Konstantin Taranov, Benjamin Rothenberger, Adrian Perrig, Torsten Hoefler

sRDMA – Efficient NIC-based Authentication and Encryption for Remote Direct Memory Access

RDMA networking is a new trend in cloud computing

Alibaba Builds High-Speed RDMA Network for AI and Scientific Computing

Alibaba Clouder June 3, 2019 19,036 0

Alibaba has built the RDMA high-speed network within its global and ultra-large data centers to support AI and scientific computing.

Why Oracle Deployed RoCE Network to serve AI & HPC workloads in Oracle Cloud Infrastructure (OCI)

November 12, 2018 Motti Beck

200 Gigabit HDR InfiniBand Boosts Microsoft Azure High-Performance Computing Cloud Instances

The HDR InfiniBand Connected Virtual Machines Deliver Leadership-Class Performance, Scalability, and Cost Efficiency for a Variety of Real-World HPC Applications

November 18, 2019 12:00 PM Eastern Standard Time

How RDMA Became the Fuel for Fast Networks

The networking technology that feeds the world's largest supercomputers and data centers spawned one of this year's top tech mergers and now it drives AI.

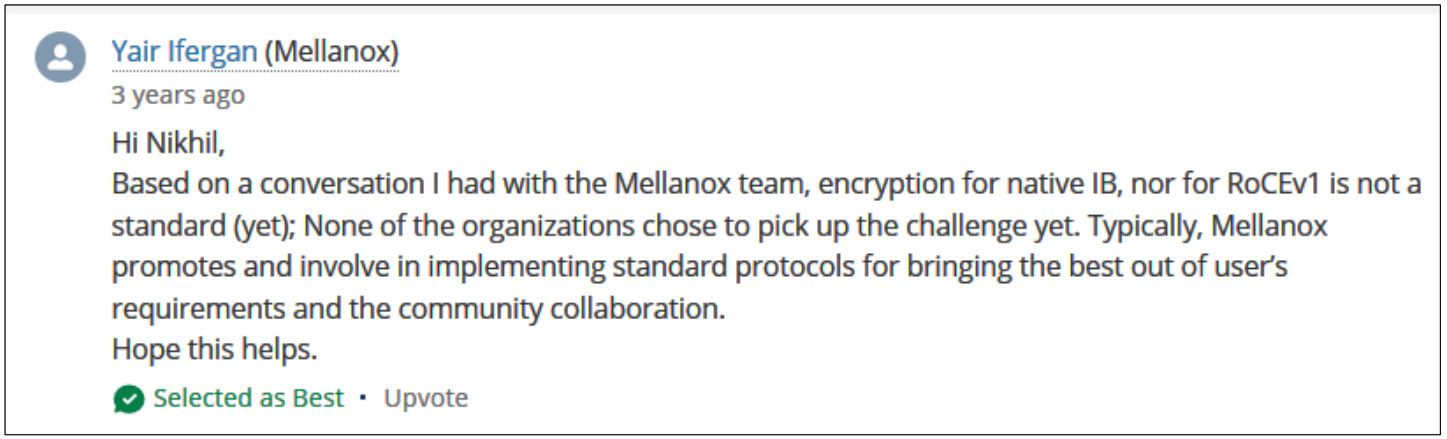
April 29, 2020 by RICK MERRITT

RDMA security considerations

RFC4297 – Remote Direct Memory Access (RDMA) over IP Problem Statement:

"The RDMA protocols must permit integration with Internet security standards, such as IPsec and TLS. "

December 2005



Yair Ifergan (Mellanox)
3 years ago

Hi Nikhil,
Based on a conversation I had with the Mellanox team, encryption for native IB, nor for RoCEv1 is not a standard (yet); None of the organizations chose to pick up the challenge yet. Typically, Mellanox promotes and involve in implementing standard protocols for bringing the best out of user's requirements and the community collaboration.
Hope this helps.

Selected as Best • Upvote

March 2017

IPSec does not support RDMA

July 2020

Can application-level security be used?

- **One-sided RDMA requests are completely performed by the NIC**
  - No CPU involvement on the destination machine

- **Two-sided communication is also offloaded to the NIC**
  - Packets cannot be discarded by the NIC
  - Received data consumes resources of the connection
  - CPU is responsible for verifying the received data negating RDMA advantages

sRDMA – secure RDMA communication

- **sRDMA is lightweight security extension to RDMA which uses symmetric key cryptography to provide**
  - Header Authentication
  - Packet Authentication
  - Payload encryption
  - Memory protection

- **sRDMA effectively prevents:**
  - Eavesdropping
  - Spoofing attacks
  - Replay attacks
  - Man in the middle attacks

- **sRDMA is back compatible with classical RDMA and can be easily adapted by**
  - native InfiniBand
  - RoCEv1
  - RoCEv2

sRDMA – secure QP connection

- **sRDMA introduces a new Secure Reliably Connected Queue Pair**
  - The application installs symmetric keys to a QP connection and required level of protection

```
// Secure attributes are the key and the security code
struct ibv_secure_attr {
  enum ibv_qp_crypto qp_crypto; // crypto type
  uint8_t sym_key[MAX_KEY_LENGTH];    // symmetric key
};
```

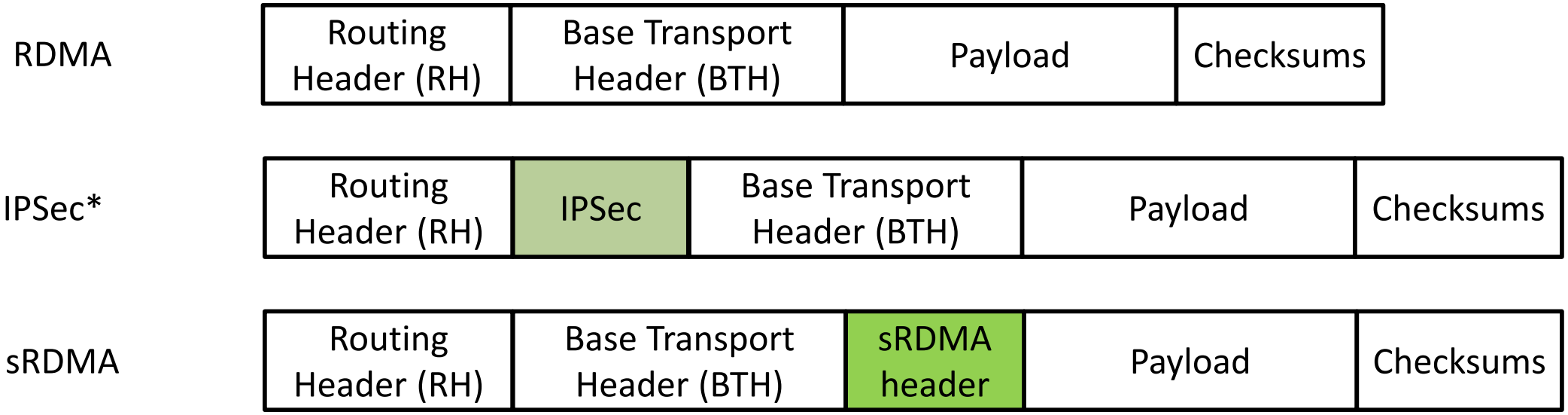
  - Supported security codes:

```
/*Message authentication codes */
  /* Hash-based MACs*/
   /* Header authentication*/
   IBV_HDR_HMAC_SHA1_160 = 0x1001, //EVP_sha1
   IBV_HDR_HMAC_SHA2_256 = 0x1003, //EVP_sha256
   IBV_HDR_HMAC_SHA2_512 = 0x1005, //EVP_sha512
   /* Packet authentication*/
   IBV_PCKT_HMAC_SHA1_160 = 0x2001, //EVP_sha1
   IBV_PCKT_HMAC_SHA2_256 = 0x2003, //EVP_sha256
   IBV_PCKT_HMAC_SHA2_512 = 0x2005, //EVP_sha512
```

```
/* Cipher-based MACs*/
 /* Header authentication*/
 IBV_HDR_CMAC_AES_96    = 0x3001, //EVP_aes-128-ocb
 IBV_HDR_CMAC_AES_128   = 0x3002, //EVP_aes-128-ocb
 IBV_HDR_CMAC_AES_256   = 0x3004, //EVP_aes-256-ocb
 IBV_HDR_CMAC_POLY1305 = 0x3005, //EVP_chacha20_poly1305
 /* Packet authentication*/
 IBV_PCKT_CMAC_AES_96    = 0x4001, //EVP_aes-128-ocb
 IBV_PCKT_CMAC_AES_128   = 0x4002, //EVP_aes-128-ocb
 IBV_PCKT_CMAC_AES_256   = 0x4004, //EVP_aes-256-ocb
 IBV_PCKT_CMAC_POLY1305 = 0x4005, //EVP_chacha20_poly1305
```

```
/* Authenticated Encryption*/
  IBV_AEAD_AES_96    = 0x5001, //EVP_aes-128-ocb
  IBV_AEAD_AES_128   = 0x5002, //EVP_aes-128-ocb
  IBV_AEAD_AES_256   = 0x5004, //EVP_aes-256-ocb
  IBV_AEAD_POLY1305 = 0x5005, //EVP_chacha20_poly1305
```

sRDMA Packet format



sRDMA packet format advantages:

- Routing and checksums not affected
- Secure header is processed after processing of BTH

* It does not exist yet, but it is discussed

Base Transport Header (BTH)

- **Changes to BTH**
  - We use 3 out of 7 reserved bits from BTH to indicate the presence of the secure header

bits / bytes	31-24	23-16	15-8	7-0
0-3	OpCode	SE M Pad TVer	Partition Key	
4-7	F B Reserved 6	Destination QP		
8-11	A **Reserved 7**	PSN – Packet Sequence Number		

- **Secure header size**
  - sRDMA supports 7 different MAC sizes
  - Value 0 is for back-compatibility

Size (bits)	0	96	128	160	224	256	384	512
Value	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7

Nonce and Packet Sequence Number (PSN)

bits / bytes	31-24	23-16	15-8	7-0
0-3	OpCode	SE M Pad TVer	Partition Key	
4-7	F B Reserved 6	Destination QP		
8-11	A Reserved 7	PSN – Packet Sequence Number		

- **IPSec uses nonce against replay attacks**
  - Nonce must never be reused
  - Nonce can be predictable and be transmitted in clear
- **PSN is a part of BTH**
  - PSN is only 24 bit which get reused after 80 ms on modern network devices
  - Mellanox ConnectX-5 can send up to 200 million messages per second!
- **sRDMA extends InfiniBand PSN counters to 64 bits**
  - Both sender and receiver maintain 64-bit counters,
  - But they transmit 24 least significant bits (LSB).
  - As PSNs are ordered, the endpoints can recover 64 bit sequence number from 24 LSB using sliding window.

sRDMA - Authentication and Secrecy

- **Header Authentication**

$$mac_{hdr} = MAC_{K_{A,B}}(nonce_{A \to B} \parallel RH \parallel BTH)$$

- **Packet Authentication**

$$mac_{pck} = MAC_{K_{A,B}}(nonce_{A \to B} \parallel RH \parallel BTH \parallel PAYLOAD)$$

- **Payload authenticated encryption**
  - Nonce, RH, and BTH are passed as *Additional Authenticated Data*
  - *Payload is encrypted and sent instead of plaintext*

- **Overheads of AES-128 for *N* secure QP connections**

	Key overhead	Nonce counter	Header
IPSec	16B * *N*	16B * *N*	32B
sRDMA	16B * *N*	10B * *N*	16B

Improving memory overhead – Protection Domain (PD) level keys

- **In RDMA, QP connections are created inside PDs**
  - PD groups IB resources such as QP connections and memory regions that can work together.

- **sRDMA proposes to install a key (K_{PD}) to PD, and use this key to derive QP level keys**
  - We propose to install a single key per PD, and derive QP-level keys from the PD key.
  - The key is derived using pseudorandom function (PRF) based on adapter port addresses (APA) and QPN identifiers of the endpoints.

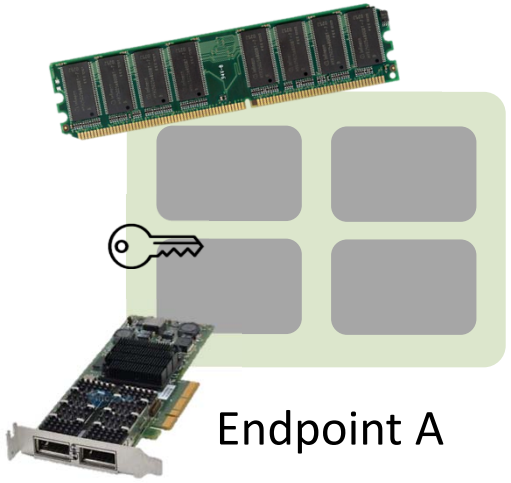
$$K_{A,B} = PRF_{K_{PD}}(APA_A \parallel QPN_A \parallel APA_A \parallel QPN_B)$$

  - Two endpoints derive the same symmetric key.
- **Overheads of AES-128 for *N* secure QP connections**

	Key overhead	Nonce counter	Header
IPSec	16B * *N*	16B * *N*	32B
sRDMA	16B * *N*	10B * *N*	16B
sRDMA + PD keys	16B	10B* *N*	16B

Extended memory protection

- **Memory protection in IBA is based on *rkey tags (32 bits)***
  - *Each one-sided RDMA request must include rkey in its request.*
  - *Any endpoint with the rkey can access the memory*

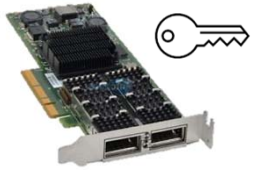


- **sRDMA proposes scalable crypto-based memory protection**
  - Access to sub-region (SR) with addresses [START, END)

$$K_{SR} = PRF_{K_{MR}}(START_{SR} \,||\, END_{SR})$$

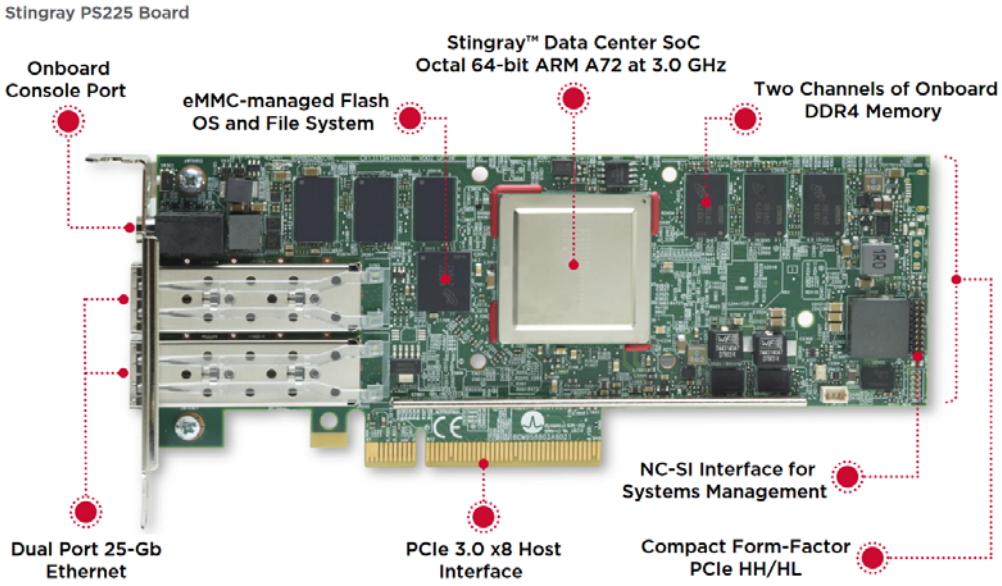
  - sRDMA does not introduce extra header and reuses the STH

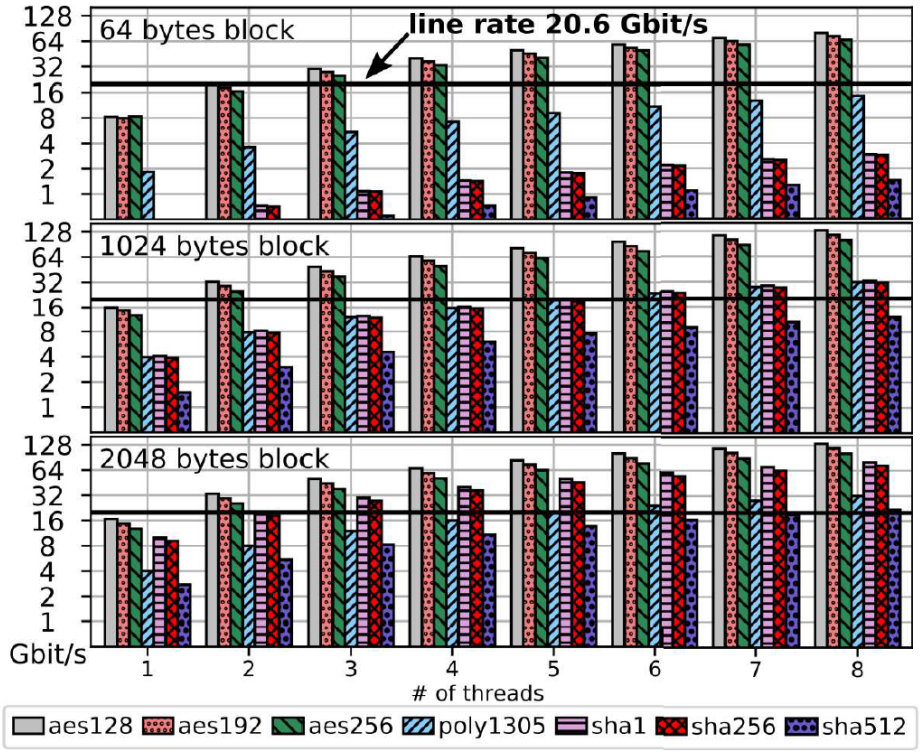
$$mac_{hdr} = MAC_{K_{A,B}}(K_{SR}||mac_{hdr})$$



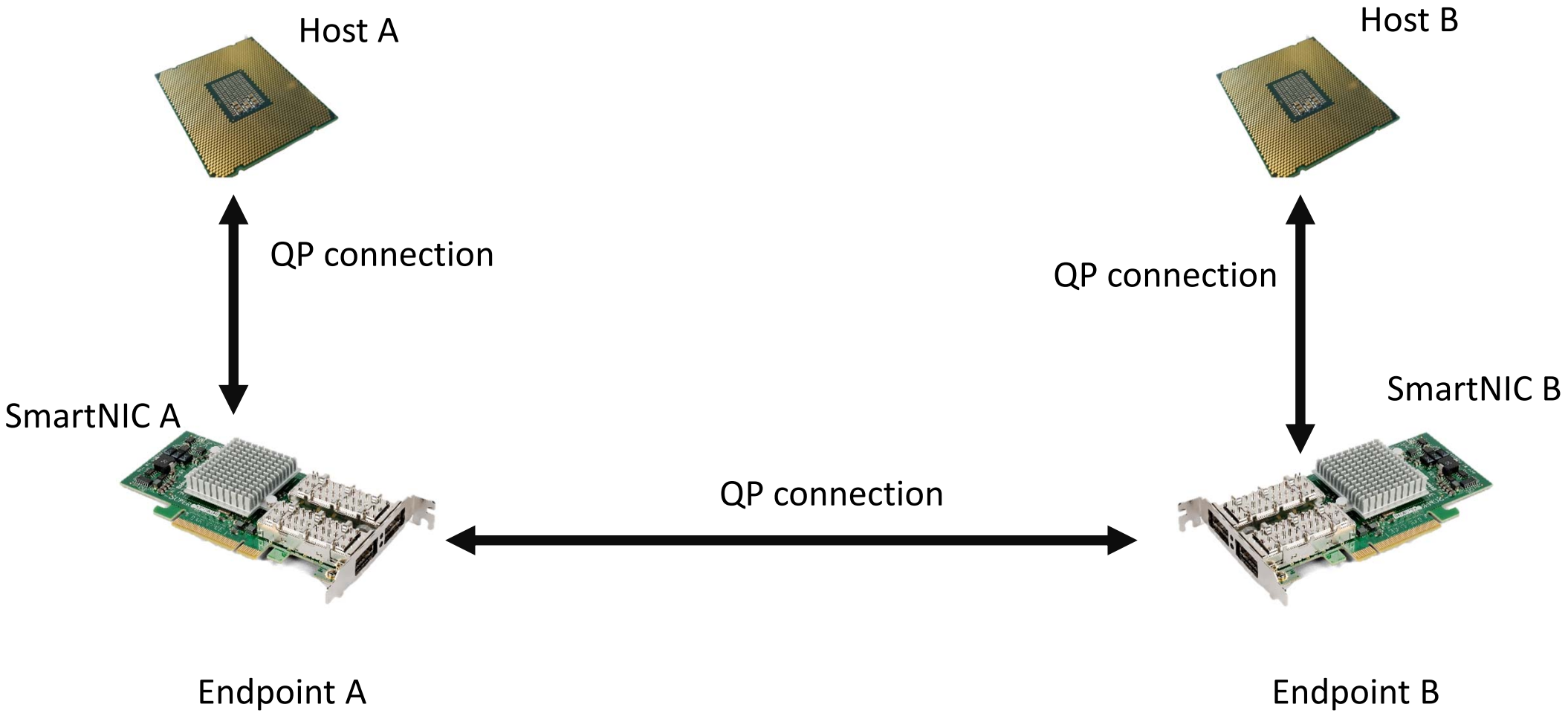
Implementation of sRDMA

- **sRDMA is implemented on Broadcom Stingray PS225**
  - Eight-core ARM A72
  - DDR4 8 GB DRAM
  - Supports crypto-acceleration

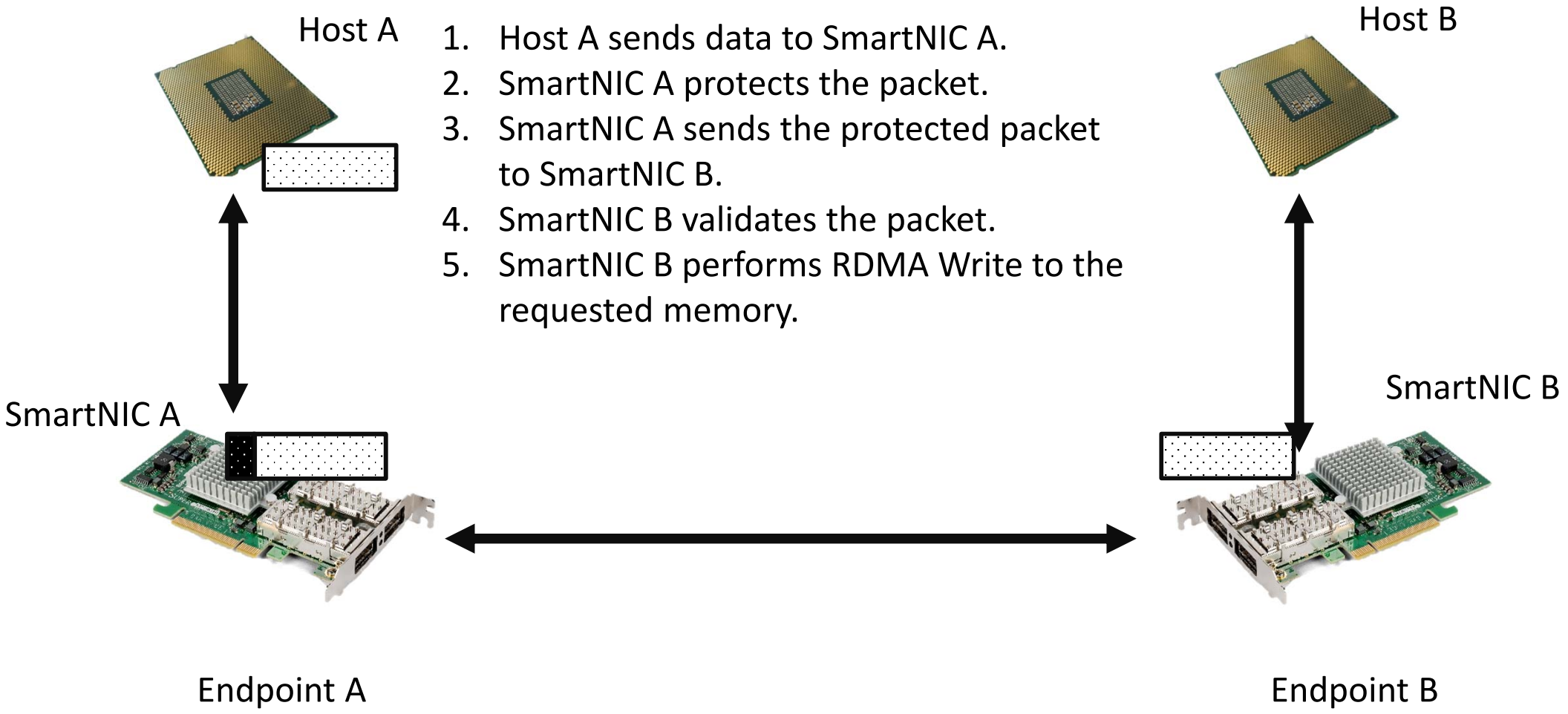




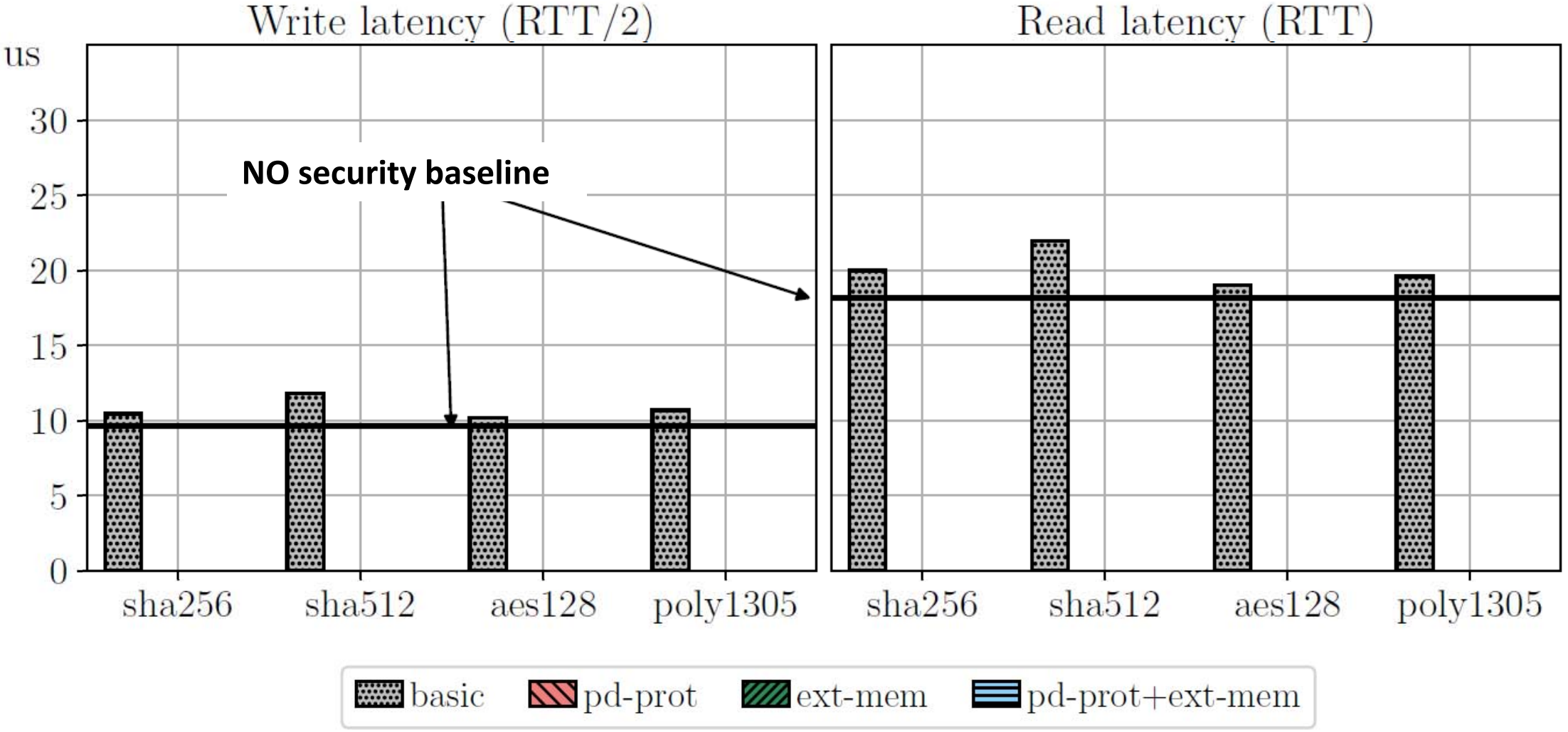
Implementation of sRDMA



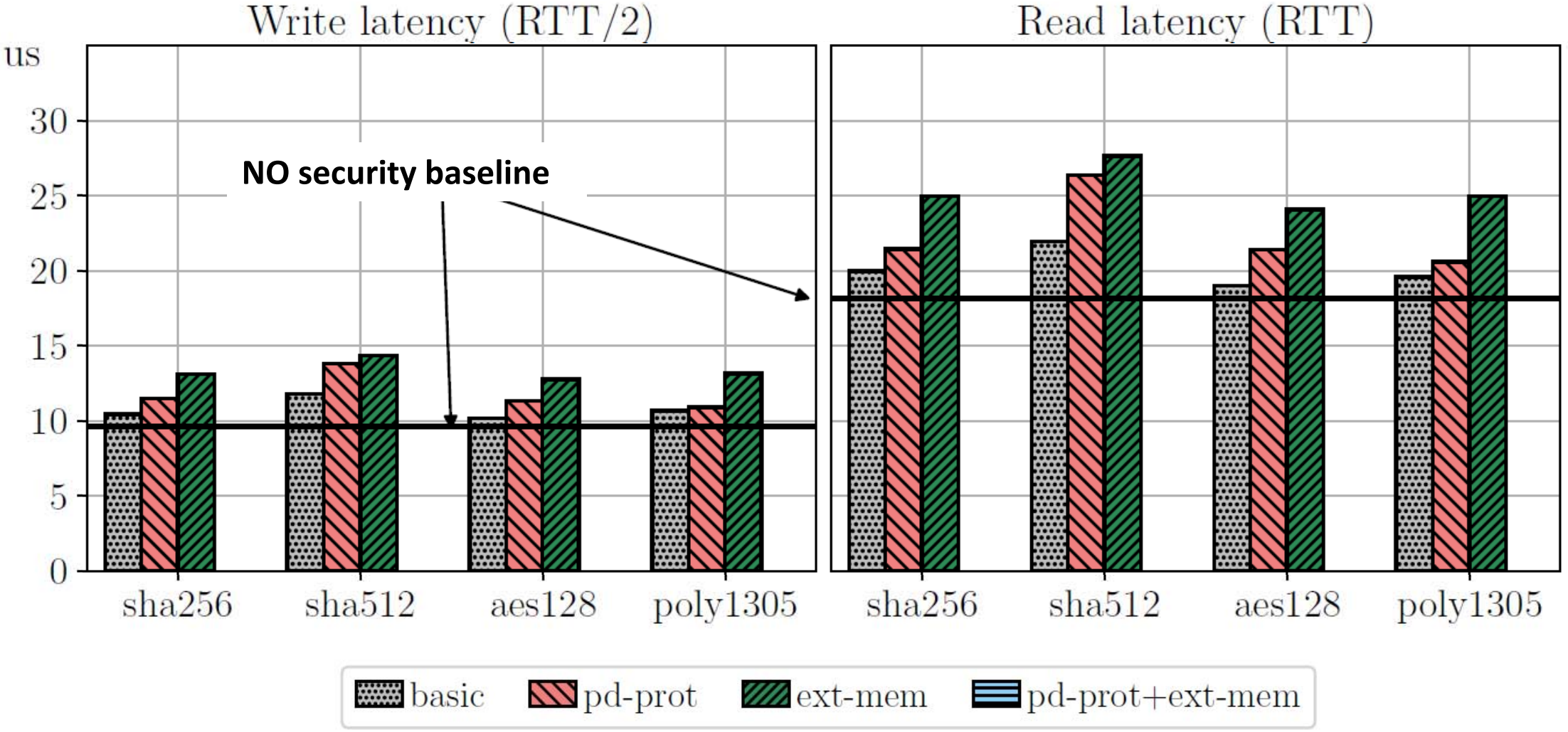
Implementation of sRDMA – RDMA Write



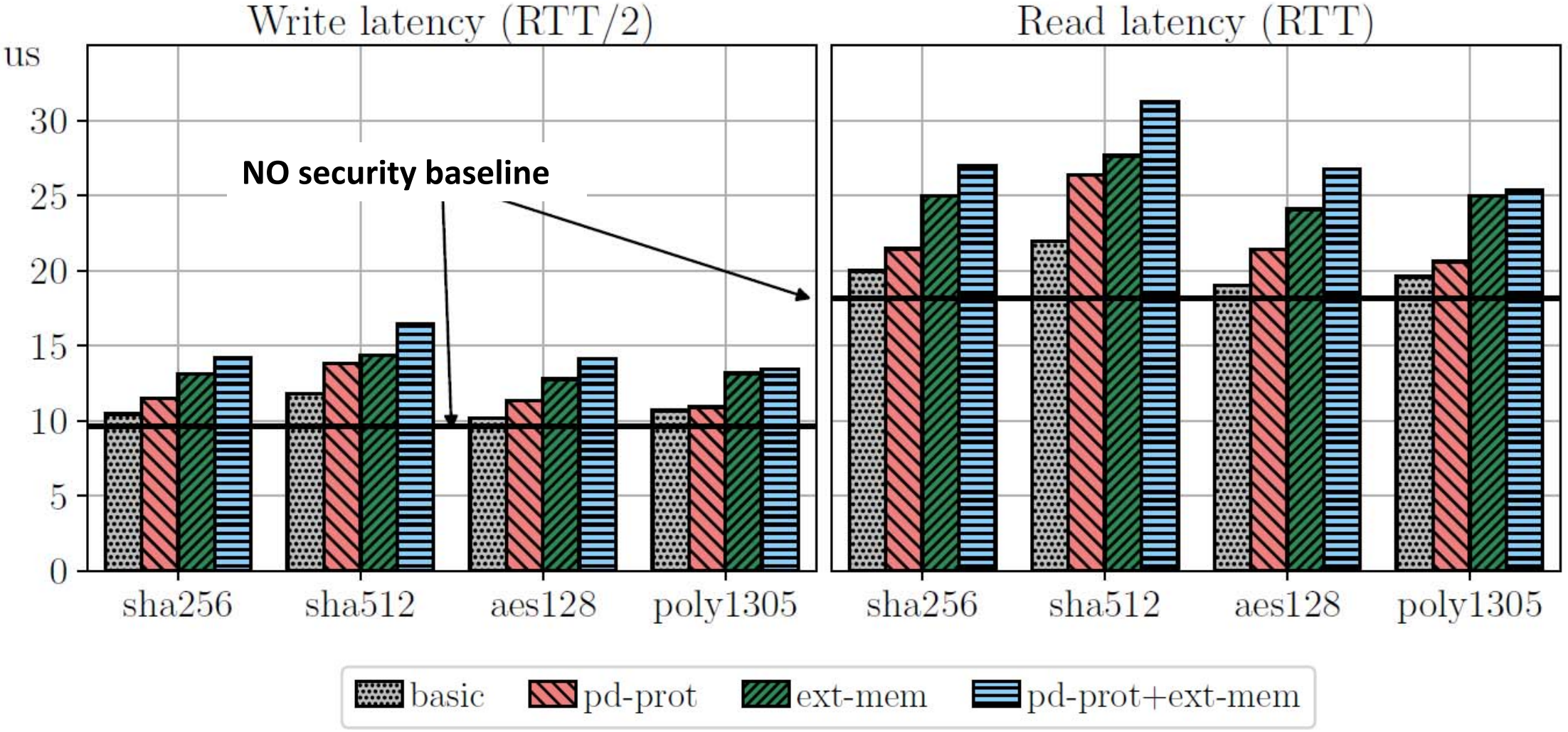
Evaluation – Source authentication latency



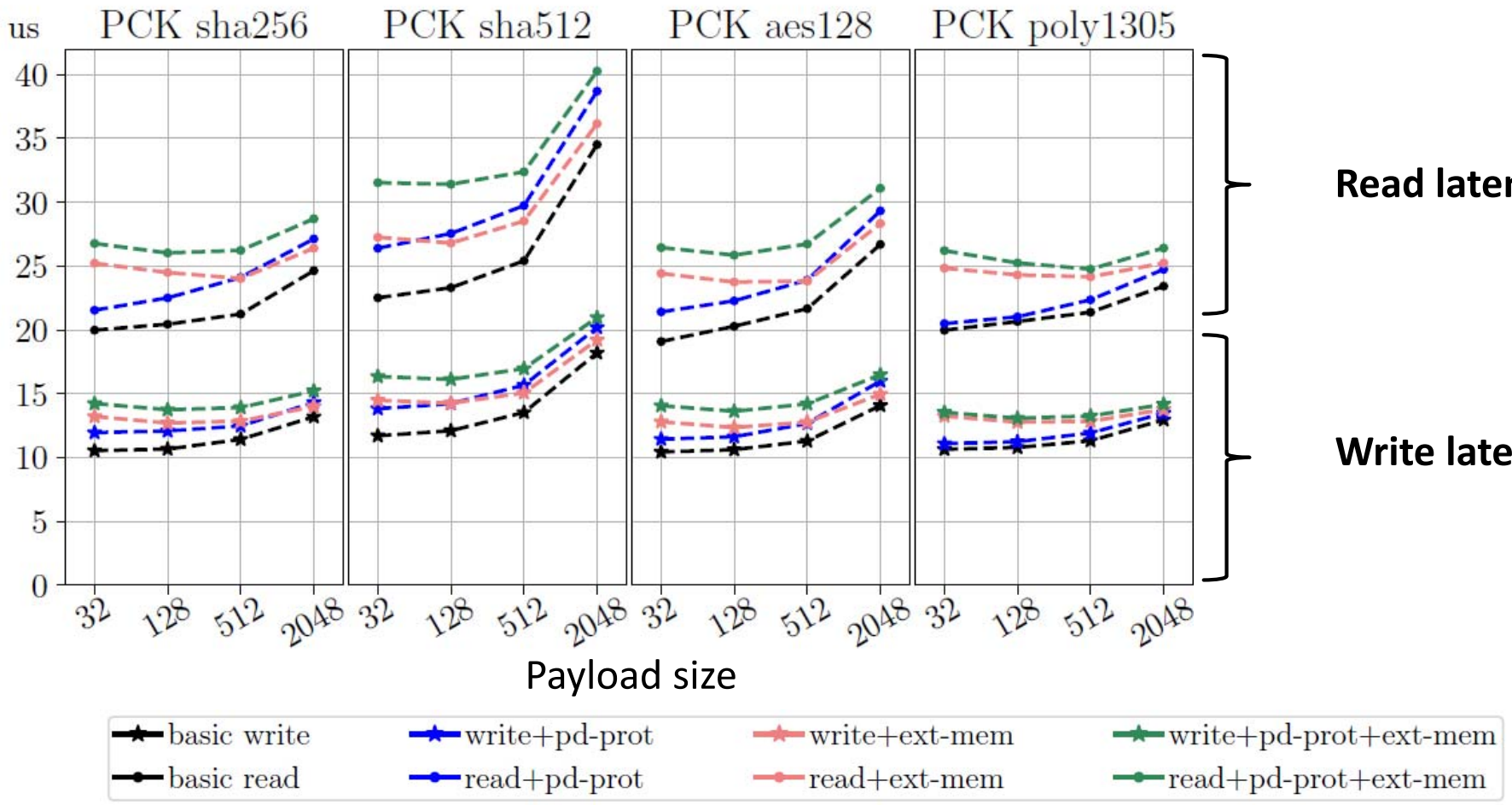
Evaluation – Source authentication latency



Evaluation – Source authentication latency



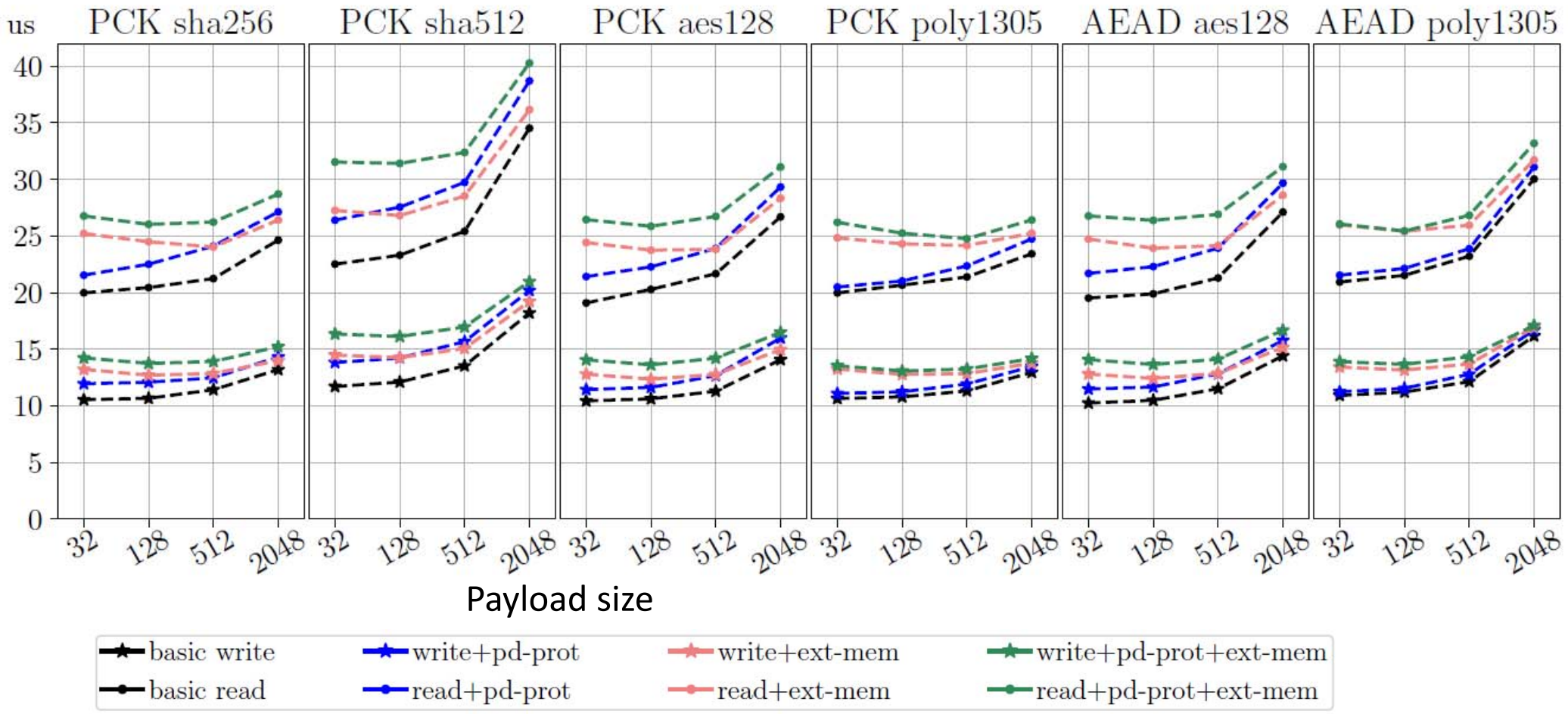
Evaluation – Packet authentication latency



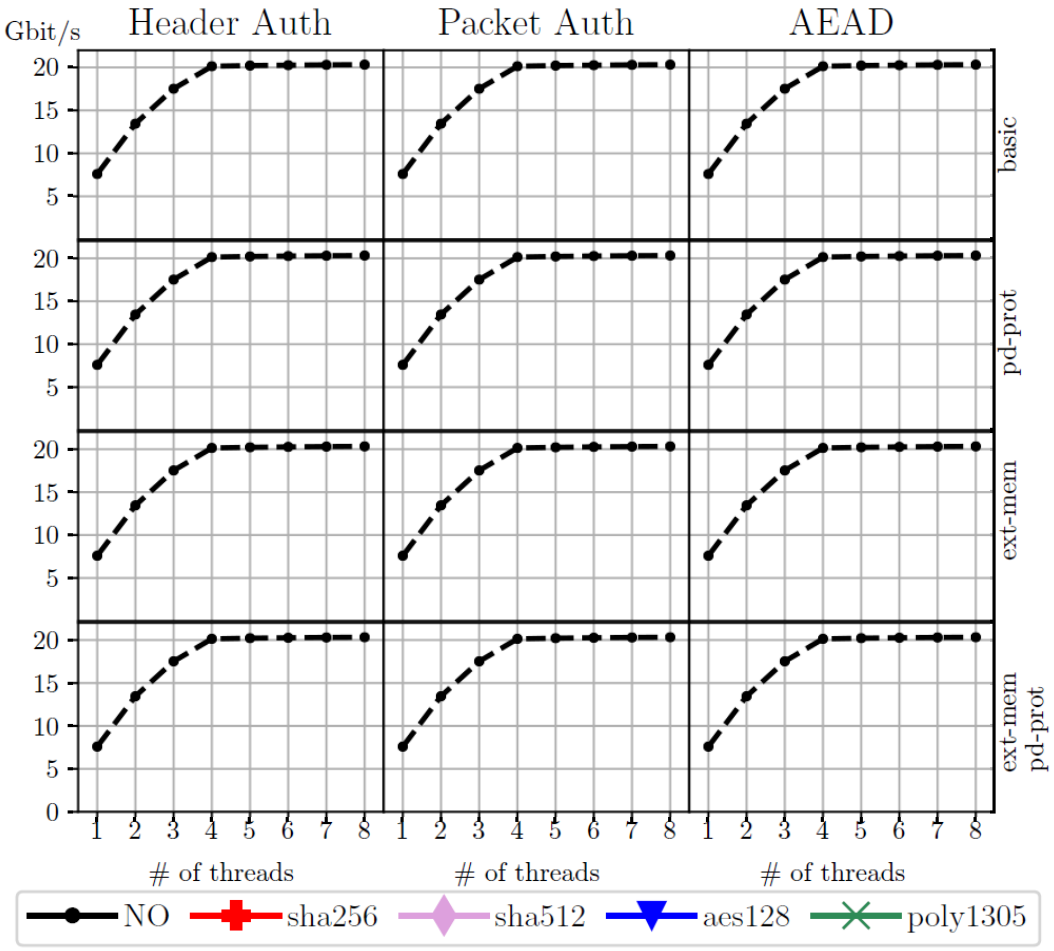
Read latency (RTT)

Write latency (RTT/2)

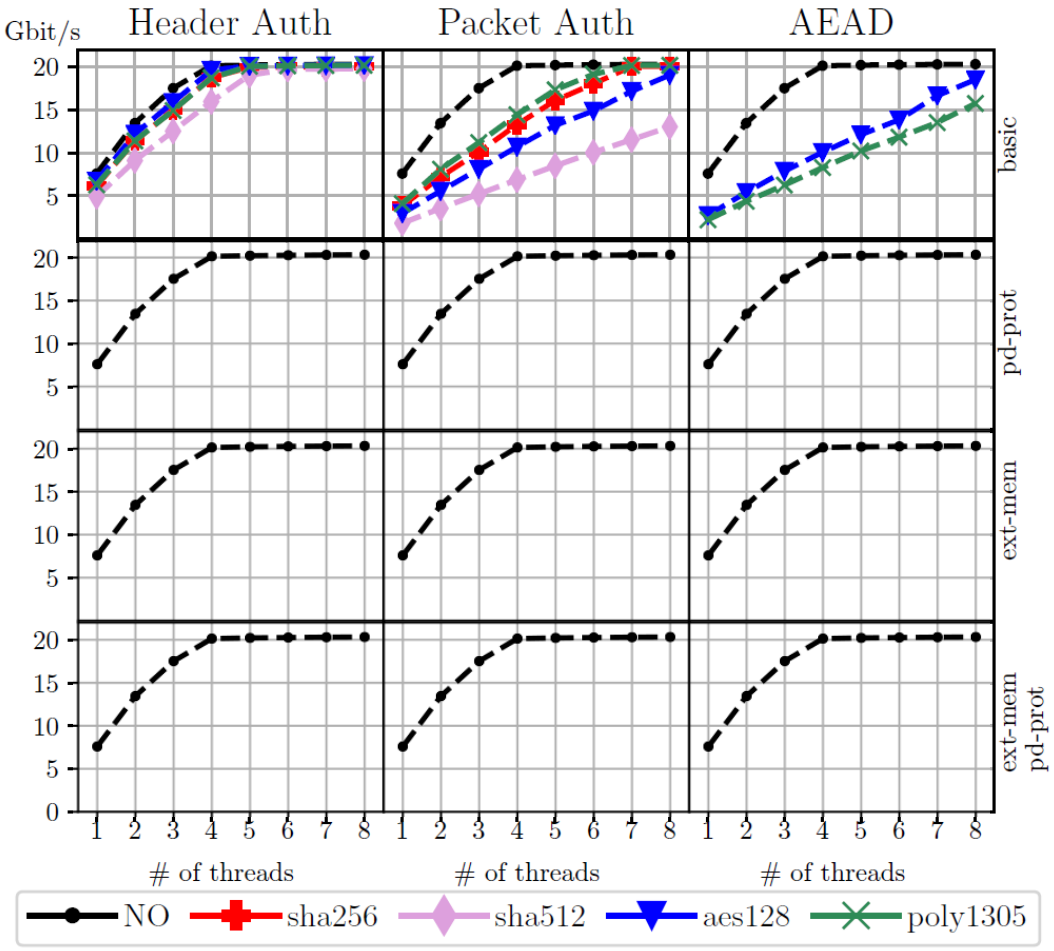
Evaluation – AEAD latency



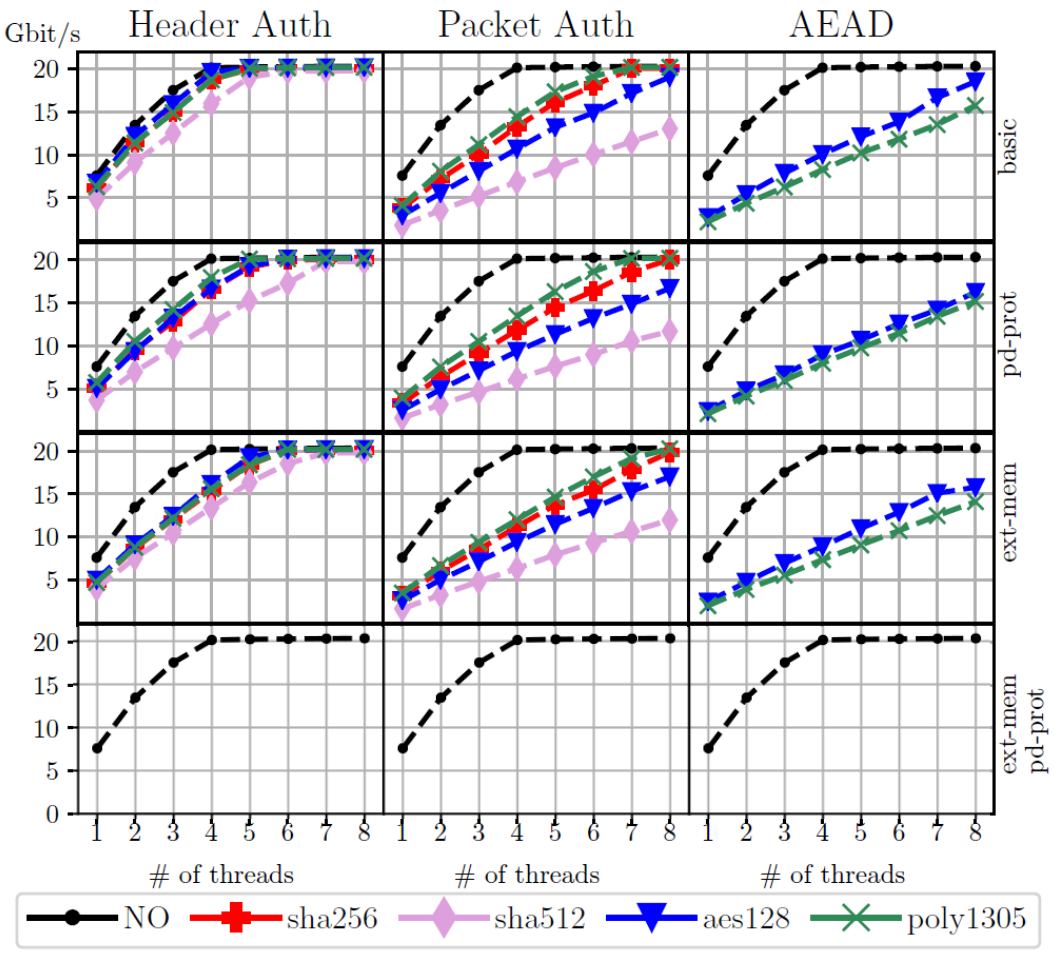
Evaluation – Write Bandwidth



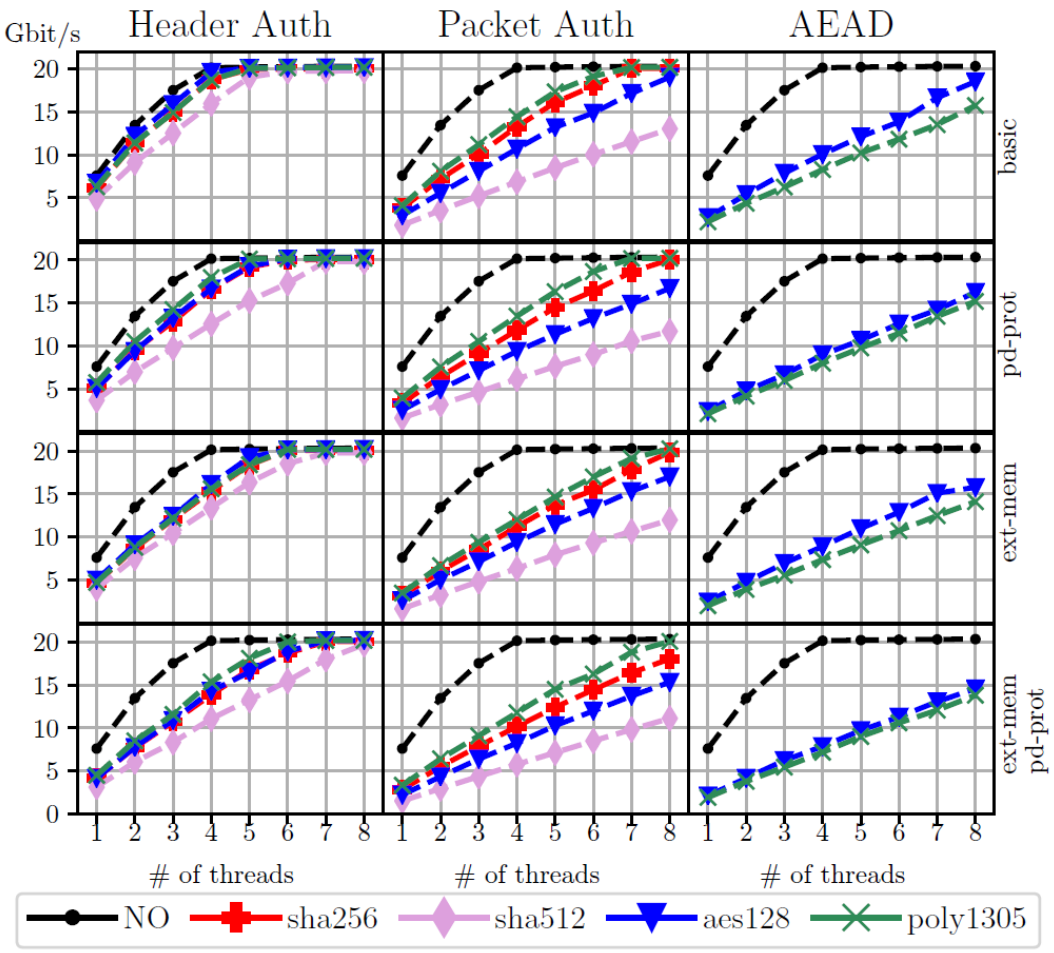
Evaluation – Write Bandwidth



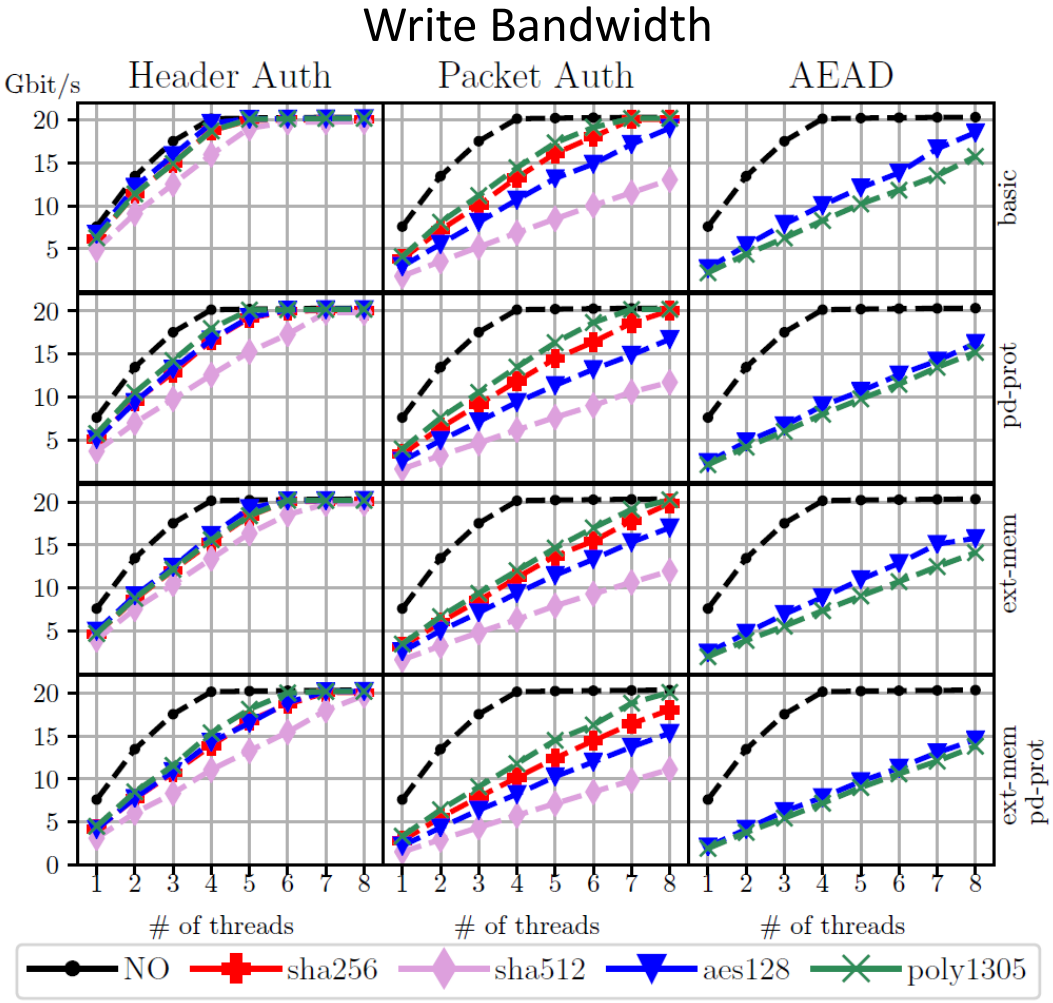
Evaluation – Write Bandwidth

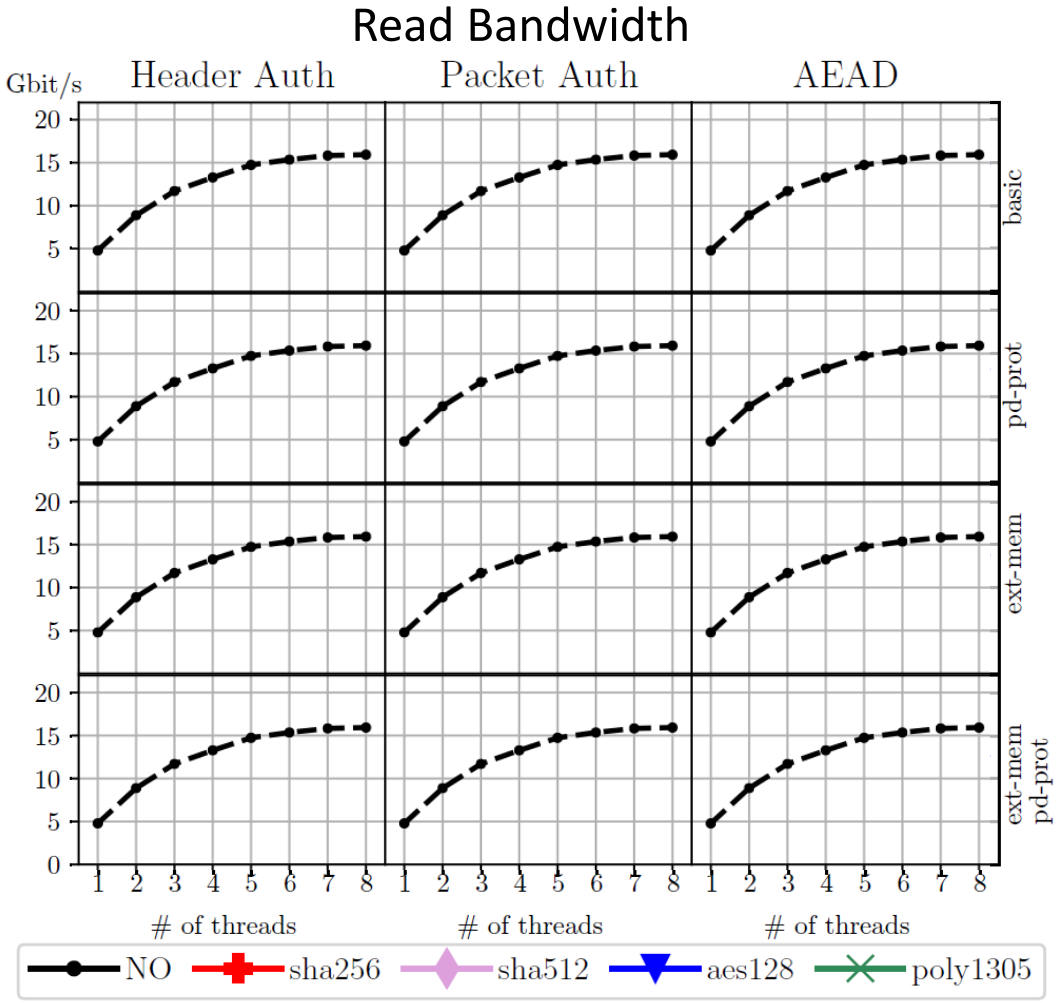


Evaluation – Write Bandwidth

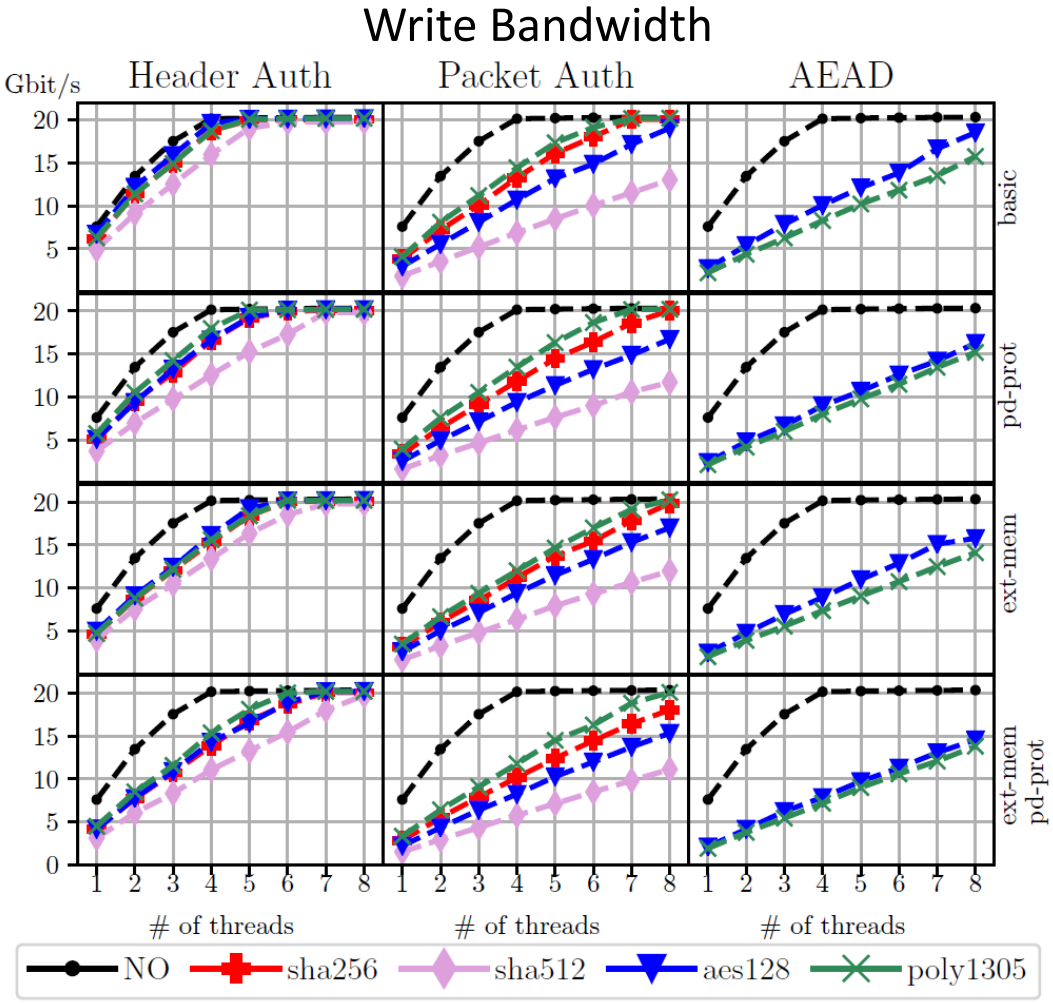


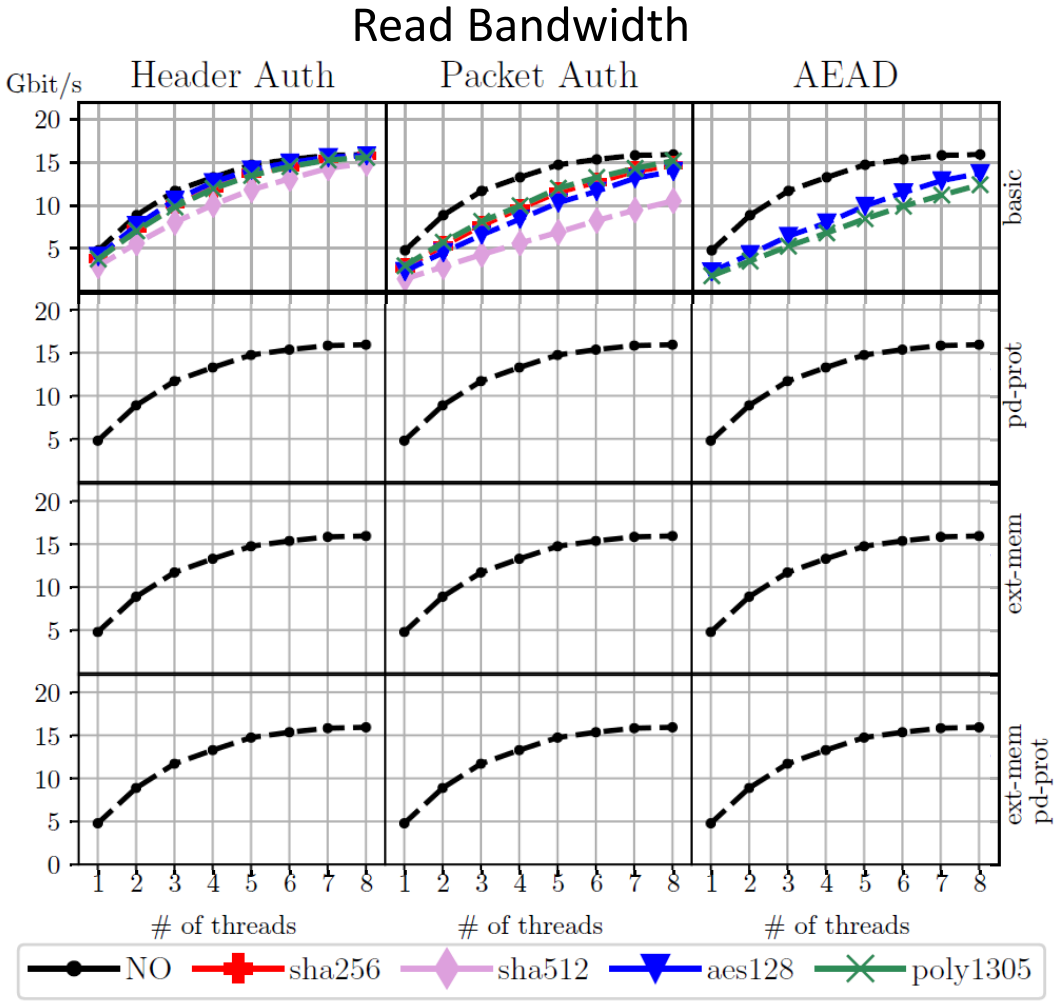
Evaluation – Read Bandwidth



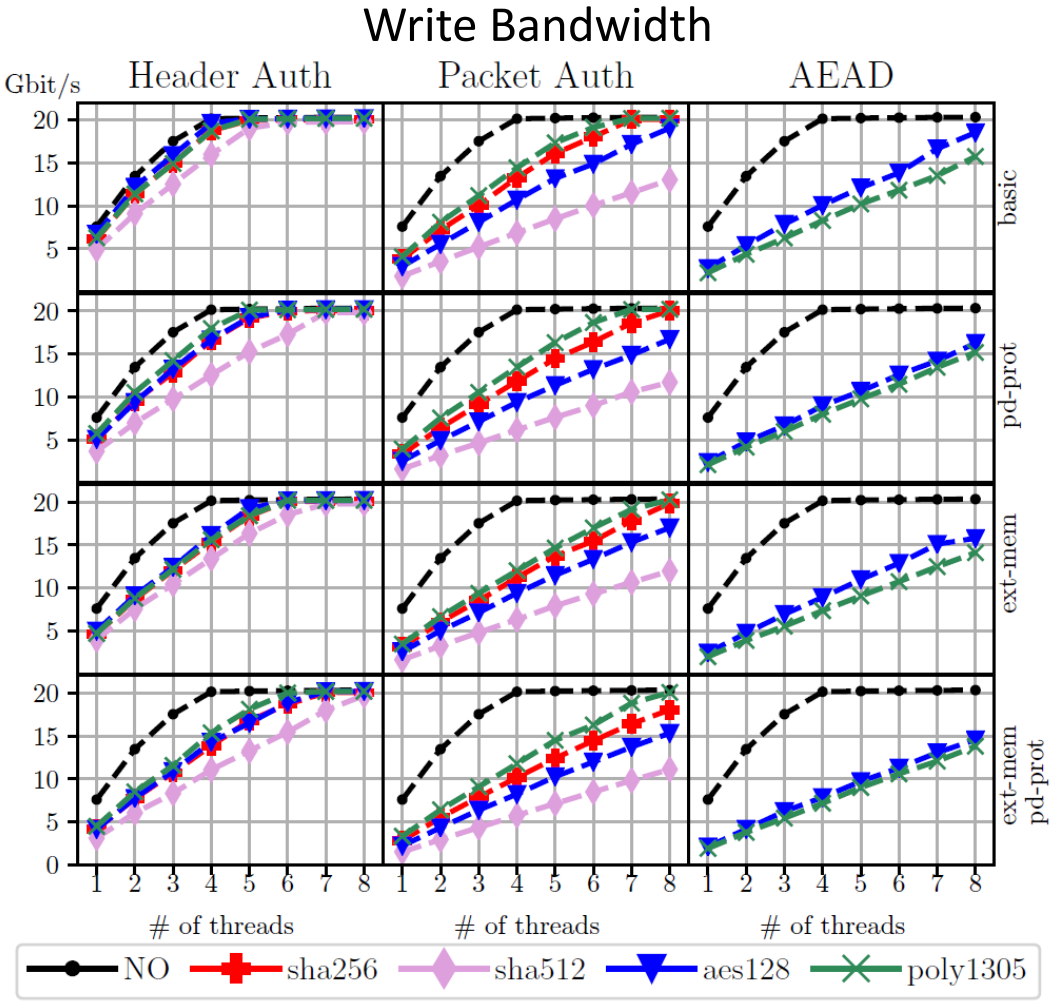


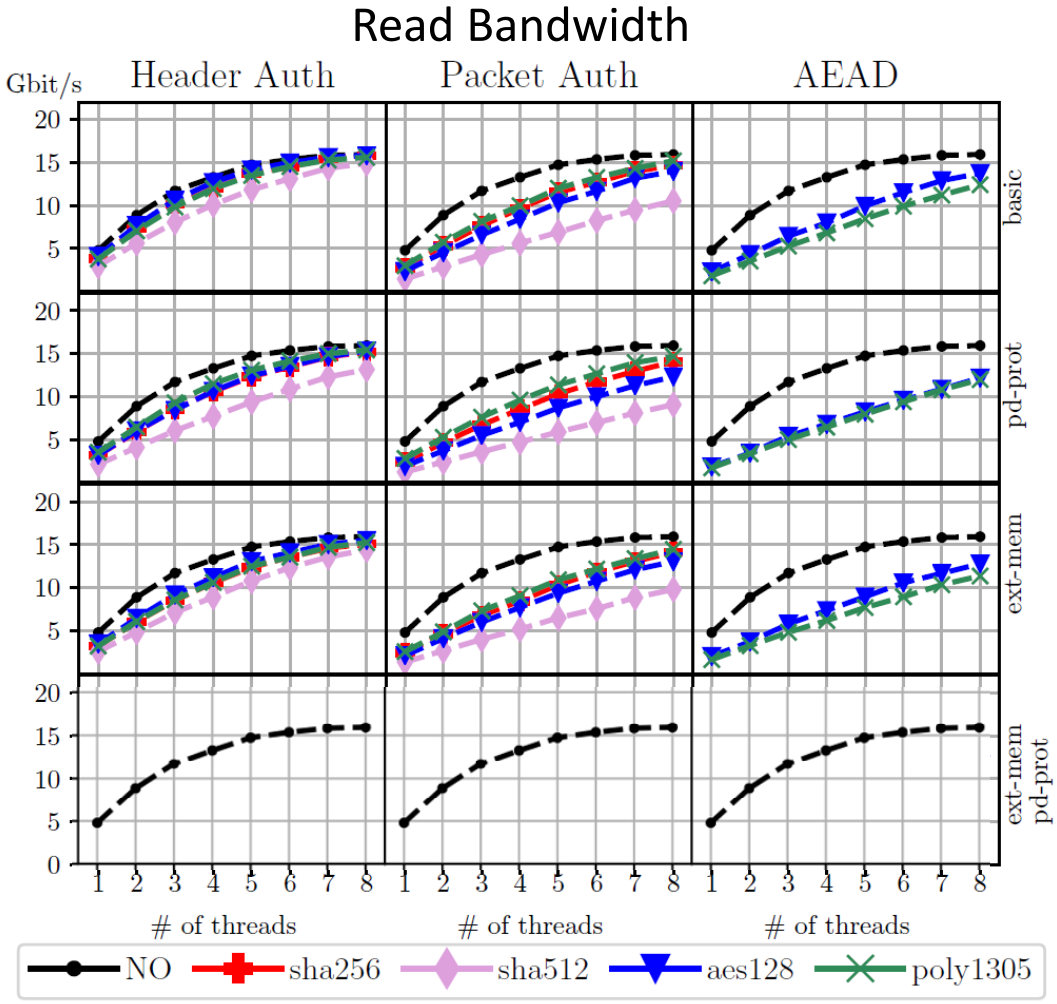
Evaluation – Read Bandwidth



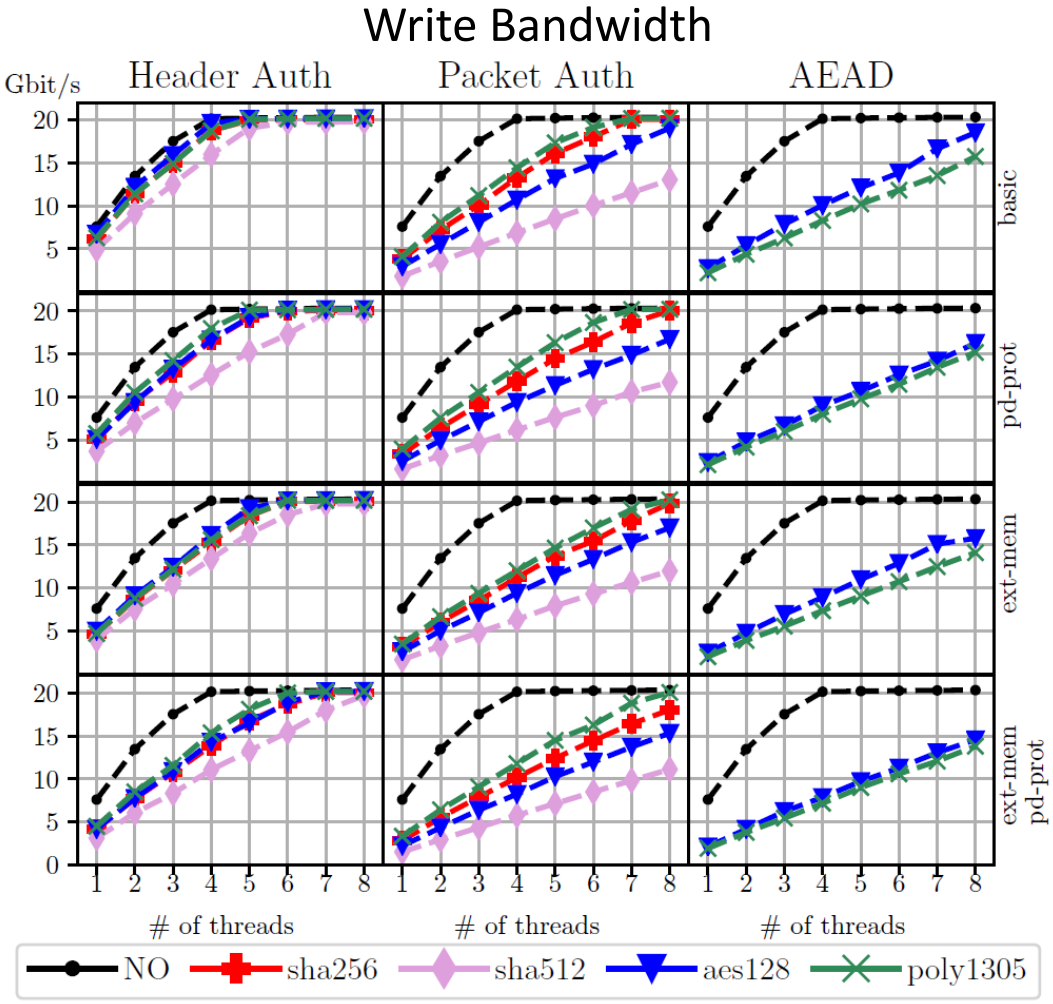


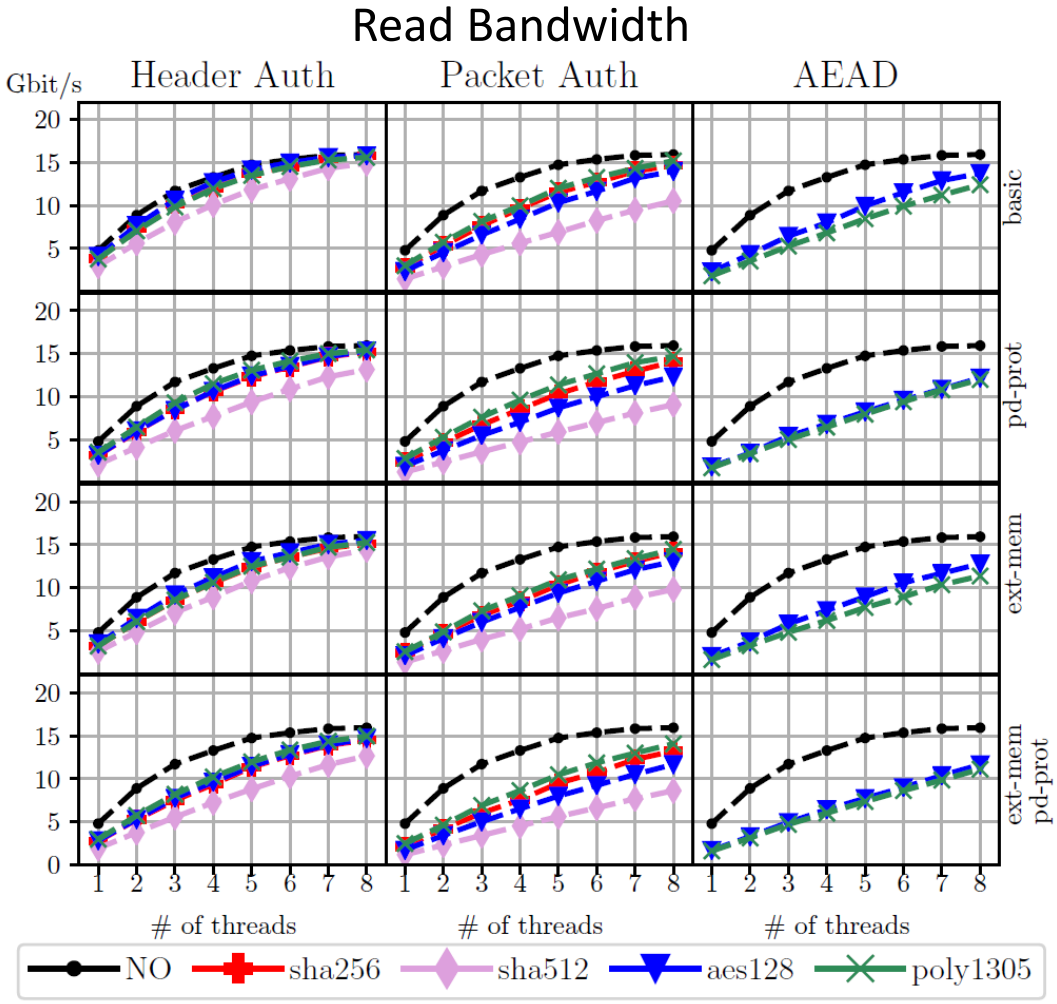
Evaluation – Read Bandwidth





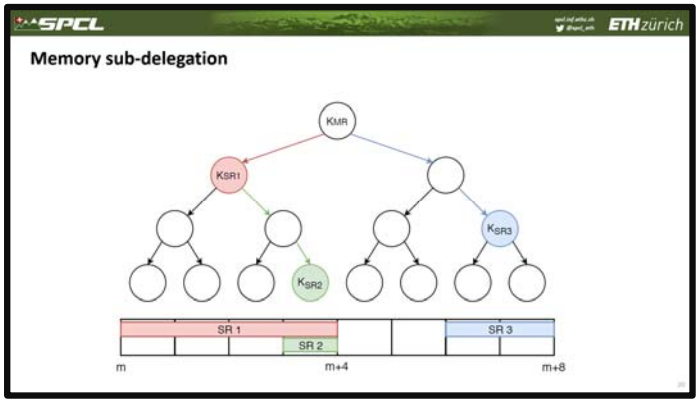
Evaluation – Read Bandwidth

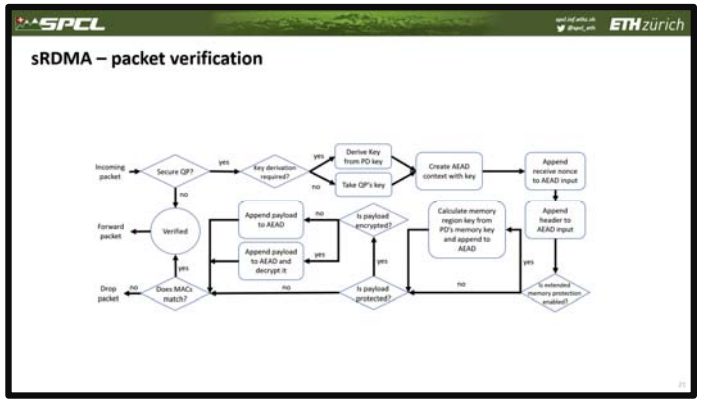


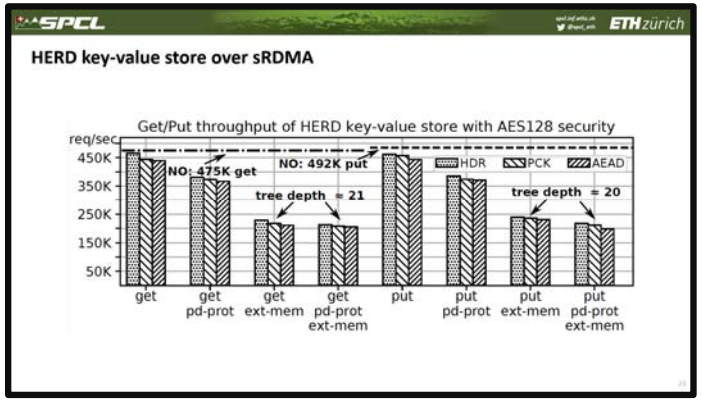


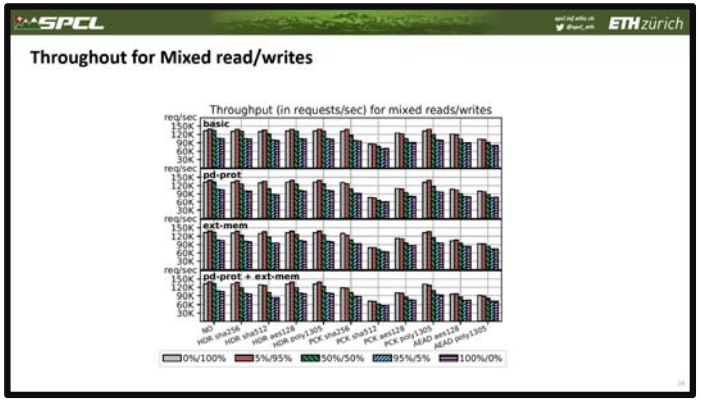
sRDMA paper also includes

- Memory Sub-delegation
- Details on the implementation
- Extra Experiments









Thank you for your attention!

- sRDMA is lightweight security extension to RDMA protocols
- sRDMA is flexible and supports various protection modes
- PD-level protection minimizes memory consumption on the NIC
- sRDMA extends memory protection of InfiniBand architecture
- sRDMA can be easily adapted to hardware

sRDMA implementation:

Network Security Group

Contact information:
Konstantin Taranov
konstantin.taranov@inf.ethz.ch