IP Security Maintenance and Extensions (IPsecME) WG

IETF 119, Tuesday, March 19th, 2024

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/ietf119/?session=32044
Notes: https://notes.ietf.org/notes-ietf-119-ipsecme
Agenda

• Note Well, technical difficulties and agenda bashing – Chairs (5 min) (15:30-15:35)
• Document Status – Chairs (5 min) (15:35-15:40)
• Presentations
  – Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2 – Scott Fluhrer (5 min) (15:40-15:45)
  – ESP Echo Protocol – Jen Linkova (15 min) (15:45-16:00)
  – Shared Use of IPsec Tunnel in a Multi-VPN Environment – Wei Pan (10 min) (16:00-16:10)
  – IKEv2 Support for Anti-Replay Status Notification – Wei Pan (10 min) (16:10-16:20)
  – Using ShangMi in the IKEv2 – Frank (Liang) XIA (10 min) (16:20-16:30)
  – IKEv2 IPv4 Downstream Fragmentation – Daniel Migault (10 min) (16:30-16:40)
  – IKEv2 DSCP Notification – Daniel Migault (10 min) (16:40-16:50)
• AOB + Open Mic (10 min) (16:50-17:00)
WG Status Report

- Published as RFCs
  - Internet Key Exchange Protocol Version 2 (IKEv2) Configuration for Encrypted DNS RFC9464

- IETF Last Call:
  - draft-ietf-ipsecme-ikev2-auth-announce

- Almost Publication Requested (waiting for document update):
  - draft-ietf-ipsecme-multi-sa-performance
WG Status Report

• Waiting for write-up / AD Followup:
  - draft-ietf-ipsecme-g-ikev2

• Work in progress:
  - draft-ietf-ipsecme-ikev2-sa-ts-payloads-opt
  - draft-smyslov-ipsecme-ikev2-qr-alt
  - draft-mglt-ipsecme-ikev2-diet-esp-extension
  - draft-mglt-ipsecme-diet-esp

• Expired:
  - draft-ietf-ipsecme-ike-tcp
Presentations

- Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer
- ESP Echo Protocol
  Jen Linkova
- Shared Use of IPsec Tunnel in a Multi-VPN Environment
  Wei Pan
- IKEv2 Support for Anti-Replay Status Notification
  Wei Pan
- Using ShangMi in the IKEv2
  Frank (Liang) XIA
- IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault
- IKEv2 DSCP Notification
  Daniel Migault
Presentations

- Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer
- ESP Echo Protocol
  Jen Linkova
- Shared Use of IPsec Tunnel in a Multi-VPN Environment
  Wei Pan
- IKEv2 Support for Anti-Replay Status Notification
  Wei Pan
- Using ShangMi in the IKEv2
  Frank (Liang) XIA
- IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault
- IKEv2 DSCP Notification
  Daniel Migault
draft-kampanakis-ml-kem-ikev2-03
Summary

• Introduces soon to be standardized, quantum-resistant ML-KEM to IKEv2 for PQ-hybrid and potentially PQ-pure key exchanges.
• Uses RFC9370. Does not define any new mechanisms.
• Requests “Transform Type 4 - Key Exchange Method Transform IDs” IANA codepoints
  • TBD36 for ML-KEM-768
  • TBD37 for ML-KEM-1024
Some details

• ML-KEM can be used in
  • IKE_INTERMEDIATE (RFC9242)
  • IKE_FOLLOWUP KE (RFC9370)
  • IKE_SA_INIT
    • ML-KEM-512 would fit in one 1460B UDP packet.
    • ML-KEM-768, barely...
    • ML-KEM-768 is a SHOULD NOT.

• Keying material is derived as per RFC9370
  • SK_d, SK_a[i/r], and SK_e[i/r]
  • SKEYSEED and KEYMAT
Open questions

• Should we get a codepoint for ML-KEM-512?
  • It would fit in a IKE_SA_INIT
  • It could be used in a PQ-pure key exchange.
  • ML-KEM-512 has lower security level (close or slightly less than 128-bits depending on analysis). TLS converged towards using ML-KEM-768 to be more conservative regarding security level.
Path Forward?

• IPSECME WG draft or
• I-D to get codepoints from Expert Review as per IANA “Transform Type 4 - Key Exchange Method Transform IDs” ’s Registration Procedure.
Thank you
Presentations

• Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer

• ESP Echo Protocol
  Jen Linkova

• Shared Use of IPsec Tunnel in a Multi-VPN Environment
  Wei Pan

• IKEv2 Support for Anti-Replay Status Notification
  Wei Pan

• Using ShangMi in the IKEv2
  Frank (Liang) XIA

• IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault

• IKEv2 DSCP Notification
  Daniel Migault
ESP, IPv4 and IPv6

IPv4

| Protocol = 17 | UDP | Dst port = 4500 | ESP |

IPv6

| Next Header 50 | ESP |

Native ESP advantages:

- no keepalives needed (if ESP is statelessly allowed)
- fewer keepalives otherwise (ESP timeouts usually higher)
Problem Statement

ESP packets do not share fate with IKE
IKE might succeed but ESP packets are dropped
Hard to detect and recover
Data traffic is blackholed
Solution Overview

The node MAY send an IPv6 ESP Echo Request packet:

- SPI = 7, Next Header = 59

The peer SHOULD respond with an ESP Echo Reply packet:

- SPI = 8, Next Header = 59
- MUST copy the data from Echo Request up to the MTU
Payload Format (Based on Section 4 RFC4443)

Identifier: Specific to the ESP ping session. Expected to be randomized.

Sequence number: allows matching Echo Requests and Echo Response

Data: May or may not be zero.
Changes to RFC4303: Processing Next Header = 59

RFC4303 “Dummy” packets:
- used for padding
- ESP packets with next header == 59 (no next header)
- A transmitter MUST be capable of generating dummy packets
- a receiver MUST be prepared to discard such packets

Proposed changes:

If a packet with next header == 59 has SPI 7 or 8, then it’s an ESP Echo packet and shall be processed as described in this document
Non-Reserved ("Production") SPIs

The proposed mechanism could be extended to "real" SPIs: how would that look?

- Specifically, a next-header of 59 would elicit a response in the paired IPsec SA?
- A Next-Header of ipv6-icmp (58) would elicit a response in the paired IPsec SA?

(do kernels even know what the associated paired IPsec SA is?)

Benefits:

- Preventing MitM attacks (replies are authenticated)
- Fate sharing with actual data flows

However, such packets are indistinguishable from “dummy” packets.
Security Considerations

The node MUST NOT fall back to unencrypted mode of communication in case of ESP Echo failure

Preventing a downgrade attack

ESP Echo Request can be used to discover IPsec speaker

....but so can be IKE INIT
Open Questions: Discovering ESP Echo Support

- Explicit (out-of-band) signal
  - E.g. a corporate VPN client is configured to use ESP Echo when connecting to the corporate servers
- [not in the draft] Announcing ESP Echo support in IKEv2
  - Is it needed?
  - Worth complexity?
Questions? Comments?
Adoption?
Open Questions: Ways to do ESP Ping within “production” SAs

- An IKEv2 Notify could be created that indicates that the sender is willing to accept ESP packets with {tunnel} mode ICMP(v6) messages.
  - The Notify would indicate an address to which an ICMPv6 message can be addressed that is willing to answer.

Do we need to also indicate a valid source address?
Presentations

- Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer
- ESP Echo Protocol
  Jen Linkova
- **Shared Use of IPsec Tunnel in a Multi-VPN Environment**
  Wei Pan
- IKEv2 Support for Anti-Replay Status Notification
  Wei Pan
- Using ShangMi in the IKEv2
  Frank (Liang) XIA
- IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault
- IKEv2 DSCP Notification
  Daniel Migault
Background

• Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space
• Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space

• When establishing IPsec tunnel via IKEv2 to protect the traffic of VPN 1 and VPN 2

  • If VPN 1 and VPN 2 share one IKE SA, when negotiating the creation of Child SA, the receiver can’t differentiate which VPN this Child SA should be associated with.

### Background

**Device A**

- **VPN 1** (10.1.1.0/24)
- **VPN 2** (10.1.1.0/24)

**Device B**

- **VPN 1** (11.1.1.0/24)
- **VPN 2** (11.1.1.0/24)

**IKE SA**

CREATE_CHILD_SA request

(TSi(0,0,10.1.1.0-10.1.1.255), TSr(0,0,11.1.1.0-11.1.1.255))

TS payloads can match both VPN 1 and VPN 2, but which VPN is this Child SA belonging to?
• Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space

• When establishing IPsec tunnel via IKEv2 to protect the traffic of VPN 1 and VPN 2
  • If VPN 1 and VPN 2 share one IKE SA, when negotiating the creation of Child SA, the receiver can’t differentiate which VPN this Child SA should be associated with.
  • If VPN 1 and VPN 2 separately use different IKE SAs, when negotiating the creation of Child SA, the receiver can differentiate which VPN this Child SA should be associated with.

**Background**

Device A

VPN 1 (10.1.1.0/24)
VPN 2 (10.1.1.0/24)

IKE SA 1

CREATE_CHILD_SA request

(TSi((0,0,10.1.1.0-10.1.1.255)), -->
TSr((0,0,11.1.1.0-11.1.1.255)))

This Child SA belongs to VPN 1, because its IKE SA belongs to VPN 1

Device B

VPN 1 (11.1.1.0/24)
VPN 2 (11.1.1.0/24)

IKE SA 2

CREATE_CHILD_SA request

(TSi((0,0,10.1.1.0-10.1.1.255)), -->
TSr((0,0,11.1.1.0-11.1.1.255)))

This Child SA belongs to VPN 2, because its IKE SA belongs to VPN 2
Background

- Assuming two Devices and two VPNs, and VPN 1 and VPN 2 are using the same IP address space.
- When establishing IPsec tunnel via IKEv2 to protect the traffic of VPN 1 and VPN 2:
  - If VPN 1 and VPN 2 share one IKE SA, when negotiating the creation of Child SA, the receiver can’t differentiate which VPN this Child SA should be associated with.
  - If VPN 1 and VPN 2 separately use different IKE SAs, when negotiating the creation of Child SA, the receiver can differentiate which VPN this Child SA should be associated with.
- Therefore, currently, different VPNs need different IPsec tunnels (different IKE SAs & Child SAs).
Problem Statement

- In 3GPP networks, full-meshed IPsec tunnels are established among base stations.

- Radio Access Network (RAN) Sharing is used to lease the infrastructure to other operators.

- **IPsec tunnels’ number is seriously boosted** as the number of base stations and operators sharing the RAN increases.
  - Assume there are $N$ neighbors and $M$ sharing operators, then the IKE SAs are $N \times M$ and the Child SAs are at least $N \times M$.
  - The limited SAs supported by the device restricts the development and evolution of services in this scenario.
Solution Overview

- **Core Concept:** Share the same IPsec tunnel for different VPNs, by adding VPN-related information in the creation of Child SA and the IPsec data packets

  - **Step 1**
    Negotiation of Support in IKE_SA_INIT
  
  - **Step 2**
    Correlate VPN with Child SA during its creation
  
  - **Step 3**
    Carry VPN info in the IPsec data packets

- **Current Design:**
  
  - Add the VPN attribute for each Traffic Selector when negotiating the traffic to be protected in Child SAs.
  
  - Carry the VPN info in the extended ESP and AH header to distinguish which VPN the inner packet belongs to.
Solution Step 1

- During the IKE_SA_INIT exchange, two peers negotiate the support of correlating VPN with IPsec SAs.

- Peers include the `VPN_BASED_TS_SUPPORTED` notify payload in the IKE_SA_INIT exchange request and response, to indicate the support of using new Traffic Selectors that contain the VPN ID field.

**IKE_SA_INIT Message Exchange Example**

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SAi1, KEi, Ni,</td>
<td></td>
</tr>
<tr>
<td>N(<code>VPN_BASED_TS_SUPPORTED</code>) --&gt;</td>
<td></td>
</tr>
</tbody>
</table>

|   <= HDR, SAr1, KEr, Nr, [CERTREQ,] |
|   N(`VPN_BASED_TS_SUPPORTED`)    |
Solution Step 2

- Two New Traffic Selectors are introduced: **TS_IPV4_ADDR_RANGE_VPN** and **TS_IPV6_ADDR_RANGE_VPN**
  - Compared with existing v4/v6 Traffic Selectors, these two new Traffic Selectors contain an additional "VPN ID" field.

- When creating Child SAs, two peers using these two new Traffic Selectors instead of the existing two.
  - Parsing Rule: First **pairing the Traffic Selectors with the same VPN ID** from the TSi and TSr payloads, then processing the paired Traffic Selectors.

---

**TS_IPV4_ADDR_RANGE_VPN and TS_IPV6_ADDR_RANGE_VPN Formats**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td>+---------------+-------------------------------+</td>
<td></td>
</tr>
<tr>
<td>TS Type</td>
<td>IP Protocol ID*</td>
<td>Selector Length</td>
</tr>
<tr>
<td>+---------------+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Start Port*</td>
<td>End Port*</td>
<td></td>
</tr>
<tr>
<td>+---------------+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>~ Starting Address*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>~ Ending Address*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-------------------------------+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VPN ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**CREATE_CHILD-SA Message Exchange Example**

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SK [SA, Ni, [KEi,]</td>
<td>HDR, SK [SA, Nr, [KEr,]</td>
</tr>
<tr>
<td>TSi(TS_IPV4_ADDR_RANGE_VPN),</td>
<td>TSi(TS_IPV4_ADDR_RANGE_VPN),</td>
</tr>
<tr>
<td>TSr(TS_IPV4_ADDR_RANGE_VPN)} --&gt;</td>
<td>TSr(TS_IPV4_ADDR_RANGE_VPN)} --&gt;</td>
</tr>
<tr>
<td>&lt;= HDR, SK [SA, Nr, [KEr,]</td>
<td>&lt;= HDR, SK [SA, Nr, [KEr,]</td>
</tr>
<tr>
<td>TSi(TS_IPV4_ADDR_RANGE_VPN),</td>
<td>TSi(TS_IPV4_ADDR_RANGE_VPN),</td>
</tr>
<tr>
<td>TSr(TS_IPV4_ADDR_RANGE_VPN)}</td>
<td>TSr(TS_IPV4_ADDR_RANGE_VPN)}</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SK [SA, Ni, [KEi,]</td>
<td>HDR, SK [SA, Nr, [KEr,]</td>
</tr>
<tr>
<td>TSi(TS_IPV6_ADDR_RANGE_VPN),</td>
<td>TSi(TS_IPV6_ADDR_RANGE_VPN),</td>
</tr>
<tr>
<td>TSr(TS_IPV6_ADDR_RANGE_VPN)} --&gt;</td>
<td>TSr(TS_IPV6_ADDR_RANGE_VPN)} --&gt;</td>
</tr>
<tr>
<td>&lt;= HDR, SK [SA, Nr, [KEr,]</td>
<td>&lt;= HDR, SK [SA, Nr, [KEr,]</td>
</tr>
<tr>
<td>TSi(TS_IPV6_ADDR_RANGE_VPN),</td>
<td>TSi(TS_IPV6_ADDR_RANGE_VPN),</td>
</tr>
<tr>
<td>TSr(TS_IPV6_ADDR_RANGE_VPN)}</td>
<td>TSr(TS_IPV6_ADDR_RANGE_VPN)}</td>
</tr>
</tbody>
</table>
Solution Step 3

- Extending the ESP and AH packet formats with an additional "VPN ID" field, to differentiate which VPN the inner traffic belongs to.

**ESP Format Extension Example**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------+</td>
<td>^Int.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security Parameters Index (SPI)</td>
<td>&quot;Cov-&quot;</td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sequence Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VPN ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------+</td>
<td></td>
<td>^</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payload Data* (variable)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>~</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>~</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td>&quot;Conf.&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Padding (0-255 bytes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+------------------------+------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pad Length</td>
<td>Next Header</td>
</tr>
<tr>
<td>+------------------------+------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Integrity Check Value-ICV (variable)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>~</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**AH Format Extension Example**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Next Header</td>
<td>Payload Len</td>
<td>RESERVED</td>
</tr>
<tr>
<td>+---------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security Parameters Index (SPI)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sequence Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VPN ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+---------------------------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>~</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Integrity Check Value-ICV (variable)</td>
<td>~</td>
<td></td>
</tr>
<tr>
<td>+-----------------------------------------------+</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10
Alternative Solutions

- **Splitting the 32-bit SPI into two sub-fields: the VPN ID sub-field and SPI sub-field**

  - When creating Child SAs, to set the VPN ID sub-field all zero and only use the SPI sub-field
  - When sending IPsec packets, to set the VPN ID sub-field with the actual VPN ID value that the inner traffic belongs to, and to set the SPI sub-field with the SPI value

  - **Advantage**
    - No ESP/AH packet format changes needed
  - **Disadvantage**
    - Scalable issue: 16-bit VPN ID is needed for future scenarios, then 16-bit SPI might not be sufficient.
    - Packet disorder: Different VPNs use different 32-bit SPI (composed of VPN ID and actual SPI) in the data packets, this will interfere with the load balance process of the on-path routers who look at the SPI field when doing the hash, and finally cause disorder at the receiver.

- **Using a notify or a traffic selector of just the VPN ID when creating the Child SAs**
  - Can’t differentiate which v4/v6 Traffic Selector is associated with which VPN.
  - May cause unwanted traffic to be included.
Further Considerations

- Is this problem worth solving?
- Suggestions, comments, reviews, co-authors, etc., are all welcome.
Presentations

- Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer
- ESP Echo Protocol
  Jen Linkova
- Shared Use of IPsec Tunnel in a Multi-VPN Environment
  Wei Pan
- IKEv2 Support for Anti-Replay Status Notification
  Wei Pan
- Using ShangMi in the IKEv2
  Frank (Liang) XIA
- IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault
- IKEv2 DSCP Notification
  Daniel Migault
IKEv2 Support for Anti-Replay Status Notification

draft-pan-ipsecme-anti-replay-notification

Wei Pan
Qi He
Paul Wouters

IETF 119
March 2024
RFC 4302 and RFC 4303 specify the details of sequence number generation and verification.

- When anti-replay is enabled on the receiver, the sender must monitor the sequence number counter, increment the counter with every message sent, and ensure the counter does not cycle.
- When anti-replay is disabled on the receiver, the sender does not need to monitor or reset the counter.

RFC 4302 and RFC 4303 also specify that, during SA establishment, IPsec implementation should notify the peer if it will not provide anti-replay protection, to avoid having the peer do unnecessary sequence number monitoring and SA setup.
Problems In High-Performance Scenarios

Current Situation

• In high-performance scenarios, high-level QoS packets may arrive earlier than a large number of low-level QoS packets, causing the low-level QoS packets being dropped due to disordered packets exceed the window size of anti-replay.

• Operators choose to disable the anti-replay function for reasons of QoS, performance, etc.

Result in Frequent Rekey

• In high-performance scenarios, 32-bit sequence numbers are consumed quickly, resulting in frequent rekeying of Child SAs.

• ESN solves this problem by extending the sequence numbers to 64 bits. But, ESN relies on the window of anti-replay to guess the high-order 32 bits of the sequence number.

• Currently, when the anti-replay is disabled, Child SAs will rekey frequently due to unnecessary sequence number monitoring and the unavailability of ESN. Solutions could be:
  • To avoid unnecessary sequence number monitoring, i.e., adding anti-replay status notification in IKEv2.
  • To allow the use of ESN when anti-replay is disabled.
Anti-replay status notification

- Peers include the **ANTI_REPLAY_STATUS** notify payload in the IKE_AUTH exchange for creating the initial Child SA or the CREATE_CHILD_SA exchange for creating the subsequent Child SAs.

- When anti-replay is disabled on both peers, neither peer needs to monitor the sequence number counter, thus avoiding frequent rekey of Child SAs.

IKE_AUTH Message Exchange Example

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr, <strong>N(ANTI_REPLAY_STATUS)</strong>}</td>
<td>HDR, SK {IDr, [CERT,] AUTH, SAr2, TSi, TSr, <strong>N(ANTI_REPLAY_STATUS)</strong>}</td>
</tr>
</tbody>
</table>

CREATE_CHILD_SA Message Exchange Example

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SK {SA, Ni, [KEi,] TSi, TSr, <strong>N(ANTI_REPLAY_STATUS)</strong>}</td>
<td>HDR, SK {SA, Nr, [KEr,] TSi, TSr, <strong>N(ANTI_REPLAY_STATUS)</strong>}</td>
</tr>
</tbody>
</table>

**Question:** Should this notify be able to convey “I can do ESN without replay protection”?
Unbind ESN From Anti-Replay

- ESN relies on the window of anti-replay to guess the high-order 32 bits of the sequence number.
- If IPsec implementations can maintain a separate window for ESN, then ESN can be used without anti-replay.
  - Due to the window size of anti-replay is not negotiable, the window size for ESN may also not need to be negotiable.
- Doing ESN without anti-replay seems to be an unilateral act, is there a need to notify the peer that “I can do ESN without replay-detection”?
- ESN can be used whichever peer disables the anti-replay, thus avoiding frequent rekey of Child SAs.
Further Considerations

• Is this problem worth solving?
  • Anti-replay status notification is required by RFC 4302 & 4303, and this draft can fulfill this requirement.
  • Unbinding ESN from anti-replay should be covered in this draft or done in a separate draft?

• Suggestions, comments, and reviews are all welcome.
Presentations

- Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer
- ESP Echo Protocol
  Jen Linkova
- Shared Use of IPsec Tunnel in a Multi-VPN Environment
  Wei Pan
- IKEv2 Support for Anti-Replay Status Notification
  Wei Pan
- Using ShangMi in the IKEv2
  Frank (Liang) XIA
- IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault
- IKEv2 DSCP Notification
  Daniel Migault
Using ShangMi in the Internet Key Exchange Protocol Version 2 (IKEv2)

draft-guo-ipsecme-ikev2-using-shangmi-00

Yanfei Guo, Liang Xia
Yu Fu

Huawei
China Unicom
China ShangMi Cryptography Algorithms

Background

• SM2: a set of cryptographic algorithms based on elliptic curve cryptography, including a digital signature, public key encryption (not defined in this draft for IKEv2) and key exchange scheme. ISO/IEC 14888-3:2018, GBT.32918.2-2016, GBT.32918.5-2017


• SM3: a set of block cipher encryption algorithms. 18033-3:2010, GBT.32907-2016
Proposal: IKEv2 Support ShangMi without any Protocol Change, Only Add Several New Transform Types and IDs

◆ New added Transform Types and IDs:

1. Transform Type 1 - Encryption Algorithm Transform IDs
   - ✓ ENCR_SM4_CBC  CBC
   - ✓ ENCR_SM4_GCM  AEAD
   - ✓ ENCR_SM4_CCM  AEAD

2. Transform Type 2 - Pseudorandom Function Transform IDs
   - ✓ PRF_HMAC_SM3

3. Transform Type 3 - Integrity Algorithm Transform IDs
   - ✓ AUTH_HMAC_SM3

4. Transform Type 4 - Key Exchange Method Transform IDs
   - ✓ curveSM2

5. IKEv2 Hash Algorithms
   - ✓ SM3

6. IKEv2 Authentication Method
   - ✓ SM2 Digital Signature
At the time of writing, there are no known weak keys for SM cryptographic algorithms SM2, SM3 and SM4, and no security issues have been found for these algorithms.

**cryptanalysis of SM2:**

**cryptanalysis of SM4 (especially for side-channel attacks):**
Comments from Paul Wouters (SEC AD):

"Thanks for the document. I believe the best way forward for these would be via the ISE. In which case the Working Group and Intended Status would need to be updated. But if the document proceeds that way, please keep the IPsecME WG in the loop. All the registries involved are "Expert Review", so it can be registered regardless of where or how the specification is published.

As for the draft itself, I have two questions.

Is the CBC variant really necessary? CBS is being made historic or deprecated for all other IETF uses (eg see TLS 1.3). Why introduce it now for IKEv2 and ESP in combination with ShangMi?

For the GCM variants, do you know if these can make use of the ghash hardware instructions? As in, would ENCR_SM4_GCM also benefit from CPU hardware instructions available?

"
Presentations

- Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer
- ESP Echo Protocol
  Jen Linkova
- Shared Use of IPsec Tunnel in a Multi-VPN Environment
  Wei Pan
- IKEv2 Support for Anti-Replay Status Notification
  Wei Pan
- Using ShangMi in the IKEv2
  Frank (Liang) XIA
- IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault
- IKEv2 DSCP Notification
  Daniel Migault
IKEv2 IPv4 Downstream Fragmentation Notification Extension

draft-liu-ipsecme-ikev2-mtu-dect

Liu, Zhang. Migault
The goal of the document is to limit the reassembly operations being performed by the egress security gateway. It defines:

- The IKEv2 Link Maximum Atomic Packet
- The Packet Too Big Extension
Illustrative Example:

1. Mid-tunnel (performed by a router on N) (only for IPv4 DF=0 TLP)
2. Egress node detects fragmentation

- a) it collects IPVersion the IP version of the first fragment as well as FragLen, the fragment length
- b1) If all segments can be reassembled and the reassembled packet is properly decrypted a Link Maximum Atomic Packet Notification (LMAP) is sent on the IKEv2 channel.
b2) If the packet is too big and cannot be fully processed PTB indicates LMTU of the router component of the egress node.

```
[IKEv2]
<--- N( LMAP [ IPVersion, FragLen] )
   N( PTB [LMTU, EMTU_R] )
```
3. Upon receiving the LMAP or optionally the ingress node

- a) Update the TMTU so that the Source performs source fragmentation with TTP packets that are not fragmented.

```
Source fragmentation
(IPv6 or IPv4)

+-----------------+
| IPs | IPd | Da |
+-----------------+
| IPS | IPd | ta |
+-----------------+
```

(TTP)
b) Performs inner fragmentation TTP packets that exceeds the TMTU and will generate some fragments.

Inner fragmentation (performed by the Ingress node)
(only for IPv4 DF=0 TTP)

```
+---+---+---+---+---+--+
|IPi|IPe|ESP|IPs|IPd|Da| (TLP)
+---+---+---+---+---+--+
```

```
+---+---+---+---+---+--+
|IPi|IPe|ESP|IPs|IPd|ta| (TLP)
+---+---+---+---+---+--+
```
In both cases the egress node does not proceed to reassembly operations:

<table>
<thead>
<tr>
<th>IPs</th>
<th>IPd</th>
<th>Da</th>
<th>(TTP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```plaintext
+------------+
| IPs | IPd | Da |
+------------+

+------------+
| IPs | IPd | ta |
+------------+
```
The draft has been presented and reviewed several times; we took all comments into consideration.

1. PTB discussion: The IKE PTB, in our view, is largely motivated by enabling the egress interface to provide the EMTU_R (see ietf-intarea-tunnels section 4.2.2.1.

2. TMTU:

- ietf-intarea-tunnels considers the router component - carrying the TTP - and the interface component - handling LTP - independent. Such independence between the Tunnel MTU (for TTP) and link layer MTU for (LTP) is provided by performing outer fragmentation when needed.

- RFC4301 considers the router component can adapt to the specific needs of the interface component. This is what we do here.
Presentations

• Post-quantum Hybrid Key Exchange with ML-KEM in the IKEv2
  Scott Fluhrer
• ESP Echo Protocol
  Jen Linkova
• Shared Use of IPsec Tunnel in a Multi-VPN Environment
  Wei Pan
• IKEv2 Support for Anti-Replay Status Notification
  Wei Pan
• Using ShangMi in the IKEv2
  Frank (Liang) XIA
• IKEv2 IPv4 Downstream Fragmentation
  Daniel Migault
• IKEv2 DSCP Notification
  Daniel Migault
Differentiated Services Field Codepoints Internet Key Exchange version 2 Notification

draft-mglt-ipsecme-dscp-np

Migault, Halpern, Parkholm, Liu
This document specifies the DSCP Notification Payload, which, in a CREATE_CHILD_SA Exchange, explicitly mentions which DSCP code points will be tunneled in the newly created tunnel.
## Illustrative Example

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Responder</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR, SK {IDᵩ, [CERT,] [CERTREQ,] [IDᵩᵠ,] AUTH, SAᵩᵠ, TSi, TSᵩ} --&gt;</td>
<td>HDR, SK {IDᵩ, [CERT,] AUTH, SArᵩ, TSi, TSᵩ} --&gt; HDR, SK {IDᵩᵠ, [CERT,] [CERTREQ,] [IDᵩᵠᵠ,] AUTH, SAᵩᵠᵠ, TSi, TSᵩᵠ}</td>
</tr>
<tr>
<td>HDR, SK {SA, Ni, KEᵩ, N(DSCP, AF11, AF3)} --&gt;</td>
<td>HDR, SK {SA, Nr, KEᵩ, N(DSCP, AF11, AF3)} --&gt; HDR, SK {SA, Ni, KEᵩ, N(DSCP, EE)} --&gt; HDR, SK {SA, Nr, KEᵩ}</td>
</tr>
<tr>
<td>HDR, SK {SA, Ni, KEᵩ, N(DSCP, EE)} --&gt;</td>
<td>HDR, SK {SA, Nr, KEᵩ}</td>
</tr>
</tbody>
</table>
Thanks!
Open Discussion

• Other points of interest?