IP Security Maintenance and Extensions (IPsecME) WG

IETF 116, Wednesday, March 29th, 2023

Chairs: Tero Kivinen
Yoav Nir

Responsible AD: Roman Danyliw

Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

• By participating in the IETF, you agree to follow IETF processes and policies.

• If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.

• As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.

• Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.

• As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:
•BCP 9 (Internet Standards Process)
•BCP 25 (Working Group processes)
•BCP 25 (Anti-Harassment Procedures)
•BCP 54 (Code of Conduct)
•BCP 78 (Copyright)
•BCP 79 (Patents, Participation)
•https://www.ietf.org/privacy-policy/ (Privacy Policy)

Administrative Tasks

We need volunteers to be:

- Two note takers

MeetEcho: https://meetings.conf.meetecho.com/ietf116/?group=ipsecme&short=&item=1

Notes: https://notes.ietf.org/notes-ietf-116-ipsecme

Agenda

- Note Well, technical difficulties and agenda bashing – Chairs (5 min) (15:30-15:35)
- Document Status – Chairs (5 min) (15:35-15:50)
- Presentations
  - Issues SA TS Payloads opt draft – Paul Wouters (10 min) (15:50-16:00)
  - Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov (10 min) (16:00-16:10)
  - Extended IKEv2 Payload Format – Valery Smyslov (20 min) (16:10-16:30)
  - Anti replay subspaces – Mohsin Shaikh (10 min) (16:30-16:40)
  - IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault (10 min) (16:40-16:50)
  - Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault (10 min) (16:50-17:00)
- If time permits
  - Inter-domain source address validation using RPKI and IPsec

WG Status Report

- Published as RFCs
  - TCP Encapsulation of Internet Key Exchange Protocol (IKE) and IPsec Packets was published as RFC9329
  - Aggregation and Fragmentation Mode for Encapsulating Security Payload (ESP) and Its Use for IP Traffic Flow Security (IP-TFS) was published as RFC9347
  - A YANG Data Model for IP Traffic Flow Security was published as RFC9348
  - Definitions of Managed Objects for IP Traffic Flow Security was published as RFC9349
- RFF Editor queue:
  - draft-ietf-ipsecme-ikev1-algo-to-historic
  - draft-ietf-ipsecme-ikev2-multiple-ke
- Publication requested:
  - draft-ietf-ipsecme-labeled-ipsec IETF Last Call

WG Status Report

- Waiting for write-up / AD Followup:
  - draft-ietf-ipsecme-add-ike
- Working Group Last Call:
  - draft-ietf-ipsecme-auth-announce
  - draft-ietf-ipsecme-g-ikev2
- Work in progress:
  - draft-ietf-ipsecme-ikev2-sa-ts-payloads-opt
  - draft-ietf-ipsecme-multi-sa-performance

Presentations

- Issues SA TS Payloads opt draft – Paul Wouters
- Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov
- Extended IKEv2 Payload Format – Valery Smyslov
- Anti replay subspaces – Mohsin Shaikh
- IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault
- Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault
- Inter-domain source address validation using RPKI and IPsec

Presentations

- **Issues SA TS Payloads opt draft – Paul Wouters**
- Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov
- Extended IKEv2 Payload Format – Valery Smyslov
- Anti replay subspaces – Mohsin Shaikh
- IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault
- Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault
- Inter-domain source address validation using RPKI and IPsec

OPTIMIZED_REKEY ISSUES FOUND

DRAFT-IETF-IPSECME-IKEV2-SA-TS-PAYLOADS-OPT

IPsecME, IETF 116

Yokohama, March 2023

Tobias Brunner, Paul Wouters

OPTIMIZED_REKEY and SPI

- REKEY: IKE and Child SA contains new SPI in the SA payload
- OPTIMIZED_REKEY: Both send new SPI in N(OPTIMIZED_REKEY)
- Difference between IKE and Child rekey a little harder to detect via size difference of SPI field

- Proposal 1: Leave as is and use SPI size (and improve text)
- Proposal 2: Use a new notify payload (OPT_REKEY_IKE_SPI) ?

OPTIMIZED_REKEY_SUPPORTED

- Sent in IKE_AUTH
- What if there are more than one IKE_AUTH exchange
  - Proposal:
    - Initiator sends it in first IKE_AUTH
    - Responder send it in the last IKE_AUTH exchange (i.e. sent it where normally the TS payloads go)

Rekeying initial Child SA

- The initial Child SA uses the KE from the IKE SA
- If Child SA negotiates PFS, it uses the IKE SA group

- OPTIMIZED_REKEY for Child SA that used PFS should:
  - Proposal 1: Use same KE as IKE SA
  - Proposal 2: Use a regular rekey before using an optimized rekey

USE_TRANSPORT, ESP_TFC_PADDING_NOT_SUPPORTED, NON_FIRST_FRAGMENTS_ALSO, etc

- Normally sent in CREATE_CHILD_SA for rekeys.
- We don’t want a changed outcome on these notifies.

- Proposal 1: omit Notifies means “keep same”, error if not.
- Proposal 2: sent Notifies, error if not same.

IPCOMP_SUPPORTED

- IPCOMP_SUPPORTED payload contains compression algorithm and the CPI.
- Algorithm could be omitted but CPI is needed.

- Proposal 1: Send IPCOMP_SUPPORTED with new CPI.
  - Reject proposals that change IPcomp algorithm
- Proposal 2: Omit IPCOMP_SUPPORTED and send a 2nd OPTIMIZED_REKEY Notify with protocol IPcomp (108) and SPI length 2 and the new CPI as SPI value.

ERROR HANDLING

- If KE payload is for a different group, or “Child SA notify change” (see previous slides), what error message to send ?
  - INVALID_KE
  - INVALIX_SYNTAX
  - NO_PROPOSAL_CHOSEN
  - A new: INVALID_REKEY_CHANGE

Presentations

- Issues SA TS Payloads opt draft – Paul Wouters
- **Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov**
- Extended IKEv2 Payload Format – Valery Smyslov
- Anti replay subspaces – Mohsin Shaikh
- IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault
- Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault
- Inter-domain source address validation using RPKI and IPsec

Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security

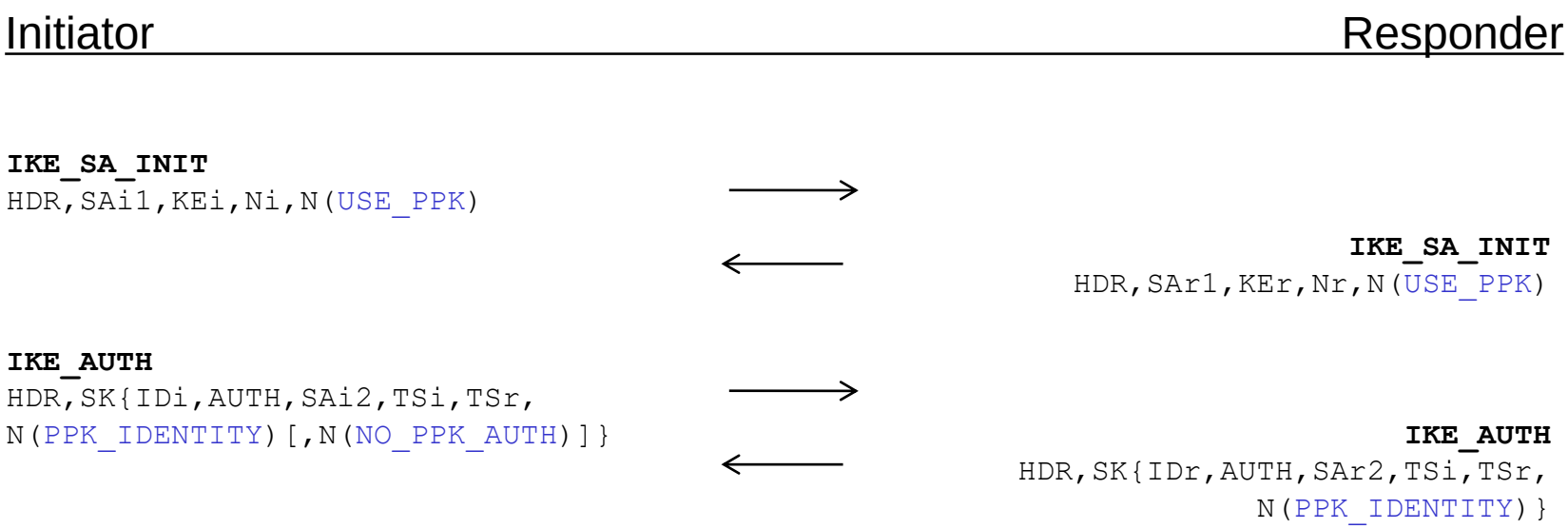
draft-smyslov-ipsecme-ikev2-qr-alt

Valery Smyslov
svan@elvis.ru

IETF 116

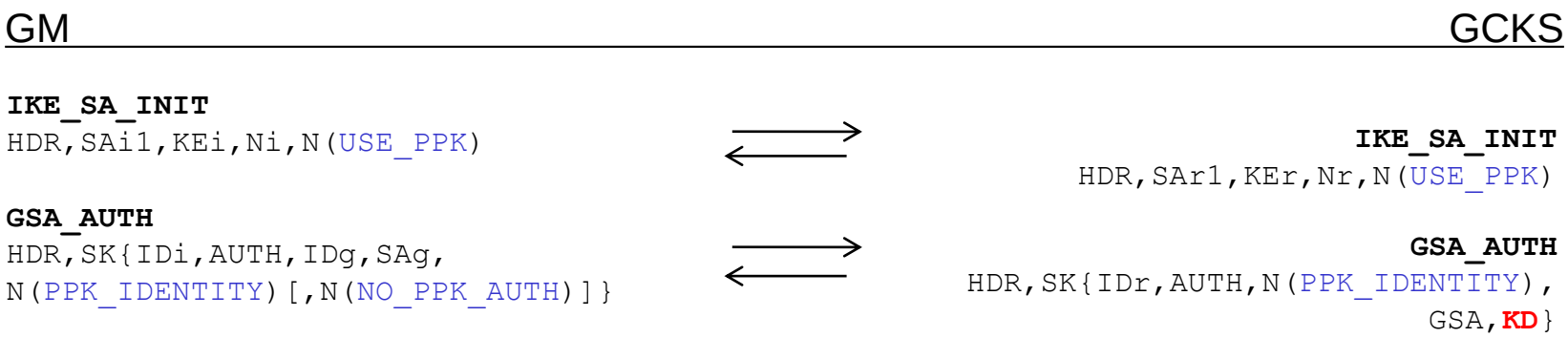
PPK for IKEv2

Defined in RFC 8784:



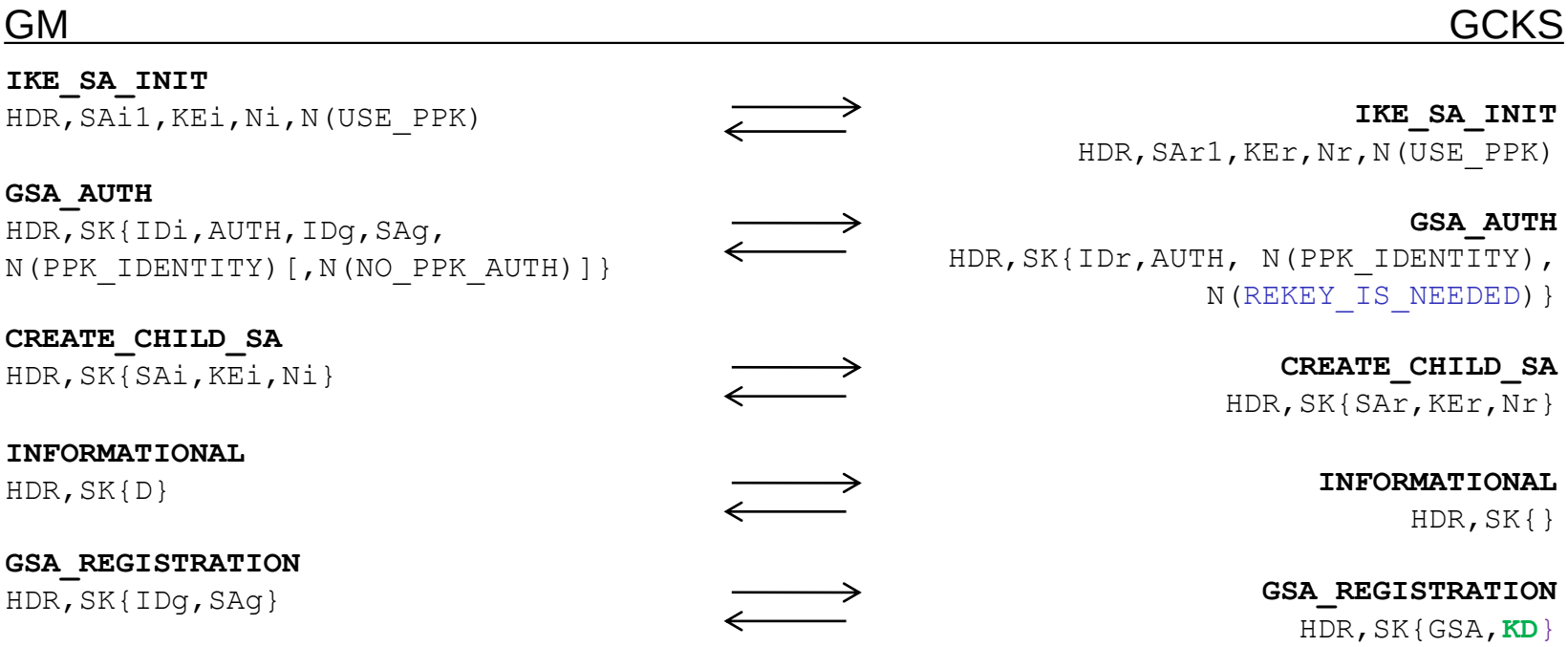
The Problem

- Initial IKE SA is not protected with PPK (WG decision)
  - it was assumed that no sensitive information was transferred over initial IKE SA, and one could immediately rekey it to get protection
- G-IKEv2 (draft-ietf-ipsecme-g-ikev2) uses initial IKE SA to immediately transfer session keys from Group Controller/Key Server (GCKS) to Group Member (GM)
  - these keys **are not protected** with PPK



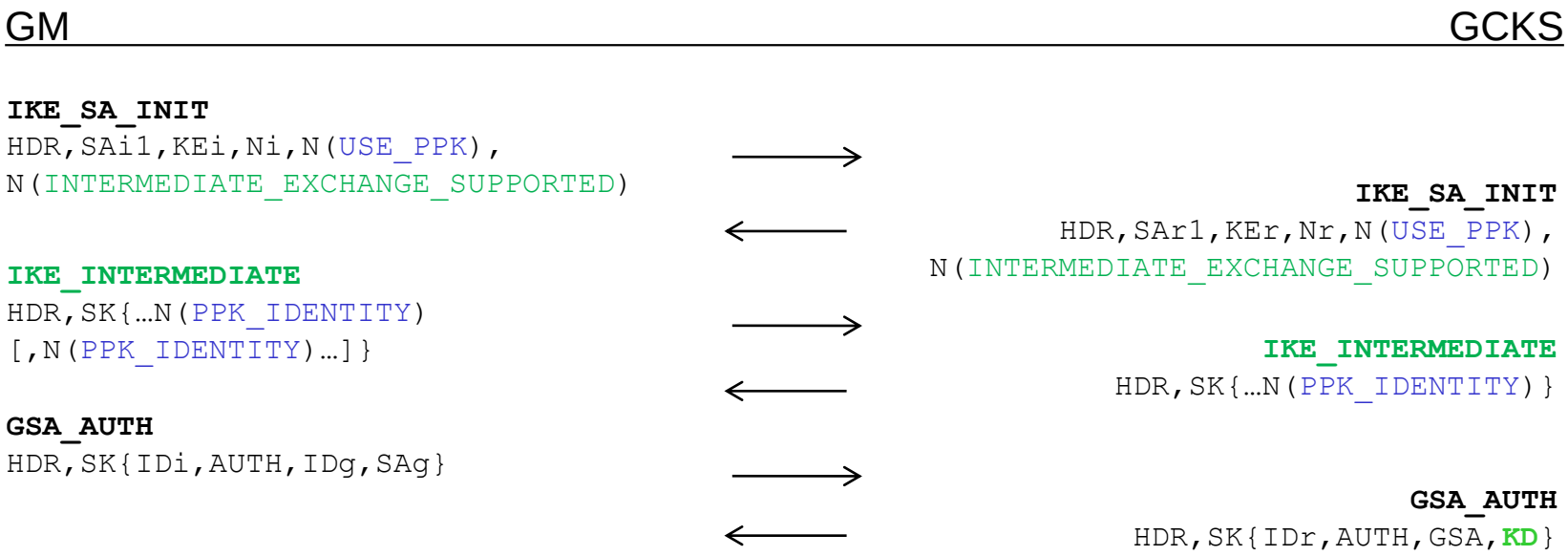
Current Use of PPK with G-IKEv2

Currently G-IKEv2 draft suggests the following sequence of exchanges to get the protection with PPK:



Alternative Approach

Proposed in draft-smyslov-ipsecme-ikev2-qr-alt:



Fallback to RFC 8784

- If the responder doesn't support this extension, then it doesn't respond with any PPK_IDENTITY in IKE_INTERMEDIATE
  - the initiator MAY fallback to RFC 8784 in this case
  - the same situation happens if the responder isn't configured with any of the proposed PPK_IDs
    - no need to fallback to RFC 8784 in this case, but allowed in the draft for simplicity
- It is possible to modify draft to distinguish between these two cases and disallow fallback if extension is supported, but no PPK found

Double PPK

- Do we need to support using both RFC 8784 and this draft's approaches for a single SA?
  - Currently is not supported in the draft
  - It seems that this is too complex with no benefits
    - Should be explicitly prohibited in the draft?

Session Keys Calculation

- RFC 8784:

```
SKEYSEED = prf(Ni | Nr, g^ir)
{SK_d` | SK_ai | SK_ar | SK_ei | SK_er | SK_pi` | SK_pr`} =
                                    prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)
SK_d = prf+ (PPK, SK_d')
SK_pi = prf+ (PPK, SK_pi')
SK_pr = prf+ (PPK, SK_pr')
```

- This proposal

```
SKEYSEED' = prf+ (PPK, SK_d)
{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr} =
                                    prf+ (SKEYSEED', Ni | Nr | SPIi | SPIr )
```

Mismatched PPK

- If PPKs with the same PPK_ID are different, then we run into the problem that the responder cannot decrypt and authenticate IKE_AUTH messages and will drop them
  - MUST be fixed in the next version of the draft
    - need to have key confirmation payload in the IKE_INTERMRDIATE exchange, perhaps prf(PPK, Ni | Nr | SPIi | SPIr)
    - Who should send it – initiator or responder? Seems like more appropriate for initiator

Comparison

- For G-IKEv2:
  - 3 exchanges instead of 5
  - 1 DH shared key computation instead of 2
  - 1 computation of AUTH in case of optional PPK instead of 2
  - initiator can propose several PPK_IDs
- Can also be used in IKEv2:
  - 3 exchanges instead of 2
    - but PPK_ID can be piggybacked if IKE_INTERMEDIATE is also used for other purposes
  - 1 computation of AUTH instead of 2 if PPK is optional
  - initiator can propose several PPK_IDs

Coexistence

- The proposed approach is **not intended to replace** the existing one, both can co-exist:
  - for G-IKEv2 the proposed approach can be a primary one (or the only one?)
  - for IKEv2 the proposed approach can be an alternative one (e.g. if IKE identities need to be protected)

Implementations

- At least 2 implementations of -06 draft exists:
  - ELVIS-PLUS
  - libreswan
- Successfully interoperated during hackathon

Thanks

- Comments? Questions?
- More details in the draft
- WG adoption?

Presentations

- Issues SA TS Payloads opt draft – Paul Wouters
- Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov
- **Extended IKEv2 Payload Format – Valery Smyslov**
- Anti replay subspaces – Mohsin Shaikh
- IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault
- Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault
- Inter-domain source address validation using RPKI and IPsec

Extended IKEv2 Payload Format

draft-smyslov-ipsecme-ikev2-extended-pld

Valery Smyslov
svan@elvis.ru

IETF 116

Problems with Existing Format

- Payload Length field occupies 2 bytes, so payload size is limited to 64 Kbytes
  - might not be enough for some PQ algorithms
  - no problem with Message size, which is limited to 4 Gbytes
- Many payloads contain substantial redundancy
  - Payload Length field occupies 2 bytes, while most payloads are shorter
  - most parameters occupy 2 bytes, while less than 256 values are defined
  - zero-filled `RESERVED` fields

Existing Proposals

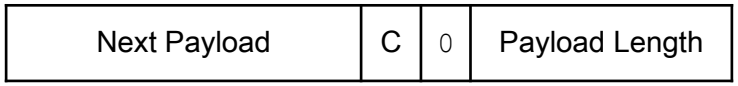
- A Larger Internet Key Exchange version 2 (IKEv2) Payload
  draft-nir-ipsecme-big-payload
- Beyond 64KB Limit of IKEv2 Payloads
  draft-tjhai-ikev2-beyond-64k-limit
- Compact Format of IKEv2 Payloads (expired)
  draft-smyslov-ipsecme-ikev2-compact
  (expired)

Extended Payload Format Overview

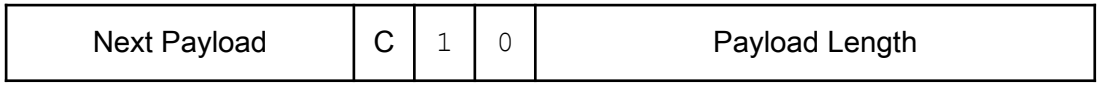
- Three formats for new Generic Payload Header
  - for small payloads (up to 64 bytes)
  - for medium size payloads (up to 8 Kbytes)
  - for large payloads (up to 512 Mbytes)
- No `RESERVED` fields
- Revise some existing payloads headers to reduce their size
  - remove unnecessary fields
- Special Format for some payloads (SA, some status notifies)

Extended Generic Payload Header Format

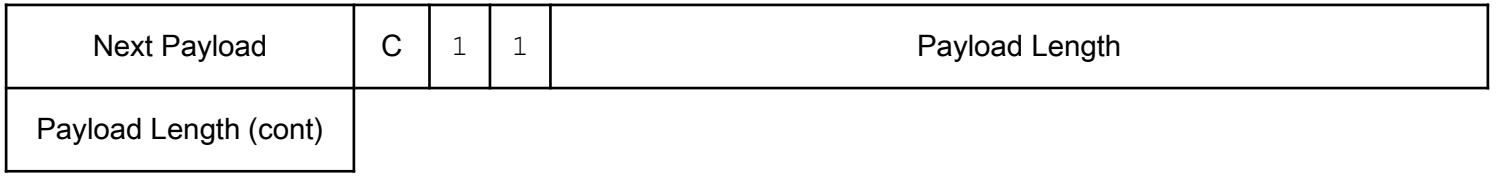
1. Small payloads (2 bytes, 6 bits for Payload Length)



2. Medium size payloads (3 bytes, 13 bits for Payload Length)



3. Large payloads (5 bytes, 29 bits for Payload Length)



Revise some Payload Headers

The following payload headers are revised:

- Key Exchange Payload
  - no `RESERVED` field (2 bytes)
- Identification, Authentication, Configuration Payloads
  - no `RESERVED` field (3 bytes)
- Traffic Selector Payload
  - no `RESERVED` field (3 bytes)
  - no `Number of TSs` field (1 byte)
- Traffic Selector
  - no `Selector Length` field (2 bytes)

Special Format for some Payloads

Special format for:

- SA Payload
  - SA Payload grows quickly as more and more new transforms are defined and offered by initiators
- Notify Payload with some Status Type Notification containing no data
  - Exchange of such payloads is a common way to negotiate support for various protocol extensions, so initial IKEv2 messages grow up as more and more extensions are defined

Both payloads contain a lot of redundancy and can be effectively compacted.

SA Payload

- No `RESERVED` fields
- No generic header in `Proposal` substructure
- Encode `Transform` substructure as variable-length structures

Transform Encoding

1-byte: for `Encryption`, `Key Exchange`, `PRF`, `ESN` Transform Types for limited number of Transform IDs

1-byte: for some future Transform Types (e.g. for G-IKEv2) and limited number of Transform IDs

2-bytes: for `Additional Key Exchange` Transform Types and for other Transform Types with Transform IDs that don't fit into 1-byte encoding

3-bytes: for Transform IDs that don't fit into 1-byte and 2-bytes encodings

5-512 bytes: for remaining Transform IDs or in case there are Attributes (other than `Key Length`)

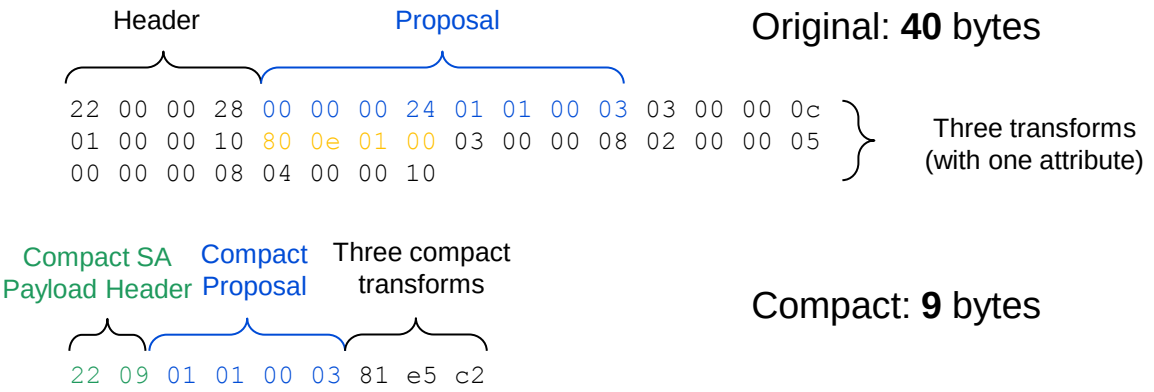
Transform Encoding Summary

Name	Format	Length	Transform Types	Transform IDs
Short	0**ttt**vvvv	1	13-20	0-15
Encryption (128)	100vvvvv	1	1	11-42
Encryption (256)	101vvvvv	1	1	11-42
KE	110vvvvv	1	4	0, 14-44
PRF	1110vvvv	1	2	2-15
ESN	111110vv	1	5	0-3
Long 1	11110**ttt tt**vvvvvv	2	1-31	0-63
Long 2	1111110**t tttt**vvvv vvvvvvvv	3	1-31	0-4095
Full	1111111**t ttttttt***l lllllll* vvvvvvvv vvvvvvvv	5 up to 512 (in case of attributes)	any	any

Example of Compact SA Payload

SA Payload with one Proposal and three Transforms:

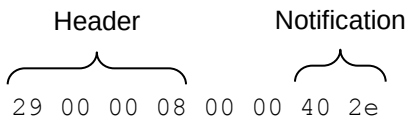
- `ENCR_AES_CCM_16` (256 bits key)
- `PRF_HMAC_SHA2_256`
- `4096-bit MODP Group`



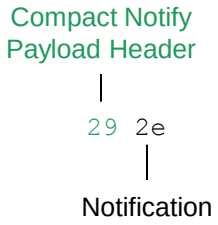
Notify Payload

Outline: encode notification in one octet (limited to first 256 status notifications) and omit all other fields from Notify Payload

Example: Notify Payload with `IKEV2_FRAGMENTATION_SUPPORTED` notification.



Original: **8** bytes



Compact: **2** bytes

Negotiation

If new format is used from the very beginning then the following options exist

- New status notify `EXTENDED_PAYLOAD_FORMAT`
  - extended format cannot be used in `IKE_SA_INIT`
  - suitable if only large payloads are needed
- New initial exchange `X_IKE_SA_INIT`
  - functionally equivalent to `IKE_SA_INIT`, but may contain payloads in extended (compact) format
  - old responders would return `INVALID_SYNTAX` notify

Transport Issues

Transport issues for transferring large payloads (> 64 Kbytes) are out of scope. Possible solutions:

- IKE over TCP combined with IKE fragmentation (to solve limitation on 64 Kbytes on a single IKE message over TCP)
- Mixed Mode (defined in draft-tjhai-ikev2-beyond-64k-limit: IKE over TCP + IKE fragmentation combined with plain ESP or ESP over UDP) can be used to avoid ESP performance degradation when used with TCP encapsulation

Discussion

- Get rid of `SPI Size` in Proposal substructure, Delete and Notify payloads (can be deducted from `Protocol ID`)?
- Get rid of `Proposal Num` in Proposal substructure?
- New Payload types or reuse existing types?
  - 9+ new payload formats – too many?
- Certificates consume a lot of space, can be compressed (out of scope)
  - RFC 8879 is an example of certificate compression

Thank you!

- Comments?
- Questions?
- Any interest in this work?

Presentations

- Issues SA TS Payloads opt draft – Paul Wouters
- Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov
- Extended IKEv2 Payload Format – Valery Smyslov
- **Anti replay subspaces – Mohsin Shaikh**
- IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault
- Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault
- Inter-domain source address validation using RPKI and IPsec

Anti-replay sequence number subspaces

draft-ponchon-ipsecme-anti-replay-subspaces

Mohsin Shaikh (presenter), Paul Ponchon, Pierre Pfister, Guillaume Solignac
IETF 116 @ Yokohama

Quick Recap

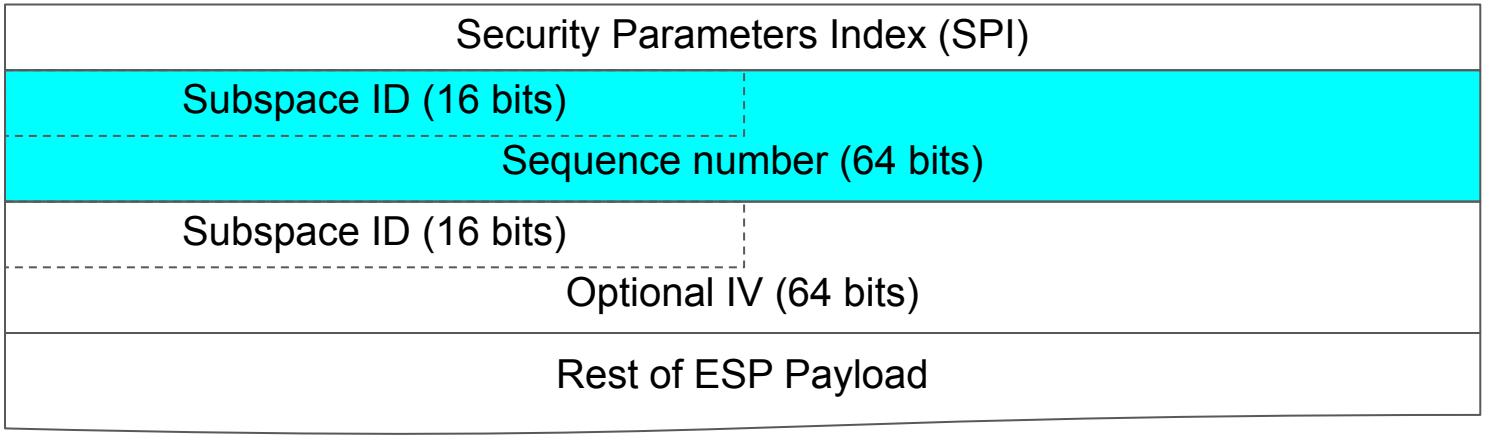
- Proposal to use multiple sequence number subspaces as an alternative to creating multiple child SAs for multi-core performance
- Additionally support QoS and multi-path

Advantages of multiple sequence subspaces

- Some users require anti-replay and PFS for compliance
- No. of IKEv2 messages exchanged:
  Multiple sequence subspaces : 4
  Multiple child SAs : 2 + (2 x no. of child SAs)
  IKEv2 touts fewer message exchanges as an advantage over IKEv1 [RFC7296 - Appendix A. Summary of Changes from IKEv1]
- Avoids repeating history. IKEv1 protocol allowed a single pair of selectors per CHILD_SA, while IKEv2 improved that by allowing multiple traffic selectors to be negotiated for one child SA. We are repeating the same IKEv1-like behaviour with multiple child SAs per core. [RFC7296 - Appendix A. Summary of Changes from IKEv1]
- Adding too many SAs may slow down per-packet SAD lookup [draft-ietf-ipsecme-multi-sa-performance-00]

Updates in draft-01 since IETF 115

- Increased the sequence number field in ESP header to 64-bits, the subspace ID to 16 bits (from 8 bits in previous draft) and reserve the top 16 bits to store subspace ID.



Updates in draft-01 since IETF 115 (contd.)

- IKE negotiation contains a new “Anti-replay subspaces” transform to negotiate the number of subspaces required
- IPR disclosure from SSH Communications Security

Test Data

- Memory usage goes up with multiple child SAs (tested between 2 docker containers using strongSwan with 1 IKE SA).

ESP proposal	aes256gcm8	aes128ctr-sha2_256
0 tunnel	9,040 kB	8,908 kB
1000 tunnels	17,996 kB	19,228 kB
10,000 tunnels	90,360 kB	98,576 kB
100,000 tunnels	824,100 kB	891,568 kB

- Assuming a 64 packet window, the amount of increase in memory per SA is (8+8 bytes) X no. of sequence subspaces.

Should the working group work on this and adopt ?

Q&A

Presentations

- Issues SA TS Payloads opt draft – Paul Wouters
- Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov
- Extended IKEv2 Payload Format – Valery Smyslov
- Anti replay subspaces – Mohsin Shaikh
- **IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault**
- Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault
- Inter-domain source address validation using RPKI and IPsec

IKEv2 Link Maximum Atomic Packet and Packet Too Big Notification Extension

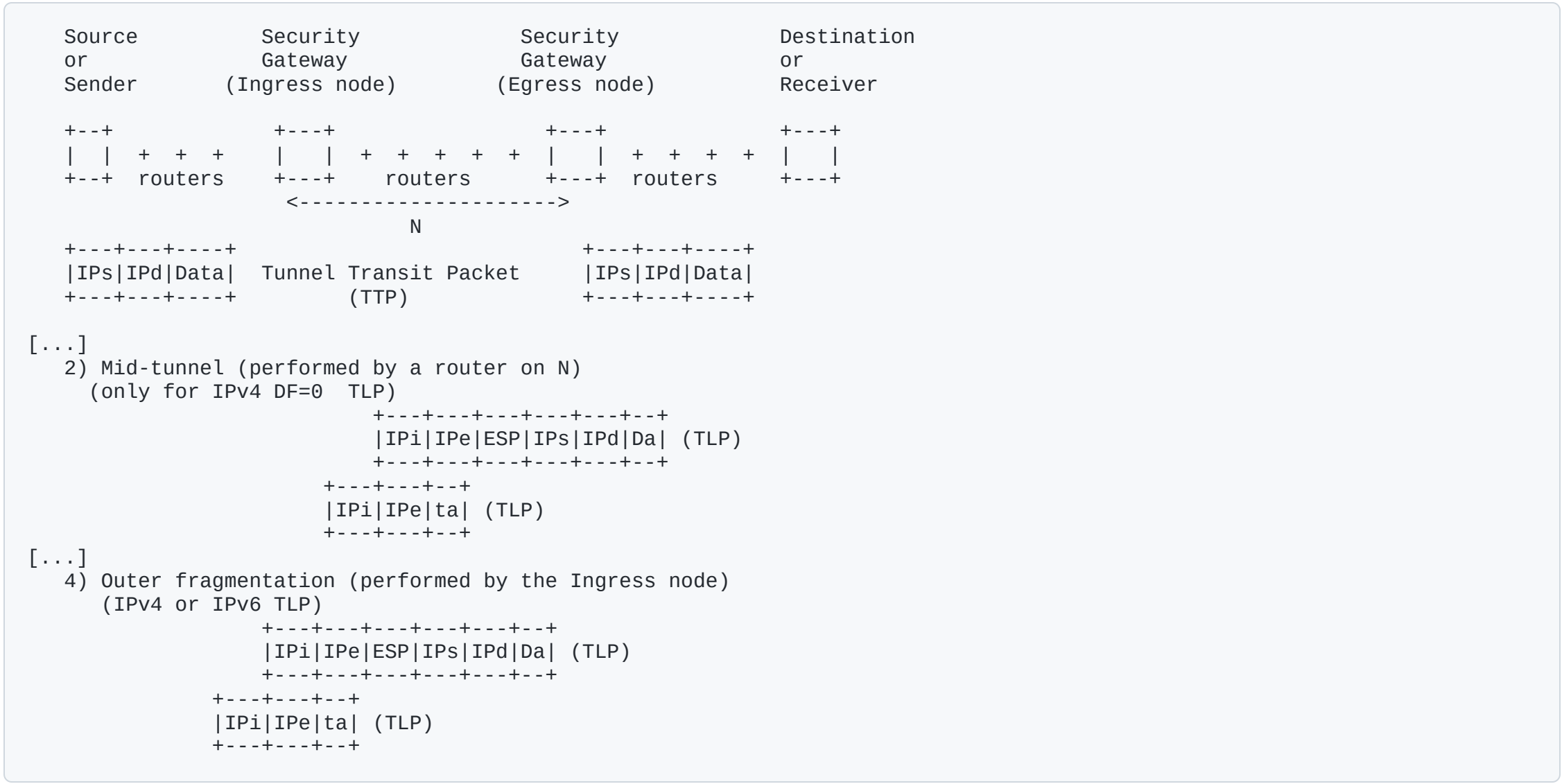
`draft-liu-ipsecme-ikev2-mtu-dect`

Liu, Zhang. Migault

Problem Statement

Fragments reassembling at the egress security gateway requires additional resources which under heavy load results in service degradations.

When Reassembly is observed ?



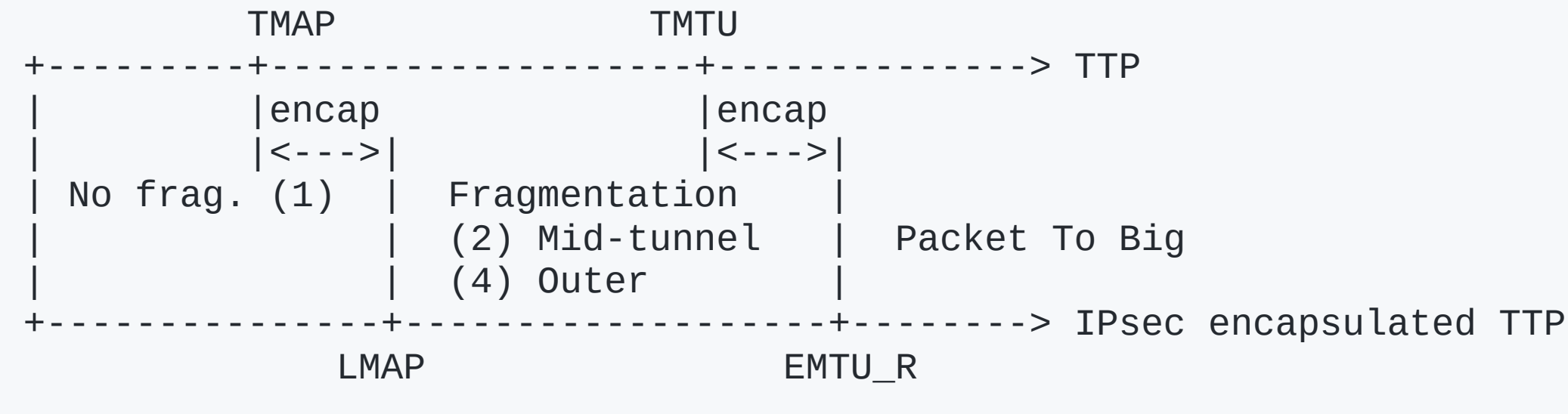
In both cases, the Egress nodes:

1. Reassembles fragments for an IPsec packet

- fragment < LMAP

2. Processes the reassembled IPsec packet

- (reassembled) IPsec encapsulated TTP < EMTU_R



We define two notification payload:

1. Link Maximum Atomic Packet Notification (LMTA)

- To inform the Ingress node of the observed LMAP

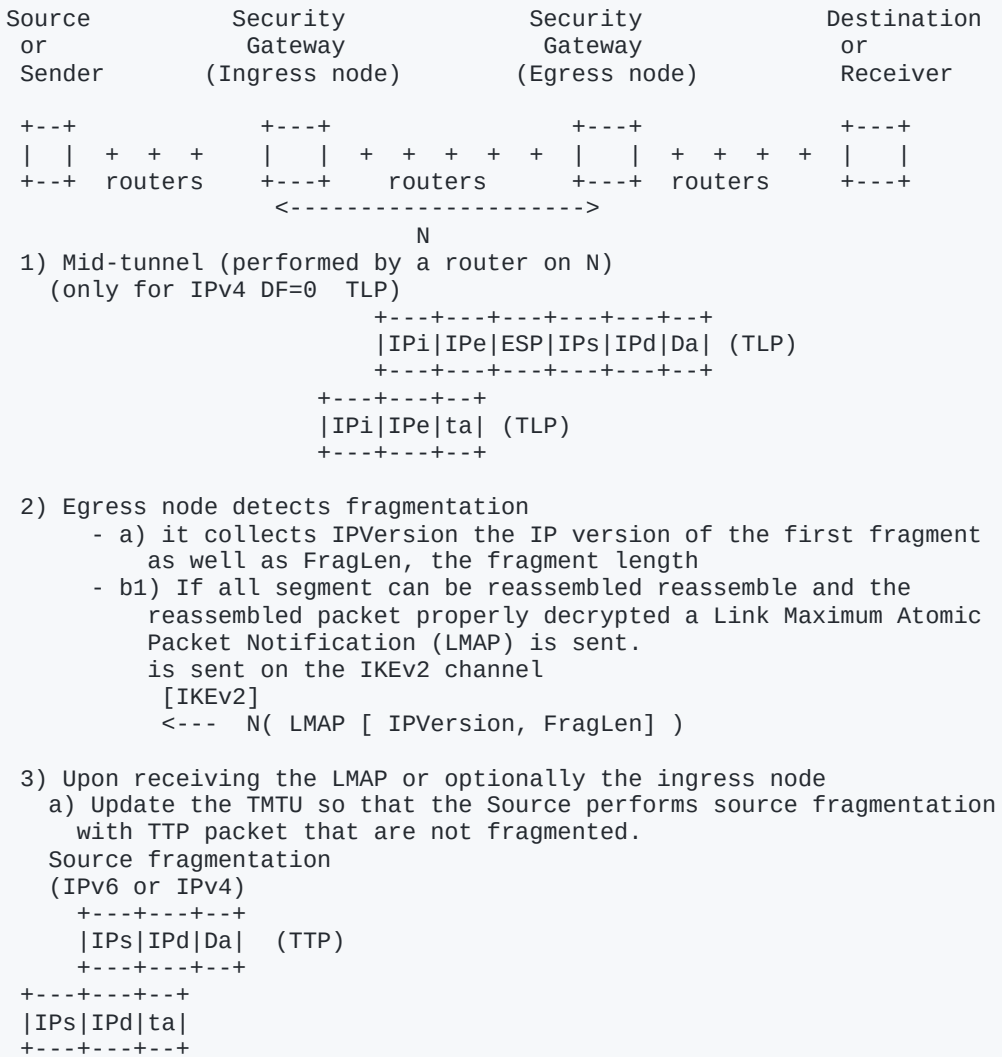
2. Packet Too Big Notification (PTB)

- to inform the Ingress node of the EMTU, LMTU

Given LMTA, EMTU_R the Ingress node is able to:

1. Compute the TMAP and TMTU
2. Inform the Source of appropriate TTP size (or perform inner fragmentation)

Illustrative Example (LMAP)



Where we are:

We considered ALL received comments with `ietf-intarea-tunnels` terminology, Generalize the protocol to IPv4 and IPv6, add PTB,...

We implemented it and it solves our issue.

Remaining discussion:

- `ietf-intarea-tunnels` considers the router component - carrying the TTP - and the interface component - handling LTP - independent. Independence of the Tunnel MTU (for TTP) and link layer MTU for (LTP) is provided by performing outer fragmentation when needed.
- [RFC4301] takes another view considers the router component can adapt to the specific needs of the interface component. This is what we do here.

We are looking for adoption

Thanks.

Presentations

- Issues SA TS Payloads opt draft – Paul Wouters
- Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov
- Extended IKEv2 Payload Format – Valery Smyslov
- Anti replay subspaces – Mohsin Shaikh
- IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault
- **Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault**
- Inter-domain source address validation using RPKI and IPsec

Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints (DSCP)

`draft-mglt-ipsecme-ts-dscp-00`

Daniel Migault, Joel Halpern, Ulf Parkholm

Goal

Specifying a new TS Type TS_DSCP for IKEv2 to negotiate DSCP as additional selectors for the SPD.

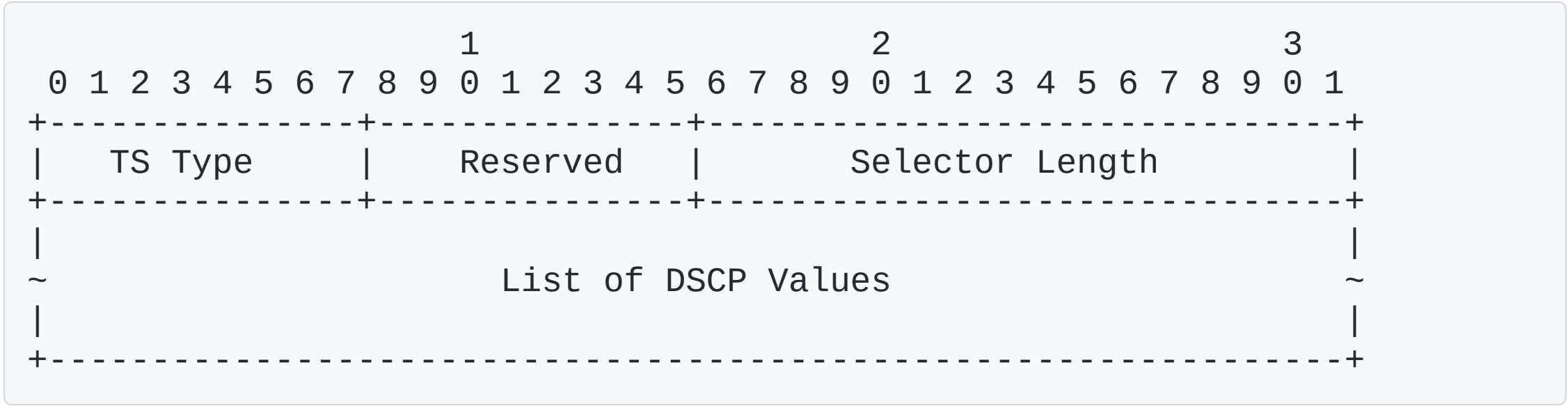
RFC4301 Section 4.1 acknowledges that aggregating traffic with mutliple DSCP over the same SA may result in inappropriate discarding of lower priority but recommends a **classifier** mechanism which dispatches the traffic over multiple SAs.

Such **classifier** results in inbound and outbound traffic may take SA negotiated via different IKEv2 sessions and thus makes:

- SA management more complex with an unnecessary SAs.

This is especially an issue with hardware implementations are designed with a limited number of SAs

Defining new TS that includes a range of acceptable DSCP: TS_DSCP_LIST



The CREATE_CHILD_SA request for rekeying a Child SA is:

```
Initiator                              Responder
-----------------------------------------------------------------
HDR, SK {N(REKEY_SA), SA, Ni, [KEi,]
    TSi, TSr}   -->
    with:
      TSi = ( TS_IPV6_ADDR_RANGE, TS_DSCP_LIST1 )
      TSr = ( TS_IPV6_ADDR_RANGE )


                                  <--  HDR, SK {SA, Nr, [KEr,]
                                           TSi, TSr}
    with:
      TSi = ( TS_IPV6_ADDR_RANGE, TS_DSCP_LIST1 )
      TSr = ( TS_IPV6_ADDR_RANGE )
```

We are looking for a call for adoption.

Thanks!

Presentations

- Issues SA TS Payloads opt draft – Paul Wouters
- Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security – Valery Smyslov
- Extended IKEv2 Payload Format – Valery Smyslov
- Anti replay subspaces – Mohsin Shaikh
- IKEv2 IPv4 Link Maximum Atomic Packet Notification Extension – Daniel Migault
- Traffic Selector for Internet Key Exchange version 2 to add support Differentiated Services Field Codepoints – Daniel Migault
- **Inter-domain source address validation using RPKI and IPsec**

An RPKI and IPsec-based AS-to-AS Approach for Source Address Validation

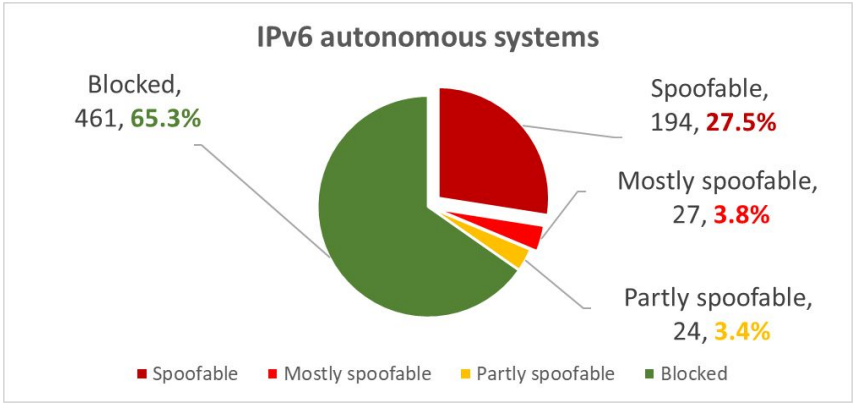
draft-xu-ipsecme-risav-00: https://datatracker.ietf.org/doc/draft-xu-ipsecme-risav/
Github: https://github.com/bemasc/risav/

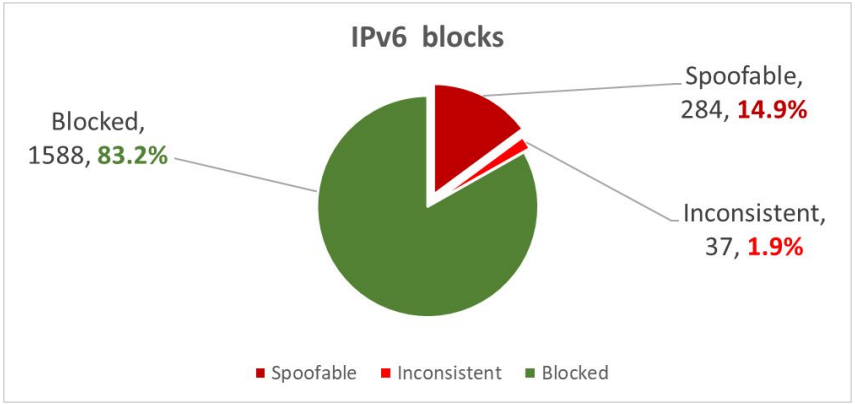
SAV question definition

Vulnerability: It is difficult to resist attacks by disabling the IP source address.

Traceability: Attackers could conceal location and identity.

Manageability: It is difficult to realize billing and other management through the IP source address.

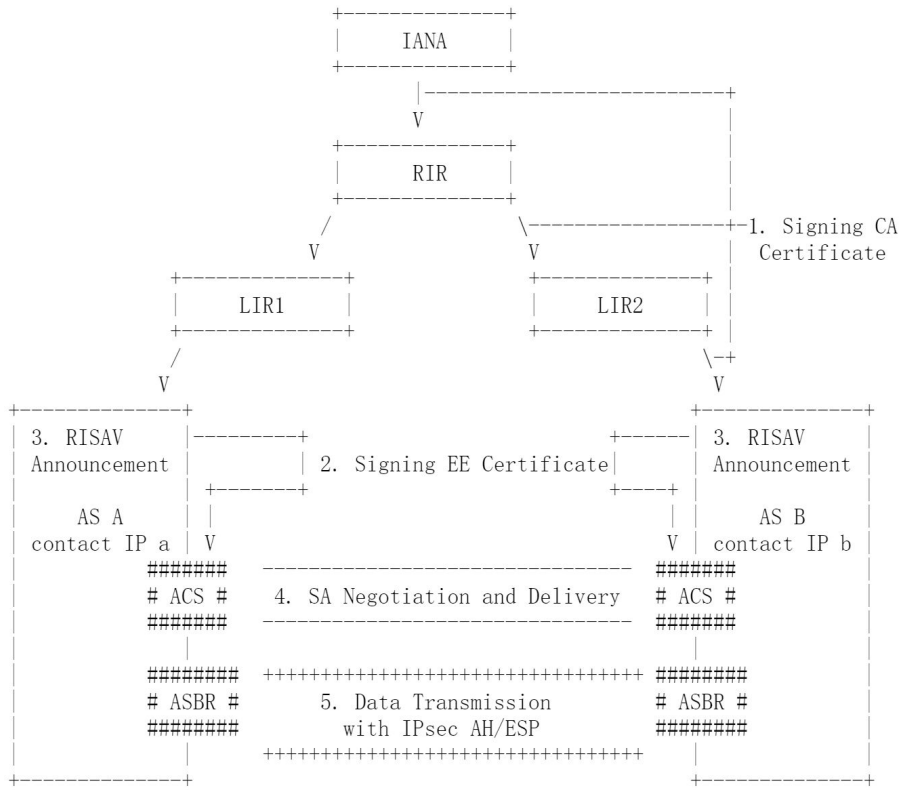




REF: https://spoofer.caida.org/summary.php

Overview

- cryptographically-based inter-AS SAV protocol
- RPKI + IPsec compatible
- add MAC at source ASBR and delete it at destination ASBR



Control plane

Enabling RISAV

- Announcing that this AS supports RISAV.
- Publishing contact IPs.
  - RISAVAnnouncement: a Signed Object, testing for indicating the reliability of contact IP.

```
RISAVAnnouncement ::= SEQUENCE {
        version [0] INTEGER DEFAULT 0,
        asID ASID,
        contactIP ipAddress,
        testing BOOLEAN }
```

- Performing IPsec session initialization (i.e. IKEv2).

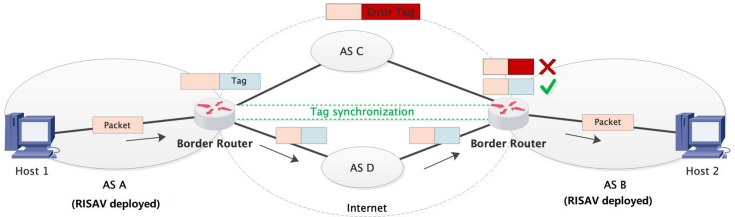
Green Channel

- A channel established only between pair ACSes.
- For rebooting quickly and imperceptible
- When it enabled, ASBRs don't perform RISAV validation.

Disabling RISAV

- Targeted Shutdown
  - NO pair of inbound-outbound SAs. => strictly unidirectional SA.
  - If one AS sends NO_ADDITIONAL_SAS to its peer, it means the peer MUST halt all further RISAV negotiation temporarily.
  - Deleting all SAs and rejecting new ones.
- Total Shutdown
  - Apply a targeted shutdown
  - Stop requiring RISAV authentication of incoming packets.
  - Remove the "RISAVAnnouncement" from the RPKI Repository.
  - Wait at least 24 hours.
  - Shut down the contact IP.

Data plane



Transport mode

```
                    1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   | Payload Len   |    RESERVED    |     Scope     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Security Parameters Index (SPI)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Sequence Number Field                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+             Integrity Check Value-ICV (variable)              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Updated AH Format.

- ONLY the “Scope” field, which identifies the scope of protection for RISAV AH, is different from the original AH.
  - 0 for IP and 1 for AS; others not defined.
- Only used for AS-to-AS communication
- Only indexed by SPI and counterpart ASN regardless of src IP or dst IP in SAD
- Transparent to the end hosts.

Tunnel mode

- ESP encapsulation
- Tunnel is built with current ASBR and ACS’s contact IP of another AS
- ASBR maintains its own SAD indexed by SPI and counterpart ASN

RISAV implementations **MUST** support transport mode, and **MAY** support tunnel mode.

- USE_TRANSPORT_MODE notification

MTU Handling and Replay Protection

Choose a **minimum** acceptable "**inner MTU**" and reject RISAV negotiations whose inner MTU is **lower than** inner MTU.

- Prior knowledge of the outer MTU
- Estimation of the outer MTU

ICMP PACKET TOO BIG(PTB)

- Transport Mode
  - MTU value reduced by the total length of RISAV AH header
- Tunnel Mode
  - Be treated as single IP hop
  - Oversize will cause generating PTB

MTU Estimation

- Initial estimation
  - PMTUD (RFC 7383)
- MTU monitoring

Traffic Selector and Replay Status

- Simplest RISAV Configuration
  - Single Child SA (**SHARING one**)
  - TSi lists all the IPs of sending AS
  - and TSr lists all the IPs of receiving AS

Enabling Replay Protection

- Sender creates many Child SAs and narrow the TSi.
- each SA is processed by a single receiving ASBR
- Tunnel Mode: route each SA to a specific ASBR using IKEv2 Active Session Redirect.
- Transport Mode:

Disabling Replay Protection

- Set the REPLAY-STATUS indication to False in CREATE_CHILD_SA notification,
- and delete the SA if….

AS IP Ranges

Others

Security Consideration

1. Threat model
   a. Reply attack
   b. Downgrade attack
2. Incremental benefit
3. Comparability
   a. IPsec
   b. Other SAVs

Operational Consideration

1. Reliability
2. Multiple ASBRs
3. Performance
4. NAT

Consistency with Existing Protocols

- IPv6
  - MTU: minimum of 1280B. {MTU-Handling}
  - Header Modification: RISAV-AH
  - IP address usage
- RPKI Usage
  - RISAV fully falls squarely within the limits of usage of RPKI key material.

Thanks

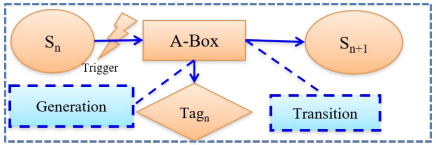
Possible Extensions

Header-only Authentication

It only authenticates the **IP source address**, **IP destination address**, etc.

An attacker could simply replace the payload, allowing it to issue an unlimited number of spoofed packets.

Time-base key rotation



Time triggers the SM transit from **S(n)** to **S(n+1)** following the algorithm defined by two parties as well as generating the tags as the side product.

Static-static ECDH negotiation

Ideas from RFC 6278

It would allow ASes to agree on shared secrets simply by syncing the RPKI database.

Pros.

- Stateless

Cons.

- Novel IPsec negotiation mechanism

Open Discussion

- Other points of interest?