Scalable NIC inline crypto for hyperscale workloads

& OCP NIC software project update

Jakub Kicinski kicinski@fb.com
Willem de Bruijn willemb@google.com

Connect. Collaborate.
Accelerate.

OCP NIC software project

Bring together NIC vendors and users to agree on standard features

Selected topics

- Telemetry
- Traffic Engineering
- Flow Steering
- Time Measurement
- Crypto Offload: **QUIC & PSP > HW spec**

opencompute.org/wiki/Networking/NIC_Software

QUIC: intro and protocol

QUIC is widely used on edge w/ some inroads in the datacenter.

- no inter-packet crypto state (unlike TLS)
- some connection state (next and acked packet number)
- special key derivation and header protection process

```
Short Header Packet {
  Header Form (1) = 0,
  Fixed Bit (1) = 1,
  Spin Bit (1),
  Reserved Bits (2),         # Protected
  Key Phase (1),             # Protected
  Packet Number Length (2),  # Protected
  Destination Connection ID (0..160),
  Packet Number (8..32),     # Protected
  Protected Payload (0..24), # Skipped Part
  Protected Payload (128),   # Sampled Part
  Protected Payload (..),    # Remainder
}
```

QUIC: offload high level

- Linux implementation akin to TLS offload (Upper Layer Protocol)
- Support multiple connections on a single socket
- Install keys in the HW if capable
- Pass in connection parameters via control data
- Perform crypto while copying data or offload to HW
- Support UDP Segmentation Offload
- Device communication driver-specific

QUIC: offload requirements

- AES-GCM offload required, ChaCha-Poly nice to have
- Tx-only offload a viable option
  - Can be stateless (packet numbers passed in context)
- No sequential Packet Number requirement within Segmentation Offload super-frames
- Rx require packet number regeneration

PSP

Differences from IPSec ESP

Scalability: 10M+ active flows, 100K+ conn/s, stateless

Telemetry

Load balancing

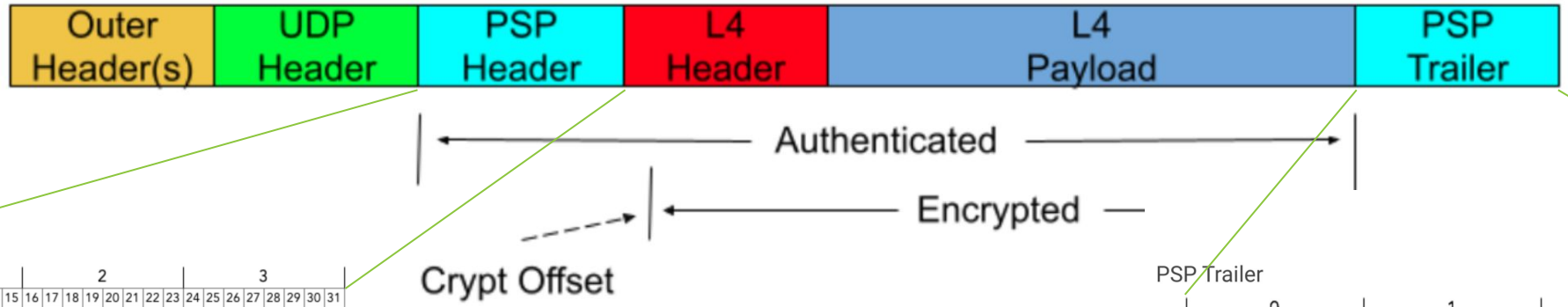
Minimal feature set

Architecture spec and preliminary src on github

Protocol

AES-GCM, FIPS

8B UDP + 16B PSP header + 16B ICV trailer



PSP Header Version 0

	0								1								2								3							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Next Header								Hdr Ext Len								R	Crypt Offset							S	D	Version				V	1
1	Security Parameters Index (SPI)																															
2	Initialization Vector (IV) 63:32																															
3	Initialization Vector (IV) 31:0																															
4	Virtualization Cookie (VC) [Optional] 63:32																															
5	Virtualization Cookie (VC) [Optional] 31:0																															

PSP Trailer

	0								1								2								3							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Integrity Checksum Value (ICV) 127:96																															
1	Integrity Checksum Value (ICV) 95:64																															
2	Integrity Checksum Value (ICV) 63:32																															
3	Integrity Checksum Value (ICV) 31:0																															

Key management

Storage

- 10M connections is 256B * 2* 10M = 5GB

Performance

- insertion
- removal
- tail latency SLOs

Key management: stateless offload

Rx Derived session keys

- hidden master key, SA key (KDF-CM), per-packet IV: SPI + picosec tstamp

Key Rotation

- IV and SPI overflow
- 2 master keys
- notify processes & sessions

Tx Key lookup

- stateful: key table index as descriptor field
- stateful: flow matching
- [preferred] stateless: key as descriptor field

wireline protocol: agnostic

Implementing PSP

SPI to connection mapping

Monitoring & telemetry

- crypt_off (4B)
- device counters
- detecting cleartext

Software fallback

- no TSO, FPU

OS quirks: bonding, ipvlan, sk2dev mapping

Timestamp

Offload

.ndo Driver API

Prerequisite: segmentation offload

- replicate tunnel headers
- protocol independent tunnel segmentation offload:
  - future protocols and optional variants
- GSO_PARTIAL: fixed length field: two packets

Encrypt: write unique IV + ICV after segmentation

Crypto offload: OCP draft spec

Robust: standardize shared mechanisms (AES-GCM, TSO) instead of protocols

Cover all common protocols: QUIC, PSP, IPSec, TLS, WireGuard, ..

Shared features

- Algorithms: AES-GCM, ChaChaPoly1305
- Key management
- Scalability & Performance
- Telemetry
- Tunneling & Segmentation offload

Join the effort

Bring your own ideas, critique others, share deployment experiences and help define the ecosystem.

Proposals for other features actively invited.
Whether in planning, development or already deployed.

mailing list: ocp-all.groups.io/g/OCP-Networking

meetings: 2nd Monday of the month, 10am PT (info, link)

wiki: opencompute.org/wiki/Networking/NIC_Software