IP Security Maintenance and Extensions (IPsecME) WG

IETF 115, Wednesday, November 9th, 2022

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/ietf115/?group=ipsecme&short=&item=1

Notes: https://notes.ietf.org/notes-ietf-115-ipsecme

Agenda

- Note Well, technical difficulties and agenda bashing – Chairs (5 min) (15:00-15:05)
- Document Status – Chairs (5 min) (15:05-15:10)
- Presentations
  - Multi-SA update (5 min) (15:10-15:15)
  - IPsec workshow report (10 min) (15:15-15:25)
  - IPComp Extension (10 min) (15:25-15:35)
  - New IKEv2 payload format (15 min) (15:35-15:50)
  - Revised Cookie Processing (10 min) (15:50-16:00)
  - Inter-domain source address validation using RPKI and IPsec (15 min) (16:00-16:15)
  - IKEv2 Optional SA&TS Payloads in Child Exchange (10 min) (16:15-16:25)
  - IPsec anti-replay subspaces (10 min) (16:25-16:35)
- If time permits
  - Traffic selector with DSCP
  - MTU fragmentation

WG Status Report

- RFF Editor queue:
  - draft-ietf-ipsecme-iptfs
  - draft-ietf-ipsecme-mib-iptfs
  - draft-ietf-ipsecme-yang-iptfs
  - draft-ietf-ipsecme-rfc8229bis
- Publication requested:
  - draft-ietf-ipsecme-ikev1-algo-to-historic IETF Last Call
  - draft-ietf-ipsecme-ikev2-multiple-ke IESG Eval
- Waiting for write-up / Chair review:
  - draft-ietf-ipsecme-labeled-ipsec
- Working Group Last Call:
  - draft-ietf-ipsecme-add-ike
- Work in progress:
  - draft-ietf-ipsecme-auth-announce
  - draft-ietf-ipsecme-g-ikev2

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

Presentations

- **Multi-SA update – Paul Wouters**
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

DRAFT-PWOUTERS-MULTI-SA-PERFORMANCE

IPsec, IETF 115
November 2022

Antony Antony, Tobias Brunner, Steffen Klassert, Paul Wouters

Draft history

- -00 had multi CPU and QoS
- -01 removed QoS, updated max negotiation logic
- -02 Added TS_MAX_QUEUE notify error message
- -03 updated Linux code reference
- -04 no changes
- -05 removed special Fallback SA, send both NOTIFYs per SA

- No more issues left, overdue for WG Adoption call :)

Implementation Status:

- Linux kernel XFRM implementation
  - Including per-cpu (on-demand) ACQUIRE messages
- Libreswan implementation
  - Basic: implements preconfigured number of IPsec SAs
- Strongswan implementation
  - Basic: implements preconfigured number of IPsec SAs
- See draft Implementation Status for links to software

Presentations

- Multi-SA update – Paul Wouters
- **IPsec workshow report – Steffen Klassert**
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

IPsec workshop 2022
November 3th – 4th

Report

Steffen Klassert

Some background about the event

- Funded by IPsec and Network Security Association
- Yearly event
- Held first time in 2018
- Semi public (< 20 attendees)
- Topics: IPsec Implementation + Protocol

FIPS requirements for AES-GCM
(Paul Wouters)

- Question: Can we use the same key for more than 2^32 packets?
- Yes: Can use 2^64 packets even in FIPS mode
- Limitation: 8 octets ICV is limited to 2^32 packets in FIPS mode

Decorrelated policies – avoid overlapping policies
(Tero Kivinen)

- Overview of decorrelated policies with examples
- Gives a flat SPD structure without priorities
- Makes lookups more efficient

Full IPsec datapath HW offload
(Leon Romanovsky)

- Offload lookups, encapsulation and crypto operations to HW
- Offloading API for Linux exists
- Nvidia/Mellanox CX-7 supports this offload type
- Linux + CX-7 can run the full datapath offload

Linux forwarding fastpath with packet bulking
(Pablo Neira Ayuso / Steffen Klassert)

- Use Netfilter flowtable
- Skip full L3 datapath

- Create packet bulks (packets matching same SA processed together)
- Run on small code loops (cache frindly)

- Gives good performance results (factor 5)

ANIMA and use of IPsec
(Michael Richardson)

- Introduction to ANIMA
- IPsec usage in ANIMA

- Problem: Cross network namespaces with VTI interfaces

- Proposed solution: Use xfrm interfaces
- xfrm interfaces were created to replace VTI

IPTFS
(Christian Hopps)

- Introduction to IPTFS
- Presentation/Discussion about state of Linux implemention:
    - Aggregation / Fragmentation supported
    - Constant rate sending not yet supported

Draft-pwouters-ipsecme-multi-sa-performance (Steffen Klassert)

- Crtitsm on ML: Fallback SA is treated special
- Proposed solution: Remove fallback SA from the darft
- No architectural changes
    - ‚Low hanging fruit’
    - Can continue without charter changes

Re-designing ESP
(Steffen Klassert)

- Lot of proposals around to support multi cpu case, QoS classes, HW offloads
- Need separate anti-replay windows
- Proposed solutions:
  - Use some bits from SPI
  - Use some bits of anti-replay window
  - Add new field to ESP
- Google publishsed PSP for HW offload

» **Time to rethink ESP, maybe create ESP-v4**

Standardizing BEET mode
(Steffen Klassert / Antony Antony)

- Draft-nikander-esp-beetmode-09 (from 2008) unfinished
- BEET mode is implemented im Linux
- People use it!

» **Continue the work on BEET mode**

Presentations

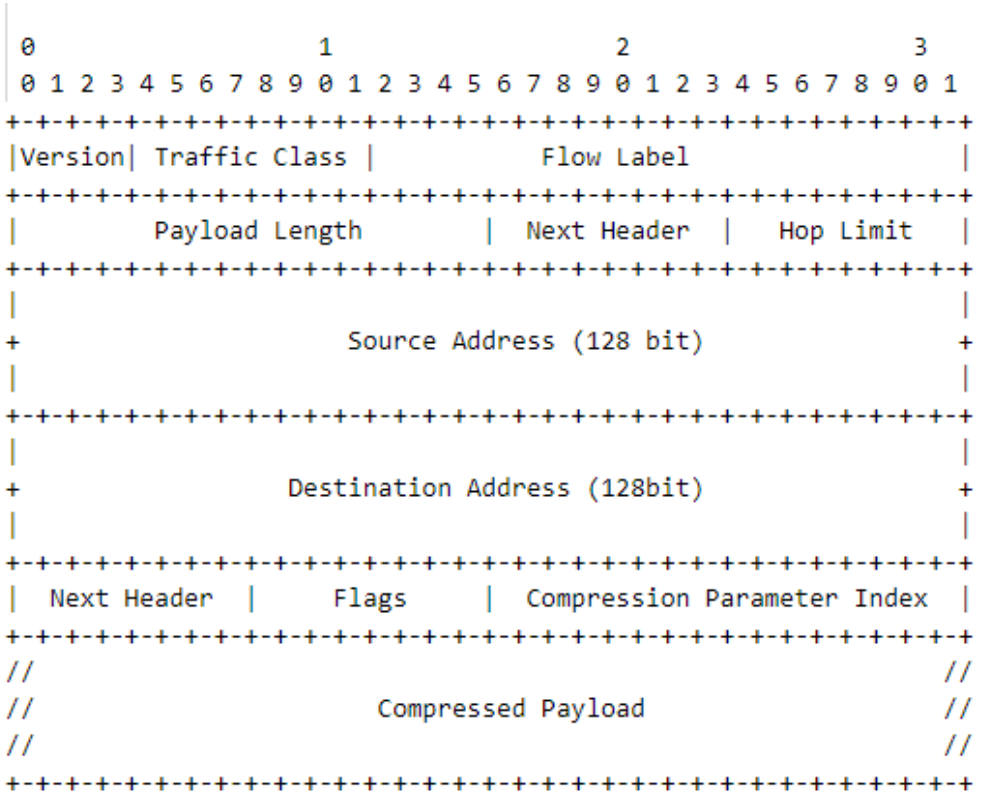
- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- **IPComp Extension – Hang Shi**
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

IPComp excluding transport layer

Hang Shi/Cheng Li/Meng Zhang/Xiaobo Ding

Background on IPComp

- IP Payload Compression Protocol(IPComp) compress IP payload to save bandwidth
- Next header = original next header
- Flags: Must be 0
- Compression Parameter Index(CPI) to indicate compression algorithm



Value	Transform ID	References
0	RESERVED	[RFC2407]
1	IPCOMP_OUI	[RFC2407]
2	IPCOMP_DEFLATE	[RFC2407]
3	IPCOMP_LZS	[RFC2407]
4	IPCOMP_LZJH	[RFC3051]
5-47	Reserved for approved algorithms	
48-63	Reserved for private use	
64-255	Unassigned	

https://www.iana.org/assignments/isakmp-registry/isakmp-registry.xhtml#isakmp-registry-11

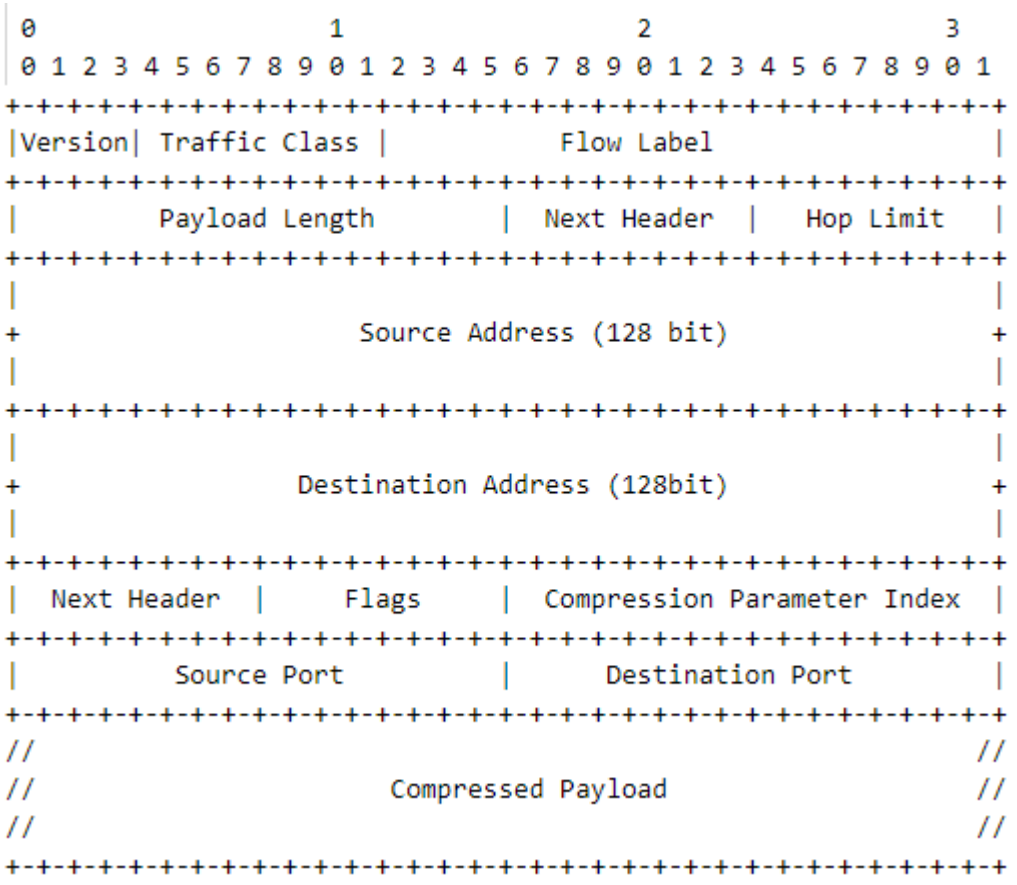
Problem1: incompatible with network functions

- Layer 4 information(Source port + Destination port) is compressed
- NAT, Firewall, ACL may need to inspect layer 4 info
- Can not deploy between IPComp nodes

Extension 1: four-bytes exclusion

- Exclude ports info from the compression range.
- Option 1: Change Flags, 0->1 bit indication
- Option 2: Change CPI, duplicate each compression algorithm codepoint

Value	Transform ID	References
0	RESERVED	[RFC2407]
1	IPCOMP_OUI	[RFC2407]
2	IPCOMP_DEFLATE	[RFC2407]
3	IPCOMP_LZS	[RFC2407]
4	IPCOMP_LZJH	[RFC3051]
TBD	IPCOMP_OUI with four bytes exclusion	This document
TBD	IPCOMP_DEFLATE with four bytes exclusion	This document
TBD	IPCOMP_LZS with four bytes exclusion	This document
TBD	IPCOMP_LZJH with four bytes exclusion	This document



Problem 2: Out-of-order processing

- If a flow is IPComp enabled but compression does not produce shorter payload, RFC 3713 says: sent uncompressed without IPComp header
- Out of order, packets with the IPComp header will go through decompression co-processor first

Extension 2: Uncompressed Payload

- Add IPComp header even if the payload is sent uncompressed
- Use a new CPI value for uncompressed packet

Comments?

- Currently, the CPI codepoint is allocated in the IPSec registry and negotiated use IKE, but …
- Compression is not related to security, CPI value does not have to be allocated by IKE, maybe BGP? Decouple with IPSec?
- For transport exclude L4 info, CPI or flag?

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- **New IKEv2 payload format – Valery Smyslov**
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

New IKEv2 Payload Format?

Valery Smyslov
svan@elvis.ru

IETF 115

Existing Format Limitation

- 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

Existing Format Redundancy

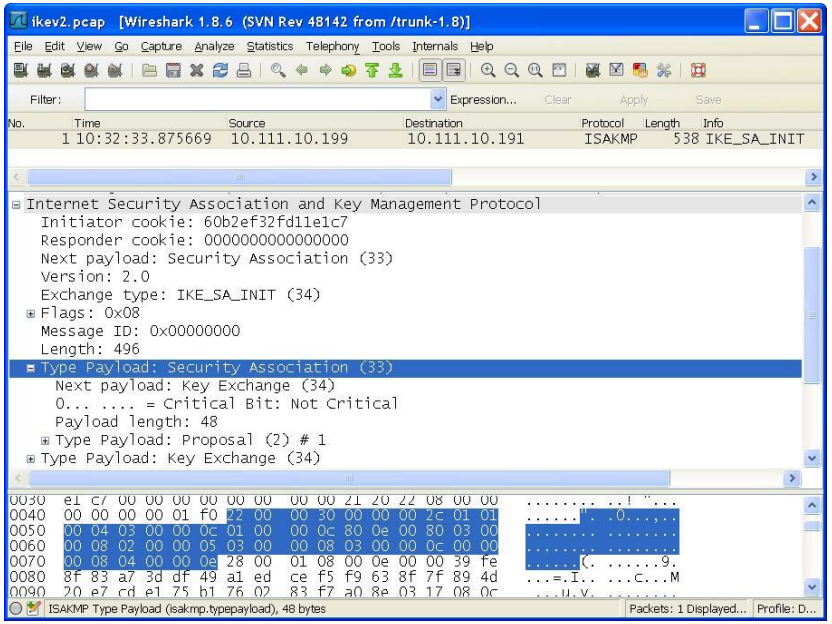
Many payloads contain substantial redundancy

- 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

Example: SA Payload on the right contains one Proposal with four Transforms:

- `ENCR_AES_CBC` (128 bits)
- `PRF_HMAC_SHA2_256`
- `AUTH_HMAC_SHA2_256_128`
- `2048-bit MODP Group`

Payload size is **48** bytes, among which **24** bytes are zeroes.



Lifting 64 Kbytes Size Limit

- Would allow using PQ algorithms with long public keys and signatures
  - Classic McEliece is NIST round 4 candidate, it is also recommended by some national state organizations (e.g. BSI in Germany)
- Would allow transferring large chunks of data (e.g. in CP payload)

Making Payloads Smaller

- Would decrease power and network bandwidth consumption (important for IoT devices)
- Would decrease chances of IP fragmentation in `IKE_SA_INIT` and chances of IKE fragmentation in the following exchanges
  - these chances grow as the number of transforms proposed by initiator increases making SA payload larger, e.g. when draft-ietf-ipsecme-ikev2-multiple-ke is used with full range of PQ algorithms with different parameters

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
  draft-smyslov-ipsecme-ikev2-compact
  (expired)

"A Larger Internet Key Exchange version 2 (IKEv2) Payload"

- addresses only 64Kbytes limitation
- generic solution suitable for any payload
  - payloads in new and old formats can be mixed in a message
- explicitly negotiated via exchange of notifies in `IKE_SA_INIT`
  - cannot be used in initial exchange (`IKE_SA_INIT`)
- relatively easy to implement (depending on base IKEv2 code)
  - no implementations exists (?)

“Beyond 64KB Limit of IKEv2 Payloads”

- addresses only 64Kbytes limitation
- suitable only for some payloads (`KE, AUTH, CERT`)
  - existing payload format is preserved
  - Encrypted Payload is mangled (zero payload length)
- no explicit negotiation, implicitly negotiated in `IKE_SA_INIT` by selecting transforms with large public keys
  - cannot be used in initial exchange (`IKE_SA_INIT`)
- relies on mandatory use of IKE fragmentation
- relatively easy to implement
  - implementations exist

“Compact Format of IKEv2 Payloads”

- addresses redundancy of IKE payloads
- suitable for any payload
  - compact and standard payloads can be mixed in a message
- some payloads have special, extremely compact format
- no negotiation, new initial exchange is used (`ALT_IKE_SA_INIT` instead of `IKE_SA_INIT`)
  - can be used in new initial exchange (`ALT_IKE_SA_INIT`)
  - initiator can revert to `IKE_SA_INIT` if this extension is not supported by responder (based on receiving of fatal error or on timeout)
- moderately difficult to implement
  - can be implemented as post-/pre- message processing
  - no implementations exist

Questions

- Do we want to revise IKE payload format?
- If yes, then what problems should be addressed:
  - remove 64K limitation?
  - decrease IKEv2 messages redundancy?
  - both?
- Any interest in this work?

Thanks!

Backup Slides

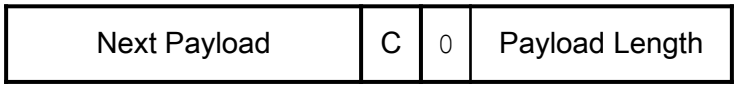
Possible new payload format that would support large payloads and also would make IKE messages smaller by eliminating some redundancy

New Format Overview

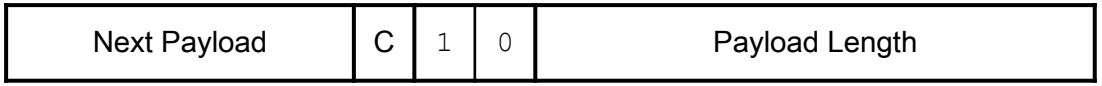
- Three formats for new Generic Payload Header
  - for small payloads (up to 64 bytes)
  - for medium size payloads (up to 8 Kbytes)
  - for large payloads (up to 512 Mbytes)
- No RESERVED fields
- Revise existing payloads headers to reduce their size
  - remove unnecessary fields
- Special Format for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

New Generic Payload Header

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)

Revised Existing Payload Headers

The following payload headers can be revised:

- Key Exchange, Identification, Authentication, Configuration
  - remove `RESERVED` field
- Notify
  - remove `SPI Size` field (can be deducted from Protocol ID)
- Delete
  - remove `SPI Size` field (can be deducted from Protocol ID)
  - remove `Num of SPIs` field (can be deducted from Payload Length)
- Traffic Selector
  - remove `RESERVED` field
  - remove `Number of TSs` field (can be deducted from Payload Length)

Special Format

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.

(*) Inspired by draft-smyslov-ipsecme-ikev2-compact

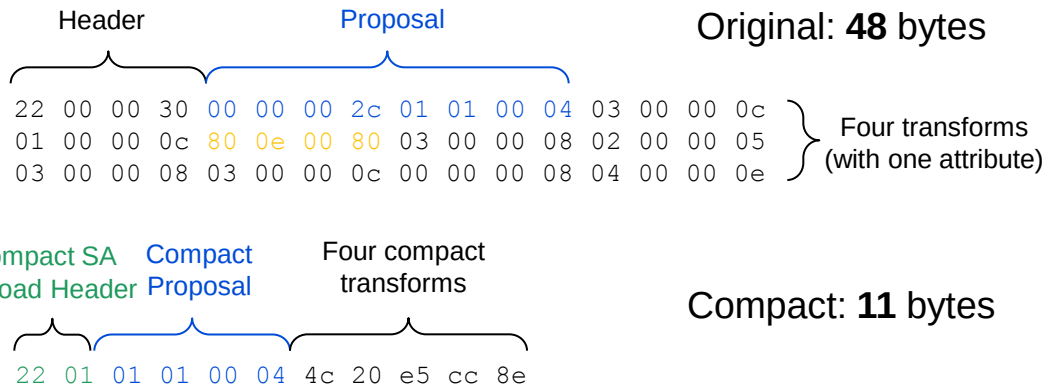
SA Payload

Outline:

- Remove all `RESERVED` fields
- Remove `Length` fields in substructures (where they are unnecessary)
- Encode all currently defined transforms w/o attributes using one octet (both `Transform Type` and `Transform ID`)
- Encode currently defined `Encryption` transforms having `Key Length` attribute using two octets
- Leave possibility to encode arbitrary (even not yet defined) `Transform Type` and `Transform ID`, as with regular format

Example: SA Payload with one Proposal and four Transforms:

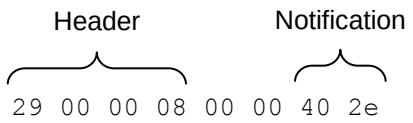
- `ENCR_AES_CBC` (128 bits)
- `PRF_HMAC_SHA2_256`
- `AUTH_HMAC_SHA2_256_128`
- `2048-bit MODP Group`



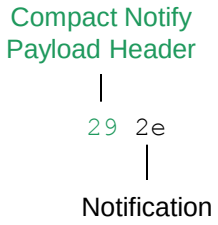
Notify Payload

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 major IKE version (v3)
  - old responders would return `INVALID_MAJOR_VERSION`
- New type of initial exchange (e.g. `ALT_IKE_SA_INIT`)
  - old responders would return `INVALID_SYNTAX`
- New critical payload in the `IKE_SA_INIT`, followed by payloads in new format
  - old responders would return `UNSUPPORTED_CRITICAL_PAYLOAD`

Discussion

- We don’t need to assign new payload types except for special format payloads (SA and empty status Notify), do we? What about revised payloads?
- Transport issues for transferring large payloads are out of scope
  - IKE over TCP combined with IKE fragmentation (to solve limitation on 64 Kbytes on a single IKE message over TCP)
  - mixed mode (draft-tjhai-ikev2-beyond-64k-limit: IKE over TCP combined with plain ESP or ESP over UDP) can be used to avoid ESP performance degradation of TCP encapsulation
- Certificates consume a lot of space, can be compressed
  - RFC 8879 is an example of certificate compression

Thanks again!

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- **Revised Cookie Processing – Valery Smyslov**
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

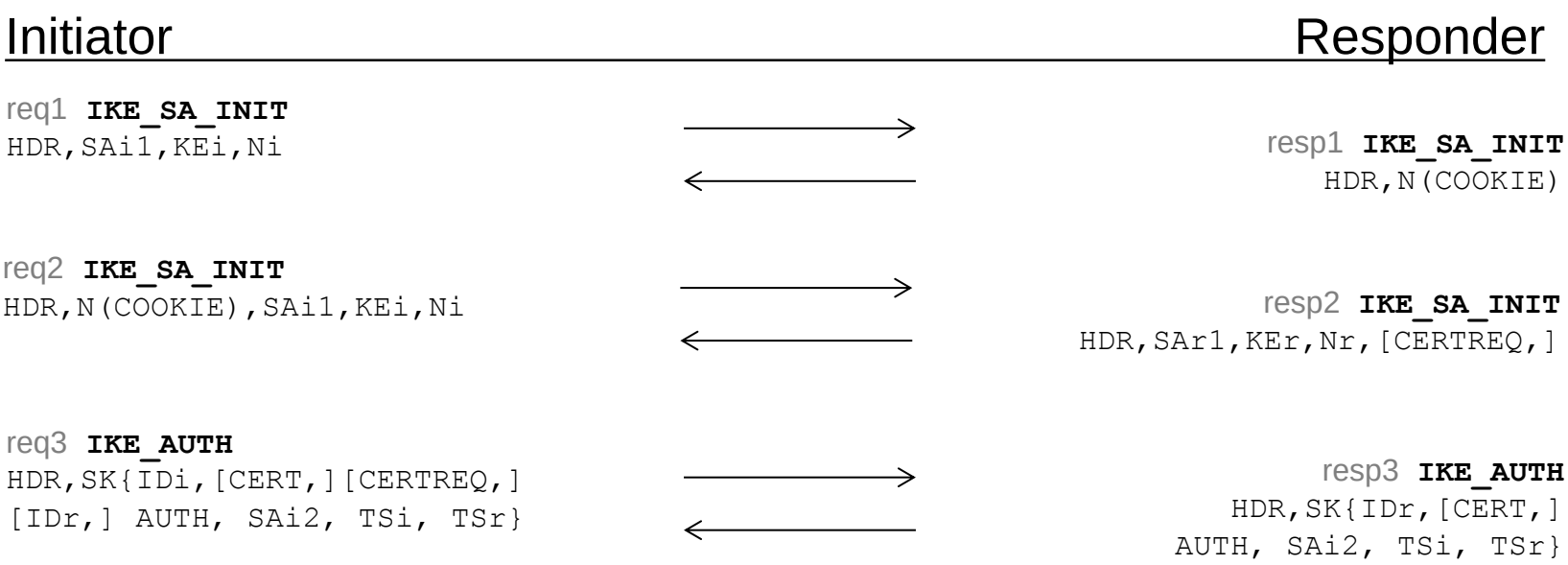
Revised Cookie Processing in IKEv2

draft-smyslov-ipsecme-ikev2-cookie-revised

Valery Smyslov
svan@elvis.ru

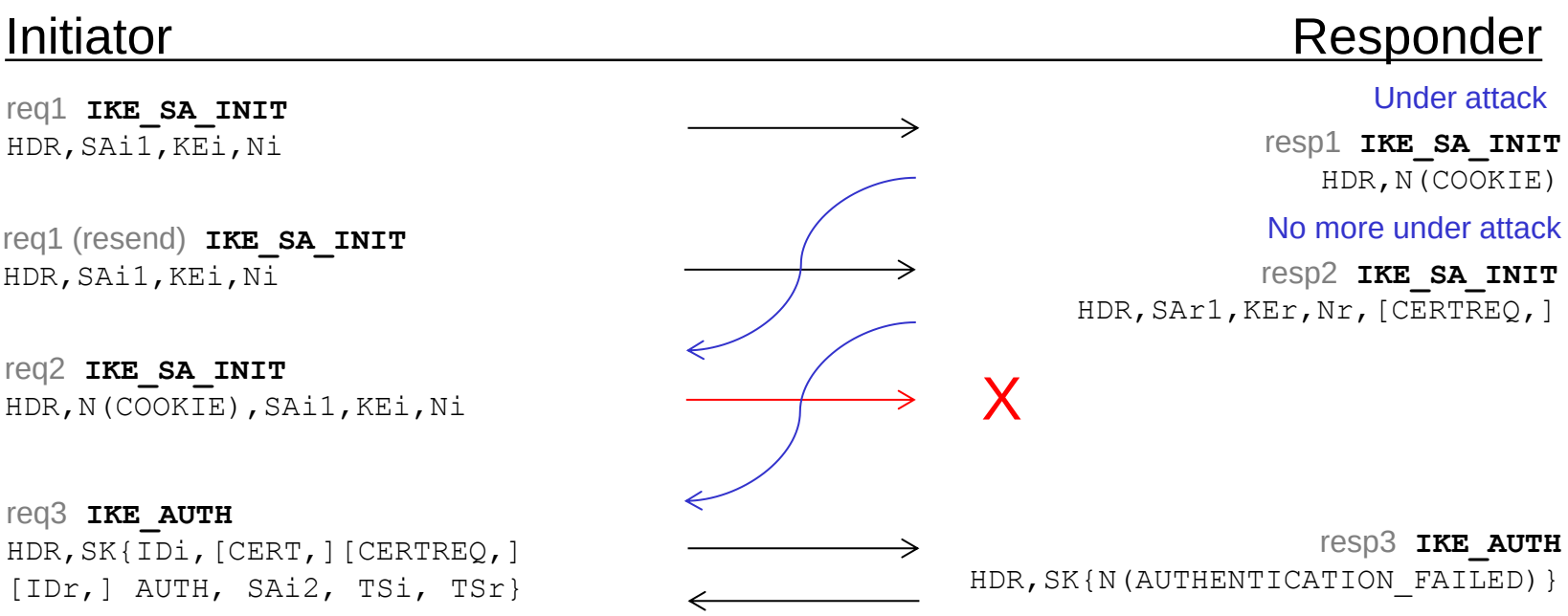
IETF 115

Using Cookies in IKEv2



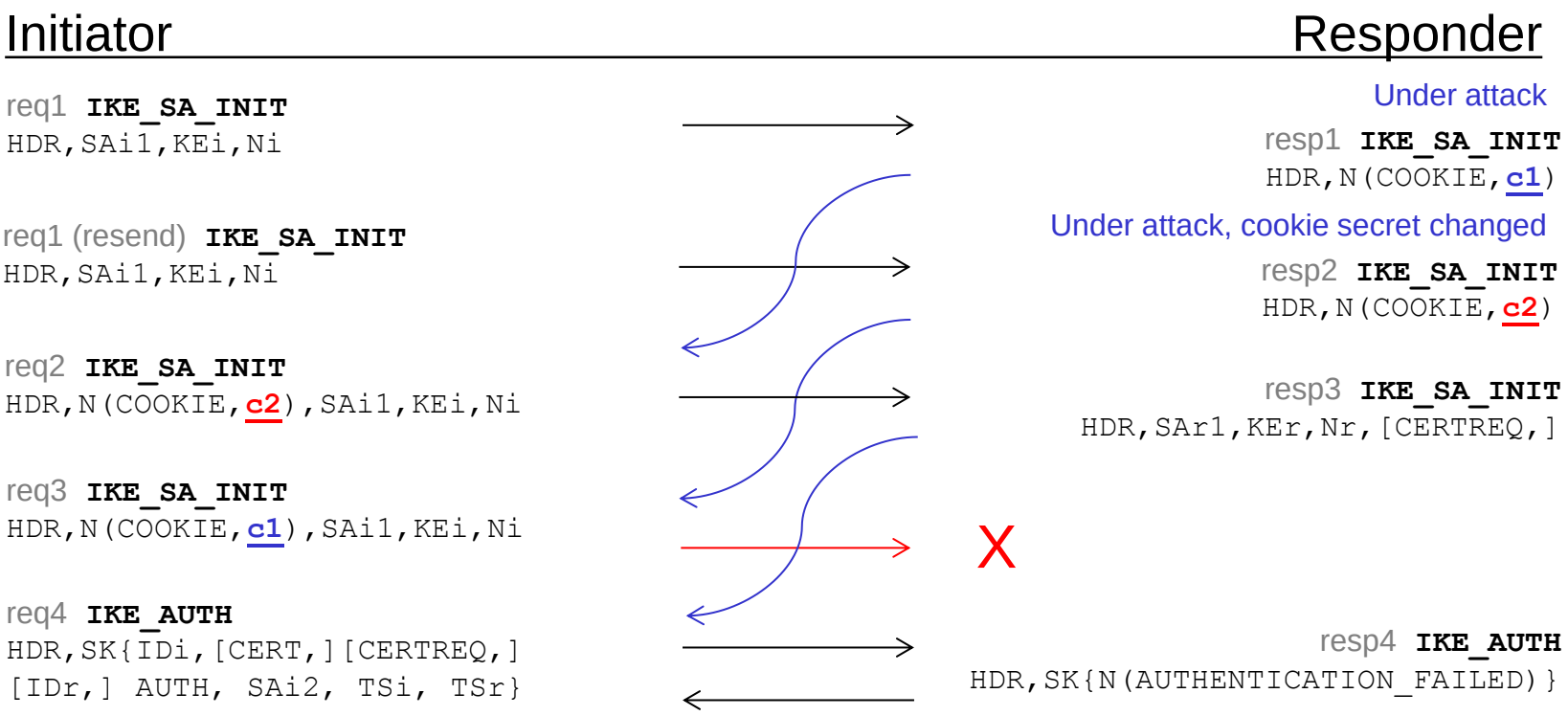
According to RFC 7296, the most recent `IKE_SA_INIT` request is included in the `AUTH` payload calculation in the `IKE_AUTH` exchange. In this example it is **`req2`** for both the initiator and the responder.

Problem Scenario 1



The most recent `IKE_SA_INIT` request sent by the initiator is **`req2`**, while the responder only received **`req1`**, so authentication would fail.

Problem Scenario 2



The most recent `IKE_SA_INIT` request sent by the initiator is **`req3`**, while the responder only received **`req2`**, so authentication would fail.

Source of the Problem

- The `IKE_SA_INIT` request can be sent several times with different content depending on the responder state
- If there is high probability of packets loss and reordering, then peers may complete the `IKE_SA_INIT` exchange having different views on what was the most recently sent `IKE_SA_INIT` request
- This request message is used in calculation of the `AUTH` payload. If peers use different messages for the calculation, the authentication would erroneously fail

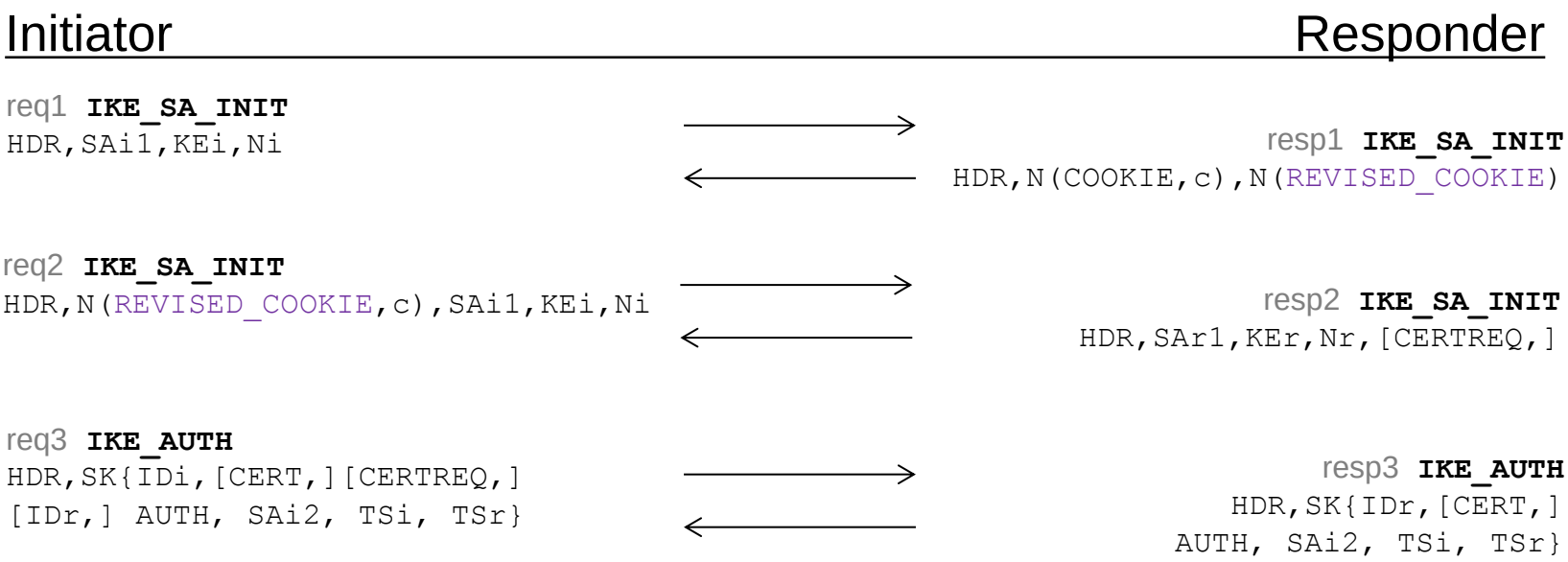
Severity of the Problem

- There are some preconditions for this problem to become noticeable
  - network with high probability of packet loss and delay
  - relatively frequent change of responder state (either changing cookie generation secret or changing responder's mind whether it is under attack)
- It might be extremely rare in normal conditions, but in stress tests we observed that up to 5% of SAs failed due to this problem
  - customers wonder why authentication sometimes fails with proper credentials
- This is a protocol flaw

Proposed Solution Overview

- Revise cookie processing by excluding Notify payload containing cookie (if present) from the `IKE_SA_INIT` request message when calculating the `AUTH` payload content
  - the cookie is already verified by the responder, no need to include it into the data to be authenticated
- For backward compatibility make the revised processing negotiable

Negotiation



Responder includes a new notification REVISED_COOKIE in the message containing COOKIE notification. If initiator also supports this extension, it returns cookie in this notification instead of COOKIE notification

Revised Cookie Processing

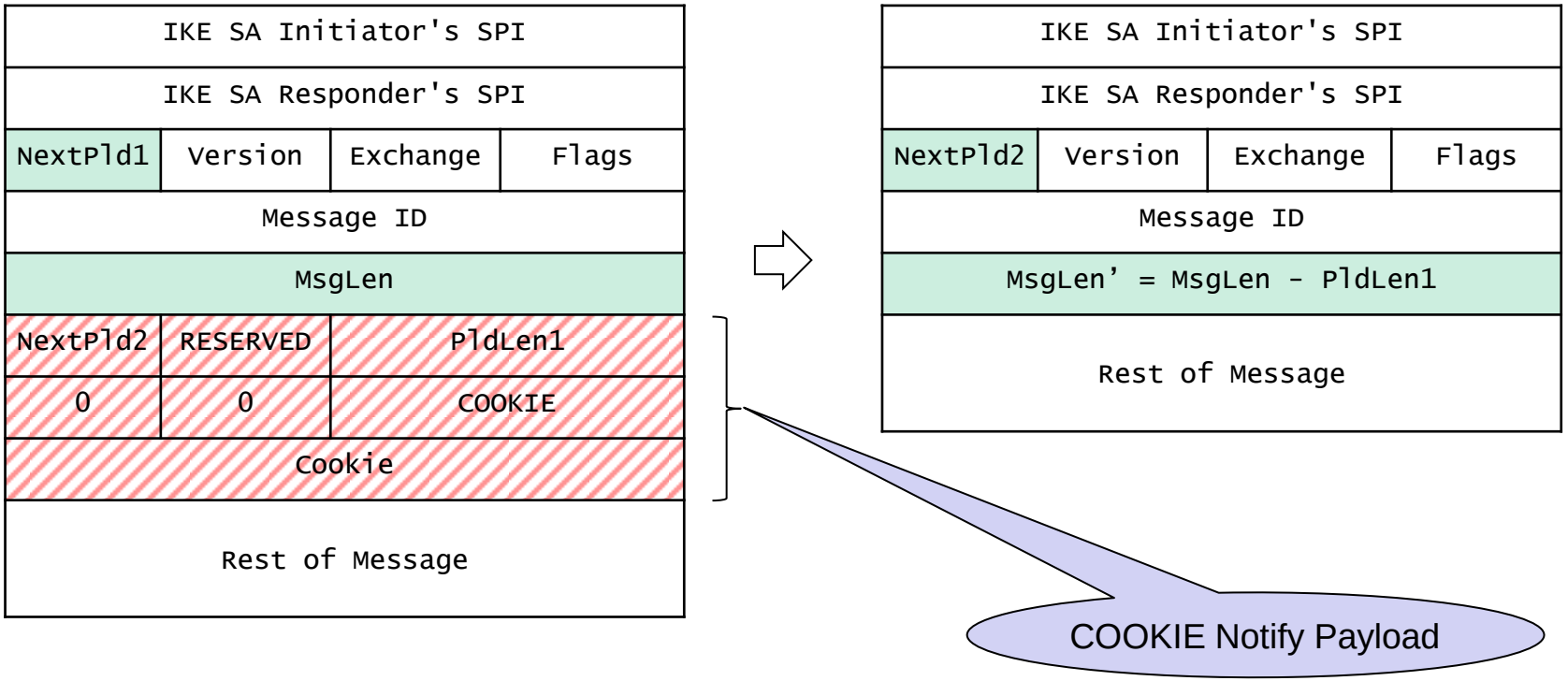
- If peers agreed upon using this extension then the cookie processing is changed
  - no changes in cookie anti-clogging function – responder still sends stateless cookie and when it is returned back by initiator it MUST be verified before message is processed

  According to RFC7296 initiator's `AUTH` payload is calculated by signing (or MAC'ing) the blob:

  ```
  InitiatorSignedOctets = RealMessage1 | NonceRData | MACedIDForI
  ```

  - if `COOKIE` Notify payload is present in RealMessage1 (i.e. in `IKE_SA_INIT` request message), then for the purpose of `AUTH` payload calculation the message is modified as if it contained no this payload

Adjusting IKE_SA_INIT Request for AUTH Payload Calculation



Thanks

- Comments? Questions?
- Is this problem worth to address?
- Is the suggested approach reasonable?
- WG adoption?

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- **Inter-domain source address validation using RPKI and IPsec – Yangfei Guo**
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

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

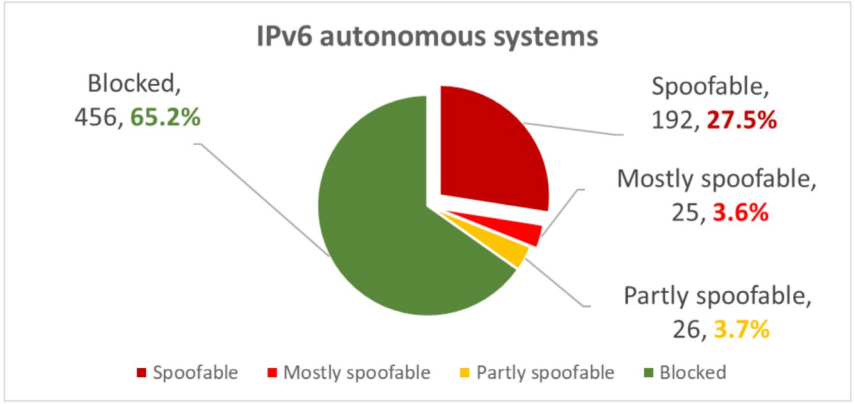
draft-xu-risav-02: https://datatracker.ietf.org/doc/draft-xu-risav/
Github: https://github.com/bemasc/draft-xu-risav/

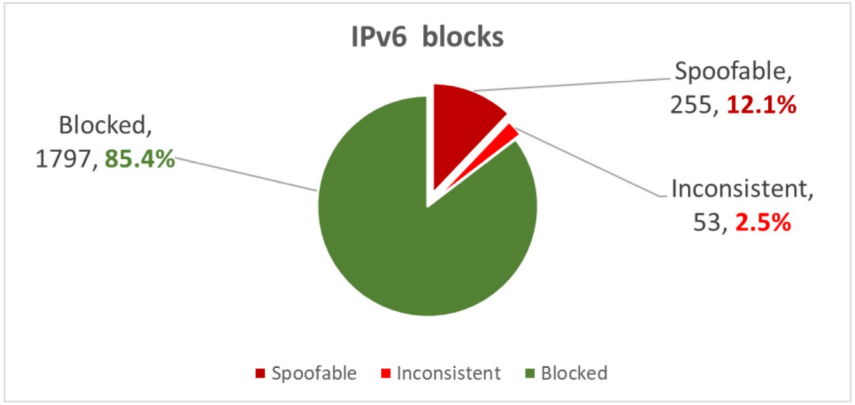
SAV question definition

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.

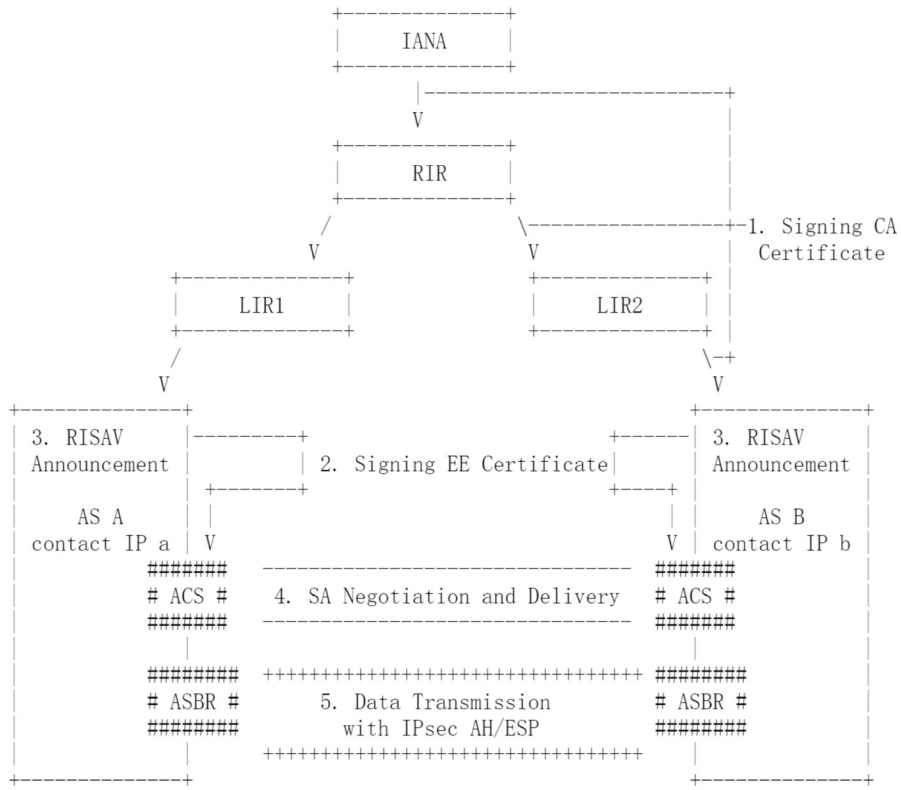




REF: https://spoofer.caida.org/summary.php

What RISAV is and how it works

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



Control plane

Enabling RISAV

- Announcing that this AS supports RISAV.
  - RISAVAnnouncement: testing for indicating the reliability of contact IP.
- Publishing contact IPs.
- Performing IPsec session initialization (i.e. IKEv2).

Disabling RISAV

- Stop requiring RISAV authentication of incoming packets.
- Remove the RISAVAnnouncement from the RPKI Repository.
- Wait at least 24 hours.
- Stop sending RISAV and shut down the contact IP.

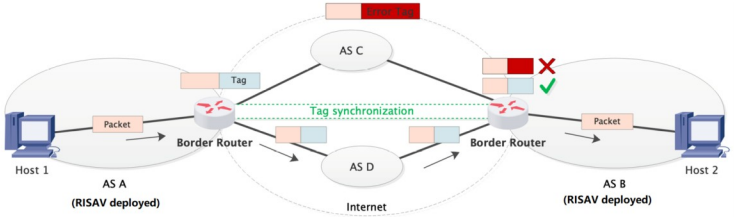
```
RISAVAnnouncement ::= SEQUENCE {
        version [0] INTEGER DEFAULT 0,
        asID ASID,
        contactIP ipAddress,
        testing BOOLEAN }
```

OPEN QUESTION:
Does IKEv2 have an authenticated permanent rejection option that would help here?

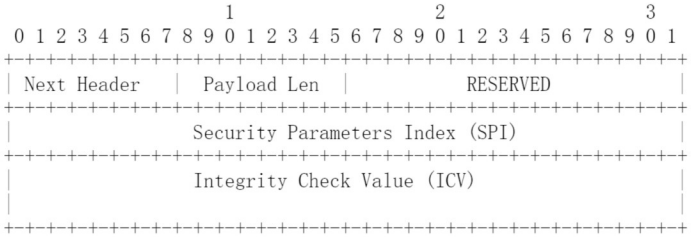
RFC 6023: CHILDLESS_IKEV2_SUPPORTED

IKEv2: TS_UNACCEPTABLE or
NO_PROPOSAL_CHOSEN

Data plane



Transport mode



- `Seq Num` field is omitted for this is presumed to be a `multi-sender SA`
- Only used for AS-to-AS communication
- Only indexed by SPI and counterpart ASN regardless of src IP or dst IP in SAD

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

PROBLEM: ICV in ESP is optional. ESP doesn't protect the source IP in default.

OPEN QUESTION: How do peers express a preference or requirement for transport or tunnel mode?

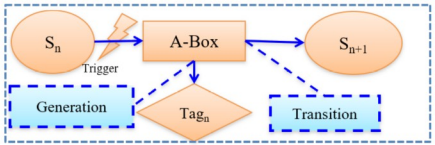
Possible Extensions

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

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. MTU
5. NAT

Open Questions

1. Does IKEv2 have an authenticated permanent rejection option that would help to disable RISAV normally and orderly?

2. How do peers express a preference or requirement for transport or tunnel mode?

3. PROBLEM: Can we negotiate an extension to ESP that covers the IP header? Or could we always send from the contact IP and encode the ASBR ID in the low bits of the SPI?

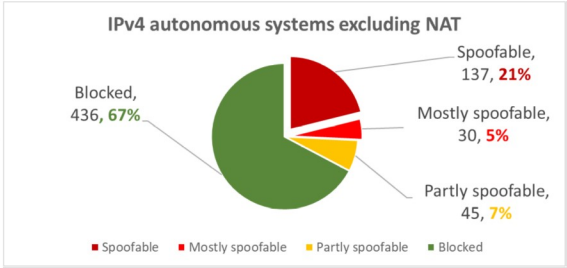
Thanks

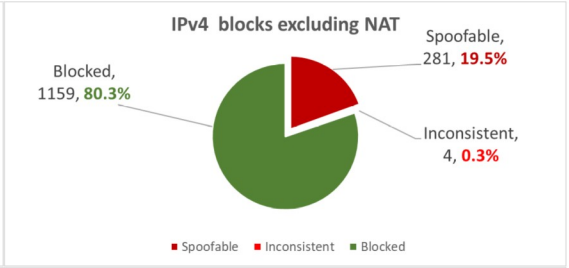
SAV question definition

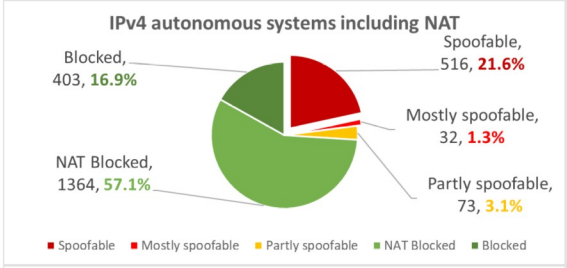
Vulnerability: It is difficult to resist attacks by disabling the IP source address.

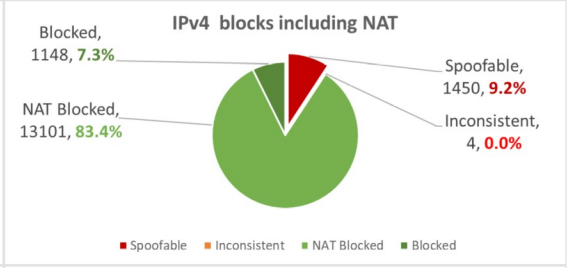
Traceability: Attackers could conceal location and identity.

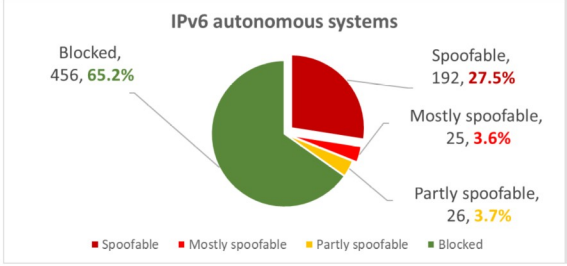
Manageability: It is difficult to realize billing and other management through the IP source address.

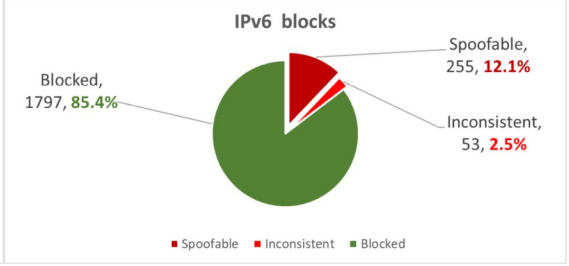












REF: https://spoofer.caida.org/summary.php

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- **IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan**
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

IKEv2 Optional SA&TS Payloads in Child Exchange

https://datatracker.ietf.org/doc/draft-kampati-ipsecme-ikev2-sa-ts-payloads-opt/

Sandeep Kampati (Microsoft)

Wei Pan (Huawei)

Paul Wouters (Aiven)

Meduri Bharath (Mavenir)

Meiling Chen (CMCC)

IETF 115

November 2022

Background:

- In RFC 7296 Section 2.8, it says:
  - `... Note that, when rekeying, the new`
  - `Child SA SHOULD NOT have different Traffic Selectors and algorithms`
  - `than the old one.`
- So, omitting the SA&TS payloads during rekey:
  - doesn’t violate RFC 7296,
  - can save the bandwidth in wire,
  - and can reduce the CPU operations.

Solution Recap (same as IETF 113):

- Negotiation of Support for OPTIMIZED REKEY

```
Initiator                           Responder
-----------------------------------------------------------------------
HDR, SK {IDi, [CERT,] [CERTREQ,]
    [IDr,] AUTH, SAi2, TSi, TSr,
    N(OPTIMIZED_REKEY_SUPPORTED)} -->
                                    <-- HDR, SK {IDr, [CERT,] AUTH,
                                        SAr2, TSi, TSr,
                                        N(OPTIMIZED_REKEY_SUPPORTED)}
```

- Optimized Rekey of the IKE SA

```
Initiator                           Responder
-----------------------------------------------------------------------
HDR, SK {N(OPTIMIZED_REKEY,newSPIi),
    Ni, KEi} -->
                                    <-- HDR, SK {N(OPTIMIZED_REKEY,newSPIr),
                                        Nr, KEr}
```

Note: The current SPI is from the IKE header.

- Optimized Rekey of Child SAs

```
Initiator                           Responder
-----------------------------------------------------------------------
HDR, SK {N(REKEY_SA,currentSPI), N(OPTIMIZED_REKEY,newSPIi),
    Ni, [KEi,]} -->
                                    <-- HDR, SK {N(OPTIMIZED_REKEY,newSPIr),
                                        Nr, [KEr,]}
```

Updates from -08 to -10

- Typos fixed:
  - TS payloads are misspelled as TA payloads.

IETF 113 IPsecME report

[saag] IPsecME WG report for SAAG

Tero Kivinen <kivinen@iki.fi> | Wed, 23 March 2022 13:08 UTC | Show header

IPsecME will be meeting on Friday after the saag so I updated the status on the datatracker to describe the current status (https://datatracker.ietf.org/group/ipsecme/about/status/).

Intermediate draft is now approved by the IESG and is now in the RFC Editor queue. Publication has been requested for IPTFS drafts (base draft, and yang and mib drafts), and the TCP Ecnapsulation (rfc8229bis) draft. Labeled IPsec and Deprecation of IKEv1 and obsoleted algorithms drafts are ready for publication and will be submitted to the IESG immediately after this IETF. Multiple Key Exchanges draft should also be ready for publication.

Group Key Management using IKEv2 has received some reviews during the WGLC, and should be ready for publication now. IKEv2 configuration for Encrypted DNS and Announcing Supported Authentication Methods in IKEv2 drafts are adopted as WG drafts.

There has been some work on the Optional SA & TS Payload in Child Exchange, and it might be ready to be adopted as WG draft.

There has not been that much happening with other new work, like modifying the base IKEv2 payload format, both to make it more compact for constrained devices, and allow it to go over 64kB payload limit.

IETF 114 IPsecME report

[saag] IETF 114 IPsecME report

Tero Kivinen <kivinen@iki.fi> | Tue, 26 July 2022 02:23 UTC | Show header

This status has also be uploaded to the datatracker at
https://datatracker.ietf.org/group/ipsecme/about/status/

Intermediate draft published as RFC 9242. Publication has been requested for IPTFS (base draft, and yang and mib drafts), TCP Encapsulation (rfc8229bis), multiple ke, and deprecation of IKEv1 and obsolete algorithms drafts. Labeled IPsec is ready for publication and will be submitted to the IESG immediately after this IETF.

Group Key Management still would benefit from more reviews, but we might go forward with anyways. IKEv2 configuration for Encrypted DNS should be ready for WGLC, but Announcing Supported Authentication Methods in IKEv2 needs more reviews before that.

The Optional SA & TS Payload in Child Exchange, and multi sa performance should be ready for WG adoption calls.

Quite a lot of charter items have been finished, so we should start working on to do rechartering, and clear out old things already finished, and add some new work to the charter. Most likely try to work on charter update in November IETF meeting and do rechartering after that.

IETF 115 IPsecME report

- “must be ready for WG adoption calls”?

Next Steps

- Ask for WG adoption
  - The authors believe current version is clear and mature.
- Interop test

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- **IPsec anti-replay subspaces – Paul Ponchon**
- Traffic selector with DSCP – Daniel Migault
- MTU fragmentation – Daniel Migault

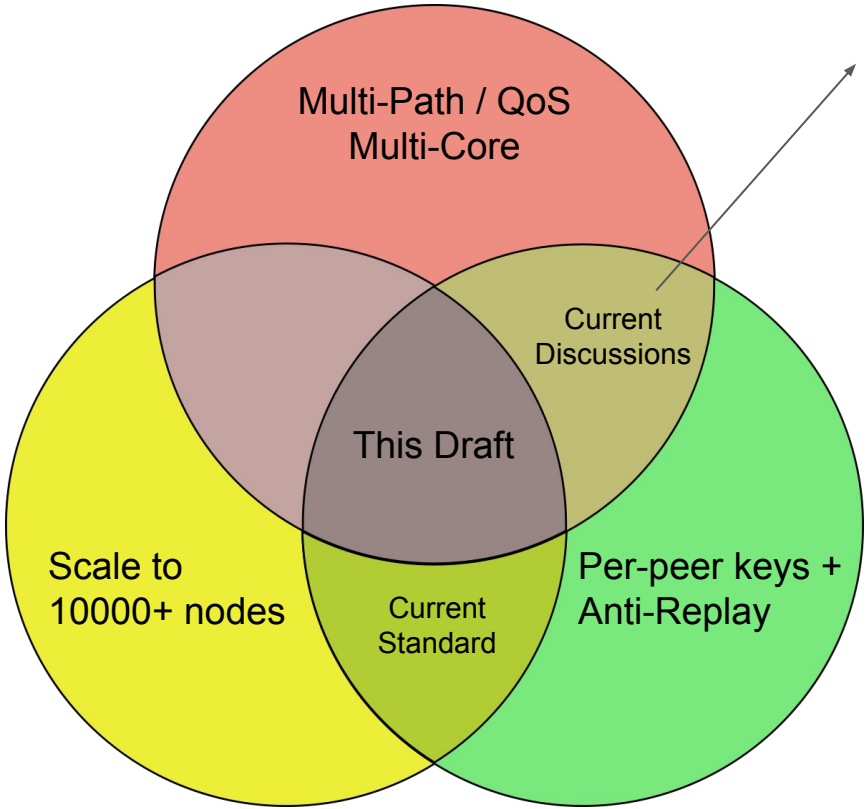
IPsec and IKE anti-replay sequence number subspaces for multi-path tunnels and multi-core processing

draft-pponchon-ipsecme-anti-replay-subspaces

Paul Ponchon (presenter), Mohsin Shaikh, Pierre Pfister, Guillaume Solignac

IETF 115 @ London

What's the problem ?



Current drafts suggest using multiple SAs

- (2019) draft-mglt-ipsecme-multiple-child-sa-00
- (2022) draft-pwouters-ipsecme-multi-sa-performance-04
- IETF 108 Presentation

But Multi-Path / QoS requirement multiplies the number of Child SAs:

- e.g., assuming 6 paths, 8 cores, 8 QoS classes: **384 Child SAs per peer**.

10.000 peers become 3.840.000 Child SAs:

- **Unnecessary load on IKE**
- **Fills CPU cache and hardware memory**

The source cause is Anti-Replay: Let’s fix it

Challenges with Anti-Replay:

- Anti-replay material cannot be efficiently shared across cores
- Multiple paths cause out-of-order packets: Packets are dropped !

Proposed solution:

- Single Child SA maintains **multiple anti-replay sequence number and vector**.
- Sender sets the subspace ID in the ESP header
- Receiver uses the subspace ID to use the appropriate anti-replay material

Discussion #1: IKE Negotiation

The draft will describe an IKE-based negotiation.

Input from the working group is welcome !

Discussion #2: subspace ID encoding

Option 1 (in current draft): 8 higher-order bits of the sequence number space

- 24 bits explicit sequence number cycles in 20 seconds at 10Gbps
- Extended Sequence Number (ESN) will be needed
- At very high speeds (e.g. 100Gbps), outages might cause resync
- More than 8 bits if ESN is made explicit

Option 2: Add a new 32 bits field between sequence number and IV

- 16 bits to be used as subspace-ID
- Remaining bits reserved for future use

Discussion #3: AES-GCM FIPS compliance

- AES-GCM for IPsec max usage count is 2^64 and enforced by RFC4106

> Moreover, as with most block cipher modes of operation, the security assurance of GCM degrades as more data is processed with a single key. Therefore, the total number of blocks of plaintext and AAD that are protected by invocations of the authenticated encryption function during the lifetime of the key should be limited. A reasonable limit for most applications would be 2^{64}, consistent with the requirement on the number of invocations in Sec. 8.3.

NIST 800-38d Appendix B

- How to make different cores share the 64 bits IV space is implementation specific
  - Only requirement is that an IV value shall only be used once.
  - A sub-field can be used to encode the sender core ID, or the subspace ID, the rest is used as a counter.

Conclusion

- We want to help the working group to solve anti-replay issues (with multi-core and multi-paths).
- Current proposed solutions rely on creating more SAs, but we believe this will show scalability limitations.

- Let’s find a common ground and work on a single solution:
  - Agree that multiplying the number of Child SAs and keymat will not scale ?
  - Allow multiple anti-replay material in the same Child SA ?
  - Encode the anti-replay subspace ID in sequence number space ? Or in a different field ?
  - Negotiate the option with IKE ?

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- **Traffic selector with DSCP – Daniel Migault**
- MTU fragmentation – Daniel Migault

Classifier for DSCP/ECN

Daniel Migault, Joel Halpern

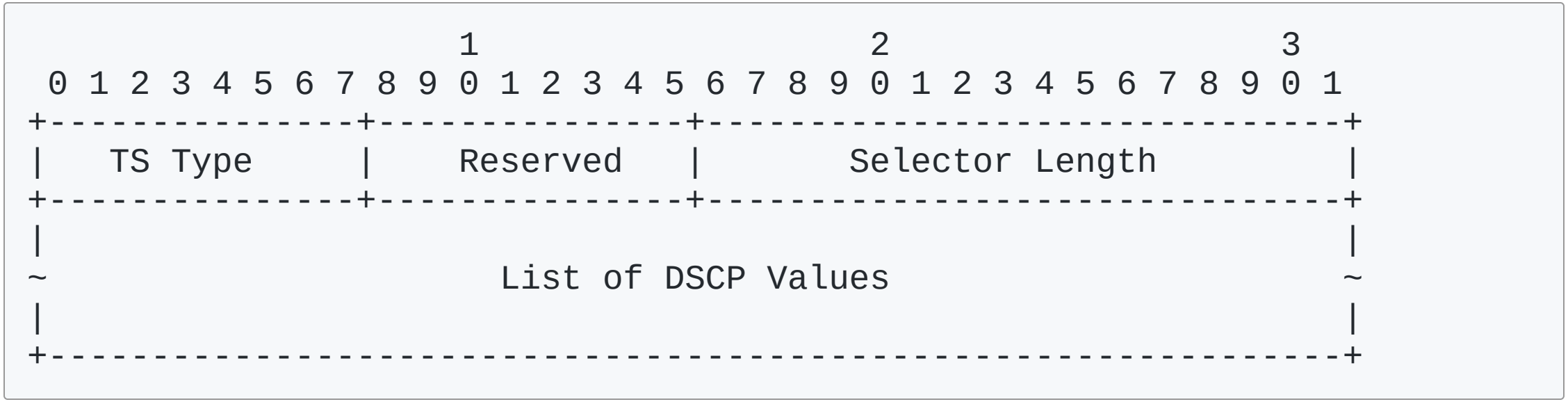
Goal: to be able to negotiate SA dedicated to a list of DSCP values.

As per RFC4301 Section 4.1,

Traffic with different DSCP value result in inappropriate discarding of lower priority packets due to the windowing mechanism used by this feature.

Although the DSCP and ECN fields are not "selectors", as that term in used in this architecture, the sender will need a mechanism to direct packets with a given (set of) DSCP values to the appropriate SA. This mechanism might be termed a "classifier".

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, TS_DSCP_LIST2 )
      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 )
```

TS_DSCP_LIST1, can be repeated or not in TSr and needs to be be mentionned only once.

TS_DSCP_LIST MUST be ignored in transport mode

If TS_DSCP_LIST is not supported a TS_UNACCEPTABLE is returned

- no fall back

DSCP values are checked against those agreed in TS

Thanks!

Presentations

- Multi-SA update – Paul Wouters
- IPsec workshow report – Steffen Klassert
- IPComp Extension – Hang Shi
- New IKEv2 payload format – Valery Smyslov
- Revised Cookie Processing – Valery Smyslov
- Inter-domain source address validation using RPKI and IPsec – Yangfei Guo
- IKEv2 Optional SA&TS Payloads in Child Exchange – Wei Pan
- IPsec anti-replay subspaces – Paul Ponchon
- Traffic selector with DSCP – Daniel Migault
- **MTU fragmentation – Daniel Migault**

Slides missing

Authors have not submitted drafts for this presentation.

Open Discussion

- Other points of interest?