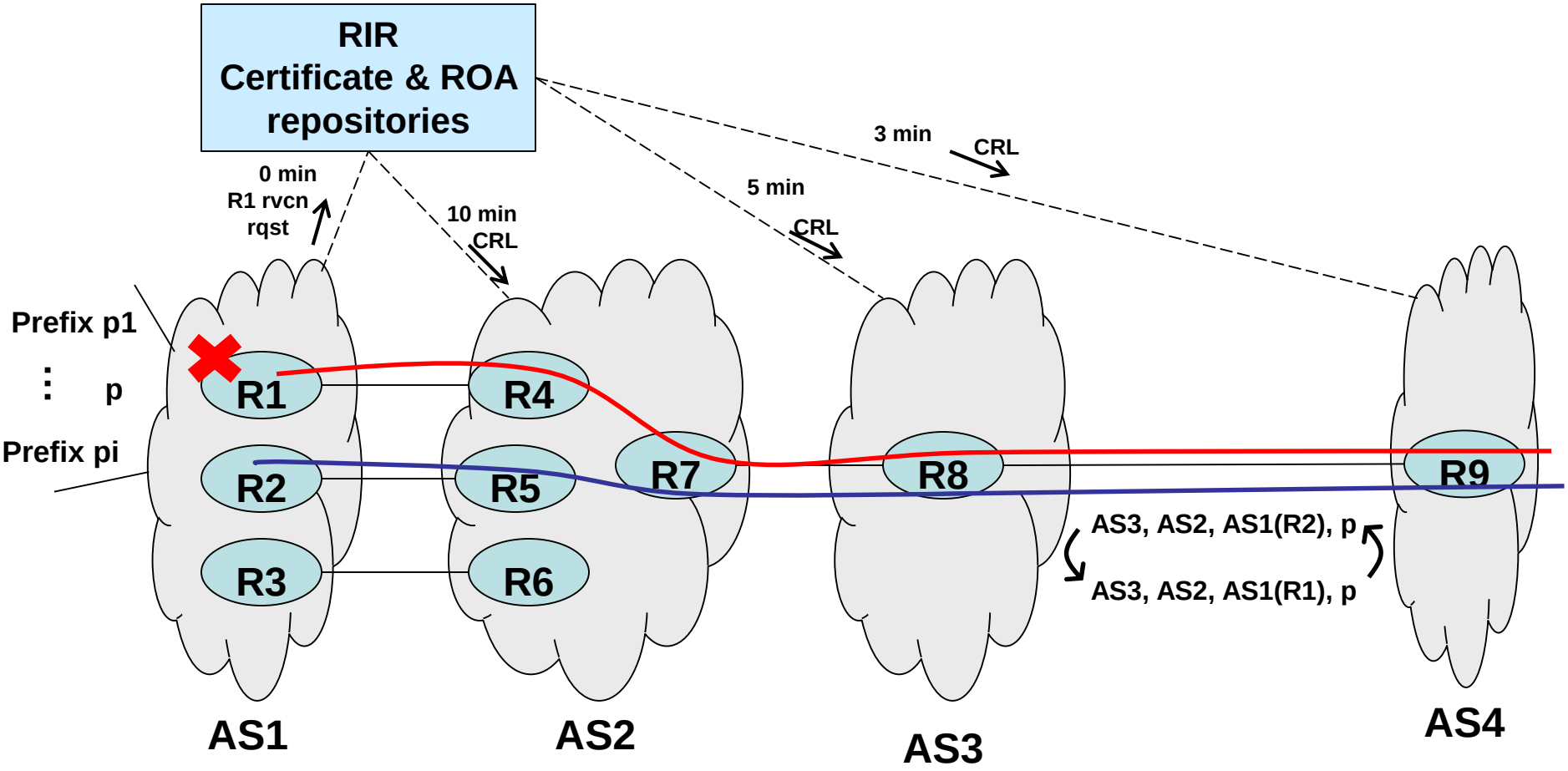


Note (Example): AS1(R2) means AS1 with signature signed with cert of R2

BGP Churn Due to Scheduling Latencies in CRL Propagation

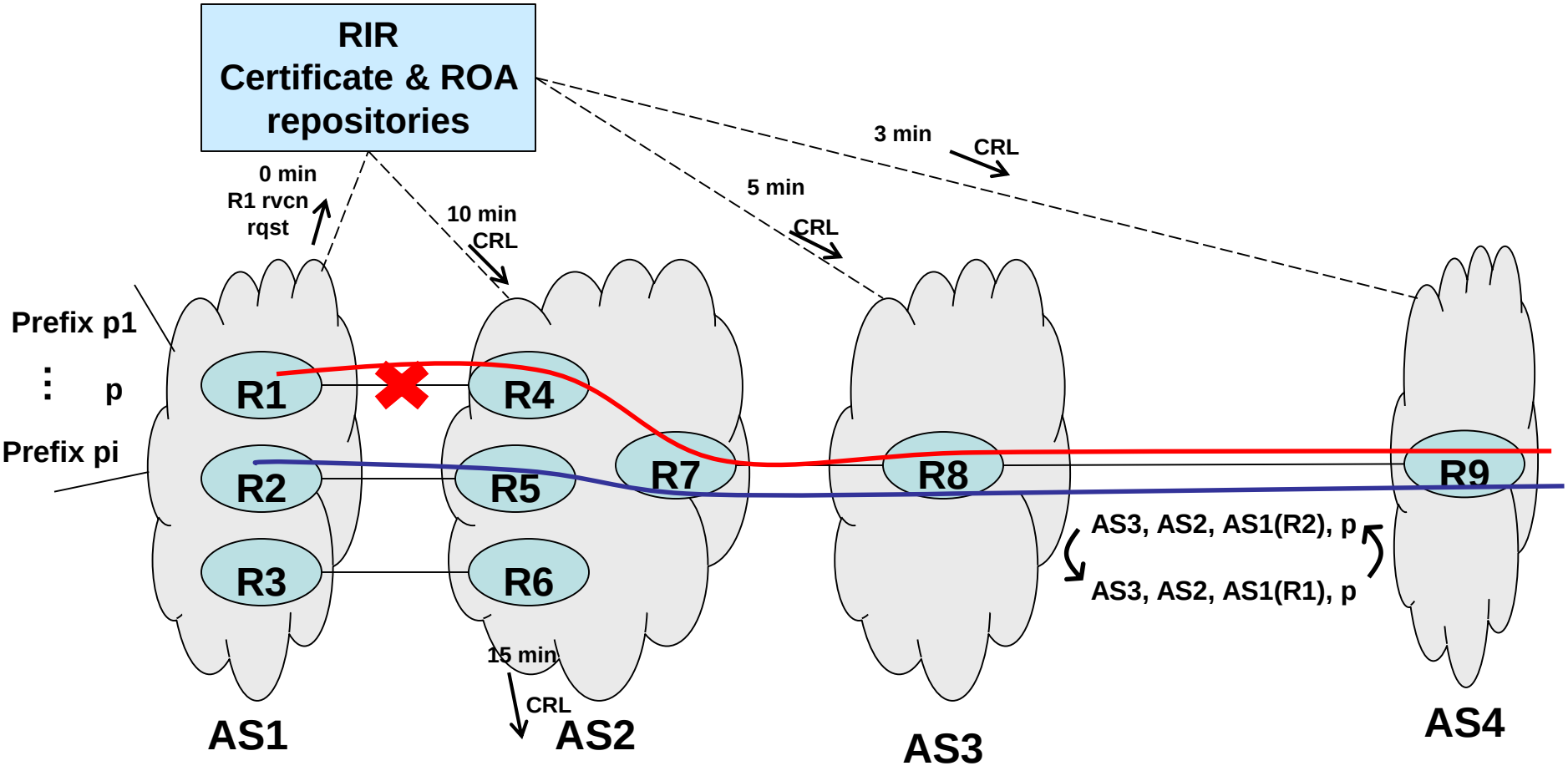
- Due to latencies involved in propagating revocation lists, in spite of multiple peerings between AS1 and AS2, breakdown in reachability becomes unavoidable
- AS4 withdraws prefixes propagated through AS1 (signed with R1's cert) for a while; AS4 artificially has no reachability to those prefixes until AS2 catches up (i.e., receives CRL)
- AS4 propagates many withdrawals unnecessarily
- AS3 also generates churn by picking alternate paths while in reality none of it should have been necessary; the problem was created because AS2 (where the multi-session BGP peering exists) learned about the CRL later than others
- Just one router's cert revocation can produce substantial BGP churn globally – because orderly dissemination of CRL is not feasible in reality
- Suggestion at September 22 meeting: Add another validation state – "Revoked" and do not jack up the RFD penalty under these circumstances

What types of DOS Attacks are Possible?



- R1's cert is revoked
- AS3 can be a malicious AS sending updates alternatively with R1-cert (invalid) and R2-cert (valid) for the period until the expiry time in the attestation allows it
- This can cause route flap damping for the prefixes p1, … , pi and render them unreachable from parts of the Internet

What types of DOS Attacks are Possible?

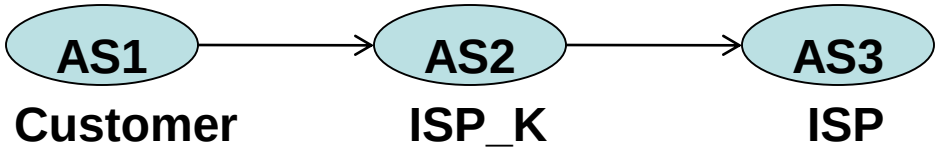


- In this case, the peering session between R1-R4 went down; AS2 switched to R2-R5 peering
- AS3 can be a malicious AS sending updates alternatively with R1-cert (valid) and R2-cert (valid) for a period of time
- This can cause route flap damping for the prefixes p1, … , pi and render them unreachable from parts of the Internet

Idea of Signing BGP Keep Alive or IPSEC

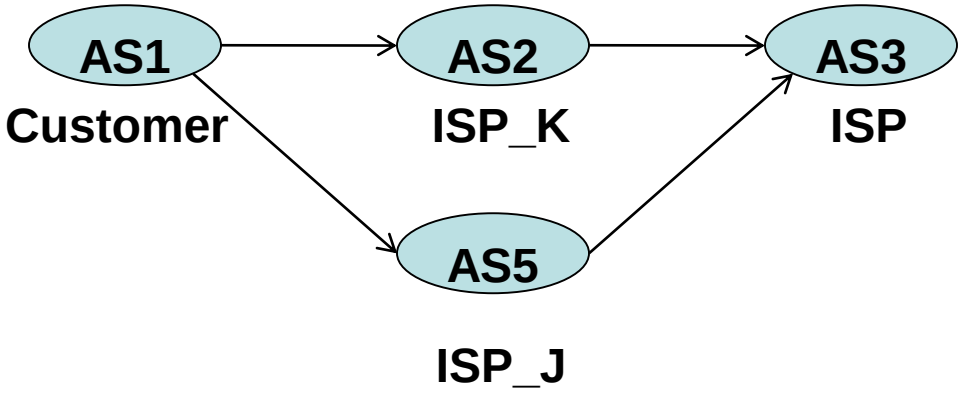
- BGP keep alive interval – configurable; typically 90s
- Peering AS signs its keep alive messages
- But if one router in an AS has its cert revoked, the peering router in the second AS does not detect invalid cert until it has received the CRL
- If CRL is based on polling which may have an interval of 30 min or so, then Keep Alive or IPSEC approach does not help
- If fast CRL indication/notification is provided to ASes, then the CRL can be possibly pulled (downloaded) quickly, and the bad peering session (due to rogue router) can be disabled without much delay
- So it seems signing BGP keep alive is not so helpful as compared to, say, **fast push-based CRL indication/notification**

Proxy Update Signing



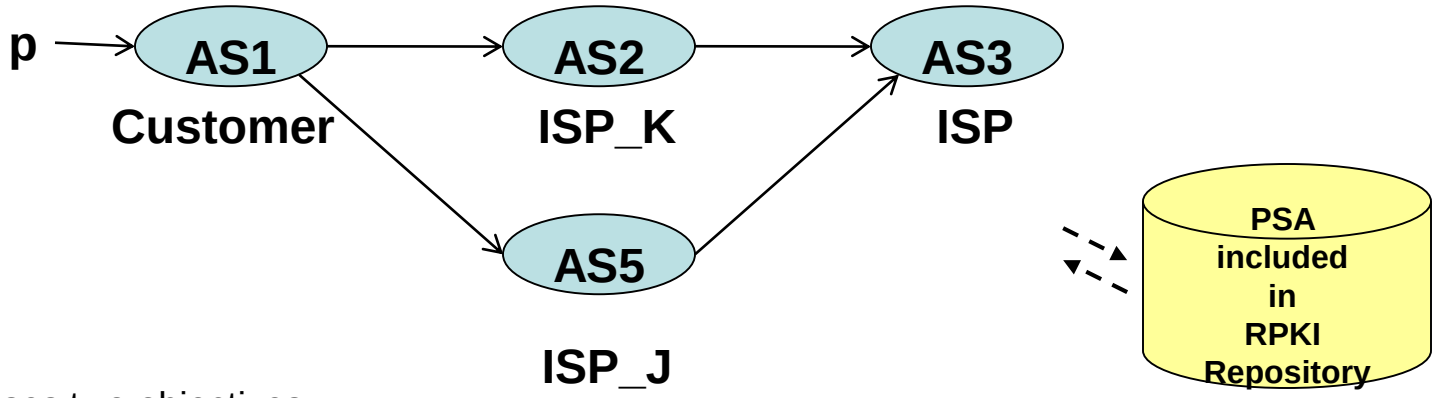
- **Solution A:** AS1 subscribes to a service offered by the ISP (name it ISP_K) that operates AS2. Under the service agreement, AS1 will have peering with AS2 and will share its private key with ISP_K (under an appropriate secure arrangement) so that AS2 can have the private key of AS1. AS1 does not do BGPSEC, but AS2 does and signs updates originated from AS1 and adds a signature on behalf of AS1.
- **Solution B:** ISP_K actually owns the ASNs of both AS1 and AS2 and hence holds certificates for both AS1 and AS2. ISP_K sublets (rents) to a customer the ASN of AS1, but continues to own and manage the cert for that ASN. The customer operates AS1, and AS1 peers with AS2. In this situation, again AS1 does not do BGPSEC, but AS2 does and signs updates originated from AS1 and adds a signature on behalf of AS1.
- Steve K's reply: "From a technical perspective I think this would work. It is not so clear if the RIRs would be happy with the arrangement, i.e., one ISP getting an ASN to sublet to another, in order to make the path validation work. But, since the rules for allocating ASNs seem to be pretty lax (which is why, in large part, we have to go to 4-byte ASNs now :-)) it may be OK anyway."

Proxy Update Signing
Sharing Private Key – A Logistics Problem



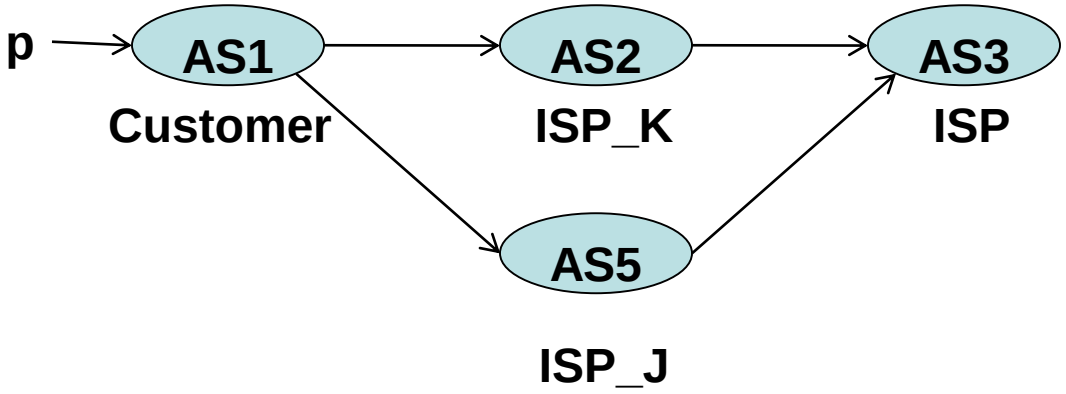
- Under separate service agreements, AS1 has peering with AS2 and AS5, and shares its private key with ISP_K and ISP_J (under an appropriate secure arrangement) so that AS2 and AS5 can have the private key of AS1. AS1 does not do BGPSEC, but AS2 and AS5 do, and each sign updates originated from AS1 and add a signature on behalf of AS1.
- One problem: If later AS1 decides to terminate its service with AS5 and does not want AS5 to know its private key anymore, then AS1 has to renew its certificate and share a new private key with AS2.

Proxy Signer Authorization (PSA)



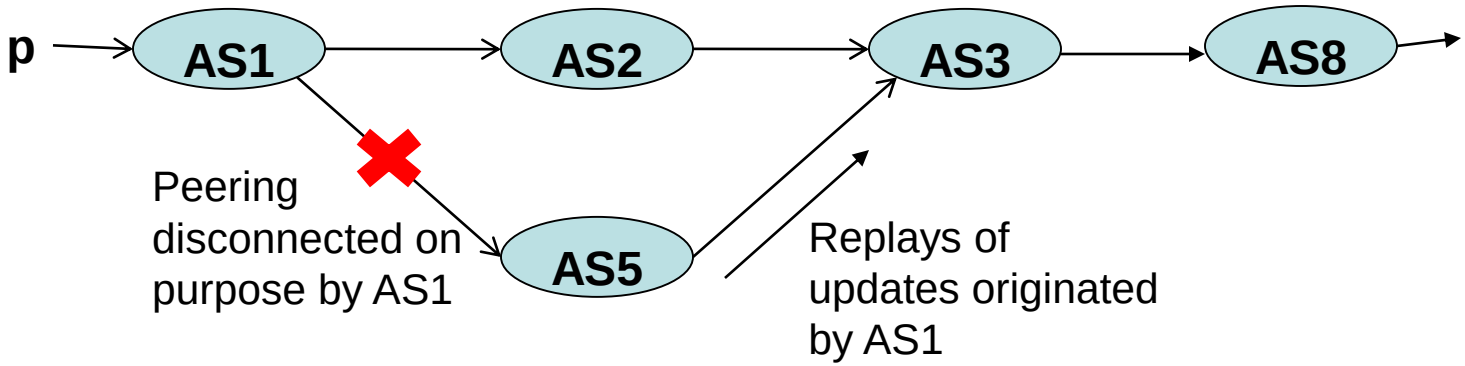
- PSA encompasses two objectives:
  - Proxy authorization for update signing
  - Prevention of Downgrade attacks – by declaring peering relations for BGPSEC peers
- Definition / Specification of PSA
  - PSA(AS m, AS n, Signer = AS n) means AS m “does” BGPSEC with AS n and AS n is proxy signer
  - PSA(AS m, AS n, Signer = AS m) means AS m does BGPSEC and AS m signs (not a proxy situation)
  - Example set for the network shown above: PSA(1, 2, Signer = 2); PSA (1, 5, Signer = 5); PSA(2, 3, Signer = 2); PSA(5, 3, Signer = 5)
- PSA as defined here encompasses the two objectives stated above
- Downside: Takes away the ability to temporarily suspend signing updates due to processor overload because it can be misconstrued as a downgrade attack!

Beaconing when Using Proxy Signer



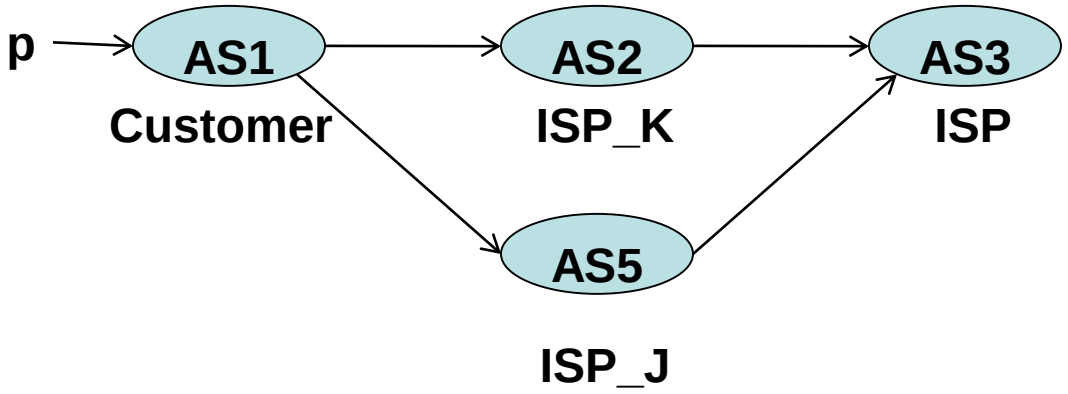
- Updates are re-beaconed when signatures expire
- Beaconing has to come from first signer
- In the above example, both AS2 and AS5 are instances of first signers
- So AS2 and AS5 will independently beacon the updates they have signed on behalf of AS1 in the event of signature expiry

Does PSA also Help Mitigate Replay Attacks?



- In this example, all ASes are BGPSEC speakers (no proxies)
- There exists (naturally) a PSA (1, 5, Signer = 1) in the PSA database
- At some point of time, AS1 stops peering with AS5
- AS1 also revokes its PSA with AS5 (that is, PSA (1, 5, Signer = 1) is revoked)
- AS5 attempts replay attacks using previously seen updates originated from AS1
- These replay attacks would NOT be successful once the CRL conveying revocation of PSA (1, 5, Signer = 1) has reached the upstream ASes
- Would PSA revocation work adequately for prevention of replay attacks? Will the need for expiry date for signatures in updates go away?

More Appropriate Name?



- PSA encompasses multiple objectives:
  - Proxy authorization for update signing
  - Prevention of Downgrade attacks – by declaring peering relations for BGPSEC peers
  - Helps mitigate replay attacks
- PSA is not just about proxy signer anymore; it more generally declares the BGPSEC capability between peers
- Call it GSA instead – Generalized Signer Authorization?

Open Issues in BGPSEC Architecture

- **Sequential Aggregate Signatures: One rolling signature over multiple AS hops**
  - Saving in bandwidth, PDU size, and may be in RIB size but computation cost will be higher as compared to one signature per-hop
- **Solutions that allow validation to be outsourced:**
  - Validating off-router box or route reflector
  - Off-router box (one per AS) doing full validation (Dave)
  - At ingress one needs the RIB-in validated, not the RIB-out (DougMont)
  - Do you feed “bleeding” control plane (BGPSEC) data into the box?
  - Off-router box is probably unlike the traditional route reflector; it sits at net edge and validates all incoming BGPSEC updates
  - Does it mean each router passes each BGPSEC update it receives to the off-router box and receives a validation result before it stores the route in the RIB-in?
  - Failure modes: Dependent on how much of the control plane is outsourced; Outboard element (or, equivalently communication to it) fails; Centralized vs. distributed philosophy (John)

Open Issues in BGPSEC Architecture (Contd.)

- so-BGP class of solutions (John's email):
  - Use of a secure map of AS relationships to "validate" updates
  - The rough consensus was not to pursue such solutions because they can't provide strong guarantees about the validity of updates
  - You can only say that an update is "plausible", not "definitely" valid

Open Issues in BGPSEC Architecture (Contd.)

- Reaction to SIG expiry vs. Reaction to Cert Expiry (DougMont)
  - React differently to SIG lifetime expirations than CERT expirations; when/where do you react and how do you react?
  - When/where - boils down to do you only check lifetimes on receipt of an update? Or do you go back to your RIB-in (either periodically, or driven by timer events) and expire announcements that previously validated?
  - The How question boils down to if the decision process treats "expired“ different than "invalid" or "no-data". May be react to expired PATH SIGs in more draconian ways because we can define the required refresh mechanisms in the protocol, and an expired PATH SIG would represent a BGPSEC failure of some sort.
  - May be do not react in draconian ways to expired CERTs, because that creates a tighter fate share between BGP and RPKI maintenance mechanisms than we would be comfortable with .... at least until we can prove that this global RPKI has much better maintenance properties than most.