High-Speed Hardware Architectures and Fair FPGA Benchmarking of CRYSTALS-Kyber, NTRU, and Saber

Viet Ba Dang, Kamyar Mohajerani, and Kris Gaj

George Mason University
USA

Co-Authors & Primary Designers

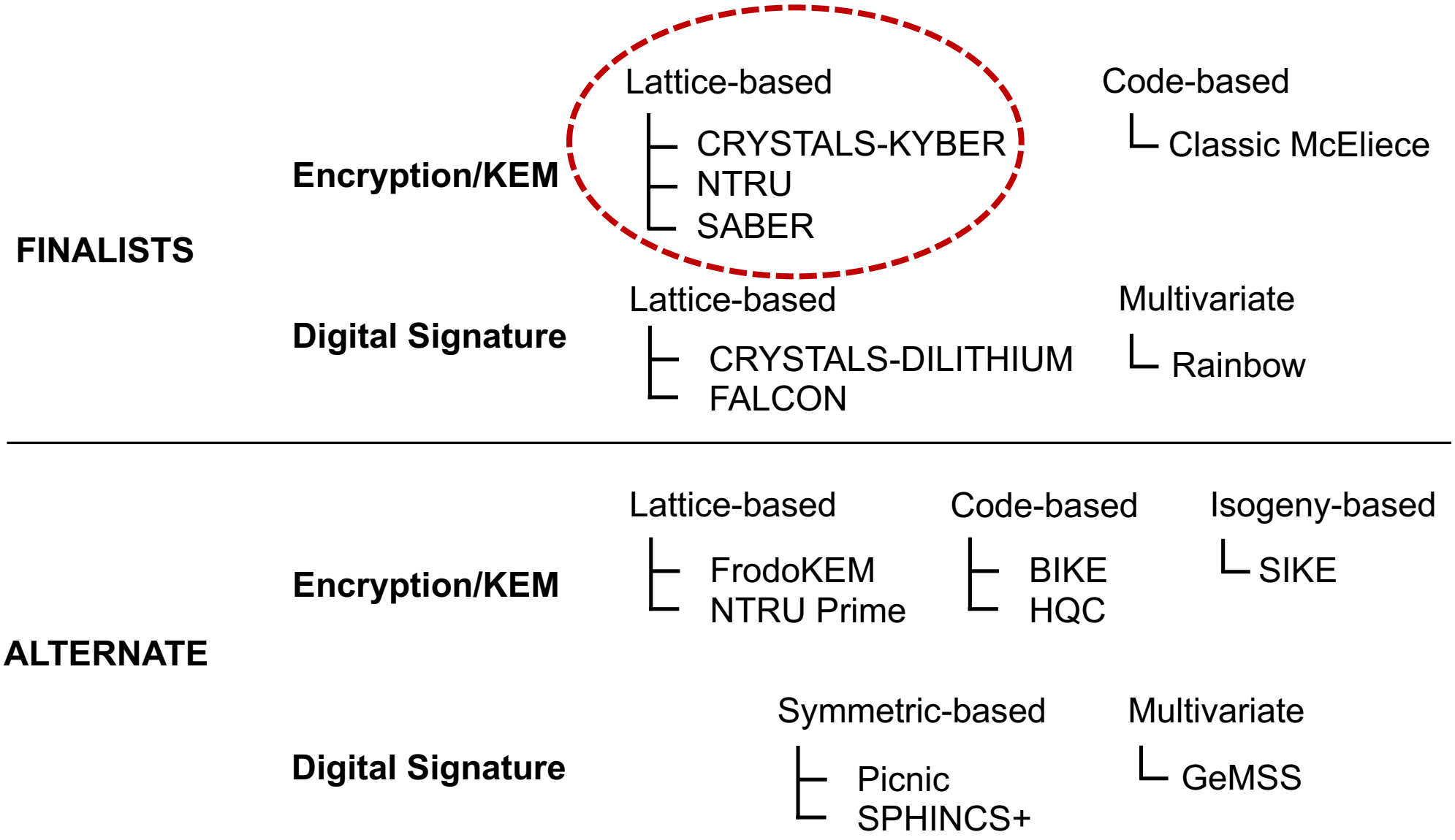
Viet
Ba Dang

NTRU, Saber

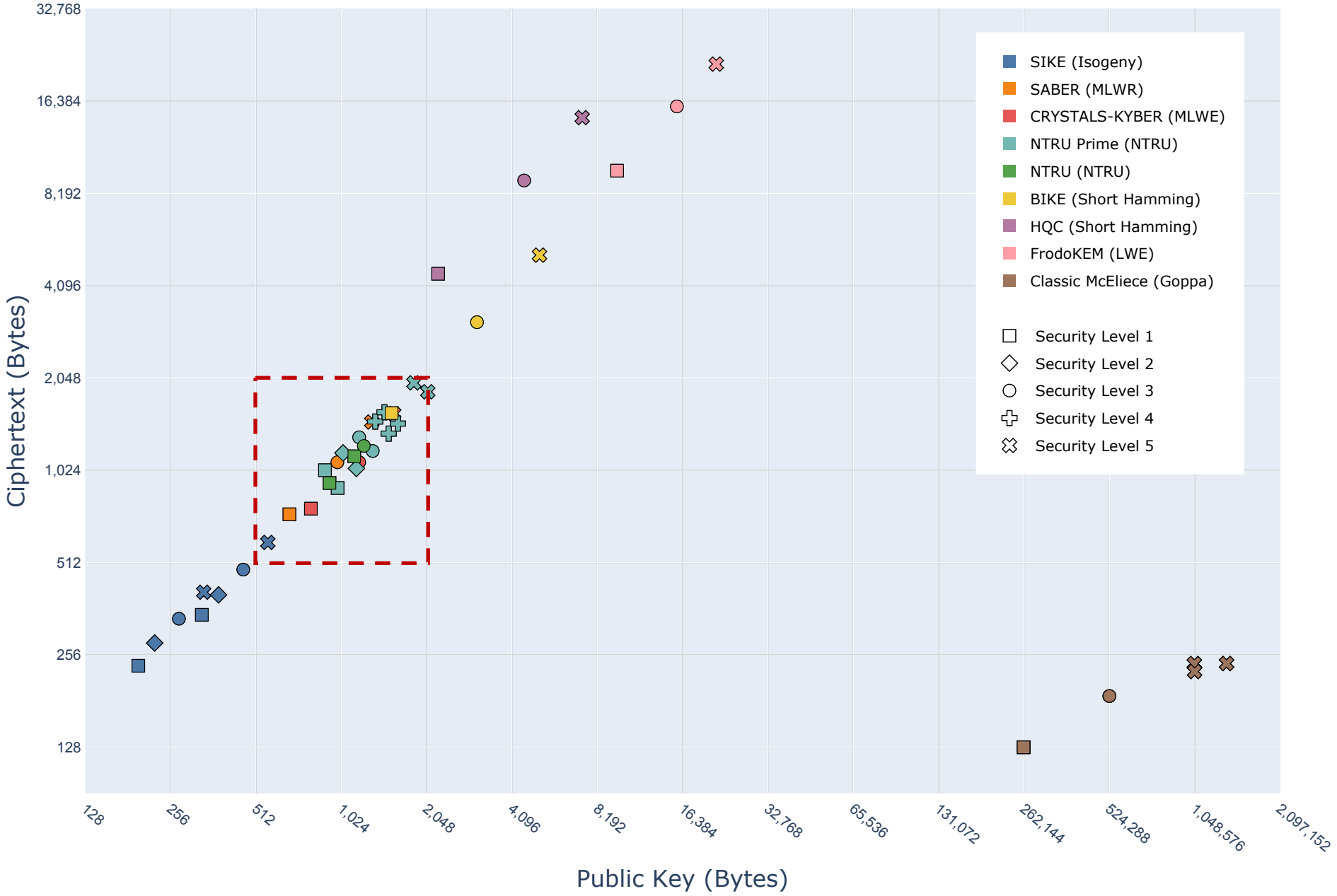
Kamyar
Mohajerani

CRYSTALS-Kyber

Round 3 Candidates & Focus of This Project



Public-Key and Ciphertext Sizes



Our Major Contributions

- NTRU-HRSS & NTRU-HPS:
  First pure hardware implementation

- Saber:
  The most-efficient hardware implementation in terms of both speed and area

- CRYSTALS-Kyber:
  The most-efficient high-speed implementation in terms of speed and latency x area

- All 9 Round-3 KEMs:
  Ranking and speed vs. area graphs
  based on the most-efficient hardware implementations

Benchmarking Methodology

Design Approach

Register-Transfer Level (RTL)

- HW: VHDL, Verilog, Chisel
- Industry standard
- Highest-performance
- Best trade-offs between speed vs. area
- Long development time

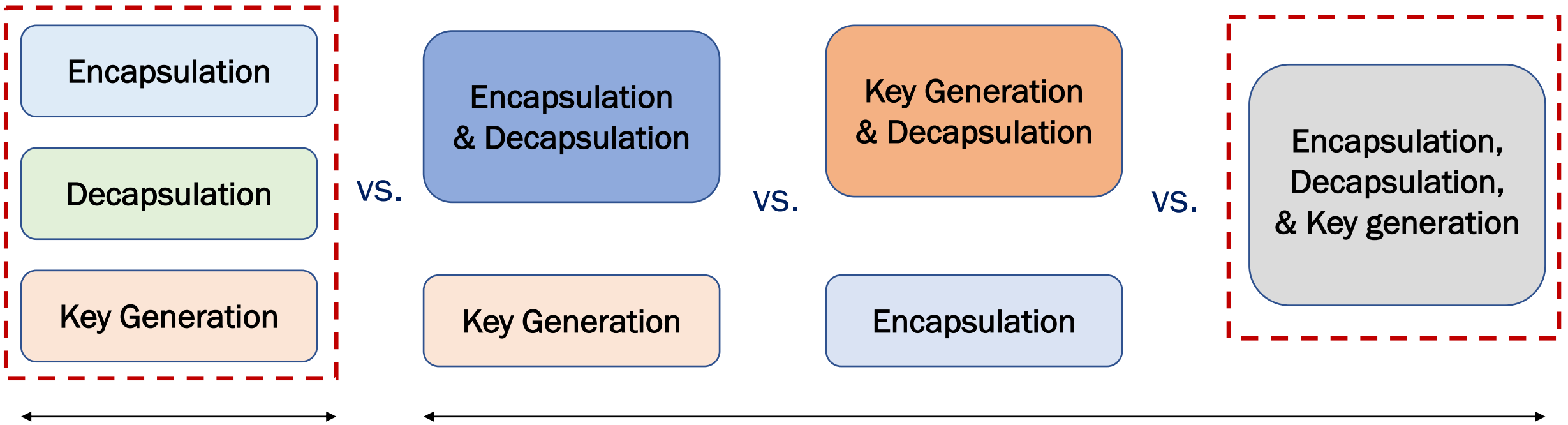
High-Level Synthesis (HLS)

- HW: C, C++, System C
- Short development time
- Lower performance in terms of speed and/or area (for PQC, some reports showing 2-4 orders of magnitude difference)

Software/Hardware Co-Design (SW/HW)

- SW: C, assembly
- HW RTL: VHDL, Verilog, Chisel
- HW HLS: C, C++, System C
- Short development time
- Communication overhead
- Strong dependence on a partitioning scheme
- Inconclusive results

Operations Supported by Each Core

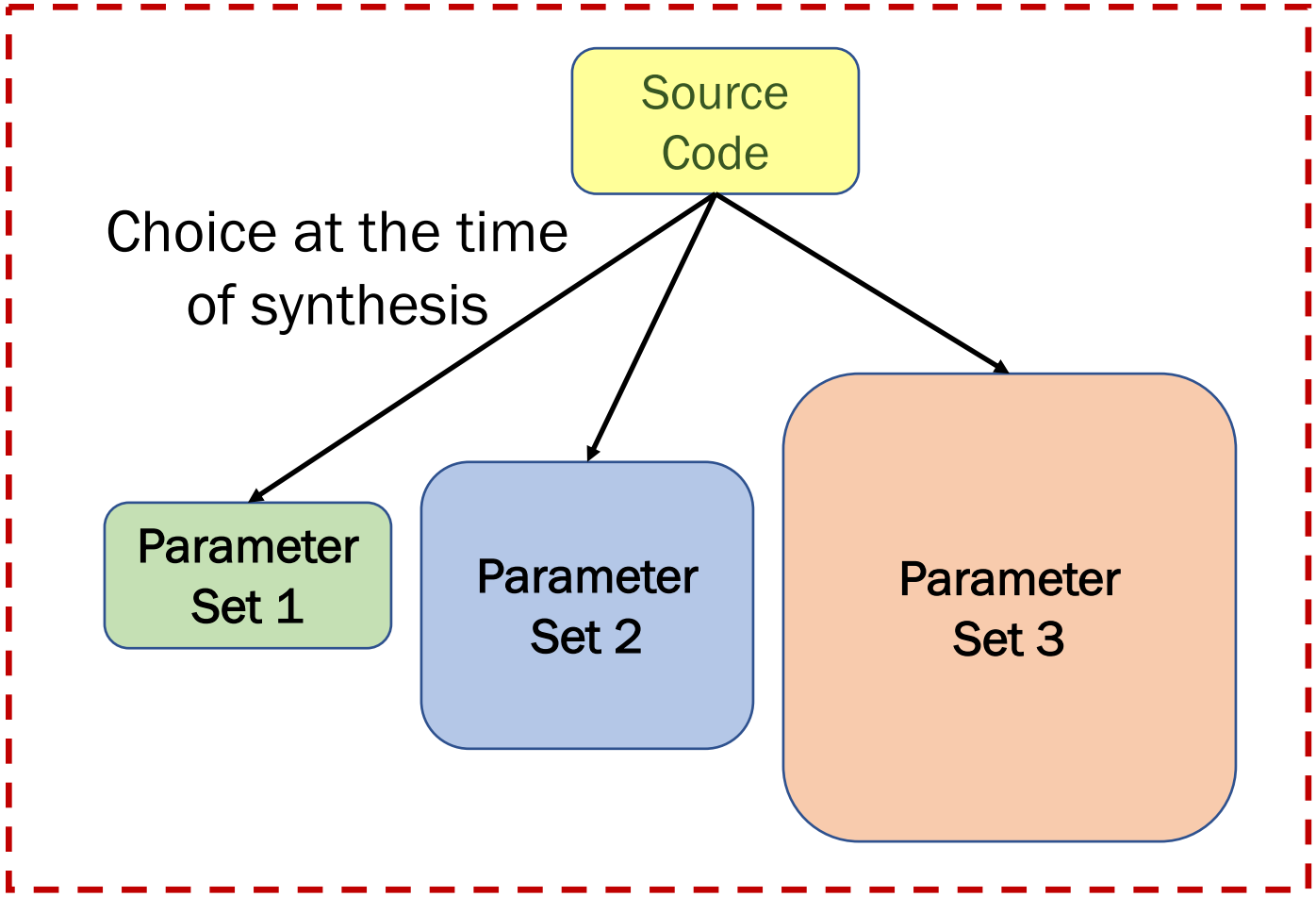


Demonstrate Area of Each Operation

Demonstrate potential for resource sharing

Each core can operate with its own maximum clock frequency

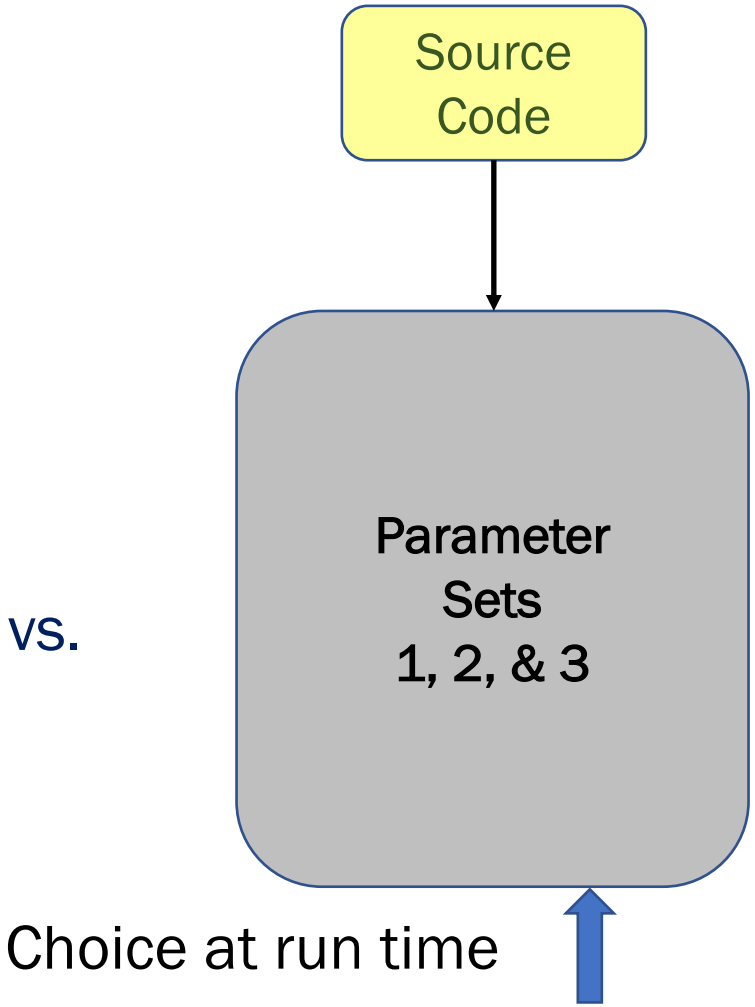
Parameter Sets Supported by Each Core



Multiple result sets with minimal effort

3 areas, 3 clock frequencies

vs.



1 area, 1 clock frequency

Design Space Exploration

Lightweight

Primary Metrics:

- Area
- Power

Secondary Metrics:

- Latency
- #Operations_per_s

Balanced

Primary Metrics:

- Latency · Area
- #Operations_per_s / Area

High-speed

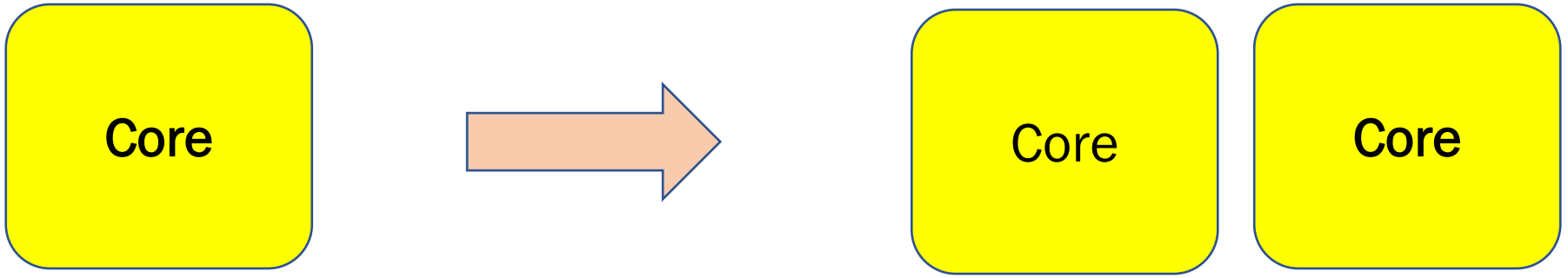
Primary Metrics:

- Latency
- #Operations_per_s

Secondary Metrics:

- Area
- Power

Latency vs. #Operations per seconds



#Operations per second: x 2 (doubles)
Latency: x 1 (stays the same)

Application with strict latency requirements:

Establishing secure communication between two autonomous vehicles

Application with strict #operations per second requirements:

High-traffic servers for popular internet security protocols

FPGA Platforms & Tools

Platforms:

Artix-7:	XC7A200T-3,	28 nm technology
134,600 LUTs	365 BRAMs	740 DSPs
Zynq UltraScale+:	ZU7EV-3,	16 nm technology
230,400 LUTs	312 BRAMs	1,728 DSPs

Tools:

Vivado WebPack 2020.1 (free)

In PQC, the use of LUTs typically most limiting ⇒ Area represented by #LUTs

All results reported after placing & routing

General Approach

- All designs started about 1 year ago
- Two designers working very closely with each other
- All design decisions made independently from other groups
- No use of any vendor-specific intellectual property (IP) cores
- No use of any code developed by other groups
- Portability to other FPGA platforms and ASICs
- Freedom to use any design flows (including open-source)
- Complete documentation
- Publishing source-code after a publication in a journal or a conference with proceedings

Design Choices

Choice of a Polynomial Multiplier

CRYSTALS-Kyber

k x NTT-based

k= 2, 3, 4
for Security Levels 1, 3, 5

+ Karatsuba
during pointwise multiplication

NTRU

Toom-Cook

Toom-3 + Karatsuba

Based on 15·d DSP units

d= 2, 3

Schoolbook

when one polynomial ternary, i.e., w/ coefficients {-1, 0, 1}

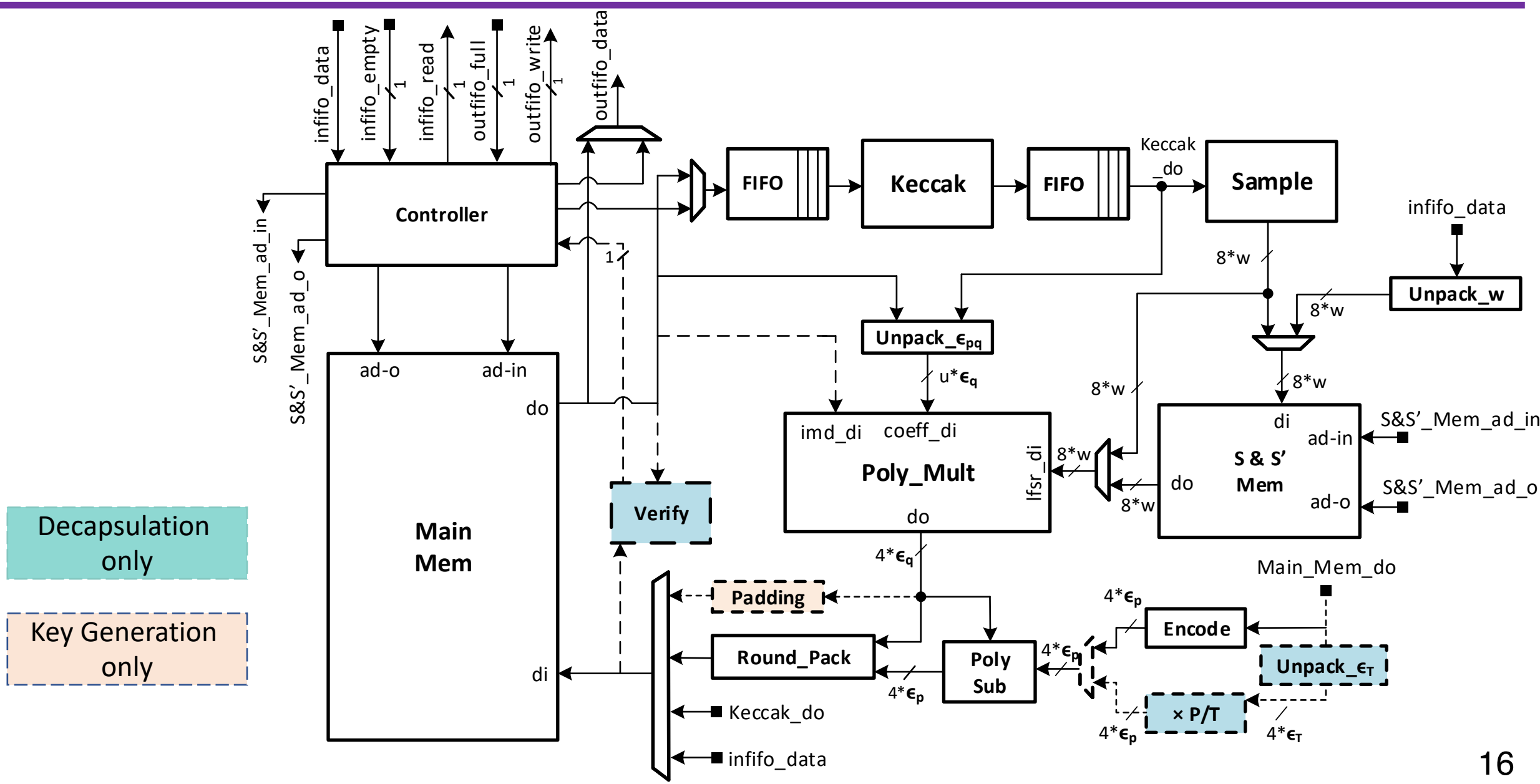
Saber

Schoolbook

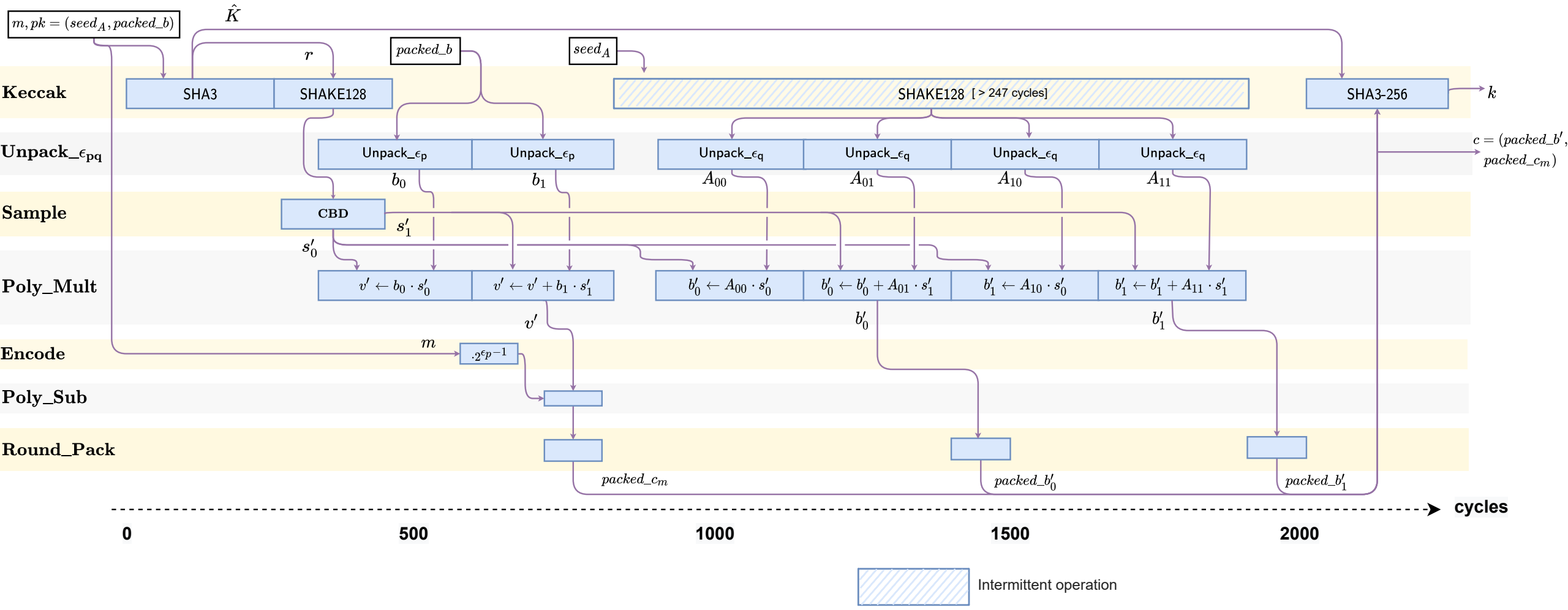
u – unrolling factor
(#coefficients of B multiplied by A)

u= 1, 2, 4

Saber: Block Diagram

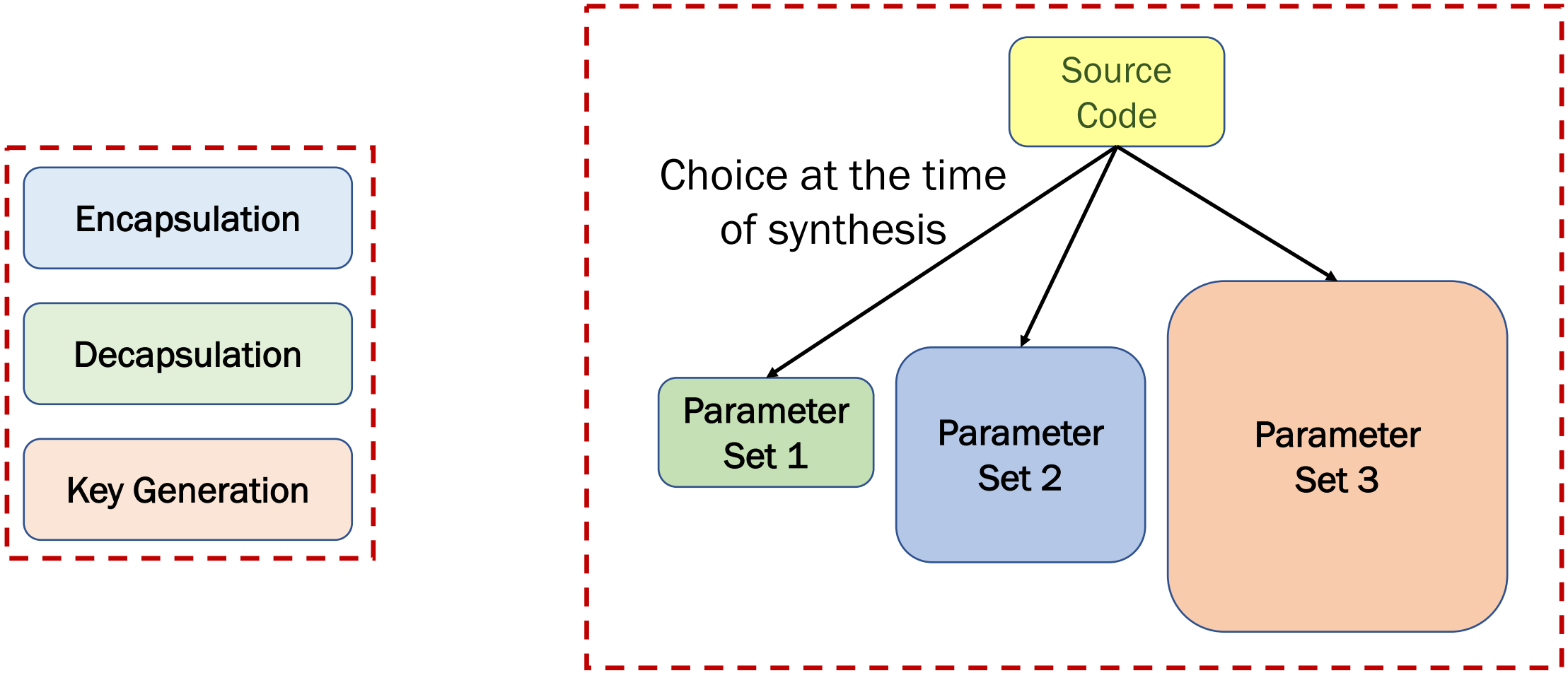


Saber Encapsulation: Scheduling Diagram



Results

Ranking of Kyber, NTRU, & Saber: Assumptions



- 1 operation and 1 security level supported by each core
- 9 cores per algorithm

Ranking of 3 KEM Finalists on Artix-7

Key Generation								
Level 1			*Level 3*			*Level 5*		
Algorithm	**Time [us]**	**Ratio**	**Algorithm**	**Time [us]**	**Ratio**	**Algorithm**	**Time [us]**	**Ratio**
Saber	9.5	1.00	Kyber	12.0	1.00	Kyber	16.2	1.00
Kyber	10.0	1.05	Saber	15.9	1.33	Saber	28.8	1.78
NTRU-HRSS	323.8	34.08	NTRU-HPS	516.6	43.05			
NTRU-HPS	370.6	39.01						

Encapsulation								
Level 1			*Level 3*			*Level 5*		
Algorithm	**Time [us]**	**Ratio**	**Algorithm**	**Time [us]**	**Ratio**	**Algorithm**	**Time [us]**	**Ratio**
Saber	12.7	1.00	Kyber	17.0	1.00	Kyber	21.7	1.00
NTRU-HRSS	13.9	1.09	Saber	22.0	1.29	Saber	34.5	1.59
Kyber	14.7	1.16	NTRU-HPS	35.2	2.07			
NTRU-HPS	28.4	2.24						

Decapsulation								
Level 1			*Level 3*			*Level 5*		
Algorithm	**Time [us]**	**Ratio**	**Algorithm**	**Time [us]**	**Ratio**	**Algorithm**	**Time [us]**	**Ratio**
Saber	16.4	1.00	Kyber	22.2	1.00	Kyber	26.4	1.00
Kyber	20.5	1.25	Saber	27.5	1.24	Saber	41.9	1.59
NTRU-HPS	47.0	2.87	NTRU-HPS	63.8	2.87			
NTRU-HRSS	55.2	3.37						

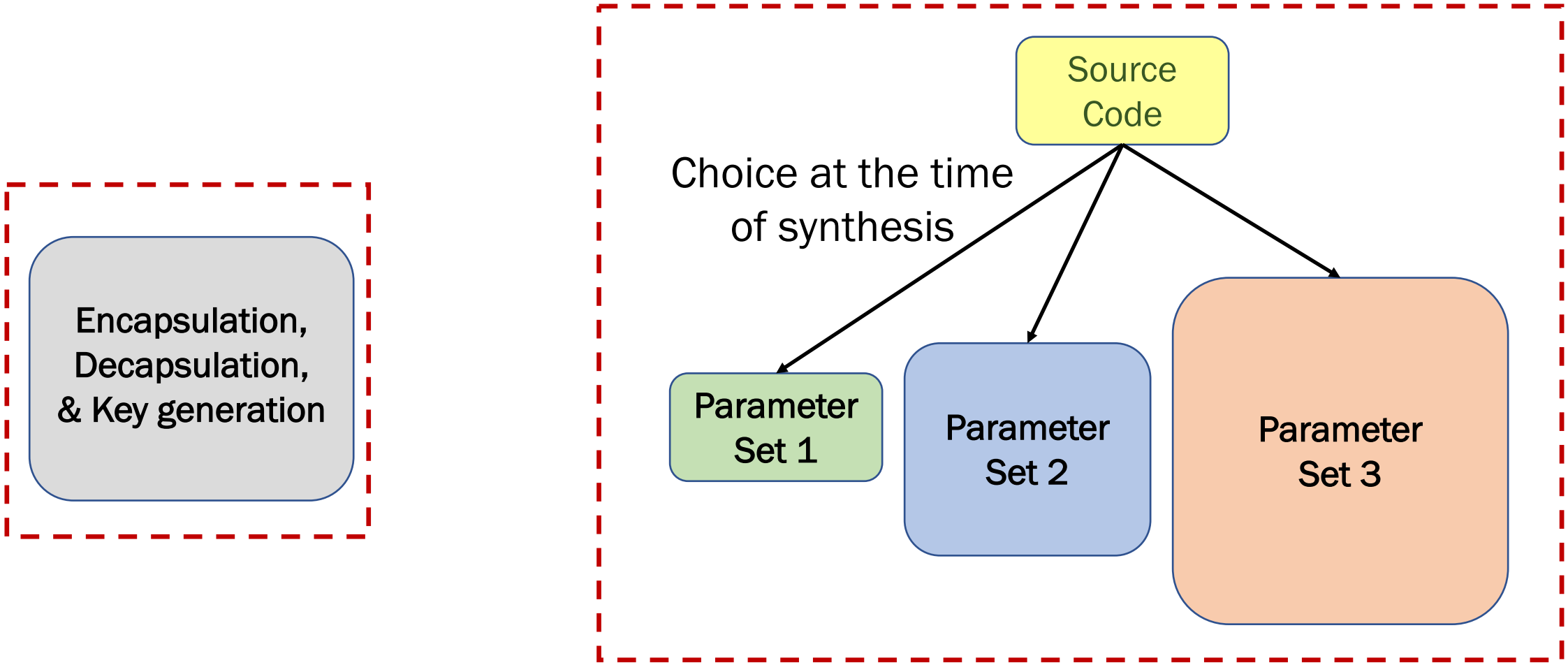
Why is Kyber Better Than Saber at Levels 3 & 5?

- In Kyber, the NTT-based multiplier is quite small and sequential
- In Saber, the schoolbook multiplier is big and parallel
- For Kyber, it is justifiable to use 2, 3, and 4 multipliers

  at the security levels 1, 3, and 5, respectively
- For Saber, it is not justifiable to use a bigger unrolling factor for

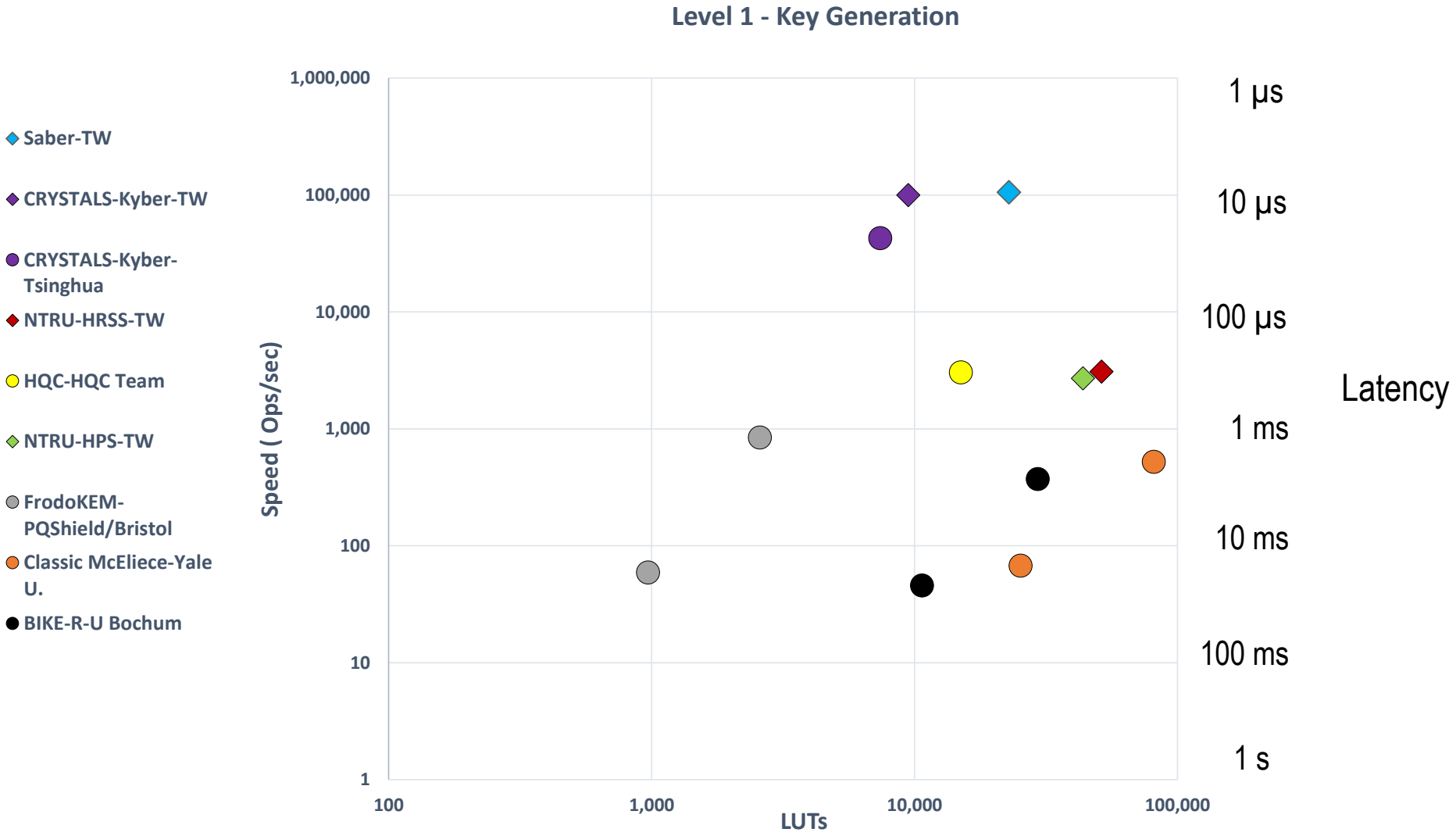
  higher security levels

Latency vs. Area: Assumptions

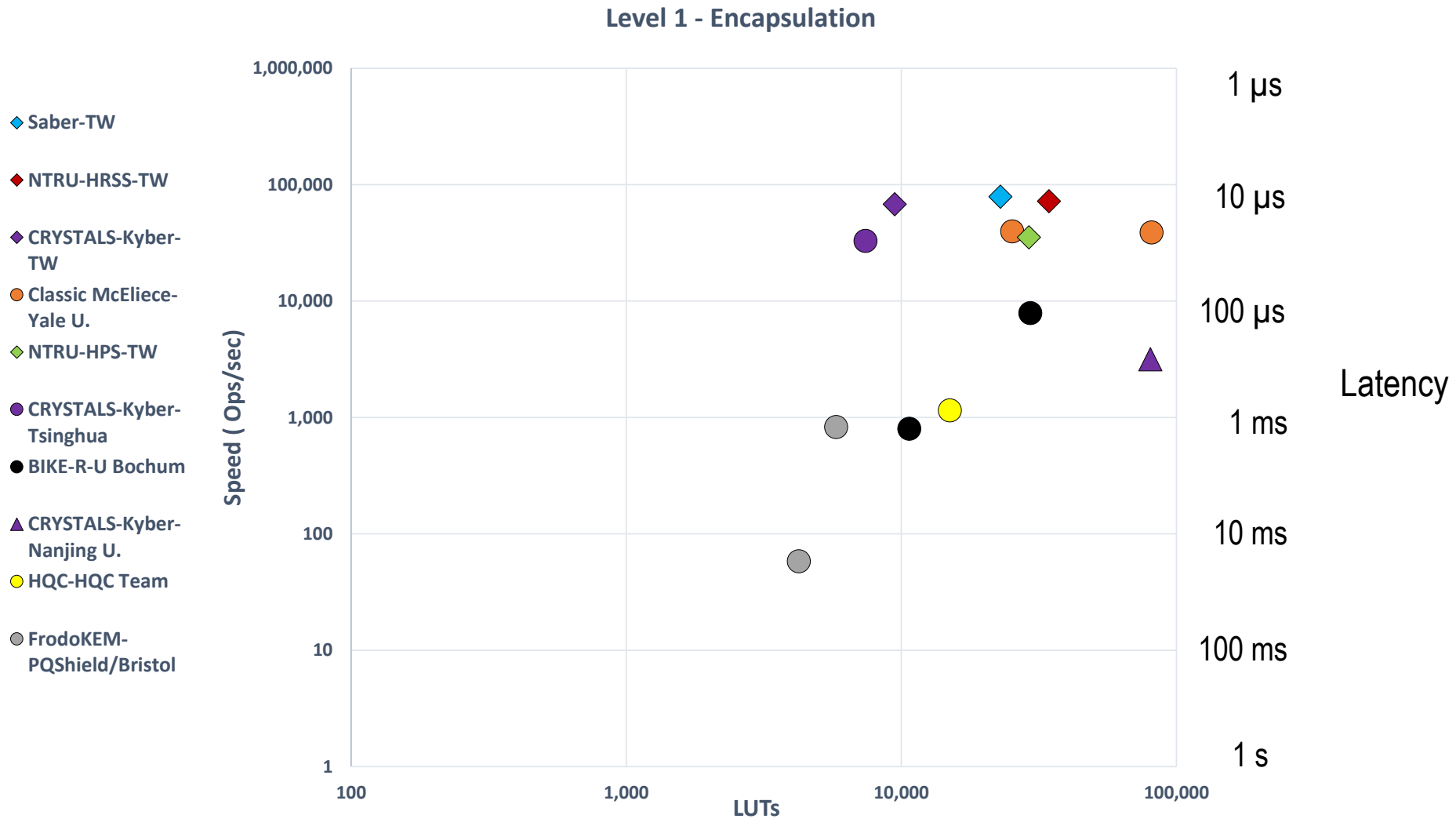


- 3 operations and 1 security level supported by each core
- 3 cores per algorithm

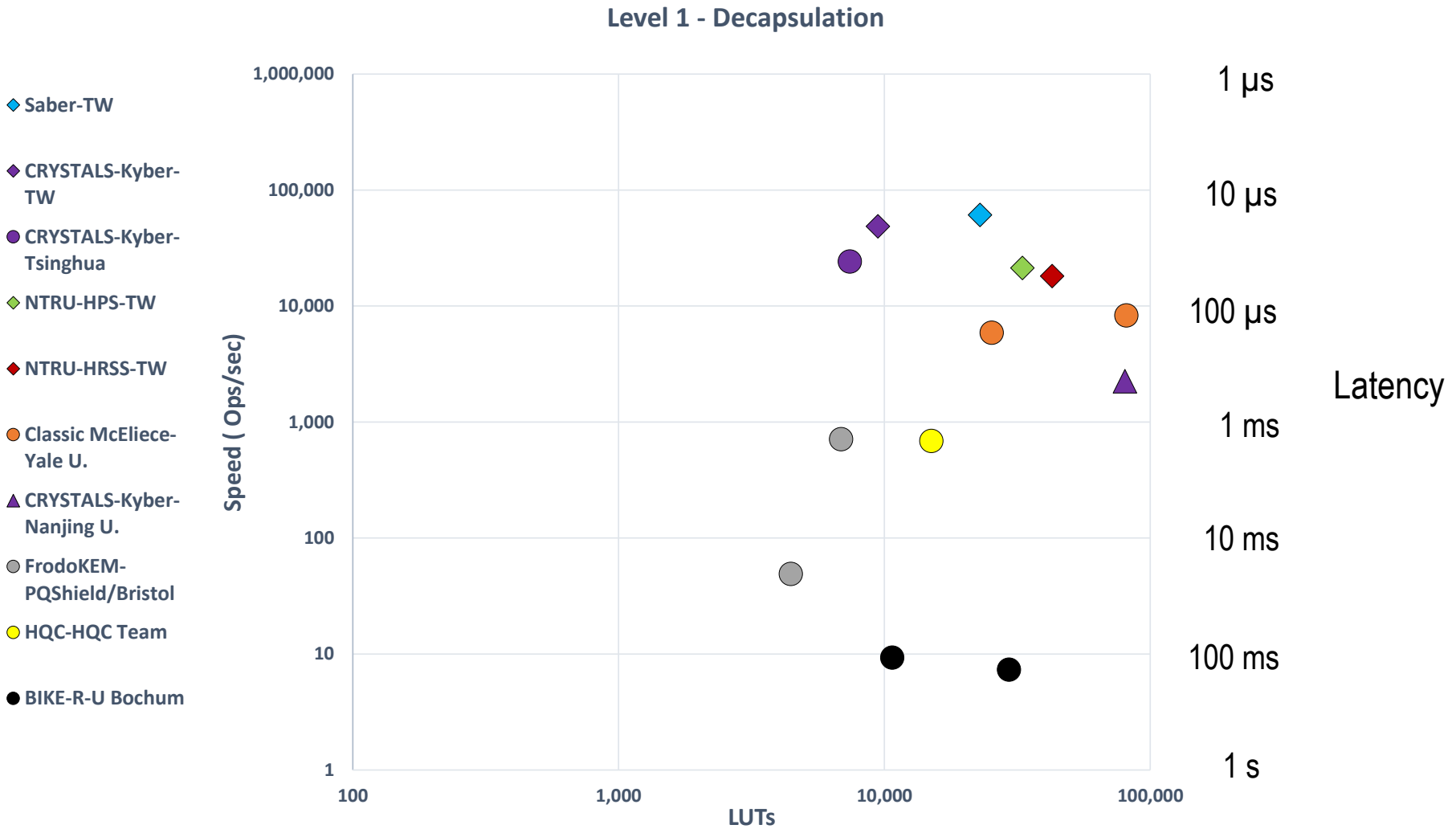
Level 1: Key Generation on Artix-7



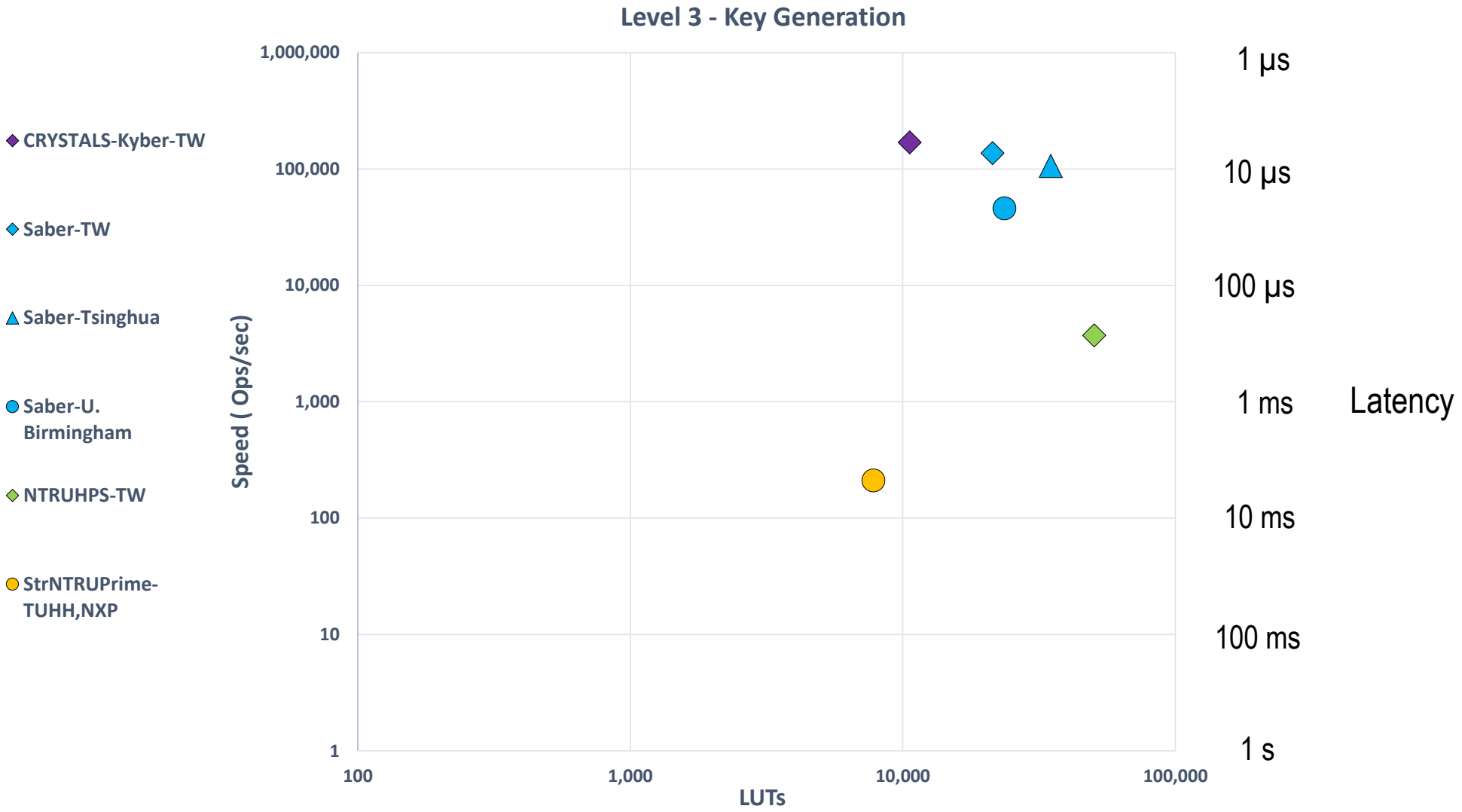
Level 1: Encapsulation on Artix-7



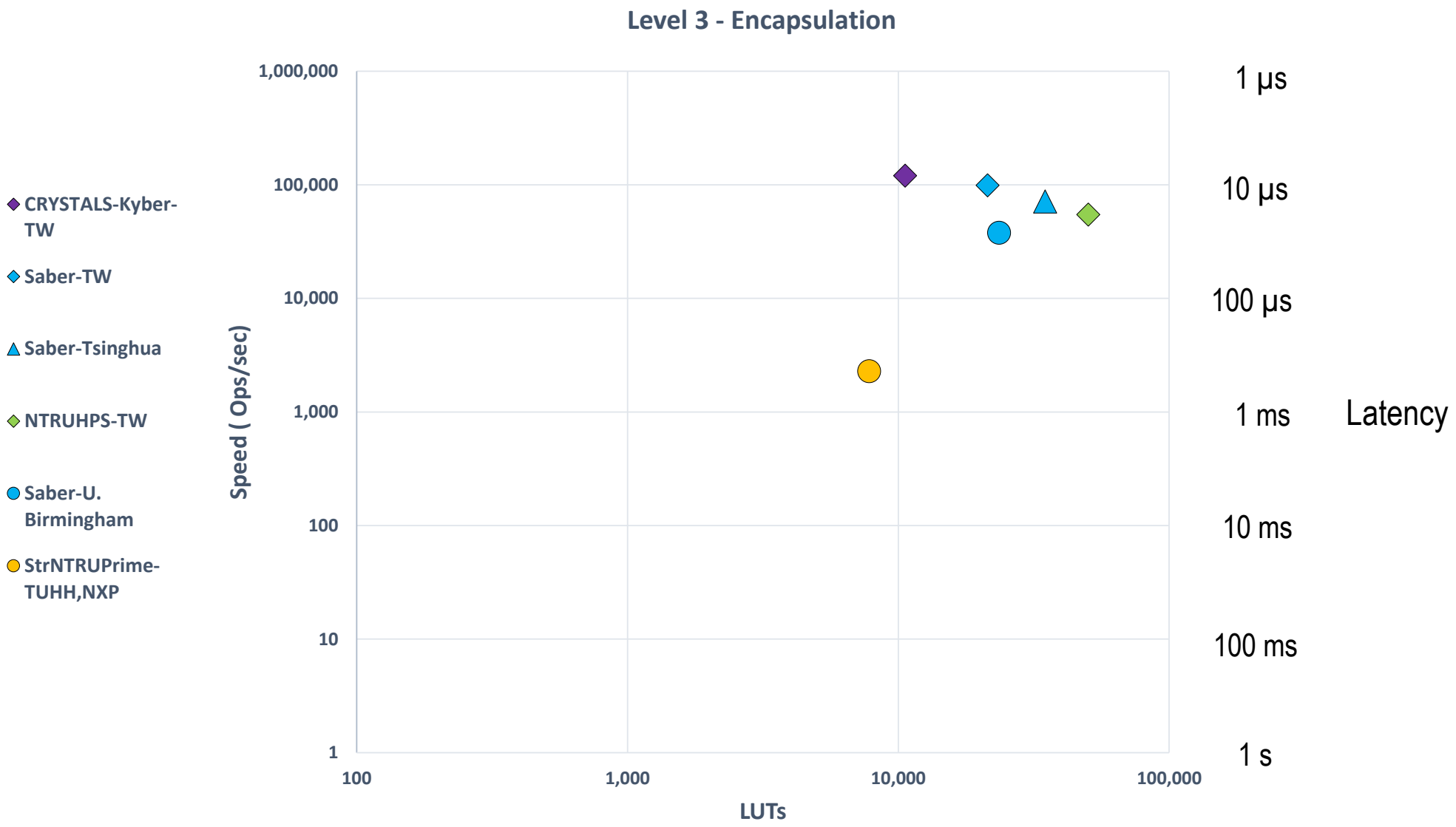
Level 1: Decapsulation on Artix-7



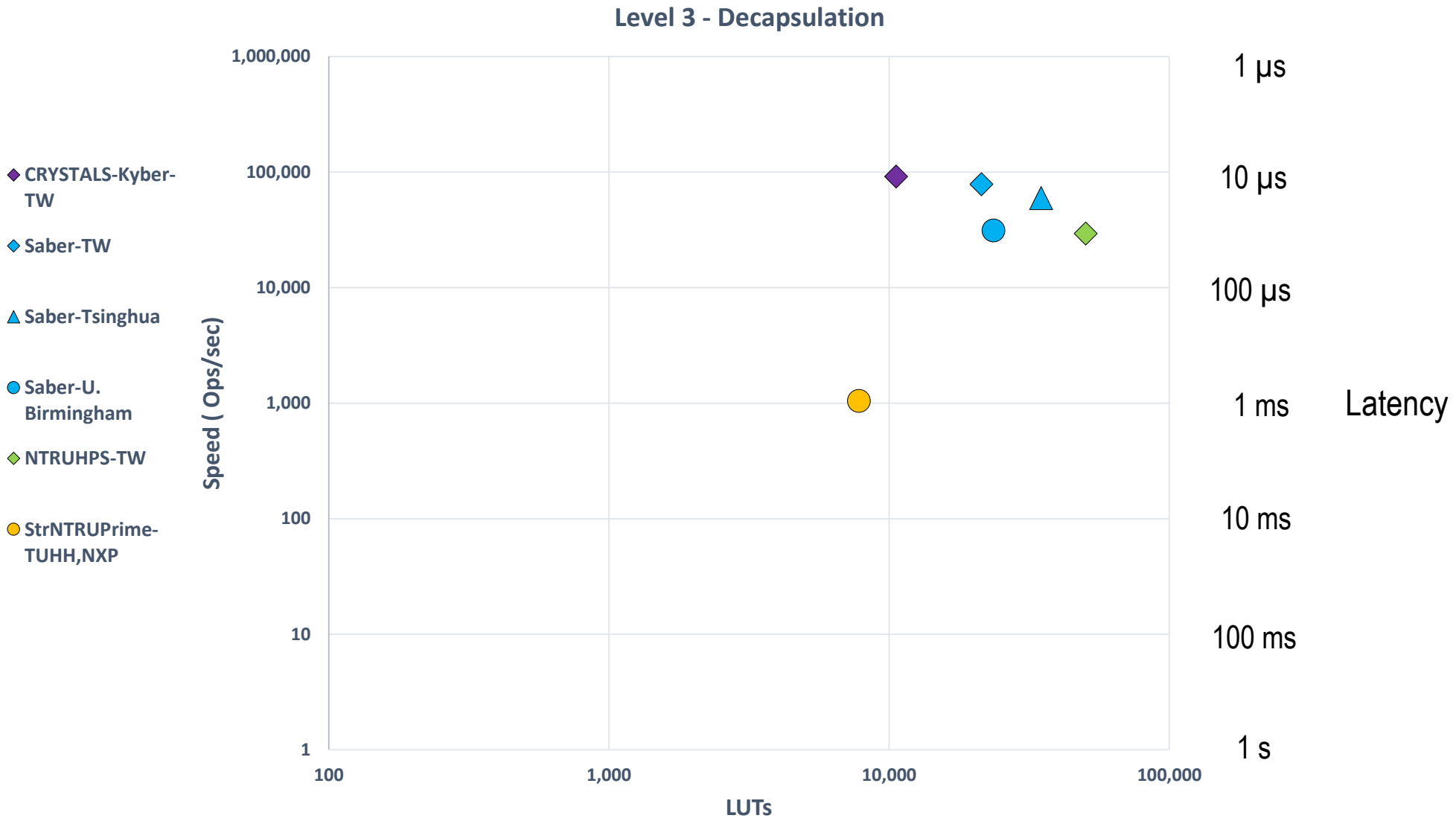
Level 3: Key Generation on Zynq UltraScale+



Level 3: Encapsulation on Zynq UltraScale+



Level 3: Decapsulation on Zynq UltraScale+



Comparison with Previous Work

Previous Work on CRYSTALS-Kyber

CRYSTALS-Kyber-Tsinghua:

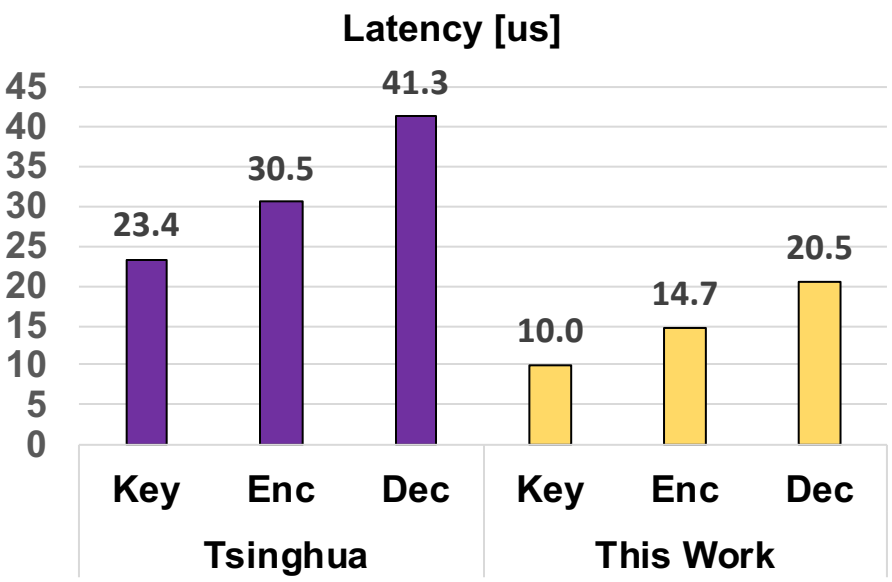
"A Compact Hardware Implementation of CCA-Secure Key Exchange Mechanism CRYSTALS-KYBER on FPGA"

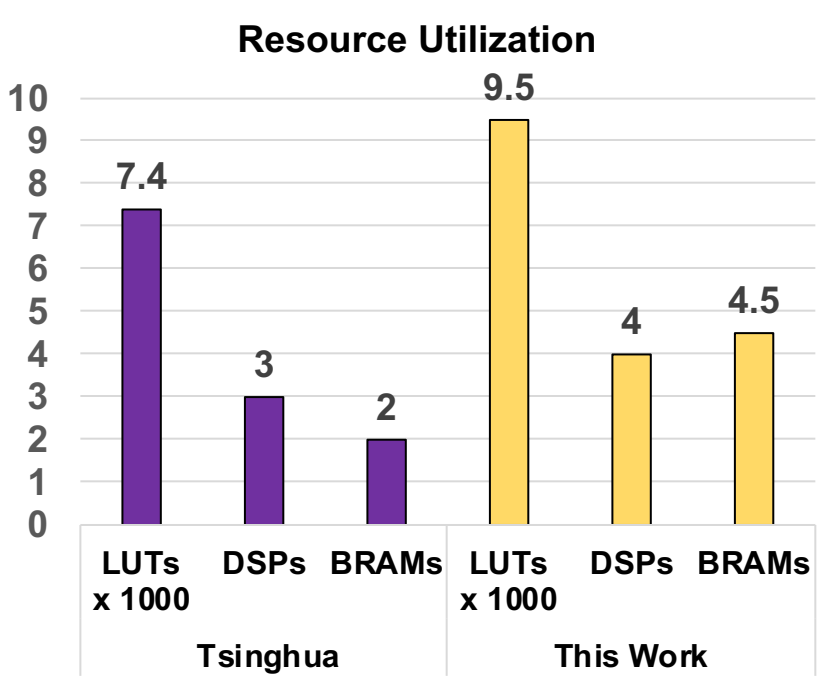
by Yufei Xing and Shuguo Li

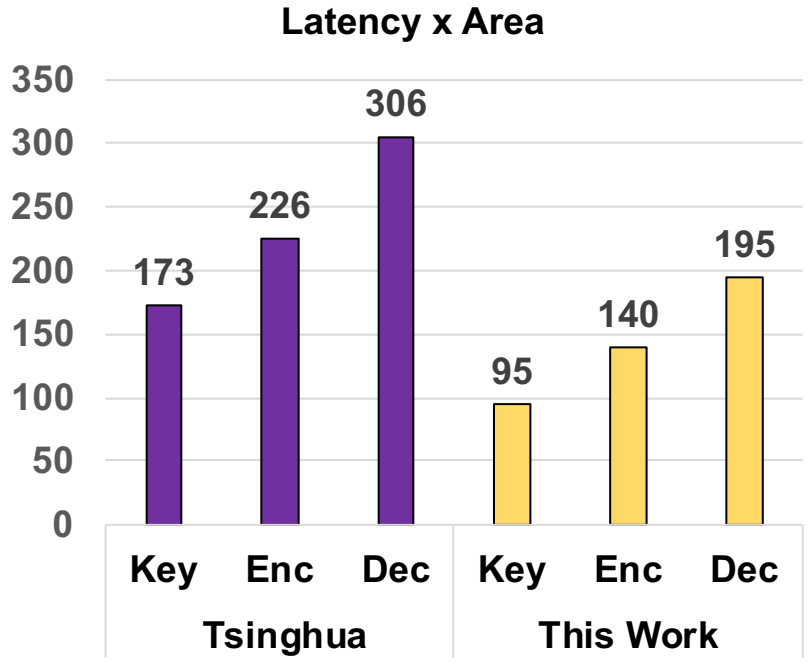
Institute of Microelectronics, Tsinghua University, Beijing, China

TCHES, vol. 2021, no. 2, Feb. 2021

CRYSTALS-Kyber: Comparison with Previous Work







Notation: Key : Key Generation Enc : Encapsulation Dec : Decapsulation

Assumptions: Security Level 1 Artix-7 Area represented by LUTs x 1000

Improvement in Latency: 2.3 (Key), 2.1 (Enc), 2.0 (Dec)

Improvement in Latency x Area: 1.8 (Key), 1.6 (Enc), 1.6 (Dec)

Previous Work on Saber

Saber-U.Birmingham:

"High-speed Instruction-set Coprocessor for Lattice-based Key Encapsulation Mechanism: Saber in Hardware,"

Sujoy Sinha Roy and Andrea Basso

University of Birmingham, UK

TCHES, vol. 2020, no. 4, Aug. 2020

Saber-Tsinghua:

"LWRpro: An Energy-Efficient Configurable Crypto-Processor for Module-LWR,"

by Yihong Zhu[1], Min Zhu[2], Bohan Yang[1], Wenping Zhu[1], Chenchen Deng[1],

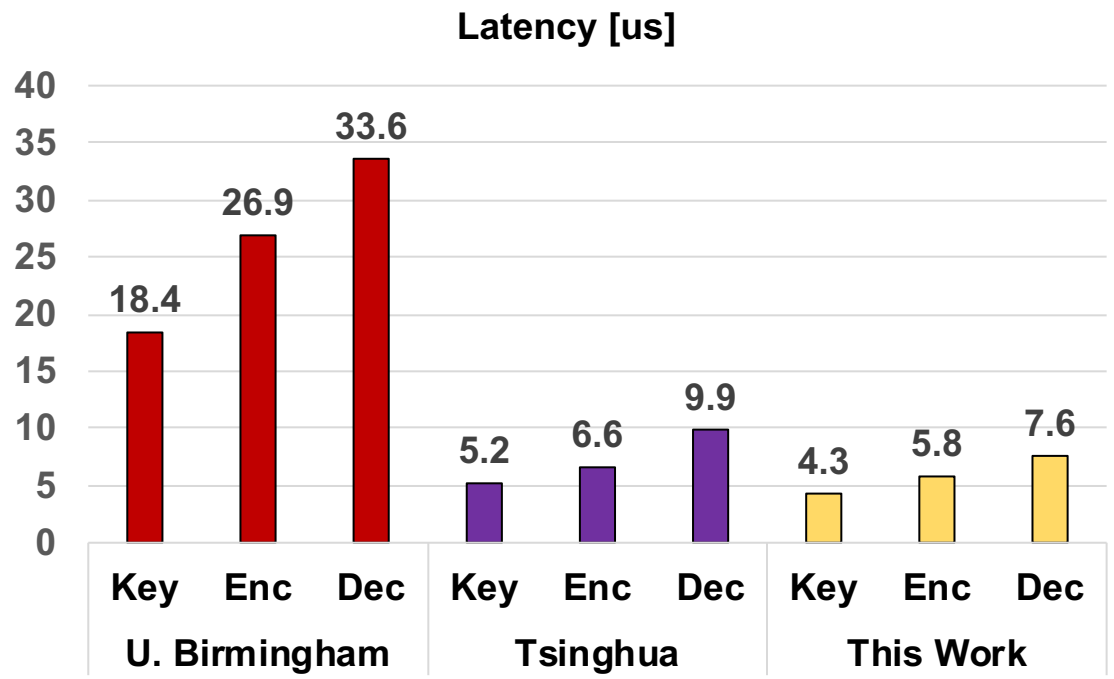
Chen Chen[1], Shaojun Wei[1], and Leibo Liu[1]

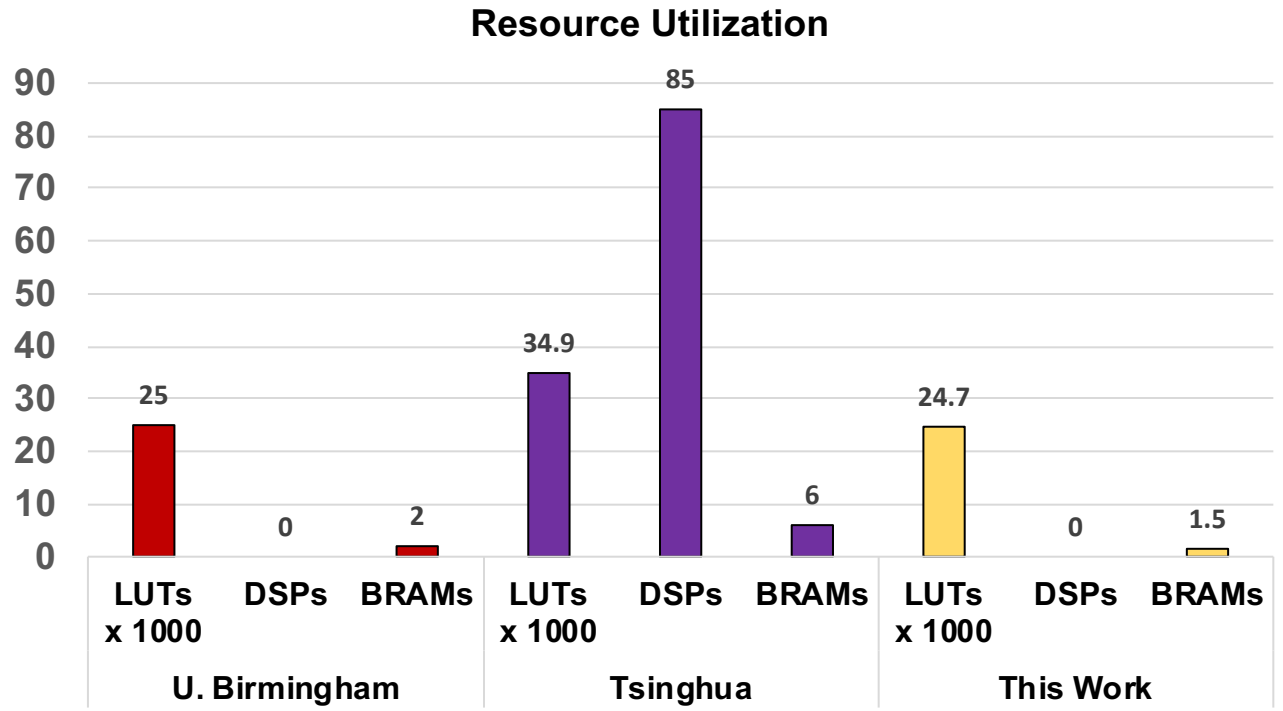
[1] Tsinghua University, Beijing, China

[2] Wuxi Micro Innovation Integrated Circuit Design Company Ltd., Wuxi, China

IEEE Transactions on Circuits and Systems-I: Regular Papers, vol. 68, no. 3, Mar. 2021

Saber: Comparison with Previous Work





Notation: Key : Key Generation Enc : Encapsulation Dec : Decapsulation

Assumptions: Security Level 1 Artix-7 Area represented by LUTs x 1000

Improvement in Latency over U. Birmingham/Tsinghua: 4.3/1.2 (K), 4.6/1.1 (E), 4.4/1.3 (D)

Improvement in Area over U. Birmingham/Tsinghua: 1.0/1.4

Conclusions

Conclusions

- New hardware implementations of 3 lattice-based finalists supporting fair rankings
- Among the 4 finalist KEMs, CRYSTALS-Kyber and Saber

  significantly outperform NTRU and Classic McEliece for at least a subset of all operations
- The differences between CRYSTALS-Kyber and Saber are relatively small, with
  - Saber slightly faster at Security Level 1
  - CRYSTALS-Kyber faster at Security Levels 3 & 5
  - CRYSTALS-Kyber smaller than Saber in terms of the #LUTs
- Four finalist KEMs outperform all alternate KEMs

Q&A

Thank You!

Questions?  Comments?

CERG: http://cryptography.gmu.edu

ATHENa: http://cryptography.gmu.edu/athena

Choose: PQC