2021 OFA Virtual Workshop

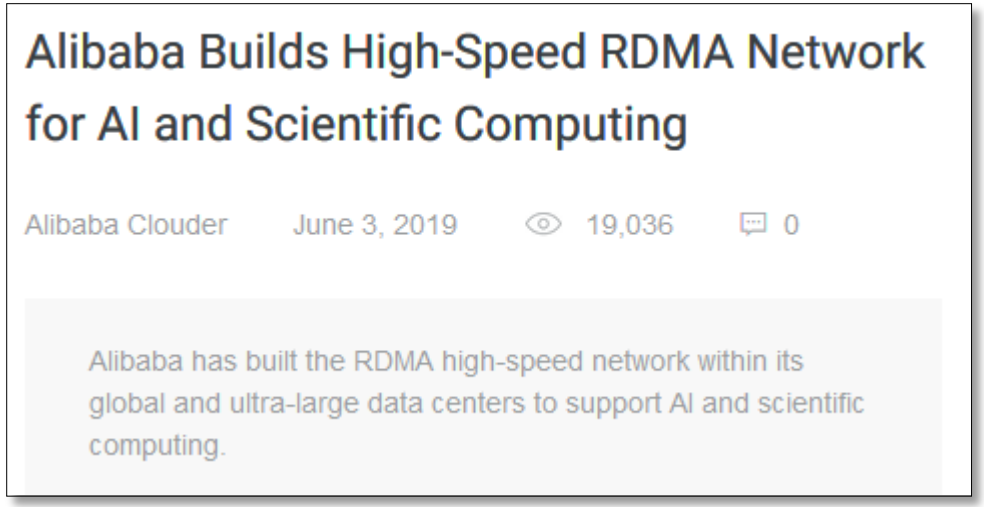
sRDMA: SECURE TRANSPORT FOR REMOTE DIRECT MEMORY ACCESS

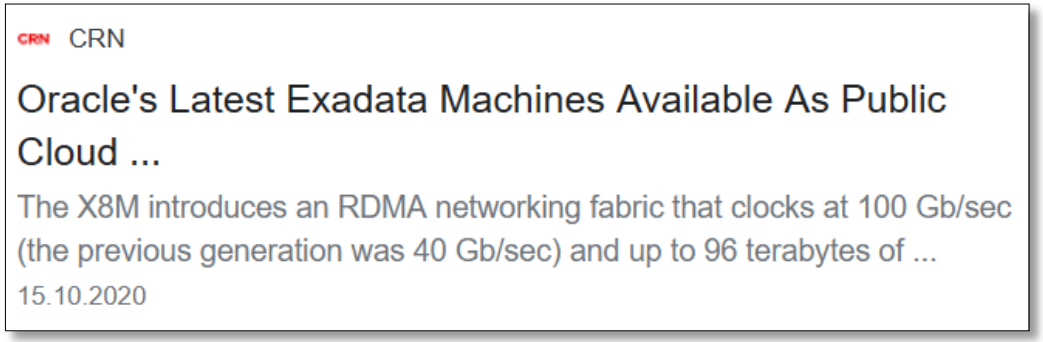
Konstantin Taranov

ETH Zurich

ETH zürich

RDMA IS A TREND IN PUBLIC CLOUDS





RDMA IS A TREND IN TOP SYSTEMS CONFERENCES

DSLR'18
DrTM+H'18
NAM-DB'17
FaRM'14
HERD'14
CoRM'21
Octopus'17
Wukong'16
RAMCloud'15
FileMR'14
FaSST'16
RDMP-KV'20
Catfish'19
XSTORE'20
ccNUMA'18
HydraDB'15
RDMA
Hermes'20
Derecho'19
Grappa'15
A1'20
DrTM+R'16
Crail'19
SparkRDMA'14
C-Hint'14
ScaleRPC'19
Dare'15
Storm'19
DaRPC'14
DrTM'15
TH-DPMS'16
Hyperloop'18
APUS'17

designed for performance – lower latency, higher bandwidth, lower CPU utilization etc.

IS INFINIBAND SECURE?

4 adversary models **10 vulnerabilities** **8 mitigations**

Implemented 6 attacks

	T1	T2	T3	T4	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	M1	M2	M3	M4	M5	M6	M7	M8
A1	◑	●	●	○	○	○	○	○	◑	◑	○	●	○	○	✢	✗	✗	✗	✗	✓	✗	✓
A2	◑	●	●	○	○	○	○	○	◑	◑	○	●	○	○	✢	✗	✗	✗	✗	✓	✗	✓
A3	●	◑	●	○	●	◑	◑	●	○	○	●	◑	●	◑	✗	✓	✗	✓	✓	✗	✗	✗
A4	●	○	○	○	○	○	○	○	●	○	○	○	○	○	✗	✗	✗	✗	✗	✗	✢	✗
A5	●	◑	●	○	○	○	○	○	○	○	●	○	○	○	✗	✗	✢	✗	✗	✗	✢	✗
A6	○	○	○	●	○	○	○	○	○	○	●	○	○	◑	✗	✗	✢	✗	✗	✗	✗	✗

V1 weak rkey
V2 static init.
V3 shared key gen.
V4 weak mem. rand.
V5 lin. inc. QPN
V6 fixed starting PSN
V7 lim. attack detect.
V8 no enc./auth.
V9 single PD
V10 ODP enabled
M1 rand. QPN
M2 rand. rkey
M3 HW counters
M4 mem. win. type 2
M5 multiple PDs
M6 enc./auth. in IB
M7 resource const.
M8 in-network filt.

● enables attack ◑ facilitates attack ○ does not affect attack
✓ mitigates attack ✢ increases attack complexity ✗ does not mitigate attack

Rothenberger et al.: ReDMArk: Bypassing RDMA Security Mechanisms, Usenix Security'21

FEATURES OF SECURE TRANSPORT

- **Integrity**
  - verifies that the data has not been forged or tampered with
- **Authentication**
  - ensures that the parties exchanging information are who they claim to be
- **Encryption**
  - hides the data being transferred from third parties.
- **Replay attack protection**
  - Each unique packet is processed only one time.

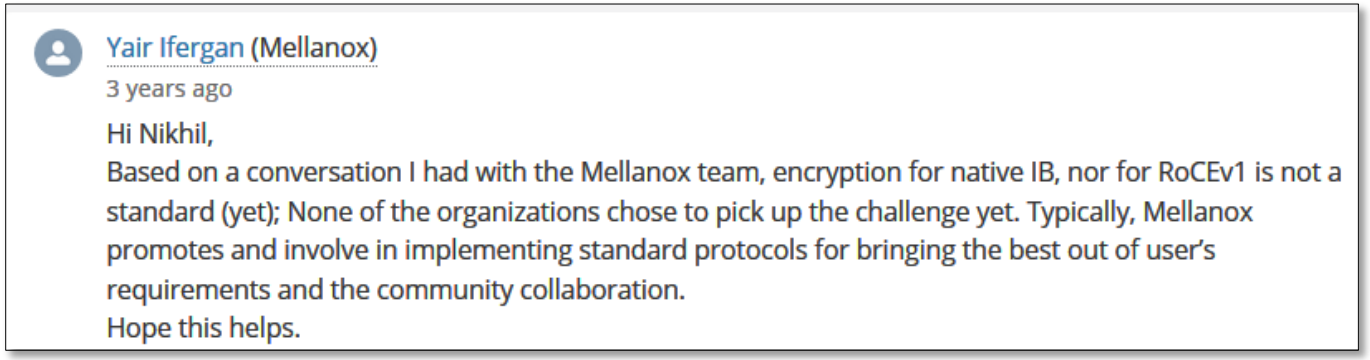
INFINIBAND SECURITY CONSIDERATIONS

RFC4297 – Remote Direct Memory Access (RDMA) over IP Problem Statement: December 2005

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

Manhee Lee et al., Security Enhancement in InfiniBand Architecture (IPDPS'05): April 2005

"Replacing 32 bit checksum with a message authentication code"



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.

March 2017

IPsec over RoCE August 2020

"Offered by Mellanox ConnectX-6 DX network adapter"

Arjun Singhvi et al., 1RMA: Re-Envisioning Remote Memory Access (SIGCOMM'20): August 2020

"A new RDMA protocol with a secure transport"

CAN APPLICATION-LEVEL SECURITY BE USED?

- **One-sided RDMA requests are completely performed by the NIC**
  - No CPU involvement on the destination machine
  - The data must be encrypted/decrypted by the host CPU
  - No protection against replay attacks

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

InfiniBand is in urgent need of a secure transport

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 backward-compatible with InfiniBand Architecture (IBA), and can be easily adapted by**
  - Native InfiniBand
  - RoCEv1
  - RoCEv2

Taranov et al. sRDMA -- Efficient NIC-based Authentication and Encryption for Remote Direct Memory Access, Usenix ATC'20

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

```
//Recognized attribute masks by ibv_modify_qp
enum ibv_qp_attr_mask {
  ... // previous masks
  IBV_QP_SECURE = 1 << 26,
}
```

```
//Recognized attributes by ibv_modify_qp
struct ibv_qp_attr {
  ... // previous fields
  struct ibv_secure_attr secure_attr;
}
```

```
// 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
};
```

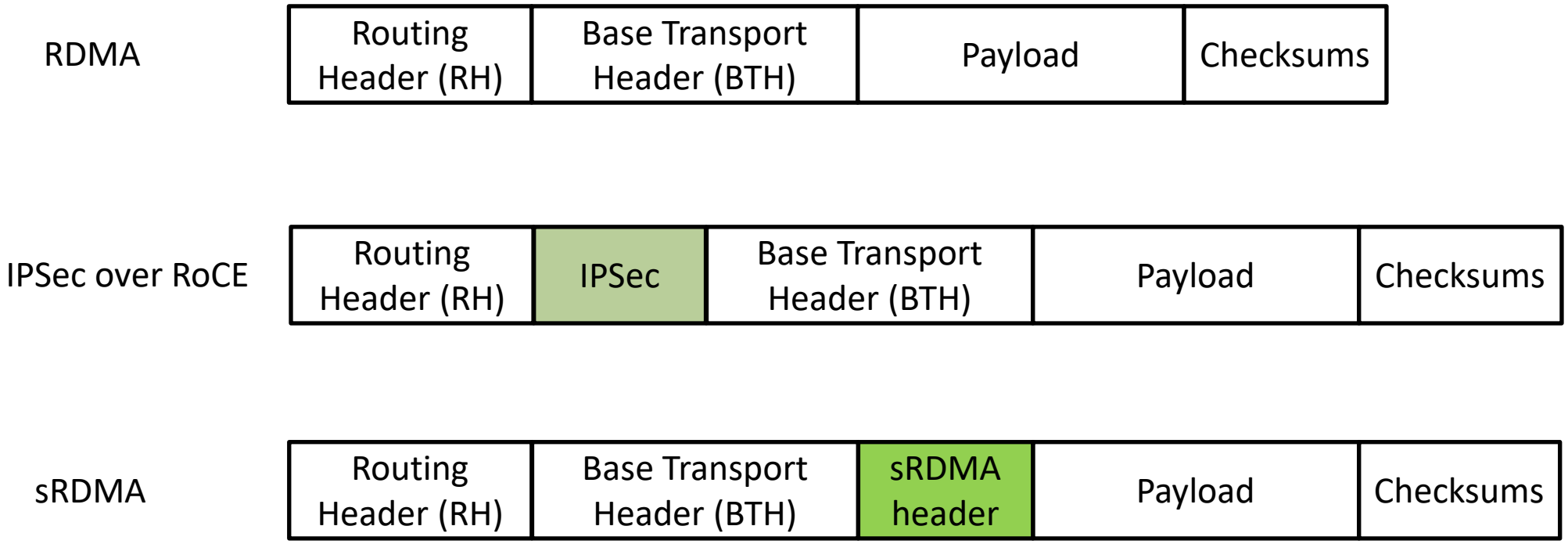
  - Supported security codes:

```
/*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
```

PACKET FORMAT



sRDMA is an extension to InfiniBand architecture:

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

sRDMA VS IPSEC OVER ROCE

- Larger headers in IPSec compared to sRDMA
- IPSec is only for RoCE, native InfiniBand is not supported
- IPSec over RoCE is unnatural: IPSec is not aware of QP connections
- IPSec cannot support additional features such as PD sharing or memory protection
- IPSec policies are installed per remote IP, sRDMA policies are installed per QP connection.
- IPSec policies are managed by OS, sRDMA policies are managed by an application.

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

bytes \ bits	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 backward-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)

bytes \ bits	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 packets 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 – SECURE PROTECTION DOMAIN

- **sRDMA introduces secure protection domains (sPDs)**

```
//allocate secure pd.
//All resources created on this pd will be secure.
struct ibv_pd *
ibv_alloc_secure_pd(struct ibv_context *context,
                    ibv_secure_pd_attr* attr);
```

```
struct ibv_secure_pd_attr {
  uint32_t type; // IBV_SECURE_PD | IBV_SECURE_MEM
  uint8_t pd_key[MAX_KEY_LENGTH];  // pd protection key
  uint8_t mem_key[MAX_KEY_LENGTH]; // memory protection key
  uint8_t max_memtree_depth; // maximum depth of OWF tree
};
```

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

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

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

EXTENDED MEMORY PROTECTION

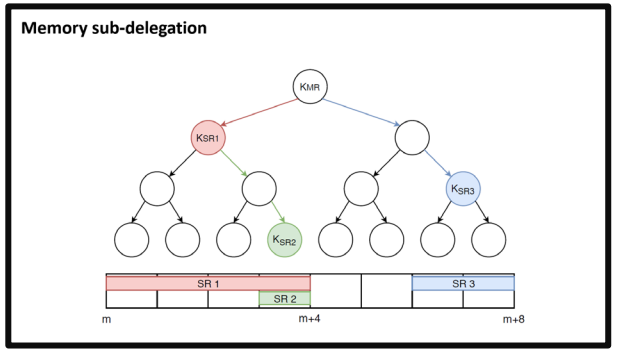
- **Memory protection in IBA is based on *rkey tags (32 bits)***
  - *Each one-sided RDMA request must include rkey in its request*
  - *Any endpoint with the rkey can access the memory*
  - *For fine-grained access control, Memory windows type 2 can be pinned to a single QP*

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

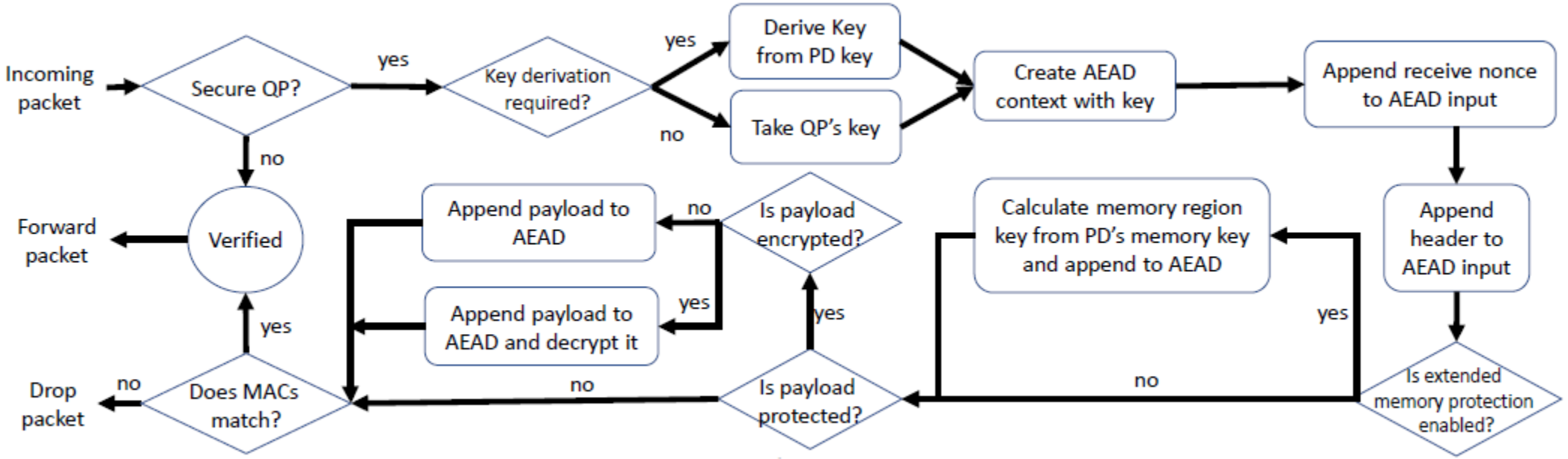
$$K_{SR} = PRF_{K_{MR}}(START_{SR} \,||\, END_{SR})$$

  - sRDMA does not introduce extra header and reuses the STH

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

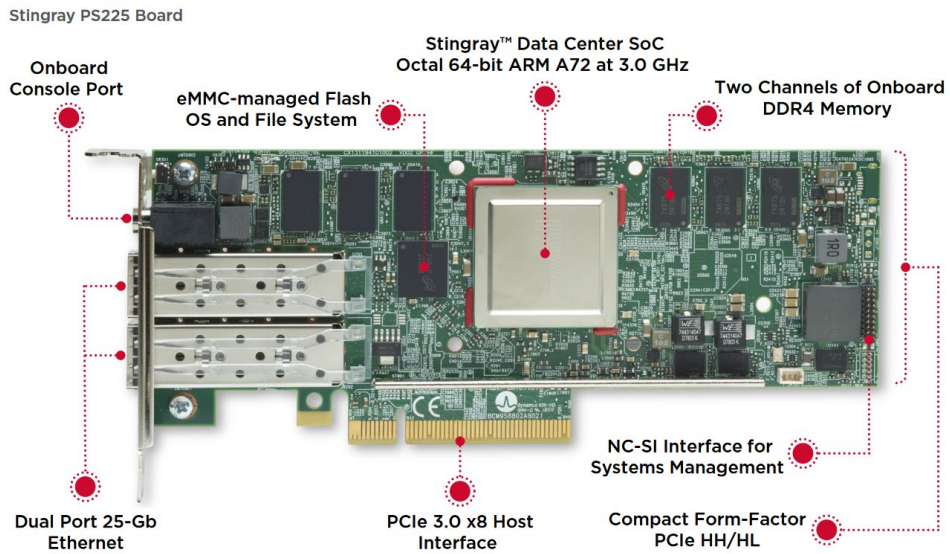


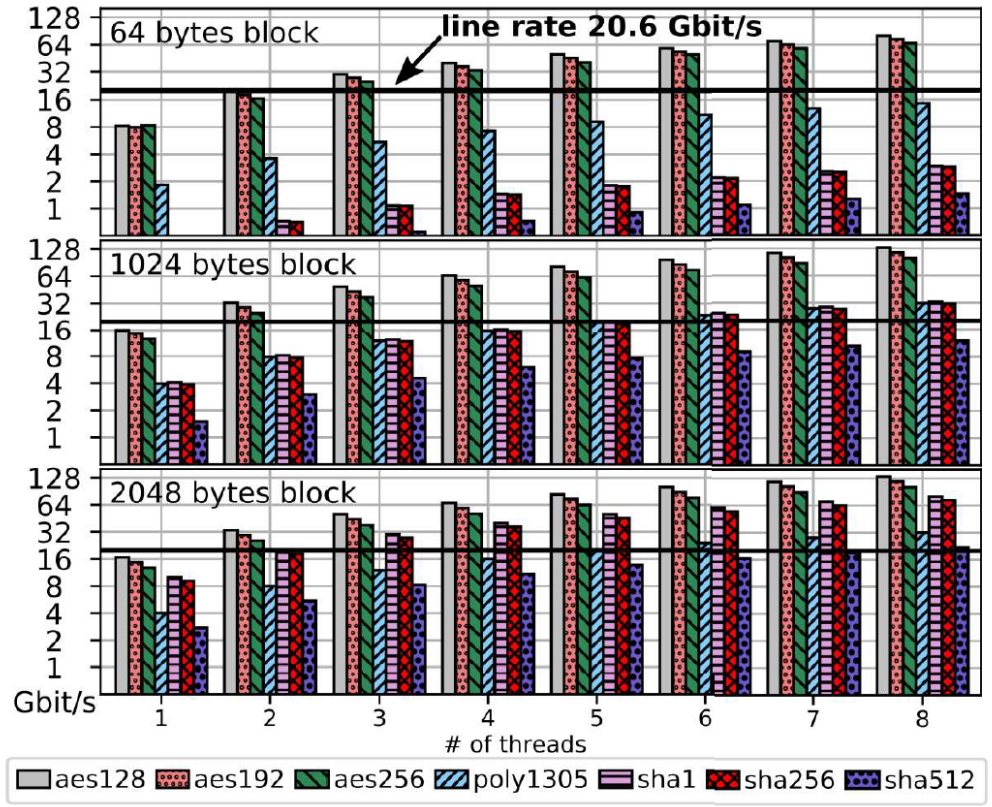
PROCESSING OF INCOMING PACKETS



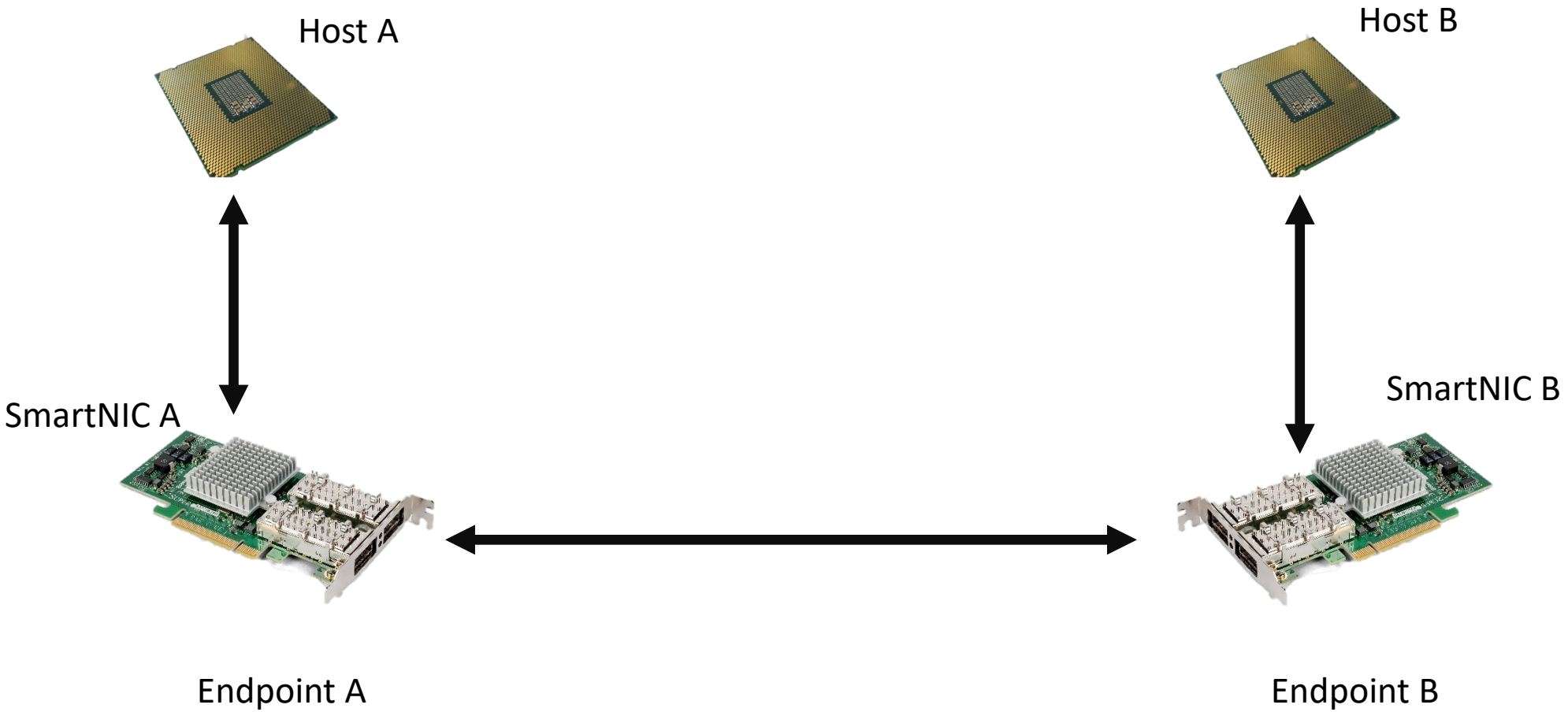
IMPLEMENTATION OF sRDMA

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

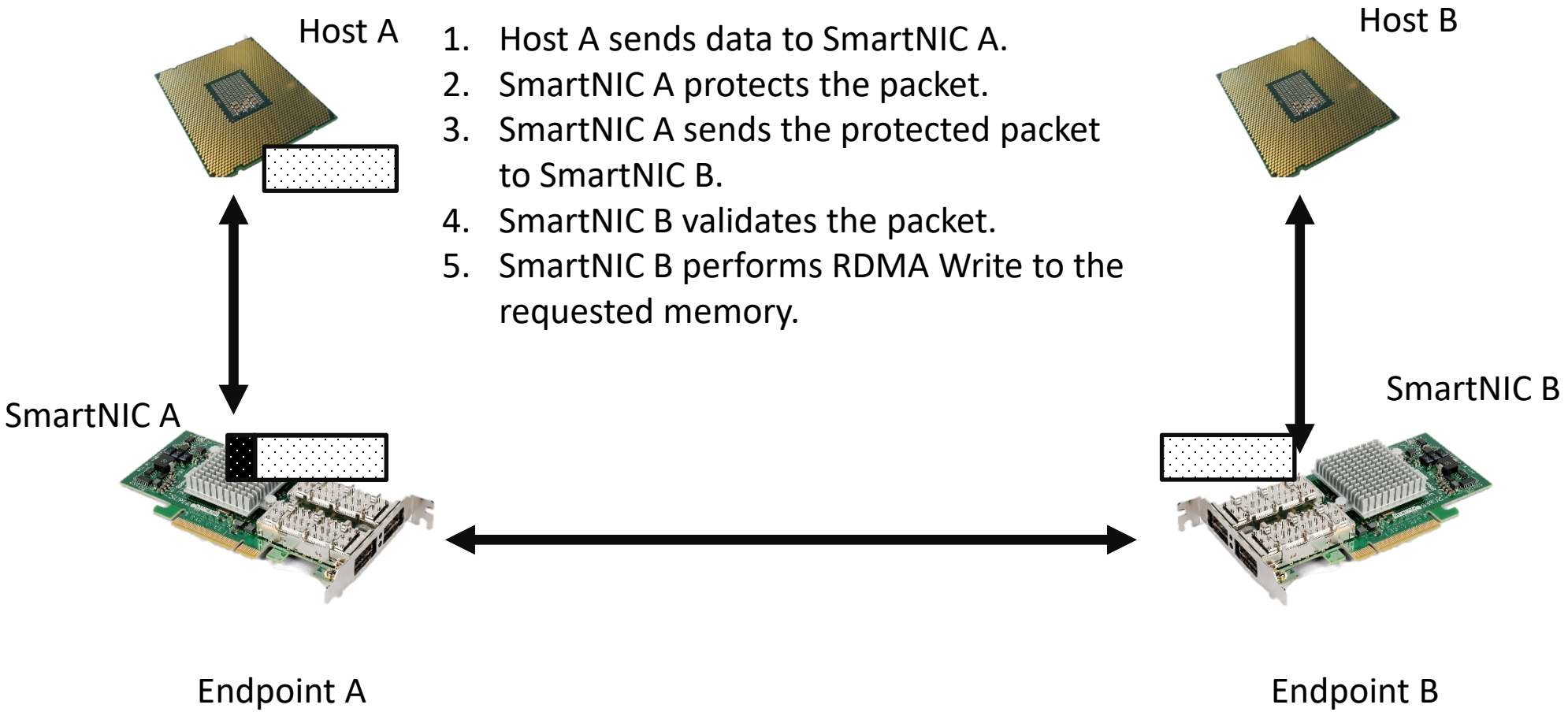




IMPLEMENTATION OF sRDMA

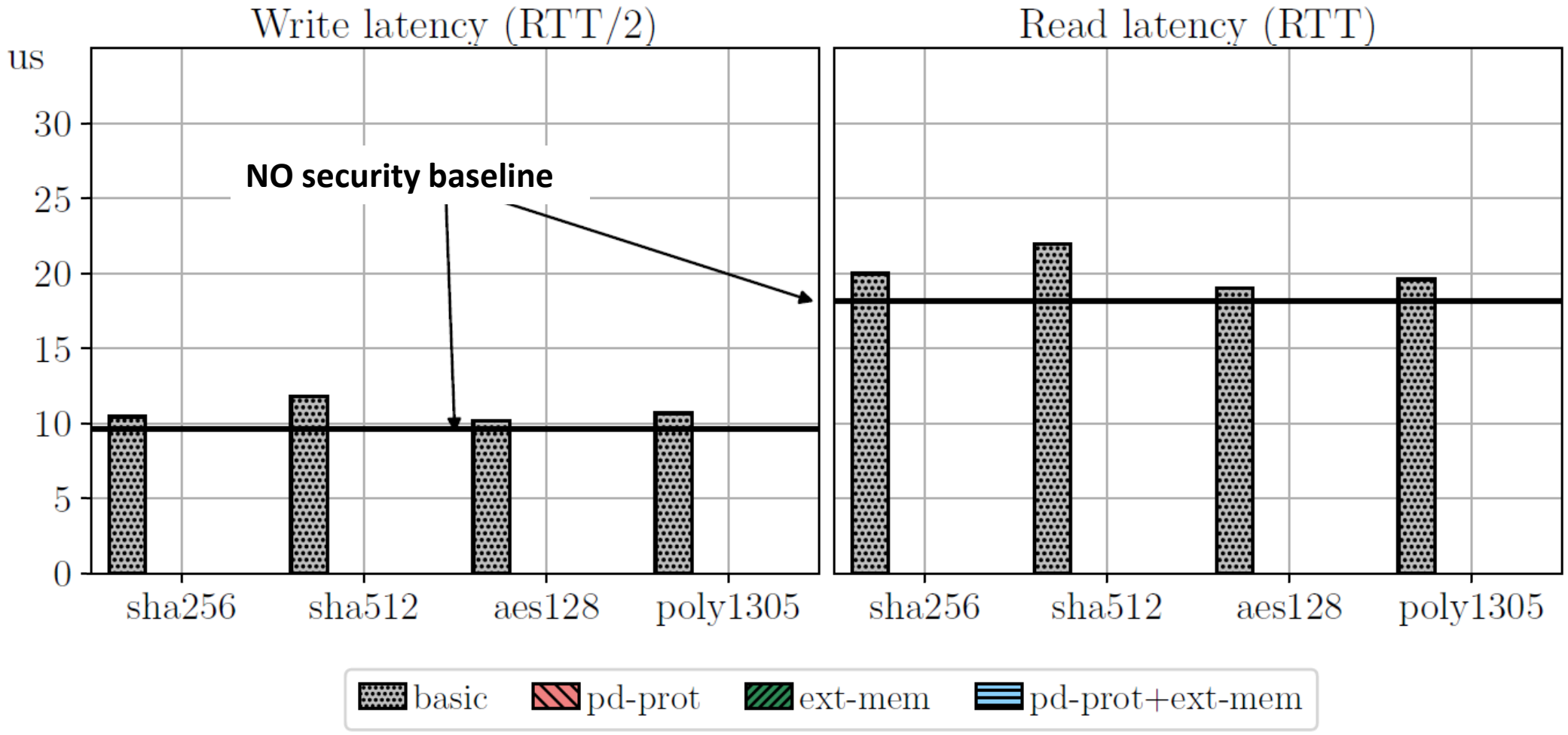


IMPLEMENTATION OF sRDMA

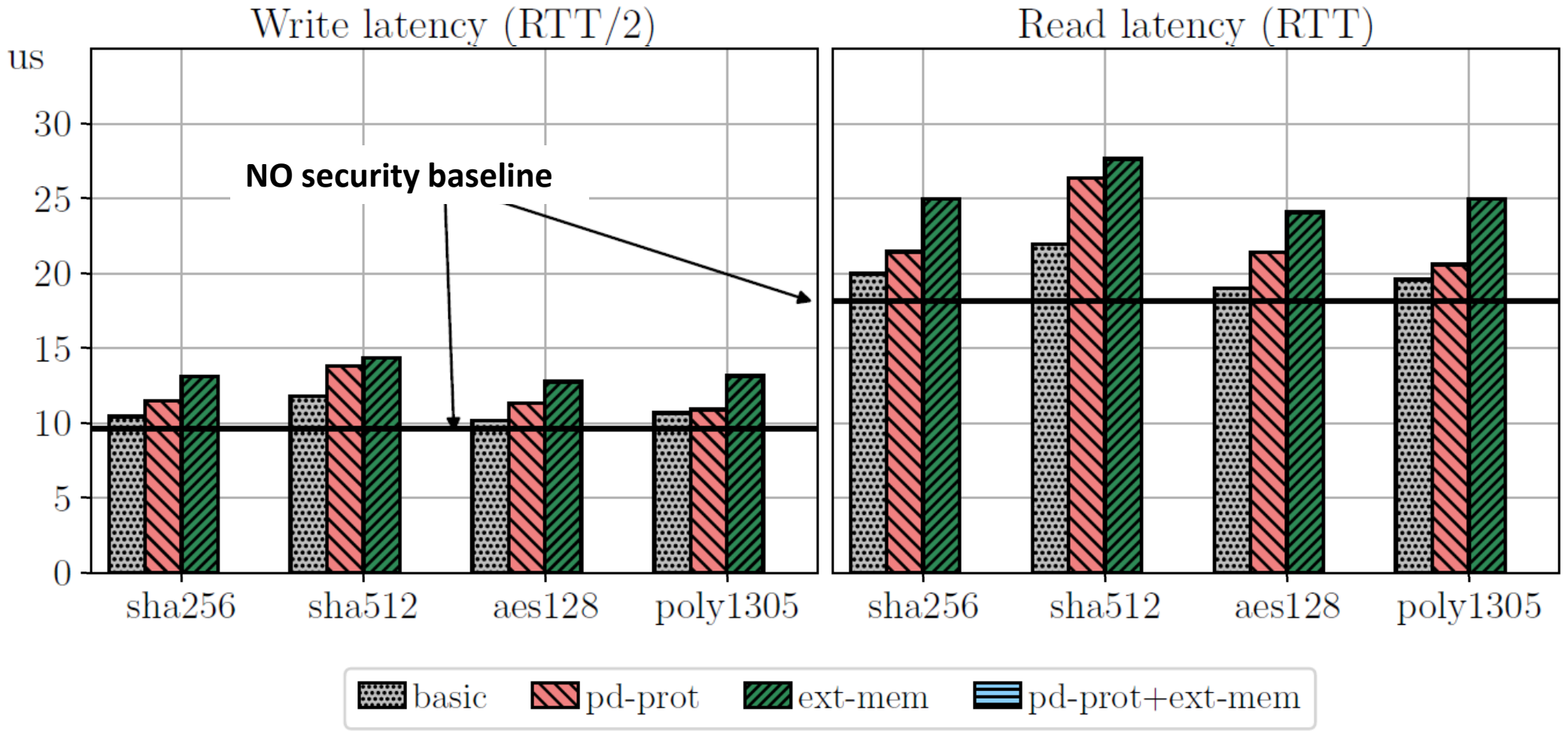


1. Host A sends data to SmartNIC A.
2. SmartNIC A protects the packet.
3. SmartNIC A sends the protected packet to SmartNIC B.
4. SmartNIC B validates the packet.
5. SmartNIC B performs RDMA Write to the requested memory.

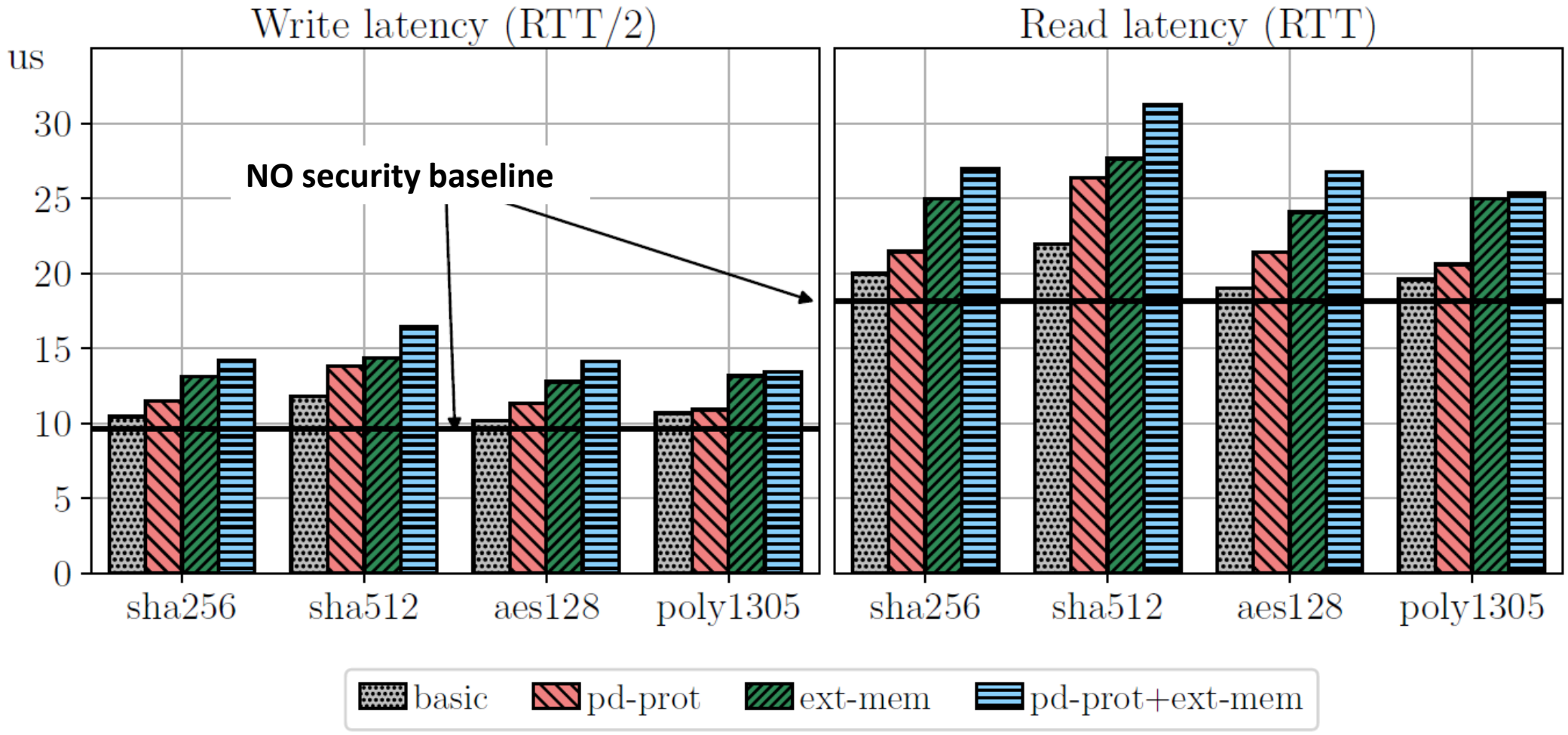
SOURCE AUTHENTICATION LATENCY



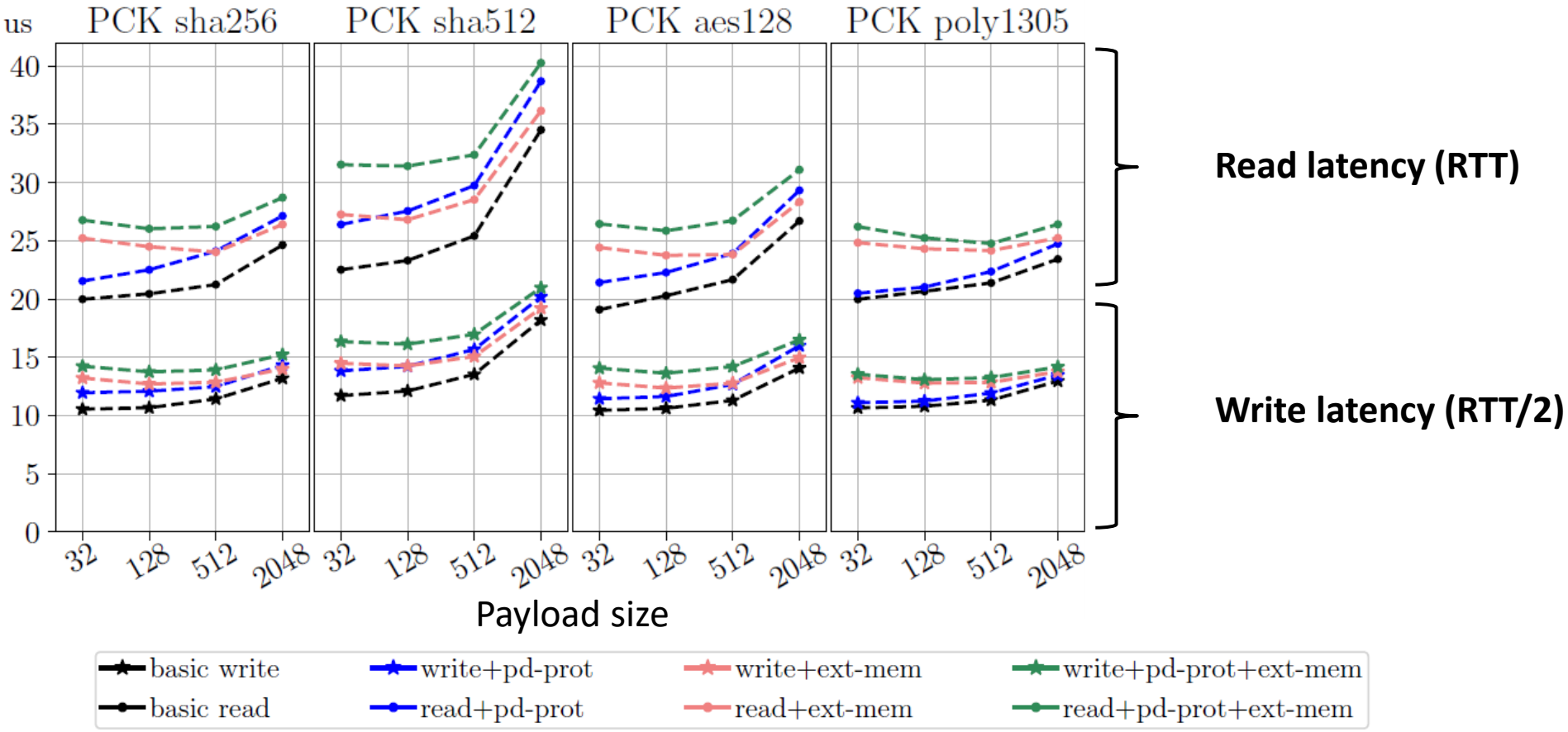
SOURCE AUTHENTICATION LATENCY



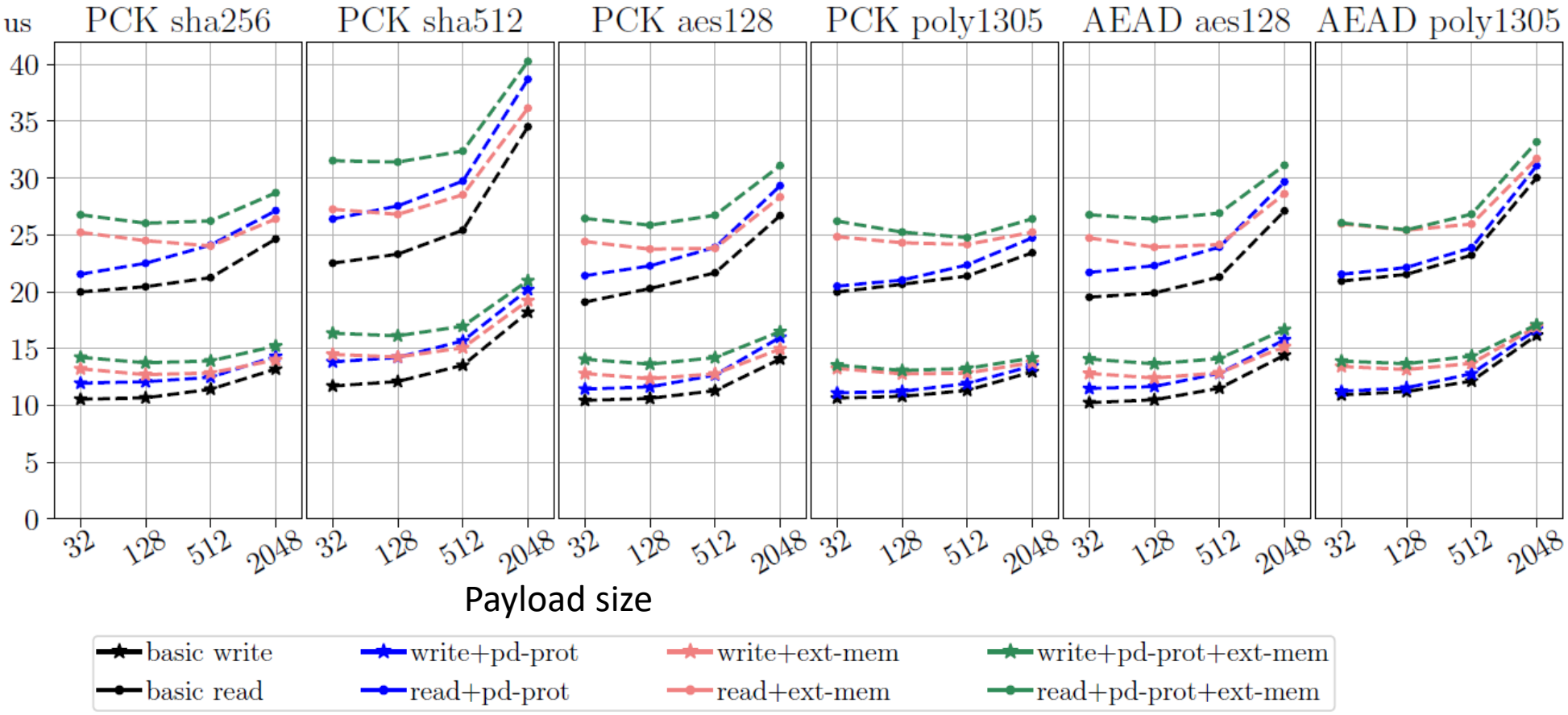
SOURCE AUTHENTICATION LATENCY



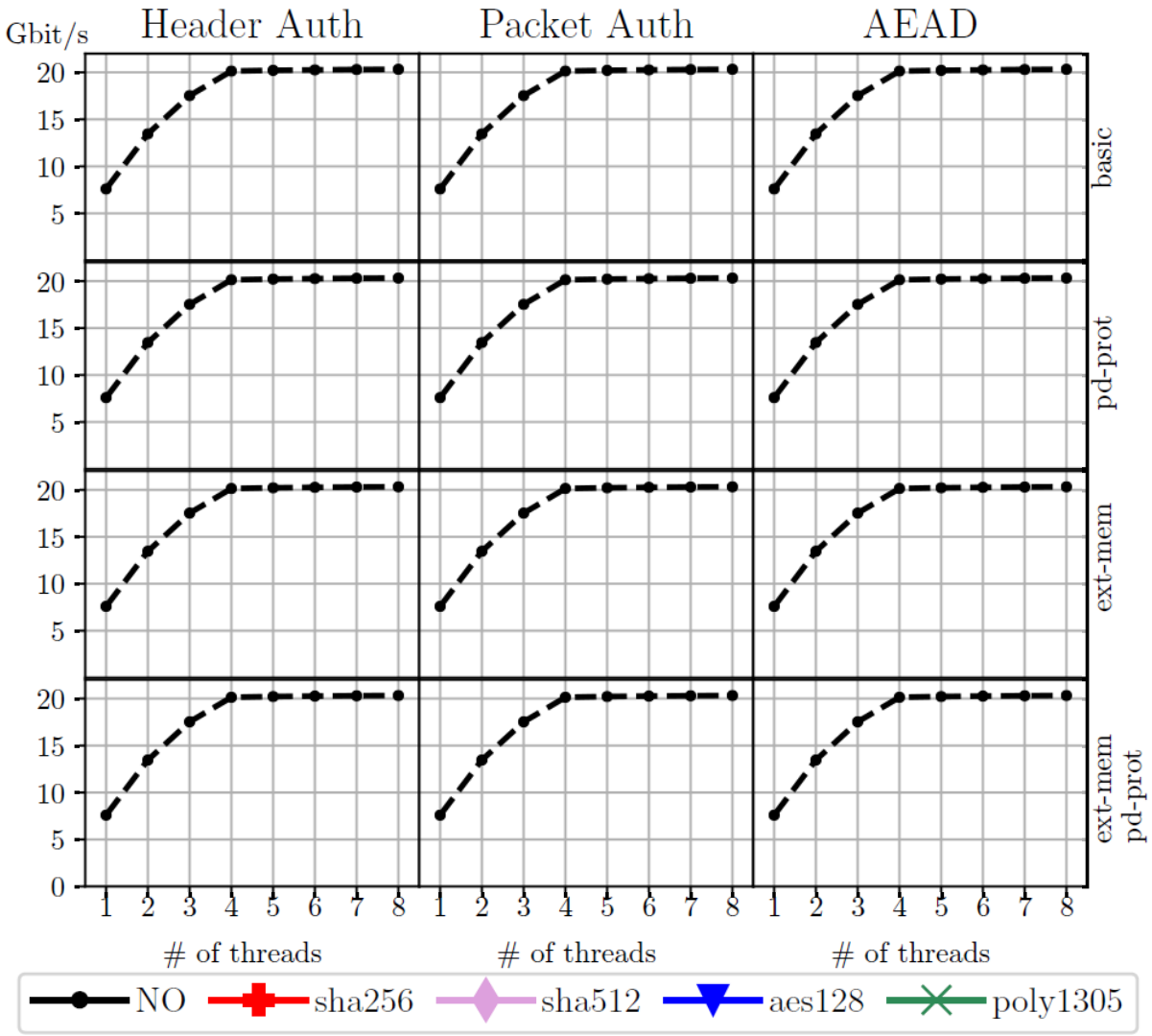
PACKET AUTHENTICATION LATENCY



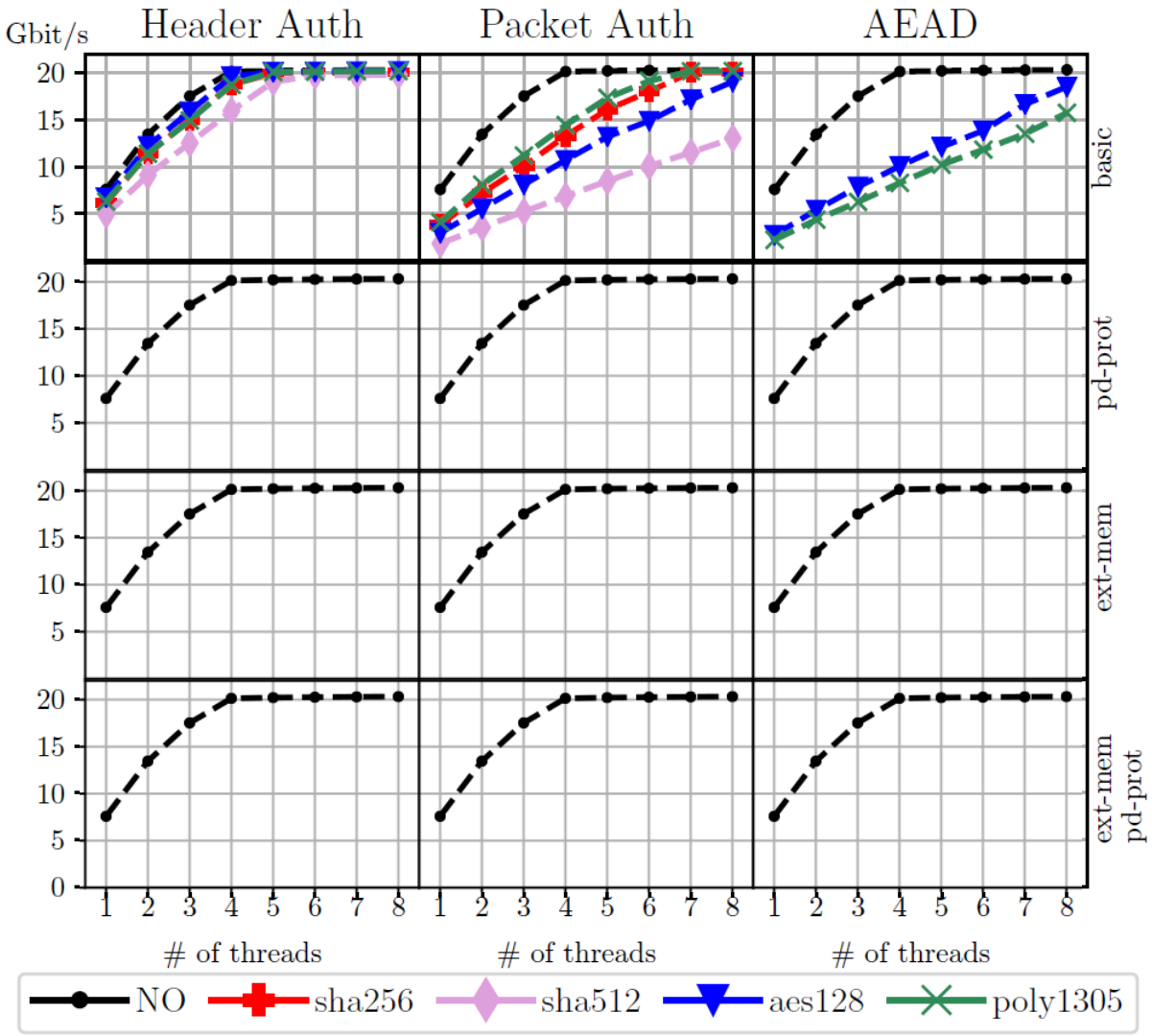
AEAD LATENCY



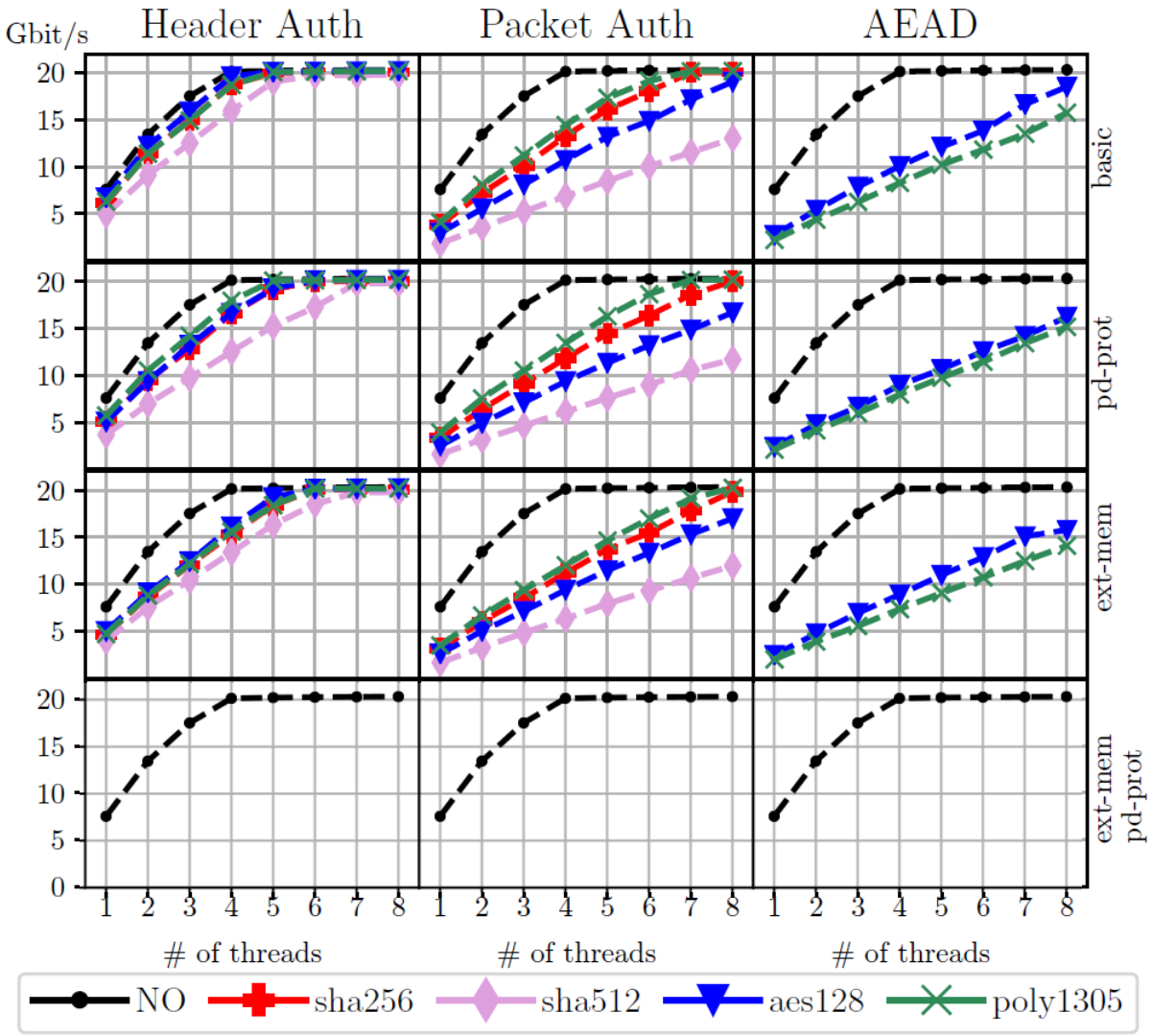
WRITE BANDWIDTH



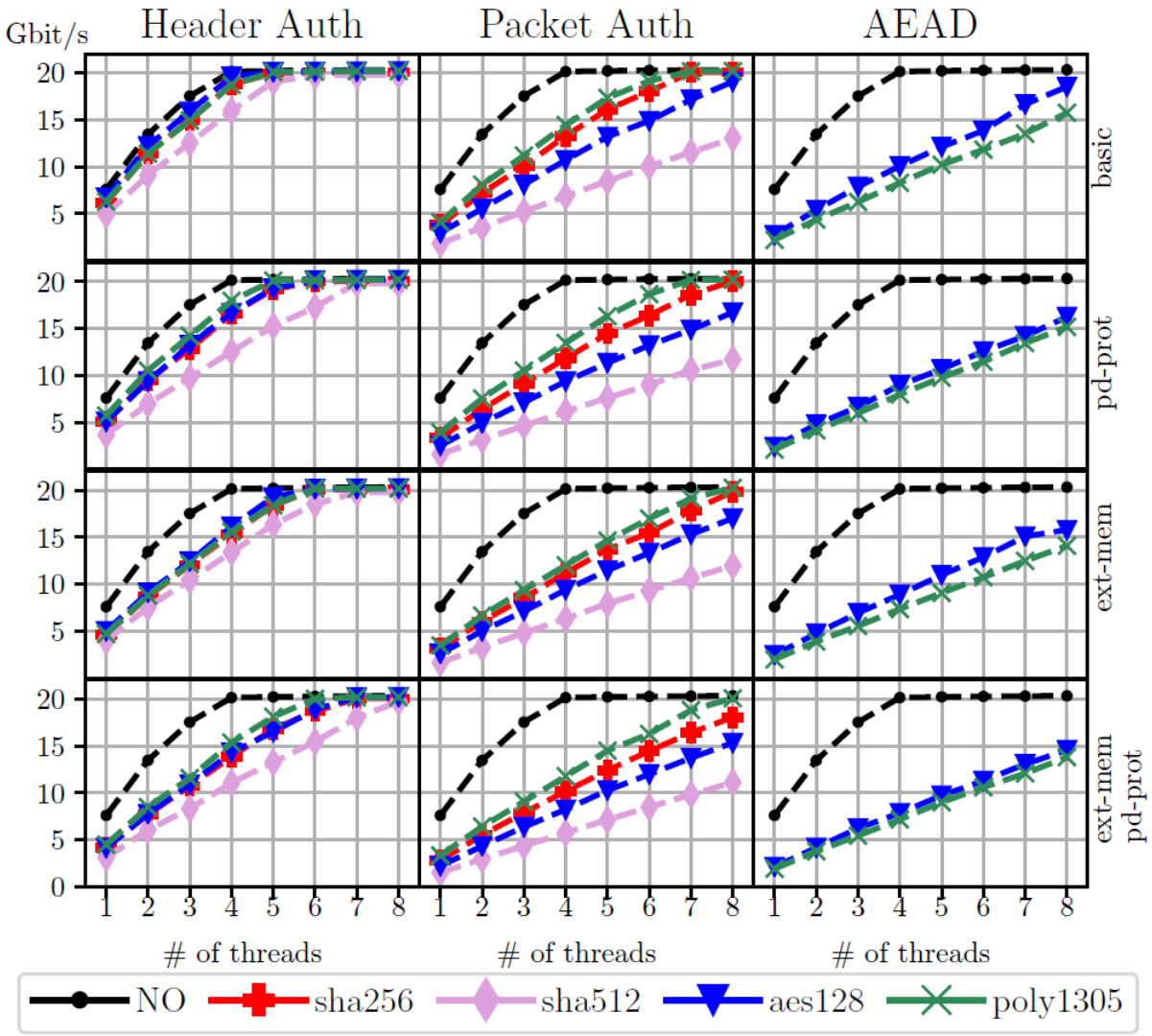
WRITE BANDWIDTH



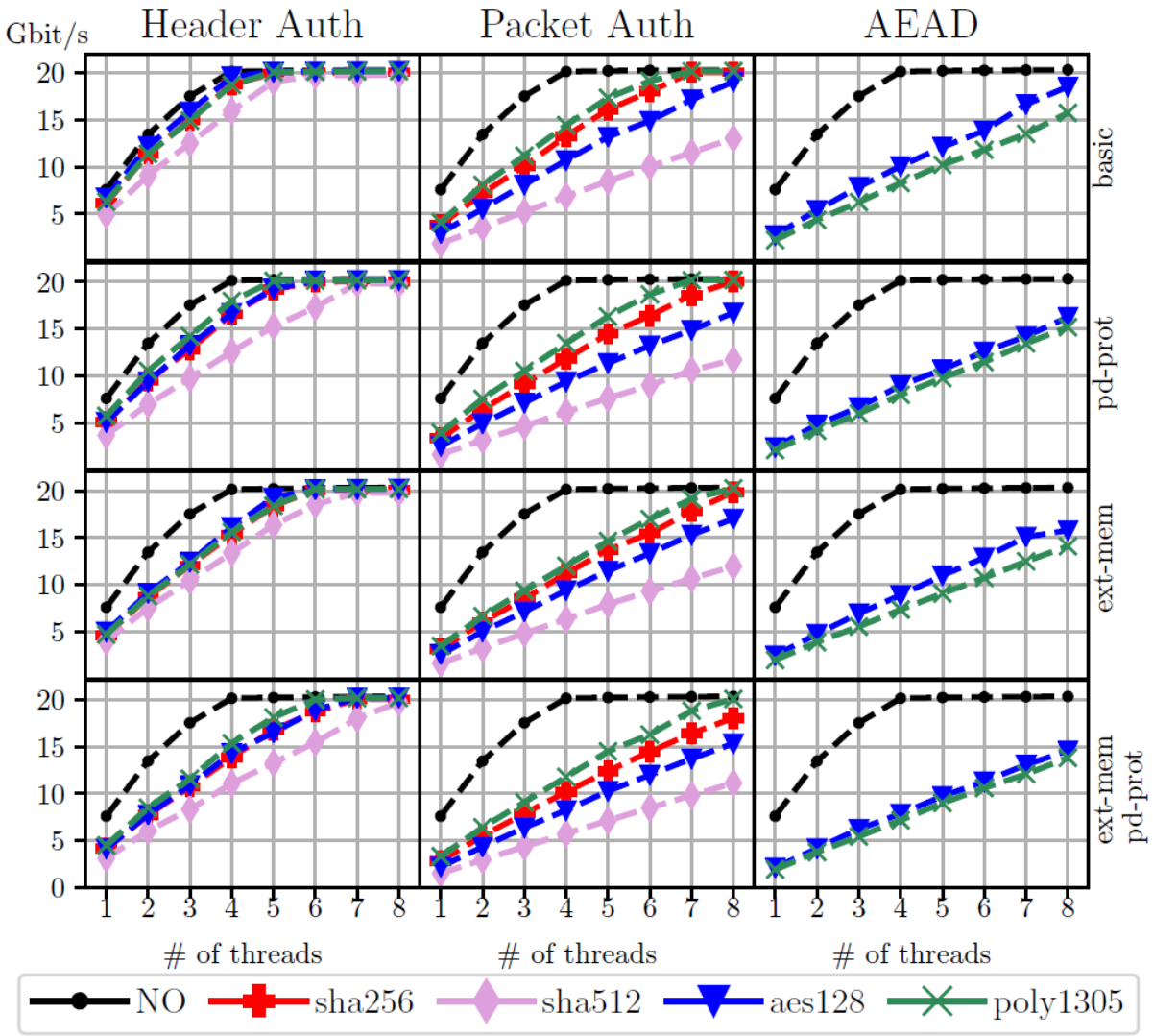
WRITE BANDWIDTH



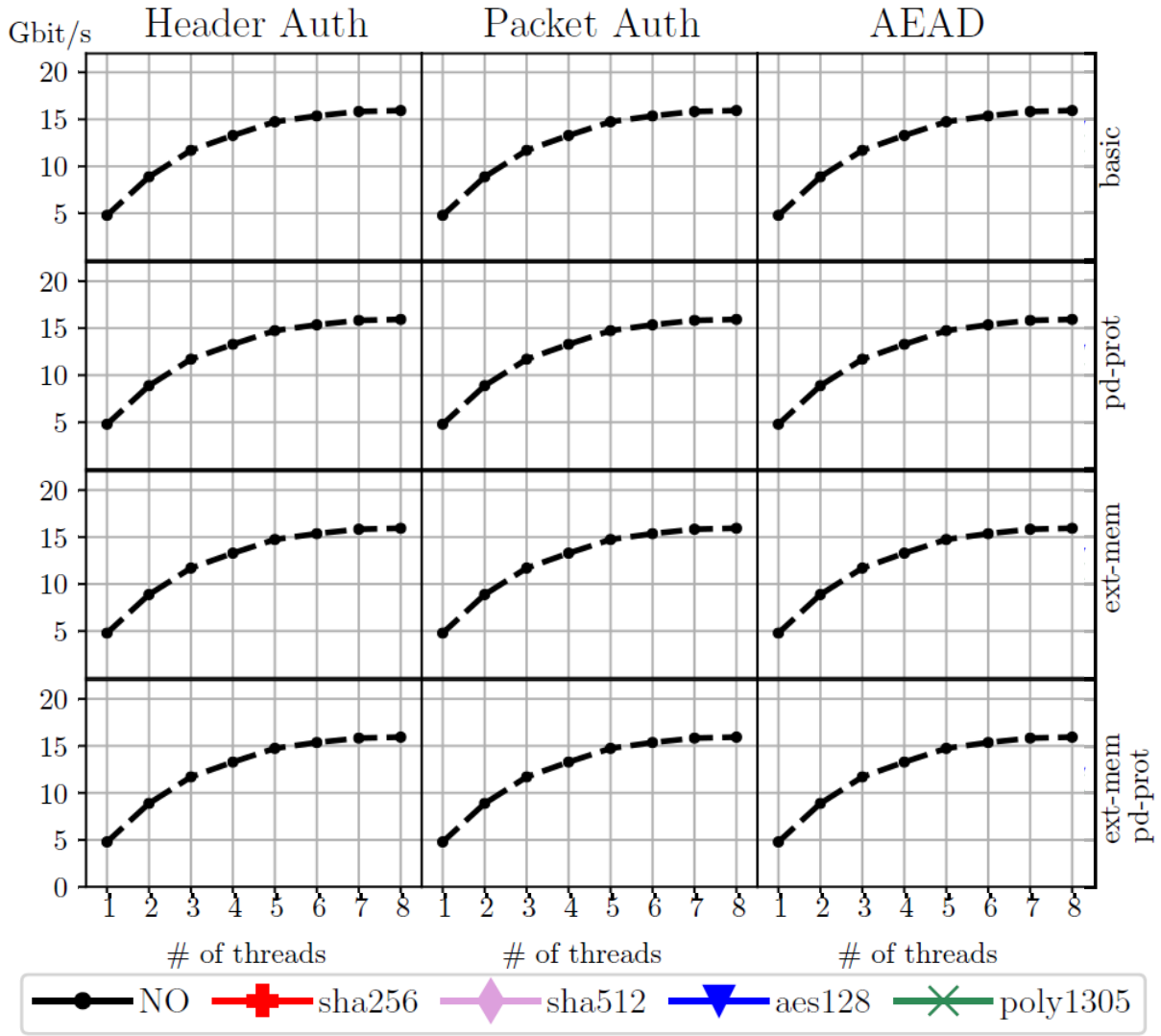
WRITE BANDWIDTH



READ BANDWIDTH

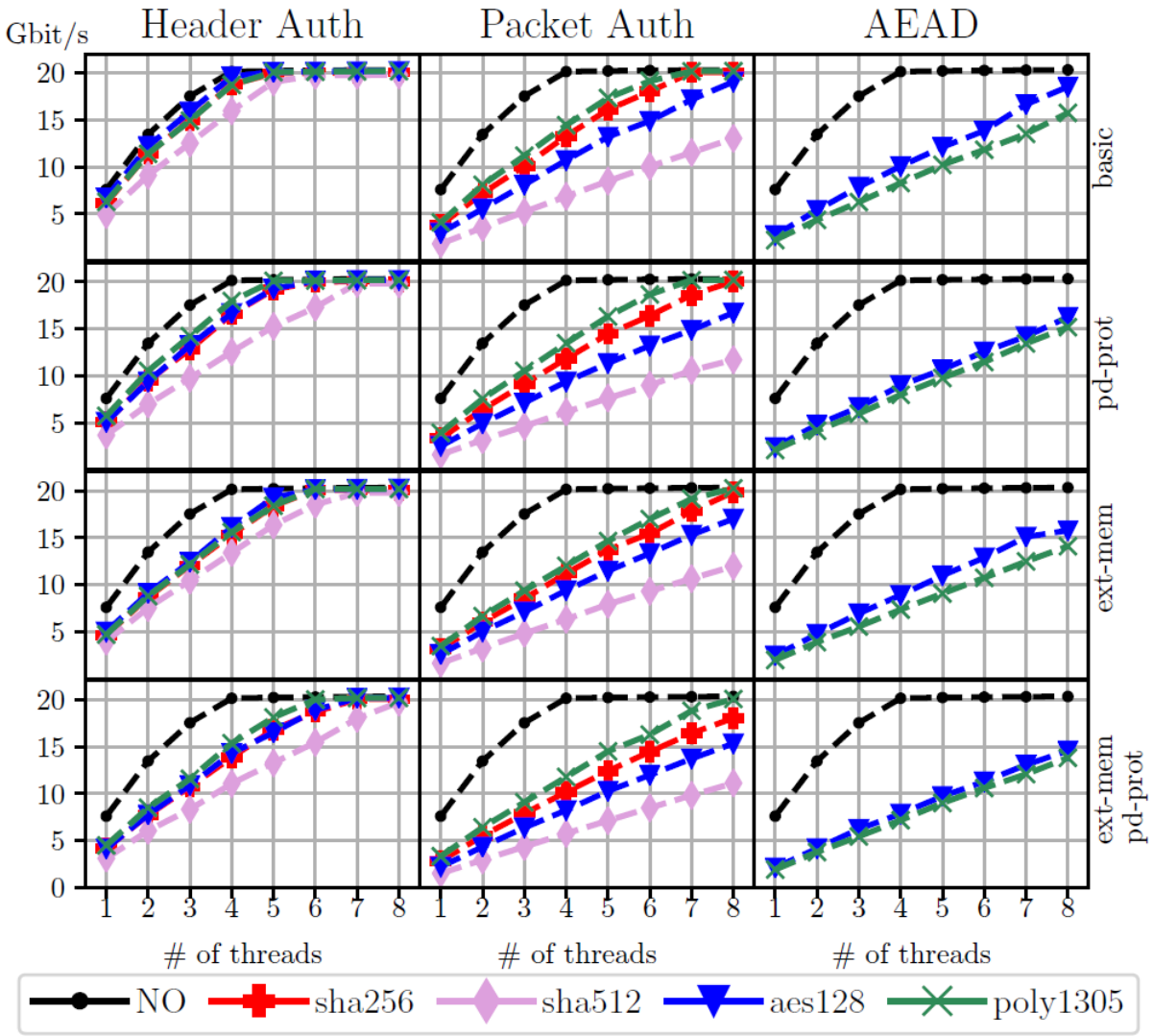


Write Bandwidth

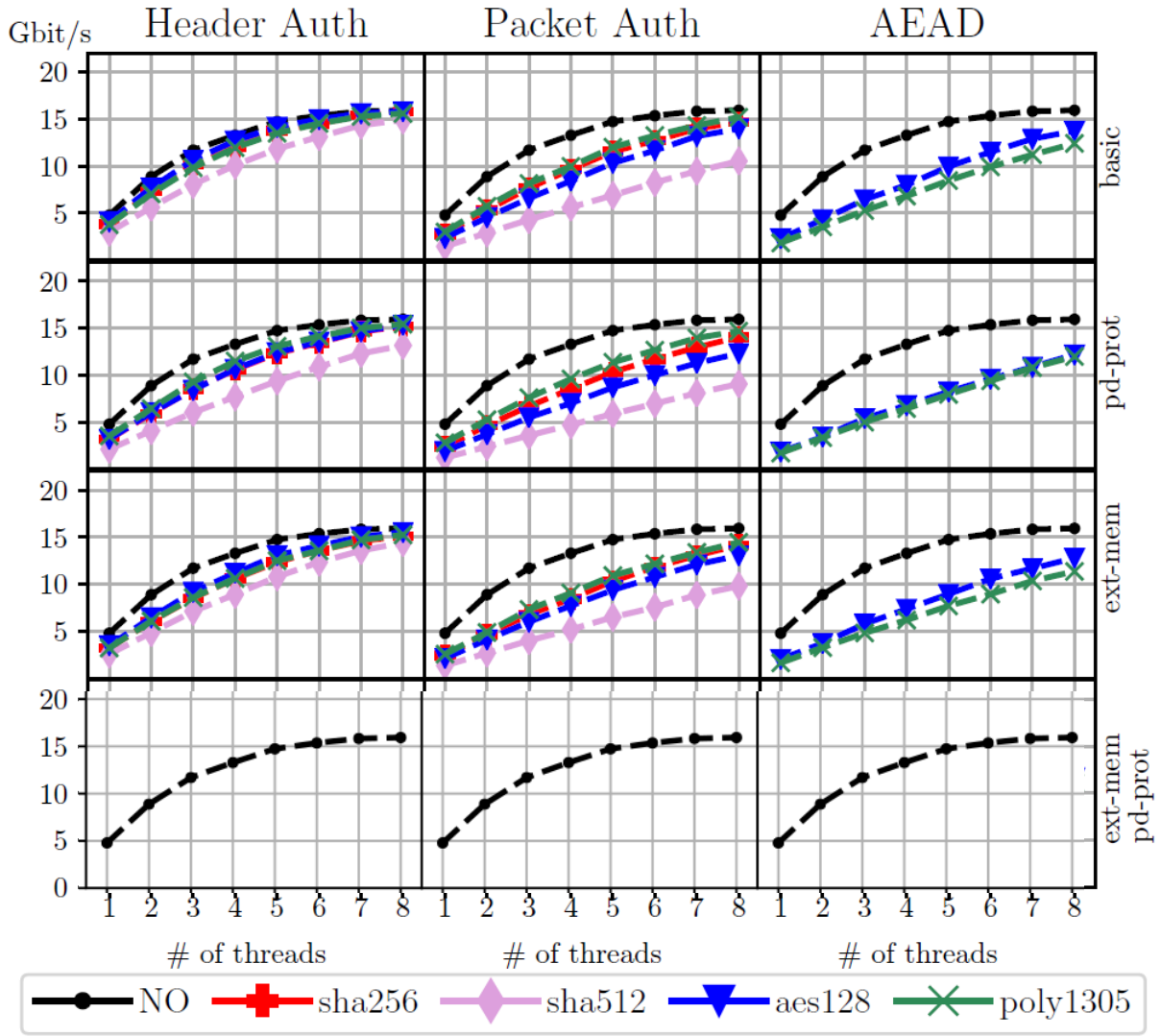


Read Bandwidth

READ BANDWIDTH

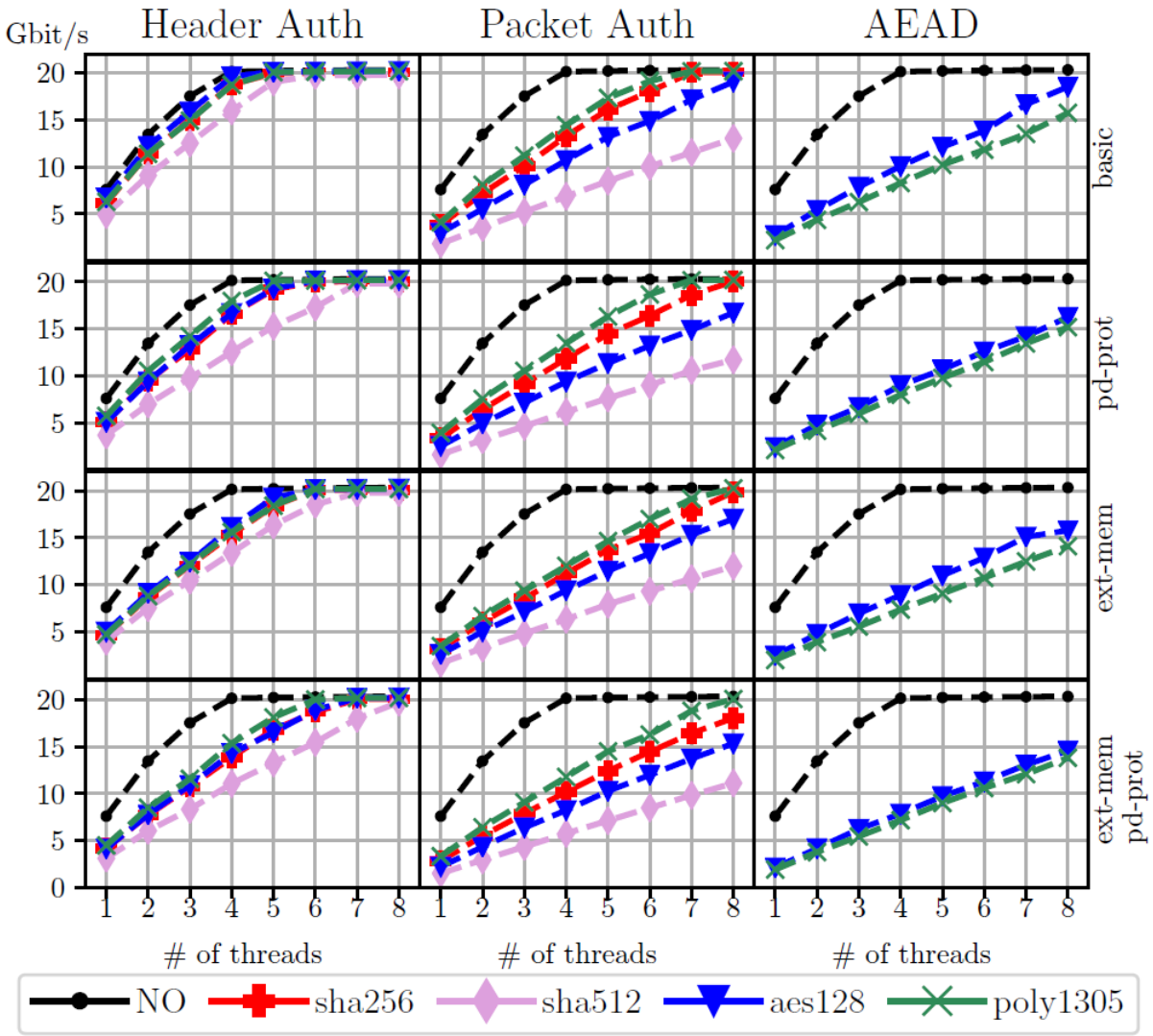


Write Bandwidth

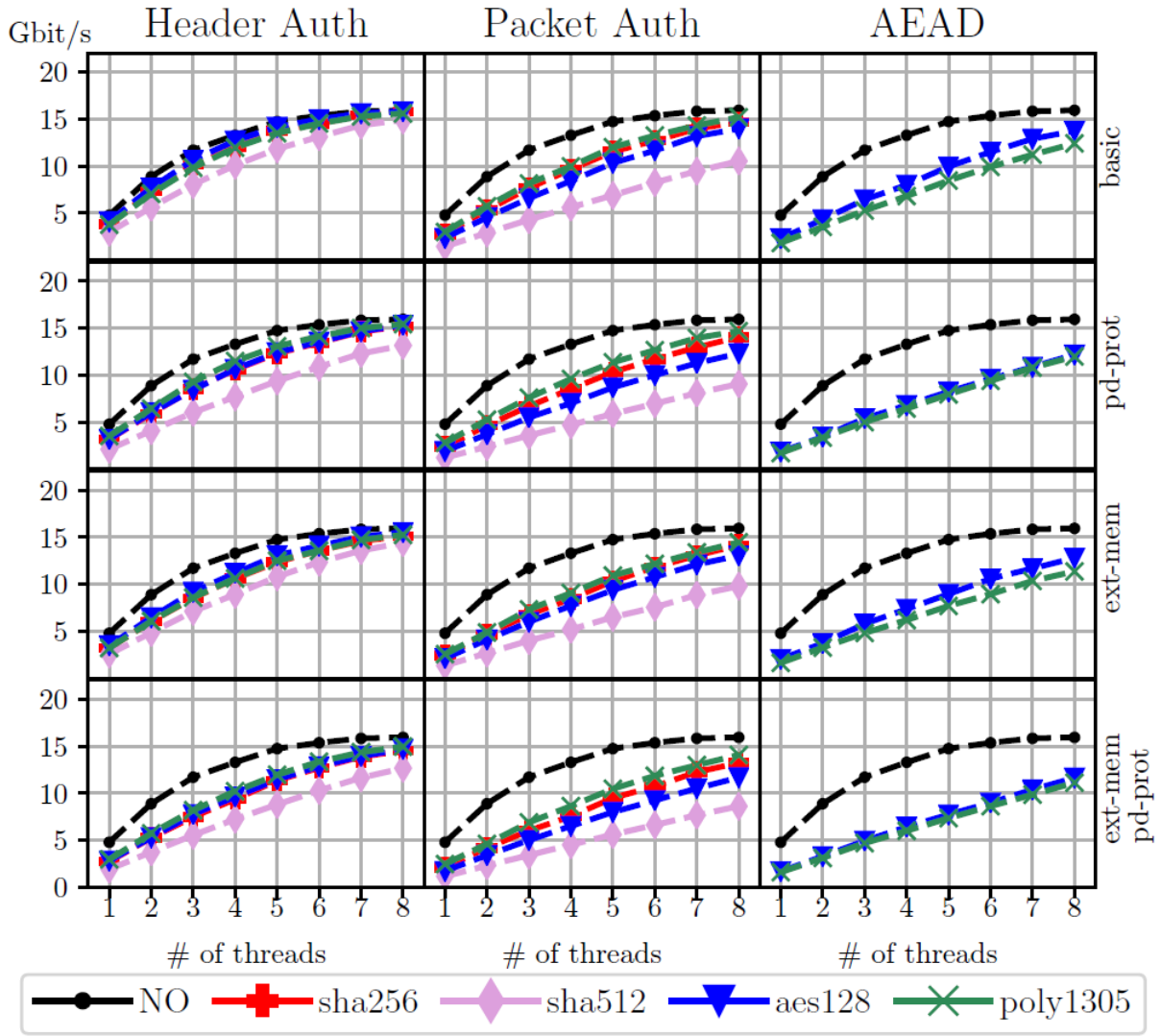


Read Bandwidth

READ BANDWIDTH



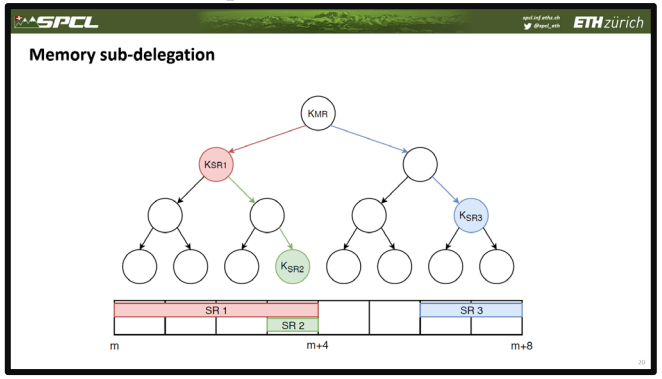
Write Bandwidth

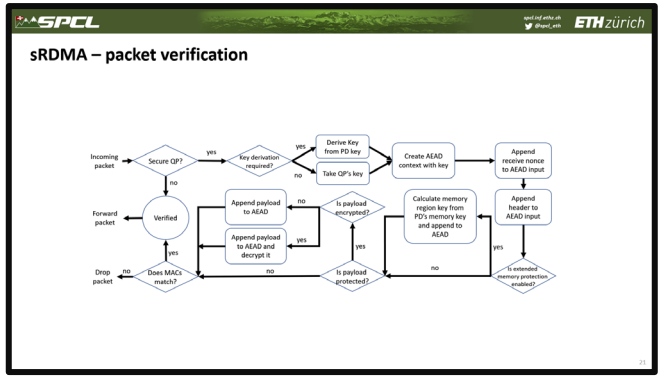


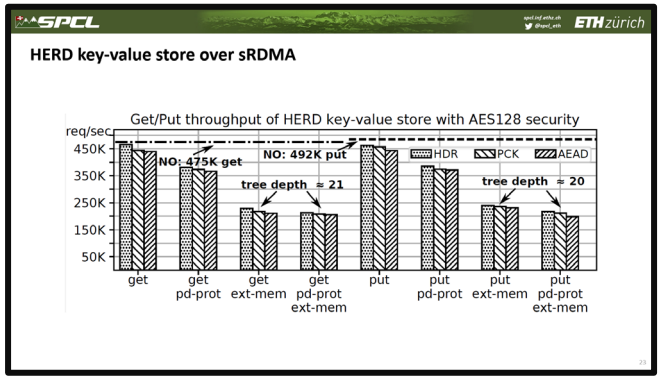
Read Bandwidth

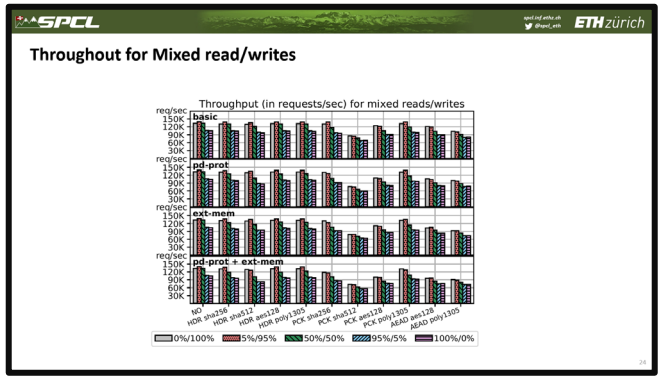
sRDMA PAPER ALSO INCLUDES

- **Memory sub-delegation (clients can pass access like capabilities)**
- **Details on the implementation**
- **Additional experiments**









Taranov et al. sRDMA -- Efficient NIC-based Authentication and Encryption for Remote Direct Memory Access, Usenix ATC'20

2021 OFA Virtual Workshop

THANK YOU

Konstantin Taranov

ETH Zurich

ETH zürich