IP Security Maintenance and Extensions (IPsecME) WG

IETF 108, Tuesday, July 28, 2020

Chairs:	Tero Kivinen
	Yoav Nir
Responsible AD:	Benjamin Kaduk

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

Bluesheets

We need volunteers to be:

- Two note takers
- One jabber scribe

Jabber: xmpp:ipsecme@jabber.ietf.org?join

MeetEcho: https://meetings.conf.meetecho.com/ietf108/?group=ipsecme&short=&item=1

Notes: https://codimd.ietf.org/notes-ietf-108-ipsecme

Agenda

- Note Well, technical difficulties and agenda bashing –
  Chairs (5 min) (11:00-11:05)
- Document Status – Chairs (5 min) (11:05-11:10)
- Work items
  - TCP Encapsulation in IKE and IPsec - RFC8229bis –
    Valery Smyslov (15 min) (11:10-11:25)
  - IKEv2 Configuration for Encrypted DNS –
    Valery Smyslov (10 min) (11:25-11:35)
  - Announcing Supported Authentication Methods in IKEv2 –
    Valery Smyslov (10 min) (11:35-11:45)
  - Proposed improvements to ESP –
    Michael Rossberg (15 min) (11:45-12:00)
  - IP Traffic Flow Security –
    Christian Hopps (10 min) (12:00-12:15)
  - YANG model for IP Traffic Flow Security –
    Christian Hopps (15 min) (12:15-12:30)
- AOB + Open Mic (12:30-12:40)

WG Status Report

Published as RFC:

- draft-ietf-ipsecme-implicit-iv as RFC8750
- draft-ietf-ipsecme-qr-ikev2 as RFC8784

Publication requested:

- draft-ietf-ipsecme-ipv6-ipv4-codes

Work in progress:

- draft-ietf-ipsecme-g-ikev2
- draft-ietf-ipsecme-ikev2-intermediate
- draft-ietf-ipsecme-ikev2-multiple-ke
- draft-hopps-ipsecme-iptfs
- draft-ietf-ipsecme-labeled-ipsec

Presentations

- **TCP Encapsulation in IKE and IPsec - RFC8229bis – Valery Smyslov**
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Announcing Supported Authentication Methods in IKEv2 – Valery Smyslov
- Proposed improvements to ESP – Michael Rossberg
- IP Traffic Flow Security – Christian Hopps
- YANG model for IP Traffic Flow Security – Christian Hopps

TCP Encapsulation of IKE and IPsec Packets Update

`draft-smyslov-ipsecme-rfc8229bis`

Valery Smyslov
svan@elvis.ru

Tommy Pauly
tpauly@apple.com

IETF 108

TCP Encapsulation in IKEv2

- Defined in RFC 8229
- Modifies IKEv2 behavior in various situations:
  - original Initiator is responsible for restoring TCP connection if it is broken
  - with MOBIKE if IP address is changed then first try UDP and then switch to TCP
  - NAT keepalives are redundant
  - IKE Fragmentation is redundant
  - etc.
- However, some nuances in using TCP are missing. Most of them affect performance, however few are essential for reliability and interoperability
- This draft is intended to replace RFC 8229 adding missing clarifications

Retransmissions

- RFC 7296 requires exchange initiator to retransmit request periodically until either response is received or the SA is deemed to have failed
- TCP is reliable protocol, there is generally no need to retransmit
- Moreover, in congested networks retransmitting requests can increase congestion making things worse
- However, if TCP connection is lost and then restored, then IKE implementation must retransmit all outstanding requests

Using COOKIE and PUZZLE

- Using COOKIE allows responder to make sure the initiator’s IP address is real
- In general COOKIE is not useful with TCP:
  - TCP itself verifies that initiator’s IP address is real
  - TCP creates state on responder before first packet ever reaches IKE, that violates stateless nature of COOKIE
- Using PUZZLE still makes sense
- If COOKIE (or PUZZLE) request is sent by responder:
  - TCP connection should be immediately closed by responder (to keep responder stateless)
  - COOKIE calculation must not include initiator’s port number (since it will most probably be different)

Error Handling in IKE_SA_INIT

- RFC 7296 advises initiator not to act immediately if error notification is received in IKE_SA_INIT because it can be forged; instead wait for more responses
- With TCP this makes little sense:
  - if this is genuine message from responder, then other responses won't be sent
  - if TCP is hijacked and this is message is forged by attacker, then genuine response won't be received or will be corrupted (because TCP sequence numbers will already be consumed by attacker's message)

Interaction with MOBIKE

- RFC 4555 defines MOBIKE protocol
- RFC 8229 recommends, that if IP is changed, then initiator first tries to send UPDATE_IP_ADDRESSES notify using UDP and then switches to TCP if no response is received
- Clarifications on the NAT_DETECTION_*_IP content and Message ID are still missing
- When switching to TCP:
  - the content of the NAT_DETECTION_*_IP notifications must be recalculated if source/destination ports differ from UDP's
  - Message ID for TCP-based exchange must remain the same as for (failed) corresponding UDP-based one

Interaction with High Availability Clusters

- RFC 6311 defines IKE Message ID & ESP SN synchronization mechanism between IKE peer and HA cluster:
  - when cluster failover takes place the new active node initiates INFORMATIONAL exchange containing new Message IDs & SN gap
- In case of cluster failover the existing TCP connection is most likely broken and the new active node cannot initiate the exchange until the client restores it (by sending fresh IKE or ESP packet):
  - client is unaware of the fact that the connection is broken, so if it has nothing to send, the connection won't be restored for a long time, and the cluster would eventually tear down the IKE SA
- Clients should periodically send Liveness Check messages if the partner is HA cluster and there is no outgoing ESP traffic

Thank you!

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

Presentations

- TCP Encapsulation in IKE and IPsec - RFC8229bis – Valery Smyslov
- **IKEv2 Configuration for Encrypted DNS – Valery Smyslov**
- Announcing Supported Authentication Methods in IKEv2 – Valery Smyslov
- Proposed improvements to ESP – Michael Rossberg
- IP Traffic Flow Security – Christian Hopps
- YANG model for IP Traffic Flow Security – Christian Hopps

IKEv2 Configuration for Encrypted DNS

`draft-btw-add-ipsecme-ike`

Mohamed Boucadair (Orange)
Tirumaleswar Reddy (McAfee, Inc.)
Dan Wing (Citrix Systems, Inc.)
Valery Smyslov (ELVIS-PLUS)

July 2020, IETF#108

Agenda

- Context
- A Sample Use Case
- IKE Configuration Attribute for Encrypted DNS
- Next Steps

Problem Description

- Several schemes to encrypt DNS have been specified
  - DNS over TLS (RFC 7858)
  - DNS over DTLS (RFC 8094)
  - DNS over HTTPS (RFC 8484)

- …And others are being specified:
  - DNS over QUIC (draft-ietf-dprive-dnsoquic)

- ***How to securely provision clients to use Encrypted DNS? This use can be within or outside the IPsec tunnel***

A Sample Use Case: DNS Offload

- VPN service providers can offer publicly accessible Encrypted DNS
  - the split-tunnel VPN configuration allows the client to access the DoH/DoT servers hosted by the VPN provider ***without traversing the tunnel***

A Sample Use Case: Protecting Internal DNS Traffic

- DoH/DoT ensures DNS traffic is ***not susceptible to internal attacks***
  - see draft-arkko-farrell-arch-model-t-03#section-3.2.1
- encrypted DNS can benefit to Roaming Enterprise users to ***enhance privacy***
  - With DoH/DoT the visibility of DNS traffic is limited to only the parties authorized to act on the traffic (“Zero Trust Architecture”)

Using IKE to Configure Encrypted DNS on Clients

- New configuration attribute `INTERNAL_ENC_DNS` is defined to convey encrypted DNS information to clients:
  - Encrypted DNS type (e.g., DoH/DoT)
  - Scope of encrypted DNS use
  - One or more encrypted DNS server IPv6 addresses
    - For IPv4 addresses are encoded using IPv4-mapped IPv6 address format defined in RFC4291
  - Fully qualified authentication domain name

- The `INTERNAL_ENC_DNS` attributes are exchanged in `IKE_AUTH` exchange along with other configuration attributes

Attribute Format

Scope bit
0: Outside the tunnel
1: Within the tunnel

```
                     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
+-+-----------------------------+-------------------------------+
|R|         Attribute Type      |             Length            |
+-+-------------+---------------+-------------------------------+
|S|Enc DNS Type | Num addresses |                               |
+-+-------------+---------------+                               +
|                         IPv6 Addresses                        ~
|                               +-------------------------------+
~                               |                               |
+-------------------------------+                               |
|                                                               |
~                  DNS Authentication Domain Name               ~
|                                                               |
+---------------------------------------------------------------+
```

1: DoT
2: DoH
…

Interaction with Split DNS IKE Extension

- RFC 8598 *Split DNS Configuration for the Internet Key Exchange Protocol Version 2 (IKEv2)* requires `INTERNAL_IP*_DNS` attribute(s) to be present when `INTERNAL_DNS_DOMAIN` is included

- It is **no more needed** if `INTERNAL_ENC_DNS` attribute is present

Next Steps

- Comments?
- Questions?
- Suggestions for progressing the document?

Thank you

Backup Slides

DoH Specifics

- DoH servers may support more than one URI Template
- The DoH server may also host several DoH services (e.g., no-filtering, blocking adult content)
  - These services can be discovered as templates
- The client uses a well-known URI "resinfo" to discover these templates:

$$\text{https://}\underbrace{\text{doh.example.com}}_{\text{Authentication Domain Name}}\text{/.well-known/}\underbrace{\text{resinfo}}_{\text{To be assigned by IANA}}$$

- Discovering the well-known URI is out of scope of this draft and is discussed in draft-btw-add-rfc8484-clarification

Presentations

- TCP Encapsulation in IKE and IPsec - RFC8229bis – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- **Announcing Supported Authentication Methods in IKEv2 – Valery Smyslov**
- Proposed improvements to ESP – Michael Rossberg
- IP Traffic Flow Security – Christian Hopps
- YANG model for IP Traffic Flow Security – Christian Hopps

Announcing Supported Authentication Methods in IKEv2

`draft-smyslov-ipsecme-ikev2-auth-announce`

Valery Smyslov

svan@elvis.ru

IETF 108

Authentication in IKEv2

- Unlike IKEv1, authentication method in IKEv2 is not negotiated, each peer is free to use whichever method he thinks is appropriate
- Generally works well if there is only one way of doing authentication or there is no ambiguity in choosing among several of them
- If peers can use several methods to authenticate each other, it is possible that initiator selects authentication method unsupported by the responder
  - less likely in the opposite direction, but still possible

The Problem

- The problem was first encountered when RSA-PSS signature format appeared in IKEv2
  - newer initiators tried to use PSS signatures while older responders didn’t support it, sending back **`AUTHENTICATION_FAILED`**
  - if initiators knew responders’ capabilities they would have chosen PKCS#1 and the SA succeeded

Source of the Problem

- Currently there is no way for the peers to explicitly indicate the supported authentication methods
  - it is possible to guess them via indirect means, e.g. `CERTREQ` content, but this is unreliable
- With new signature formats and authentication methods appearing in the future (including PQ and hybrid ones) the situation of mis-selecting may happen more often

Proposed Solution

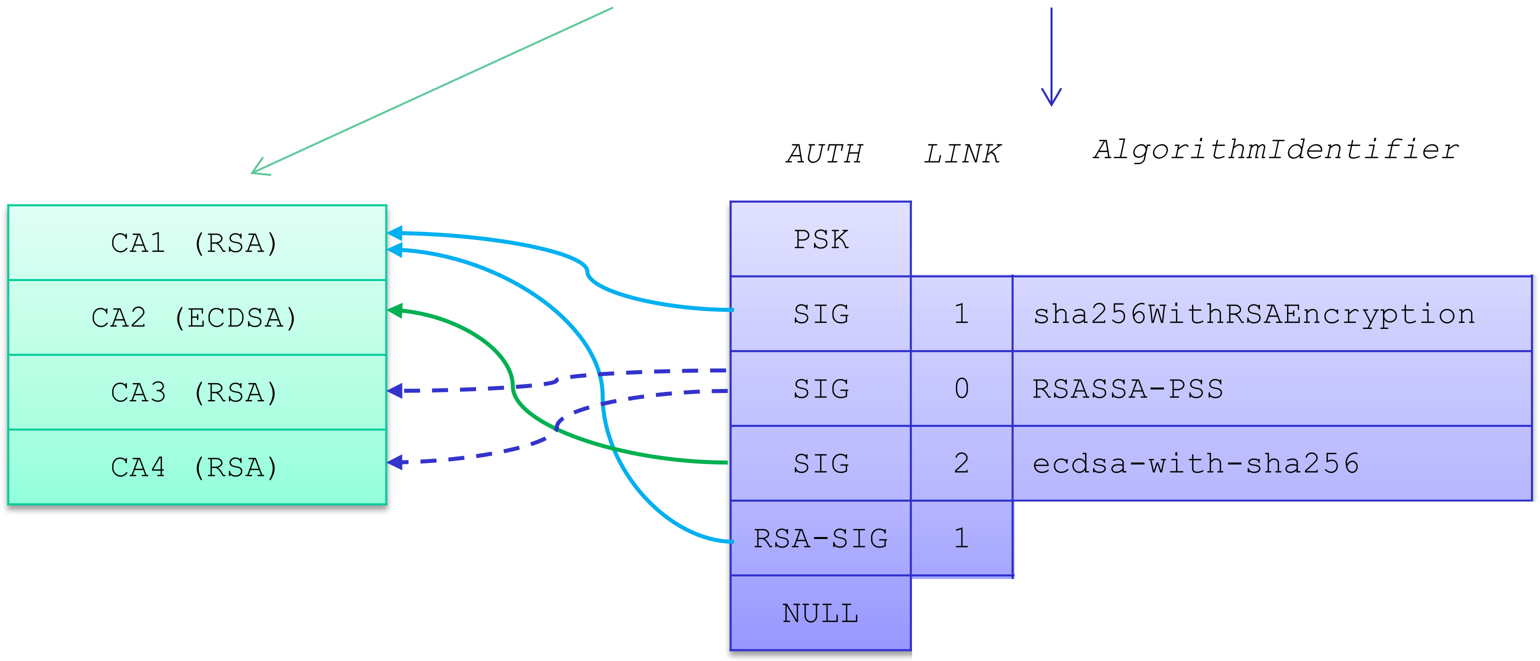
- Add new optional notification `SUPPORTED_AUTH_METHODS` to indicate the supported authentication methods
  - for certificate-based authentication add an ability for the peers to indicate which signing algorithms can be used with each of CA in the `CERTREQ` payload
  - avoid creating new IANA registries

SUPPORTED_AUTH_METHODS Notification Format

- Notification data consists of a list of supported authentication methods in the following formats:
  1. Two-octet format for the methods that are not linked to `CERTREQ` payload (`PSK`, `NULL`)
  2. Three-octet format that allows optional linking to `CERTREQ` payload (`RSA-SIG` etc.)
  3. Multi-octet format that allows optional linking to `CERTREQ` payload and specifying ASN.1 `AlgorithmIdentifier` for use with particular CA (`SIG`)
- The linking to CAs is done by specifying the CA number within the `CERTREQ` payload the method can be used with

SUPPORTED_AUTH_METHODS Notification Format Illustration

HDR,SAr1,KEr,Nr,CERTREQ,N(SUPPORTED_AUTH_METHODS)



Exchanges (Option 1)

Initiator		Responder
IKE_SA_INIT HDR,SAi1,KEi,Ni	→ ←	**IKE_SA_INIT** HDR,SAr1,KEr,Nr,[CERTREQ,] [N(SUPPORTED_AUTH_METHODS)(…)]
IKE_AUTH HDR,SK{IDi,[CERT,][CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr, [N(SUPPORTED_AUTH_METHODS)(…)]}	→ ←	**IKE_AUTH** HDR,SK{IDr,[CERT,] AUTH, SAi2, TSi, TSr}

Exchanges (Option 2)

```
Initiator                                                         Responder
-----------------------------------------------------------------------------
IKE_SA_INIT
HDR,SAi1,KEi,Ni                  --->
                                 <---                          IKE_SA_INIT
                                        HDR,SAr1,KEr,Nr,[CERTREQ,]
                                        [N(SUPPORTED_AUTH_METHODS)]

IKE_INTERMEDIATE
HDR,SK{…}                        --->
                                 <---                     IKE_INTERMEDIATE
                                                                HDR,SK{…,
                                         N(SUPPORTED_AUTH_METHODS)(…)}

IKE_AUTH
HDR,SK{IDi,[CERT,][CERTREQ,]     --->
[IDr,] AUTH, SAi2, TSi, TSr,     <---                             IKE_AUTH
[N(SUPPORTED_AUTH_METHODS)(…)]}                     HDR,SK{IDr,[CERT,]
                                                AUTH, SAi2, TSi, TSr}
```

Thanks

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

Presentations

- TCP Encapsulation in IKE and IPsec - RFC8229bis – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Announcing Supported Authentication Methods in IKEv2 – Valery Smyslov
- **Proposed improvements to ESP – Michael Rossberg**
- IP Traffic Flow Security – Christian Hopps
- YANG model for IP Traffic Flow Security – Christian Hopps

Michael Rossberg · Michael Pfeiffer

Technische Universität Ilmenau, Germany

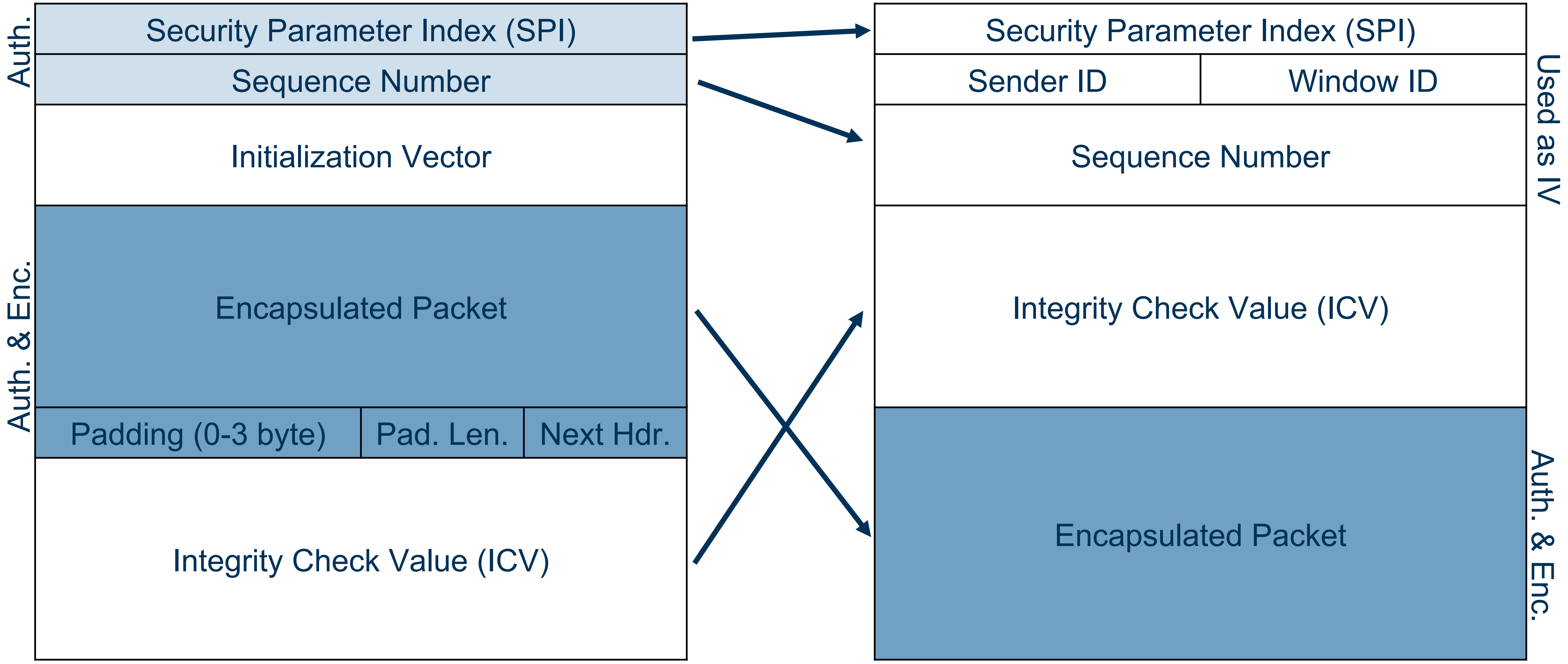
PROPOSED IMPROVEMENTS TO ESP AIDING DATA CENTER DEPLOYMENTS

Motivation

- Scenario: ESP as is in data centers
- Due to handling of sequence numbers:
  - Limited parallelism
  - No multicast replay protection
  - Issues with QoS
- Due to trailer: complex protocol handling
  - Fragments
  - Segments
  - Alignment

→ Approach: change ESP a “little” → New protocol/version/mode?

Packet Layout for Tunnel Mode



Resulting Packet Layout for Tunnel Mode

→ Multiple replay windows per SA

- Allows scaling over CPU cores,
- Multicast replay protection &
- Replay window per QoS class

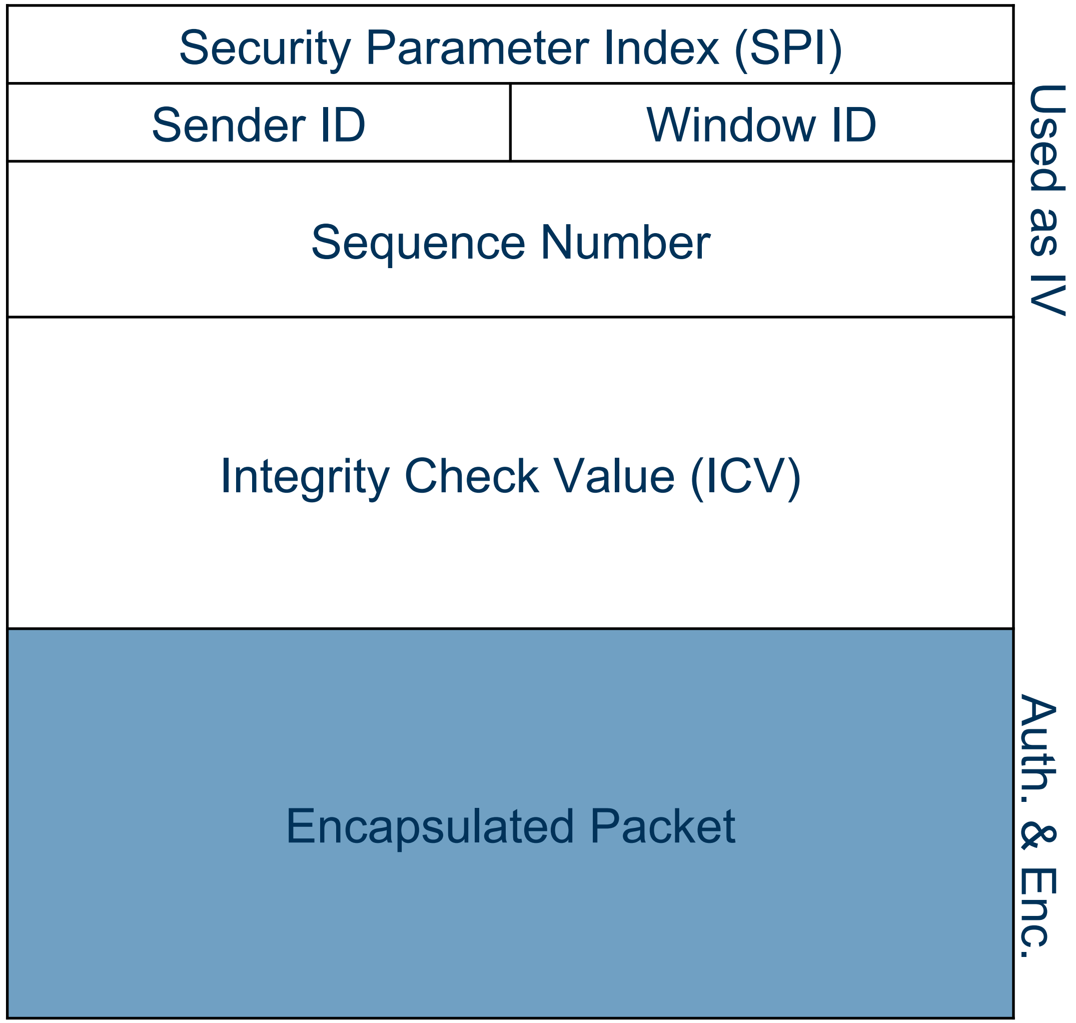
→ Full 64-bit sequence counter

→ No trailer required

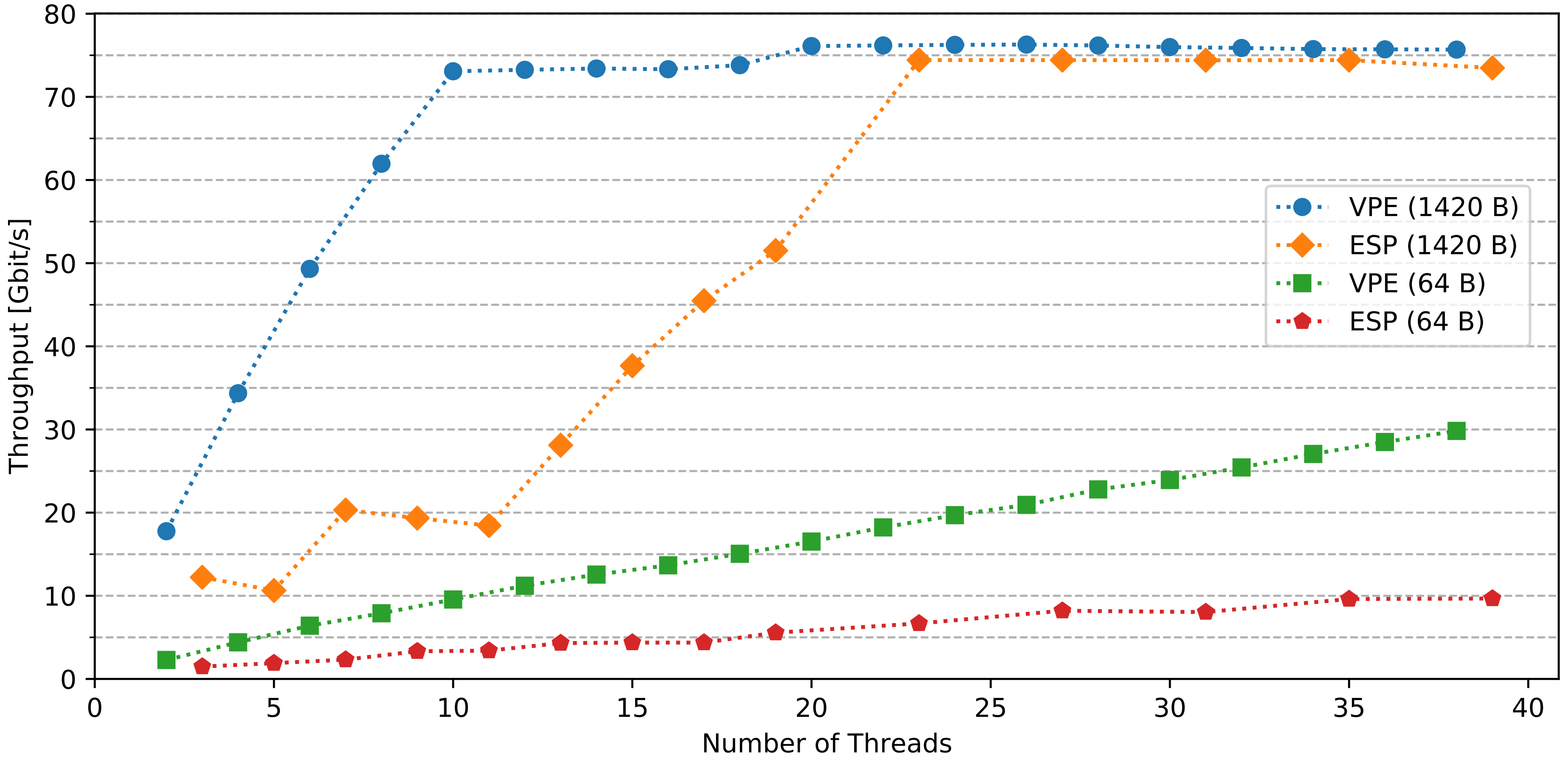
→ Implicit padding if required

→ No AAD required

Could be negotiated during IKE & coexist



Multi-Core Throughput



THANKS FOR LISTENING!

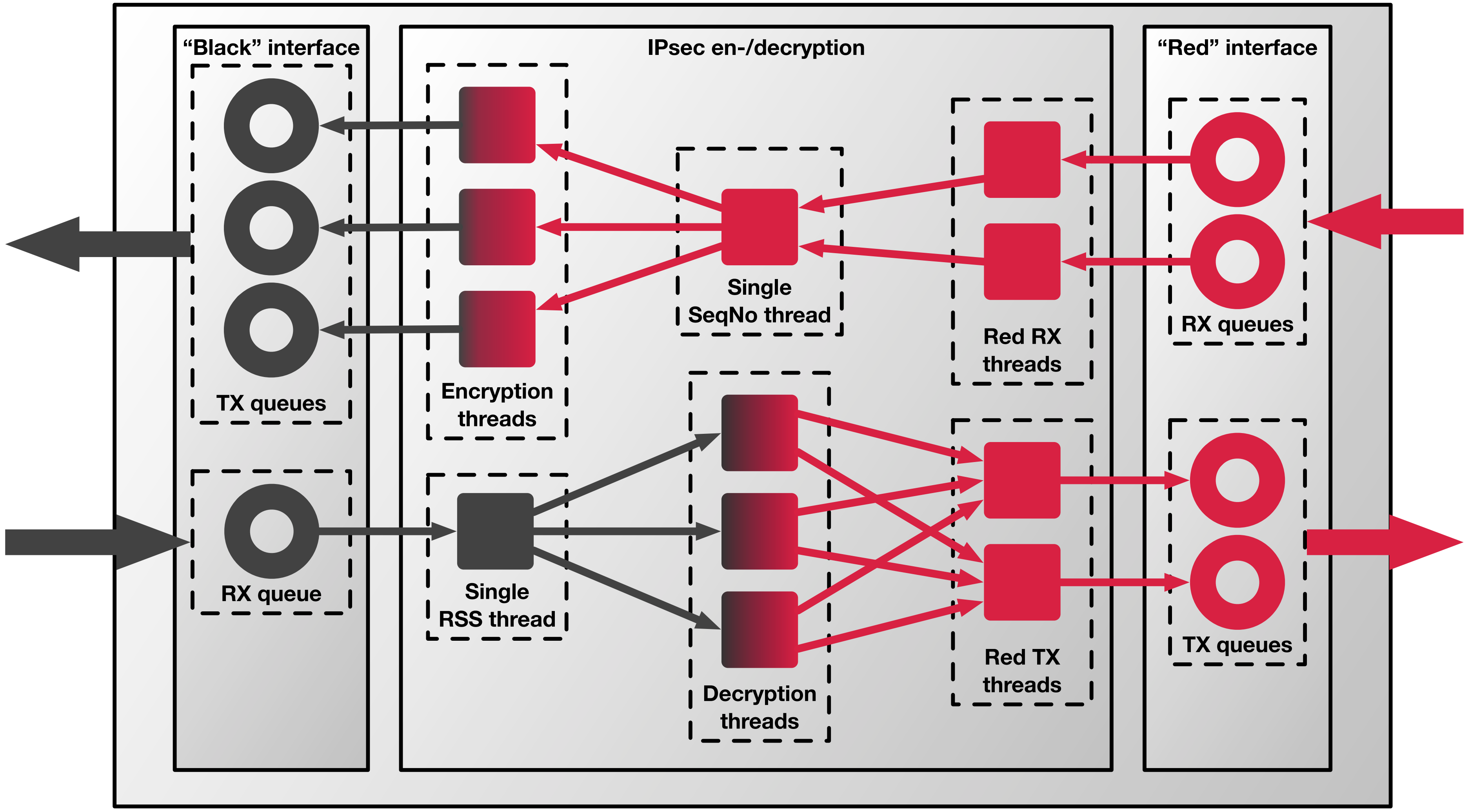
Contact:

michael.rossberg@tu-ilmenau.de & michael.pfeiffer@tu-ilmenau.de

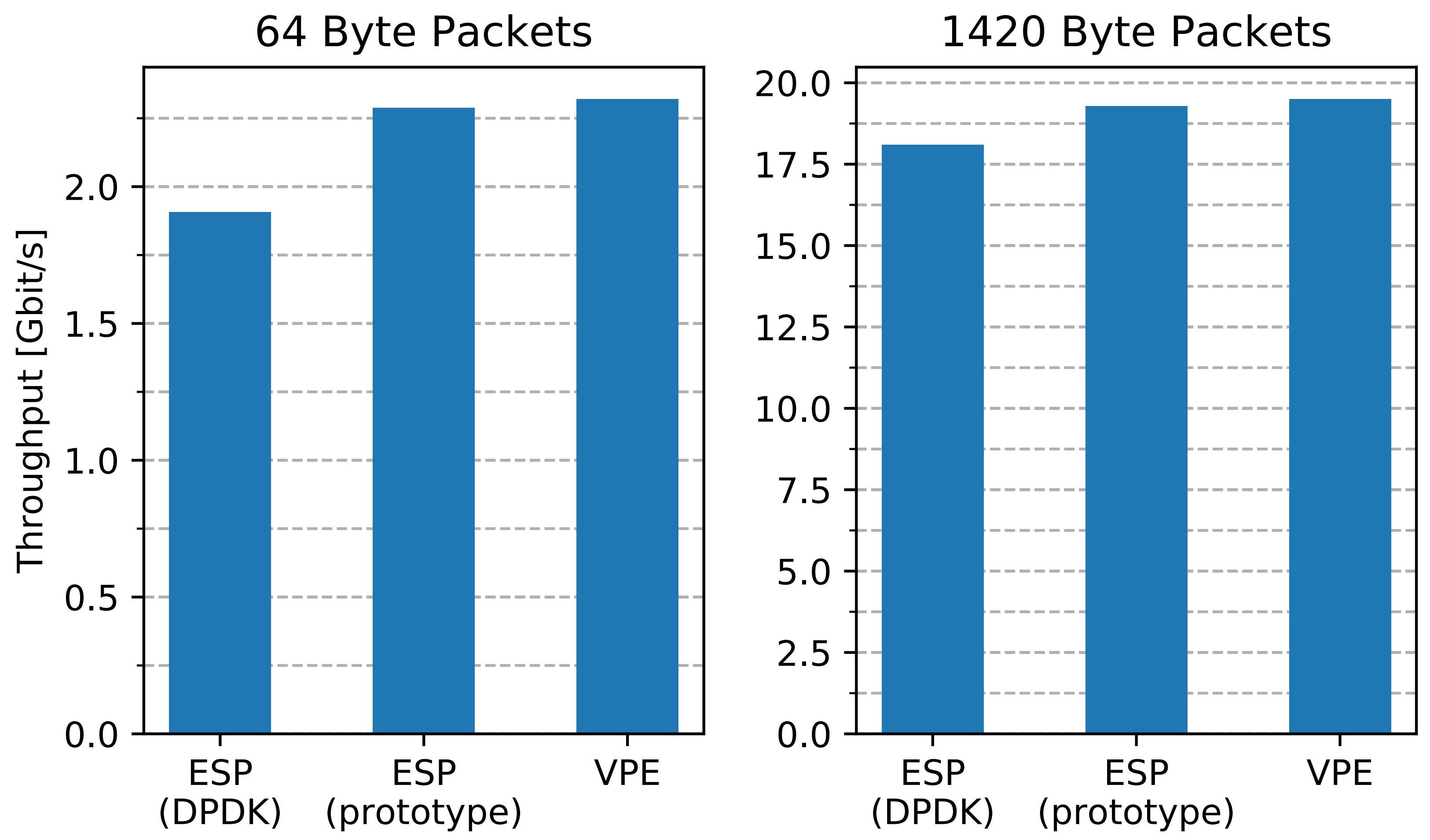
Further details:

[1] https://telematik.prakinf.tu-ilmenau.de/files/packetformat.pdf
[2] https://telematik.prakinf.tu-ilmenau.de/files/VPE.pdf

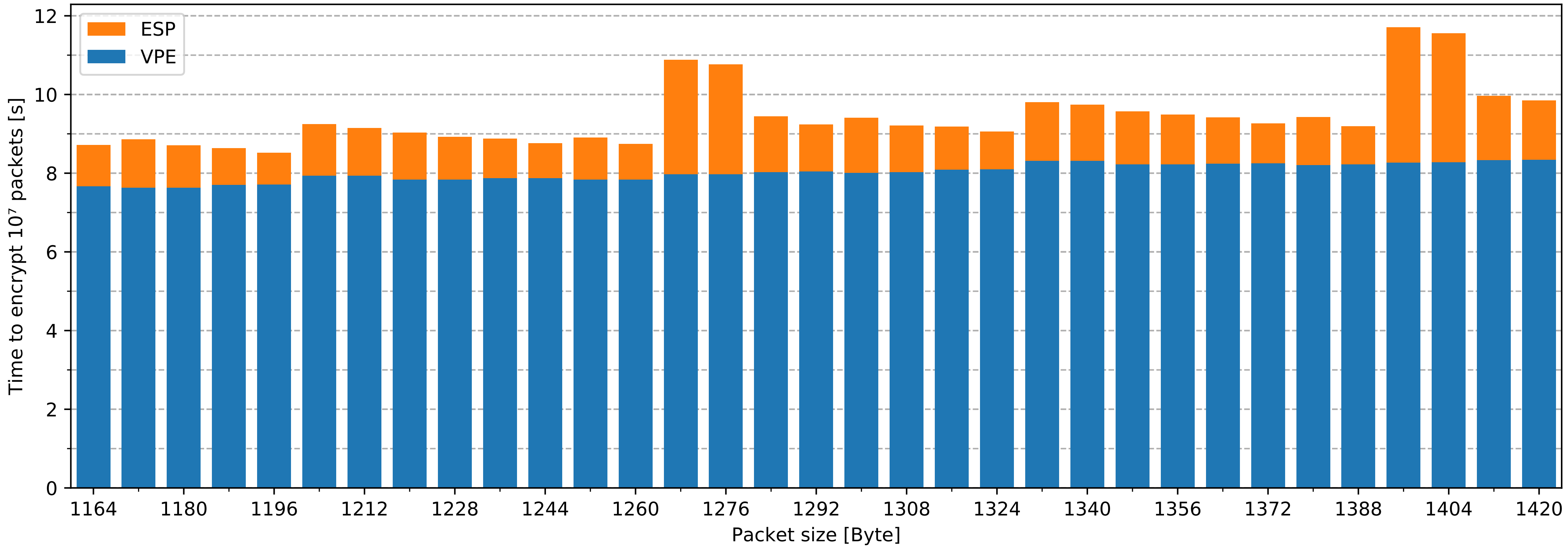
Parallel ESP processing: Threading model



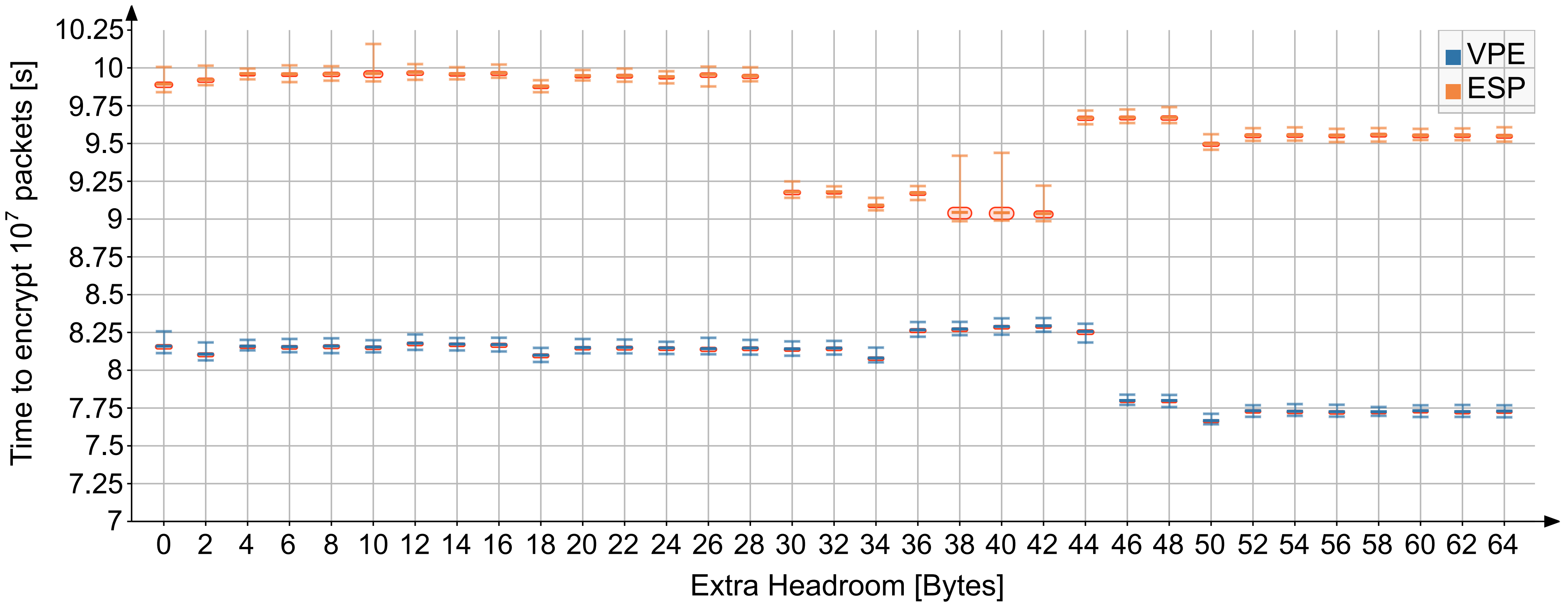
Non-parallel “well-behaved” throughput



Processing Time vs. Packet Size



Processing Time vs. Additional Headroom



Presentations

- TCP Encapsulation in IKE and IPsec - RFC8229bis – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Announcing Supported Authentication Methods in IKEv2 – Valery Smyslov
- Proposed improvements to ESP – Michael Rossberg
- **IP Traffic Flow Security – Christian Hopps**
- YANG model for IP Traffic Flow Security – Christian Hopps

Christian Hopps

LabN Consulting, LLC

IP Traffic Flow Security

Improving IPsec Traffic Flow Confidentiality

IETF 107 – “draft-ipsecme-iptfs-01”

Update Since IETF 106

- draft-ietf-ipsecme-iptfs-01 published March 2, 2020
    - Prior to IETF 107 to address IETF 106 comments.
- Notable Changes
    - IKEv2 Transform changed to Notification
    - Added Sub-Type octet

IKEv2 USE_IPTFS Notification

- Use notification during IKE_AUTH and CREATE_CHILD_SA for enabling IPTFS.
- Similar to USE_TRANSPORT_MODE (et al.) method.
- Required flags payload
- If required flags are not understood or supported then IPTFS mode is not enabled by responder or initiator deletes now established SA.

IKEv2 USE_IPTFS Notification Required Flags

```
+-+-+-+-+-+-+-+-+
|0|0|0|0|0|0|C|D|
+-+-+-+-+-+-+-+-+
```

- **C** :: Congestion control bit. If set, the sender is requiring that congestion control information MUST be returned to it periodically
- **D** :: Don’t Fragment bit. if set, the sender of the notify message does not support receiving packet fragments

IPTFS_PROTOCOL Payload Format

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-
|   Sub-type    | ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

- **Sub-Type** :: An octet indicating the payload format.

Non-Congestion Control Payload Format

```
                    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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-Type (0)  |    Reserved   |          BlockOffset          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       DataBlocks ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

- **Sub-Type** :: An octet (value 0) indicating this payload format.

Congestion Control Payload Format

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-Type (1)  |   Reserved   |E|          BlockOffset          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              RTT              |             Delay             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         LossEventRate                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          LastSeqNum                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       DataBlocks ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

- **Sub-Type** :: An octet (value 1) indicating this payload format
- **E** :: ECN bit were used in calculating the **LossEventRate**
  - Same definition as before, just moved

Open Issues/Last Meeting Comments

- IP Number
  - Discussed on list a couple times. Waiting for chairs to forward the request.
  - Summary:
    - Use WESP consumes bandwidth, still have need for next-header number.
    - Get a number, start process early, our use is valid, IETF process should not block technically better choices.
    - Can fallback to overloading another IP protocol number for ESP only use.
- Transport Mode
  - To be defined in separate document
  - Will not conflict with this tunnel mode based document
    - sub-type, flags, or the mode itself can be used to differentiate any header changes

Other Issues/Notes

- Datablock (inner packet) alignment.
  - Con: Complicates encap/decap specification and code
  - Con: Wastes bandwidth
  - Pro: Aligning internal packets allows less rigorous whitebox code to work.
    - Ends up not being an issue as copy-out of packet header is required even when using indirect buffer chains.
    - ASICs “copy” so don’t care.
  - Thus: haven’t needed this during implementation.
- Open source implementation
  - VPP/DPDK implementation to be published in 2020
  - Congestion Control
  - IKEv2
- Open to collaboration/interoperability testing.

Moving Forward

- Any remaining comments?
- Ready for WGLC?

Questions and Comments

Backup Slides

Transport Mode

- Motivation is common GRE/IPsec-Transport Use
- Some interest in generic transport mode.
- What IP header fields to support
  - Simple
    - No fields – GRE Support
      - If the packet header is different then the last, pad current IPTFS out and start new one
      - If is inefficient due to frequent header differences, then use tunnel mode.
    - All Fields
      - IP header replicated inside payload for each packet
      - Similar to tunnel mode, but less efficient.
  - Complex
    - IP Header compression Ideas (deviations, etc)
      - Complex solution in need of a problem?
- Enough separable work to publish as a separate document.

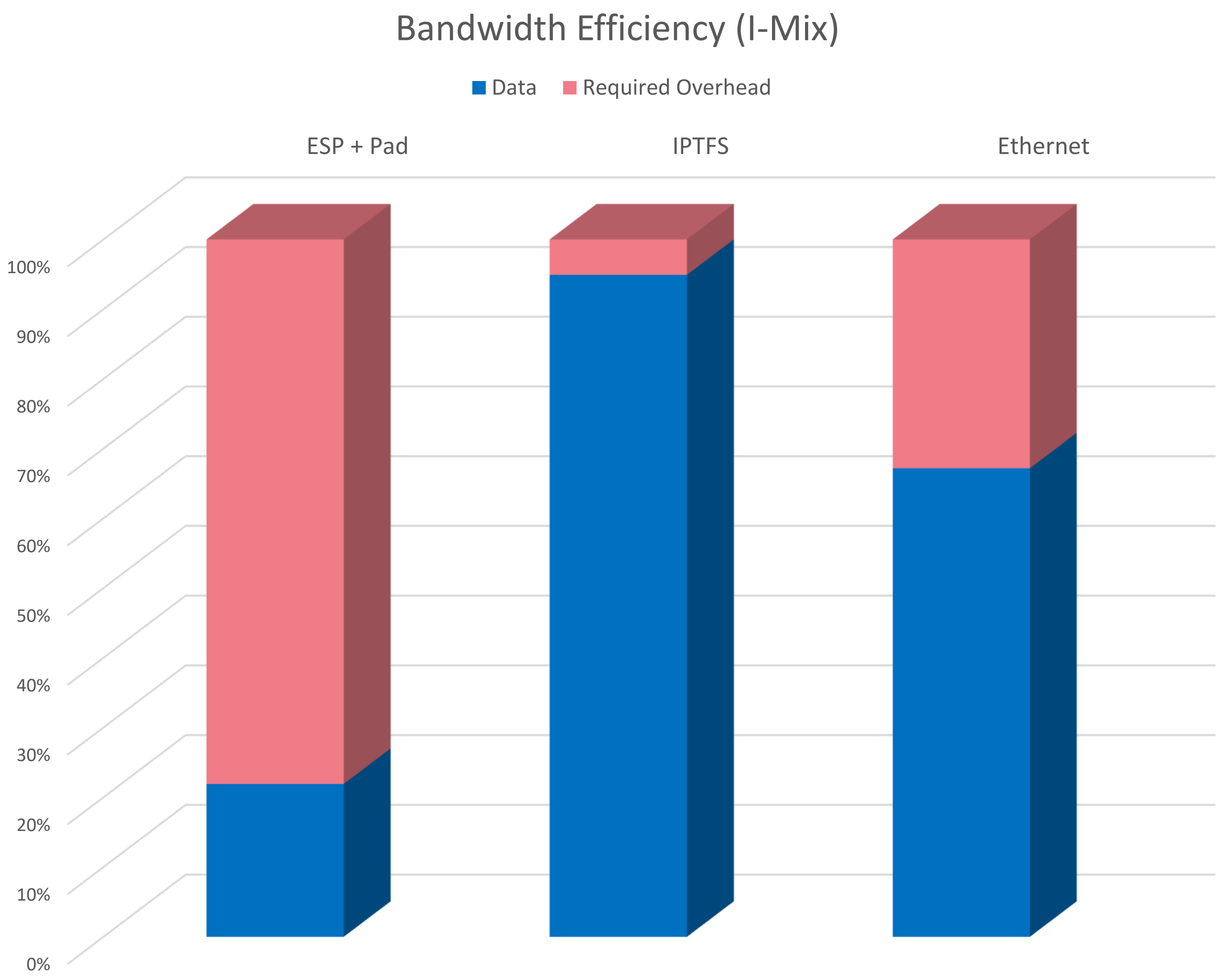
Comparison Data

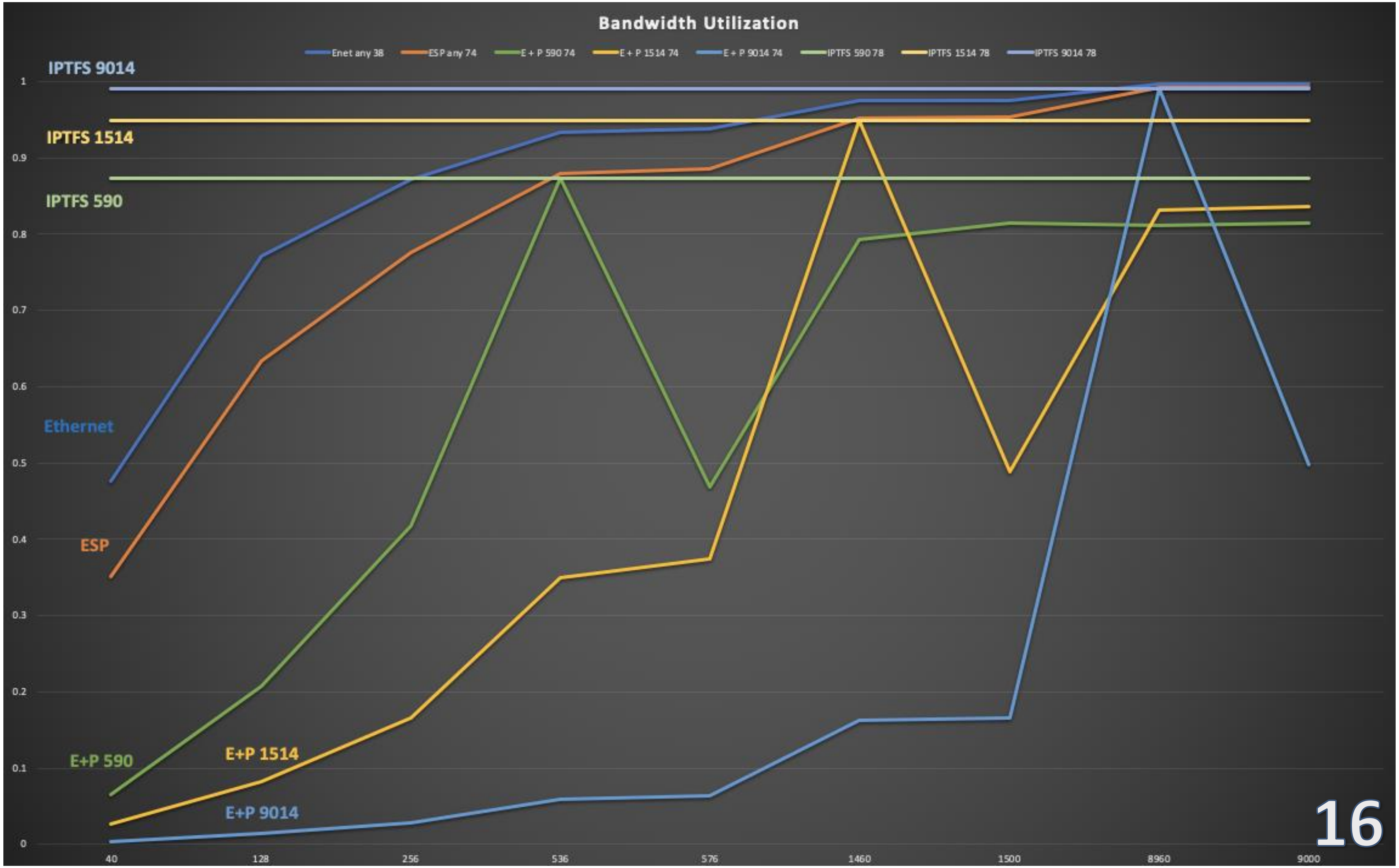
Why is this Needed?

- Current Solution: ESP + Padding 1:1
- Not Deployable.

Solution Cost (I-Mix)

	ESP + Pad	**IPTFS**	**Enet**
Bandwidth Used	1Gb	1Gb	1Gb
I-Mix Throughput	219Mb	943Mb	672Mb




Bandwidth Utilization
Enet any 38
ESP any 74
E + P 590 74
E + P 1514 74
E + P 9014 74
IPTFS 590 78
IPTFS 1514 78
IPTFS 9014 78
IPTFS 9014
IPTFS 1514
IPTFS 590
Ethernet
ESP
E+P 590
E+P 1514
E+P 9014
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
40
128
256
536
576
1460
1500
8960
9000

Overhead Comparison in Octets

Type	ESP+Pad	ESP+Pad	ESP+Pad	IP-TFS	IP-TFS	IP-TFS
L3 MTU	576	1500	9000	576	1500	9000
PSize	540	1464	8964	536	1460	8960
40	500	1424	8924	3.0	1.1	0.2
128	412	1336	8836	9.6	3.5	0.6
256	284	1208	8708	19.1	7.0	1.1
536	4	928	8428	40.0	14.7	2.4
576	576	888	8388	43.0	15.8	2.6
1460	268	4	7504	109.0	40.0	6.5
1500	228	1500	7464	111.9	41.1	6.7
8960	1408	1540	4	668.7	245.5	40.0
9000	1368	1500	9000	671.6	246.6	40.2

Overhead as Percentage of Inner Packet

Type	ESP+Pad	ESP+Pad	ESP+Pad	IP-TFS	IP-TFS	IP-TFS
MTU	576	1500	9000	576	1500	9000
PSize	540	1464	8964	536	1460	8960
40	1250.0%	3560.0%	22310.0%	7.46%	2.74%	0.45%
128	321.9%	1043.8%	6903.1%	7.46%	2.74%	0.45%
256	110.9%	471.9%	3401.6%	7.46%	2.74%	0.45%
536	0.7%	173.1%	1572.4%	7.46%	2.74%	0.45%
576	100.0%	154.2%	1456.2%	7.46%	2.74%	0.45%
1460	18.4%	0.3%	514.0%	7.46%	2.74%	0.45%
1500	15.2%	100.0%	497.6%	7.46%	2.74%	0.45%
8960	15.7%	17.2%	0.0%	7.46%	2.74%	0.45%
9000	15.2%	16.7%	100.0%	7.46%	2.74%	0.45%

Bandwidth Utilization over Ethernet

	Enet	ESP	E + P	E + P	E + P	IPTFS	IPTFS	IPTFS
	any	any	590	1514	9014	590	1514	9014
Size	38	74	74	74	74	78	78	78
40	47.6%	35.1%	6.5%	2.6%	0.4%	87.3%	94.9%	99.1%
128	77.1%	63.4%	20.8%	8.3%	1.4%	87.3%	94.9%	99.1%
256	87.1%	77.6%	41.7%	16.6%	2.8%	87.3%	94.9%	99.1%
536	93.4%	87.9%	87.3%	34.9%	5.9%	87.3%	94.9%	99.1%
576	93.8%	88.6%	46.9%	37.5%	6.4%	87.3%	94.9%	99.1%
1460	97.5%	95.2%	79.3%	94.9%	16.2%	87.3%	94.9%	99.1%
1500	97.5%	95.3%	81.4%	48.8%	16.6%	87.3%	94.9%	99.1%
8960	99.6%	99.2%	81.1%	83.2%	99.1%	87.3%	94.9%	99.1%
9000	99.6%	99.2%	81.4%	83.6%	49.8%	87.3%	94.9%	99.1%

Latency

- Latency values seem very similar
- IP-TFS values represent max latency
- IP-TFS provides for constant high bandwidth
- ESP + padding value represents min latency
- ESP + padding often greatly reduces available bandwidth.

	ESP+Pad	ESP+Pad	IP-TFS	IP-TFS
	1500	9000	1500	9000
40	1.14 us	7.14 us	1.17 us	7.17 us
128	1.07 us	7.07 us	1.10 us	7.10 us
256	0.97 us	6.97 us	1.00 us	7.00 us
536	0.74 us	6.74 us	0.77 us	6.77 us
576	0.71 us	6.71 us	0.74 us	6.74 us
1460	0.00 us	6.00 us	0.04 us	6.04 us
1500	1.20 us	5.97 us	0.00 us	6.00 us

Presentations

- TCP Encapsulation in IKE and IPsec - RFC8229bis – Valery Smyslov
- IKEv2 Configuration for Encrypted DNS – Valery Smyslov
- Announcing Supported Authentication Methods in IKEv2 – Valery Smyslov
- Proposed improvements to ESP – Michael Rossberg
- IP Traffic Flow Security – Christian Hopps
- **YANG model for IP Traffic Flow Security – Christian Hopps**

Donald Fedyk
Christian Hopps
LabN Consulting, LLC

YANG Model for IP Traffic Flow Security

IETF 108 – “draft-fedyk-ipsecme-yang-iptfs-00”

IP-TFS Configuration

- Congestion Control
  - Boolean
- Packet Size (L3 Packet size)
  - Fixed Size
  - Use Path MTU (set or lowers fixed)
- Bit rate
  - L3 Bit rate or
  - L2 Bit rate
- Allow fragmentation
  - Of Inner packets using data blocks and IP TFS offsets

Packet Transmission Frequency
= Bit rate/Packet size

Note these are minimal controls vendors or future work may augment

IP-TFS Config augment `ipsec-ike`

```
module: ietf-ipsecme-iptfs
  augment /ike:ipsec-ike/ike:conn-entry
          /ike:spd/ike:spd-entry
          /ike:ipsec-policy-config/ike:processing-info
          /ike:ipsec-sa-cfg:
    +--rw traffic-flow-security
       +--rw congestion-control?   boolean
       +--rw packet-size
       |  +--rw use-path-mtu?        boolean
       |  +--rw outer-packet-size?   uint16
       +--rw (tunnel-rate)?
       |  +--:(l2-bitrate)
       |  |  +--rw l2-bitrate?     uint64
       |  +--:(l3-bitrate)
       |     +--rw l3-bitrate?     uint64
       +--rw dont-fragment?        boolean

  augment /ike:ipsec-ike/ike:conn-entry
          /ike:child-sa-info:
    +--ro traffic-flow-security
       +--ro congestion-control?   boolean
       +--ro packet-size
       |  +--ro use-path-mtu?        boolean
       |  +--ro outer-packet-size?   uint16
       +--ro (tunnel-rate)?
       |  +--:(l2-bitrate)
       |  |  +--ro l2-bitrate?     uint64
       |  +--:(l3-bitrate)
       |     +--ro l3-bitrate?     uint64
       +--ro dont-fragment?        boolean
```

User Provided Config

Operational (Actual) Config

IP-TFS Config augment `ipsec-ikeless`

```
augment /ikeless:ipsec-ikeless
        /ikeless:spd/ikeless:spd-entry
        /ikeless:ipsec-policy-config/ikeless:processing-info
        /ikeless:ipsec-sa-cfg:
   +--rw traffic-flow-security
      +--rw congestion-control?   boolean
      +--rw packet-size
      |  +--rw use-path-mtu?        boolean
      |  +--rw outer-packet-size?   uint16
      +--rw (tunnel-rate)?
      |  +--:(l2-bitrate)
      |  |  +--rw l2-bitrate?     uint64
      |  +--:(l3-bitrate)
      |     +--rw l3-bitrate?     uint64
      +--rw dont-fragment?        boolean
```

User Provided Config
(same as IKE, under spd-entry grouping)

```
augment /ikeless:ipsec-ikeless
        /ikeless:sad/ikeless:sad-entry:
   +--ro traffic-flow-security
      +--ro congestion-control?   boolean
      +--ro packet-size
      |  +--ro use-path-mtu?        boolean
      |  +--ro outer-packet-size?   uint16
      +--ro (tunnel-rate)?
      |  +--:(l2-bitrate)
      |  |  +--ro l2-bitrate?     uint64
      |  +--:(l3-bitrate)
      |     +--ro l3-bitrate?     uint64
      +--ro dont-fragment?        boolean
```

Operational (Actual) Config
(diff from IKE, now under SAD entry)

Operational Statistics

- Outer IPsec Packet – IPsec Counters
  - tx IPsec packets and octets
  - rx IPsec packets and octets
  - rx dropped packet counts
  - rx error counts/type
- Inner IP Packets – IP-TFS Counters
  - tx packets and octets
  - tx extra pad packets and octets
  - tx all pad packets and octets
  - rx packets and octets
  - rx extra pad packets and octets
  - rx all pad packets and octets
  - rx errored packets
  - rx missed packets
  - rx incomplete inner packets

IP-TFS Protocol Overhead = Outer Packet Octets - Inner Packet Octets - Pad Octets

Statistics augment **`ipsec-ike`** (all-new)

```
augment /ike:ipsec-ike/ike:conn-entry/ike:child-sa-info:

   +--ro tx-packets?                    uint64 {ipsec-stats}?
   +--ro tx-octets?                     uint64 {ipsec-stats}?
   +--ro tx-drop-packets?               uint64 {ipsec-stats}?
   +--ro rx-packets?                    uint64 {ipsec-stats}?
   +--ro rx-octets?                     uint64 {ipsec-stats}?
   +--ro rx-drop-packets?               uint64 {ipsec-stats}?
   +--rw rx-dropped-packet-detail {ipsec-stats}?
   |  +--ro sa-non-exist?    uint64
   |  +--ro queue-full?      uint64
   |  +--ro auth-failure?    uint64
   |  +--ro malform?         uint64
   |  +--ro replay?          uint64
   |  +--ro large-packet?    uint64
   |  +--ro invalid-sa?      uint64
   |  +--ro policy-deny?     uint64
   |  +--ro other-reason?    uint64
   +--ro tx-inner-packets?              uint64 {iptfs-stats}?
   +--ro tx-inner-octets?               uint64 {iptfs-stats}?
   +--ro tx-extra-pad-packets?          uint64 {iptfs-stats}?
   +--ro tx-extra-pad-octets?           uint64 {iptfs-stats}?
   +--ro tx-all-pad-packets?            uint64 {iptfs-stats}?
   +--ro tx-all-pad-octets?             uint64 {iptfs-stats}?
   +--ro rx-inner-packets?              uint64 {iptfs-stats}?
   +--ro rx-inner-octets?               uint64 {iptfs-stats}?
   +--ro rx-extra-pad-packets?          uint64 {iptfs-stats}?
   +--ro rx-extra-pad-octets?           uint64 {iptfs-stats}?
   +--ro rx-all-pad-packets?            uint64 {iptfs-stats}?
   +--ro rx-all-pad-octets?             uint64 {iptfs-stats}?
   +--ro rx-errored-packets?            uint64 {iptfs-stats}?
   +--ro rx-missed-packets?             uint64 {iptfs-stats}?
   +--ro rx-incomplete-inner-packets?   uint64 {iptfs-stats}?
```

$\}$ IPsec Statistics

$\}$ IP-TFS Statistics

Statistics augment **`ipsec-ikeless`** (all-new)

```
augment /ikeless:ipsec-ikeless/ikeless:sad/ikeless:sad-entry:
   +--ro tx-packets?                    uint64 {ipsec-stats}?
   +--ro tx-octets?                     uint64 {ipsec-stats}?
   +--ro tx-drop-packets?               uint64 {ipsec-stats}?
   +--ro rx-packets?                    uint64 {ipsec-stats}?
   +--ro rx-octets?                     uint64 {ipsec-stats}?
   +--ro rx-drop-packets?               uint64 {ipsec-stats}?
   +--rw rx-dropped-packet-detail {ipsec-stats}?
   |  +--ro sa-non-exist?   uint64
   |  +--ro queue-full?     uint64
   |  +--ro auth-failure?   uint64
   |  +--ro malform?        uint64
   |  +--ro replay?         uint64
   |  +--ro large-packet?   uint64
   |  +--ro invalid-sa?     uint64
   |  +--ro policy-deny?    uint64
   |  +--ro other-reason?   uint64
   +--ro tx-inner-packets?              uint64 {iptfs-stats}?
   +--ro tx-inner-octets?               uint64 {iptfs-stats}?
   +--ro tx-extra-pad-packets?          uint64 {iptfs-stats}?
   +--ro tx-extra-pad-octets?           uint64 {iptfs-stats}?
   +--ro tx-all-pad-packets?            uint64 {iptfs-stats}?
   +--ro tx-all-pad-octets?             uint64 {iptfs-stats}?
   +--ro rx-inner-packets?              uint64 {iptfs-stats}?
   +--ro rx-inner-octets?               uint64 {iptfs-stats}?
   +--ro rx-extra-pad-packets?          uint64 {iptfs-stats}?
   +--ro rx-extra-pad-octets?           uint64 {iptfs-stats}?
   +--ro rx-all-pad-packets?            uint64 {iptfs-stats}?
   +--ro rx-all-pad-octets?             uint64 {iptfs-stats}?
   +--ro rx-errored-packets?            uint64 {iptfs-stats}?
   +--ro rx-missed-packets?             uint64 {iptfs-stats}?
   +--ro rx-incomplete-inner-packets?   uint64 {iptfs-stats}?
```

IPsec Statistics

IP-TFS Statistics

Existing IPsec YANG

- ietf-i2nsf-sdn-ipsec-flow-protection
  - Only active/published IPsec YANG model
  - https://tools.ietf.org/html/draft-ietf-i2nsf-sdn-ipsec-flow-protection-07
  - Submitted to IESG for Publication
  - Defines
    - ietf-ipsec-common@2019-08-05.yang
    - ietf-ipsec-ike@2019-08-05.yang
    - ietf-ipsec-ikeless@2019-08-05.yang
  - IP-TFS YANG augments this model
- Also Expired: draft ietf-tran-ipsecme-yang-01
  - https://tools.ietf.org/html/draft-tran-ipsecme-yang-01

Open Issue – SDN IPsec model

- The SDN model provides for an IKE and IKE-less operation
- IKE module intentionally missing a Security Association Database
  - Reason given: centralized controler (SDN) doesn't care about SAs
  - Has `child-sa-info` to hold connections SA related info
- IKE module missing SA information
  - `child-sa-info` only has pfs-groups and lifetime values
  - no information on selected transforms, etc
- Existing model (IKE/IKE-less) missing Basic IPsec counters
  - Missing from IKE-less SAD entries
  - Also missing under IKE `child-sa-info`

Open Issue – SDN IPsec model (cont)

- Could easily be modified to allow for more general use.
- Move SAD into common model prior to publishing
  - IKE could then refer to the CHILD_SA in child-sa-info
  - Would provide for missing SA info (transforms, etc)
- Move SPD into common model prior to publishing
  - IKE still utilizes SPDs
  - SPDs are operational data that the user may wish to query
- Otherwise, probably need to rename modules to add "sdn" to their names

SDN IPsec proposed changes (ikeless/common)

```
module: ietf-ipsec-ikeless

  +--rw ipsec-ikeless
     +--rw spd
     |  +--rw spd-entry* [name]
     |     +--rw name
     |     +--rw direction?
     |     +--rw reqid?
     |     ...
     +--rw sad
        +--rw sad-entry* [name]
           +--rw name
           +--rw reqid?
           +--rw ipsec-sa-config
           ...
  notifications:
    +---n sadb-acquire
    +---n sadb-expire
    +---n sadb-seq-overflow
    +---n sadb-bad-spi
```

$\Rightarrow$

```
module: ietf-ipsec-common

  +--rw ipsec-common
     +--rw spd
     |  +--rw spd-entry* [name]
     |     +--rw name
     |     +--rw direction?
     |     +--rw reqid?
     |     ...
     +--rw sad
        +--rw sad-entry* [name]
           +--rw name
           +--rw reqid?
           +--rw ipsec-sa-config
           ...
```

SDN IPsec proposed changes (IKE)

```
module: ietf-ipsec-ike

  +--rw ipsec-ike
     +--rw pad
     |  +--rw pad-entry* [name]
     |     +--rw name
        ...
     +--rw conn-entry* [name]
     |  +--rw name
     |  +--rw local
     |  |  +--rw local-pad-entry-name?
     |  +--rw remote
     |  |  +--rw remote-pad-entry-name?
        ...
     |  +--rw spd
     |  |  +--rw spd-entry* [name]
     |  |     +--rw name
     |  |     +--rw ipsec-policy-config
     |  |        +--rw anti-replay-window?
     |  |        +--rw traffic-selector
                 ...
     |  +--rw child-sa-info
     |  |  +--rw pfs-groups*
     |  |  +--rw child-sa-lifetime-soft
     |  |  +--rw child-sa-lifetime-hard
```

```
module: ietf-ipsec-ike

  +--rw ipsec-ike
     +--rw pad
     |  +--rw pad-entry* [name]
     |     +--rw name
        ...
     +--rw conn-entry* [name]
     |  +--rw name
     |  +--rw local
     |  |  +--rw local-pad-entry-name?
     |  +--rw remote
     |  |  +--rw remote-pad-entry-name?
        ...
     |  +--rw spd
     |  |  +--rw spd-entry* [leaf-list references to common spd]
     |  +--rw child-sa-info
     |  |  +--rw pfs-groups*                pfs-group
     |  |  +--sad-entry [reference to common sad entry]
```

IP-TFS YANG post changes

- IP-TFS config augments ipsec-common SPD entry
  - Previously under ike:conn-entry/ike:spd-entry
  - Previously under ikeless:spd/ikeless:spd-entry
- IP-TFS oper-config augments ipsec-common SAD entry
  - Previously under ike:conn-entry/ike:child-sa-info
  - Previously under ikeless:sad/ikeless:sad-entry
- IP-TFS oper-statistics augments ipsec-common SAD entry
  - Previously not available under ike
  - Previously under ikeless:sad/ikeless:sad-entry
- IP-TFS oper-statistics augment child-sa-info
  - For aggregate statistics
  - Same as before

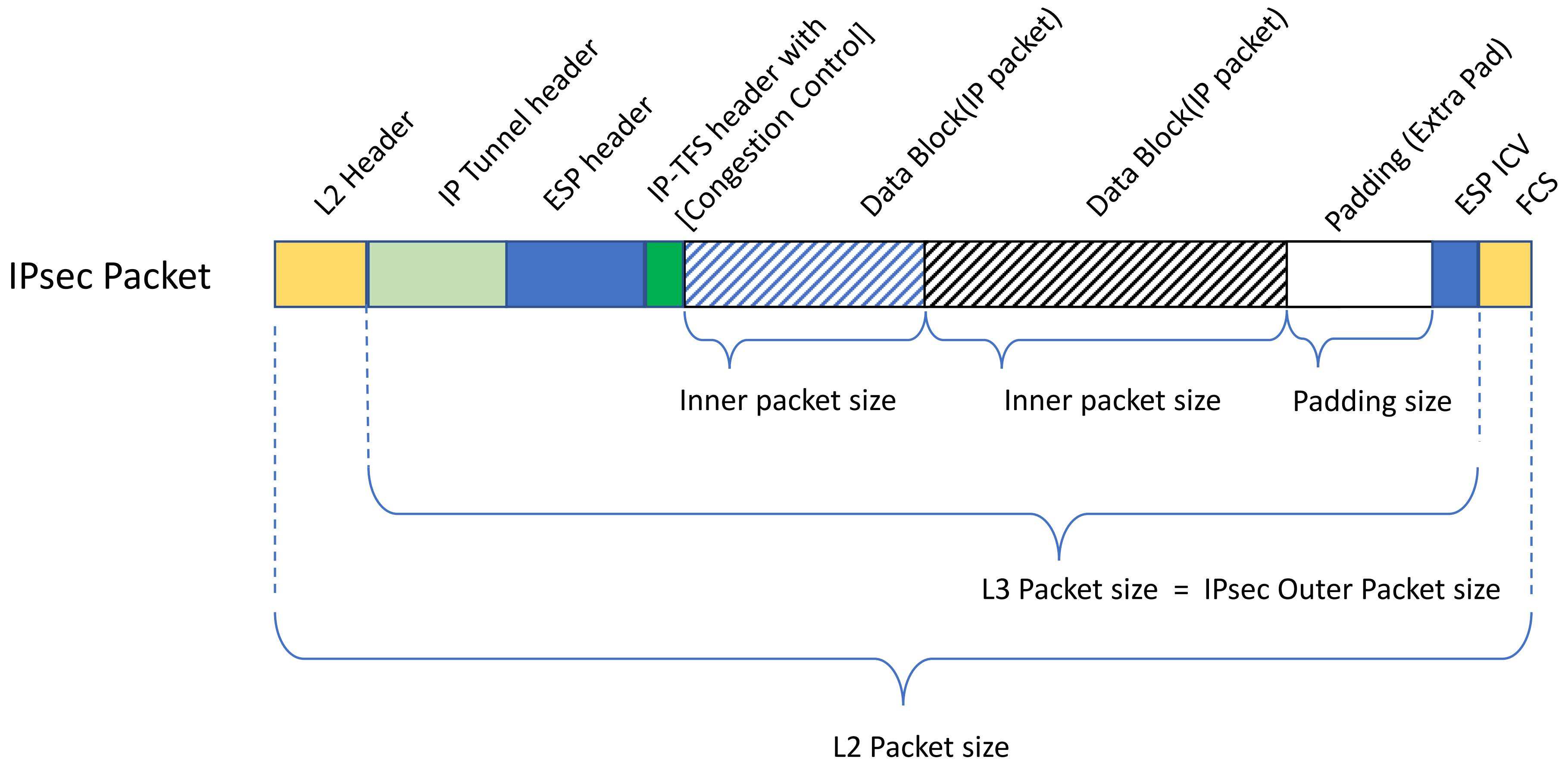
Comments / Questions?

Backup Slides

Context: IPsec Traffic Flow Security (IP-TFS)

- Provide Configuration Control and Statistics for IP-TFS
  - https://tools.ietf.org/html/draft-ietf-ipsecme-iptfs-01
- TFS in a Nutshell
  - Uses Packet Confidentiality of Tunnel Mode
  - Adds fixed size packets with aggregation and padding
  - Adds fixed transmission interval
  - Can be run with Congestion control
  - Provides Aggregation of inner packets
  - Utilizes Fragmentation of inner packets for efficiency
  - Tunnel Ingress controls packet format and frequency
    - A Self describing data block format allows sender traffic pattern flexibility

IP –TFS Tunnel Mode Packets - Summary



Open Discussion

- Other points of interest?