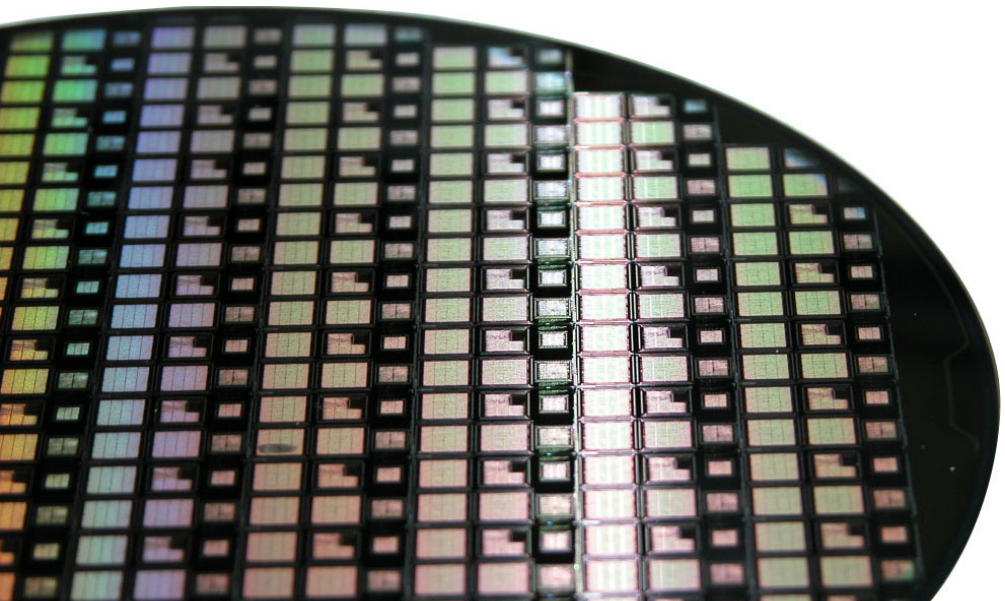
Towards Evaluating High-Speed ASIC Implementations of CAESAR Candidates for Data at Rest and Data in Motion

Work in Progress

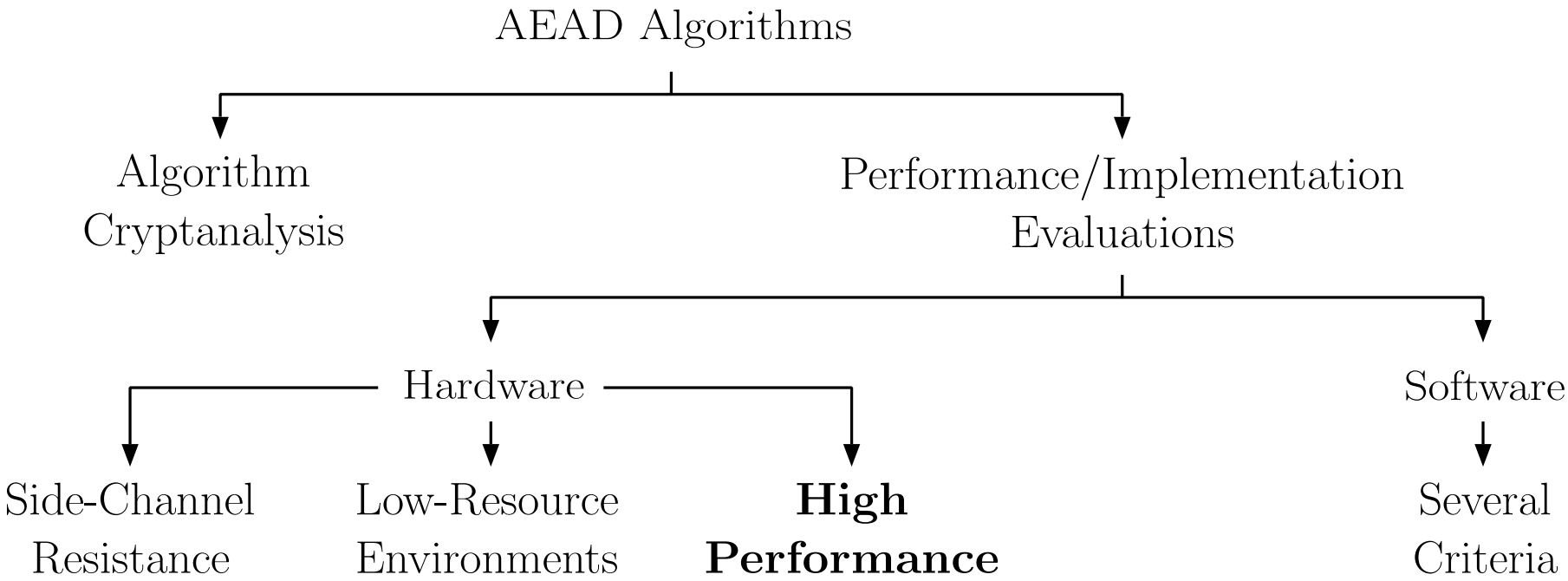
Singapore, 28. September 2015



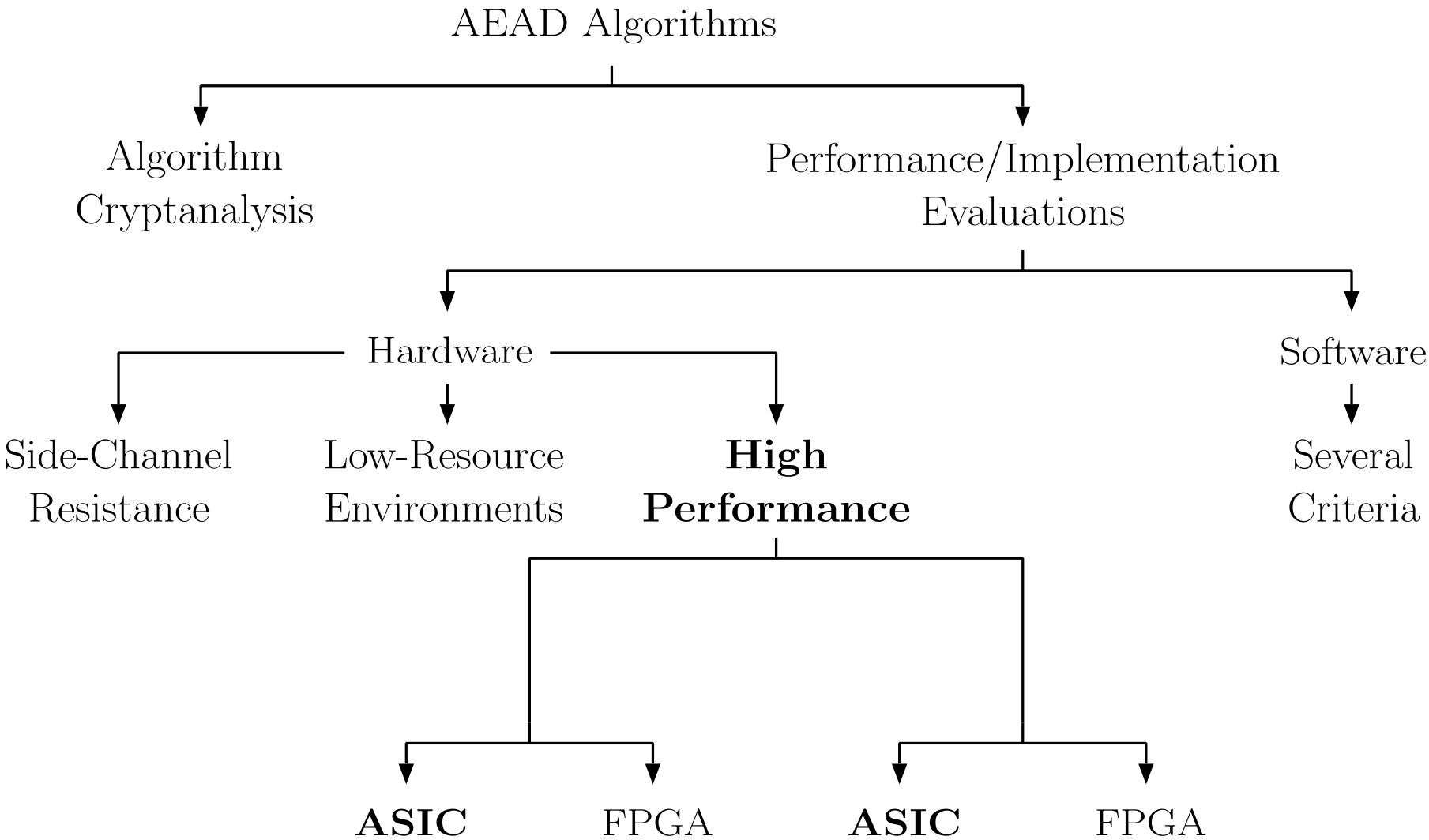
Michael Muehlberghuber
Frank K. Gürkaynak

ETH zürich
Integrated Systems Laboratory (IIS)

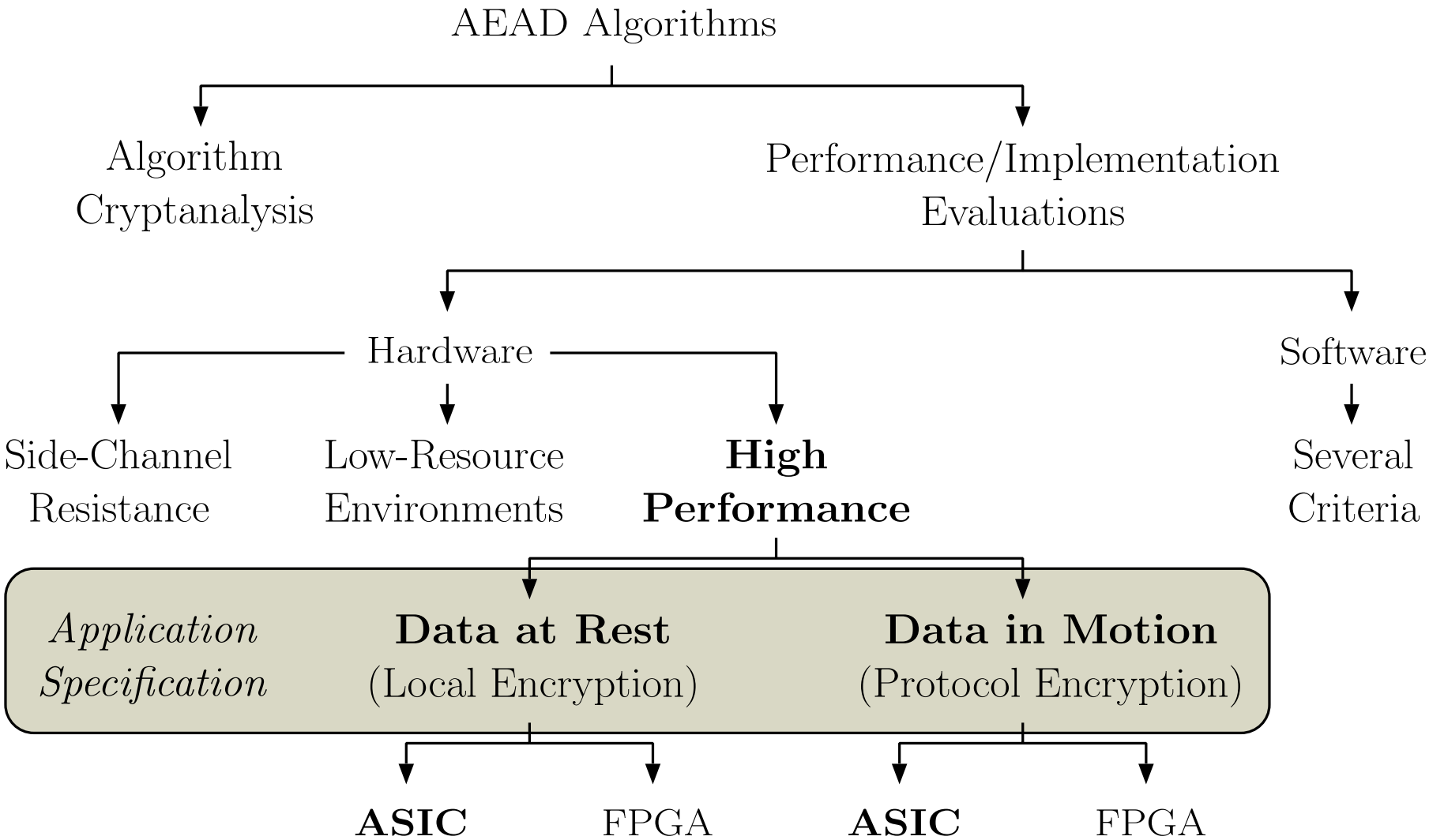
Potential CAESAR Evaluation Criteria



Potential CAESAR Evaluation Criteria



Potential CAESAR Evaluation Criteria



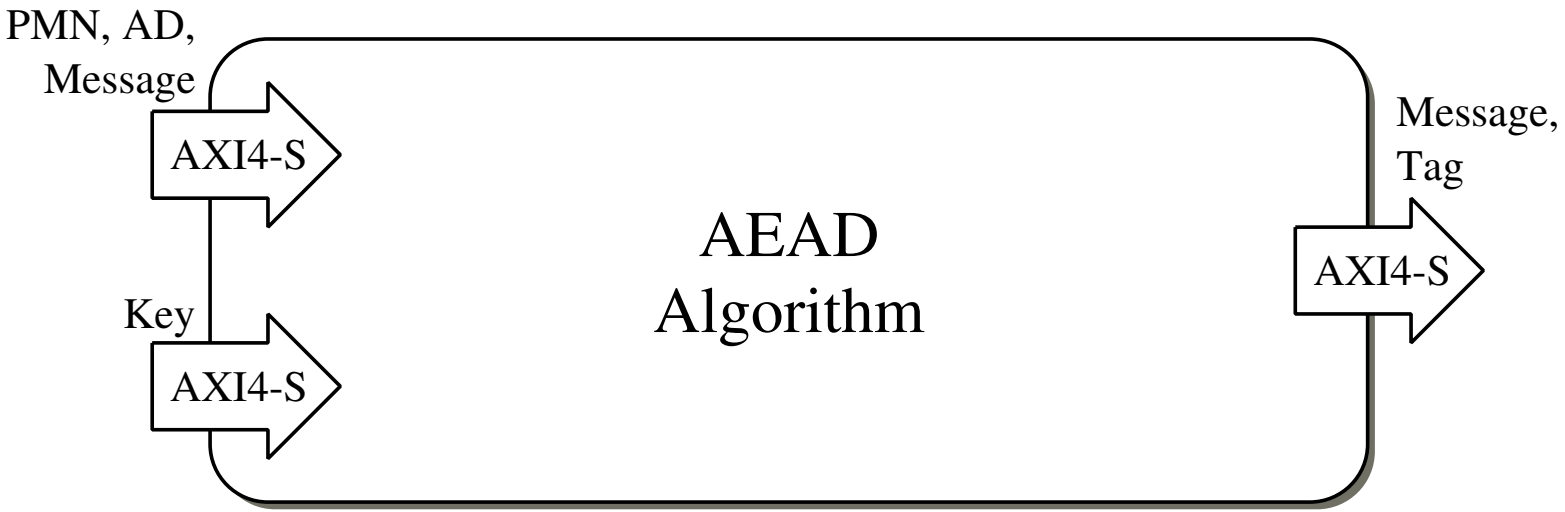
Outline

General Hardware Design Requirements

- Prioritize authors' suggested primary algorithm versions
- Both encryption and decryption must be supported

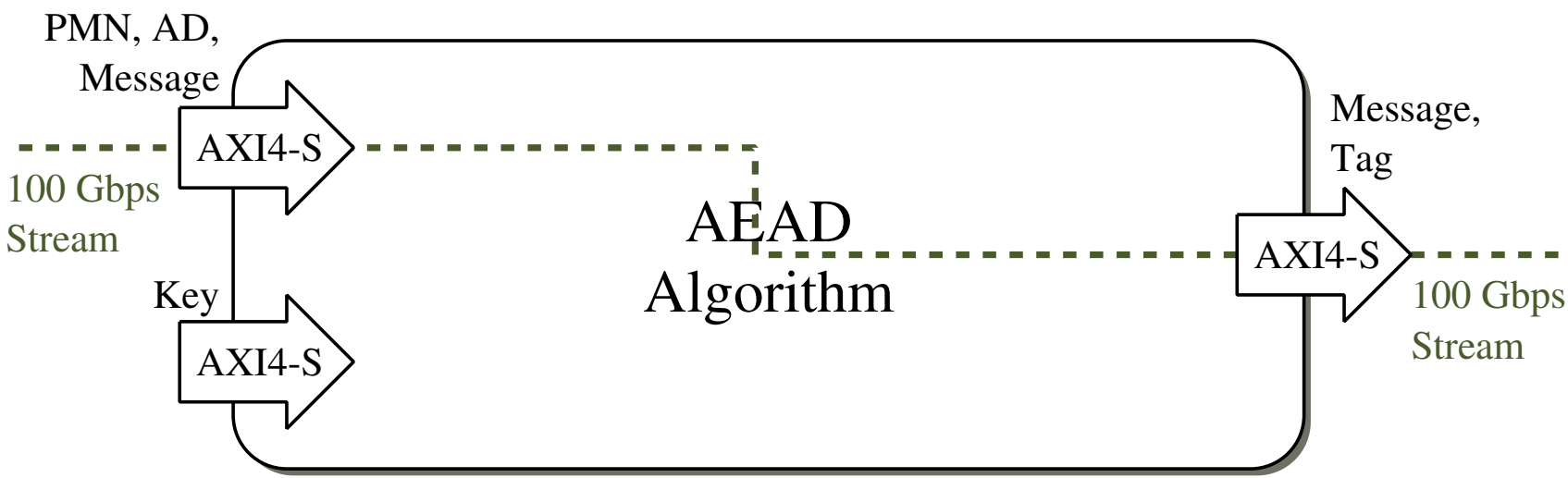
General Hardware Design Requirements

- Prioritize authors' suggested primary algorithm versions
- Both encryption and decryption must be supported
- Consistent I/O interface: Three AXI-4 Stream interfaces



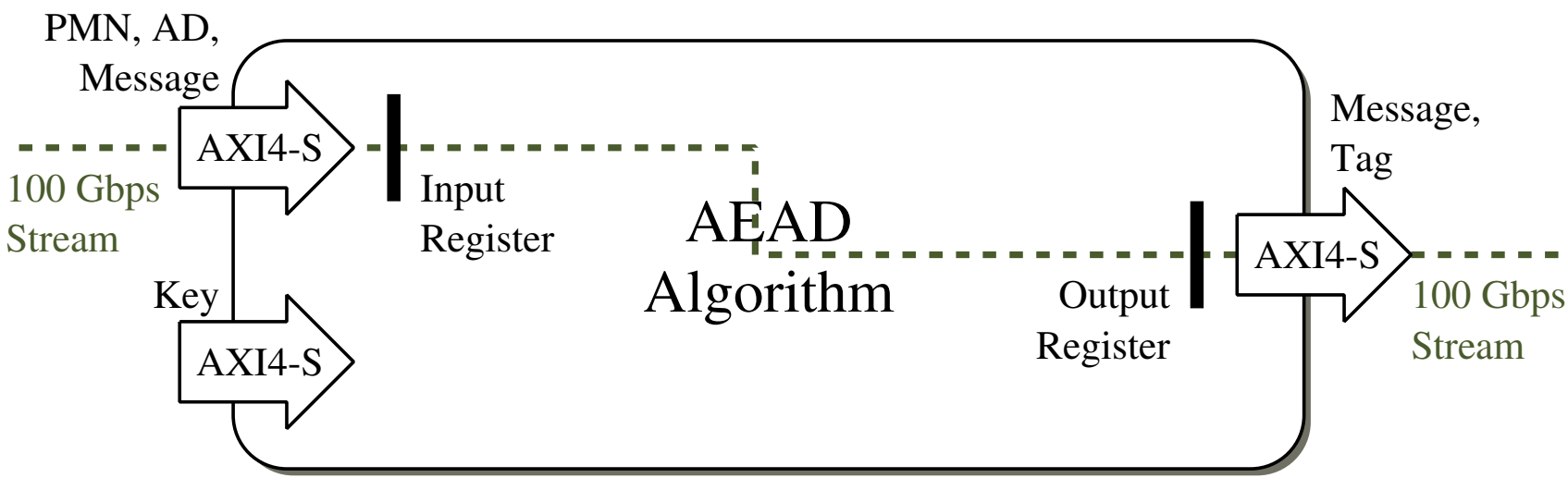
General Hardware Design Requirements

- Prioritize authors' suggested primary algorithm versions
- Both encryption and decryption must be supported
- Consistent I/O interface: Three AXI-4 Stream interfaces



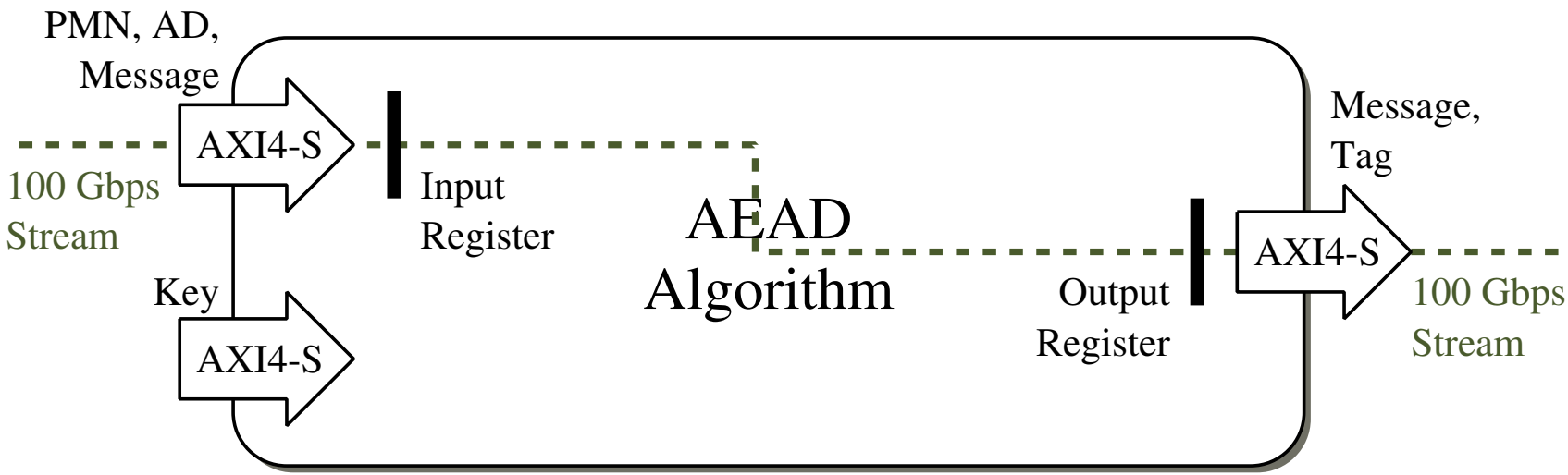
General Hardware Design Requirements

- Prioritize authors' suggested primary algorithm versions
- Both encryption and decryption must be supported
- Consistent I/O interface: Three AXI-4 Stream interfaces
  - *Comparatively* short I/O delays
  - Stallable architecture designs



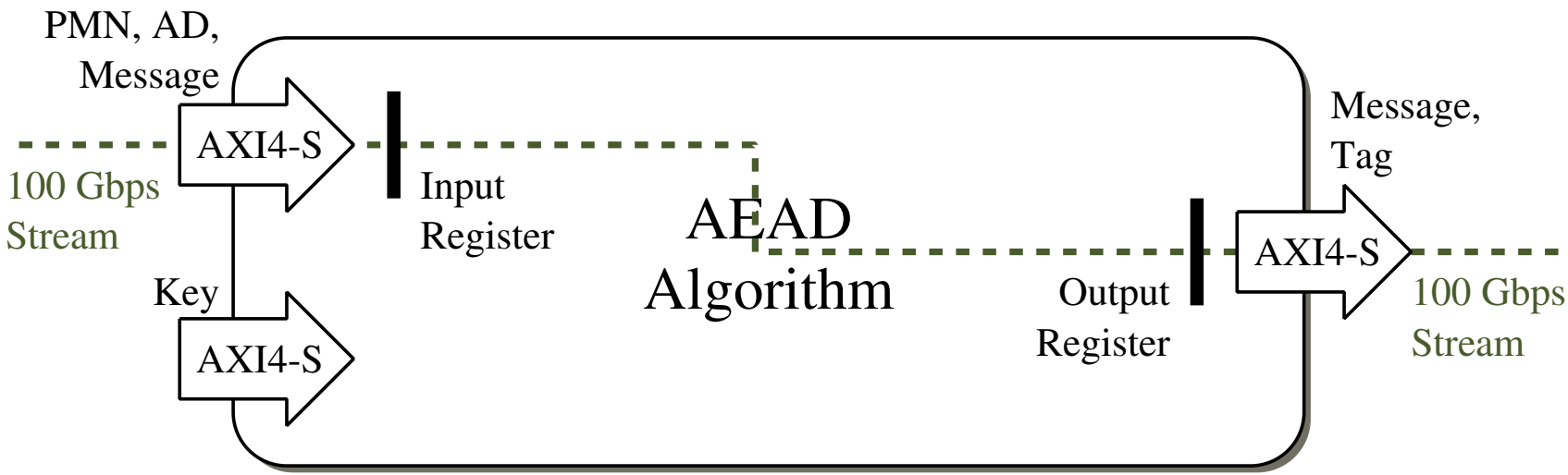
General Hardware Design Requirements

- Prioritize authors' suggested primary algorithm versions
- Both encryption and decryption must be supported
- Consistent I/O interface: Three AXI-4 Stream interfaces
  - *Comparatively* short I/O delays
  - Stallable architecture designs
- Target technology: 65 nm by UMC (aiming at techn. indep.)



General Hardware Design Requirements

- Prioritize authors' suggested primary algorithm versions
- Both encryption and decryption must be supported
- Consistent I/O interface: Three AXI-4 Stream interfaces
  - *Comparatively* short I/O delays
  - Stallable architecture designs
- Target technology: 65 nm by UMC (aiming at techn. indep.)
- No *fancy* hardware archit. transf. (sub-round pipelining, . . .)



Use Case Scenarios

Data at Rest (Local Encryption)

- Huge amount of data is available on site
- Negligible cipherkey and public message number (PMN) changes
- Large amount of associated (AD) and message data

Use Case Scenarios

Data at Rest (Local Encryption)

- Huge amount of data is available on site
- Negligible cipherkey and public message number (PMN) changes
- Large amount of associated (AD) and message data

Typically used within previous work (whether intentional or not)

Use Case Scenarios

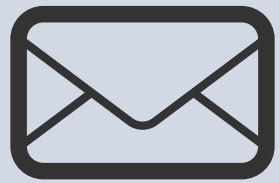
Data at Rest (Local Encryption)

- Huge amount of data is available on site
- Negligible cipherkey and public message number (PMN) changes
- Large amount of associated (AD) and message data

Typically used within previous work (whether intentional or not)

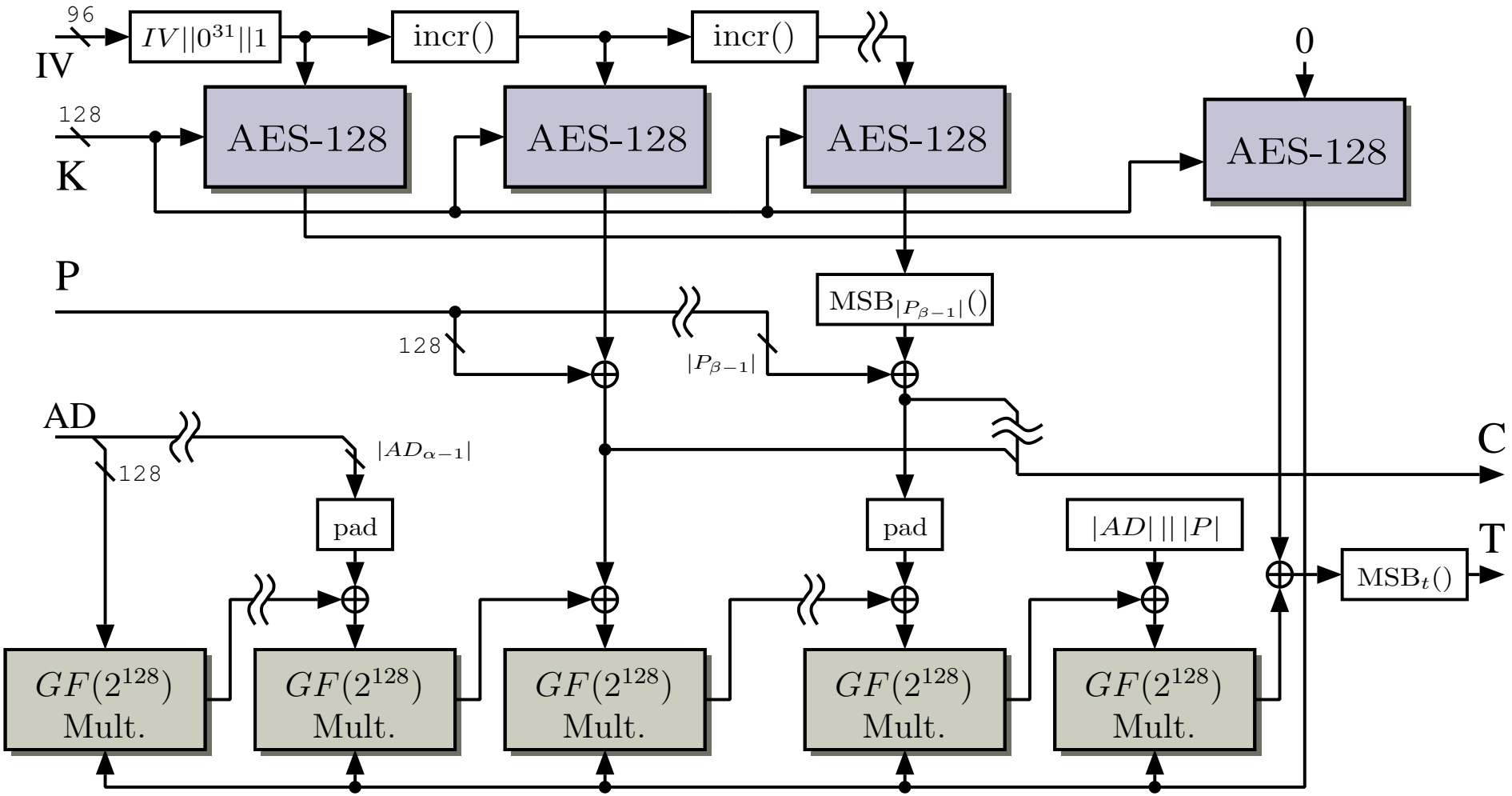
Data in Motion (Protocol Encryption) - 100 Gbps Ethernet

- Rare cipherkey changes
- PMN changes for every transmitted packet
- AD and message data size adheres to a certain range
- Minimum amount of AD stemming from the header

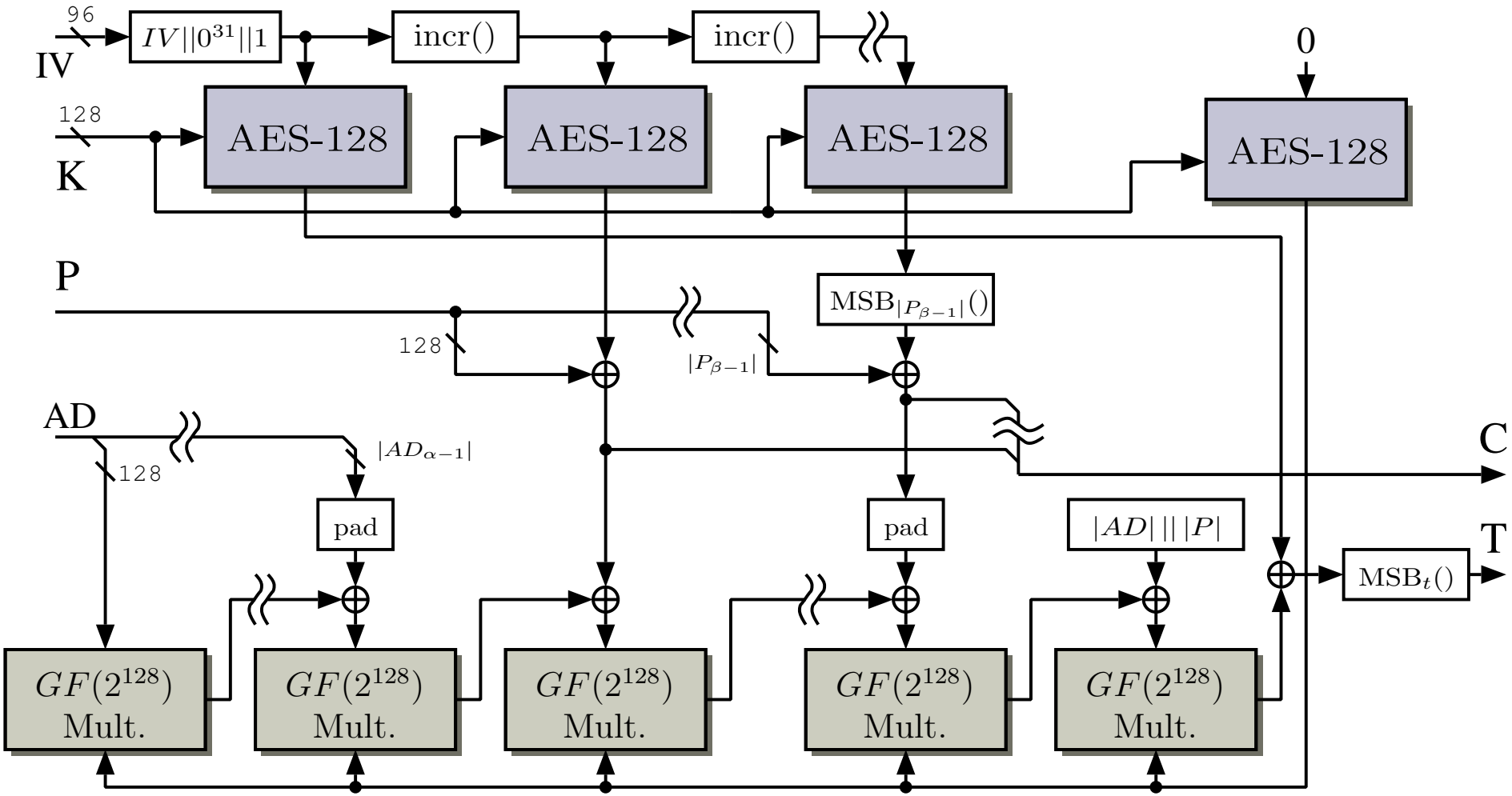


Use Case I
Data at Rest

GCM-AES-128

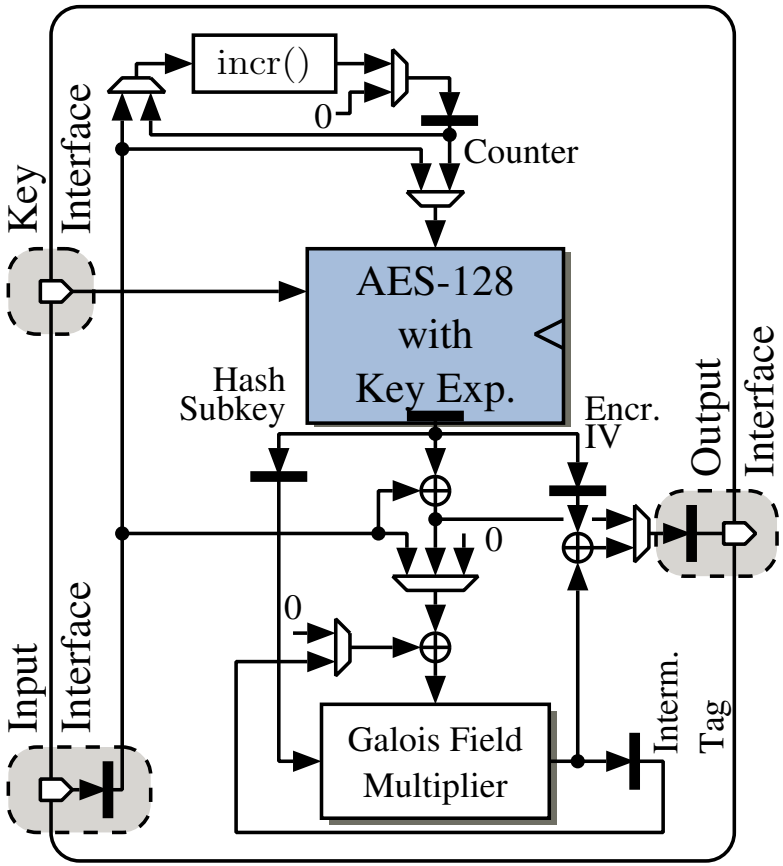

96
IV
$IV||0^{31}||1$
incr()
incr()
0
128
K
AES-128
AES-128
AES-128
AES-128
P
$\text{MSB}_{|P_{\beta-1}|}()$
128
$|P_{\beta-1}|$
AD
128
$|AD_{\alpha-1}|$
C
pad
pad
$|AD||\,|P|$
T
$\text{MSB}_t()$
$GF(2^{128})$ Mult.
$GF(2^{128})$ Mult.
$GF(2^{128})$ Mult.
$GF(2^{128})$ Mult.
$GF(2^{128})$ Mult.

GCM-AES-128

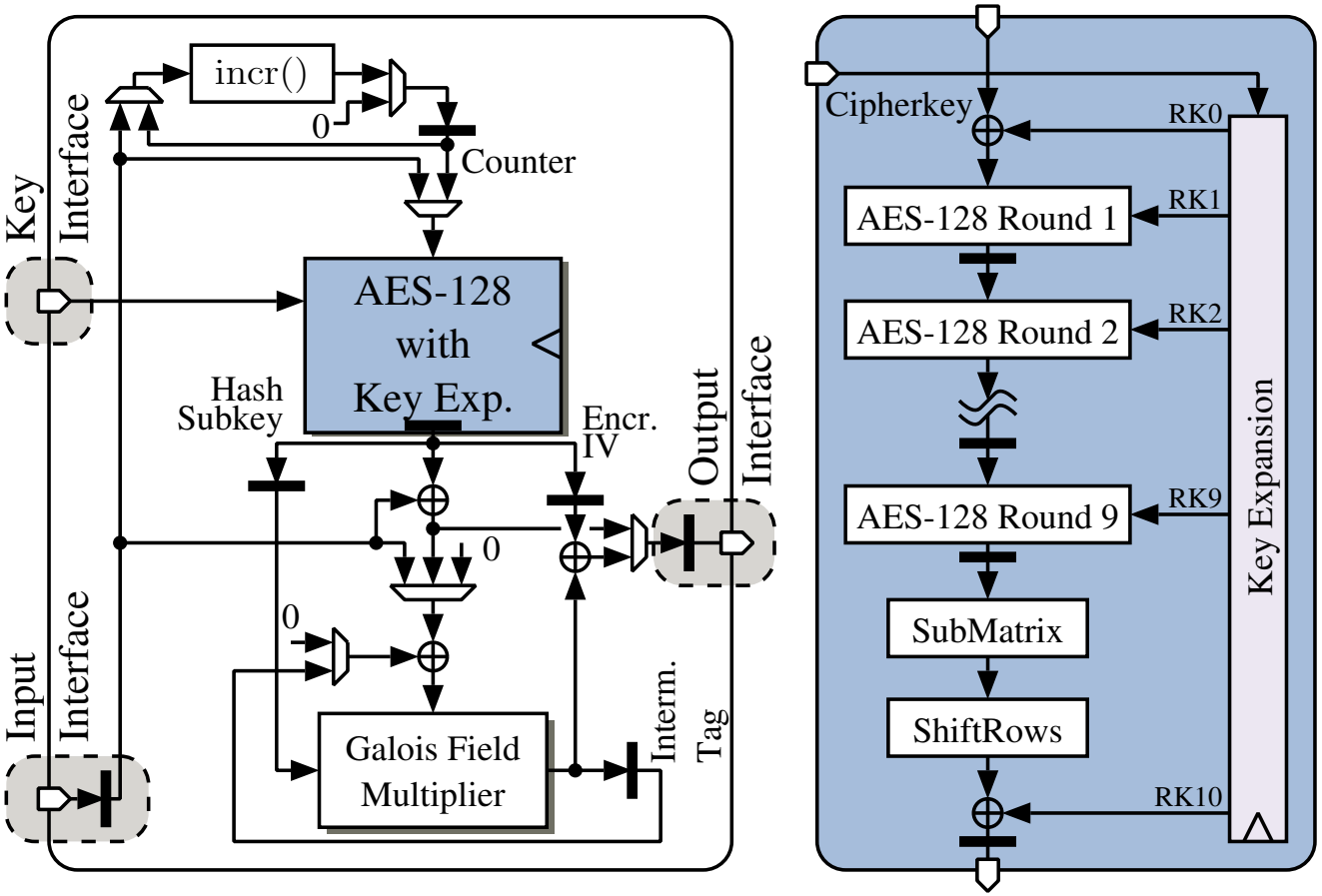


- Major blocks: Block cipher and $GF(2^{128})$ multiplier

100 Gbps GCM-AES - Data at Rest Design

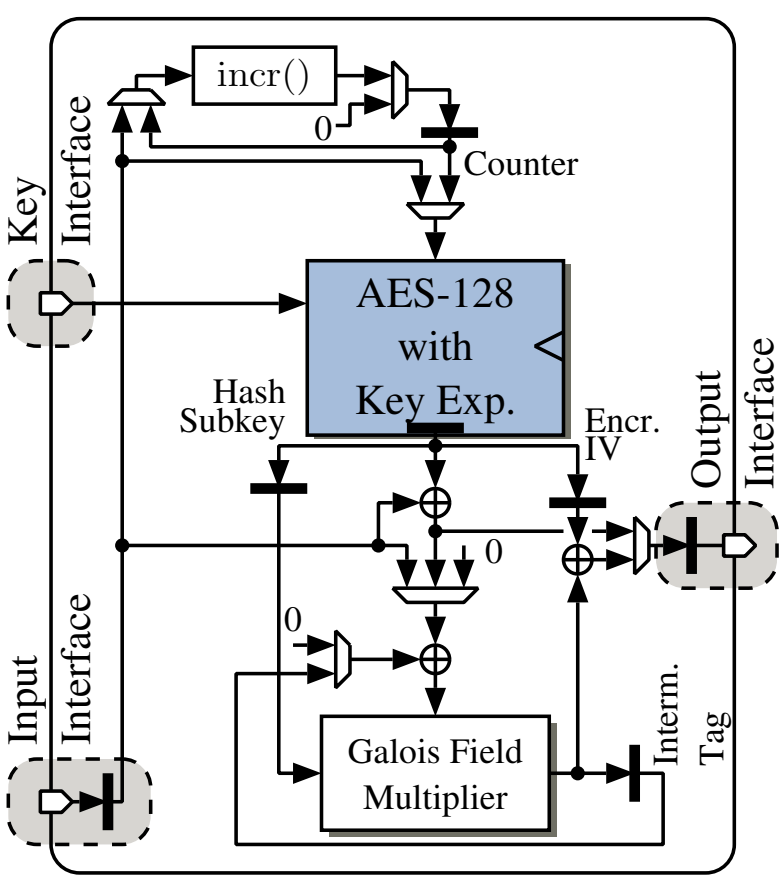


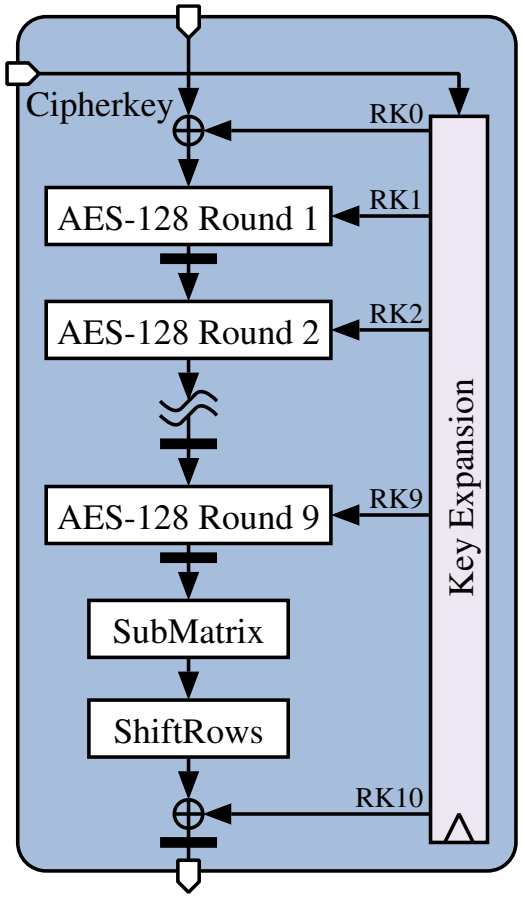
100 Gbps GCM-AES - Data at Rest Design

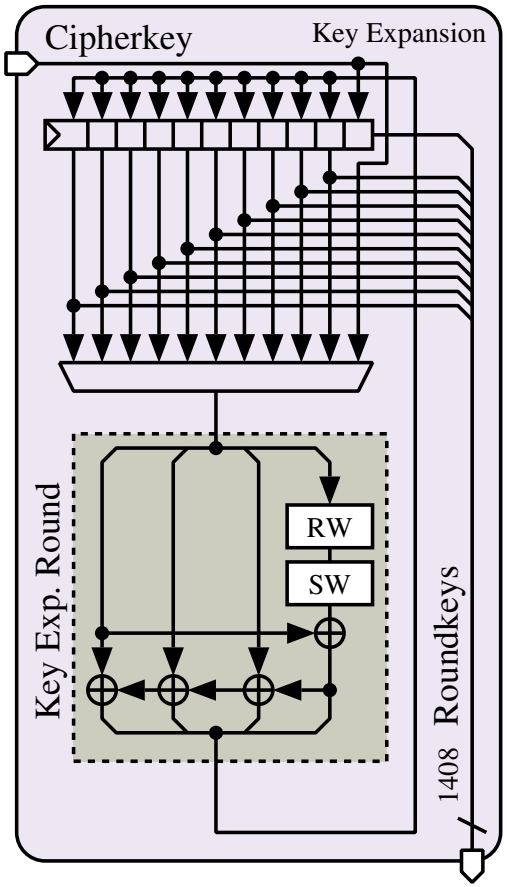


- Fully-unrolled, single-core AES-128

100 Gbps GCM-AES - Data at Rest Design

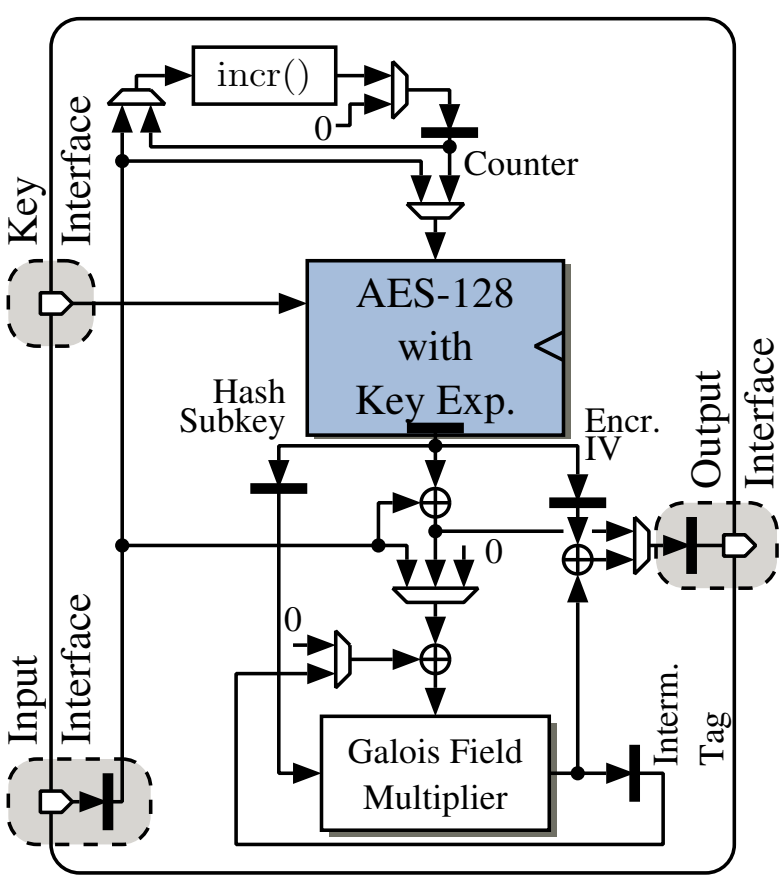


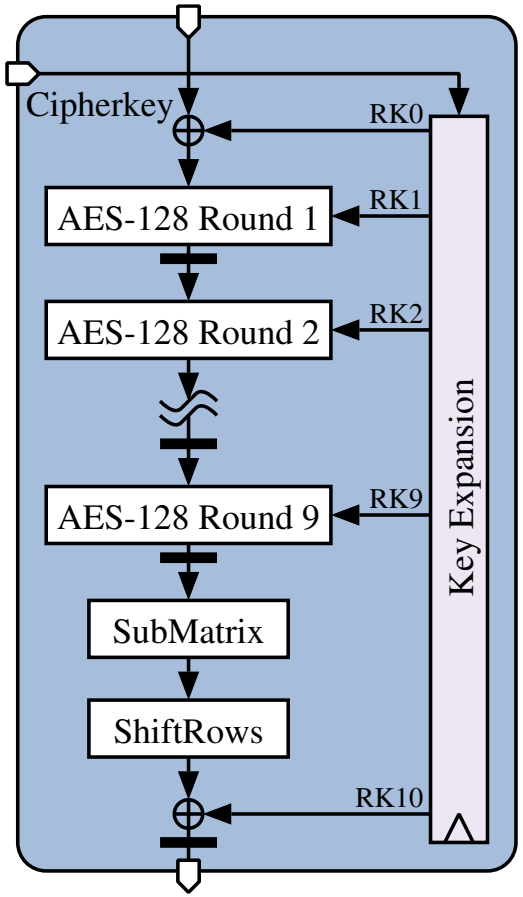


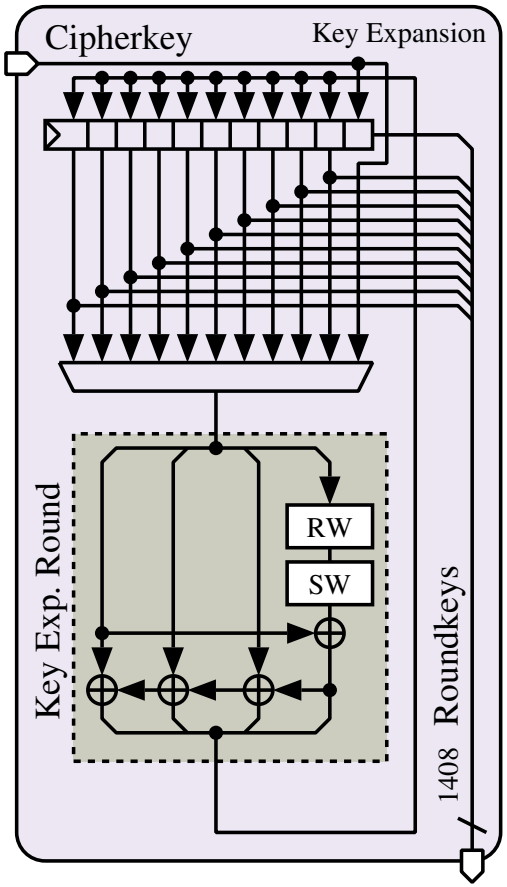


- Fully-unrolled, single-core AES-128 with iterative key expansion

100 Gbps GCM-AES - Data at Rest Design



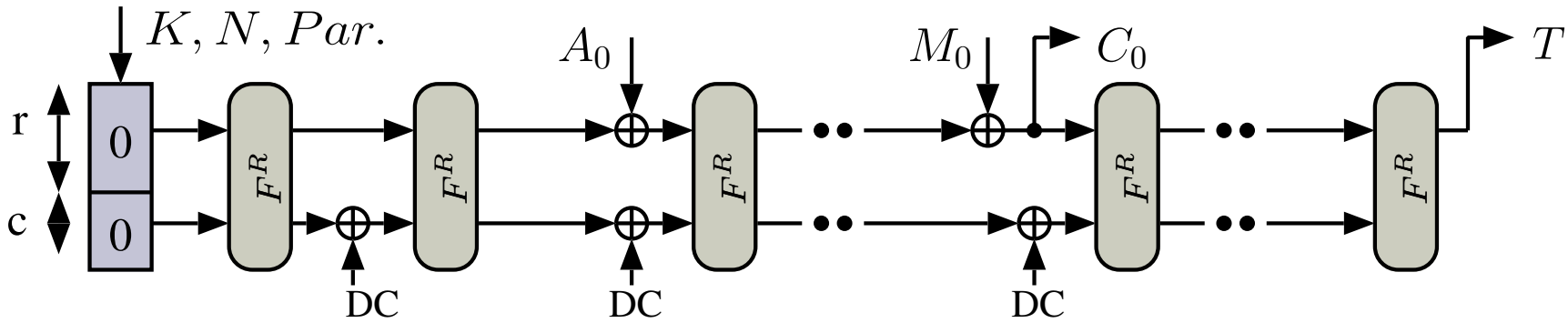




- Fully-unrolled, single-core AES-128 with iterative key expansion
- Combinational, bit-parallel $GF(2^{128})$ multiplier

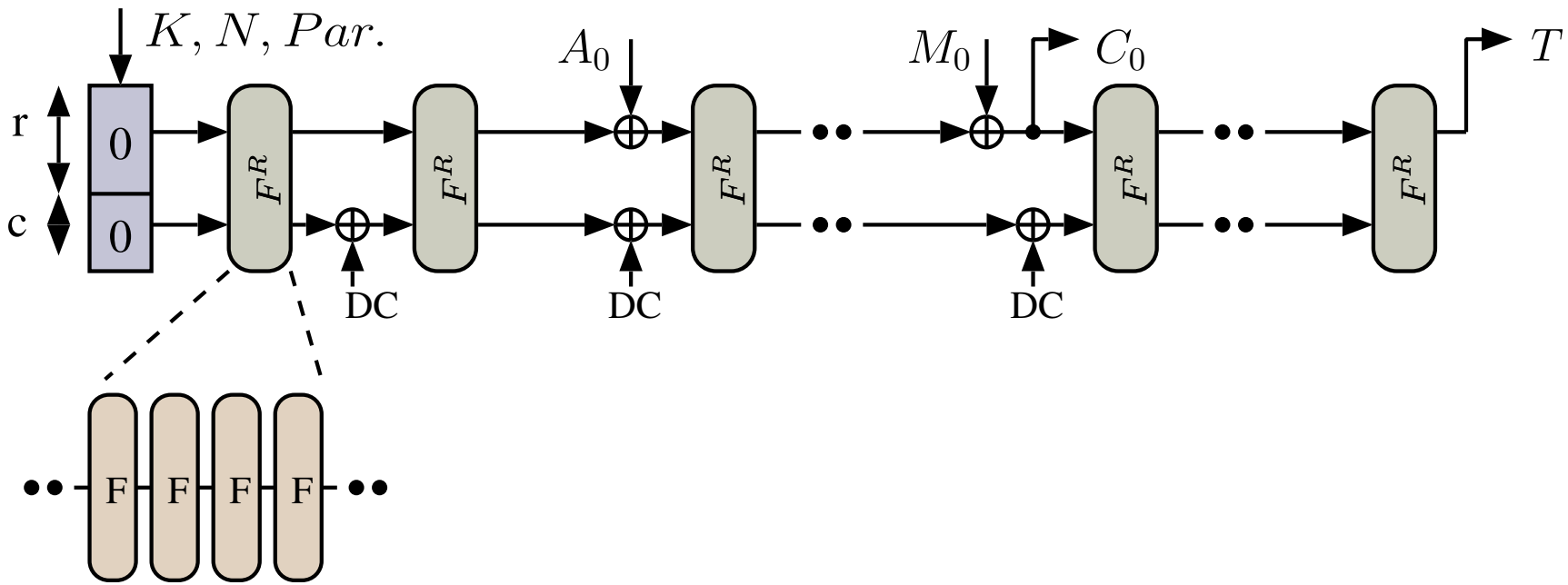
NORX [1]

- Permutation-based algorithm using the *monkeyDuplex* [2] construction
- Primary recommendation: NORX64-4-1
  - 16 word state, each word 64 bits, 4 rounds
  - Key size = 256 bits, tag size = 128 bits



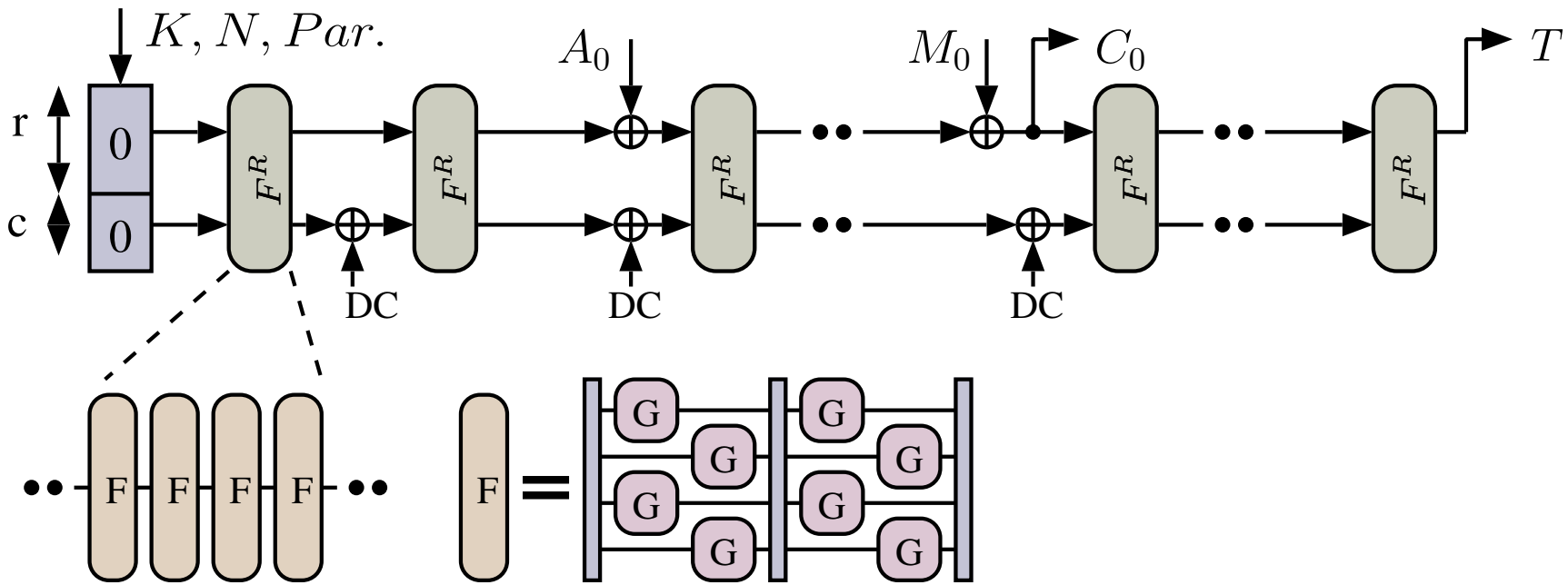
NORX [1]

- Permutation-based algorithm using the *monkeyDuplex* [2] construction
- Primary recommendation: NORX64-4-1
  - 16 word state, each word 64 bits, 4 rounds
  - Key size = 256 bits, tag size = 128 bits



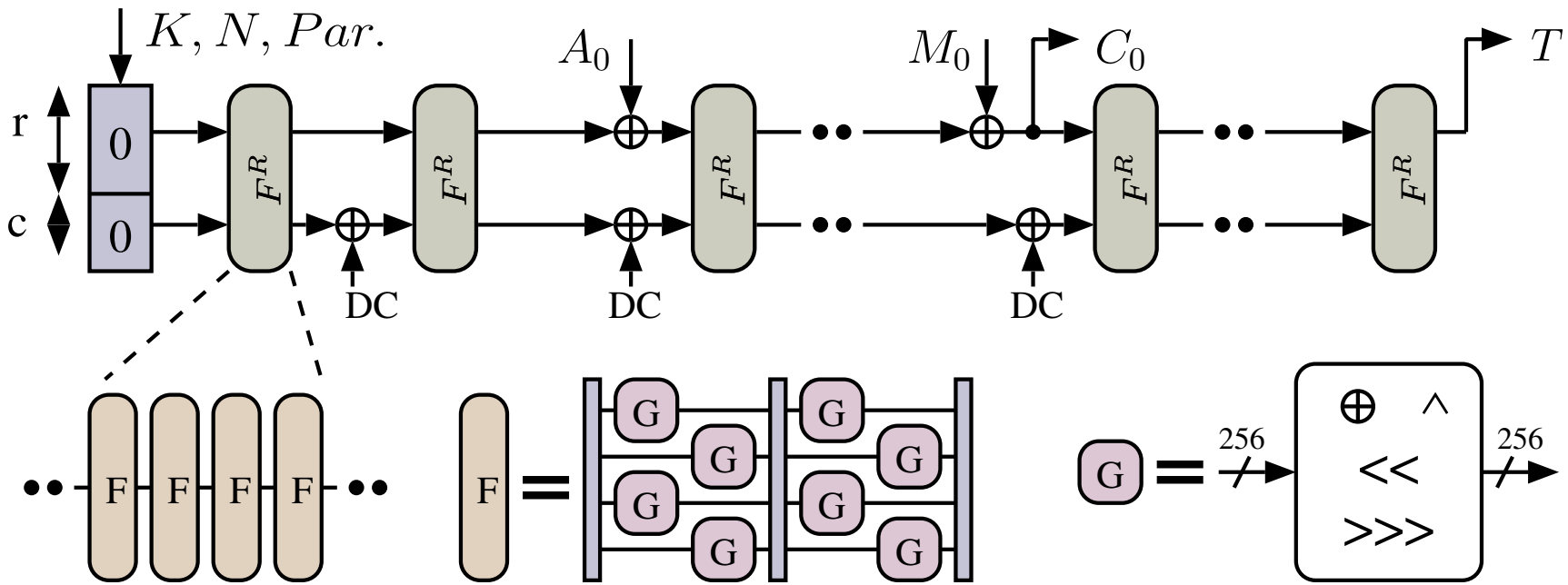
NORX [1]

- Permutation-based algorithm using the *monkeyDuplex* [2] construction
- Primary recommendation: NORX64-4-1
  - 16 word state, each word 64 bits, 4 rounds
  - Key size = 256 bits, tag size = 128 bits

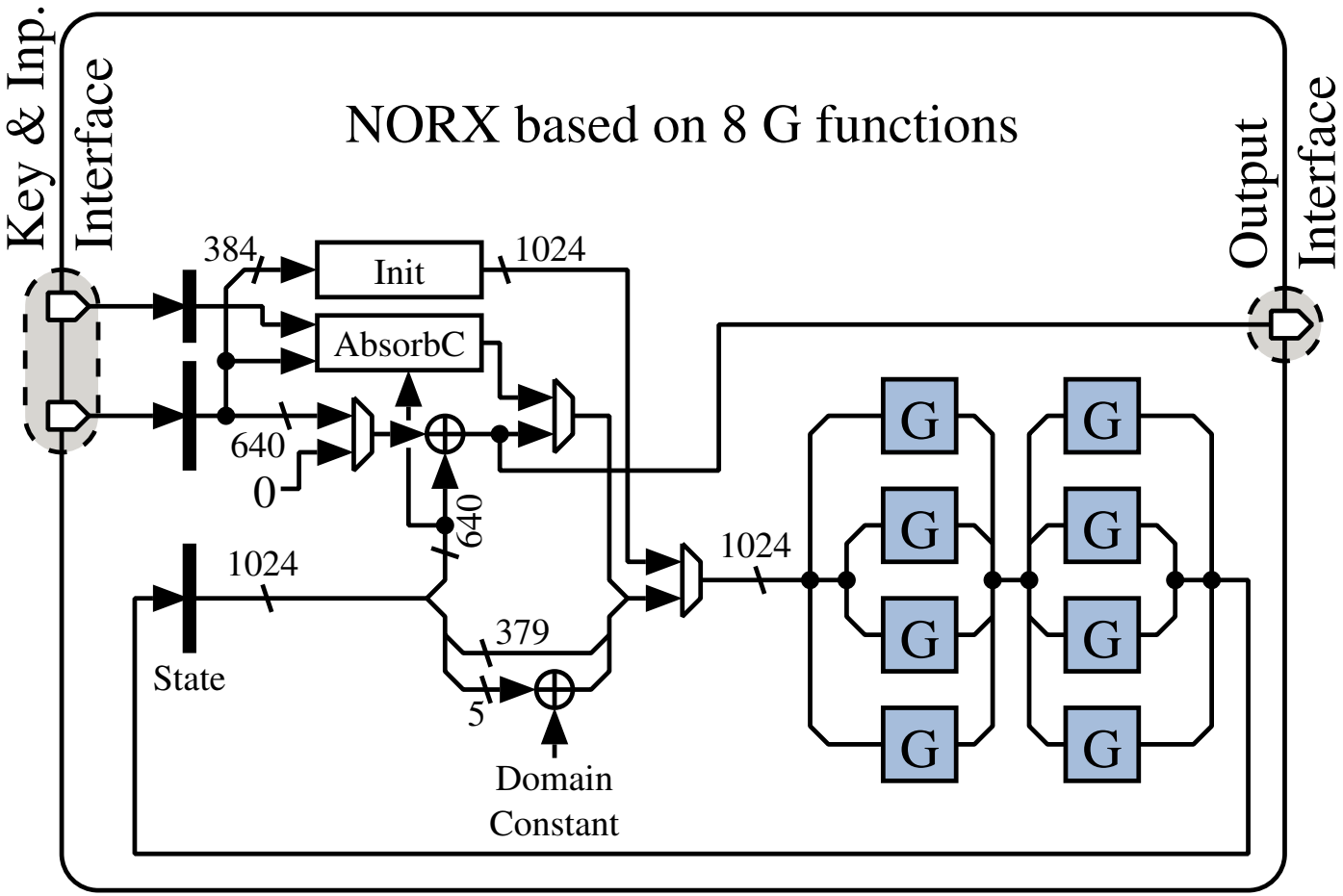


NORX [1]

- Permutation-based algorithm using the *monkeyDuplex* [2] construction
- Primary recommendation: NORX64-4-1
  - 16 word state, each word 64 bits, 4 rounds
  - Key size = 256 bits, tag size = 128 bits

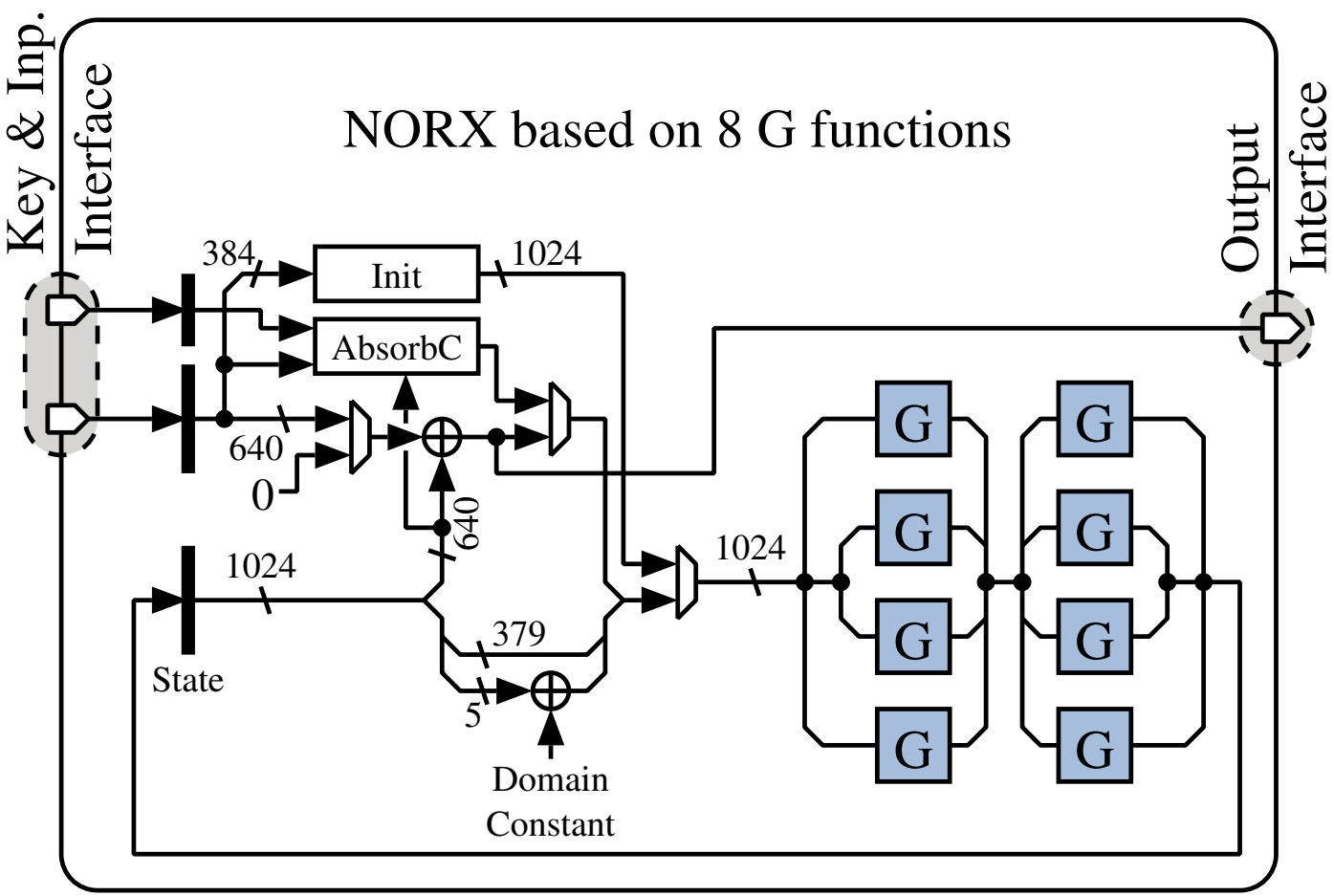


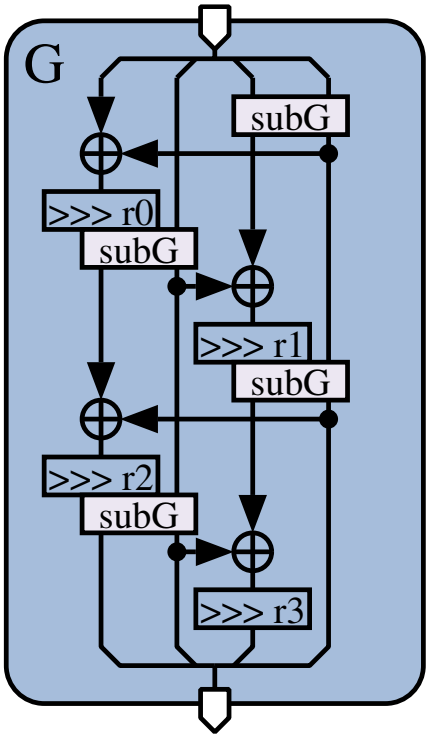
100 Gbps NORX - Data at Rest Design



- NORX architecture based on 8 G functions
- Large fan-outs due to 1024 bit state
- *Comparatively* short I/O timings

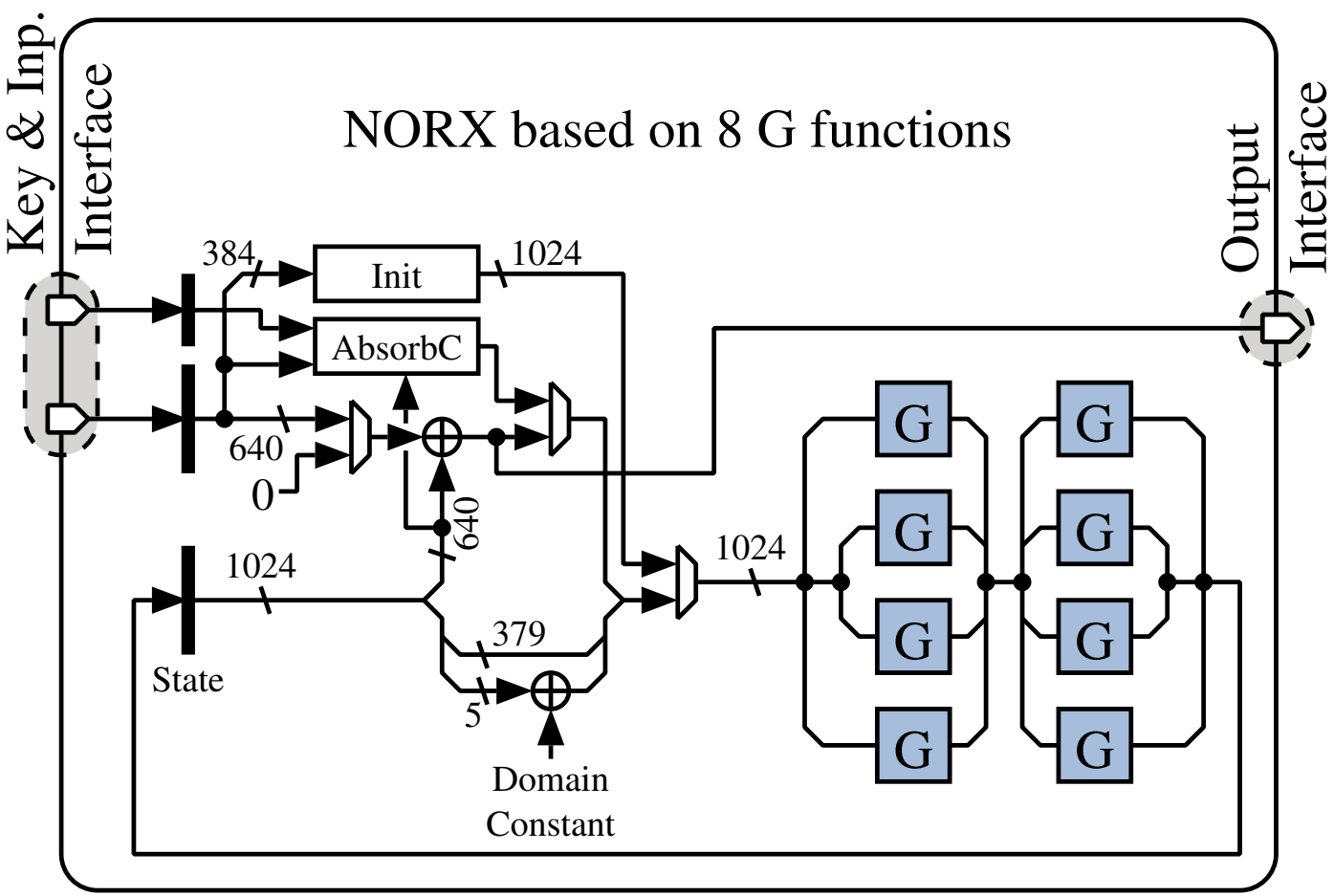
100 Gbps NORX - Data at Rest Design

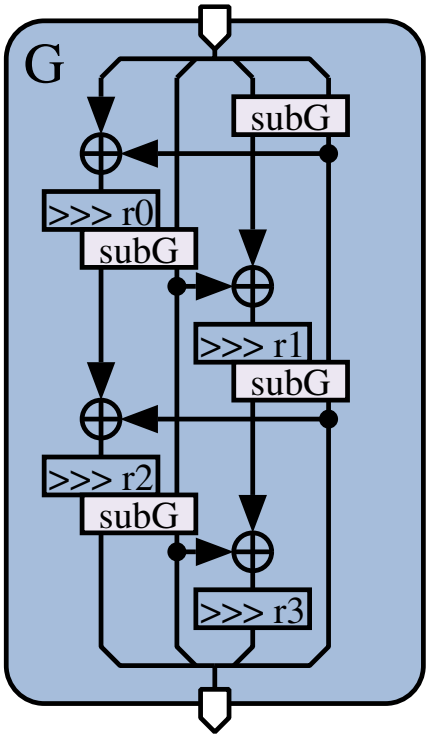


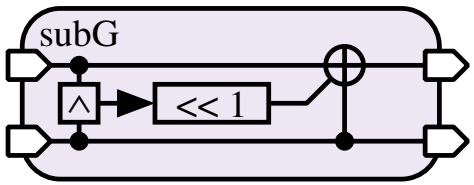


- NORX architecture based on 8 G functions
- Large fan-outs due to 1024 bit state
- *Comparatively* short I/O timings

100 Gbps NORX - Data at Rest Design

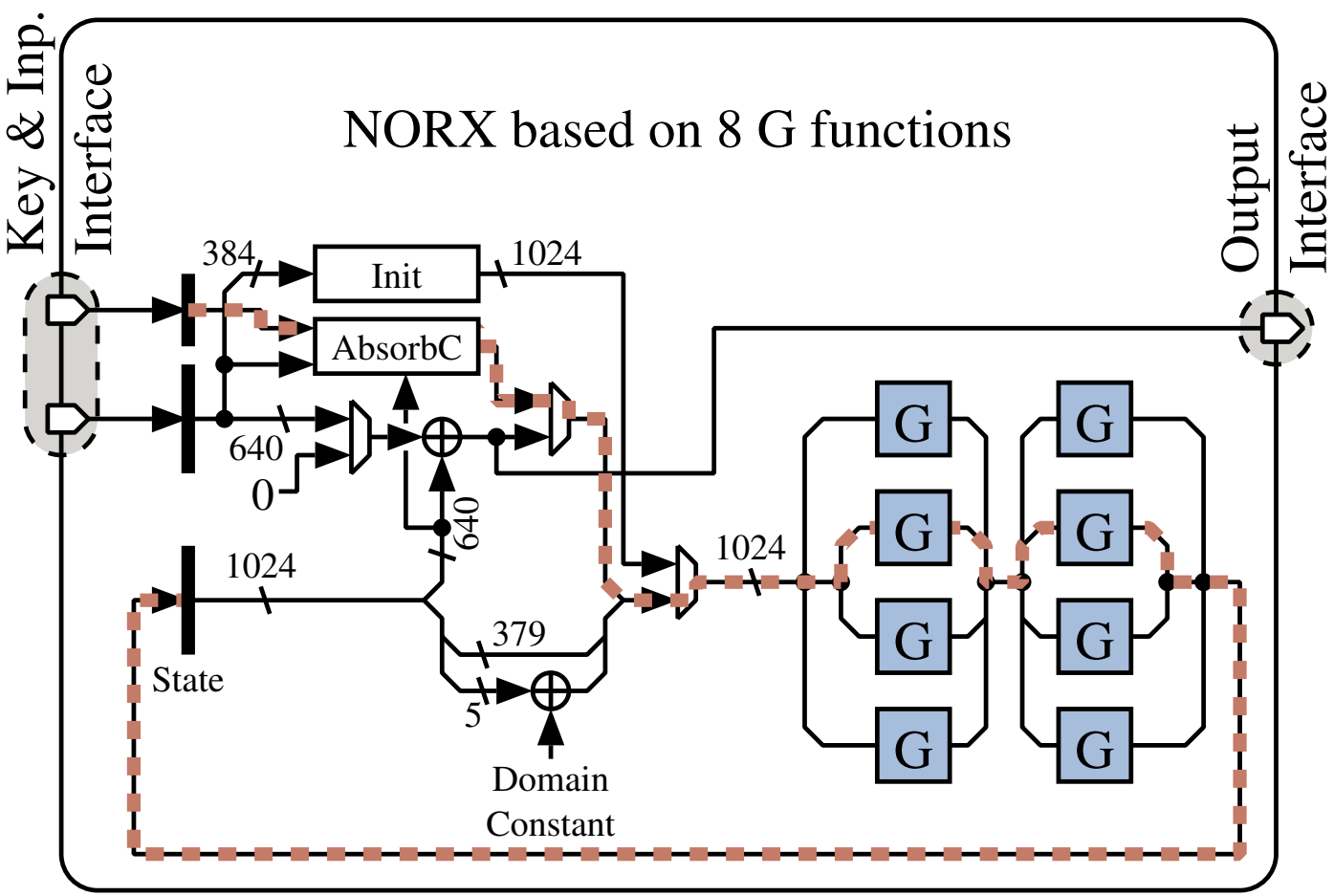


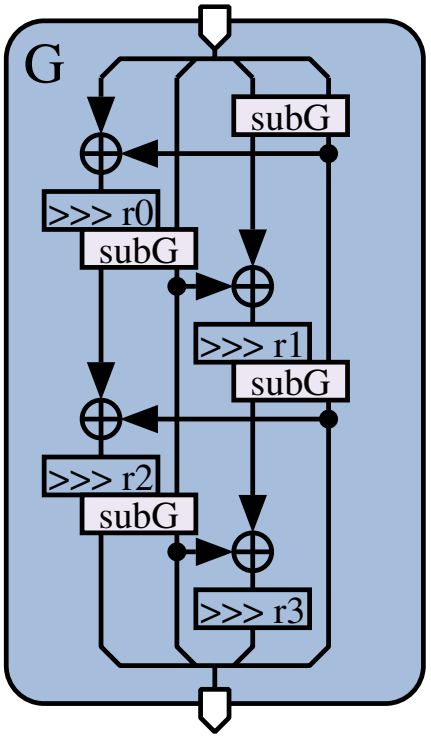


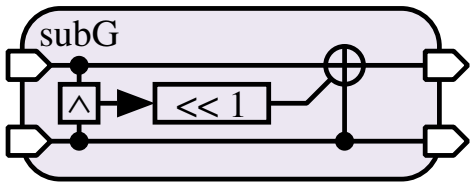


- NORX architecture based on 8 G functions
- Large fan-outs due to 1024 bit state
- *Comparatively* short I/O timings

100 Gbps NORX - Data at Rest Design

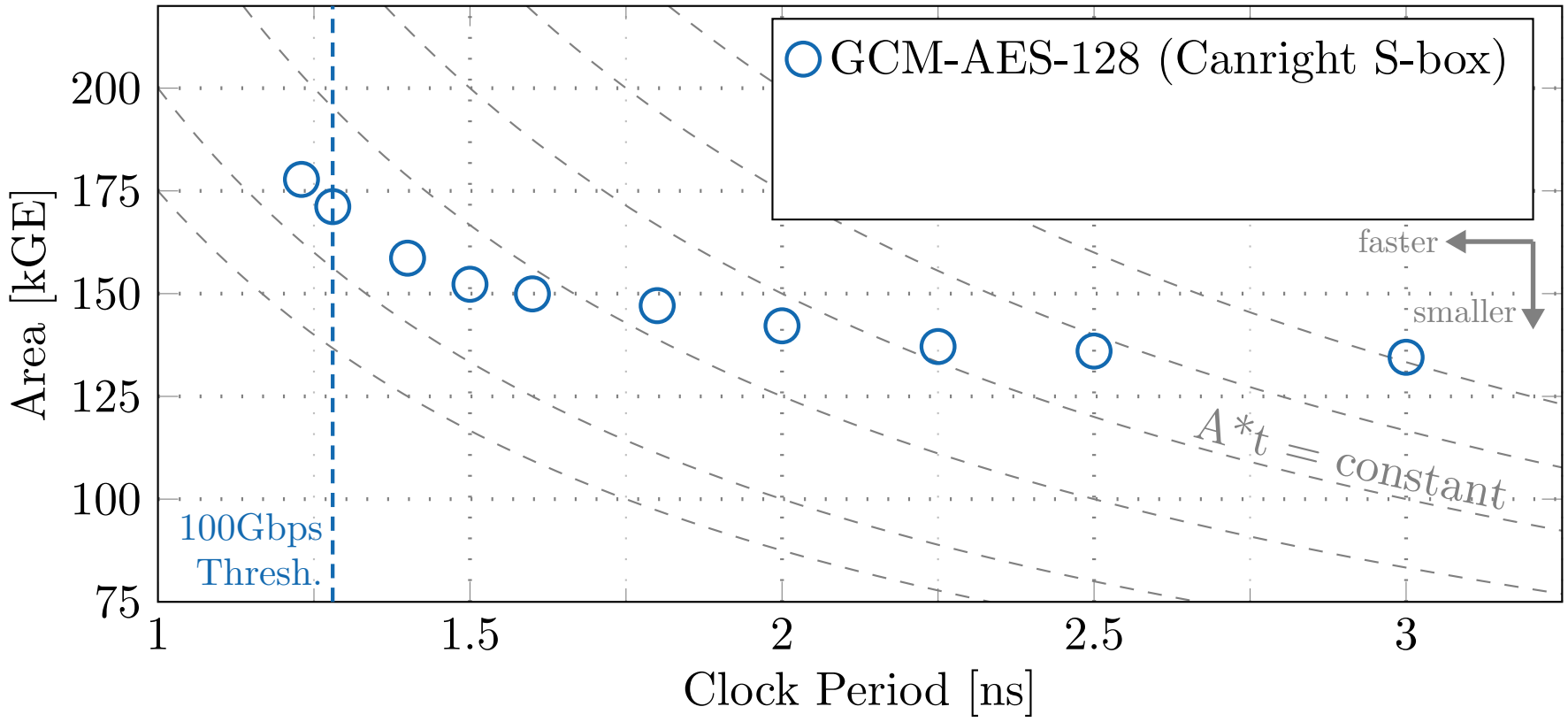






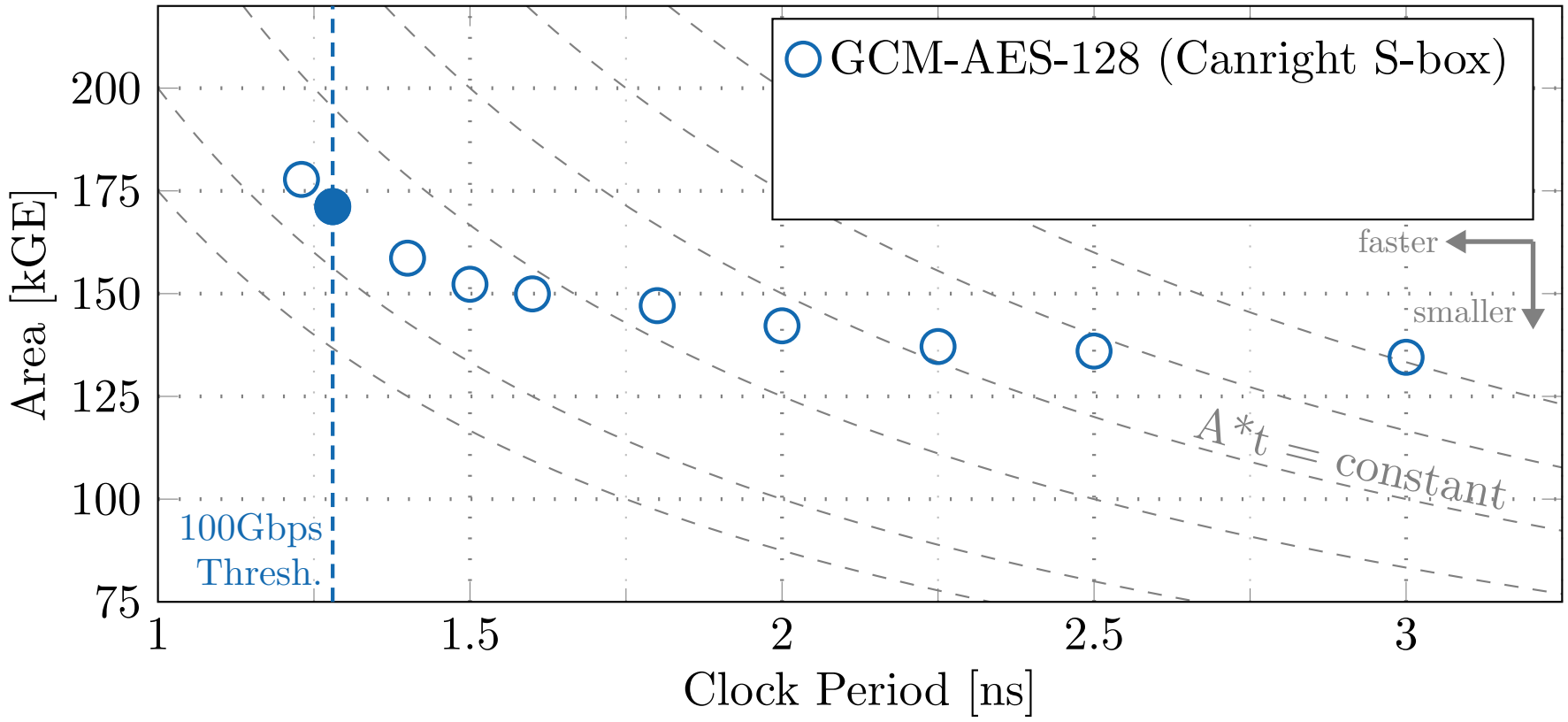
- NORX architecture based on 8 G functions
- Large fan-outs due to 1024 bit state
- *Comparatively* short I/O timings

Data at Rest - Synthesis Results (UMC 65 nm)



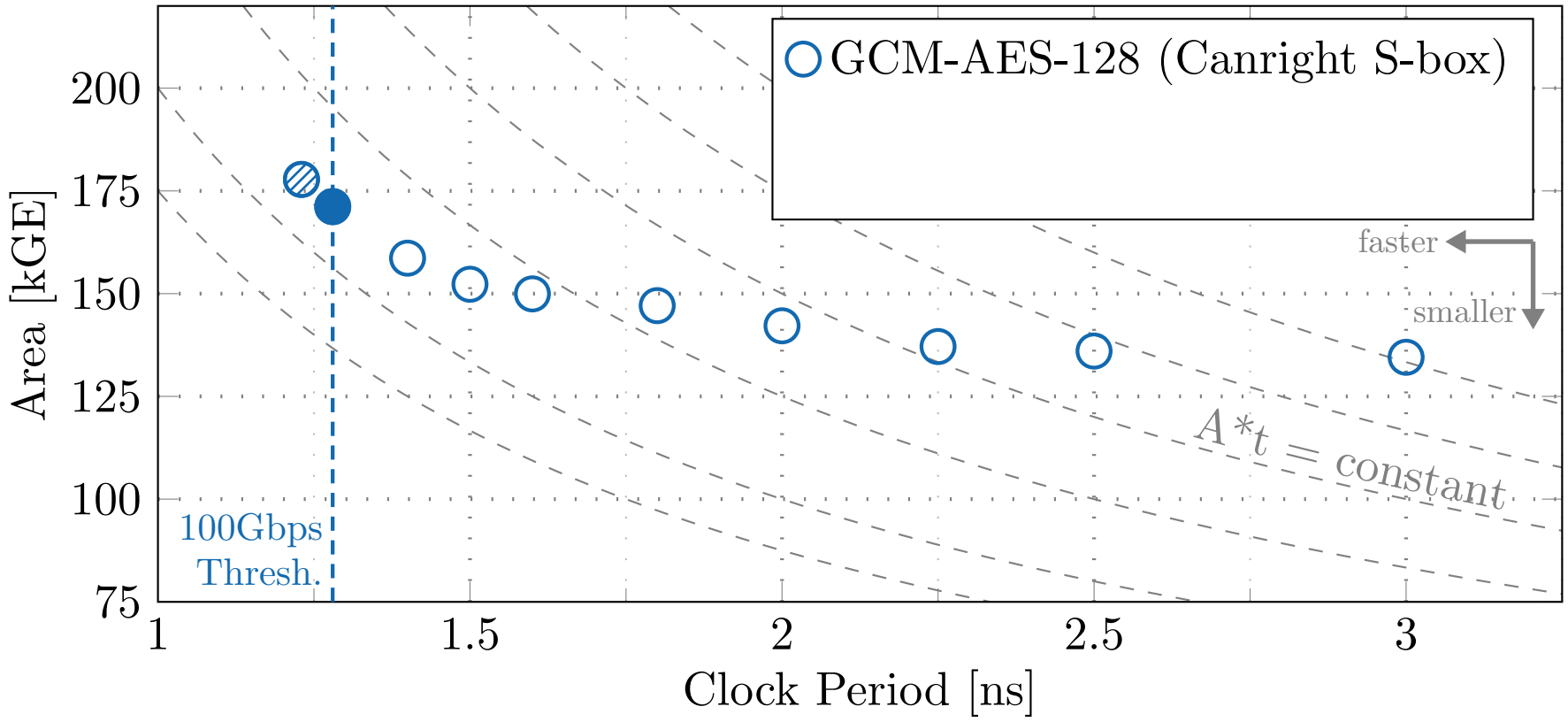
Design	100 Gbit/s Performance				Maximum TP Performance				
	Area [kGE]	f_{100} [MHz]	Efficiency [kbps/GE]	[%]	Area [kGE]	f_{max} [MHz]	TP [Gbps]	Efficiency [kbps/GE]	[%]

Data at Rest - Synthesis Results (UMC 65 nm)



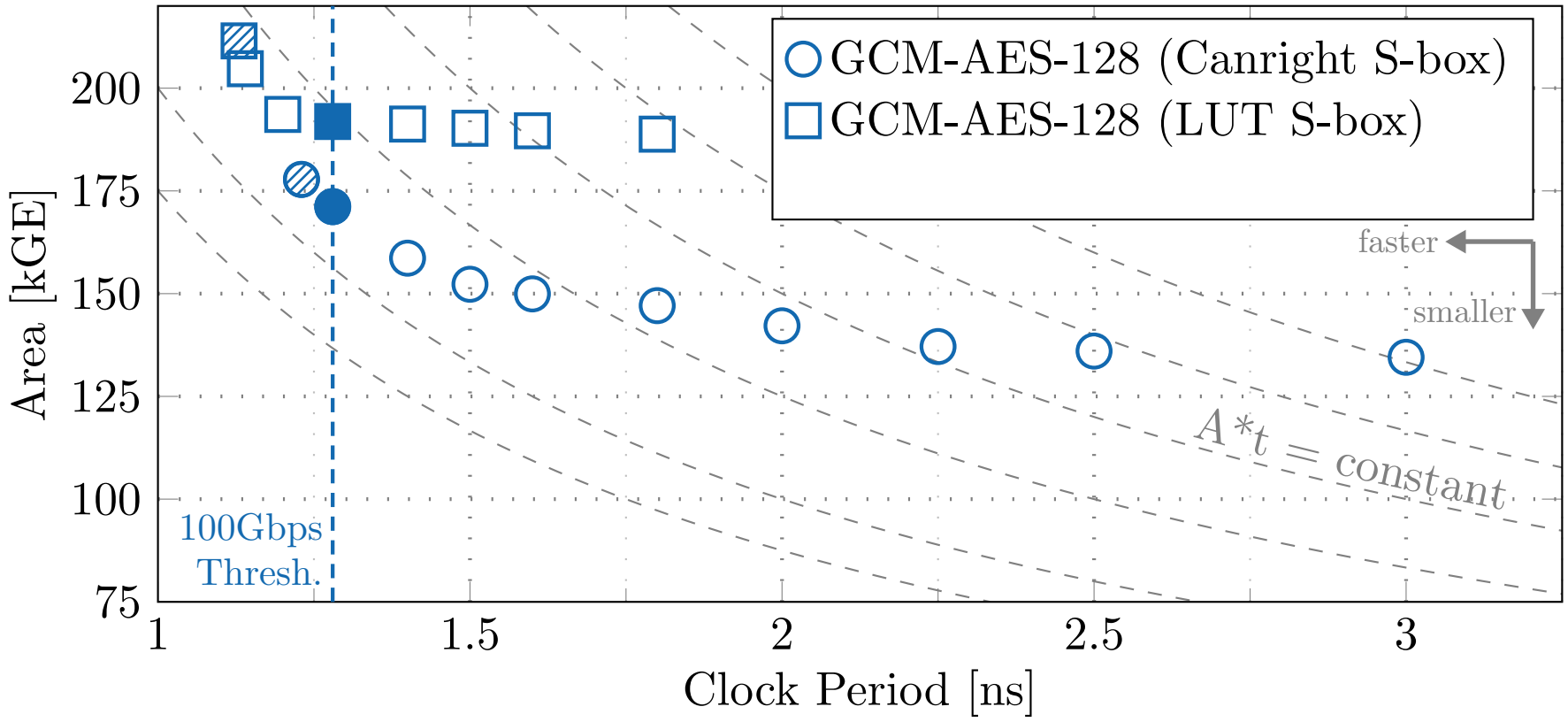
Design	100 Gbit/s Performance				Maximum TP Performance				
	Area [kGE]	f_{100} [MHz]	Efficiency [kbps/GE]	[%]	Area [kGE]	f_{max} [MHz]	TP [Gbps]	Efficiency [kbps/GE]	[%]
GCM-AES (Can)	171.2	781.3	● 584.2	**100**					

Data at Rest - Synthesis Results (UMC 65 nm)



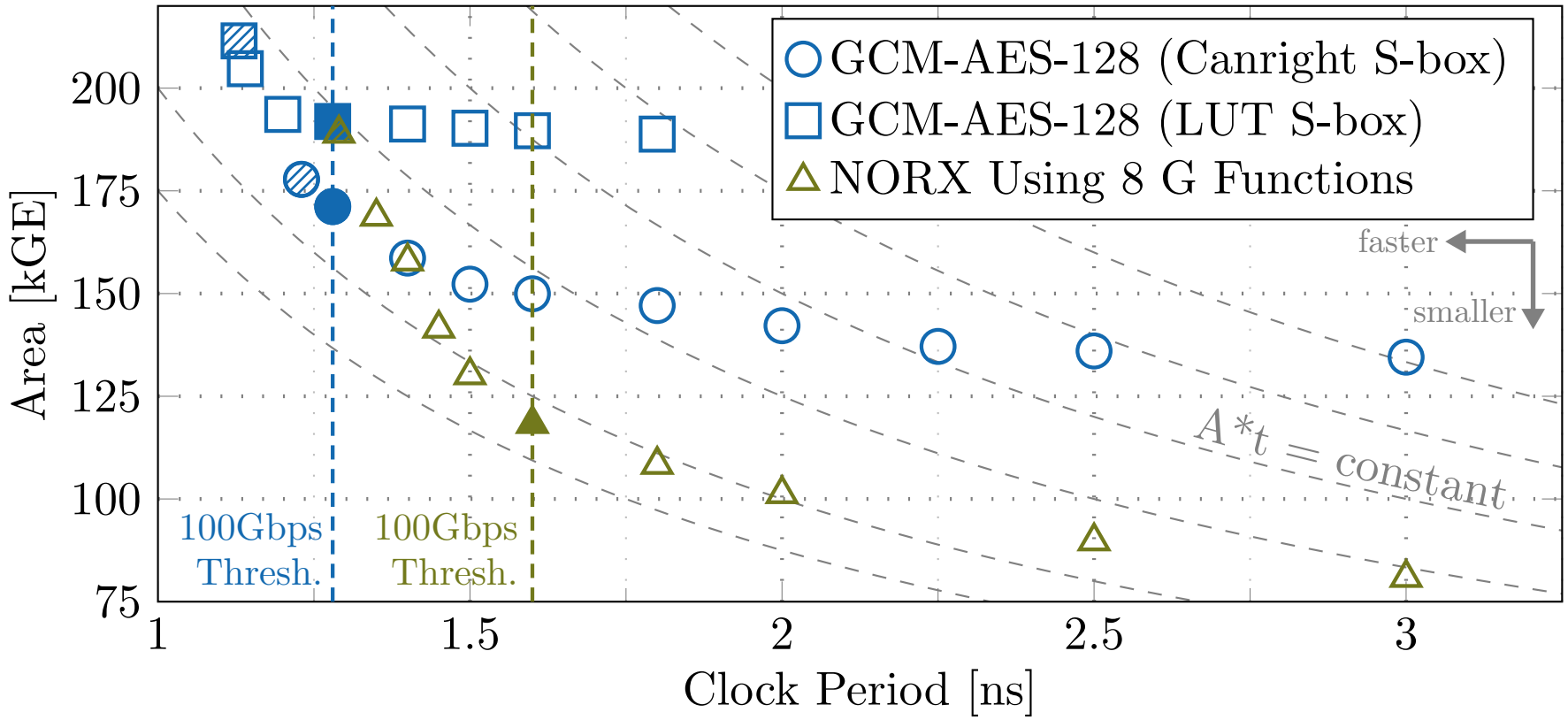
Design	100 Gbit/s Performance				Maximum TP Performance				
	Area [kGE]	f_{100} [MHz]	Efficiency [kbps/GE]	[%]	Area [kGE]	f_{max} [MHz]	TP [Gbps]	Efficiency [kbps/GE]	[%]
GCM-AES (Can)	171.2	781.3	● 584.2	**100**	177.8	813.0	104.1	◍ 585.3	**100**

Data at Rest - Synthesis Results (UMC 65 nm)



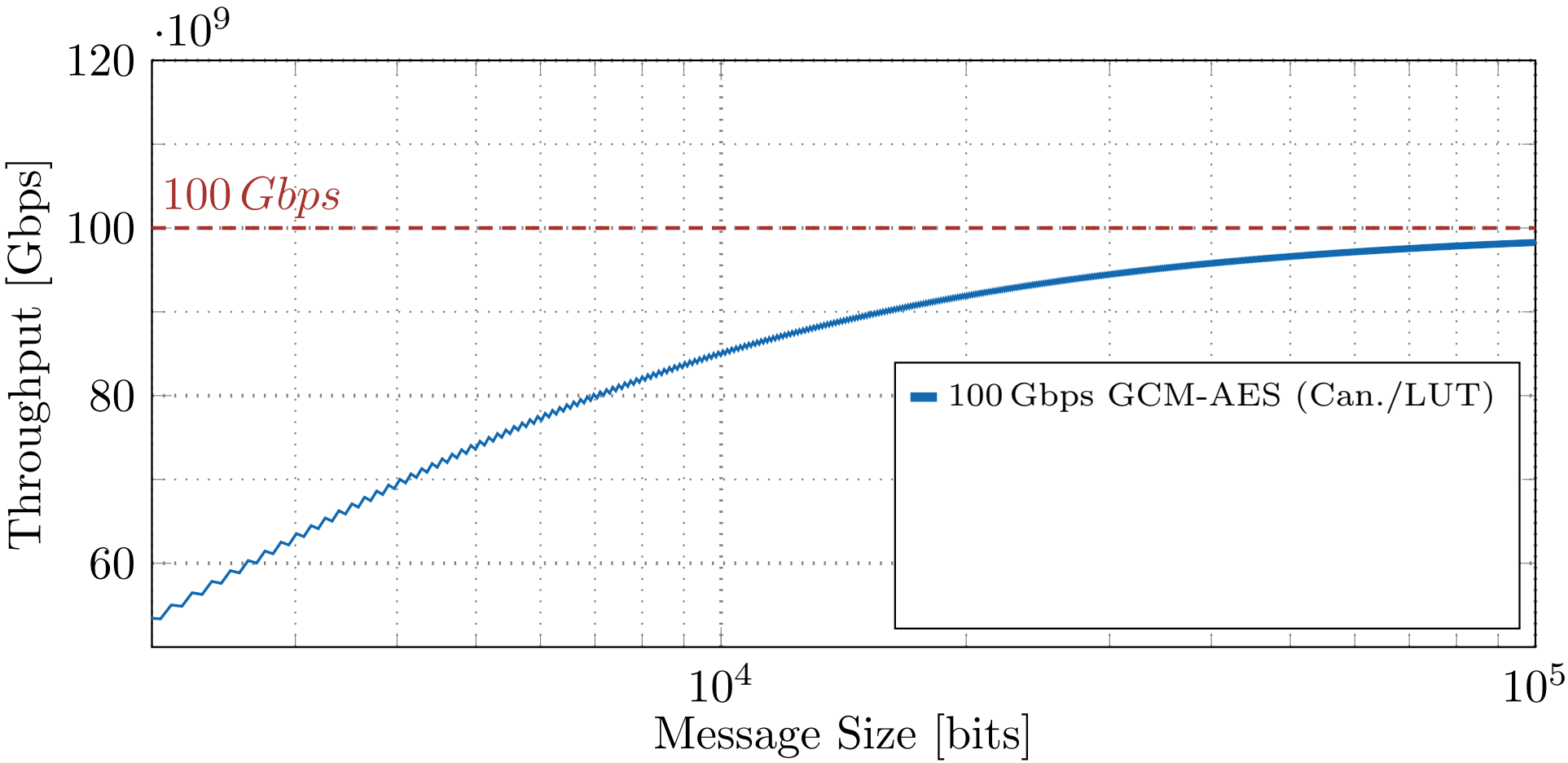
Design	100 Gbit/s Performance				Maximum TP Performance				
	Area [kGE]	f_{100} [MHz]	Efficiency [kbps/GE]	[%]	Area [kGE]	f_{max} [MHz]	TP [Gbps]	Efficiency [kbps/GE]	[%]
GCM-AES (Can)	171.2	781.3	● 584.2	**100**	177.8	813.0	104.1	▨ 585.3	**100**
GCM-AES (LUT)	191.9	781.3	■ 521.1	89	211.5	885.0	113.3	▨ 535.6	92

Data at Rest - Synthesis Results (UMC 65 nm)

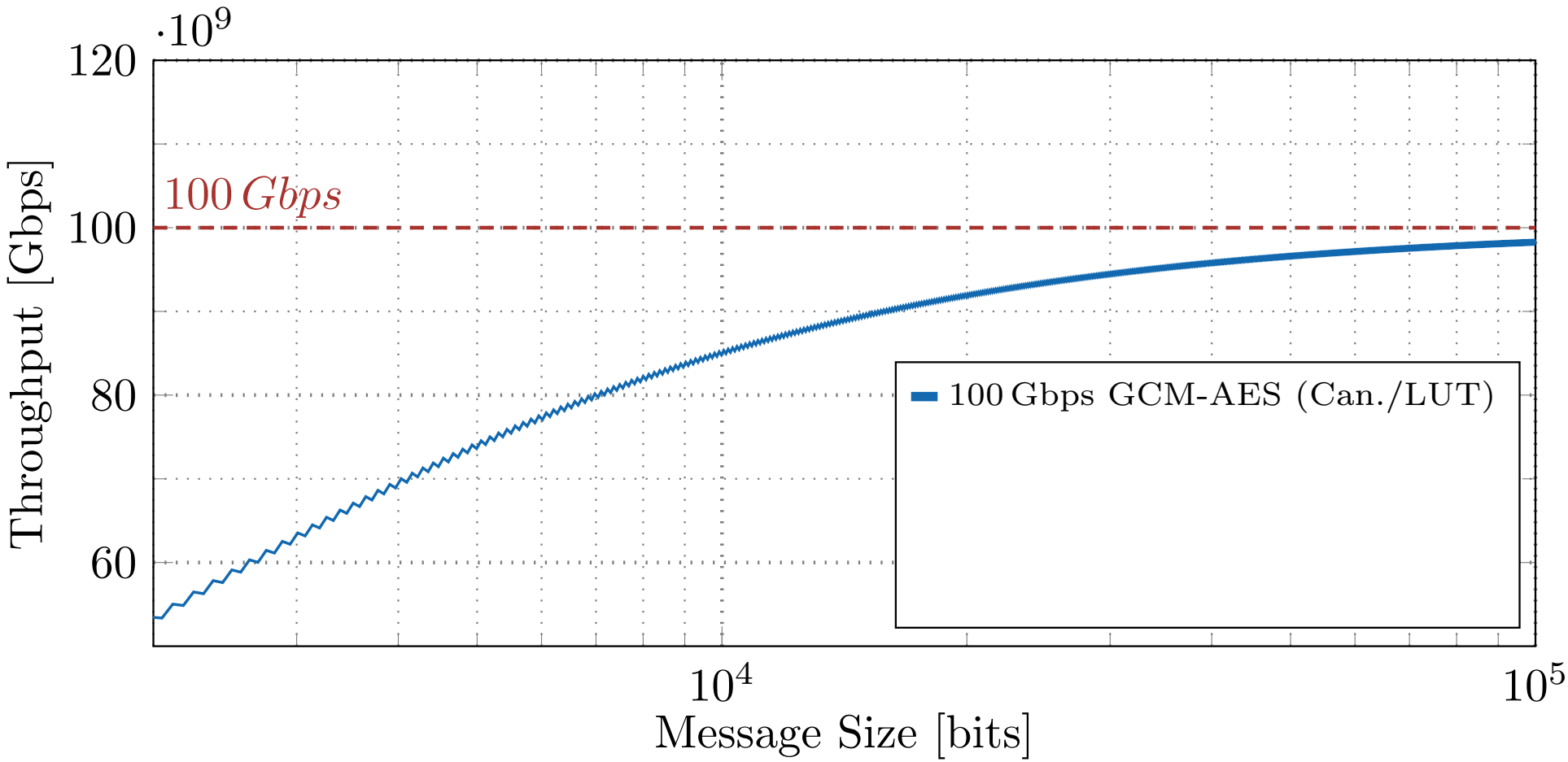


Design	100 Gbit/s Performance				Maximum TP Performance				
	Area [kGE]	f_{100} [MHz]	Efficiency [kbps/GE]	[%]	Area [kGE]	f_{max} [MHz]	TP [Gbps]	Efficiency [kbps/GE]	[%]
GCM-AES (Can)	171.2	781.3	● 584.2	**100**	177.8	813.0	104.1	◍ 585.3	**100**
GCM-AES (LUT)	191.9	781.3	■ 521.1	89	211.5	885.0	113.3	▨ 535.6	92
NORX-64-4-1 (8G)	118.3	625.0	▲ 845.3	145	189.0	775.5	124.0	△ 656.4	112

Data at Rest - Message Size Performance

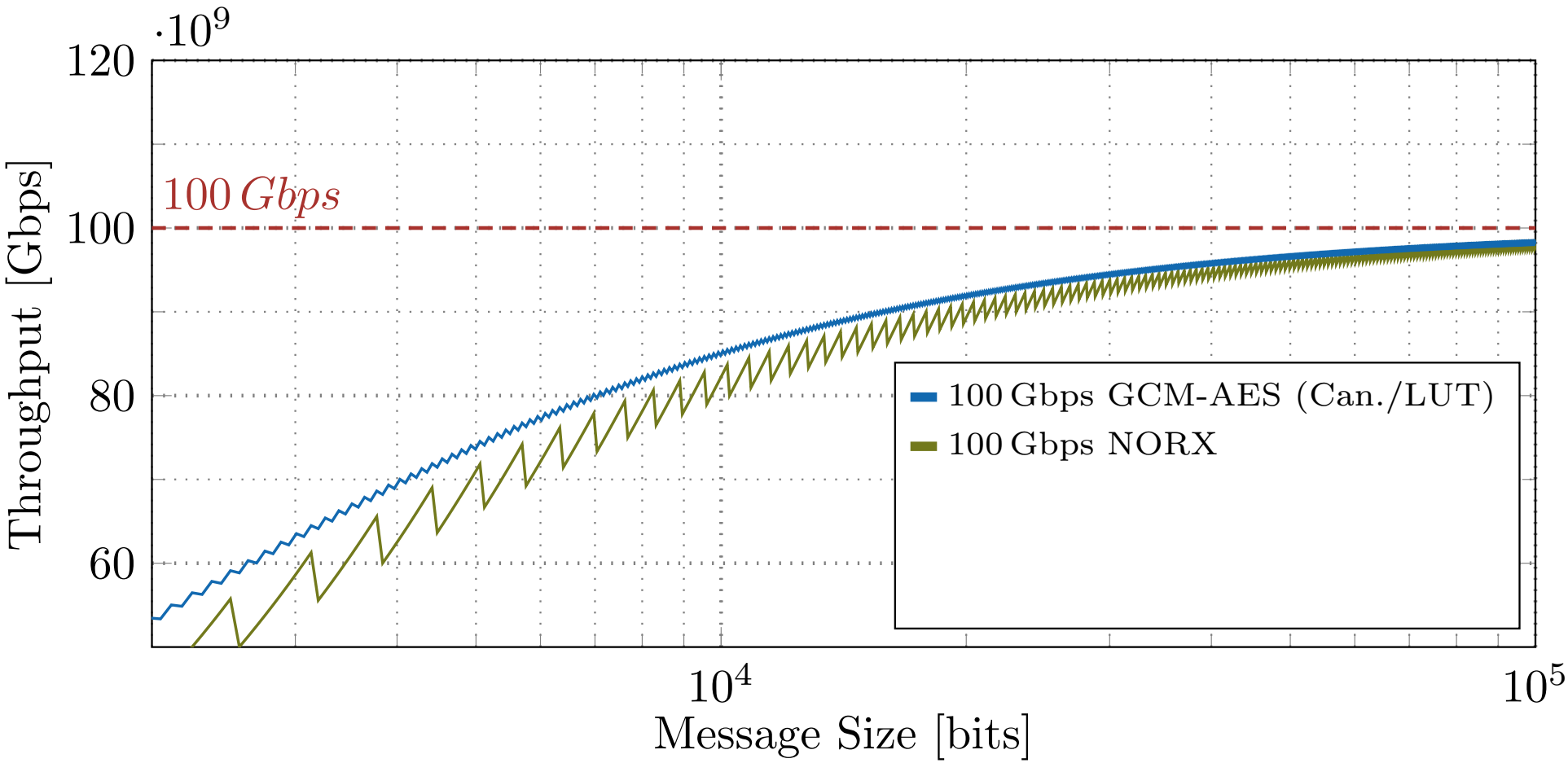


Data at Rest - Message Size Performance



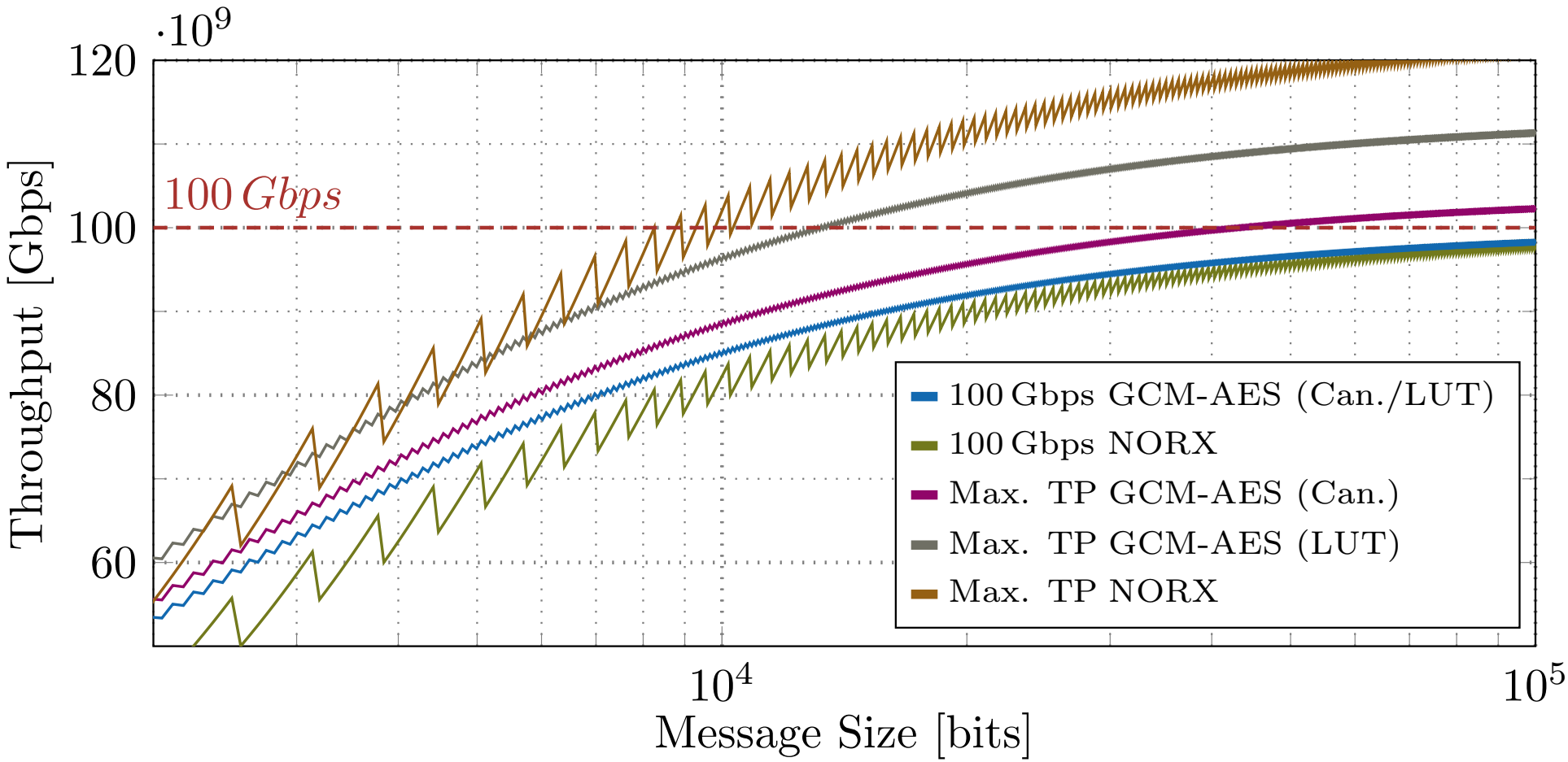
- Considering processing time required for initialization with PMNs
- Considering processing time required for tag generation

Data at Rest - Message Size Performance



- Considering processing time required for initialization with PMNs
- Considering processing time required for tag generation

Data at Rest - Message Size Performance



- Considering processing time required for initialization with PMNs
- Considering processing time required for tag generation

Project - Current State

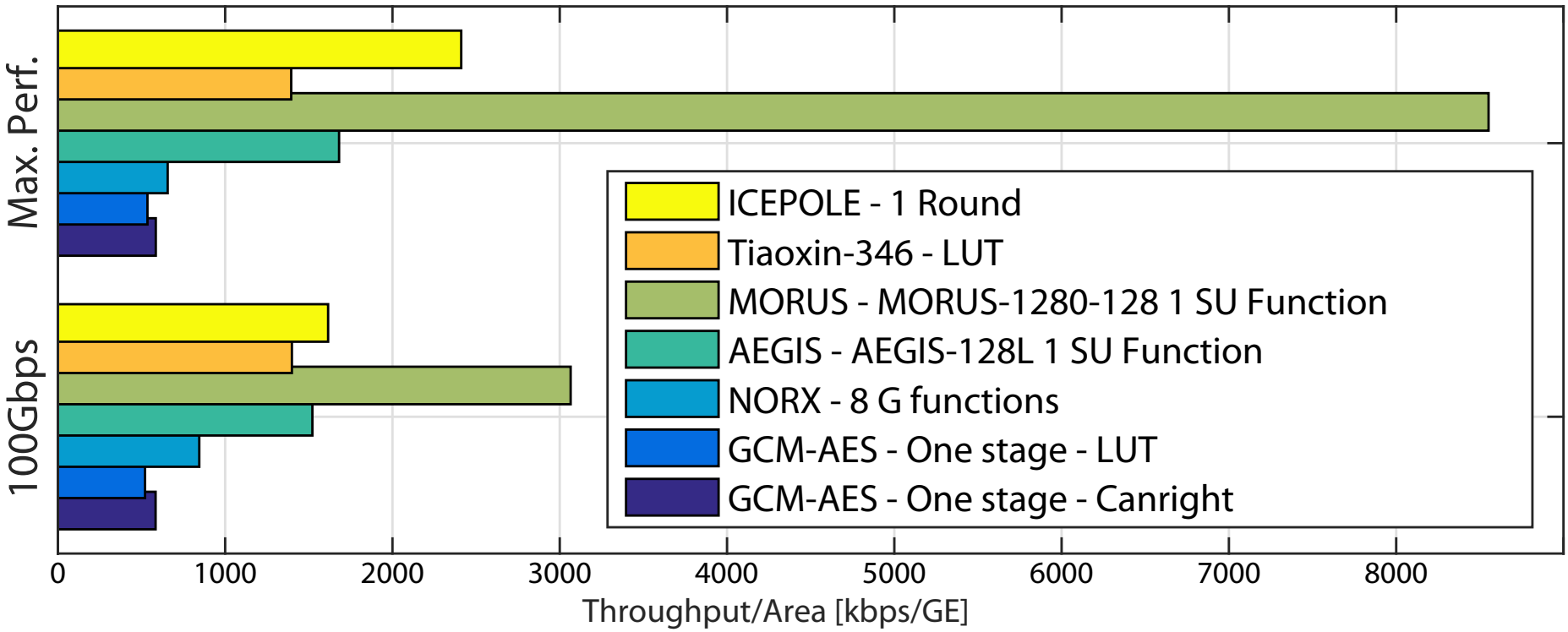
Data at Rest - Implemented 2nd-Round Candidates

- GCM-AES-128 (ref. impl.)
- AEGIS
- MORUS
- ICEPOLE
- NORX
- Tiaoxin-346

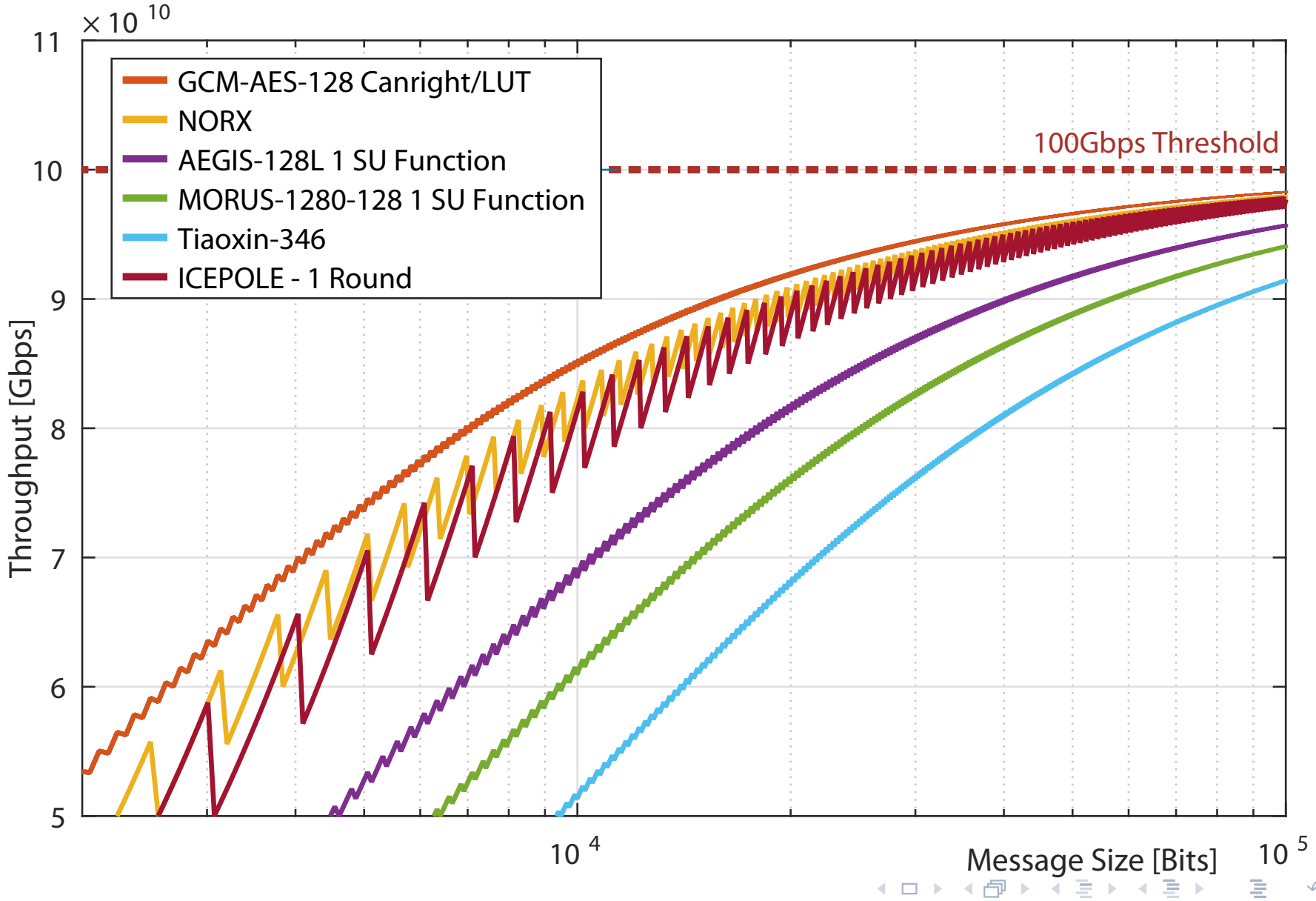
Project - Current State

Data at Rest - Implemented 2nd-Round Candidates

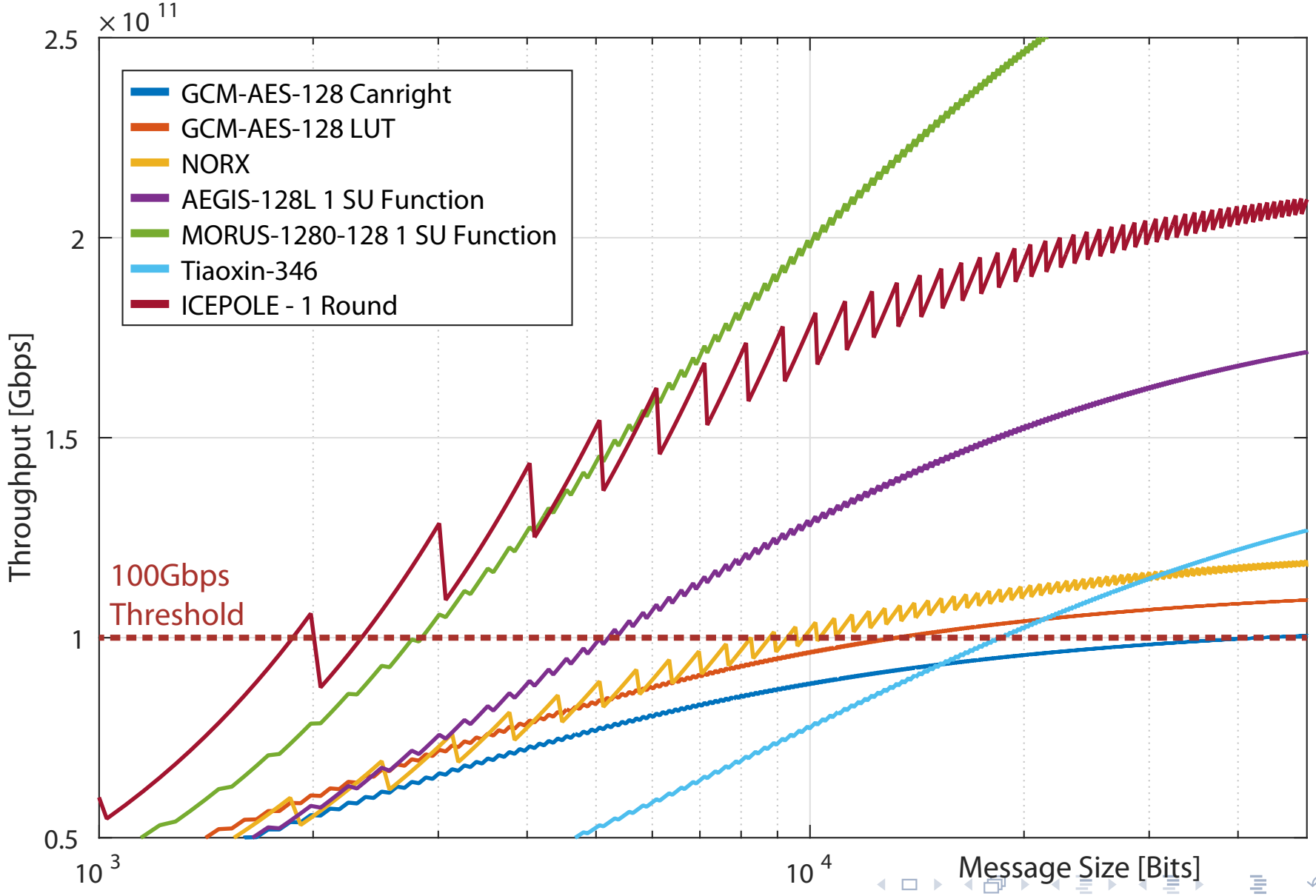
- GCM-AES-128 (ref. impl.)
- AEGIS
- MORUS
- ICEPOLE
- NORX
- Tiaoxin-346

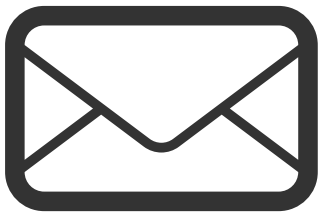


Data at Rest - 100 Gbps Performance



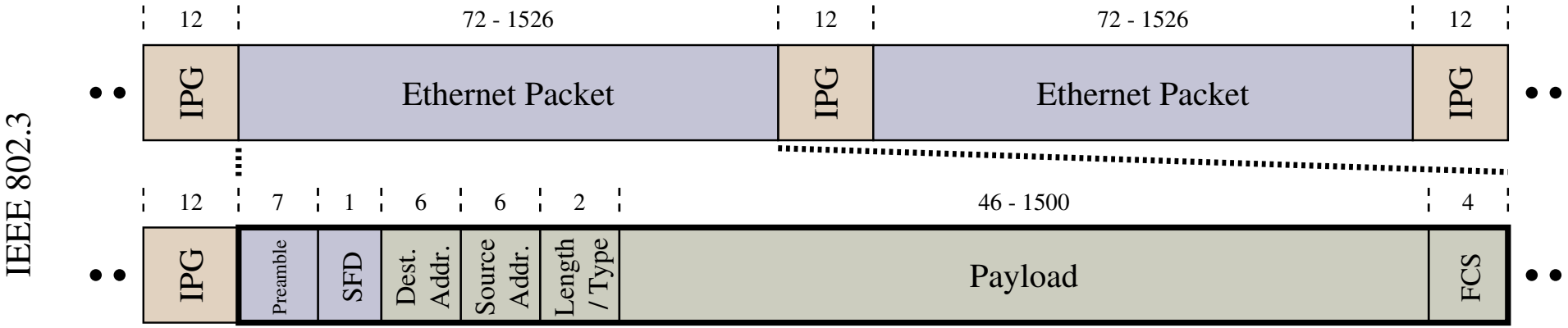
Data at Rest - Max. TP Performance



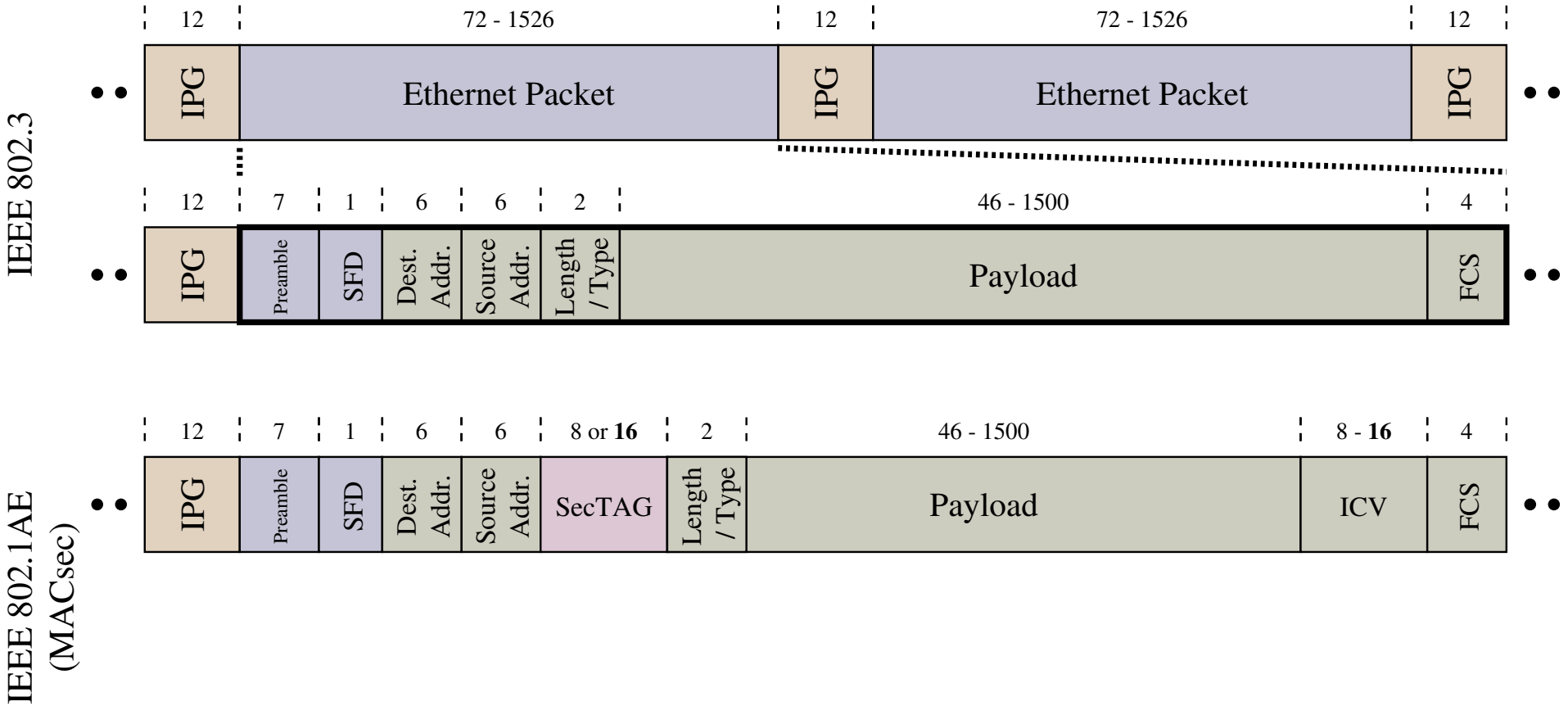


Use Case II
Data in Motion

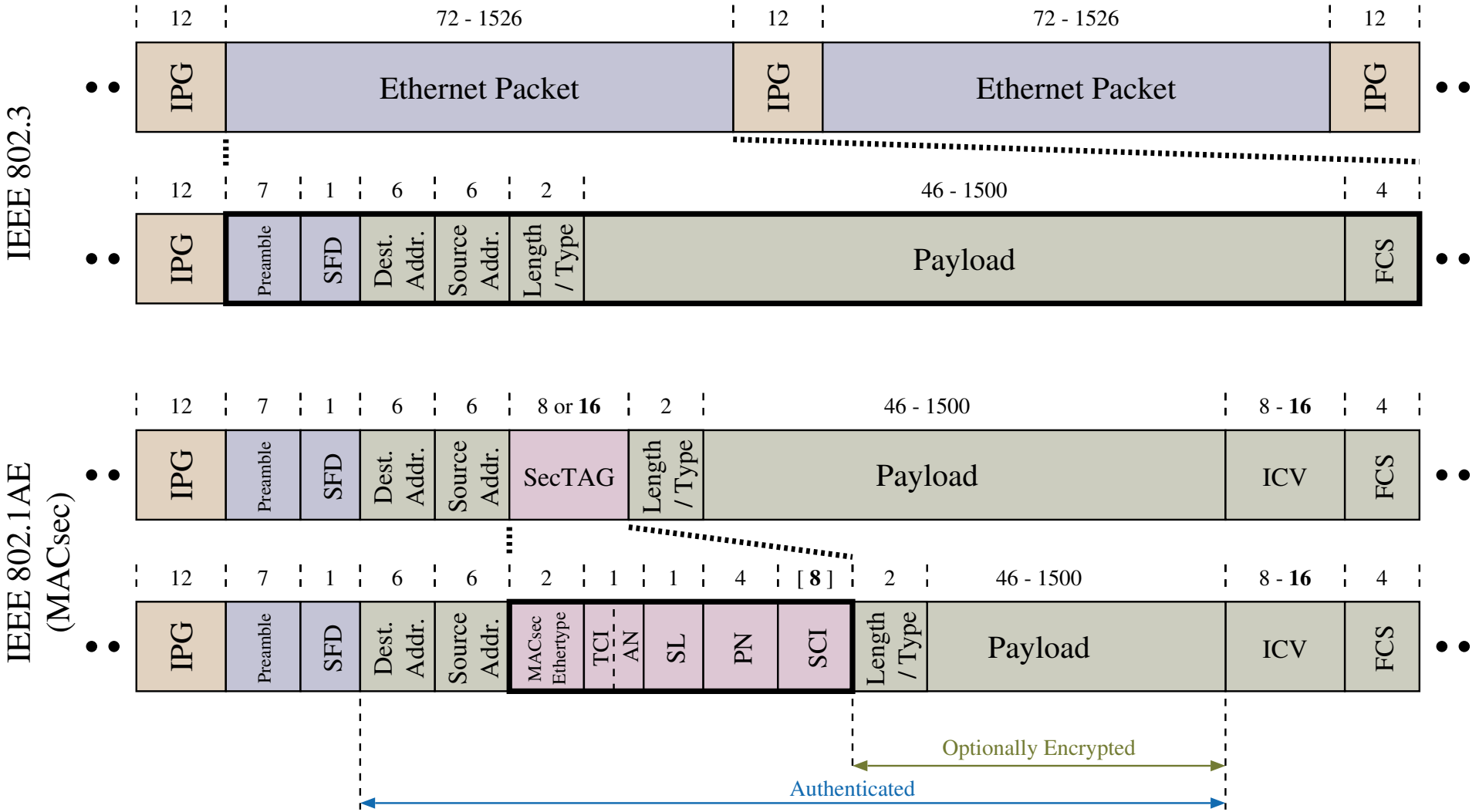
Ethernet Revisited - IEEE 802.3 and MACsec



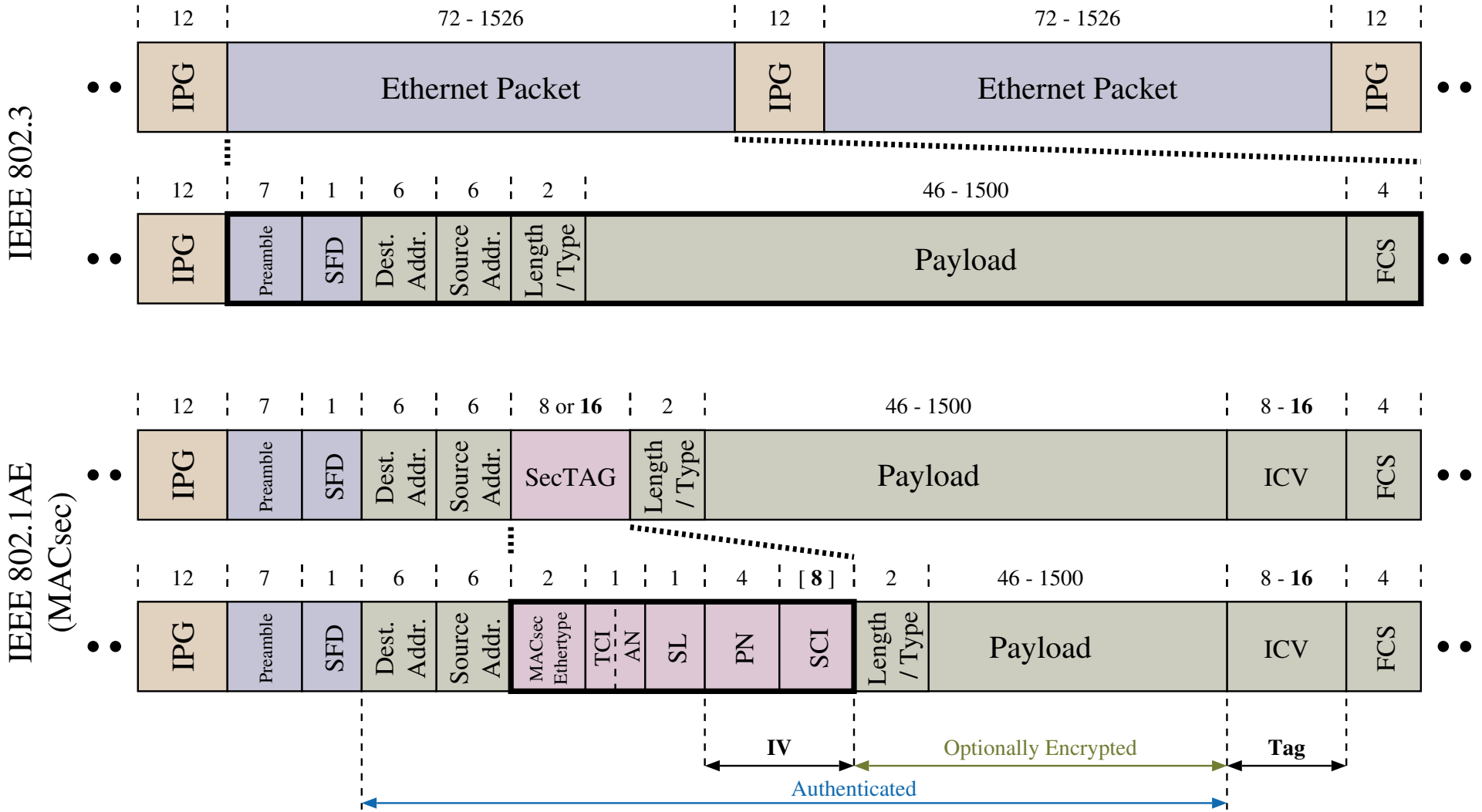
Ethernet Revisited - IEEE 802.3 and MACsec



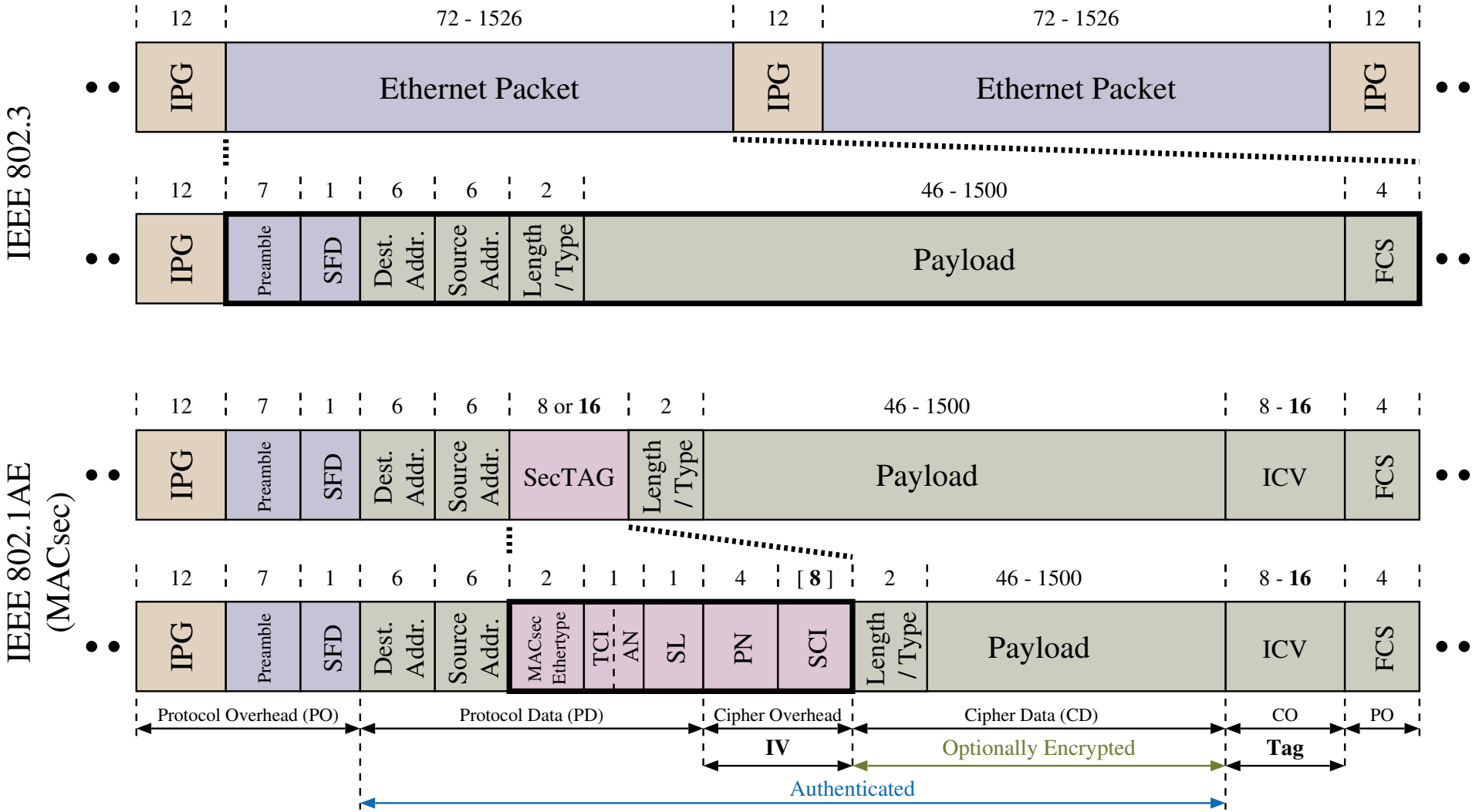
Ethernet Revisited - IEEE 802.3 and MACsec



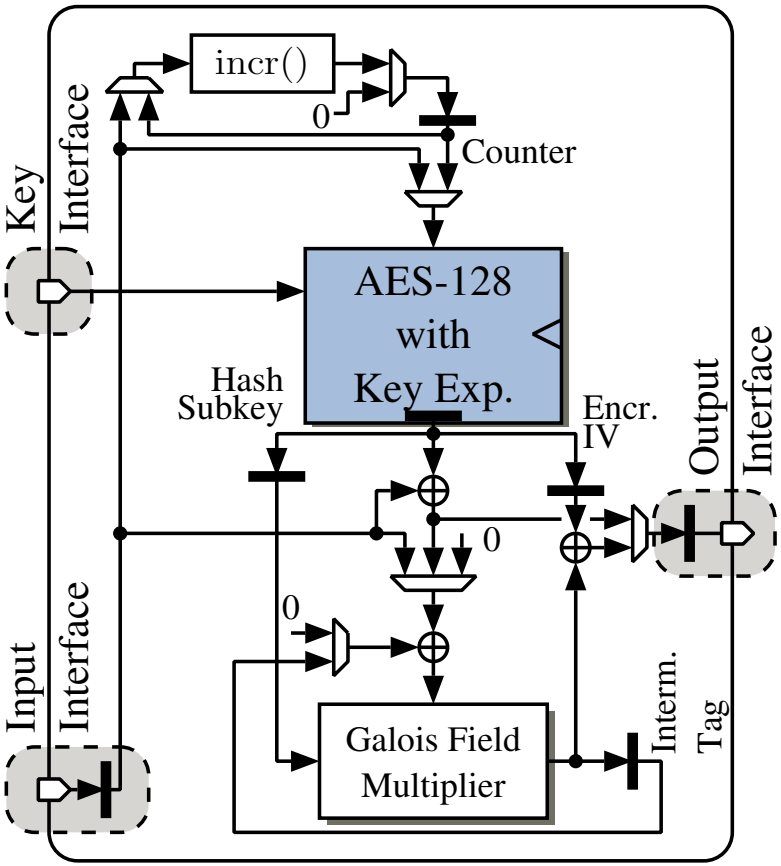
Ethernet Revisited - IEEE 802.3 and MACsec

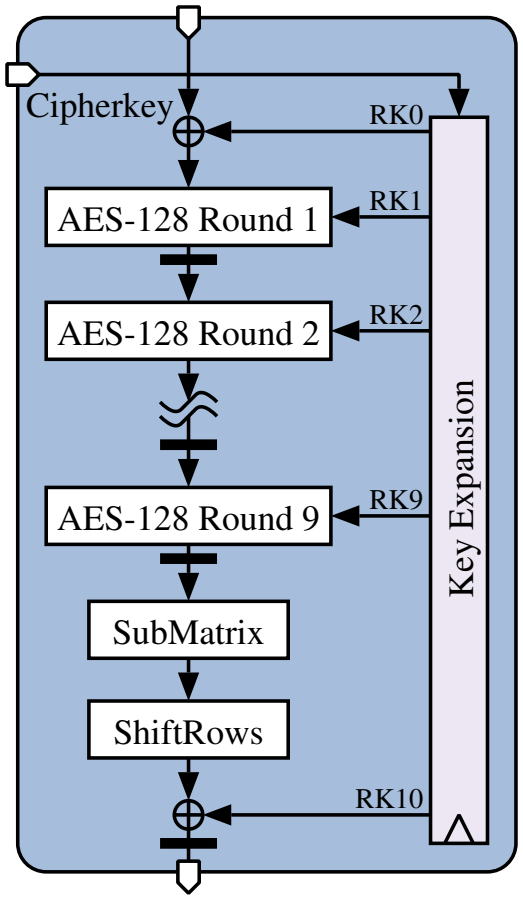


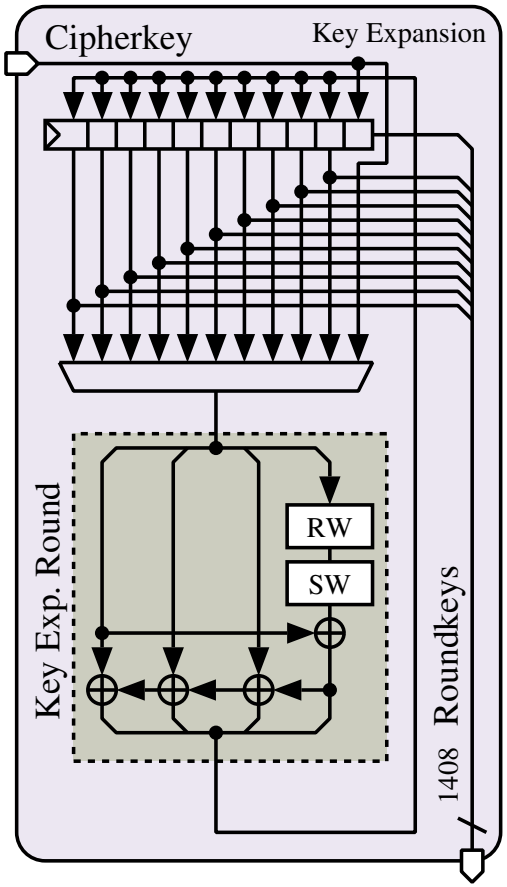
Ethernet Revisited - IEEE 802.3 and MACsec



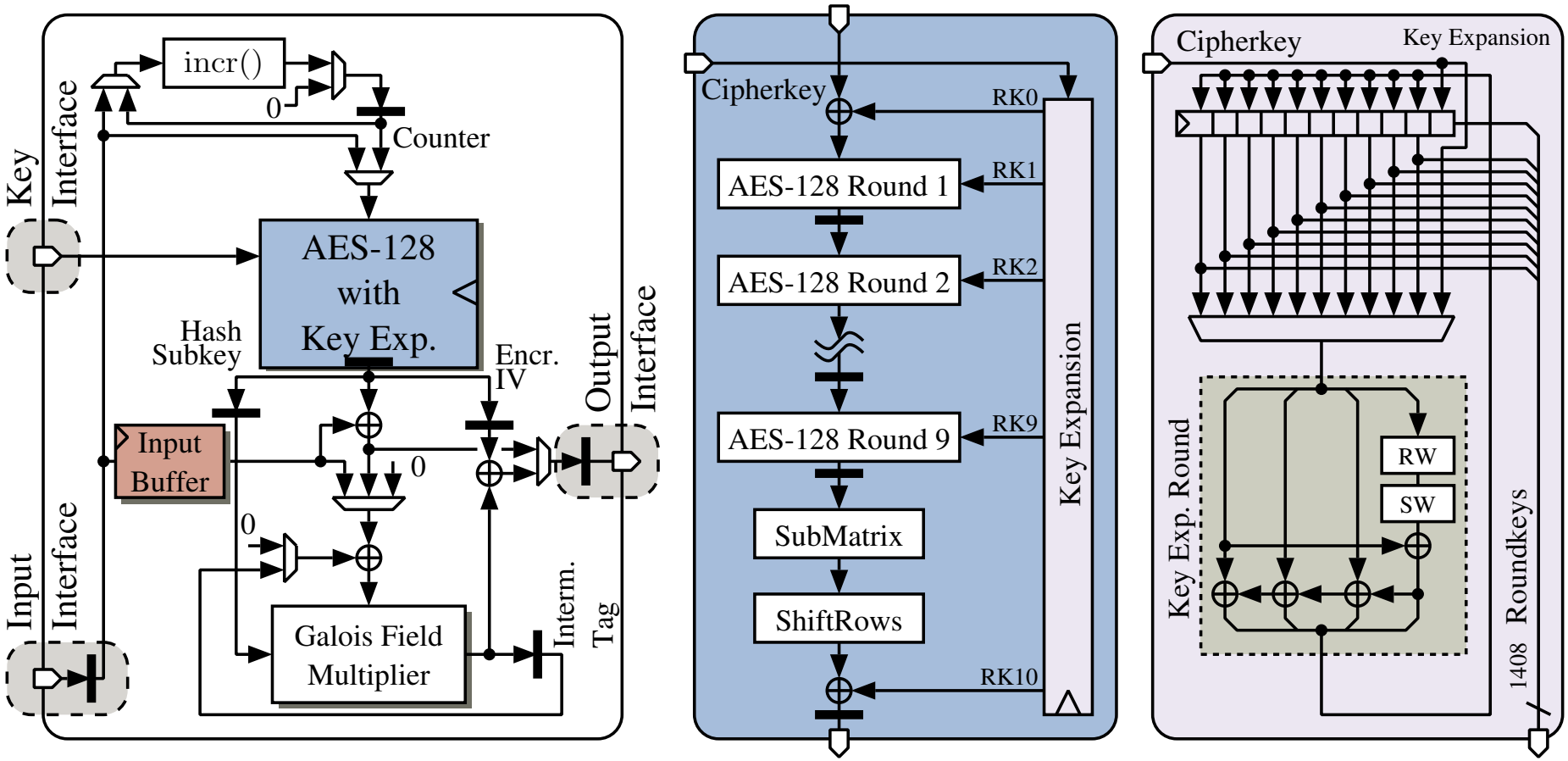
100 Gbps GCM-AES - Data in Motion Design







100 Gbps GCM-AES - Data in Motion Design



- Data input buffering according to block cipher latency
- Minor adaptions to the controlling

Project - Outlook (*WiP*)

Data in Motion - 100 Gbps Ethernet

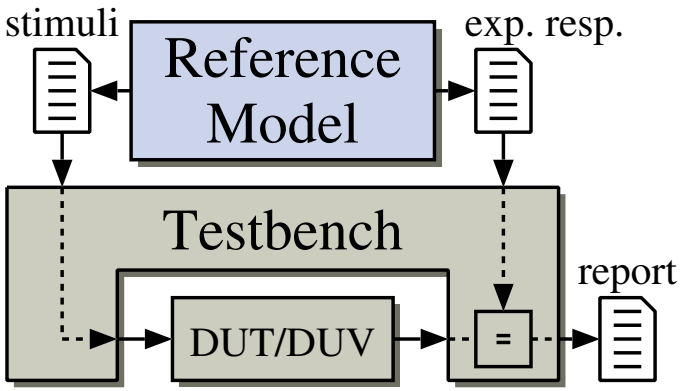
- Adapting the data at rest architectures to provide 100 Gbps Ethernet communication
- Functional verification of candidate designs

Ultimate Goal

- Add additional 2nd-round candidates
- Compare all implemented candidates against the GCM-AES reference architecture

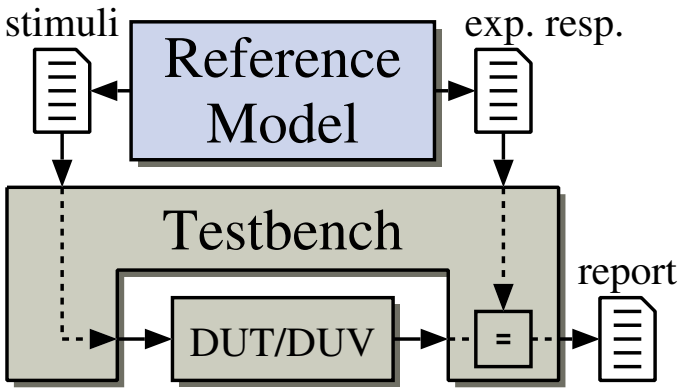
Which candidates can significantly beat GCM-AES in both the data at rest and data in motion 100 Gbps scenario?

Our HDL Verification Approach



- High-level language (C, C++,...)
- Hardware description language

Our HDL Verification Approach

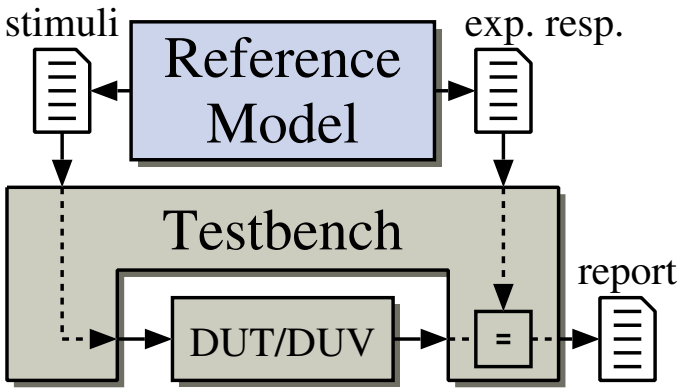


- High-level language (C, C++,...)
- Hardware description language

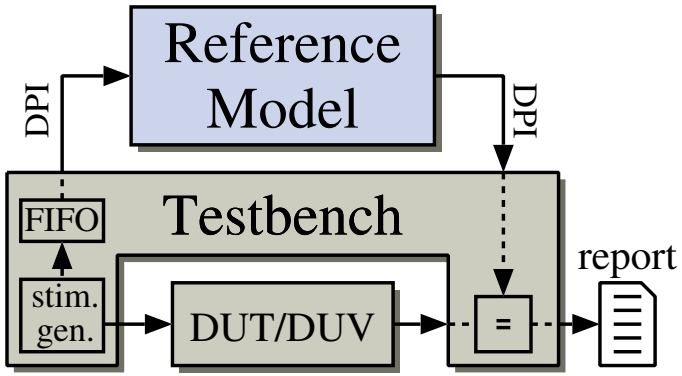
File-based testbench

- Intermediate files
- Re-implement functionality
- Minimize source of errors

Our HDL Verification Approach



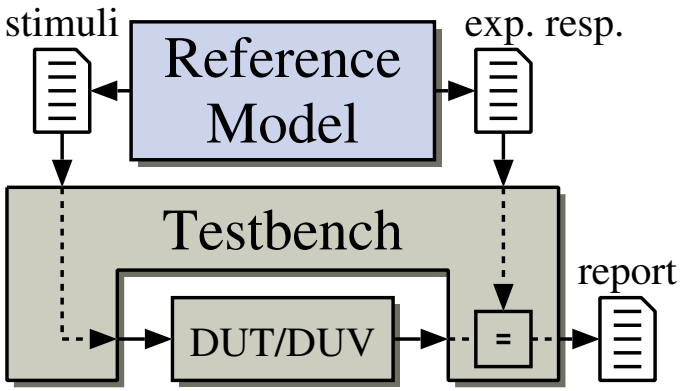
High-level language (C, C++,...)
Hardware description language



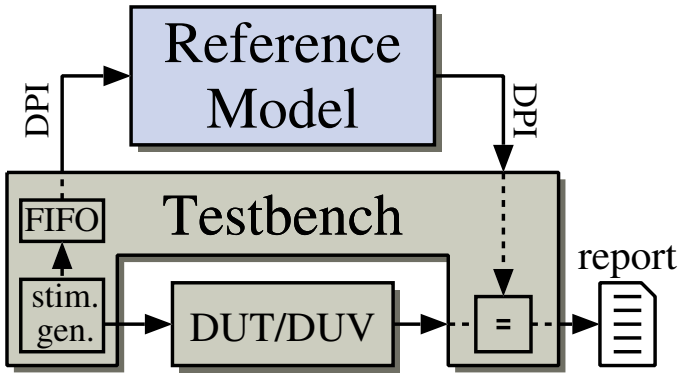
File-based testbench

- Intermediate files
- Re-implement functionality
- Minimize source of errors

Our HDL Verification Approach



High-level language (C, C++,...)
Hardware description language



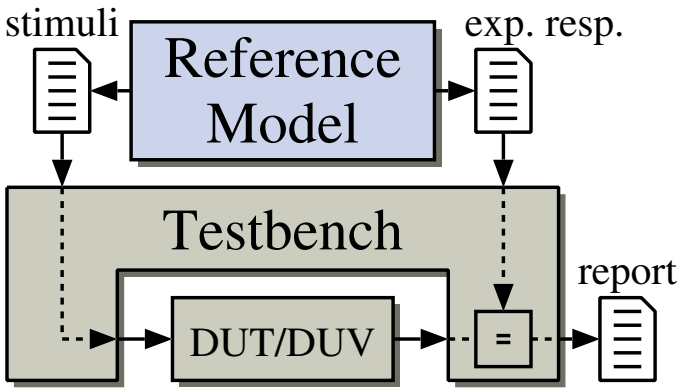
File-based testbench

- Intermediate files
- Re-implement functionality
- Minimize source of errors

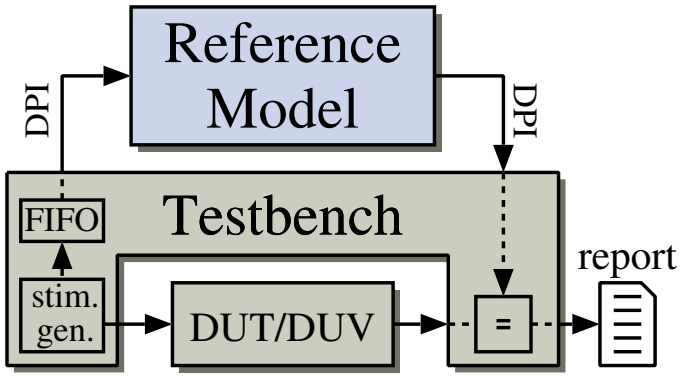
DPI-based testbench

- Direct Programming Interface (DPI)
- No intermediate files
- Reuse of C functions, C-like coding
- CAESAR: Coherent C API
- Powerful and extendable

Our HDL Verification Approach



High-level language (C, C++,...)
Hardware description language



File-based testbench

- Intermediate files
- Re-implement functionality
- Minimize source of errors

DPI-based testbench

- Direct Programming Interface (DPI)
- No intermediate files
- Reuse of C functions, C-like coding
- CAESAR: Coherent C API
- Powerful and extendable

`https://iis-git.ee.ethz.ch/mbgh/caesar-tb`

Take Home Message

1. Specify the **field of application** of hardware implementations

Take Home Message

1. Specify the **field of application** of hardware implementations

2. **TP/area metric** without a goal in mind is often misleading

Take Home Message

1. Specify the **field of application** of hardware implementations
2. **TP/area metric** without a goal in mind is often misleading
3. Initial results of **100 Gbps data at rest** and **data in motion** (Ethernet) ASIC designs

Take Home Message

1. Specify the **field of application** of hardware implementations
2. **TP/area metric** without a goal in mind is often misleading
3. Initial results of **100 Gbps data at rest** and **data in motion** (Ethernet) ASIC designs
4. Simplify your HDL **verification approach**

Questions

Contact and References

Michael Muehlberghuber
Integrated Systems Laboratory,
Gloriastrasse 35, CH-8092 Zurich
mbgh@iis.ee.ethz.ch

[1] J.-P. Aumasson et al. *NORX*. https://norx.io/.

[2] G. Bertoni et al. *Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications*. SAC 2011.