SCA Evaluation and Benchmarking of Finalists in the NIST Lightweight Cryptography Standardization Process

Kamyar Mohajerani, Luke Beckwith, Abubakr Abdulgadir, Eduadro Ferrufino, Jens-Peter Kaps, and Kris Gaj

Sixth NIST Lightweight Cryptography Workshop 2023

https://cryptography.gmu.edu

Acknowledgments

- This work is partially supported by the Department of Commerce (NIST) Grant no. 70NANB18H219

Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Authors

SCA Evaluations

Benchmarking

Jens-Peter Kaps

GMU

Kris Gaj

GMU

Bakry Abdulgadir

PQ Secure

Eddie Ferrufino

GMU

Kamyar Mohajerani

GMU

Luke Beckwith

GMU and

PQ Secure

Motivation for General Framework

- No single group is likely to have resources and expertise to develop and evaluate SCA-protected implementations of all 10 LWC finalists.
- Self-evaluation by developers may be insufficient and/or error-prone.
- Collective responsibility of the cryptographic engineering community to contribute to the evaluation process and make it as transparent and fair as possible.
- Contributions by multiple groups will make:
  - each group's workload more manageable
  - coverage of implementation platforms more complete
  - results more credible

Benefits for the Cryptographic Community

- Choosing the right algorithm can save the community countless man-hours
- Revealing and highlighting implementation and evaluation methods that rarely get fully disclosed and published
  - Most implementations open-source
  - Most evaluations transparent and reproducible
- Progress in automated generation of protected implementations
- The developed protected implementations can become benchmarks for new attacks and leakage assessment methods

General Approach

1. Call for Side-Channel Security Evaluation Labs
2. Call for Protected Hardware Implementations, targeting low-cost modern FPGAs
3. Call for Protected Software Implementations, targeting low-cost modern embedded processors

- Draft versions announced on lwc-forum in mid-December 2021
- Final versions published in mid-January 2022
- Deadlines in mid-March 2022
- Results presented to NIST on October 27, 2022 and announced in lwc-forum on November 1, 2022

Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Side-Channel Security Evaluation Labs

- We called for groups capable and willing to serve as side-channel security evaluation labs to identify their capabilities and contribute to the evaluation process
- Submitters were expected to have access to the equipment used for side-channel leakage assessment and/or attacks, experience, and human resources necessary to perform security analysis
- Labs that reported results are shown on next slides. There are
  - 2 labs that supported both hard and software implementations
  - 4 labs that supported only hardware implementations, and
  - 1 lab that supported only software implementations.
- Detailed lab specifications are posted on our webpage at https://cryptography.gmu.edu/athena/index.php?id=LWC.

Side-Channel Security Evaluation Labs for HW

Team	Evaluation Platform	Target FPGA Family	Target Boards	Leakage Assessment Methods	Attacks
IAIK, Graz University of Technology, Austria	NewAE ChipWhisperer	Artix-7	NewAE CW305	t-test	
Cryptology and Computer Security Laboratory, Shanghai Jiao Tong University, China	Riscure Inspector, NewAE ChipWhisperer, SAKURA	Kintex-7, Spartan-6,	SAKURA-G, SAKURA-X	t-test, chi-squared test, DL-LA	CPA, TA, MIA, DL-based methods
Hardware Security and Cryptographic Processor Lab,Tsinghua University, China	SAKURA	Kintex-7, Spartan-6	SAKURA-G, SAKURA-X	NICV, t-test, chi-squared test	SPA, DPA, CPA, MIA, TA, LRA, etc.

Side-Channel Security Evaluation Labs for HW

Team	Evaluation Platform	Target FPGA Family	Target Boards	Leakage Assessment Methods	Attacks
Secure-IC, France	Secure-IC Analyzr, SAKURA	Spartan-6	SAKURA-G	ISO/IEC 17825:2016	
CERG, George Mason University, USA	FOBOS3	Artix-7	NewAE CW305	t-test	
Ruhr-University Bochum, Germany	PROLEAD and other simulation-based probing security leakage-detection tools				

Side-Channel Security Evaluation Labs for SW

Team	Evaluation Platform	Target Processor	Leakage Assessment Methods	Attacks
Cryptology and Computer Security Laboratory, Shanghai Jiao Tong University, China	Riscure Inspector, NewAE ChipWhisperer	ARM Cortex-M4F, ATxmega128D4, ATmega128A	t-test, chi-squared test, DL-LA	CPA, TA, MIA, DL-based methods
Hardware Security and Cryptographic Processor Lab,Tsinghua University, China		ARM Cortex-M4F, ARM Cortex-M3	NICV, t-test, chi-squared test	SPA, DPA, CPA, MIA, TA, LRA, etc.
CESCA Lab, Radboud University, Netherlands	Riscure Inspector, NewAE ChipWhisperer	ARM Cortex-M4F, ATxmega128D4	t-test, chi-squared test, DL-LA	SPA, DPA, CPA, TA; DEMA; DFA, FI attacks

Overview

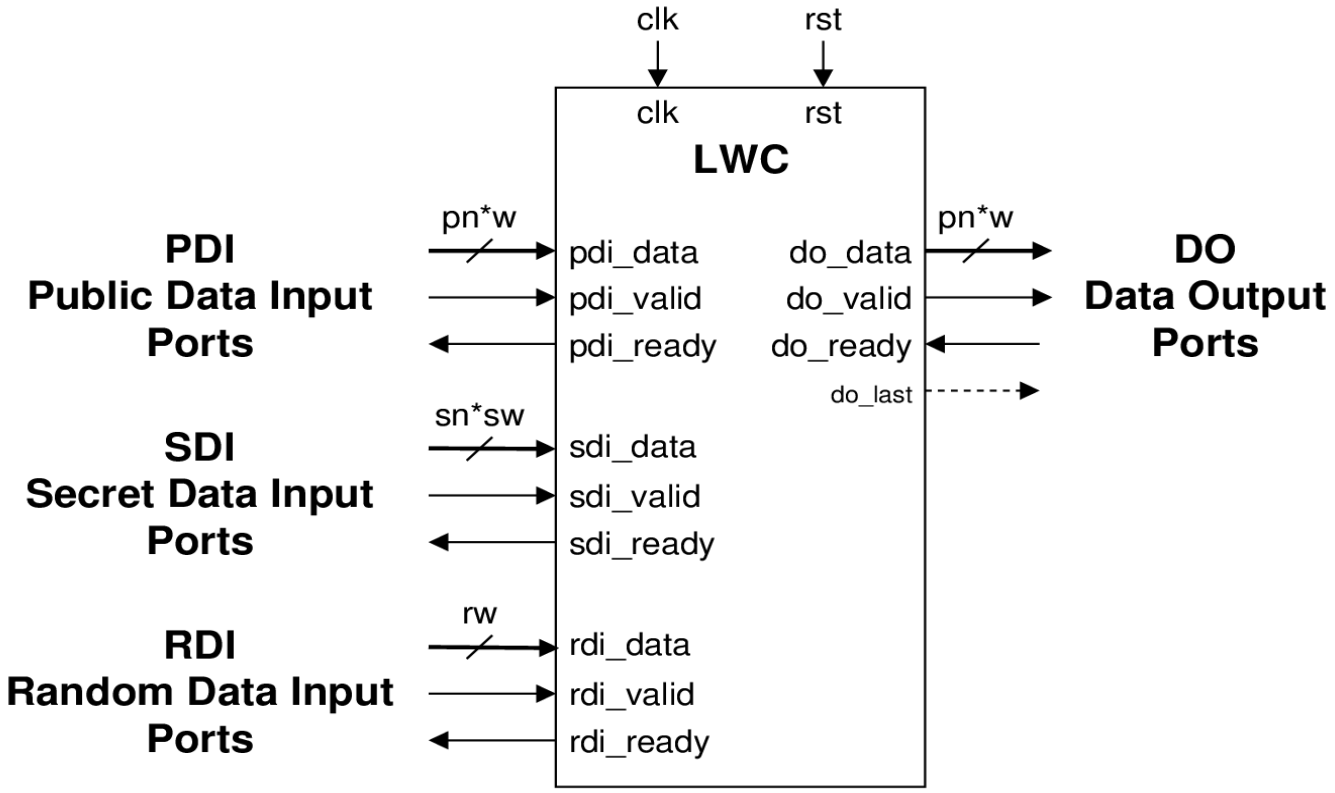
- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Protected Hardware Implementations

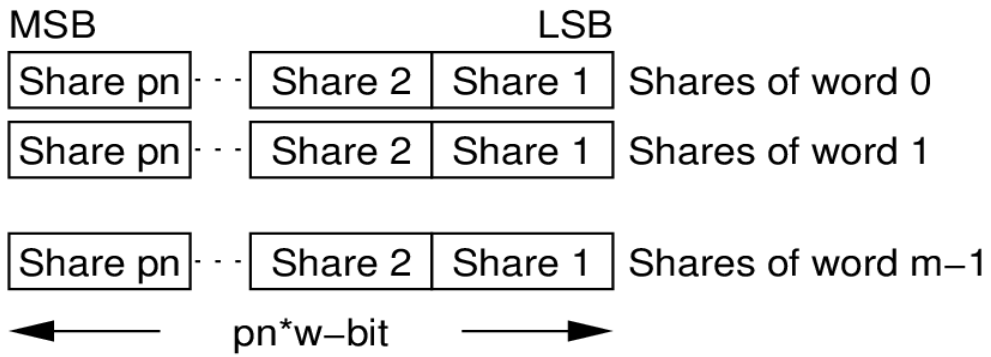
- Submitted designs should demonstrate strong resistance against side-channel attacks when implemented on low-cost modern FPGAs
- A potential for porting the designs to ASIC (Application-Specific Integrated Circuit) technology
- All submitted implementations evaluated by one or more Side-Channel Security Evaluation Labs

Protected HW Implementations Submission Requirements

- Compliant with the Extended LWC Hardware API, v1.1 (January 2022) or later
- Interface



- Pre-shared Data



Features Supporting Leakage Assessment Methods

Feature	Proposed Approach
Division of inputs into shares	outside of LWC
Combining shares into outputs	outside of LWC
Passing leakage detection test dependent on	side-channel countermeasures
Random Data Input ports	yes
Overhead of DRBG in terms of area, power, energy	excluded
Sharing DRBG with other units	easy
Changing the source of random bits	easy

Protected Hardware Implementations – Variants

- Variants = Different versions of the design that correspond to
  - different algorithms of the same family
  - different sizes of keys, nonces, tags, etc.
  - different parameters of the interface, such as w and sw
  - different hardware architectures (e.g., basic iterative, unrolled, folded, pipelined, etc.),
  - different protection methods against side-channel attacks,
  - different orders of protection against side-channel attacks

Protected HW Implementations Available for Evaluation

LWC Candidates	Team	No. of variants	Protection Method	Protection Order	Availability	License
ISAP	IAIK, Graz University of Technology, Austria	6	mode-level DPA resistance	N/A	GitHub	GPL-3.0
Ascon, Elephant, GIFT-COFB, ISAP, PHOTON-Beetle, Romulus, SPARKLE, TinyJAMBU, Xoodyak	Ruhr-University Bochum, Germany	Ascon, Xoodyak: 6 Others: 3	HPC2	1, 2, 3	GitHub	GPL-3.0
TinyJAMBU, Xoodyak,	CERG, George Mason University, USA	1	DOM	1	GitHub (TinyJAMBU) GitHub (Xoodyak)	GPL-3.0
Ascon	IAIK, Graz University of Technology, Austria	1	DOM	1, 2	Per request (Unprotected)	GPL-3.0
Xoodyak	Hardware Security and Cryptographic Processor Lab, Tsinghua University, Beijing, China	2	DOM, TI	1	GitHub	GPL-3.0

Missing Protected Hardware Implementations

- Missing semi-automatically generated implementations:
  - Grain128-AEAD
- Missing manually-designed protected hardware implementations:
  - Elephant
  - GIFT-COFB
  - Grain128-AEAD
  - Photon-Beetle
  - Romulus
  - Sparkle

Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Protected Software Implementations

- Submitted designs should demonstrate strong resistance against side-channel attacks when executed on low-cost modern embedded processors
- Compliant with the NIST API defined in Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process, published in August 2018
- The code can contain assembly language instructions specific to a given Instruction Set Architecture (ISA)
- No dependence on any external headers or libraries, including cryptographic libraries (e.g., OpenSSL), outside of the C99 standard
- All submitted implementations evaluated by one or more Side-Channel Security Evaluation Labs

Protected SW Implementations Available for Evaluation

LWC Candidates	Team	No. of variants	Protection Method	Protection Order	Availability	License
ISAP	ISAP Team	5	mode-level DPA resistance	N/A	GitHub	CCO-1.0
Ascon	Ascon Team	6	Masking, share rotation, mode-level security	2	GitHub	CCO-1.0
GIFT-COFB	Alexandre Adomnicai	1	Boolean masking	1	GitHub	CCO-1.0
Romulus	Alexandre Adomnicai	3	Boolean masking	1	GitHub	CCO-1.0
Xoodyak	HW Security and Crypto-graphic Processor Lab, Tsinghua University, Beijing, China	1	ISW Scheme	1	GitHub	CCO-1.0

CCO-1.0: Creative Commons version 1

Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Side-Channel Evaluations of Protected HW Implementations

Implementation	Lab	Target	Oscilloscope	Freq [MHz]	Sampl. Freq. [MS/s]	Reso-lution [bits]	Meas. Type	Test
Ascon_Bochum_d1	CERG	CW305	FOBOS3 ADC	16	80	10	power	TVLA
	IAIK	CW305	PicoScope 6404C	1	22	8	power	TVLA
	CCSL	SAKURA-X	LeCroy 610Zi		1000	8	EM	TVLA
		SAKURA-X	LeCroy 610Zi		1000	8	EM	χ^2 -test
		SAKURA-X	LeCroy 610Zi		1000	8	EM	CPA
Ascon_v1_Graz_d1	HSCP	SAKURA-G	WaveRunner 8404M	4	100	8	power	TVLA
Elephant_Bochum_d1	CERG	CW305	FOBOS3 ADC	10	50	10	power	TVLA
	IAIK	CW305	PicoScope 6404C	1	22	8	power	TVLA
GIFT_COFB_Bochum_d1	IAIK	CW305	PicoScope 6404C	1	22	8	power	TVLA
	CCSL	SASEBO-GIII			500	8	EM	TVLA
		SASEBO-GIII			500	8	EM	χ^2 -test
		SASEBO-GIII			500	8	EM	χ^2 -test
		SASEBO-GIII			500	8	EM	CPA

Side-Channel Evaluations of Protected HW Implementations

Implementation	Lab	Target	Oscilloscope	Freq [MHz]	Sampl. Freq. [MS/s]	Reso-lution [bits]	Meas. Type	Test
ISAP_Bochum_d1	CCSL	Kintex 7	LeCroy 610Zi				EM	CPA
		Kintex 7	LeCroy 610Zi				EM	TVLA
		Kintex 7	LeCroy 610Zi				EM	χ^2 -test
ISAP_Graz	CCSL	Kintex 7	LeCroy 610Zi				EM	CPA
Photon Beetle_Bochum_d1	CERG	CW305	FOBOS3 ADC	16	80	10	power	TVLA
Romulus_Bochum_d1	IAIK	CW305	PicoScope 6404C	1	22	8	power	TVLA
	CCSL	SASEBO-GIII			500	8	EM	TVLA
		SASEBO-GIII			500	8	EM	TVLA
		SASEBO-GIII			500	8	EM	χ^2 -test
		SASEBO-GIII			500	8	EM	χ^2 -test
		SASEBO-GIII			500	8	EM	CPA
		SASEBO-GIII			500	8	EM	TA

Side-Channel Evaluations of Protected HW Implementations

Implementation	Lab	Target	Oscilloscope	Freq [MHz]	Sampl. Freq. [MS/s]	Reso-lution [bits]	Meas. Type	Test
TinyJAMBU_Bochum_d1	CERG	CW305	FOBOS3 ADC	10	50	10	power	TVLA
TinyJAMBU_GMU_d1	HSCP	SAKURA-G	WaveRunner 8404M	4	100	8	power	TVLA
Xoodyak_Bochum_d1	IAIK	CW305	PicoScope 6404C	1	22	8	power	TVLA
Xoodyak_GMU_d1	Secure-IC	Arty A7	Tektronix MSO64	100	6250	12	EM	TVLA
Xoodyak_Bochum_d1	CERG	CW305	FOBOS3 ADC	10	50	10	power	TVLA

Results of Side-Channel Evaluations of Protected HW Implementations

Implementation	Lab	Test	Num. of Traces [x10^6]	Thresh. Exc.	Notes
Ascon_Bochum_d1	CERG	TVLA	10	Y(1.5M)	6 out of 1000+ samples exceed the threshold
	IAIK	TVLA	10	N	
	CCSL	TVLA	1	N	
		χ^2 -test	1	N	
		CPA	11	-	No bytes revealed
Ascon_v1_Graz_d1	HSCP	TVLA	7	N	
Elephant_Bochum_d1	CERG	TVLA	10	Y(2.7M)	3 out of 12,000+ samples exceed threshold
	IAIK	TVLA	10	N	

Results of Side-Channel Evaluations of Protected HW Implementations

Implementation	Lab	Test	Num. of Traces [x10^6]	Thresh. Exc.	Notes
GIFT_COFB_Bochum_d1	IAIK	TVLA	10	N	
	CCSL	TVLA	1	N	Classification based on a nonce bit. A similar test was also based on a bit in an intermediate value
		χ^2 -test	1	Y	Classification based on a nonce bit: threshold exceeded
		χ^2 -test	1	N	Classification based on a bit in an intermediate value
		CPA	1	-	Key not revealed
ISAP_Bochum_d1	CCSL	CPA		-	Key not revealed
		TVLA		Y	Some samples exceeding the threshold observed
		χ^2 -test		Y	Some samples exceeding the threshold observed

Results of Side-Channel Evaluations of Protected HW Implementations

Implementation	Lab	Test	Num. of Traces [x10^6]	Thresh. Exc.	Notes
ISAP_Graz	CCSL	CPA		-	Key not revealed
Photon Beetle_Bochum_d1	CERG	TVLA	10	N	t-value crossed threshold briefly before returning below threshold
Romulus_Bochum_d1	IAIK	TVLA	10	N	
	CCSL	TVLA	10	Y	Case A: Few samples exceed the threshold at 1M traces. Classification based on a nonce bit
		TVLA	1	N	Case B: No samples exceed the threshold at 1M traces. Classification based on an intermediate bit
		χ^2 -test	1	Y	Case A: Few samples exceed the threshold at 1M traces. Classification based on a nonce bit
		χ^2 -test	1	N	Case B: No samples exceed the threshold at 1M traces. Classification based on an intermediate bit
		CPA	1	-	Key not revealed
		TA	1	-	Key not revealed

Results of Side-Channel Evaluations of Protected HW Implementations

Implementation	Lab	Test	Num. of Traces [x10^6]	Thresh. Exc.	Notes
TinyJAMBU_Bochum_d1	CERG	TVLA	10	N	One sample exceeded the threshold so the test repeated again. Another sample exceeded the threshold but at another location indicating a false positive
TinyJAMBU_GMU_d1	HSCP	TVLA	10	N	
Xoodyak_Bochum_d1	IAIK	TVLA	10	N	
Xoodyak_GMU_d1	Secure-IC	TVLA	0.1	N	Classification based on an input plaintext bit
Xoodyak_Bochum_d1	CERG	TVLA	10	Y(3.2M)	10 out of 900 samples exceed the threshold

Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Side-Channel Evaluations of Protected SW Implementations

Implementation	Lab	Target	Oscilloscope	Freq. [MHz]	Sampl. Freq. [MS/s]	Reso-lution [bits]	Meas. Type	Test
Ascon_Graz_d1	CESCA	STM32F407	Pico 3206D	168	100-1000	8	EM	CPA
Ascon_Graz_d2	CCSL	STM32F303	Pico 3206D		62.5	16	EM	TVLA
		STM32F303	Pico 3206D		62.5	16	EM	χ^2 -test
		STM32F303	Pico 3206D		62.5	16	EM	CPA
GIFT_COFB_Adominicai	CCSL	STM32F303			125	16	EM	TVLA
		STM32F303			125	16	EM	χ^2 -test
		STM32F303			125	16	EM	CPA
	HSCP	STM32F303		8	25	8	power	TVLA
ISAP_ISAP_Team	CESCA	STM32F407		100	100-1000	8	power	TVLA
	CCSL	STM32F303	LeCroy 610Zi				EM	CPA

Side-Channel Evaluations of Protected SW Implementations

Implementation	Lab	Target	Oscilloscope	Freq. [MHz]	Sampl. Freq. [MS/s]	Reso-lution [bits]	Meas. Type	Test
Romulus_Adominicai	HSCP	STM32F303	WaveRunner 8404M	8	25	8	power	TVLA
	CCSL	STM32F303			125	16	EM	TVLA
		STM32F303			125	16	EM	TVLA
		STM32F303			125	16	EM	DL-LA
		STM32F303			125	16	EM	DL-LA
		STM32F303			125	16	EM	CPA
		STM32F303			125	16	EM	TA

Results of Side-Channel Evaluations of Protected SW Implementations

Implementation	Lab	Test	Num. of Traces [x10^6]	Thresh. Exc.	Notes
Ascon_Graz_d1	CESCA	CPA	15	-	Second order CPA. No bytes revealed.
Ascon_Graz_d2	CCSL	TVLA	0.06	N	
		χ^2 -test	0.06	N	
		CPA	0.06	-	Key not revealed
GIFT_COFB_Adominicai	CCSL	TVLA	0.02	N	Classification based on a nonce bit. Another test was done with classification based on an intermediate bit.
		χ^2 -test	0.02	N	
		CPA	0.02	-	Key not revealed
	HSCP	TVLA	0.1	Y	Threshold exceeded. Report mentions possible causes
ISAP_ISAP_Team	CESCA	TVLA	0.1	N	Fixed key vs random key test
	CCSL	CPA		-	Key not revealed

Results of Side-Channel Evaluations of Protected SW Implementations

Implementation	Lab	Test	Num. of Traces [x10^6]	Thresh. Exc.	Notes
Romulus_Adominicai	HSCP	TVLA	0.1	Y	Threshold exceeded. Report mentions possible causes
	CCSL	TVLA	1	N	Case A: No sample exceeded the threshold for 1M traces. Classification based on a nonce bit.
		TVLA	1	N	Case B: No sample exceeded the threshold for 1M traces. Classification based on an intermediate bit.
		DL-LA		N	Case A: No sample exceeded the threshold for 1M traces. Classification based on a nonce bit.
		DL-LA		-	Case B: No sample exceeded the threshold for 1M traces. Classification based on an intermediate bit.
		CPA		-	Key not revealed
		TA		-	Key not revealed

Conclusions of SCA Evaluations

- Protected hardware implementations of 9 out of 10 finalists
- Most of them generated automatically
- Most of them pass basic leakage assessment tests (any required corrections are not likely to affect results of benchmarking)

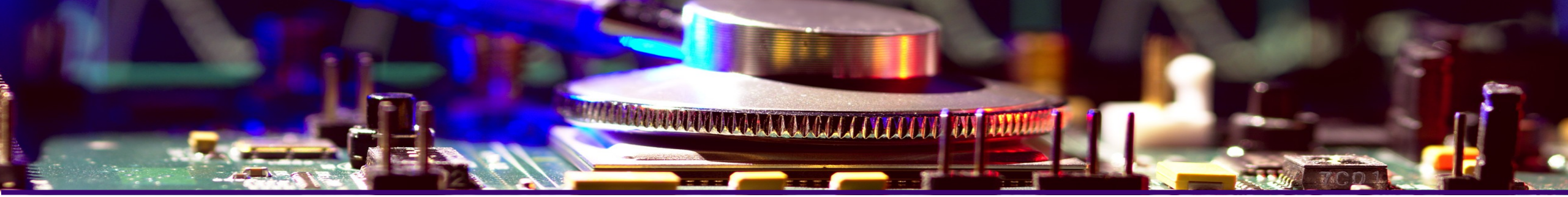
- Protected software implementations of 5 out of 10 finalists
- Two implementations fail a basic leakage assessment test
- One of the remaining ones uses a mode-level protection difficult to verify experimentally

Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

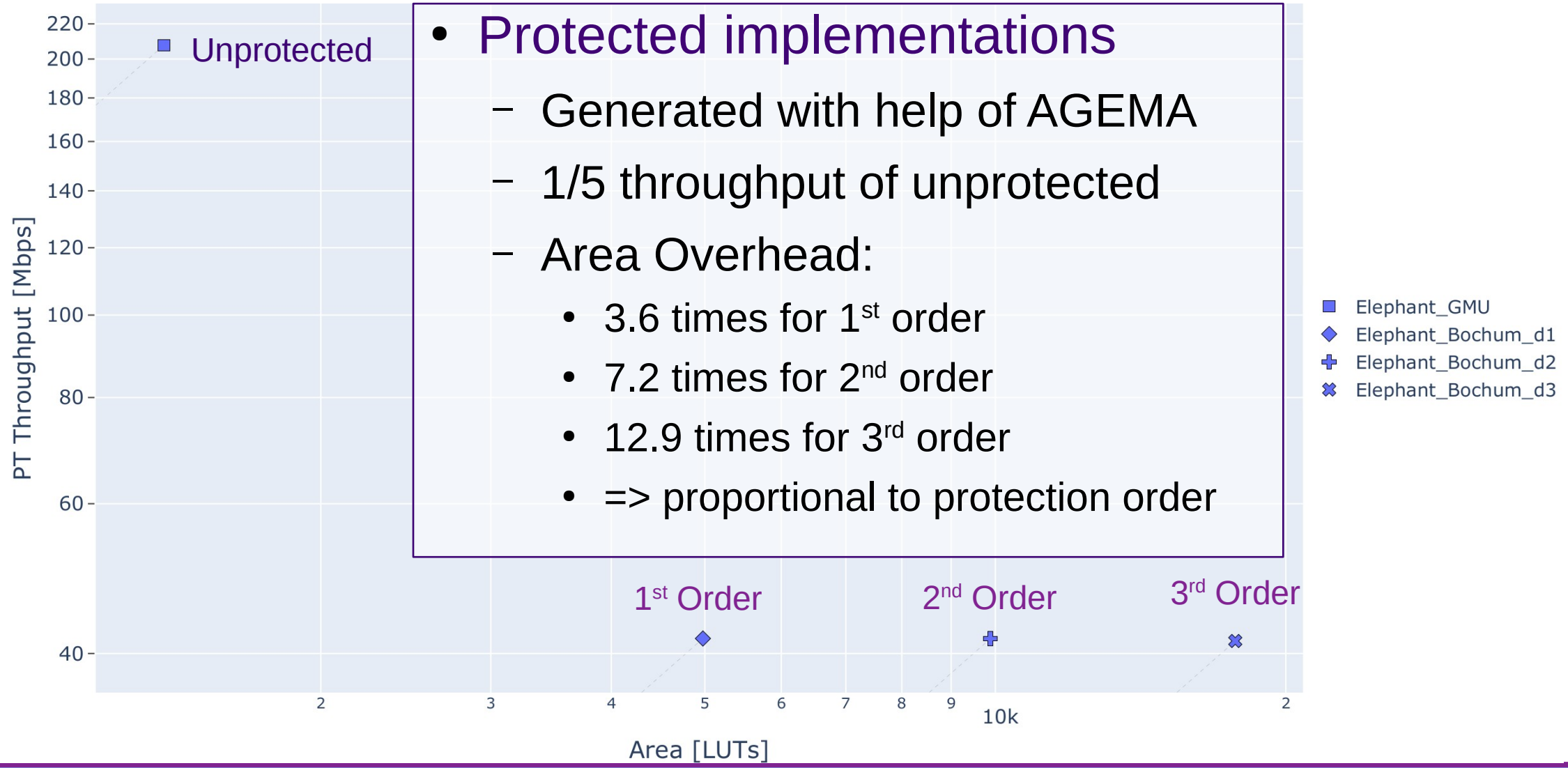
Generation of Results

- Results generated for Xilinx-Artix-7 FPGA
- Device is XC7A100T-2FTG256L of the NewAE CW305 board
- All designs compatible with GMU LWC API
- Latency in clock cycles determined using simulation
- Area and maximum frequency calculated using Xeda



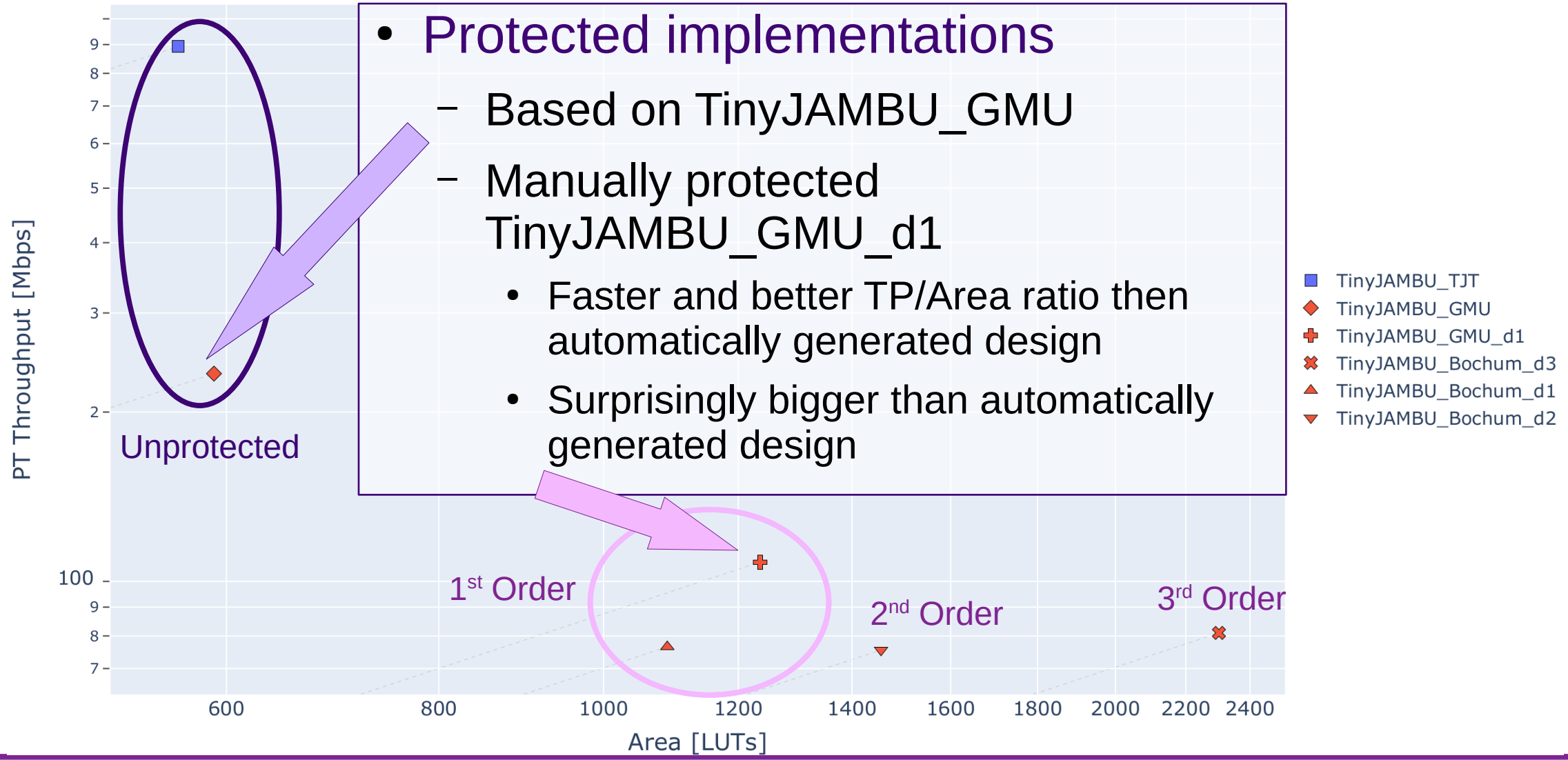
Protected vs Unprotected Hardware Designs

Elephant: PT Throughput vs. Area for Unprotected and Protected Designs



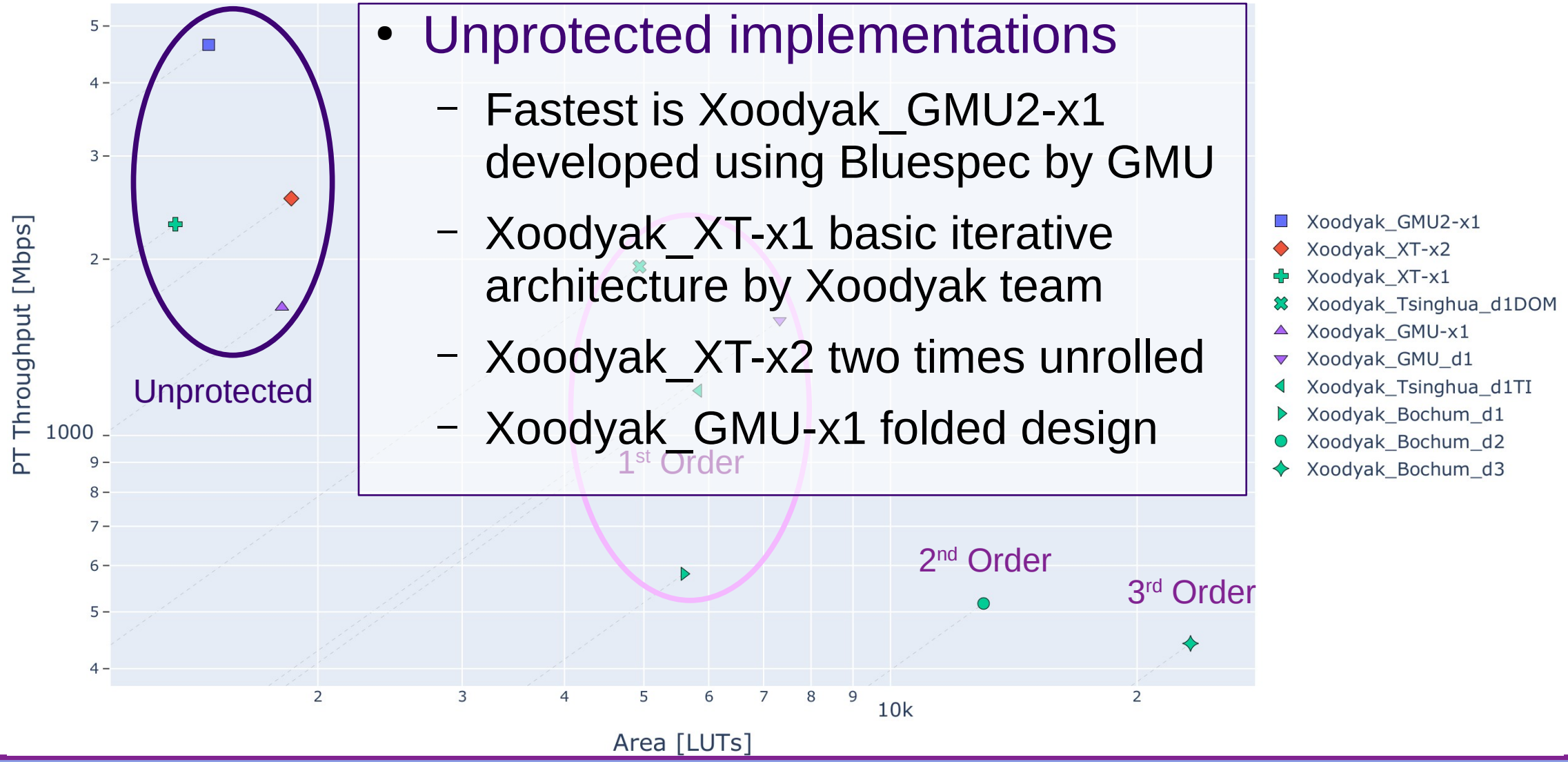
- Protected implementations
  - Generated with help of AGEMA
  - 1/5 throughput of unprotected
  - Area Overhead:
    - 3.6 times for 1st order
    - 7.2 times for 2nd order
    - 12.9 times for 3rd order
    - => proportional to protection order

TinyJAMBU: PT Throughput vs. Area for Unprotected and Protected



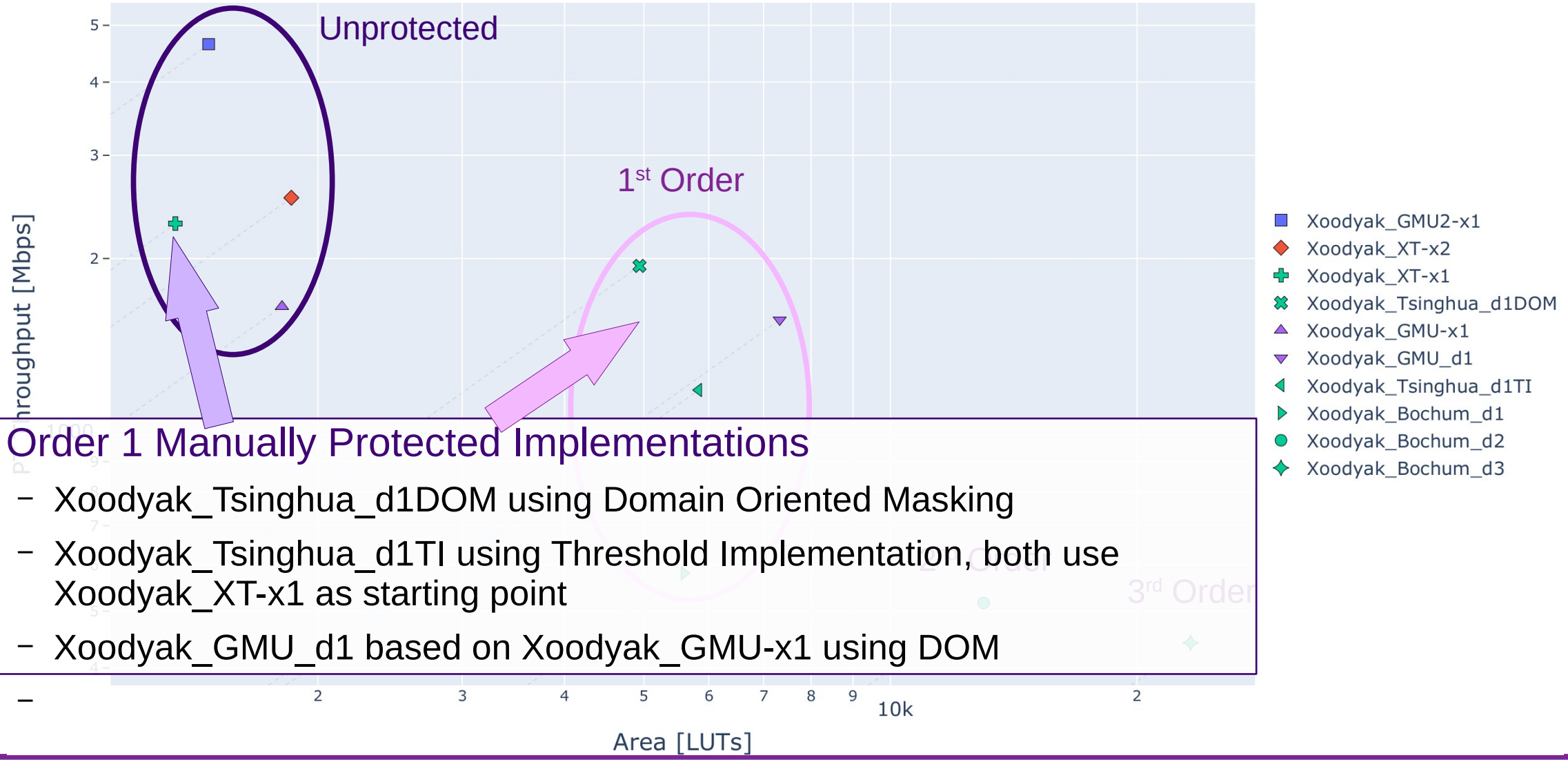
- Protected implementations
  - Based on TinyJAMBU_GMU
  - Manually protected TinyJAMBU_GMU_d1
    - Faster and better TP/Area ratio then automatically generated design
    - Surprisingly bigger than automatically generated design

Xoodyak: PT Throughput vs. Area for Unprotected and Protected Designs



- Unprotected implementations
  - Fastest is Xoodyak_GMU2-x1 developed using Bluespec by GMU
  - Xoodyak_XT-x1 basic iterative architecture by Xoodyak team
  - Xoodyak_XT-x2 two times unrolled
  - Xoodyak_GMU-x1 folded design

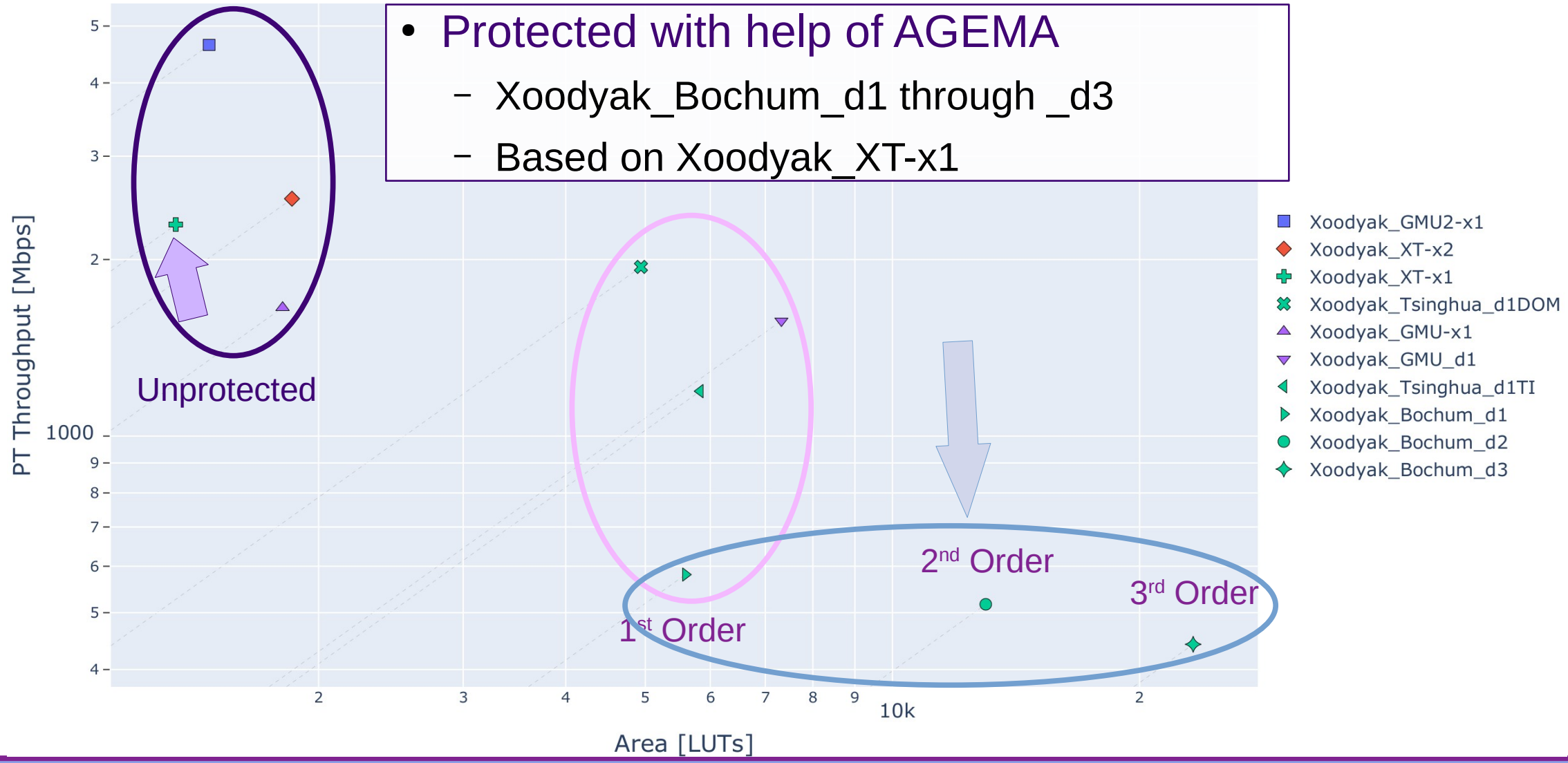
Xoodyak: PT Throughput vs. Area for Unprotected and Protected Designs



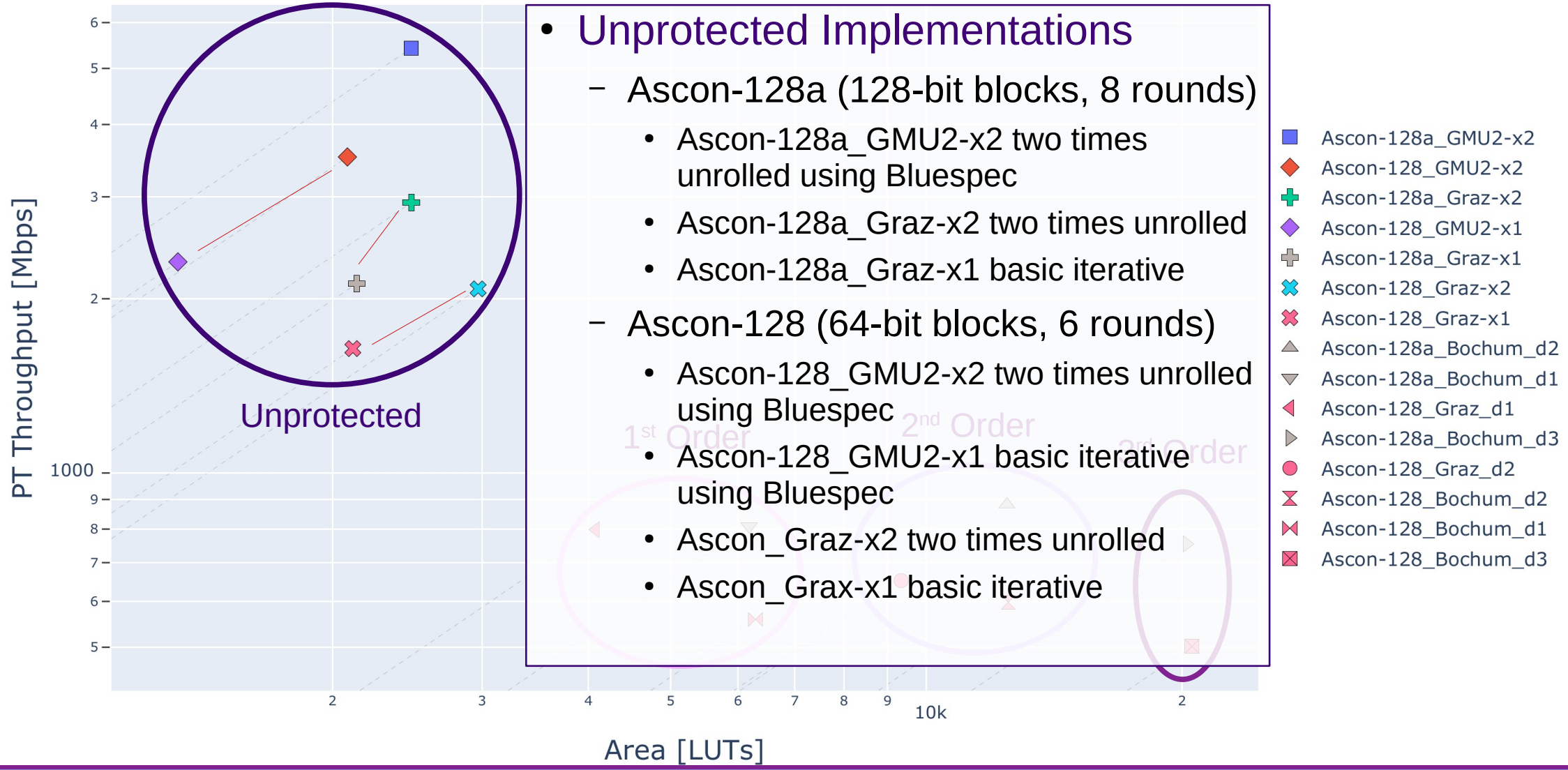
- Order 1 Manually Protected Implementations
  - Xoodyak_Tsinghua_d1DOM using Domain Oriented Masking
  - Xoodyak_Tsinghua_d1TI using Threshold Implementation, both use Xoodyak_XT-x1 as starting point
  - Xoodyak_GMU_d1 based on Xoodyak_GMU-x1 using DOM

Xoodyak: PT Throughput vs. Area for Unprotected and Protected Designs

- Protected with help of AGEMA
  - Xoodyak_Bochum_d1 through _d3
  - Based on Xoodyak_XT-x1

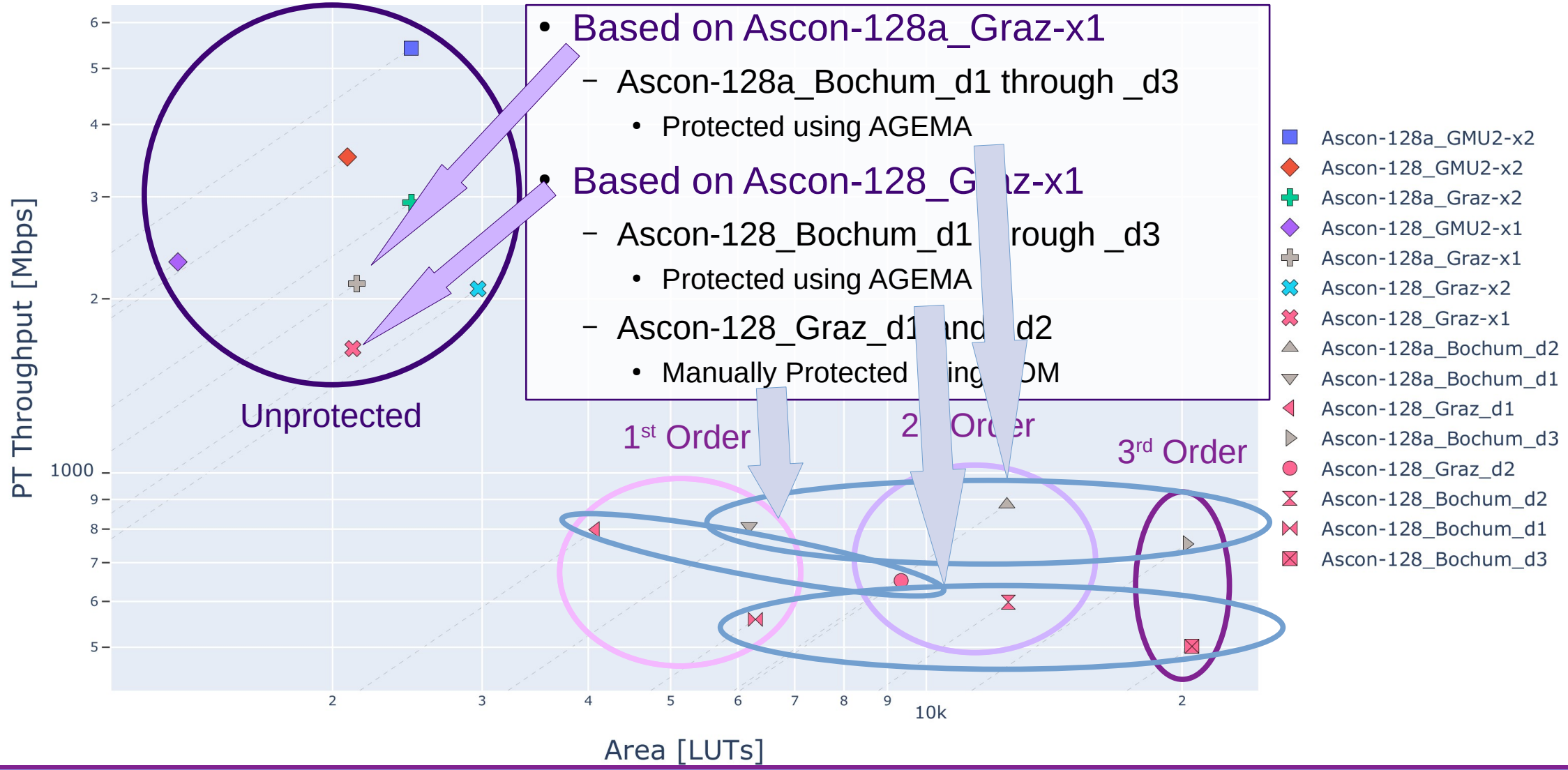


Ascon: PT Throughput vs. Area for Unprotected and Protected Designs



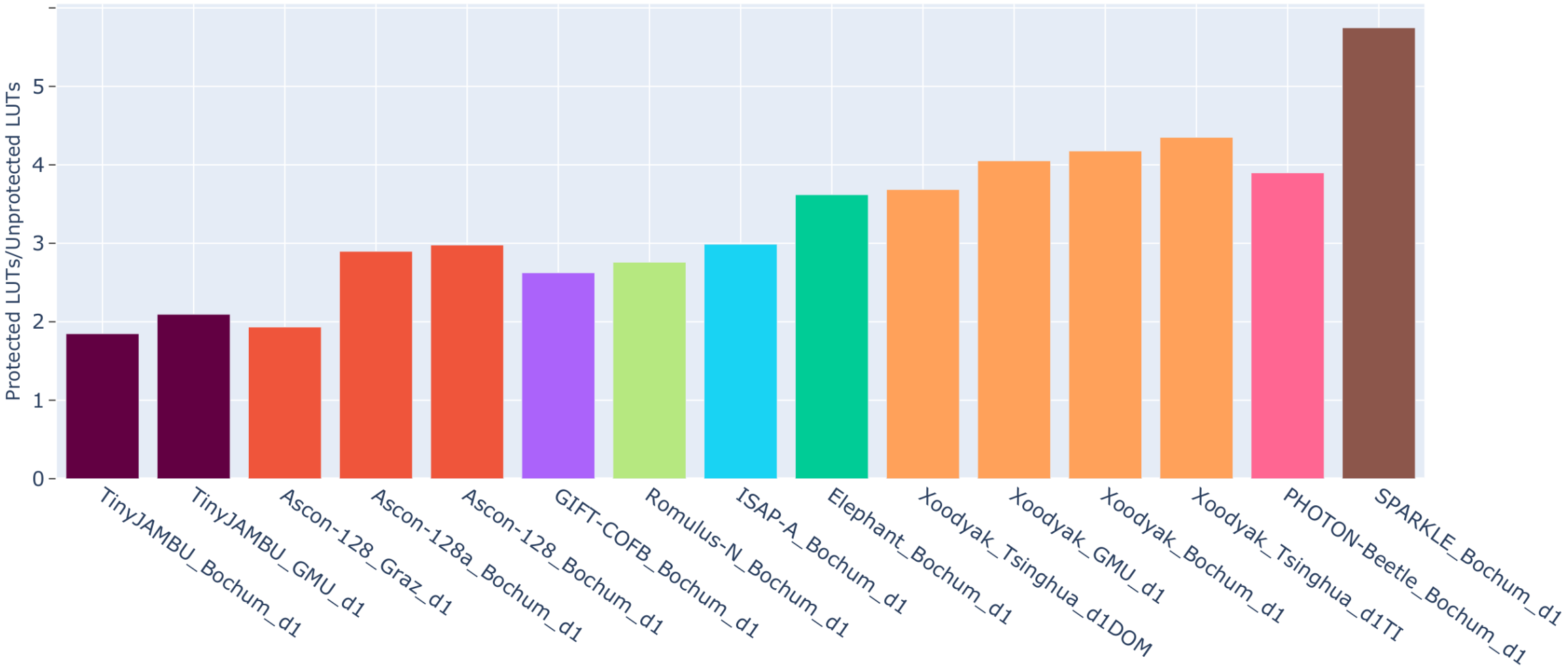
- Unprotected Implementations
  - Ascon-128a (128-bit blocks, 8 rounds)
    - Ascon-128a_GMU2-x2 two times unrolled using Bluespec
    - Ascon-128a_Graz-x2 two times unrolled
    - Ascon-128a_Graz-x1 basic iterative
  - Ascon-128 (64-bit blocks, 6 rounds)
    - Ascon-128_GMU2-x2 two times unrolled using Bluespec
    - Ascon-128_GMU2-x1 basic iterative using Bluespec
    - Ascon_Graz-x2 two times unrolled
    - Ascon_Grax-x1 basic iterative

Ascon: PT Throughput vs. Area for Unprotected and Protected Designs

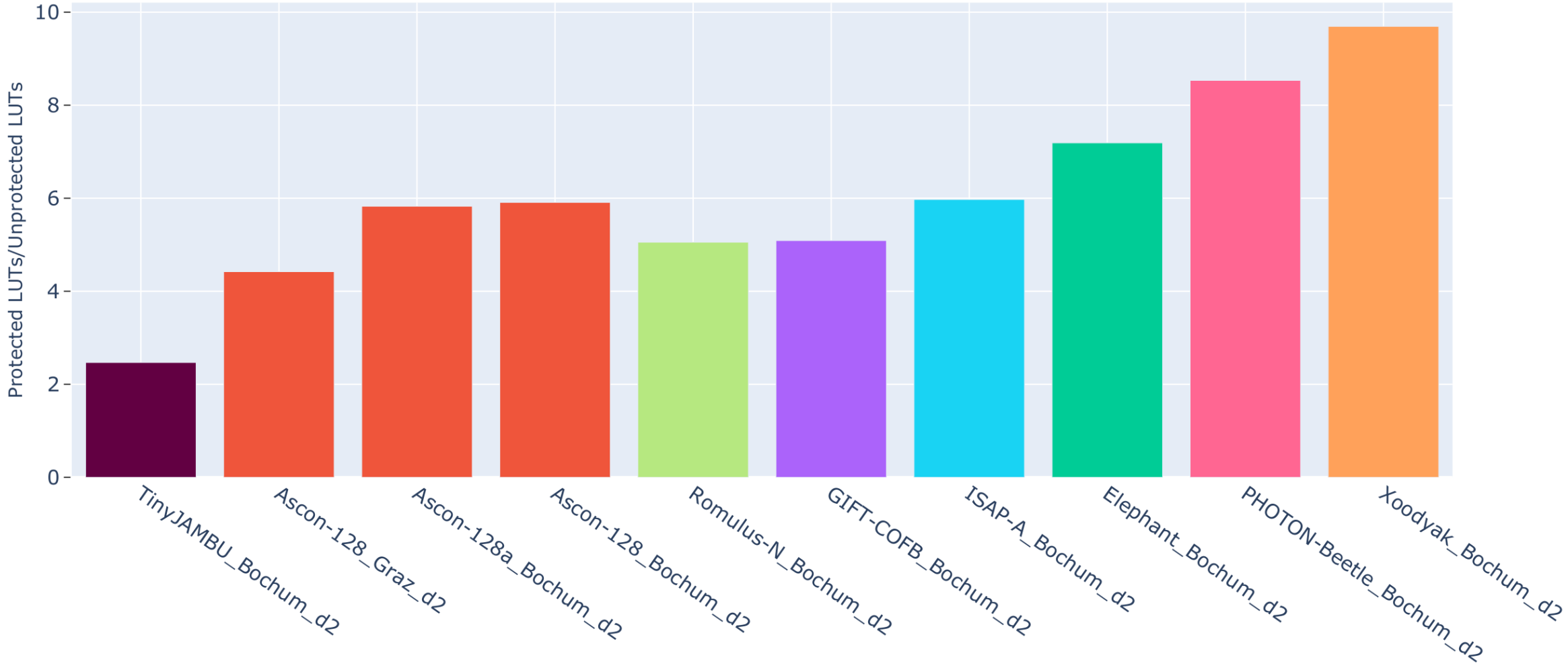


- Based on Ascon-128a_Graz-x1
  - Ascon-128a_Bochum_d1 through _d3
    - Protected using AGEMA
- Based on Ascon-128_G az-x1
  - Ascon-128_Bochum_d1 rough _d3
    - Protected using AGEMA
  - Ascon-128_Graz_d1 and d2
    - Manually Protected ing DM

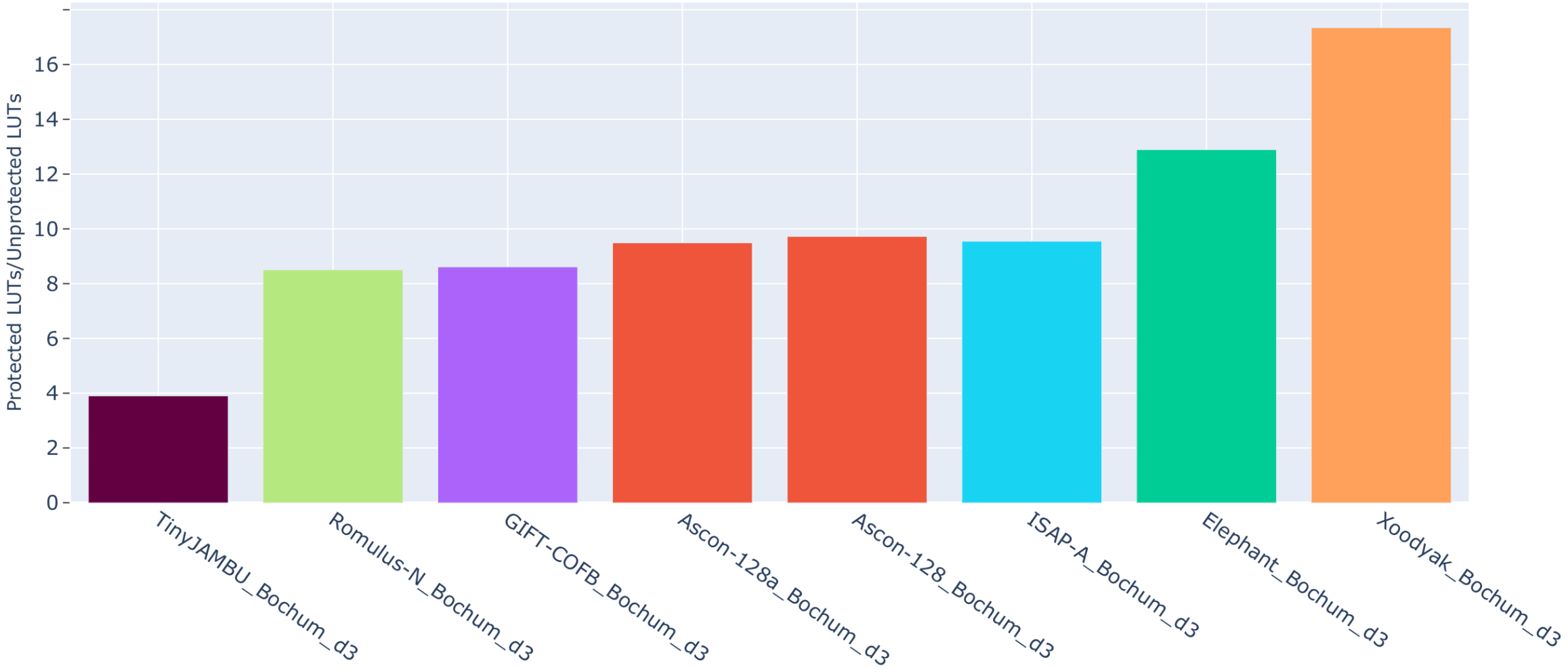
1st Order Protected Area over Unprotected Area



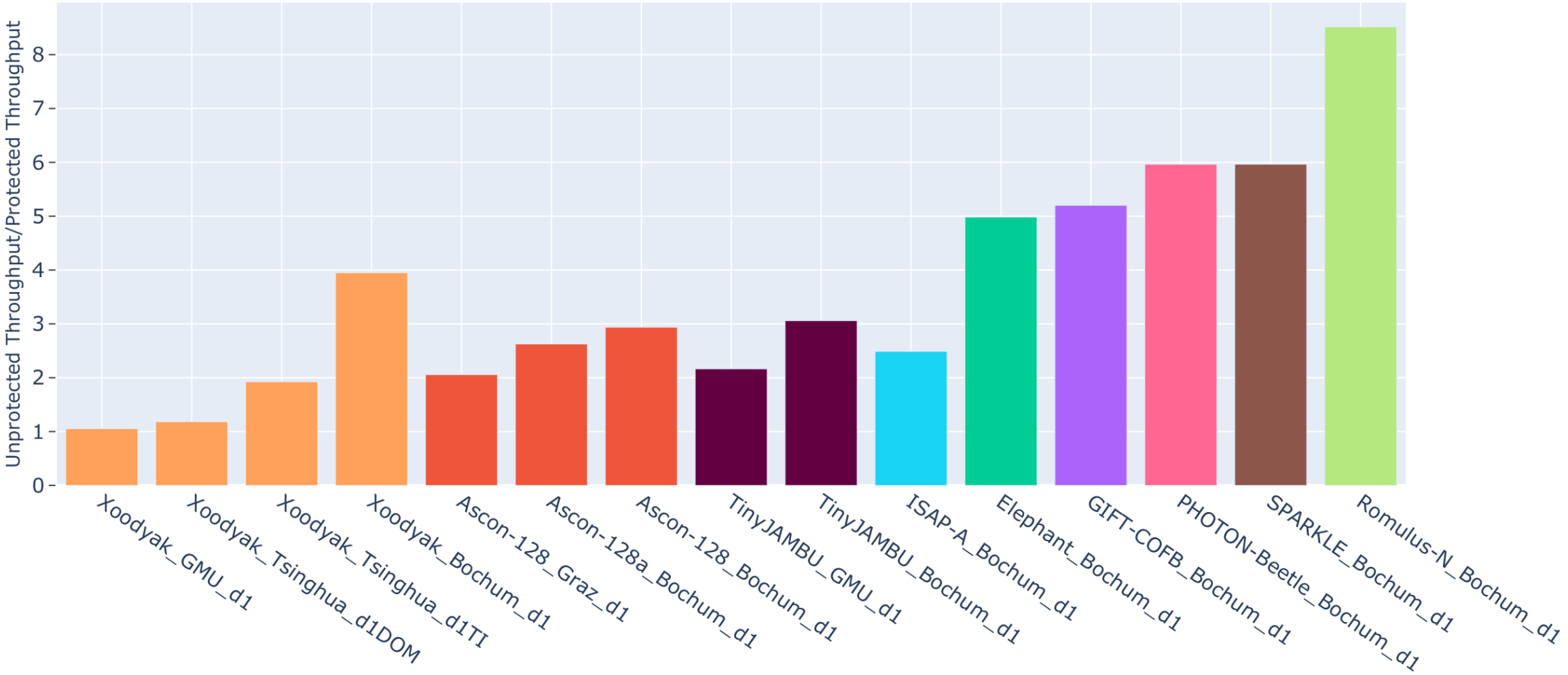
2nd Order Protected Area over Unprotected Area



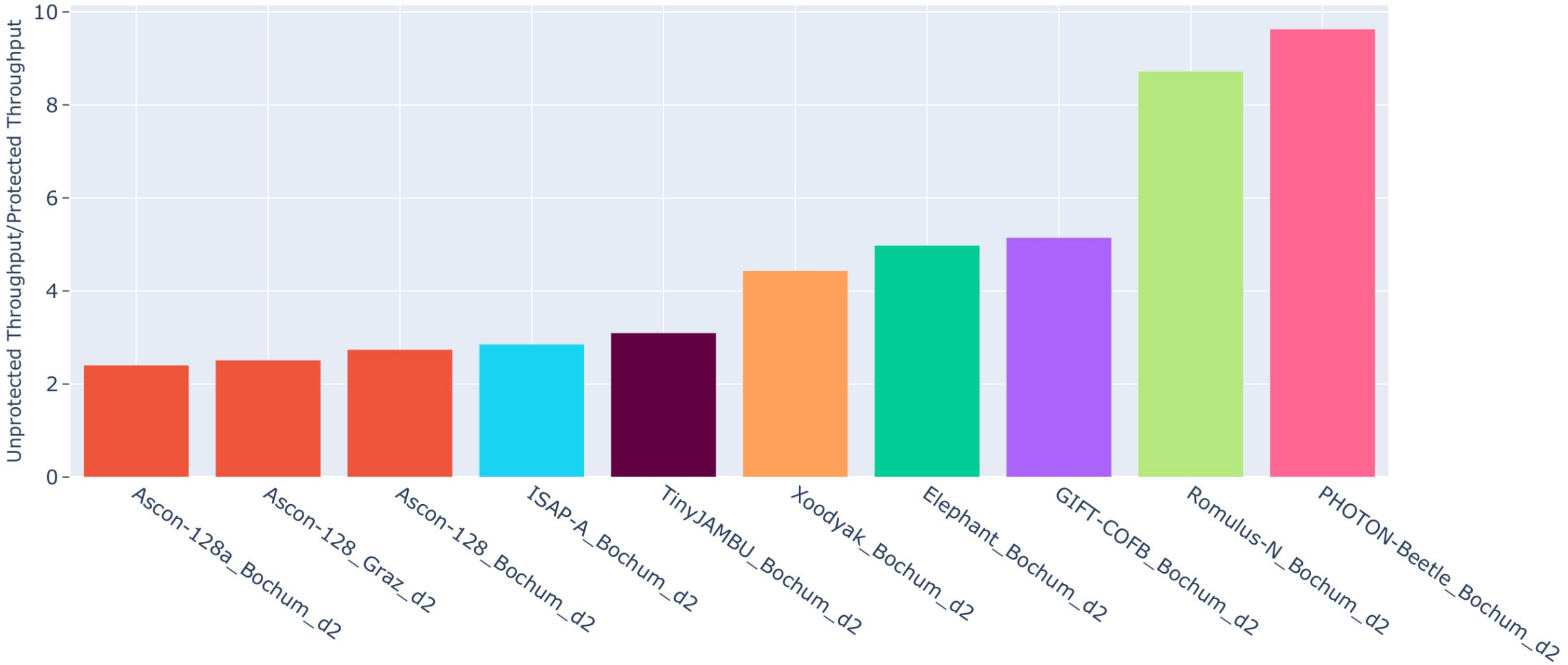
3rd Order Protected Area over Unprotected Area



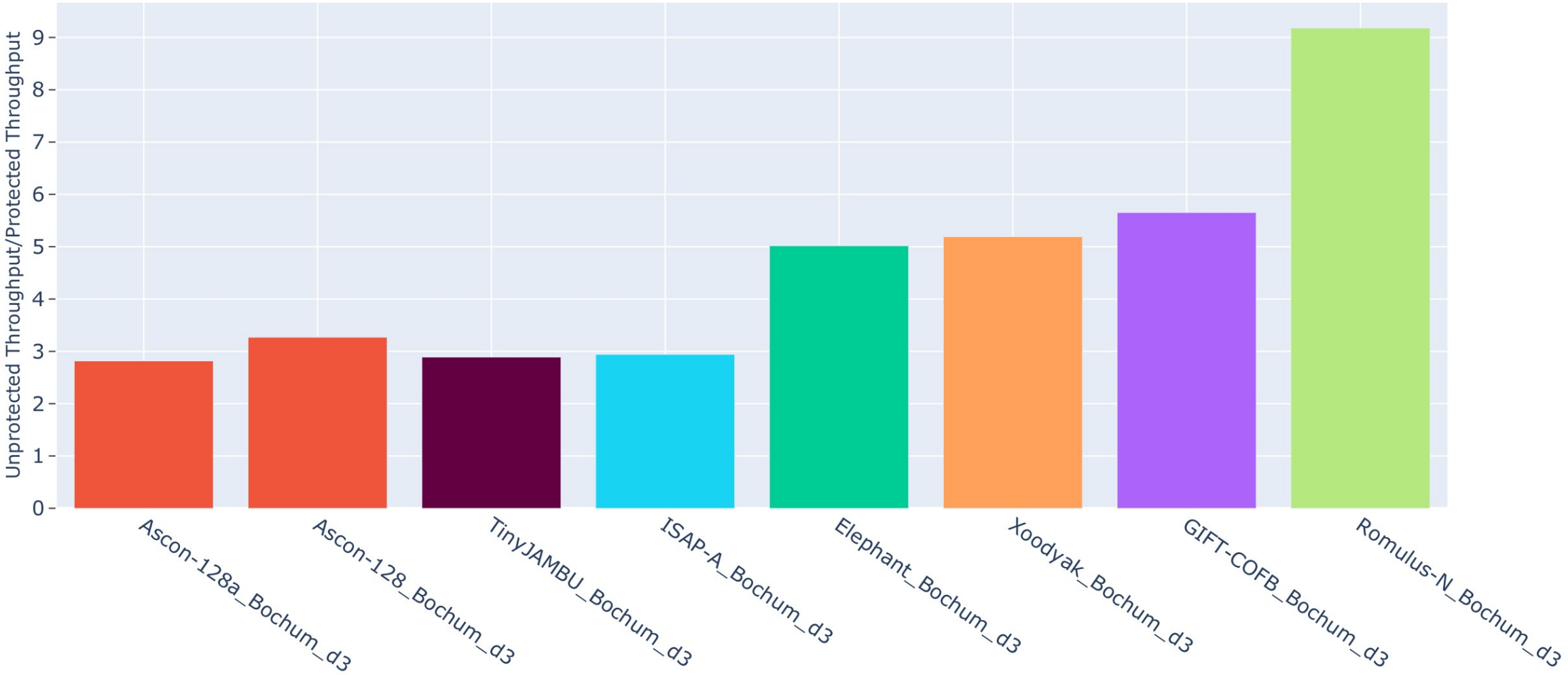
Unprotected Long-message Throughput over 1st Order Protected



Unprotected Long-message Throughput over 2nd Order Protected



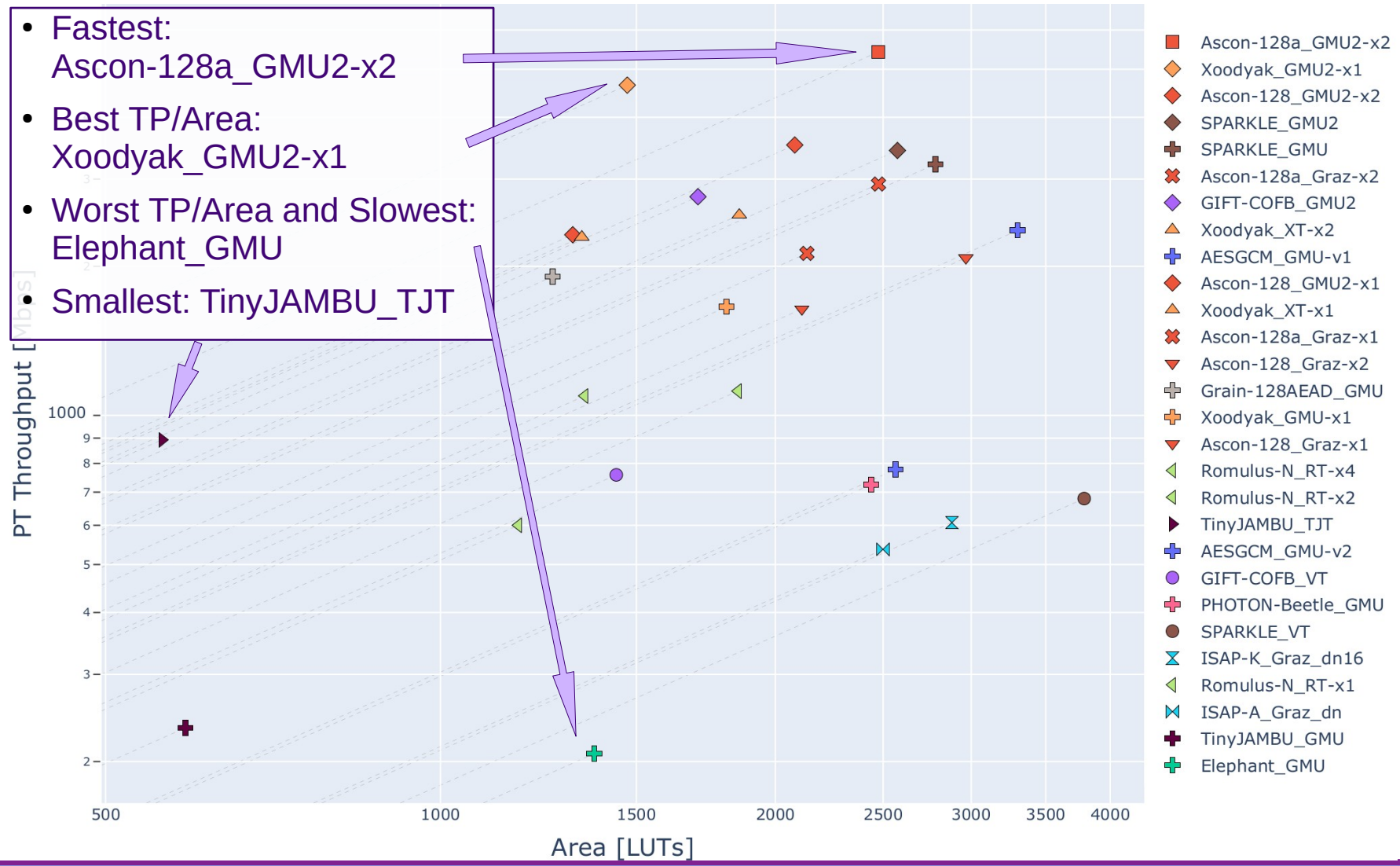
Unprotected Long-message Throughput over 3rd Order Protected



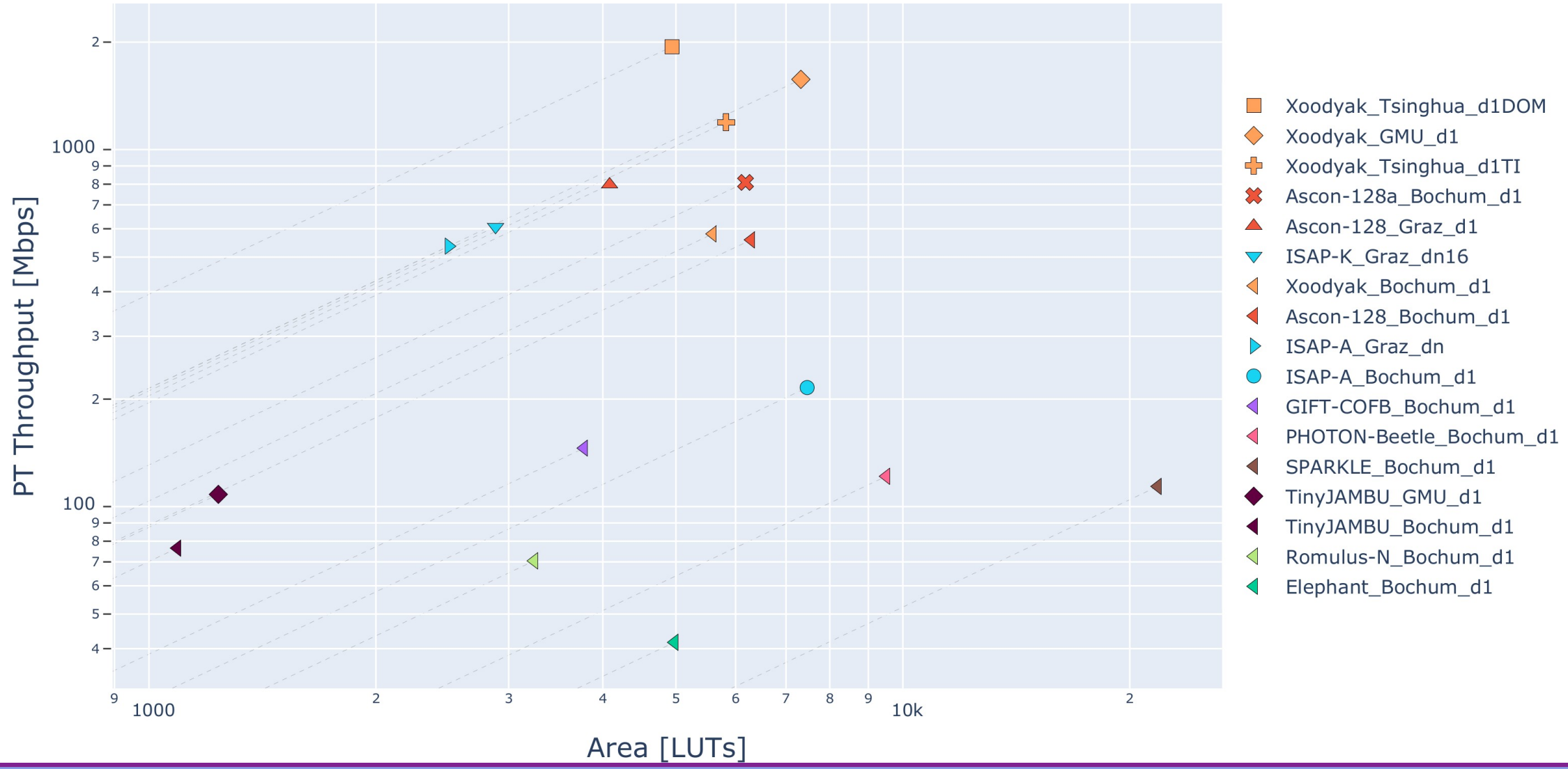
Benchmarking of Hardware Implementations

Encryption Throughput vs LUTs for Long Messages (Unprotected)

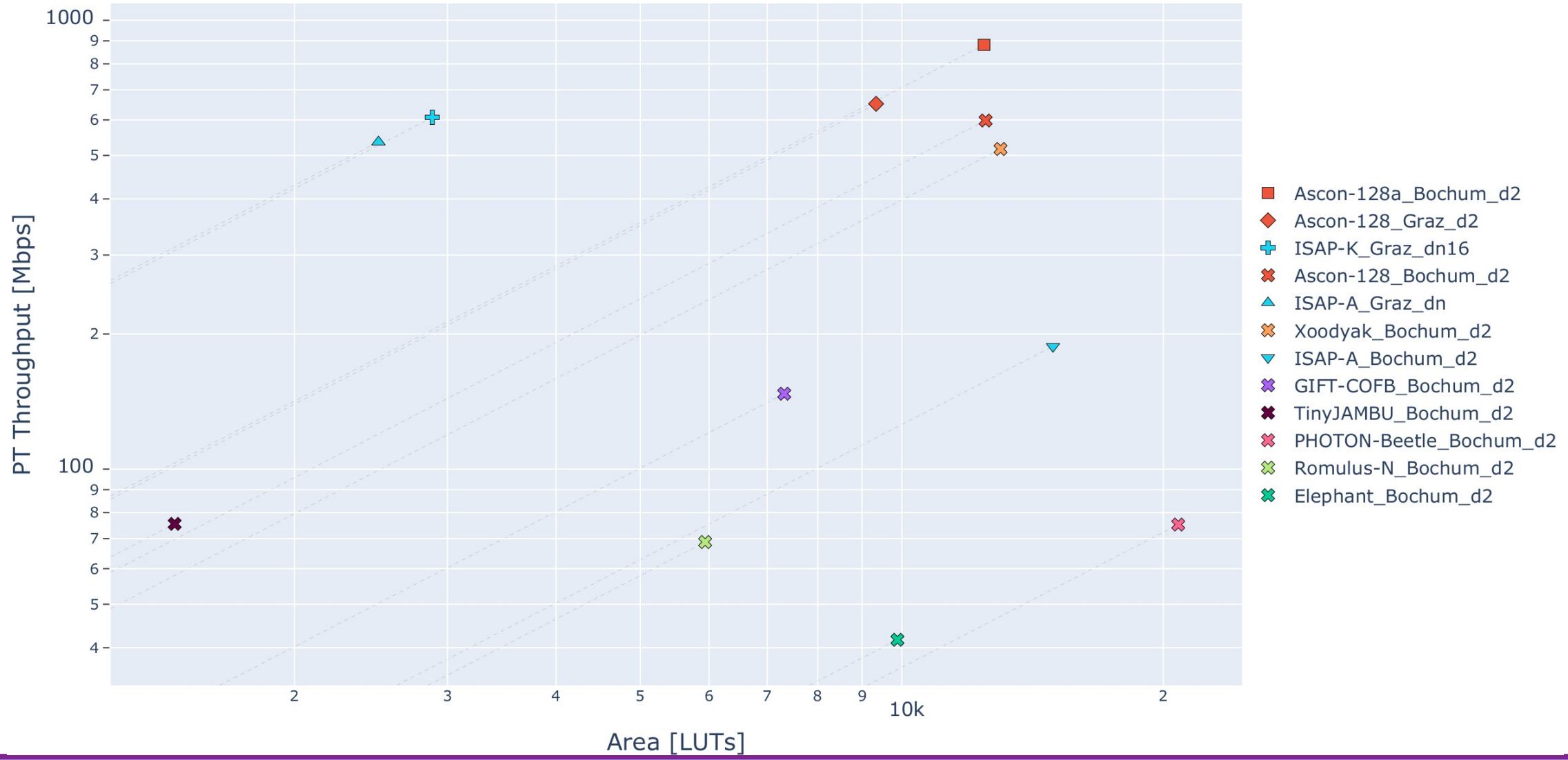
- Fastest: Ascon-128a_GMU2-x2
- Best TP/Area: Xoodyak_GMU2-x1
- Worst TP/Area and Slowest: Elephant_GMU
- Smallest: TinyJAMBU_TJT



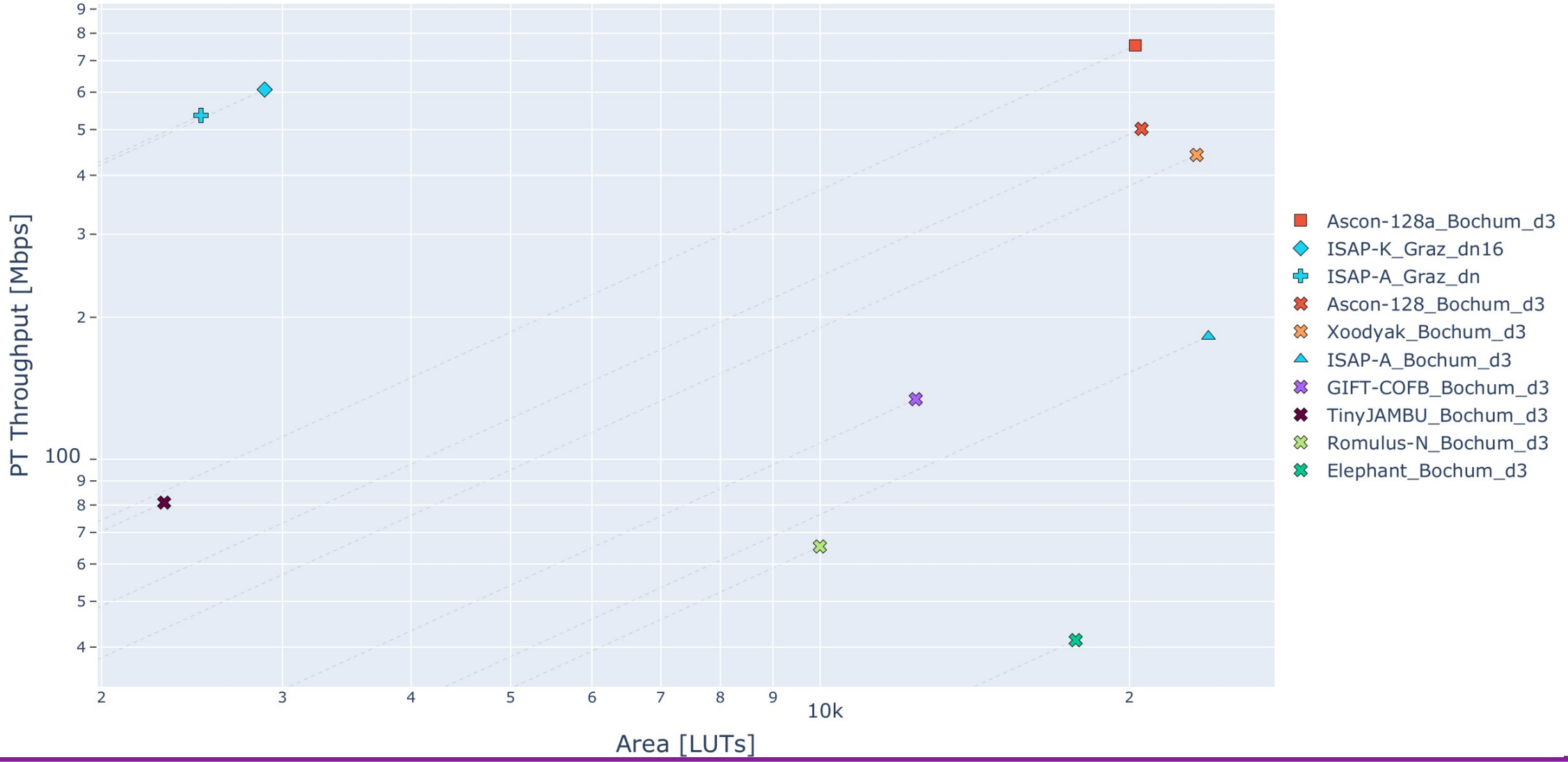
Encryption Throughput vs LUTs for Long Messages (1st Order Protected)



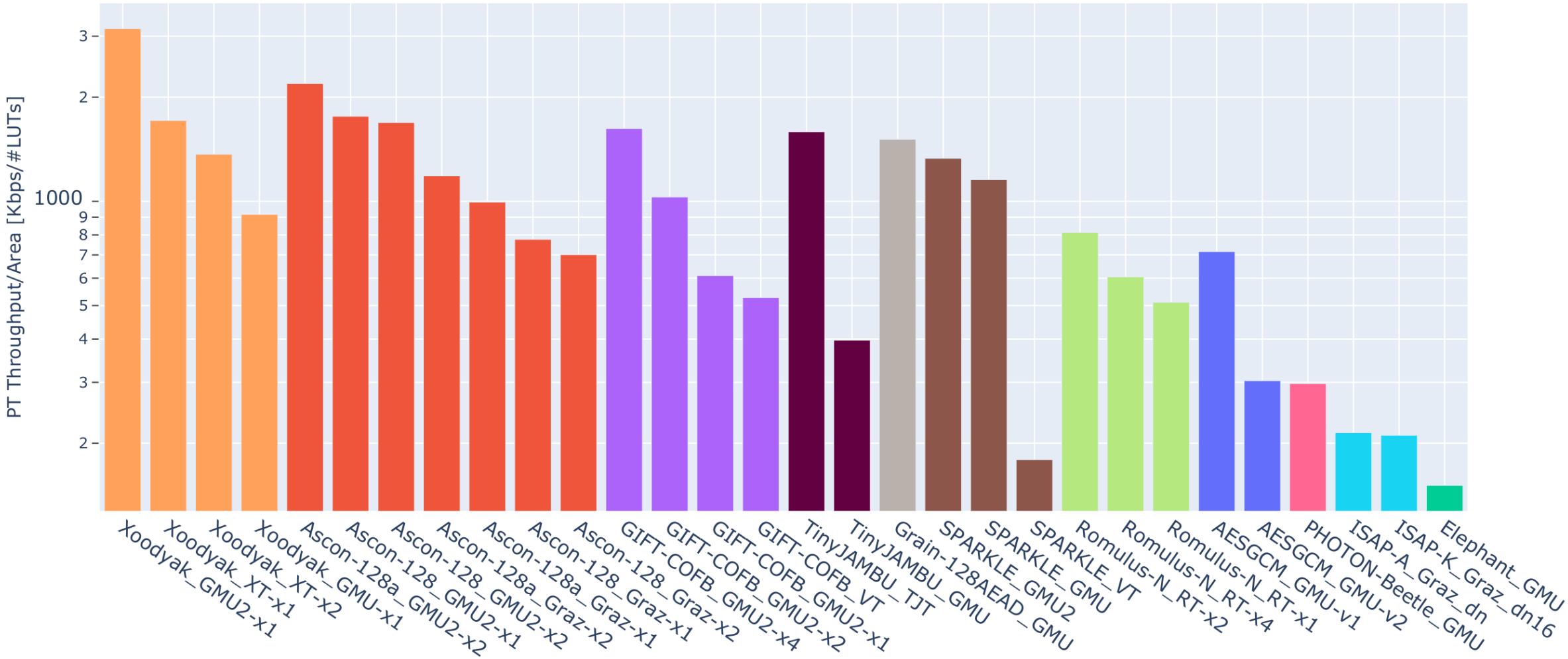
Encryption Throughput vs LUTs for Long Messages (2nd Order Protected)



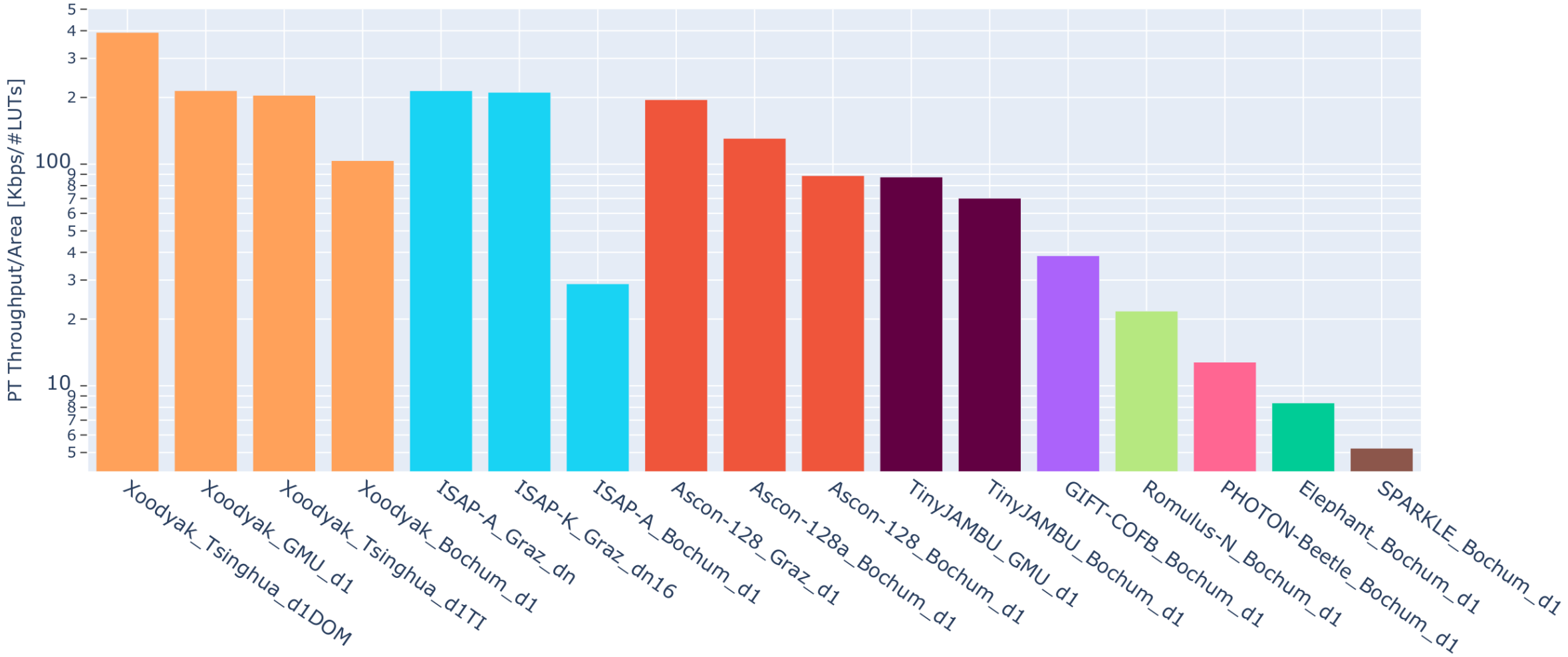
Encryption Throughput vs LUTs for Long Messages (3rd Order Protected)



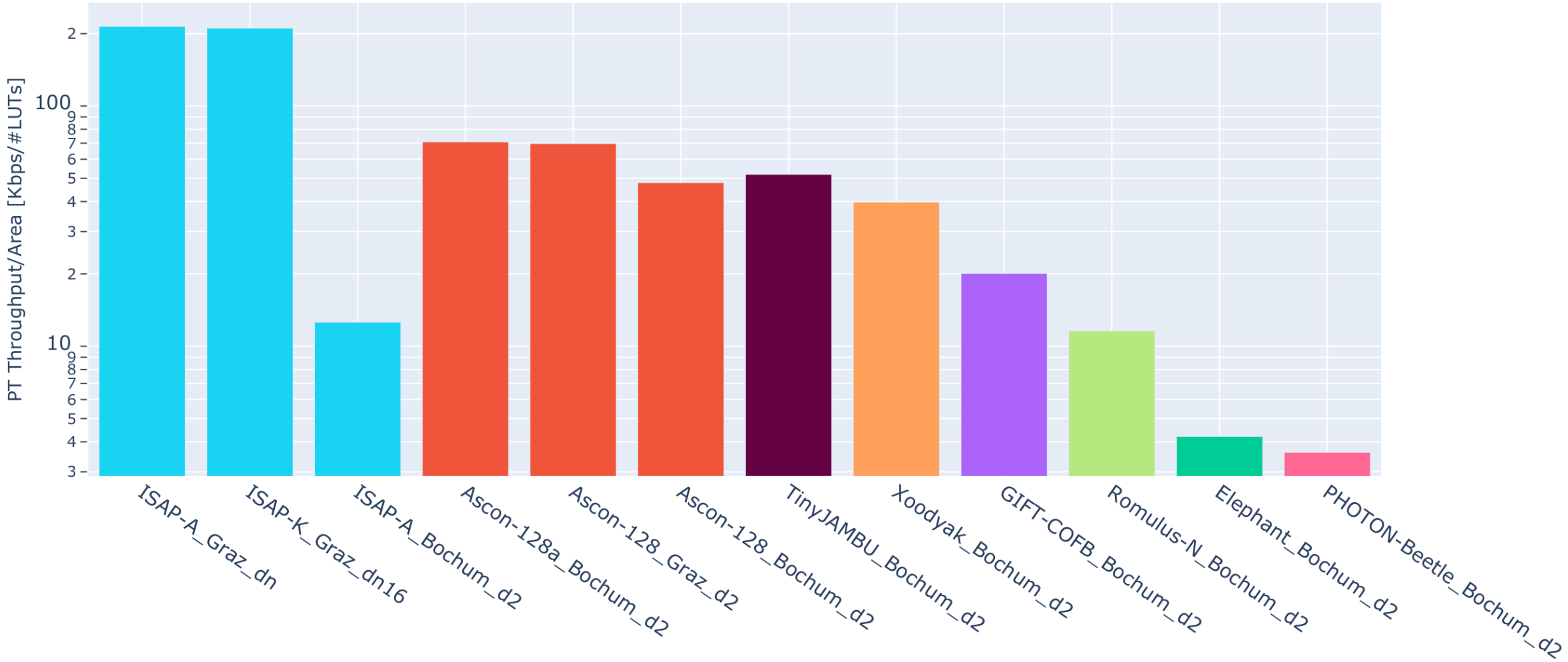
Encryption Throughput over Area for Long Messages (Unprotected)



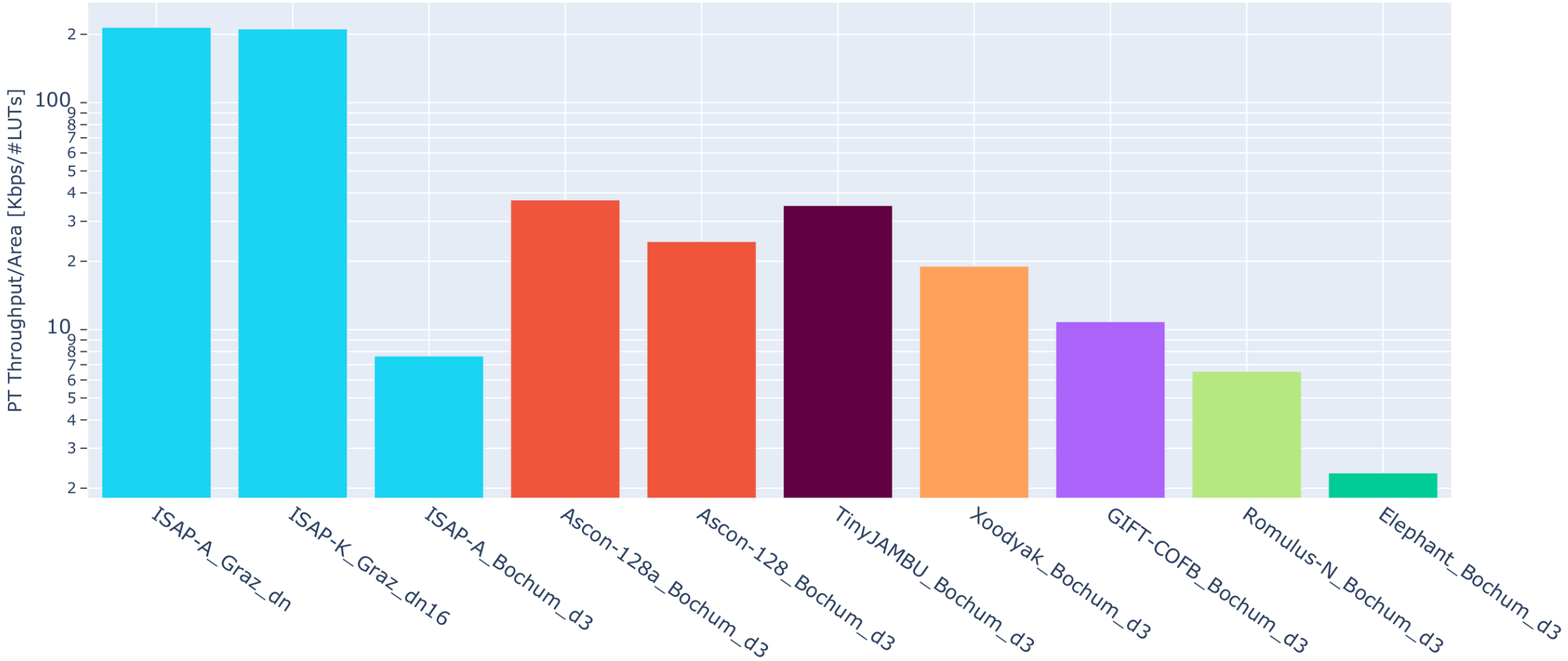
Encryption Throughput over Area for Long Messages (1st Order Protected)



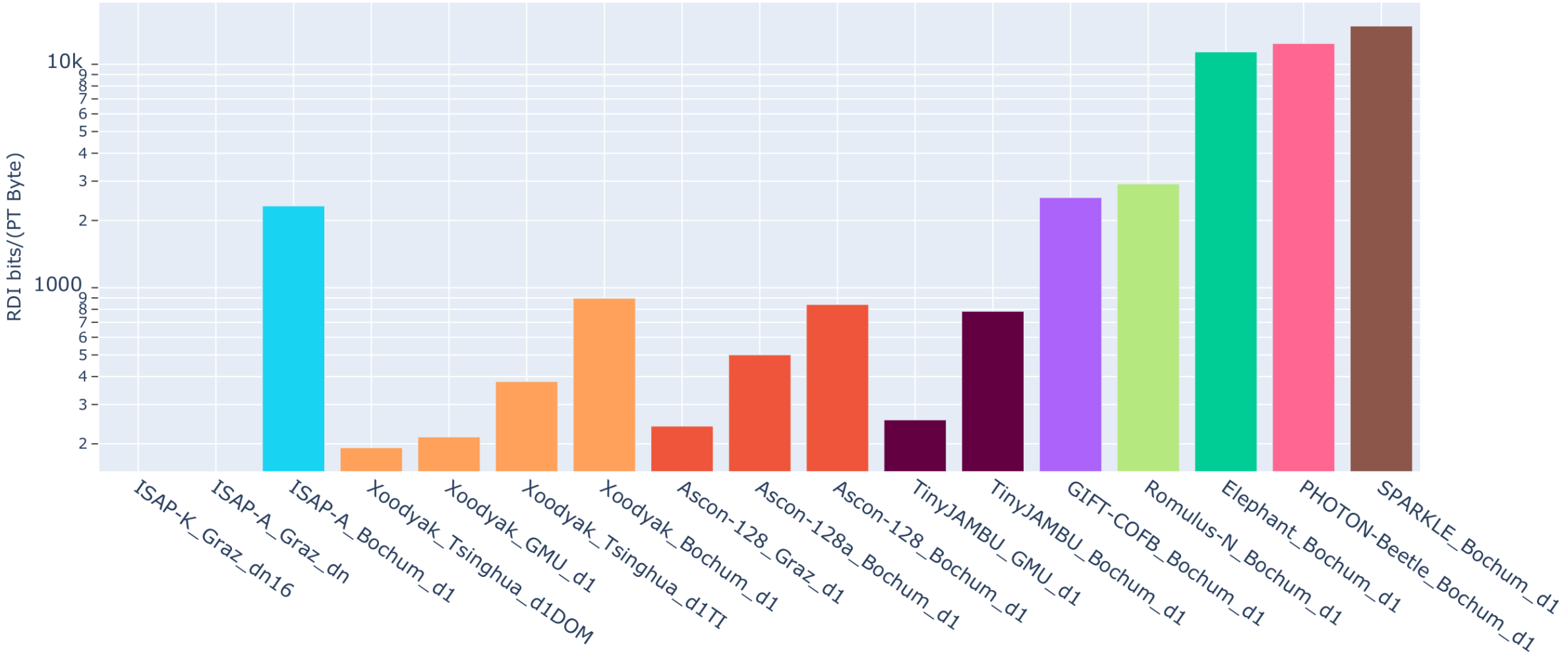
Encryption Throughput over Area for Long Messages (2nd Order Protected)



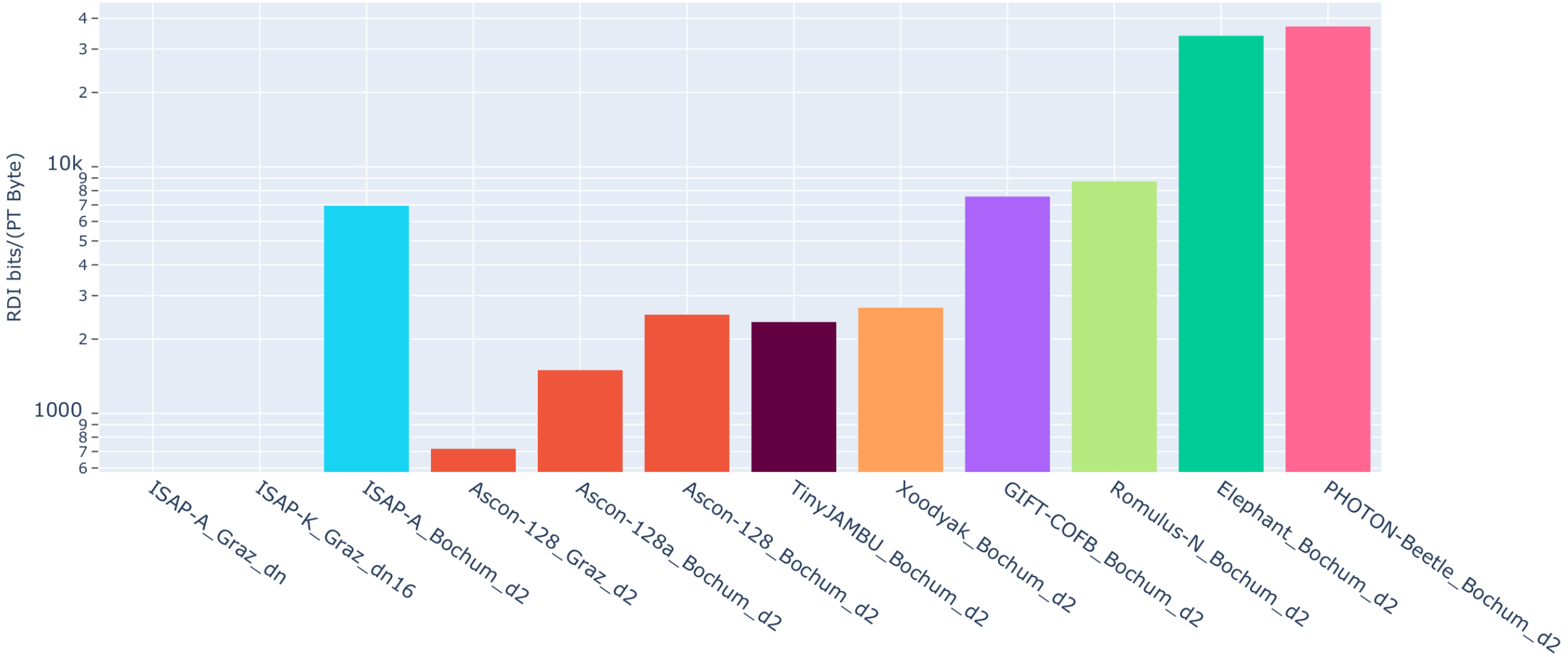
Encryption Throughput over Area for Long Messages (3rd Order Protected)



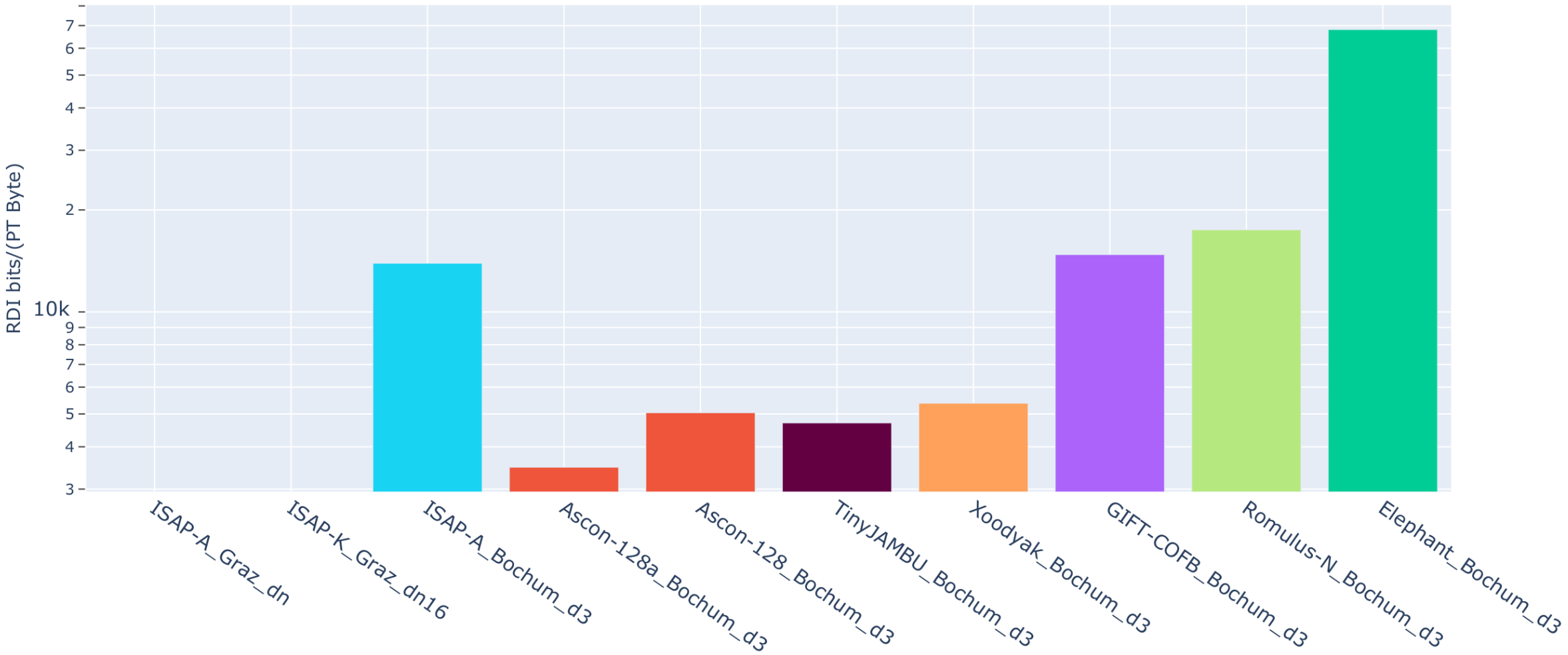
Random Bits per Plaintext Byte (1st Order Protected)



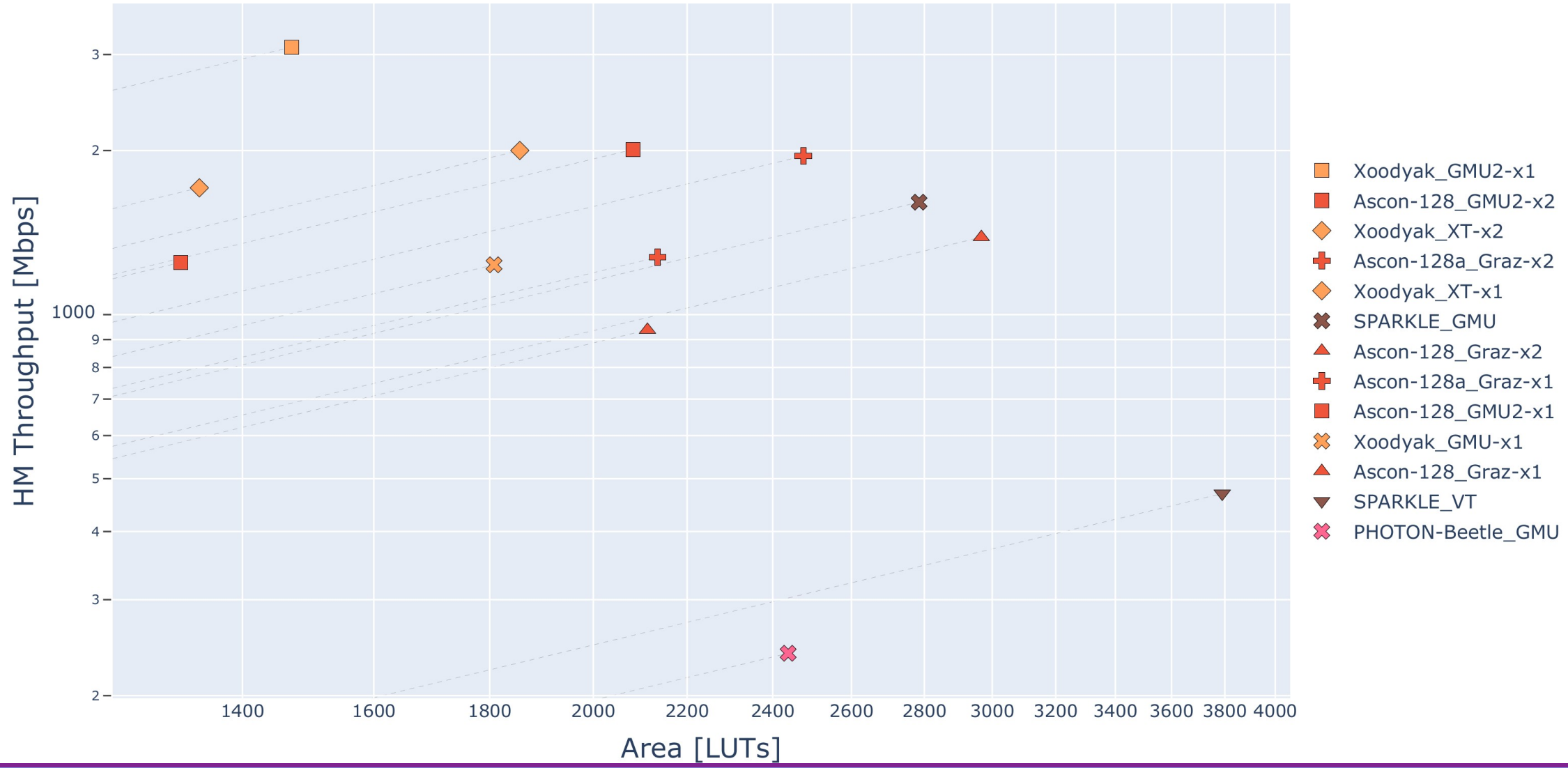
Random Bits per Plaintext Byte (2nd Order Protected)



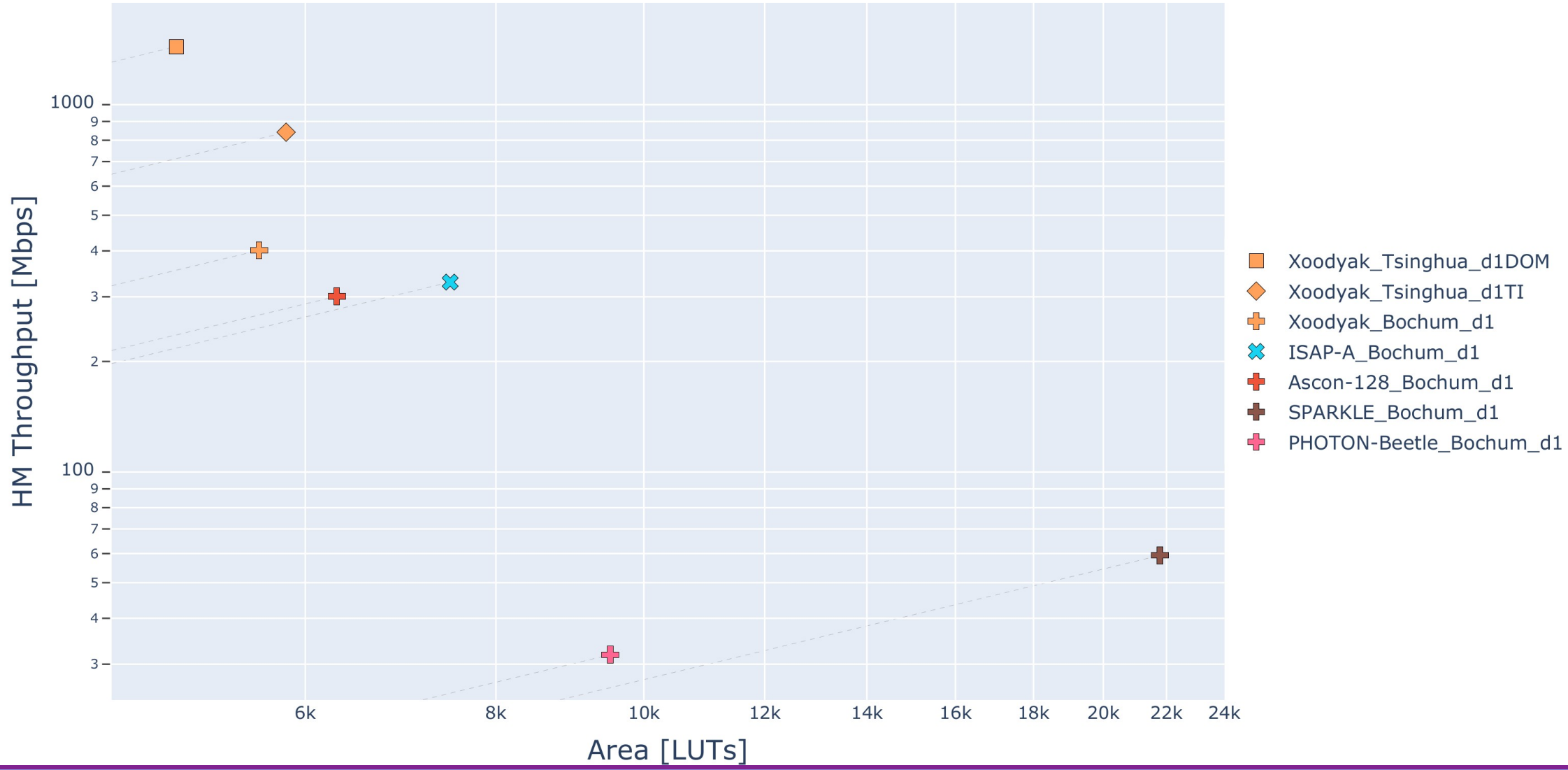
Random Bits per Plaintext Byte (3rd Order Protected)



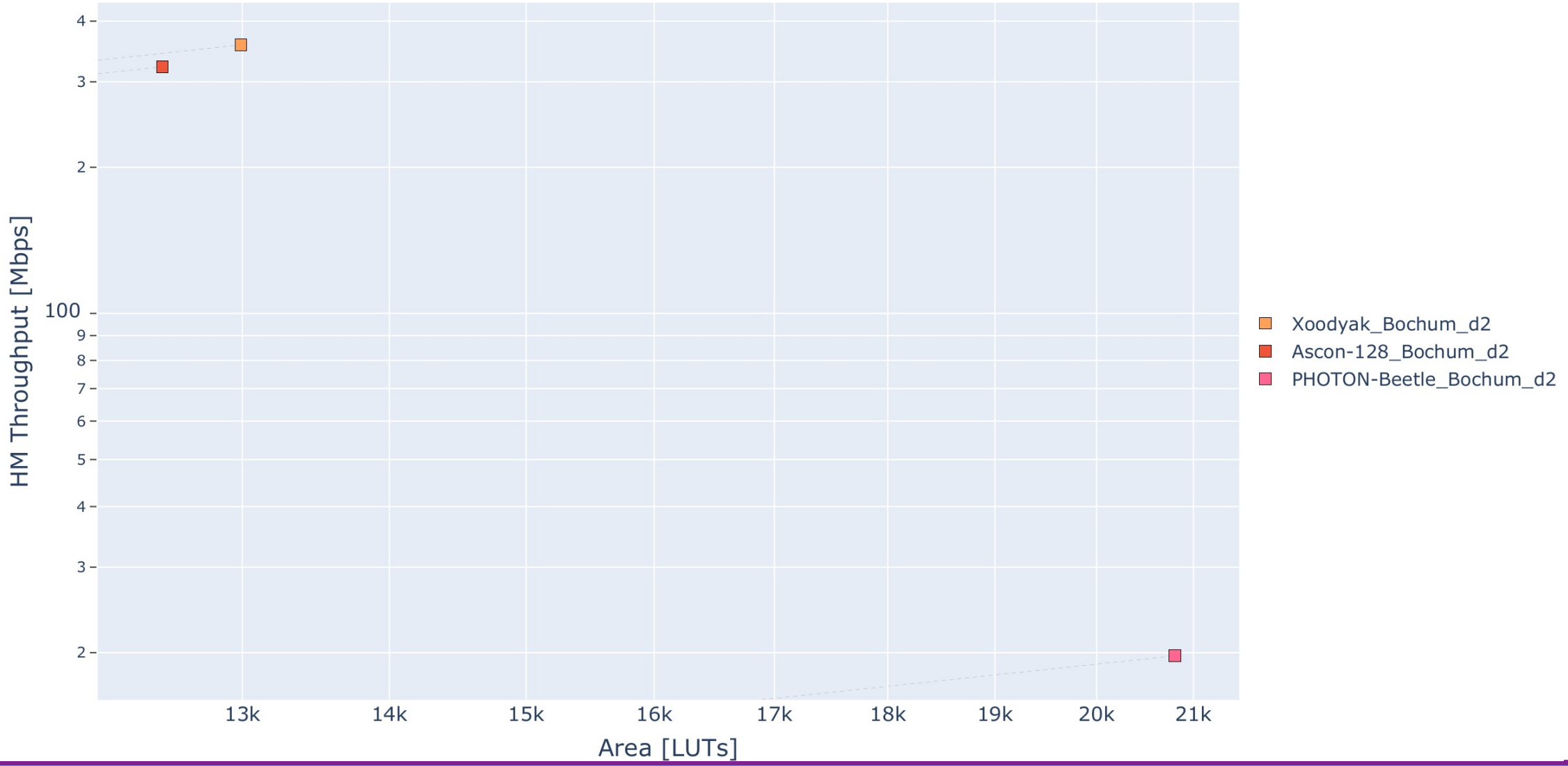
Hashing Throughput vs LUTs for Long Messages (Unprotected)



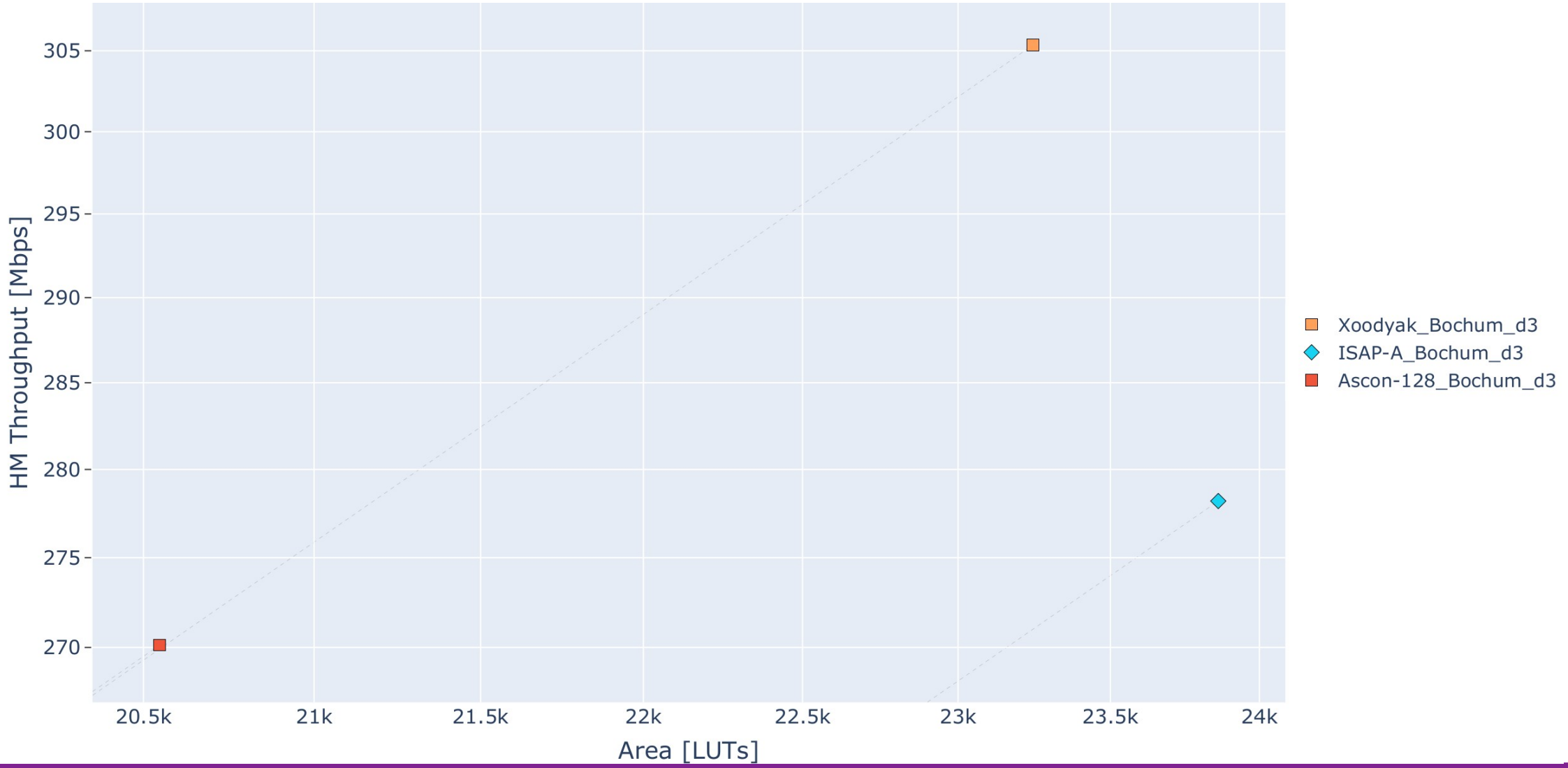
Hashing Throughput vs LUTs for Long Messages (1st Order Protected)



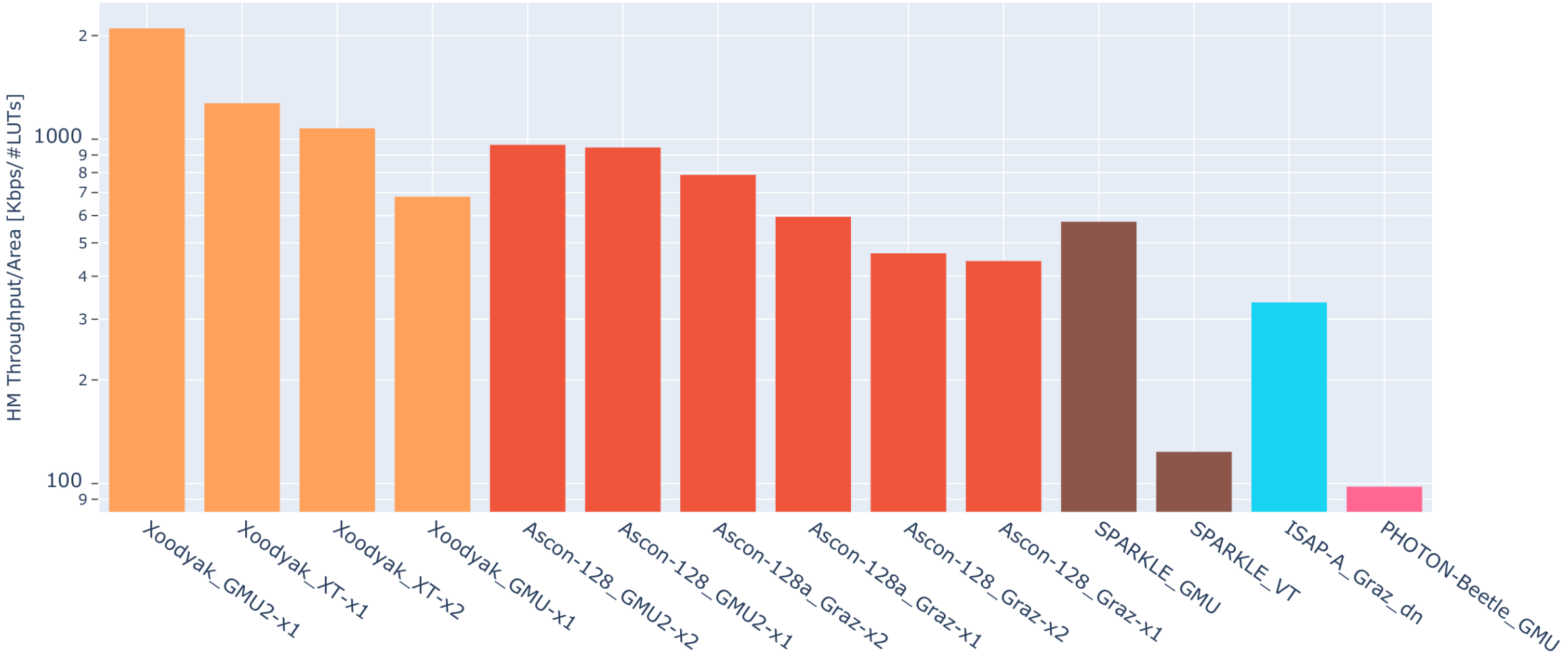
Hashing Throughput vs LUTs for Long Messages (2nd Order Protected)



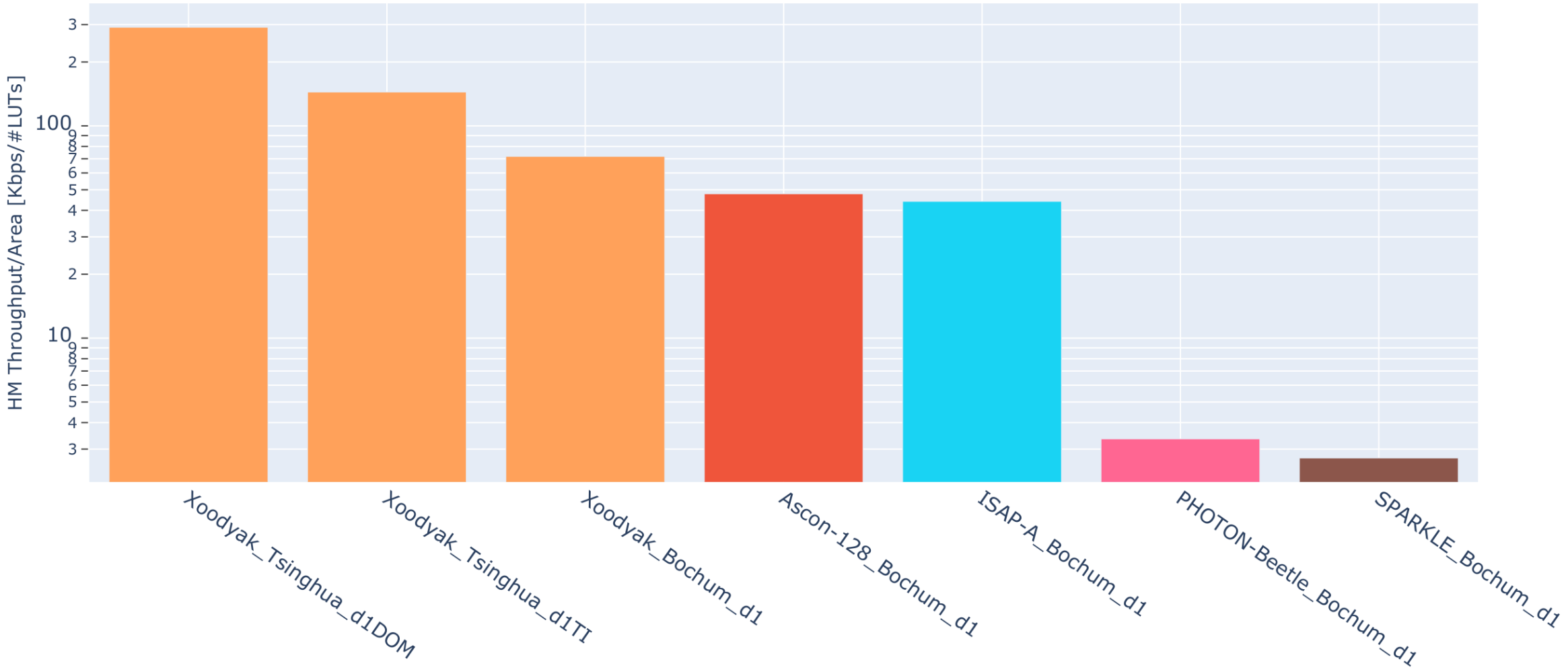
Hashing Throughput vs LUTs for Long Messages (3rd Order Protected)



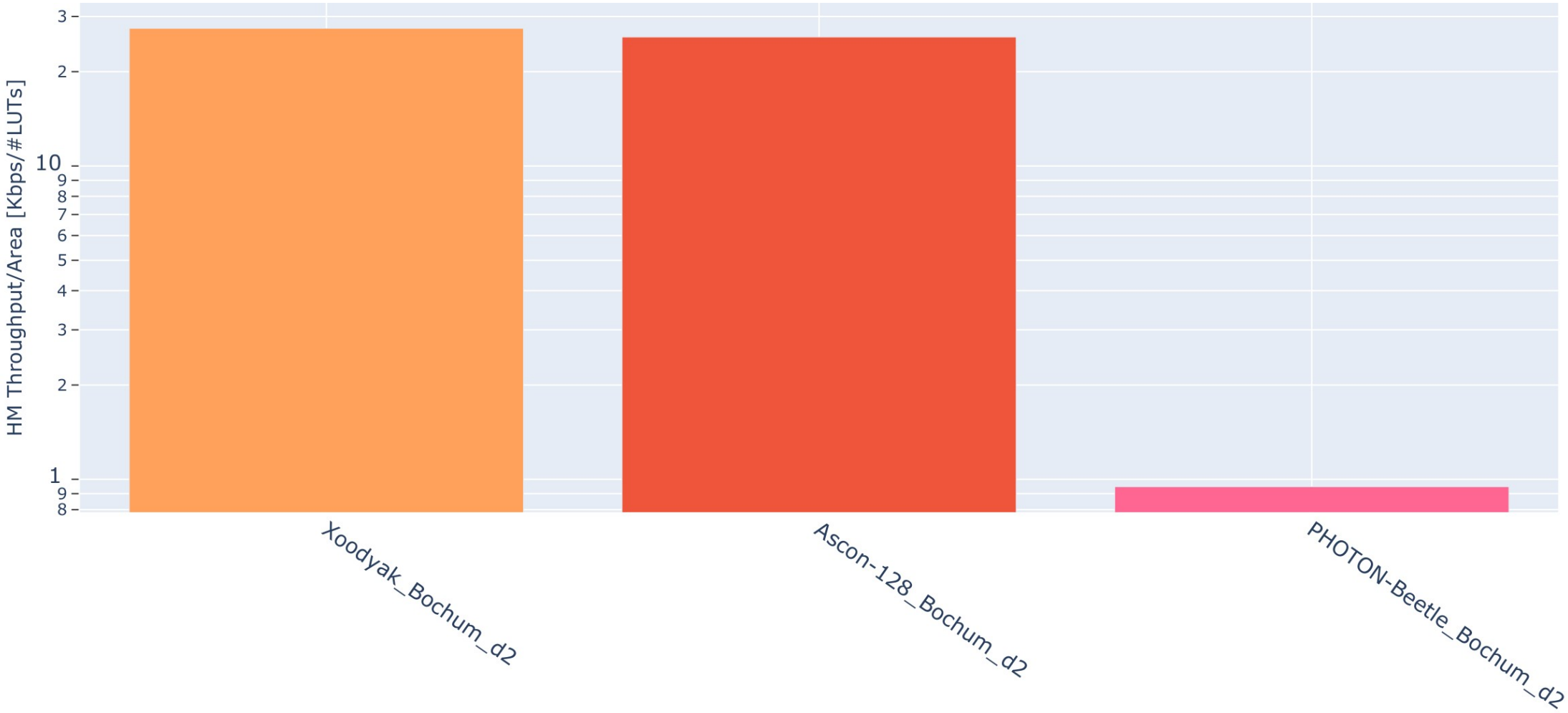
Hashing Throughput over Area for Long Messages (Unprotected)



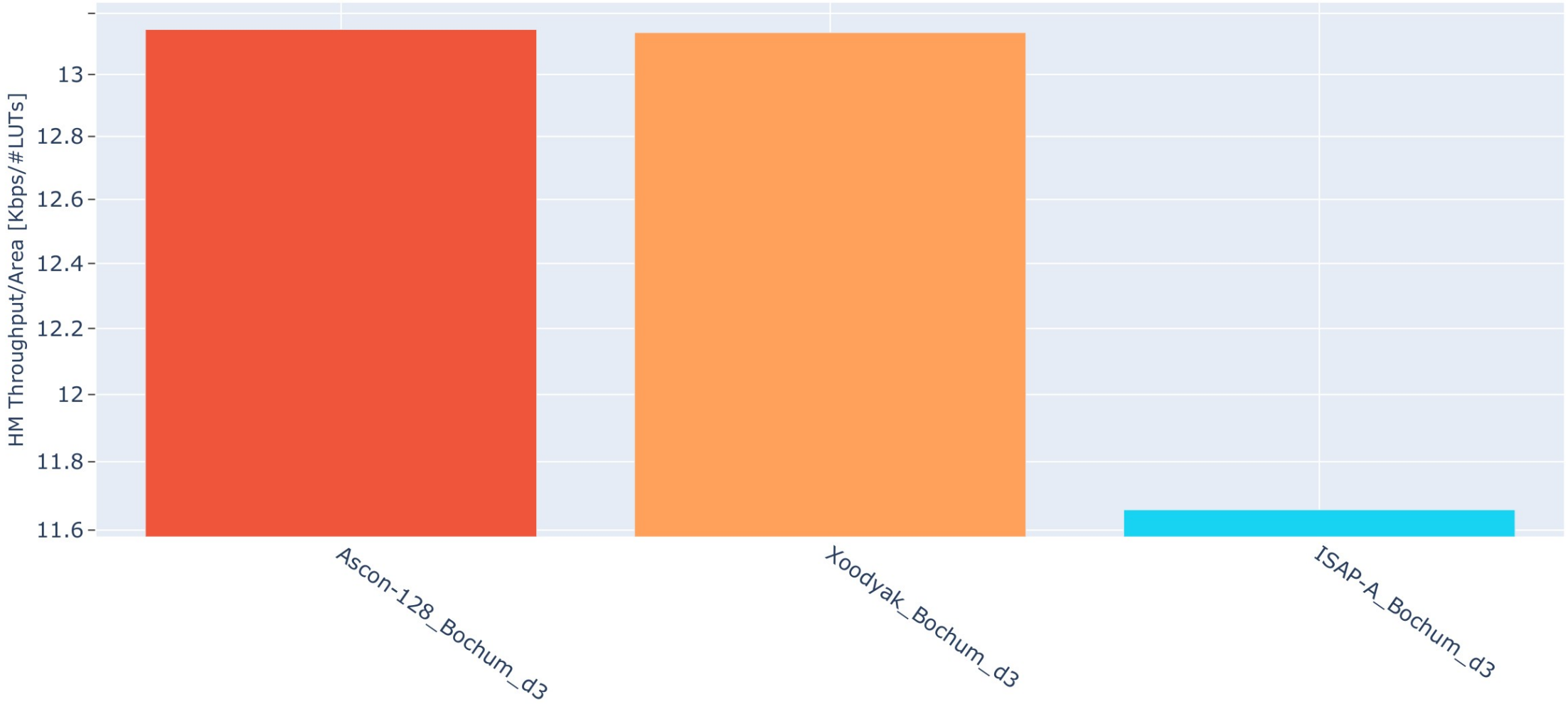
Hashing Throughput over Area for Long Messages (1st Order Protected)



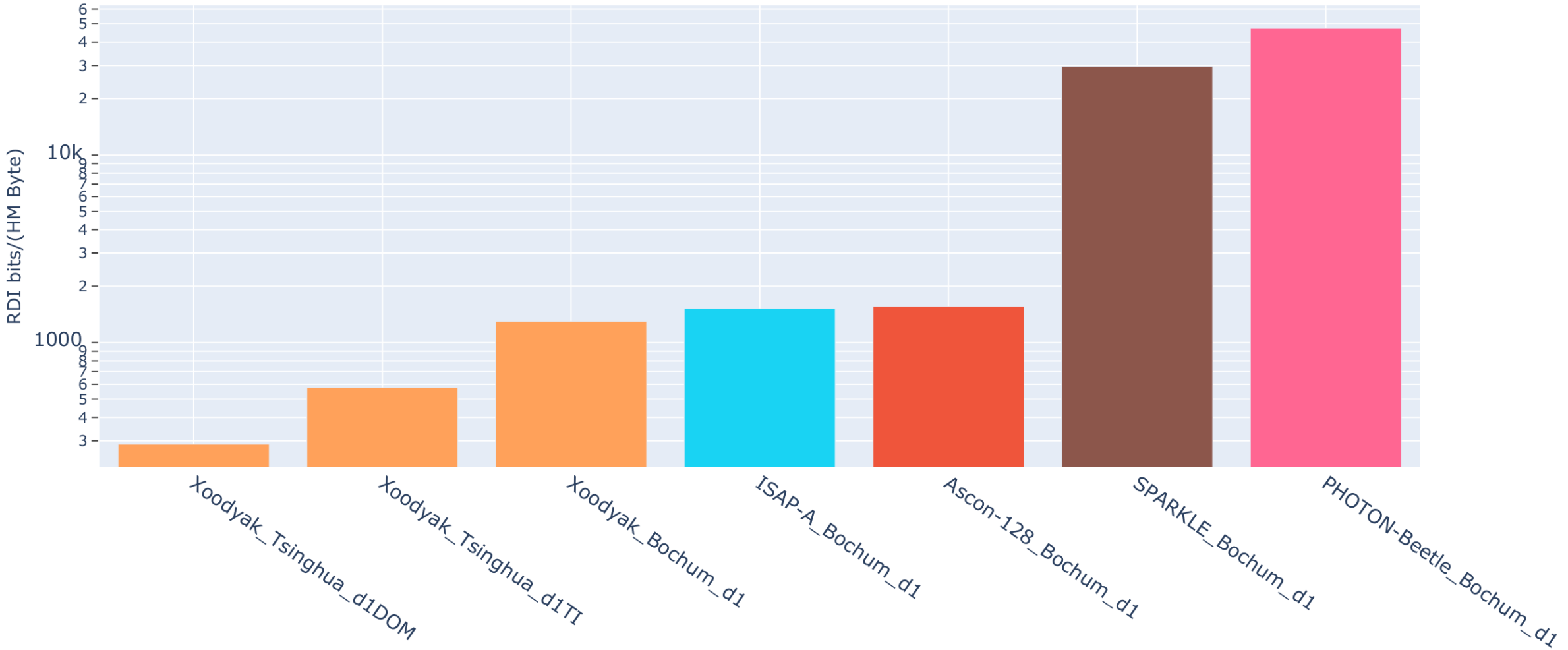
Hashing Throughput over Area for Long Messages (2nd Order Protected)



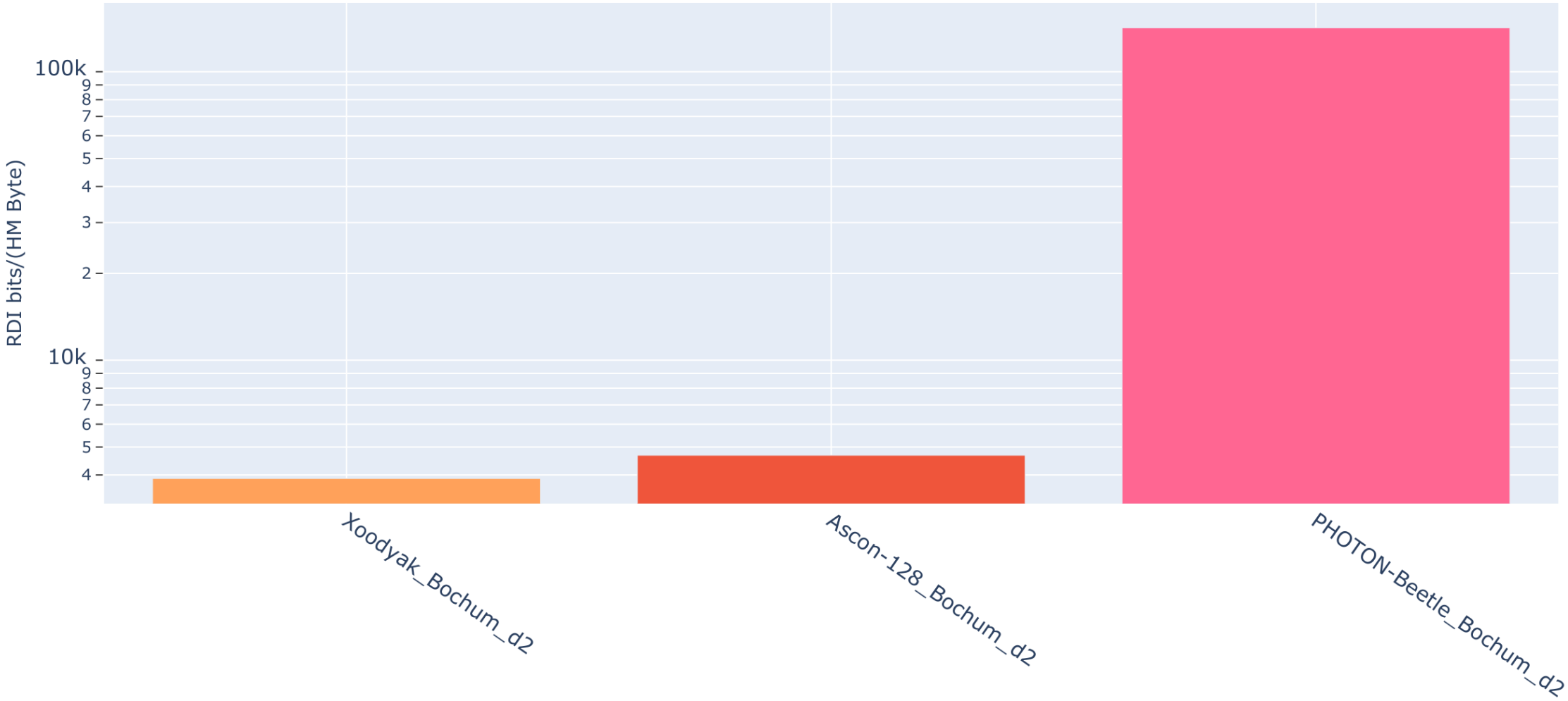
Hashing Throughput over Area for Long Messages (3rd Order Protected)



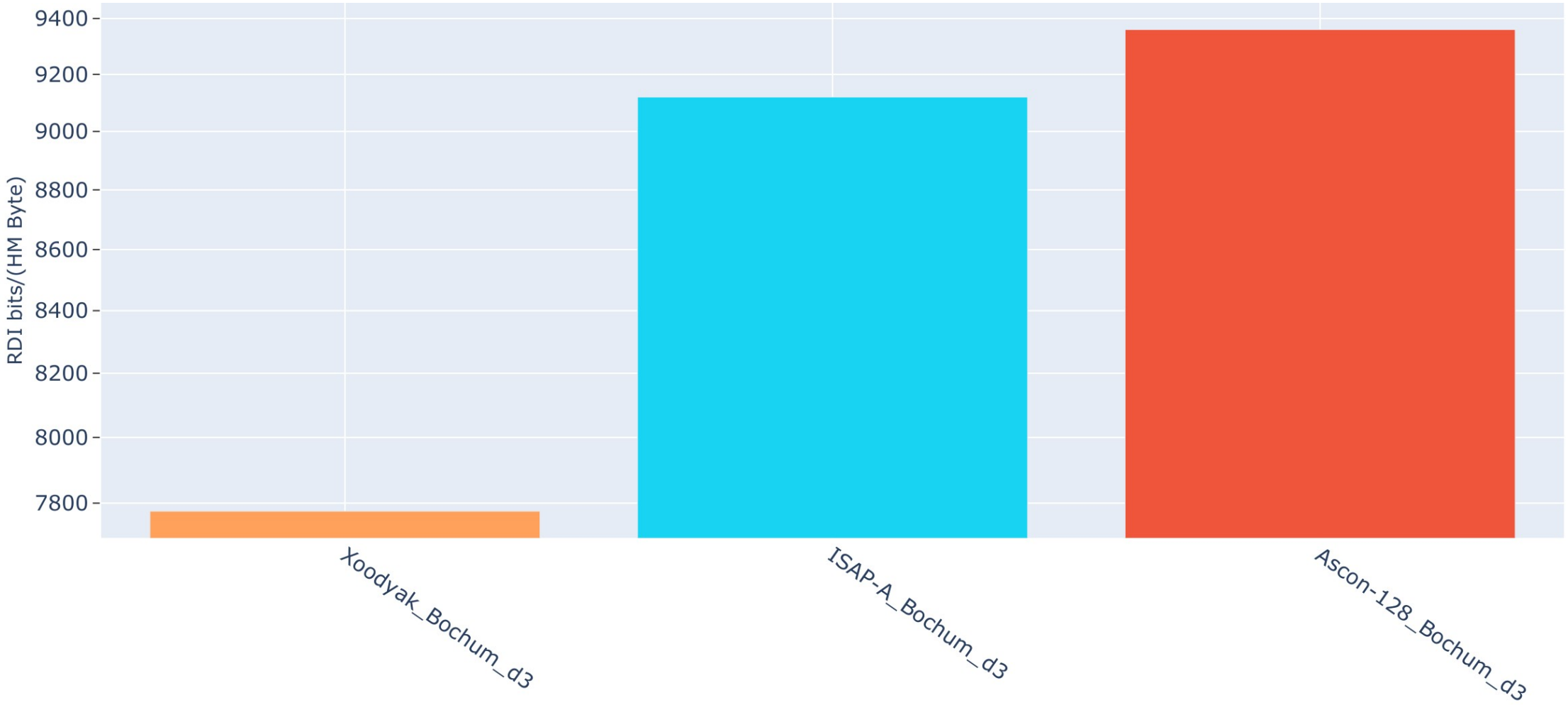
Random Bits per Hash Message Byte (1st Order Protected)



Random Bits per Hash Message Byte (2nd Order Protected)



Random Bits per Hash Message Byte (3rd Order Protected)



Overview

- Introduction
- Side-Channel Security Evaluation Labs
- Protected Hardware Implementations
- Protected Software Implementations
- SCA Evaluation Results of Hardware Implementations
- SCA Evaluation Results of Software Implementations
- Benchmarking of Hardware Implementations
- Conclusions

Conclusions

- Hardware benchmarking results demonstrate advantages of the following candidates:
- Ascon and Xoodyak:
  - High speed
  - High throughput/area ratio
  - Moderate randomness requirements
  - Support for hashing
- TinyJAMBU:
  - Low area
  - High throughput/area ratio
  - Moderate randomness requirements
- ISAP:
  - Mode-level protection against arbitrary-level DPA (no masking)
  - High throughput/area ratio among protected designs
  - Support for hashing

Website

- Lightweight Cryptography in Hardware and Embedded Systems
  https://cryptography.gmu.edu/athena/index.php?id=LWC
- Evaluation of Finalists in the NIST LWC Process
  - Full Report (also on e-print)
  - Summary of Results
  - Assignments, Commitments, and Reports of all Labs
  - Side-Channel Security Evaluation Labs Specifications
  - Protected Hardware and Software Implementations
  - Calls for Implementations & Labs
  - Documentation of the Hardware API
  - Code (Development Package) for the Hardware API
  - Unprotected Hardware Implementations