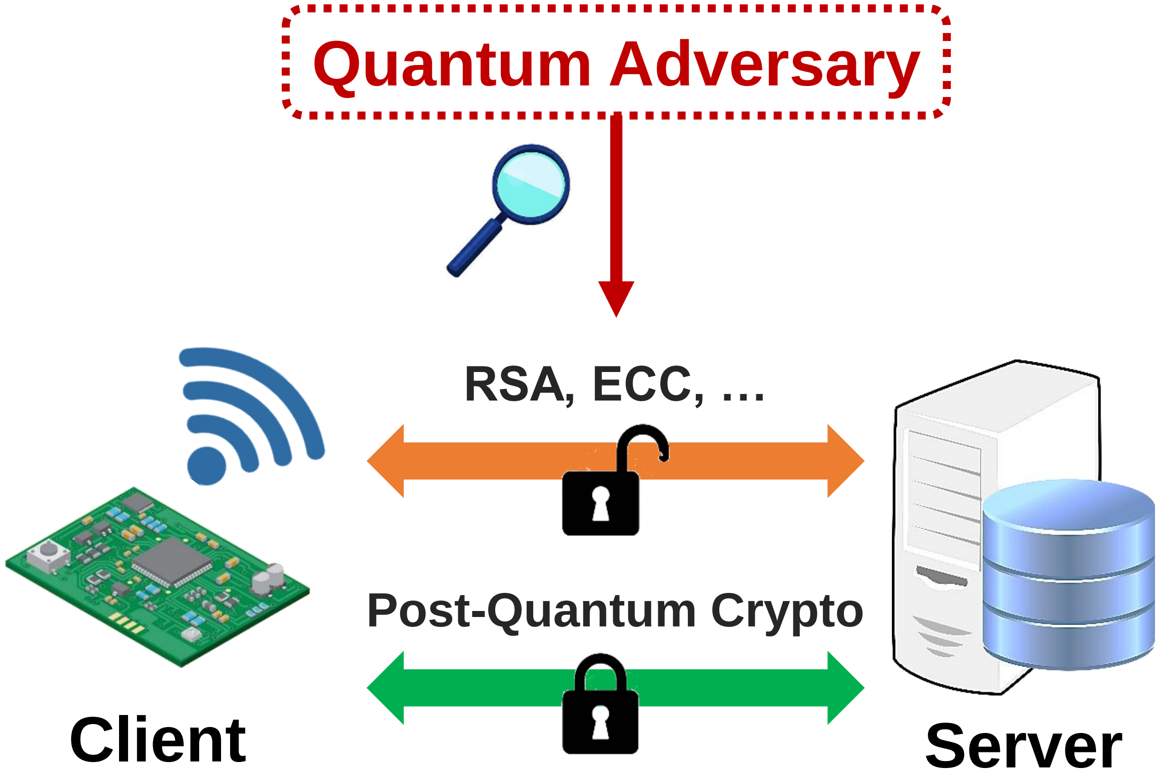
Sapphire: A Configurable Crypto-Processor for Post-Quantum Lattice-based Protocols

Utsav Banerjee[*], Tenzin S. Ukyab, Anantha P. Chandrakasan

[*]utsav@mit.edu

Massachusetts Institute of Technology

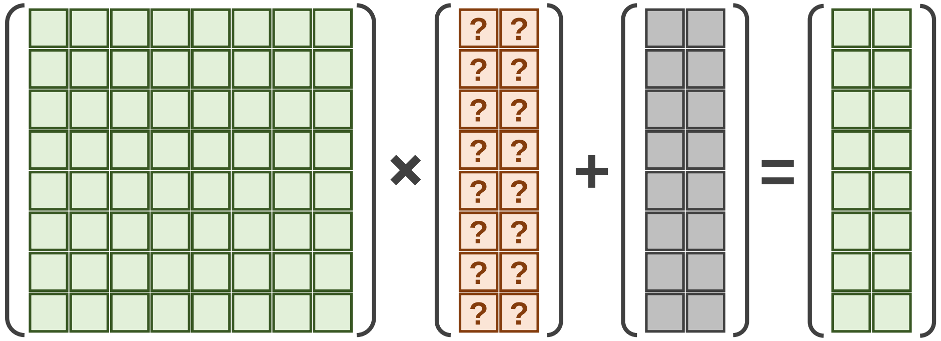
Post-Quantum Cryptography



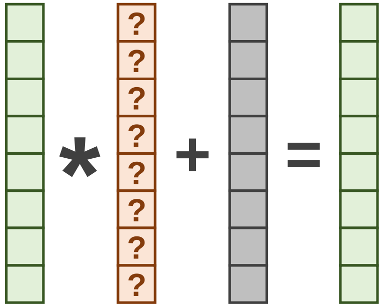
- ❑ Current public key cryptography vulnerable to quantum attacks
- ❑ NIST post-quantum crypto standardization in progress
- ❑ Round 2 has 26 candidates:
  - **Lattice-based (9 KEM + 3 Sign)**
  - Code-based (7 KEM)
  - Hash-based (1 Sign)
  - Multivariate (4 Sign)
  - Supersingular isogeny (1 KEM)
  - Zero-knowledge proofs (1 Sign)

Learning with Errors

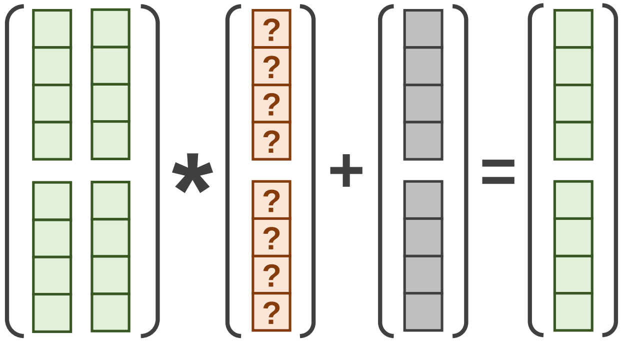
- ❑ **Learning with Errors (LWE) and its variants:**



LWE
(Standard Lattices)



Ring-LWE
(Ideal Lattices)

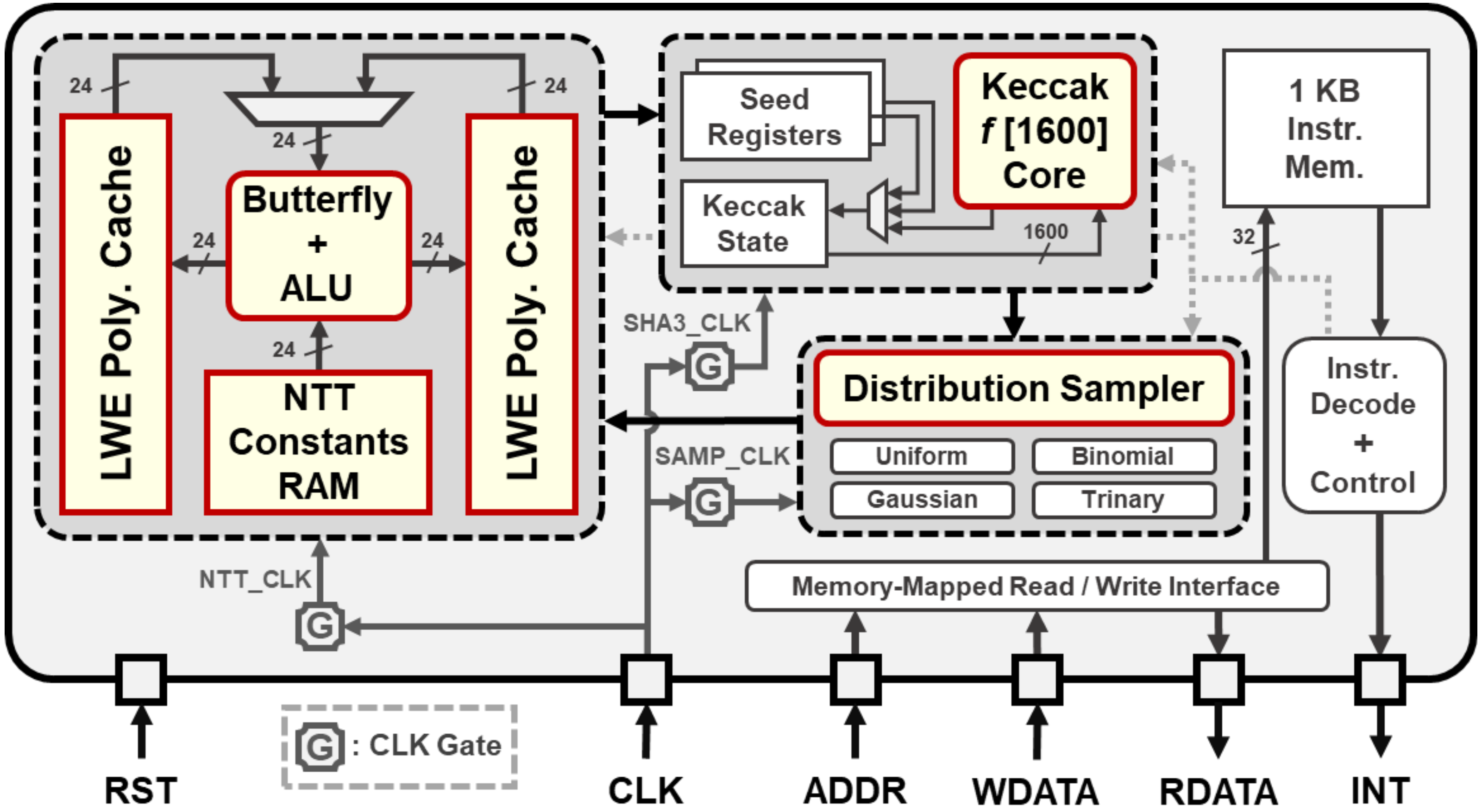


Module-LWE
(Module Lattices)

- ❑ **Computational requirements (apart from standard arithmetic):**
  - Modular arithmetic over various small primes
  - Polynomial arithmetic for Ring-LWE and Module-LWE
  - Sampling of matrices and polynomials from discrete distributions

Sapphire Crypto-Processor

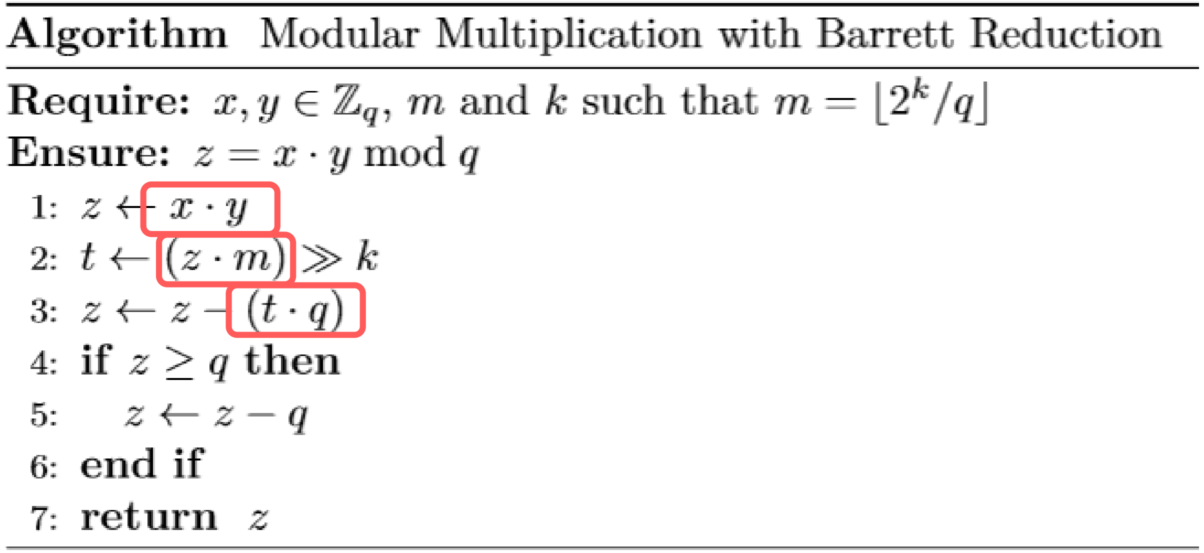
- ❑ Energy-efficient configurable lattice-crypto-processor



Outline

- Efficient Lattice-Crypto Hardware Implementation
  - Configurable Modular Multiplier
  - Area-Efficient NTT
  - Energy-Efficient Sampler
- Chip Architecture
- Measurement Results
- Side-Channel Analysis

Modular Multiplication



Algorithm Modular Multiplication with Barrett Reduction

Require: $x, y \in \mathbb{Z}_q$, m and k such that $m = \lfloor 2^k / q \rfloor$

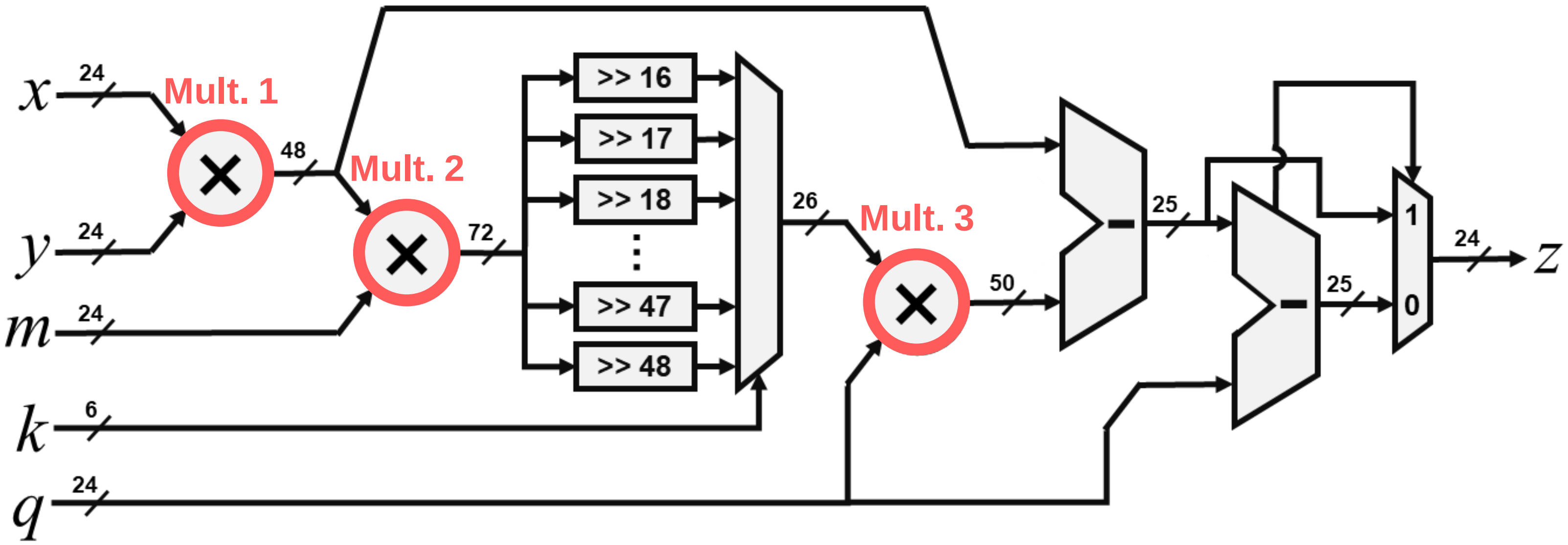
Ensure: $z = x \cdot y \bmod q$

```
$z \leftarrow \boxed{x \cdot y}$
$t \leftarrow \boxed{(z \cdot m)} \gg k$
$z \leftarrow z - \boxed{(t \cdot q)}$
if $z \geq q$ then
    $z \leftarrow z - q$
end if
return $z$
```

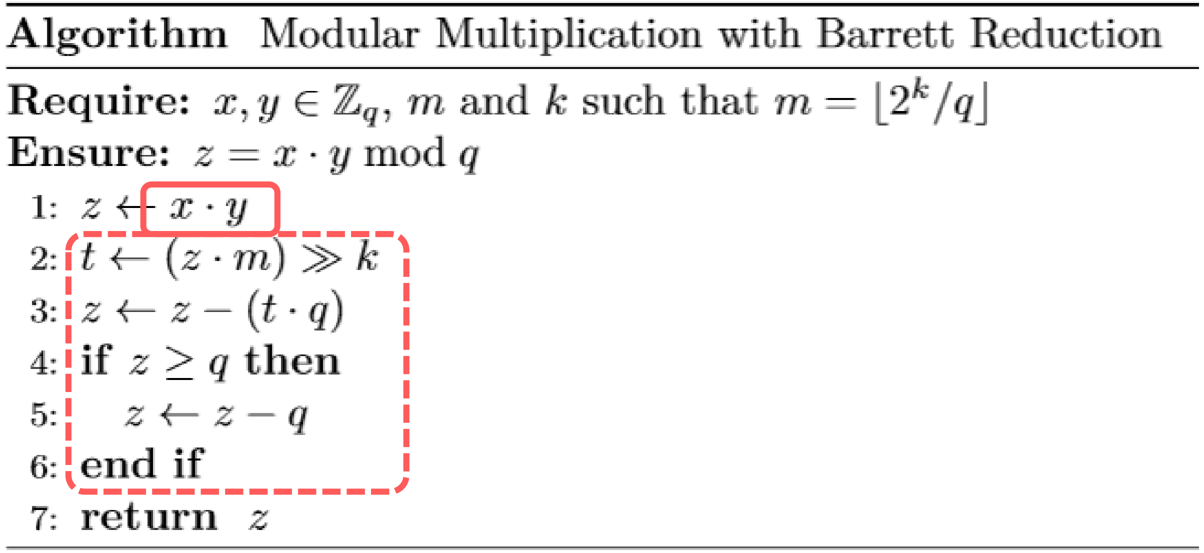
Reduction with fully configurable modulus:

- ❑ configurable parameters m, k, q
- ❑ m and q up to 24 bits
- ❑ $16 \leq k \leq 48$
- ❑ requires 2 explicit multipliers for reduction

Modular Multiplier Arch #1



Modular Multiplication



Algorithm Modular Multiplication with Barrett Reduction

Require: $x, y \in \mathbb{Z}_q$, m and k such that $m = \lfloor 2^k / q \rfloor$

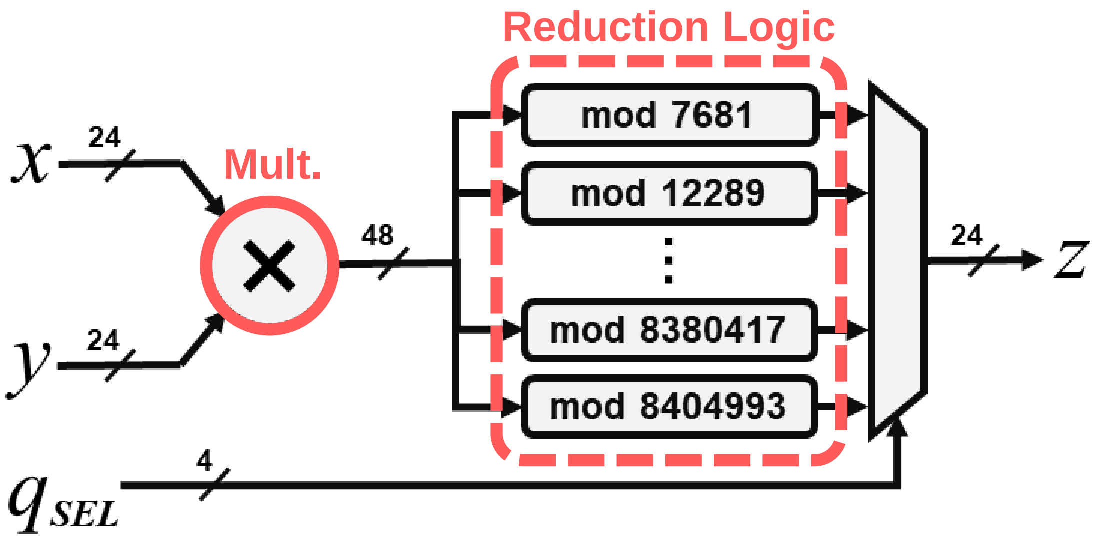
Ensure: $z = x \cdot y \bmod q$

```
$z \leftarrow x \cdot y$
$t \leftarrow (z \cdot m) \gg k$
$z \leftarrow z - (t \cdot q)$
if $z \geq q$ then
    $z \leftarrow z - q$
end if
return $z$
```

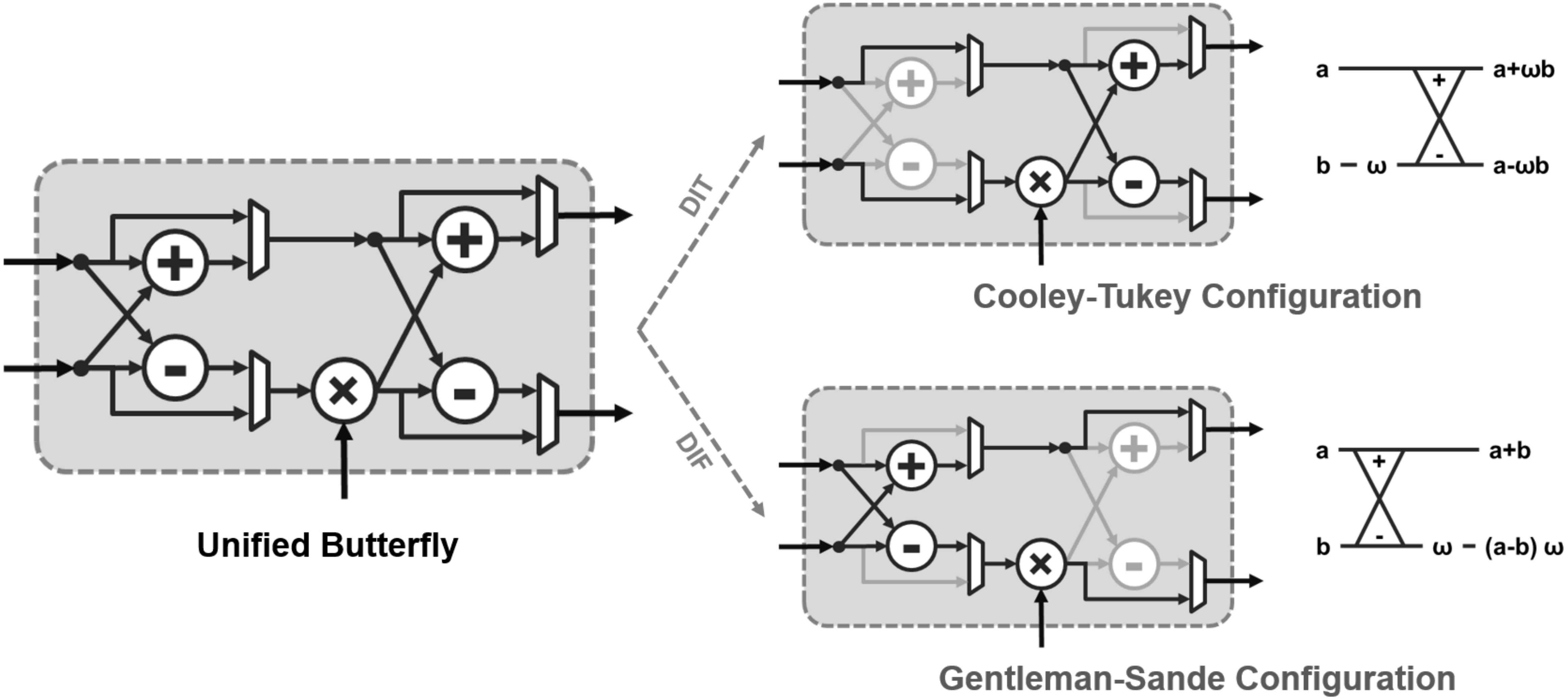
Reduction with pseudo-configurable modulus:

- ❑ choice of q from a set of primes
- ❑ reduction coded in digital logic
- ❑ requires no explicit multiplier for reduction
- ❑ up to 6× more energy-efficient

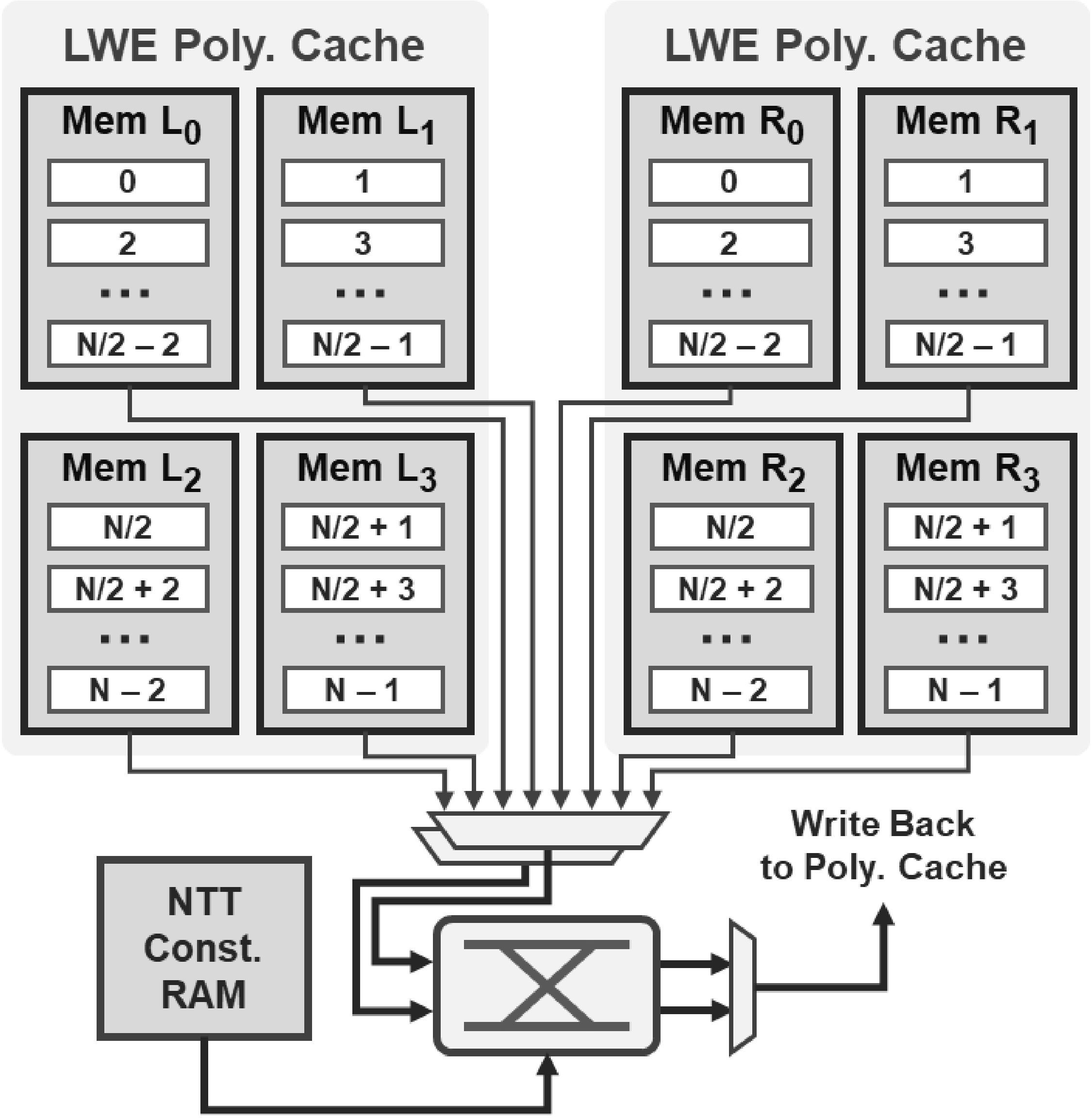
Modular Multiplier Arch #2



Unified Butterfly



Number Theoretic Transform



❑ NTT memory banks using dual-port SRAMs have large area overheads

❑ Proposed single-port SRAM-based NTT

❑ Based on constant geometry FFT data-flow

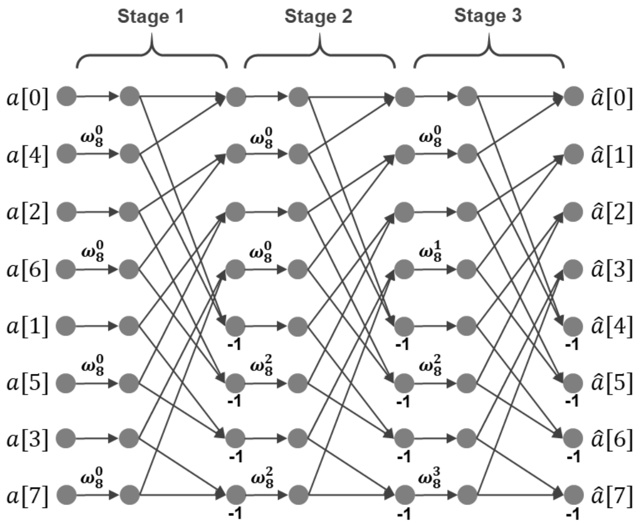
[Pease, J. ACM, 1968]

❑ Polynomials split among four single-port SRAMs based on address parity:

Mem #0	Mem #1	Mem #2	Mem #3
MSB(*addr*) = 0	MSB(*addr*) = 0	MSB(*addr*) = 1	MSB(*addr*) = 1
LSB(*addr*) = 0	LSB(*addr*) = 1	LSB(*addr*) = 0	LSB(*addr*) = 1

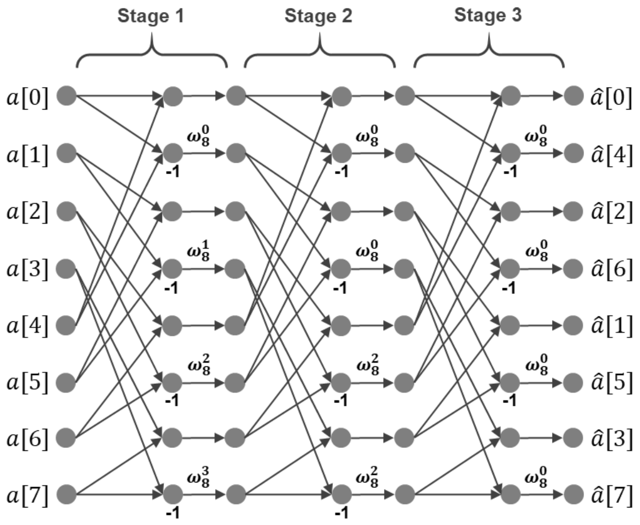
❑ Achieves > 30% area savings compared to dual-port implementation (without loss in throughput)

NTT Data Flow



	Stage 1				Stage 2				Stage 3			
Cycle Count:	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Mem L_0	R	R			W		W		R	R		
Mem L_1	R	R				W		W	R	R		
Mem L_2			R	R	W		W				R	R
Mem L_3			R	R		W		W			R	R
Mem R_0	W		W		R	R			W		W	
Mem R_1		W			R	R				W		W
Mem R_2	W			W			R	R	W		W	
Mem R_3		W					R	R		W		W

8-point Decimation-in-Time NTT



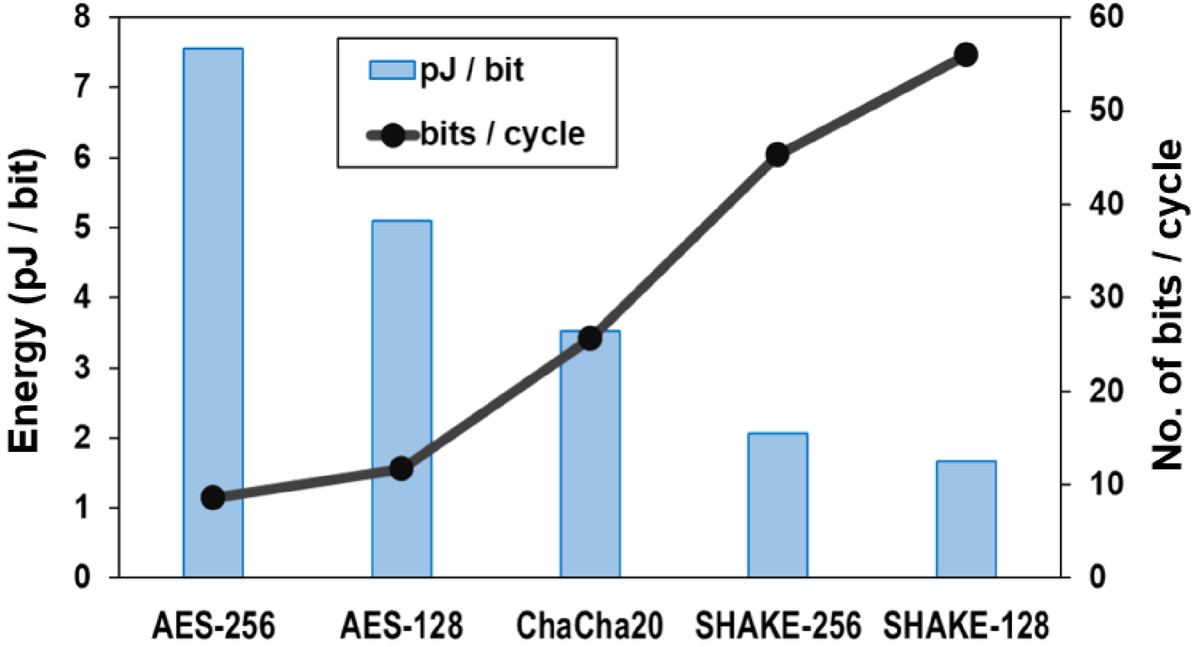
	Stage 1				Stage 2				Stage 3			
Cycle Count:	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Mem L_0	R		R		W	W			R		R	
Mem L_1		R		R	W	W				R		R
Mem L_2	R		R				W	W	R		R	
Mem L_3		R		R			W	W		R		R
Mem R_0	W	W			R		R		W	W		
Mem R_1	W	W				R		R	W	W		
Mem R_2			W	W	R		R				W	W
Mem R_3			W	W		R		R			W	W

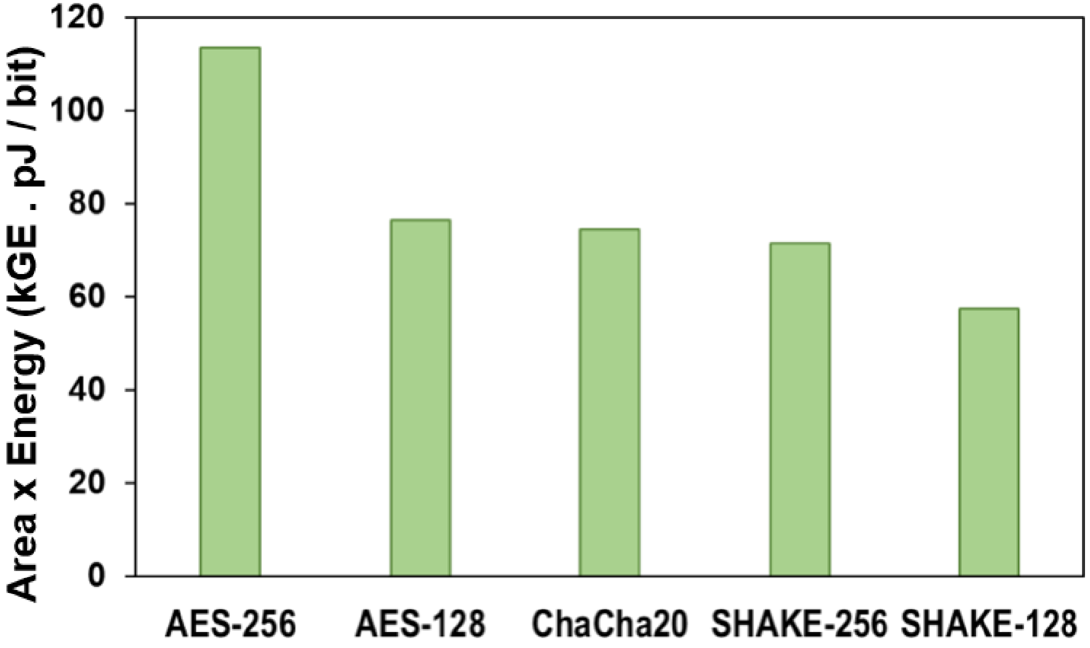
8-point Decimation-in-Frequency NTT

- ❑ One butterfly per cycle
- ❑ No read / write hazards
- ❑ No energy overheads

Energy-Efficient PRNG

Standard CS-PRNG: □ SHAKE-128 / 256 □ AES-128 / 256 □ ChaCha20

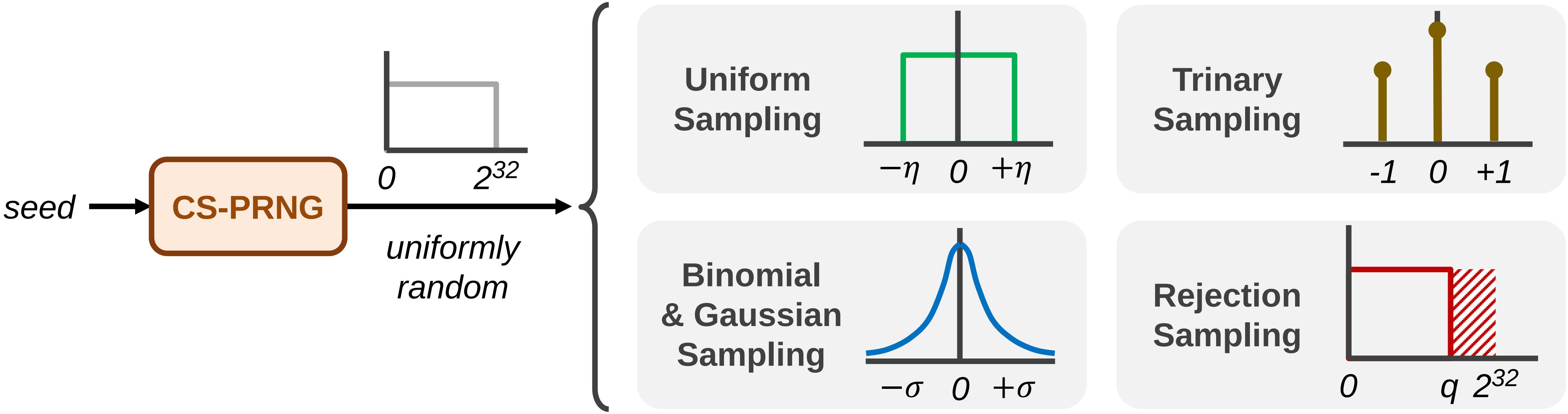


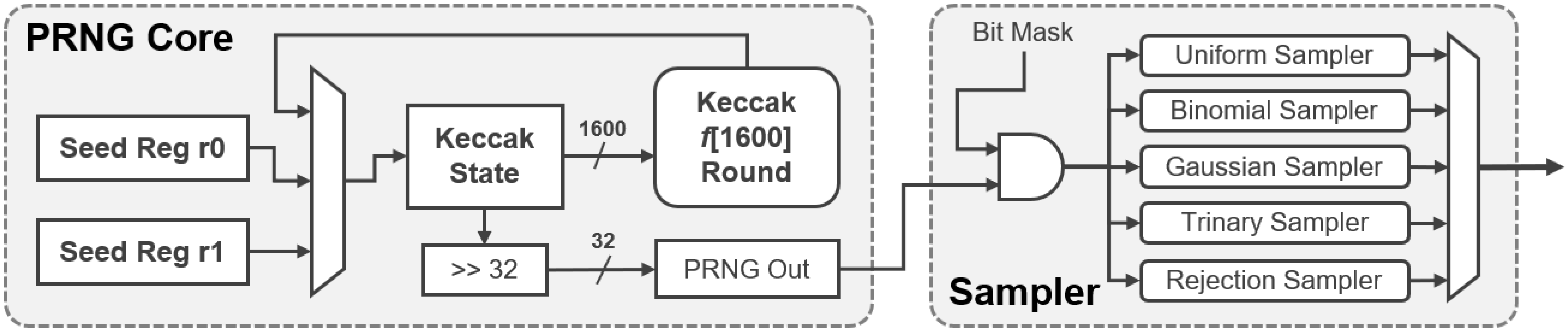


Keccak-based PRNG:

24-cycles and 2.33 nJ per round @ 1.1V

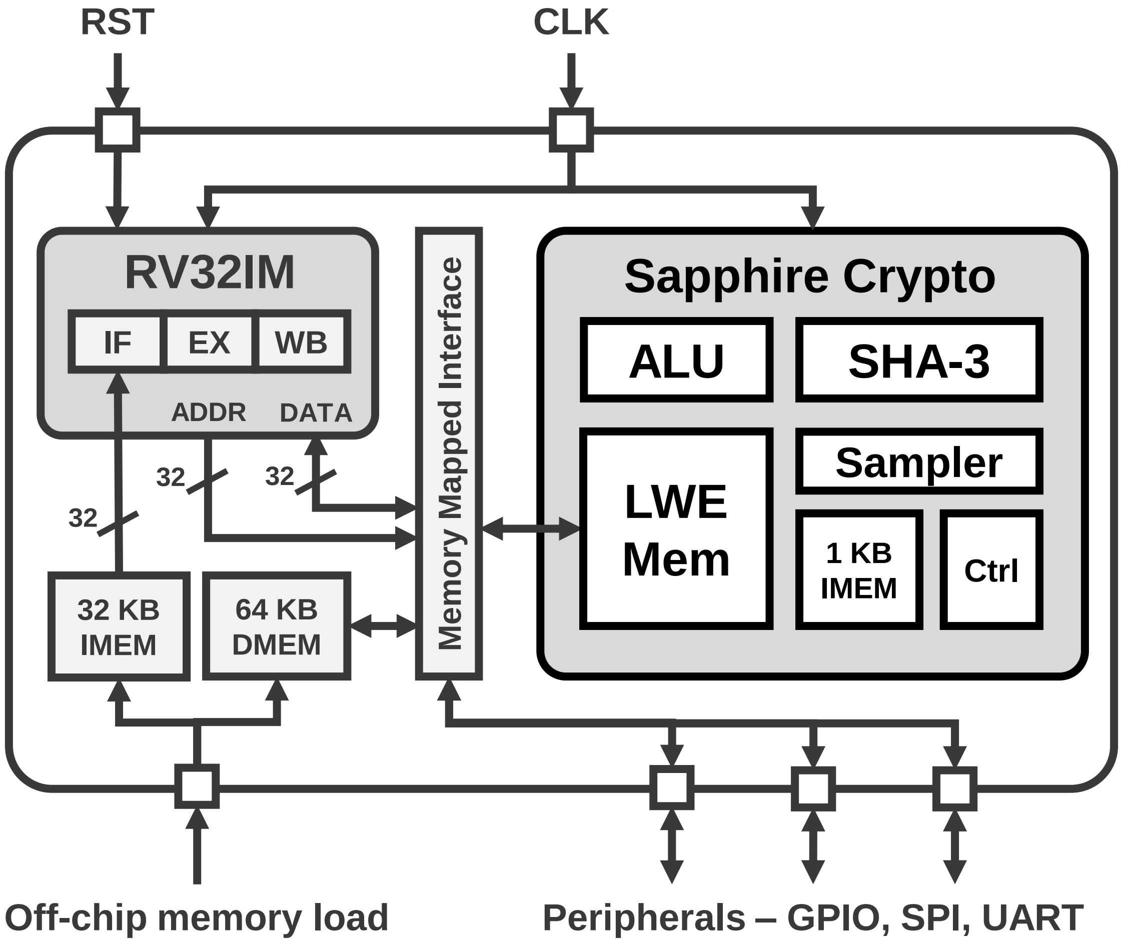
Discrete Distribution Sampler

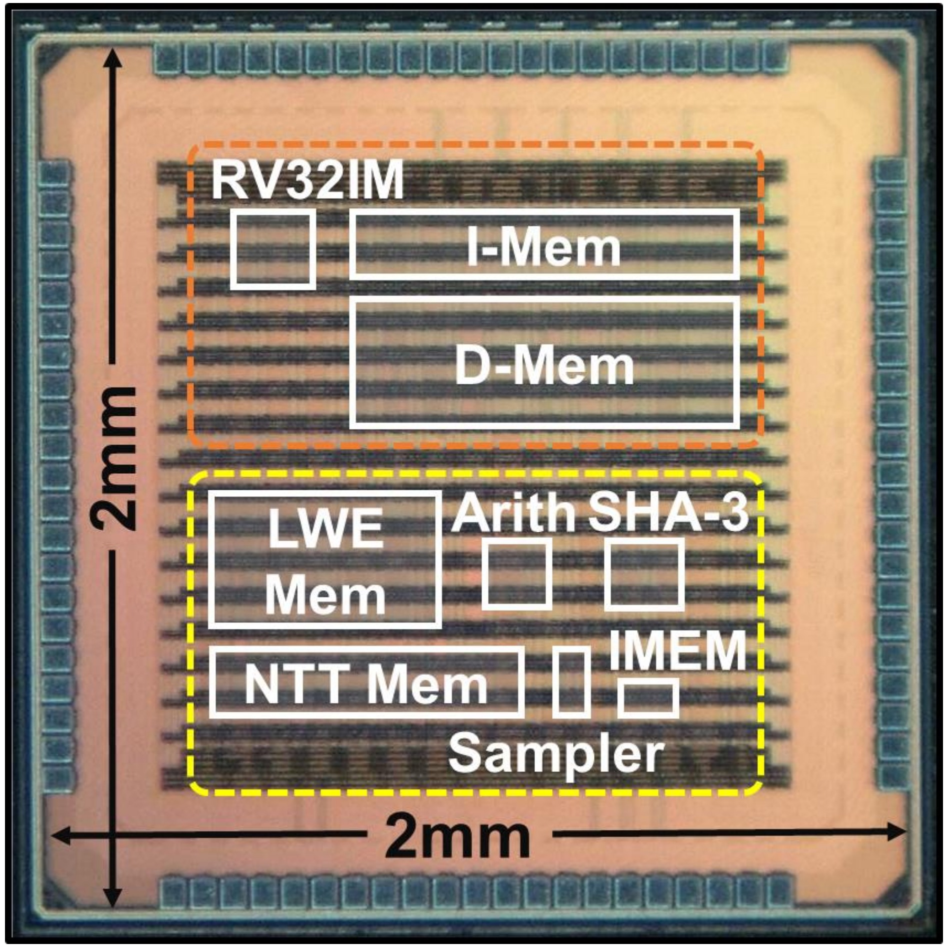




Test Chip Overview

- Crypto core integrated with RISC-V processor





Chip Micrograph

Protocol Implementations

- Following NIST Round 2 protocols were implemented on our test chip:

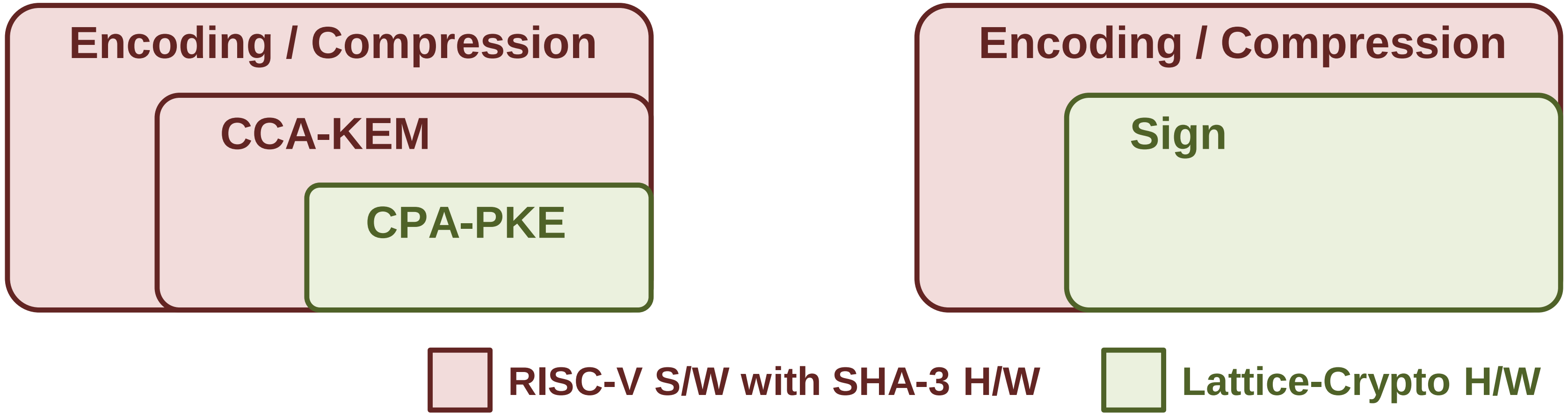
CCA-KEM	LWE	Frodo
	Ring-LWE	NewHope
	Module-LWE	CRYSTALS-Kyber

Signature	Ring-LWE	qTesla
	Module-LWE	CRYSTALS-Dilithium

- Computations shared between crypto core and RISC-V processor:

PKE / KEM:

Sign:



Encoding / Compression

CCA-KEM

CPA-PKE

Encoding / Compression

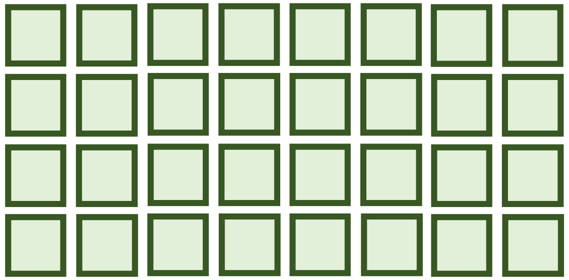
Sign

🟥 RISC-V S/W with SHA-3 H/W

🟩 Lattice-Crypto H/W

Implementation of RLWE and MLWE

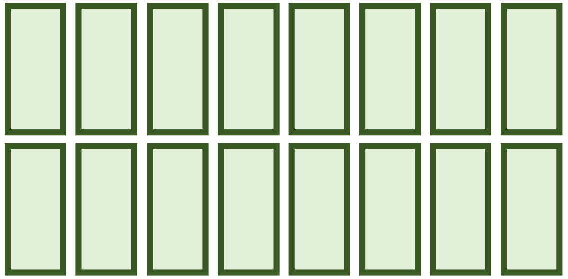
- Efficient utilization of 24 KB polynomial memory with 8192 elements



n = 256
32 polynomials

CRYSTALS-Kyber

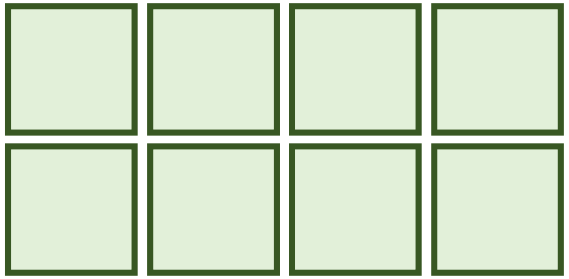
CRYSTALS-Dilithium



n = 512
16 polynomials

NewHope-512

qTesla-I



n = 1024
8 polynomials

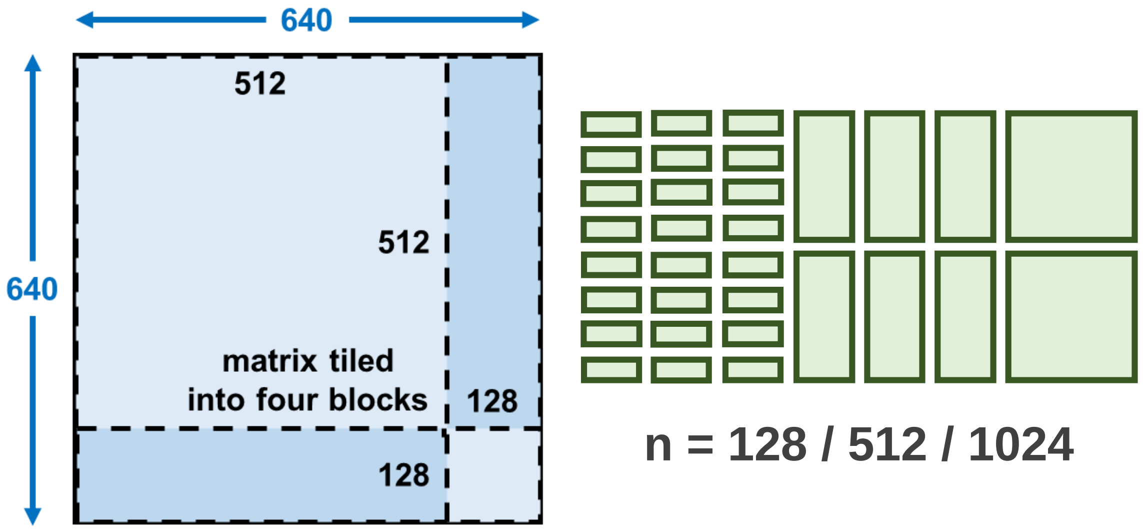
NewHope-1024

qTesla-III

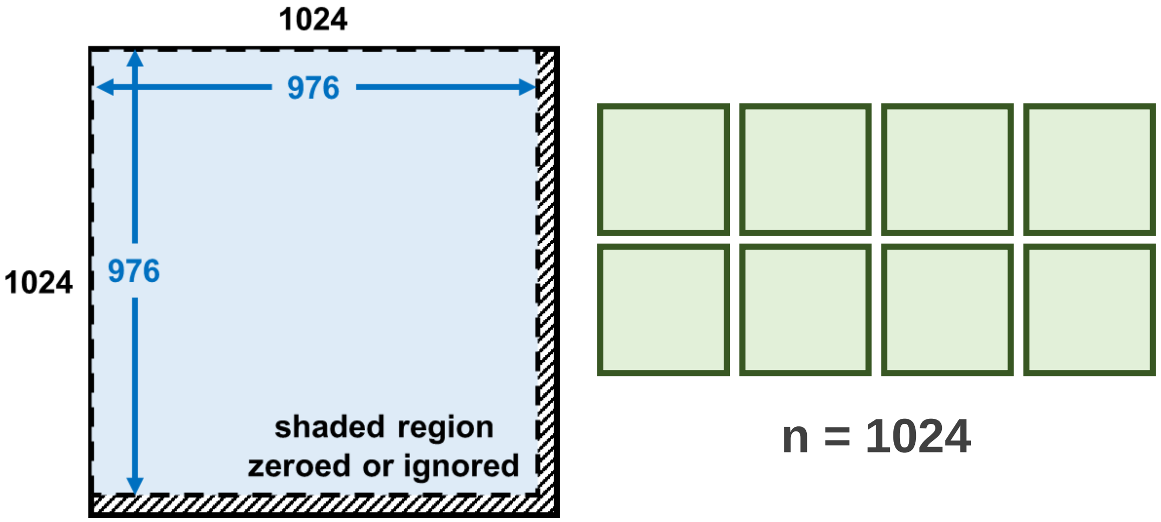
- Crypto core used to accelerate sampling and polynomial arithmetic
- Protocol scheduling, compression and encoding performed on RISC-V processor

Implementation of LWE

- Polynomial memory tiled to support non-power-of-two-size matrix manipulation



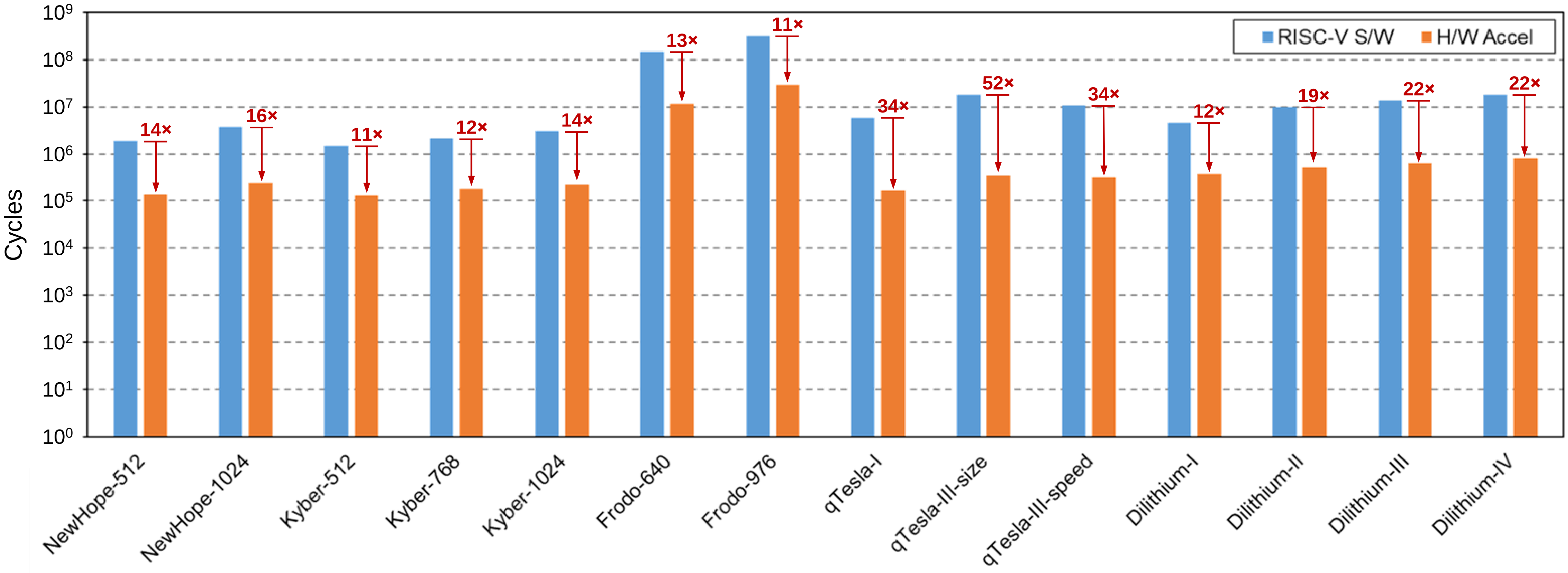
Frodo-640



Frodo-976

- Crypto core used to accelerate sampling and matrix arithmetic
- Protocol scheduling, compression and encoding performed on RISC-V processor

Protocol Evaluation Results

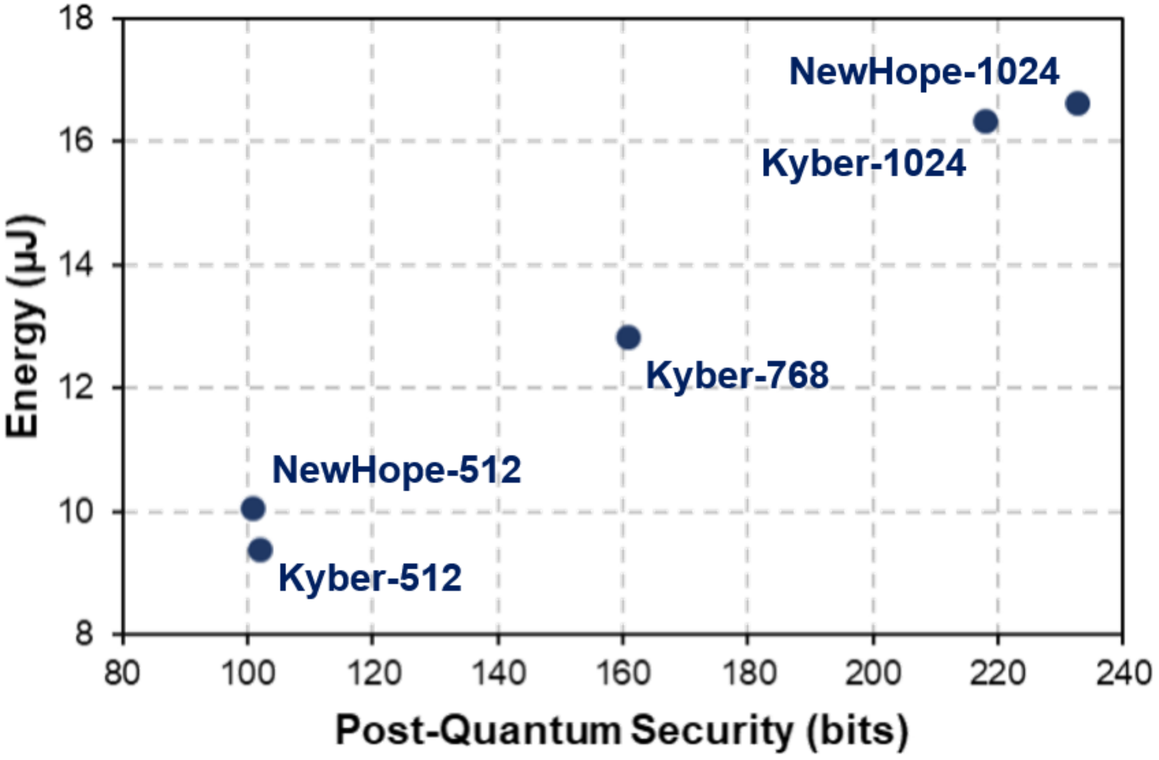


** Cycle counts for CCA-KEM-Encaps and Sign*

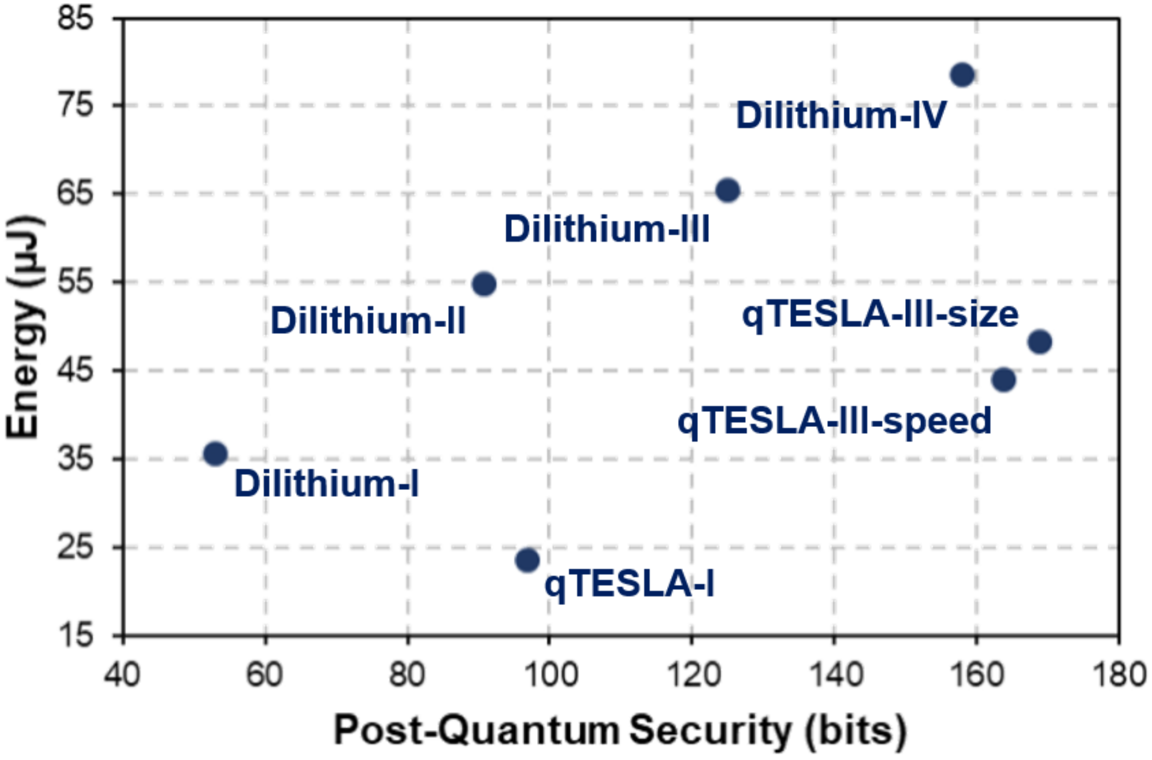
Order of magnitude improvement in energy-efficiency and performance

Protocol Evaluation Results

CCA-KEM-Encaps



Sign



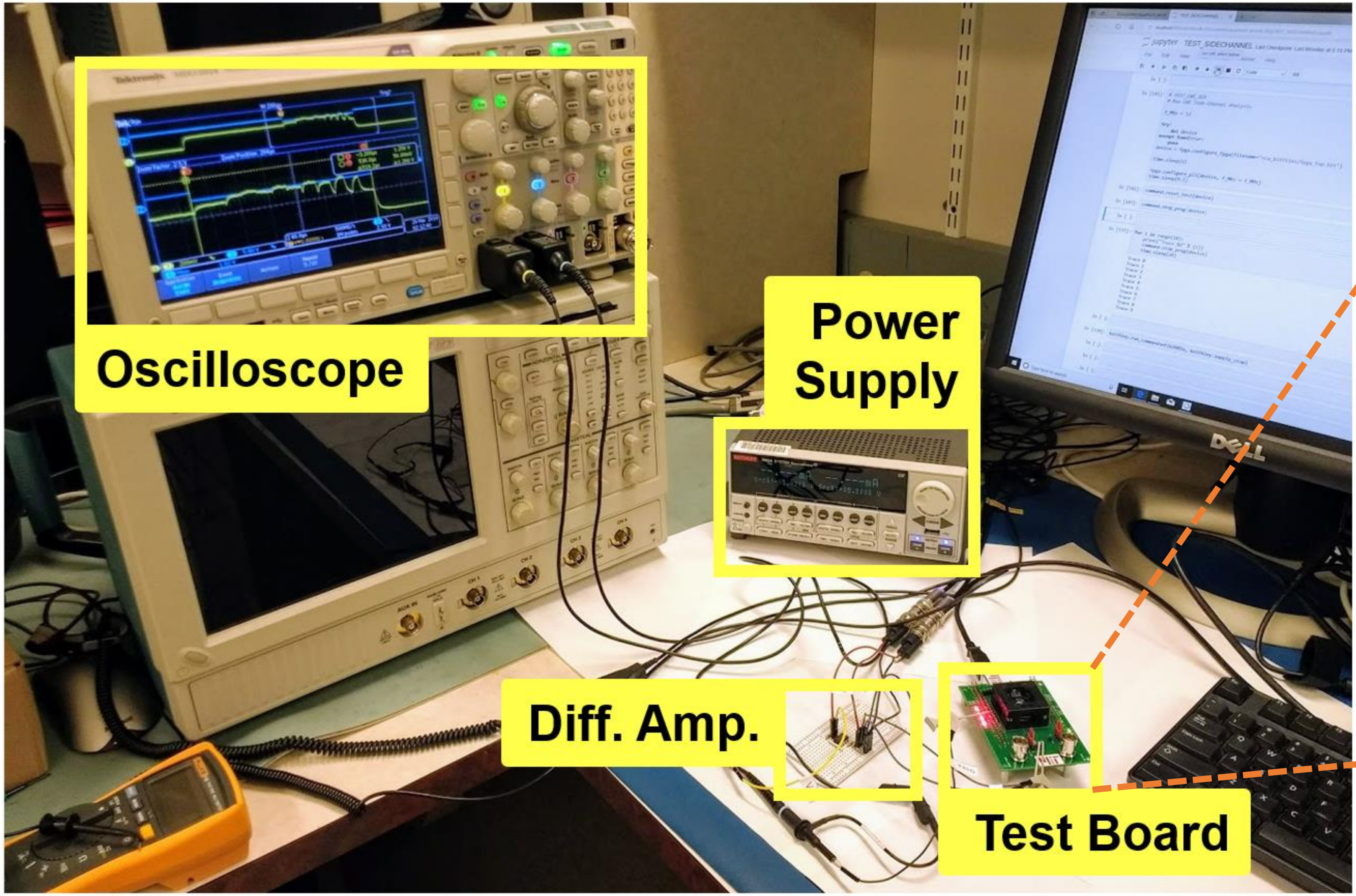
** Measured using test chip operating at 1.1 V and 72 MHz*

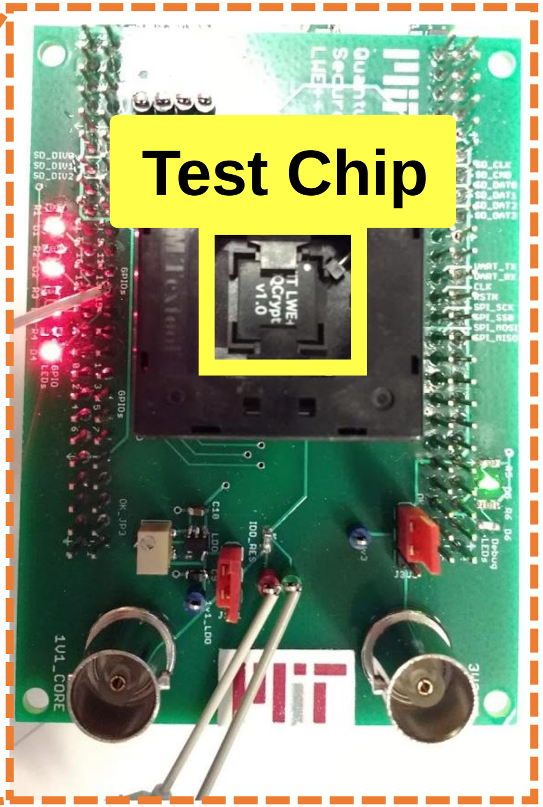
Performance Comparison

Design	Platform	Tech (nm)	VDD (V)	Freq (MHz)	Protocol	Area (kGE)	Cycles	Energy (µJ)
This work	ASIC	40	1.1	72	NewHope-512-CCA-KEM-Encaps NewHope-1024-CPA-PKE-Encrypt Kyber-512-CCA-KEM-Encaps Kyber-768-CPA-PKE-Encrypt Kyber-768-CCA-KEM-Encaps Frodo-640-CCA-KEM-Encaps Dilithium-II-Sign	106	136,077 106,611 131,698 94,440 177,540 11,609,668 514,246	10.02 12.00 9.37 10.31 12.80 1129.95 54.82
Basu et al. [BSNK19] †	ASIC	65	1.2	169 200 158	NewHope-512-CCA-KEM-Encaps Kyber-512-CCA-KEM-Encaps Dilithium-II-Sign	1273 1341 1603	307,847 31,669 155,166	69.42 6.21 50.42
Albrecht et al. [AHH+18]	SLE 78	-	-	50	Kyber-768-CPA-PKE-Encrypt Kyber-768-CCA-KEM-Encaps	-	4,747,291 5,117,996	-
Oder et al. [OG17]	FPGA	-	-	117	NewHope-1024-Simple-Encrypt	-	179,292	-
Howe et al. [HOKG18]	FPGA	-	-	167	Frodo-640-CCA-KEM-Encaps	-	3,317,760	-
Fritzmann et al. [FSM+19]	FPGA	-	-	-	NewHope-1024-CPA-PKE-Encrypt	-	589,285	-

† Only post-synthesis area and energy consumption reported

Side-Channel Analysis Setup





Test Board

Timing and SPA Side-Channels

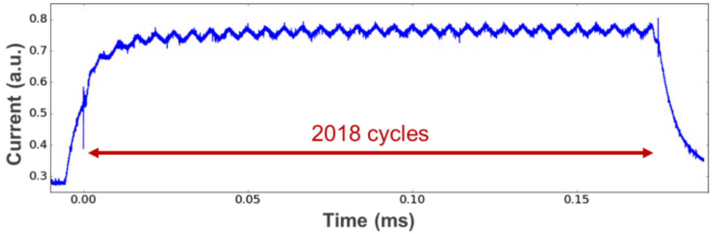
- ❑ All key building blocks constant-time by design

- ❑ Energy consumption of sampling and polynomial arithmetic follows a narrow distribution with coefficient of variation ≤ 0.5% ($= \sigma / \mu$)

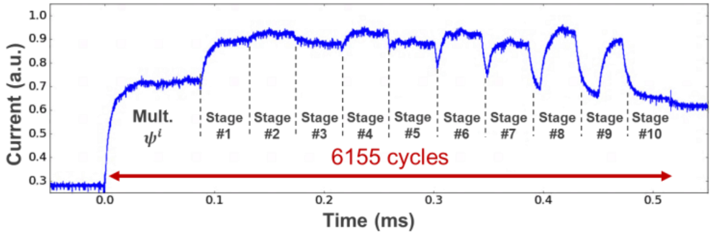
- ❑ SPA attacks target polynomial arithmetic:
  - Number Theoretic Transform
  - Coefficient-wise Multiplication
  - Coefficient-wise Addition

- ❑ SPA resistance of polynomial arithmetic evaluated using difference-of-means test with 99.99% confidence interval

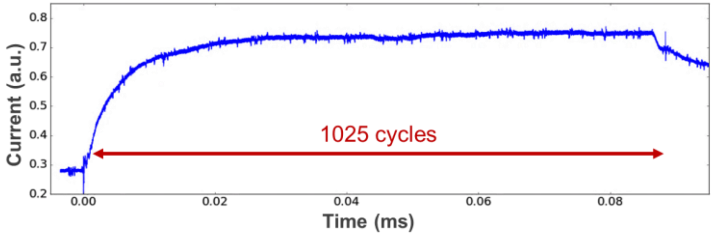
Binomial Sampling



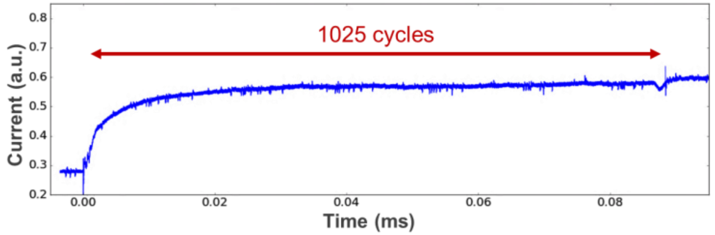
Number Theoretic Transform



Polynomial Coefficient-wise Multiplication



Polynomial Coefficient-wise Addition



Masking for DPA Security

- ❑ Protocol evaluations without any DPA countermeasures
- ❑ Masked NewHope-CPA-PKE-Decrypt based on additively homomorphic property:

[Reparaz et al, PQCrypto, 2016]

1. Generate secret message μ_r
2. Encrypt μ_r to its corresponding ciphertext $c_r = (\hat{u}_r, v_r')$
3. Compute $c_m = (\hat{u} + \hat{u}_r, v' + v_r')$ where $\mathrm{c} = (\hat{u}, v')$ is the original ciphertext
4. Decrypt c_m to obtain $\mu_m = \mu \oplus \mu_r$ where μ is the original message
5. Recover original message as $\mu = \mu_m \oplus \mu_r$

- ❑ Masked decryption using same hardware; 3× slower than unmasked version
- ❑ Masking increases decryption failure rate, which can be resolved by decreasing std. dev. σ of error distribution (at the cost of slightly lower security level)
- ❑ Leakage tests and CCA-KEM masking – work in progress

Conclusion

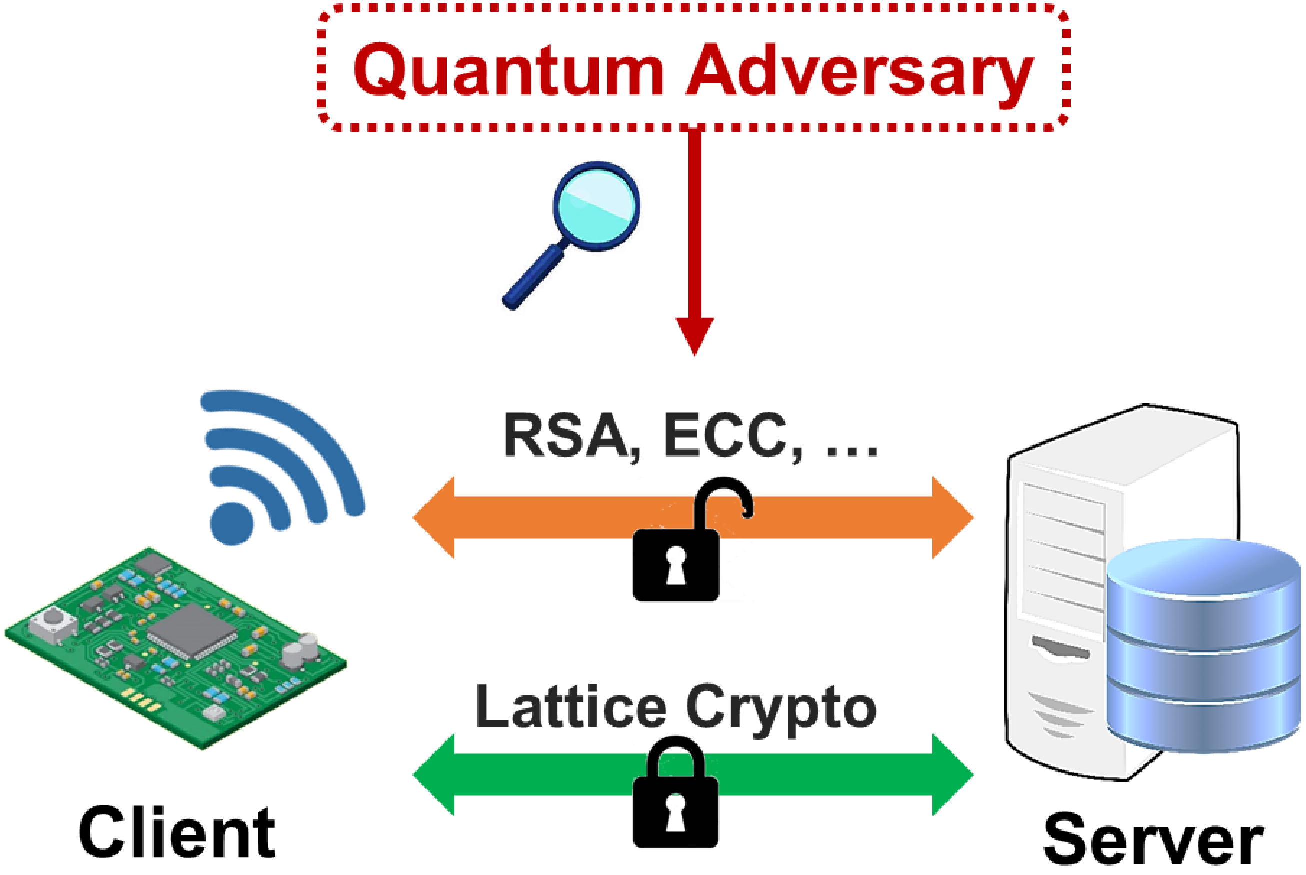
- ❑ Configurable crypto-processor for LWE, Ring-LWE and Module-LWE protocols
- ❑ Area-efficient NTT, energy-efficient sampler and flexible parameters
- ❑ ASIC demonstration of NIST Round 2 CCA-KEM and signature protocols: Frodo, NewHope, Kyber, qTesla, Dilithium
- ❑ Order of magnitude improvement in performance and energy-efficiency compared to state-of-the-art software and hardware
- ❑ Hardware building blocks constant-time and SPA-secure by design; masking can also be implemented for DPA security

Acknowledgements

❑ Texas Instruments for funding

❑ TSMC University Shuttle Program for chip fabrication

Questions


Quantum Adversary
RSA, ECC, …
Lattice Crypto
Client
Server