IP Security Maintenance and Extensions (IPsecME) WG

IETF 118, Thursday, November 9th, 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/ietf118/?group=ipsecme&short=&item=1

Notes: https://notes.ietf.org/notes-ietf-118-ipsecme

Agenda

- Note Well, technical difficulties and agenda bashing – Chairs (5 min) (13:00-13:05)
- Document Status – Chairs (5 min) (15:05-13:10)
- Adoption calls – Chairs (5 min) (13:10-13:15)
- Presentations
  - Issues with DH with IKEv2 and rekeys – Paul Wouters (15 min) (13:15-13:30)
  - Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov (10 min) (13:30-13:40)
  - Update of multiple sequence counters – Steffen Klassert (10 min) (13:40-13:50)
  - Anti-replay sequence number subspaces – Pierre Pfister (10 min) (13:50-14:00)
  - Beet mode – Antony Antony (5 min) (14:00-14:10)
  - ESP trailer adjustment – Wei Pan (5 min) (14:10-14:15)
  - Delete info – Paul Wouters (5 min) (14:15-14:20)
  - RISAV Update – Yangfei Gui (5 min) (14:20-14:25)
- AOB + Open Mic (5 min) (14:25-14:30)

WG Status Report

- Published as RFCs
  - Labeled IPsec Traffic Selector Support for the Internet Key Exchange Protocol Version 2 (IKEv2) RFC9478
- RFF Editor queue:
  - draft-ietf-ipsecme-add-ike
- Publication requested:
  - draft-ietf-ipsecme-ikev2-auth-announce

WG Status Report

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

WG Adoption calls

- draft-liu-ipsecme-ikev2-mtu-dect
- draft-mglt-ipsecme-dscp-np
- draft-mglt-ipsecme-diet-esp
- draft-mglt-ipsecme-ikev2-diet-esp-extension
- draft-smyslov-ipsecme-ikev2-cookie-revised

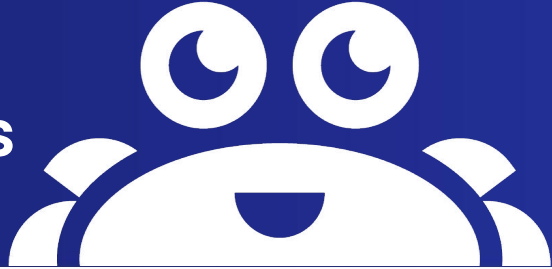
Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- Update of multiple sequence counters – Steffen Klassert
- Anti-replay sequence number subspaces – Pierre Pfister
- Beet mode – Antony Antony
- ESP trailer adjustment – Wei Pan
- Delete info – Paul Wouters
- RISAV Update – Yangfei Gui

Presentations

- **Issues with DH with IKEv2 and rekeys – Paul Wouters**
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- Update of multiple sequence counters – Steffen Klassert
- Anti-replay sequence number subspaces – Pierre Pfister
- Beet mode – Antony Antony
- ESP trailer adjustment – Wei Pan
- Delete info – Paul Wouters
- RISAV Update – Yangfei Gui

~~DH group~~ Key Exchange Method issues



- Initial Exchanges create a Child SA with DH of IKE SA setting.
  - No consistent behaviour if configuration IKE DH != Child DH
- Neither end knows if a rekey will require PFS.
  - and if so, which DH to use.

- Cause: IKE_AUTH should have contained Child DH proposal.
- Problem: Humans keep using different DH for IKE and ESP.
- Result: IPsec connection works, then fails hour(s) later.....

DH Issues: Initial Exchanges

- If negotiated IKE DH is not a valid configured Child DH:
  - On Responder:
    - If pfs=no, no issues
    - If pfs=yes, return NO_PROPOSAL_CHOSEN?
    - If pfs=yes, assume peer has same Child DH?
  - On Initiator:
    - if pfs=no, no issues
    - if pfs=yes, refuse configuration to load ?
    - if pfs=yes, send Informational Delete after negotiation ?
    - if pfs=yes, assume peer has same Child DH? (problematic)

DH Issues: REKEY of Initial Child

- Peer's configured child DH group(s) has not been negotiated yet and is unknown
- Rekey with pfs SHOULD use same DH group
- If rekey proposed Child DH is not IKE SA DH:
  - On Responder:
    - Return INVALID_KE(dh) ? [wrong, rekey DH group matches KE payload]
    - Return NO_PROPOSAL_CHOSEN ? [right, but too confusing]
    - Return another new error code? UNEXPECTED_DH_KE or something?
    - Accept any/none DH, immediately rekey IKE SA to gain pfs with IKE DH

DH Issues: Additional complications

- Microsoft Windows IKEv2 configured for DH14, rekeys with DH2
  - Probably thinks DH14 is for IKE SA, libreswan has ms-dh-downgrade=yes|no
- draft-ietf-ipsecme-ikev2-sa-ts-payloads-opt does not know PFS / DH settings
- Implementation interop issues with DH/KE transform “DH_NONE” vs no transform
- Implementation interop issues with initiator rekey using configured parameters instead of established parameters

Libreswan interoperability issues

transtions are:
AUTH -> REKEY -> REKEY
CHILD -> REKEY

esp	pfs	ms dh	w/o	AUTH CHILD	mainline REKEY AUTH CHILD	v4.5 REKEY AUTH CHILD	v4.6 REKEY AUTH CHILD	second CHILD	REKEY second CHILD or REKEY
algs	no	no	no	algs	OLD-none	algs-none	OLD-none	algs-none	OLD-none
algs	no	no	yes	algs	algs-none			algs-none	algs-none
algs	no	yes	no	algs	OLD-none	algs-none	OLD-none	algs-none	OLD-none
algs	no	yes	yes	algs	algs-none			algs-none	algs-none
algs	yes	no	no	algs	OLD-IKE	algs-IKE	OLD-IKE	algs-IKE	OLD-OLD
algs	yes	no	yes	algs	algs-IKE			algs-IKE	algs-OLD
algs	yes	yes	no	algs	OLD-IKE, OLD-none	algs-IKE algs-none	OLD-IKE algs-none	algs-IKE, algs-none	OLD-OLD, OLD-none
algs	yes	yes	yes	algs	algs-IKE, algs-none			algs-IKE, algs-none	algs-OLD, algs-none
algs-dh	no	no	no						
algs-dh	no	no	yes						
algs-dh	no	yes	no						
algs-dh	no	yes	yes						
algs-dh	yes	no	no	algs	algs-dh -or- OLD-IKE	algs-dh	OLD-IKE	algs-dh	OLD-OLD #1347
algs-dh	yes	no	yes	algs	algs-dh -or- algs-dh OLD-IKE			algs-dh	algs-dh
algs-dh	yes	yes	no	algs	algs-dh, algs-none -or- OLD-IKE, OLD-none	algs-dh, algs-none	OLD-IKE, OLD-none	algs-dh, algs-none	OLD-OLD, OLD-none
algs-dh	yes	yes	yes	algs	algs-dh, algs-none -or- algs-dh OLD-IKE algs-none			algs-dh, algs-none	algs-dh, algs-none

Possible solutions

- Disallow IKE DH != Child DH
  - does not fix install base, but will reduce problem over time.
  - Authors of RFC8247 already tried to suggest this to WG at the time :-)
- On responder, if Initial Exchange IKE DH != Child, return NO_PROPOSAL_CHOSEN
  - and use the Childless IKE SA to CREATE_CHILD_SA Child SA with proper DH
    - Causes race conditions and/or interoperability issues
- In Initial Exchanges, add DH to proposals if pfs=yes (will prob break things)
- On responder, do IKE rekey if Child DH insufficient (doesn't help initiator case)
- In IKE_AUTH, exchange a new CHILD_SA_KE(dh,..) notify
    - Return INVALID_KE / NO_PROPOSAL_CHOSEN if new notify mismatched

Questions for the IPsecME WG

- Q1: Is a new Notify CHILD_SA_KE(dh,...) worth publishing ?
- Q2: Is a new Notify Error code useful ?
- Q2: Is it useful to write up a “DH behaviour updates” doc, updating RFC 7296 ?
- Q3: Did we miss additional issues or other obvious solutions ?

Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- **Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov**
- Update of multiple sequence counters – Steffen Klassert
- Anti-replay sequence number subspaces – Pierre Pfister
- Beet mode – Antony Antony
- ESP trailer adjustment – Wei Pan
- Delete info – Paul Wouters
- RISAV Update – Yangfei Gui

Alternative Approach for Mixing PPKs in IKEv2: Update

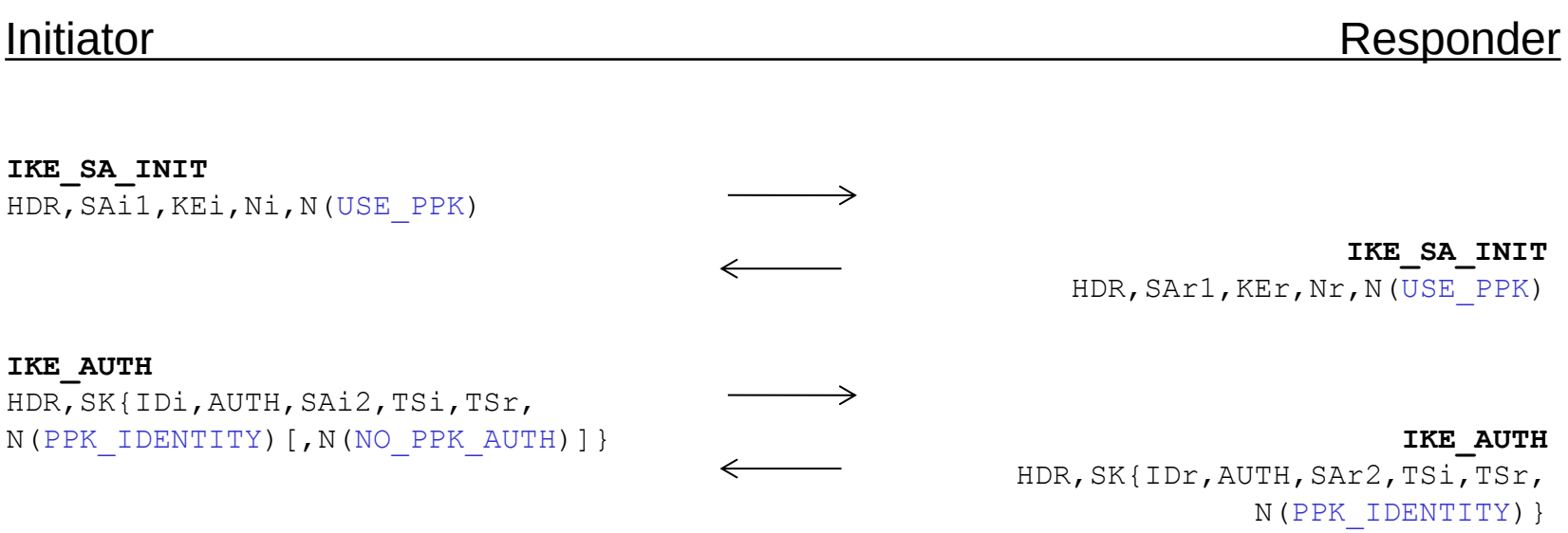
draft-smyslov-ipsecme-ikev2-qr-alt-09

Valery Smyslov
svan@elvis.ru

IETF 118

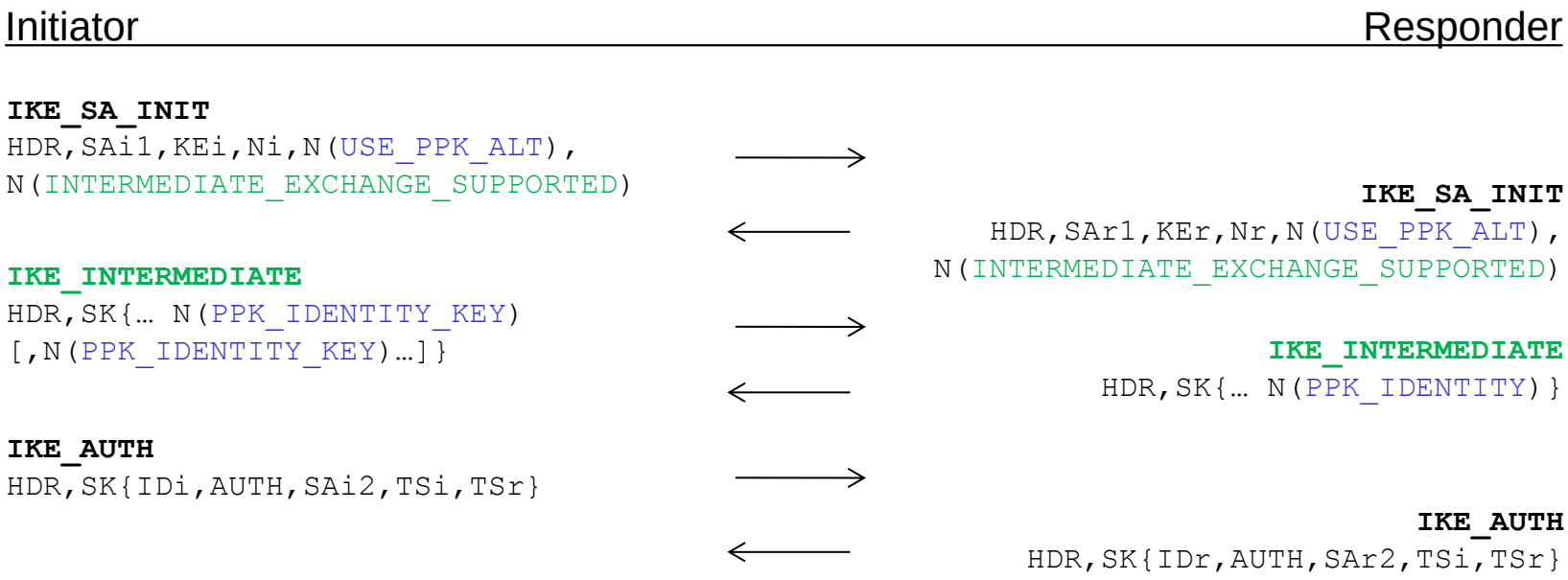
PPKs for IKEv2

Defined in RFC 8784:



Alternative Approach for PPKs in IKEv2

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



Comparison

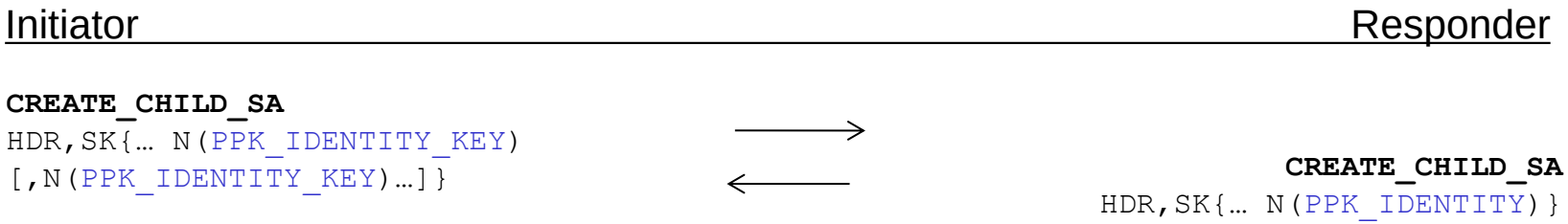
- Initial IKE SA is protected with PPK
- 3 IKE exchanges instead of 2
  - PPK_IDENTITY_KEY notification can be piggybacked if IKE_INTERMEDIATE is also used for other purposes
- 1 computation of AUTH instead of 2 if PPK is optional
- Initiator can propose several PPK_IDs
  - for each proposed PPK_ID a call to prf is required

New in -09

- Negotiation of alternative approach is now explicit and independent from RFC 8784
  - more clear protocol
  - initiator MAY propose both 8784 and alternative approach, responder MUST select one
- Use of PPKs in CREATE_CHILD_SA is defined
  - if PPK is changed before IKE SA expires (e.g. as result of QKD), then with RFC 8784 there is no way to use it other than delete IKE SA and re-create it from scratch
  - use PPKs in CREATE_CHILD_SA solves this problem (fresh PPK will only be used for confidentiality, not for authentication)

PPKs in CREATE_CHILD_SA

- Can be used for:
  - rekey IKE SA
  - rekey Child SAs
  - create additional Child SAs
- Diagram below omits content of CREATE_CHILD_SA messages for brevity



Session Keys Calculation

- For initial IKE SA

```
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)
```

- In CREATE_CHILD_SA

```
SK_d` = prf+ (PPK, SK_d)
```

  - Rekey IKE SA:

```
SKEYSEED = prf (SK_d`, g^ir (new) | Ni | Nr)
```

  - Rekey/create Child SA

```
KEYMAT = prf+ (SK_d`, [g^ir (new)] | Ni | Nr)
```

Thanks

WG adoption?

Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- **Update of multiple sequence counters – Steffen Klassert**
- Anti-replay sequence number subspaces – Pierre Pfister
- Beet mode – Antony Antony
- ESP trailer adjustment – Wei Pan
- Delete info – Paul Wouters
- RISAV Update – Yangfei Gui

ESP Problem Statement

draft-mrossberg-ipsecme-multiple-sequence-counters-01

Steffen Klassert

ESP problems in todays networks

- Replay protection and packet reordering
  - Lot of proposals to fix this
  - IETF 117
- Header and Trailer format
  - Might not fit anymore to all todays usecases
  - TODAY

Header and Trailer format

Problematic scenarios

- High-Speed Links
  - Header and trailer may and up in different cachelines
  - ESP encrypted payload alignment too short
- Software-Defined Networking (SDN)
  - Uses information from the inner transport header (ESP encrypted)

Possible solutions (High-Speed Links)

- Header and trailer may and up in different cachelines
  - Move trailer fields to the header
- ESP encrypted payload alignment too short
  - Adjusting alignment requirements to 16 or 32-byte (SMID, AVX)

Move trailer fields to the header

- Advantages:
  - Software packet processing benefits from cache locality
  - Parsing is simpler, no variable-length payload in between
- Disadvantages:
  - Larger change to the existing packet layout

Adjusting the alignment requirements

- Advantages:
  - SMID and AVX instructions operate faster
- Disadvantages:
  - Trailer might still not be aligned for SMID or AVX
  - Packets require more padding to align the trailer

» **Usefull if the trailer is removed too.**

Possible solutions (SDN)

- Uses information from the inner transport header (ESP encrypted)
  - Use an encryption offset
  - Move the ESP header between transport header and payload

Use an encryption offset

- Advantages:
  - Enables SDN usecases
  - Is optional (offset zero means encrypt everyting)
- Disadvantages:
  - Intermediate devices need to implement ESP to parse the header.

» **Google PSP uses that approach!**

Move ESP between transport header and payload

- Advantages:
  - Transparent for intermediate devices
- Disadvantages:
  - Significant change due to layering violation

» **Not recommended!**

Which way to solve the problems?

- Adjust the ESP protocol
- Define a new protocol
- Update the WESP protocol (not considered in the draft)

Adjust the ESP protocol

- Works to fix the sequence number problems
  - Some header fields interpreted diffenently
  - No change to header tailer format
- Problematic when changing header tailer format
  - No new protocol number
    - New version needs to be negotiated
  - Not transparent for middleboxes
    - SDN case does not work

Define a new protocol

- Works for sequence number problems and format change
- Maybe Google PSP + sequence number field
- Most invasive change
- Most flexible change

Update WESP protocol

- Works for sequence number problems and format change
- Not widely used
- But abuse of the original intend of WESP
- Has a version number field (2-bit)
    - Current version is 0
    - Header/trailer can be adjusted
    - Transparent to middle boxes
        - SDN case will work

Questions, suggestions?
WG adoption?

Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- Update of multiple sequence counters – Steffen Klassert
- **Anti-replay sequence number subspaces – Pierre Pfister**
- Beet mode – Antony Antony
- ESP trailer adjustment – Wei Pan
- Delete info – Paul Wouters
- RISAV Update – Yangfei Gui

IPsec and IKE anti-replay subspaces

draft-ponchon-ipsecme-anti-replay-subspaces-03

Paul Ponchon, Mohsin Shaikh, Hadi Dernaika, Pierre Pfister, Guillaume Solignac

Presenting today

IPsecme - IETF 118 - Nov 9th 2023

Quick Recap

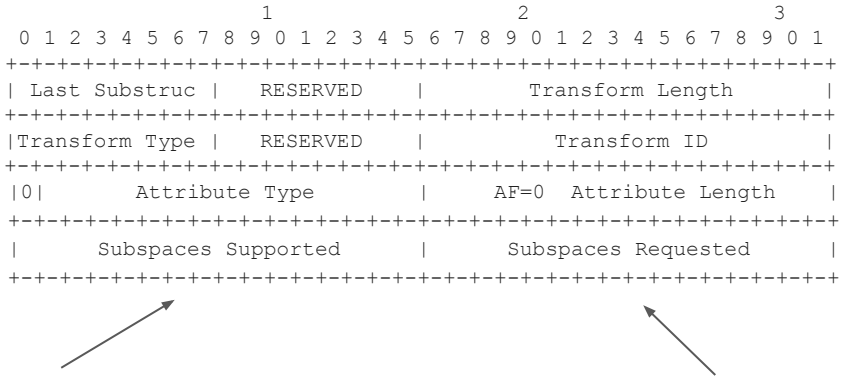
- Anti-replay protection issues with multi-path, QoS, multi-core, multicast
  - draft-mrossberg-ipsecme-multiple-sequence-counters-01

- Alternative solution is to allocate multiple SAs
  - draft-ietf-ipsecme-multi-sa-performance-02
  - Our problem is that this would create a *ton* of SAs
    - Many more keys, many more IKE messages
      - IKEv2 was designed to reduce the number of messages
      - PFS will drain resources

- Our Solution TLDR:
  - Cutting down 64 bits anti-replay sequence number into N subspaces.
  - This is an anti-replay issue, let’s fix anti-replay
    - Single SA, single RTT, all subspaces allocated at once
    - No violation of traffic selector rules. Keep change minimal in IKE.

IKE extension for Sequence-Number Subspaces (SNS) support

- Introducing the SNS Transform to ESP proposal
  - Similar to other negotiated crypto parameters, like ENCR, PRF, ESN, …



Supported inbound subspaces

Requested outbound subspaces

SNS Implementation status

- ESP with subspaces Implemented in VPP.
  - Currently working on clean-up for open-sourcing

- IKE negotiation implemented in Strongswan
  - connections.<conn>.children.<child>.esp_proposals=<esp-proposal>-snsX_Y
  - E.g. **aes128-sha256-sns16_4** indicates we support max 16 inbound subspaces and are requesting for 4 outbound subspaces.

- Meraki (closed-source) implementation
  - Targeting deployment in Cisco Meraki SD-WAN routers next year.
  - How we use subspaces:
    - **Outbound:** Per core **x** Per path (per local uplink x per remote uplink)
    - **Inbound:** Subspaces distributed across cores

Draft Status

IPRs:

- 2 IPRs disclosed with good terms
- Rumors of another patent. No-one was able to find it…

Overall discussions over last 3 IETFs

- 3 people* expressed support for adoption.
- 3 people* supportive of the work and facing similar challenges.
- 1 person expressed concerns during IETF 115, but silence since then.

How much longer are we going to have to wait to have an adoption call ?

* from different companies, excluding co-authors

Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- Update of multiple sequence counters – Steffen Klassert
- Anti-replay sequence number subspaces – Pierre Pfister
- **Beet mode – Antony Antony**
- ESP trailer adjustment – Wei Pan
- Delete info – Paul Wouters
- RISAV Update – Yangfei Gui

IPsec BEET Mode

draft-antony-ipsecme-beet-mode

IETF 118, November 2023

Antony Antony, Steffen Klassert

secunet

Goal

- Standardize IPsec BEET Mode with IKEv2
  - This has been in use for over 10 years
  - Overlooked RFC 7402 Appendix : BEET defined

History 2003 – 2009

- IETF draft-nikander-esp-beet-mode-09 expired!
  - Then HIP working group (not in IPsecME)
- Code was accepted to Linux Kernel

  Usage increased over time.

2010 - 2015

- RFC 7402 defined BEET mode in Appendix B
  - Without IKEv2

Current Use cases

- For End-to-end tunnels BEET saves bytes.
  - About 20 bytes for IPv4 (without IPv4 options)
  - 40 bytes for IPv6
- HIP RFC 7402, RFC 5202
  - Note HIP use without IKE?
- Minimal IPsec RFC 9333
- Are there more use cases that should be covered?

Software support

- Linux initial commit 2006
  - Linux kernel sees several related fixes
- strongSwan supports using private IKE Notify
- iproute2 command line tool to setup SA
  - (ip xfrm)

Next steps

- Update ID with RFC 7402 focusing on IKEv2
  - Mobile IP use cases : keep or remove?
  - NAT Use cases : keep or remove?

Questions?

- Are there any other use cases of BEET mode?
- Any other BEET mode issue to address at IETF?

BEET Pseudo-Header(PH)
esp → nextheader =94?

- Only used for IPv4 with Options or fragments.

- Linux Heders: include/uapi/linux/in.h

IPPROTO_BEETPH = **94** /* IP option pseudo header for BEET */

IPPROTO_IPIP = 4 /* IPIP tunnels (older KA9Q tunnels use 94). */

- IANA Protocol Numbers

94 IPIP IP-within-IP Encapsulation Protocol

4 IPv4 IPv4 encapsulation [RFC2003]

Backup : transport mode fragments

RFC 4301 Section 4.1

“Note: AH and ESP cannot be applied using transport mode

to IPv4 packets that are fragments. Only tunnel mode can be employed in such cases. For IPv6, it would be feasible to carry a plaintext fragment on a transport mode SA; however, for simplicity, this restriction also applies to IPv6 packets.”

Backup : IANA : 94

https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

Decimal, Keyword,Protocol,IPv6 Extension Header, Reference

94, IPIP,IP-within-IP Encapsulation Protocol, “[John_Ioannidis]”
4,IPv4 IPv4 encapsulation, [RFC2003],

[John_Ioannidis] John Ioannidis mailto:ji&tla.org 2015-01-06

Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- Update of multiple sequence counters – Steffen Klassert
- Anti-replay sequence number subspaces – Pierre Pfister
- Beet mode – Antony Antony
- **ESP trailer adjustment – Wei Pan**
- Delete info – Paul Wouters
- RISAV Update – Yangfei Gui

Considerations for Adjustments of ESP Trailer

draft-pan-ipsecme-esp-trailer-adjustment

Wei Pan
Chenyuan Fang

IETF 118

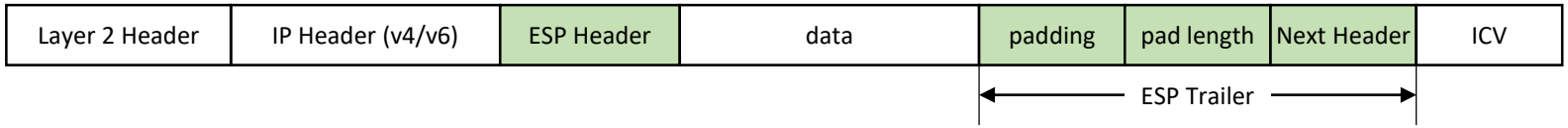
November 2023

Motivation

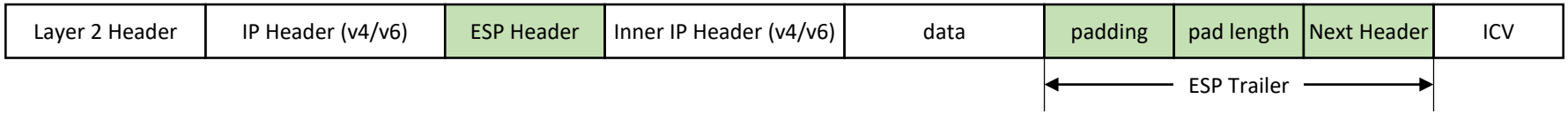
- To improve IPsec performance:
  - efficient algorithms, such as AES-GCM
  - cryptographic hardware acceleration
  - …
- Still not enough for high traffic bandwidth scenarios:
  - E.g., the **traffic between data centers can be Tbps or even higher**
- What can be considered?
  - MACsec [IEEE 802.1AE] can reach the line rate
    - The magic is that it's totally implemented by hardware (not only the encryption/decryption operations)
  - **So, implementing the whole IPsec by hardware too**

Problem Statement

- Current ESP packet format
  - Transport mode



  - Tunnel mode



- **`Next Header`** field decides how to reset the "next header" related fields in the L2 or IP header. But, it's at the end of the packet...
- The chip must cache the packet before getting the Next Header field
  - **DECRYPT THEN TRANSMIT** cannot be achieved
  - **More chip area** is needed to implement caching
    - **More chip area means more energy consumption, which is not eco-friendly**

Possible Solution 1

- Super high IPsec performance is only needed at scenarios like data centers, and these scenarios usually use ESP tunnel mode.

- A solution for ESP tunnel mode: **Judge the type of inner IP header according to its first byte**
  - In ESP tunnel mode, it's an IP packet encapsulated after ESP header.
  - The first byte of IPv4 header or IPv6 header indicates the IP version, 4 for IPv4 and 6 for IPv6.

- **Advantage**
  - Easy to implement

- **Disadvantage**
  - Only ESP tunnel mode can be supported
  - Dummy Packet function cannot be supported

Possible Solution 2

- A solution for both mode: **Move the ESP trailer immediately after ESP Header**
- **Advantage**
  - Both ESP transport and tunnel modes are supported
- **Disadvantage**
  - Significant changes to ESP protocol

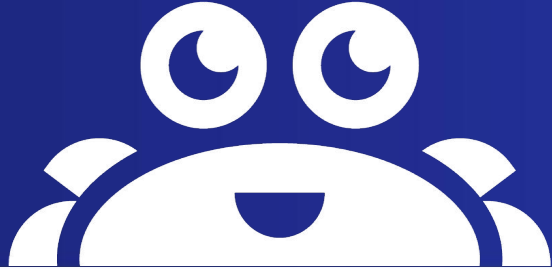
Further Considerations

- What's the reason of putting ESP trailer at the end of the packet?
- Is this problem worth solving?
  - What solution is more reasonable?
  - Any other solution?

Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- Update of multiple sequence counters – Steffen Klassert
- Anti-replay sequence number subspaces – Pierre Pfister
- Beet mode – Antony Antony
- ESP trailer adjustment – Wei Pan
- **Delete info – Paul Wouters**
- RISAV Update – Yangfei Gui

draft-pwouters-ipsecme-delete-info



- Request from customers to send and receive a signal about why an IPsec connection is going down

DELETE_REASON Notify Status Message Payload

```
                     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 Payload  !C!  RESERVED     !         Payload Length        !
+---------------+---------------+-------------------------------+
!  Protocol ID  !   SPI Size     !      Notify Message Type       !
+---------------+---------------+-------------------------------+
!            Downtime           !                               !
+-------------------------------+  Reason Message               ~
~                                                               !
+-------------------------------+-------------------------------+
```

draft-pwouters-ipsecme-delete-info

- Is the two octet “seconds” useful or over-engineered?
- Free form text or IANA registry with “reasons” ?
  - Why not both?
- Currently listed example reasons:
  - SERVICE_SHUTDOWN, SERVICE_RESTART
  - HOST_SHUTDOWN, HOST_RESTART
  - CONFIGURATION_CHILD_REMOVED
  - CONFIGURATION_IKE_REMOVED
  - ADMINISTRATIVELY_DOWN
  - IDLE_TIMEOUT
  - INITIAL_CONTACT_REPLACED
  - SIMULTANEOUS_REKEY
  - RE_AUTHENTICATED
  - REDIRECTION_ACCEPTED
  - LIFETIME_EXCEEDED

Presentations

- Issues with DH with IKEv2 and rekeys – Paul Wouters
- Alternative Approach for Mixing PPKs in IKEv2 – Valery Smyslov
- Update of multiple sequence counters – Steffen Klassert
- Anti-replay sequence number subspaces – Pierre Pfister
- Beet mode – Antony Antony
- ESP trailer adjustment – Wei Pan
- Delete info – Paul Wouters
- **RISAV Update – Yangfei Gui**

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

Ke Xu, Jianping Wu, Yangfei Guo, Benjamin M. Schwartz, Haiyang Wang

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

IETF 118 Nov. 2023

What RISAV is and How it works

WHAT

An approach for inter-AS Source Address Validation with RPKI and IPSEC.

HOW

RPKI

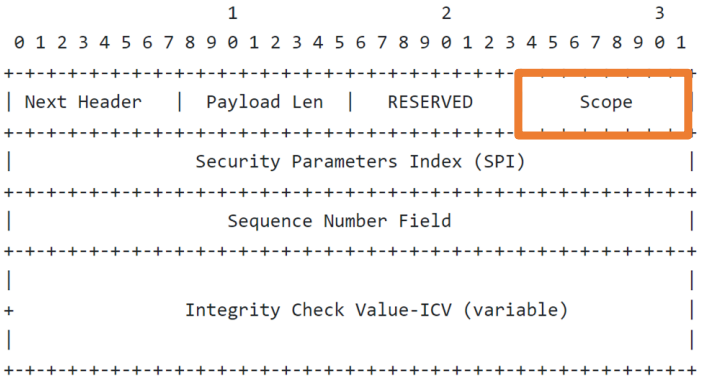
1. Define a config format for Contact IP and ASID.
2. Each participant publishes their config in the RPKI database.
3. All participants sync the RPKI database as usual.

IPSEC

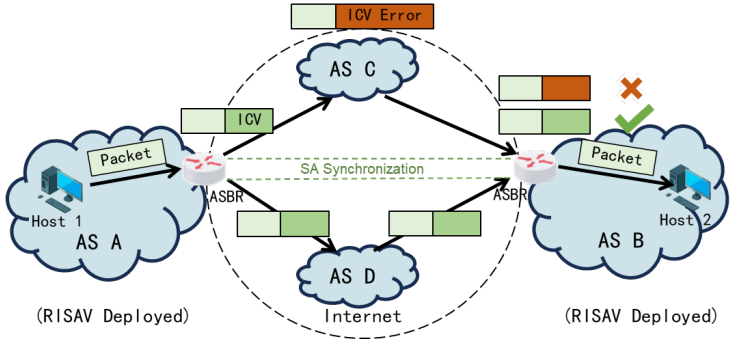
4. Each participant connects to all the other participants by IKE.
5. Data plane communicates with IPsec encapsulation.

COST: $O(N^2)$ IPsec associations with $O(N)$ human work.

Changes to IPsec



- AH
  - Reserved(2B) => Reserved(1B) + Scope(1B)
  - Scope identifies the scope of protection for RISAV AH.
    - 0 for IP and 1 for AS;
    - others are not defined
  - Only used for AS-to-AS communication
- IKE
  - Only indexed by SPI and counterpart ASN regardless of src IP or dst IP in SAD
- Transparent to the end hosts.



- ESP
  - Encryption
    - ESP Encryption
    - ESP_NULL Encryption
  - Tunnel is built with current ASBR and ACS's(AS Control Server) contact IP of another AS
- IKE
  - 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

Closing remarks

- *Changes are acceptable?*

- RISAV treats the Internet as a true “network of networks”.
- RISAV provides clear benefits for participants even when only fractionally deployed.
  - e.g. if x% of networks have RISAV, joining RISAV reduces your amplification-reflection attack volume by x% (on average).
- The design has been getting simpler as other IPsec drafts propose solutions to key protocol scalability challenges.
- Suggestions are welcomed.

Thanks

Open Discussion

- Other points of interest?