Pham Laboratory
Integrated circuit design laboratory

UEC TOKYO

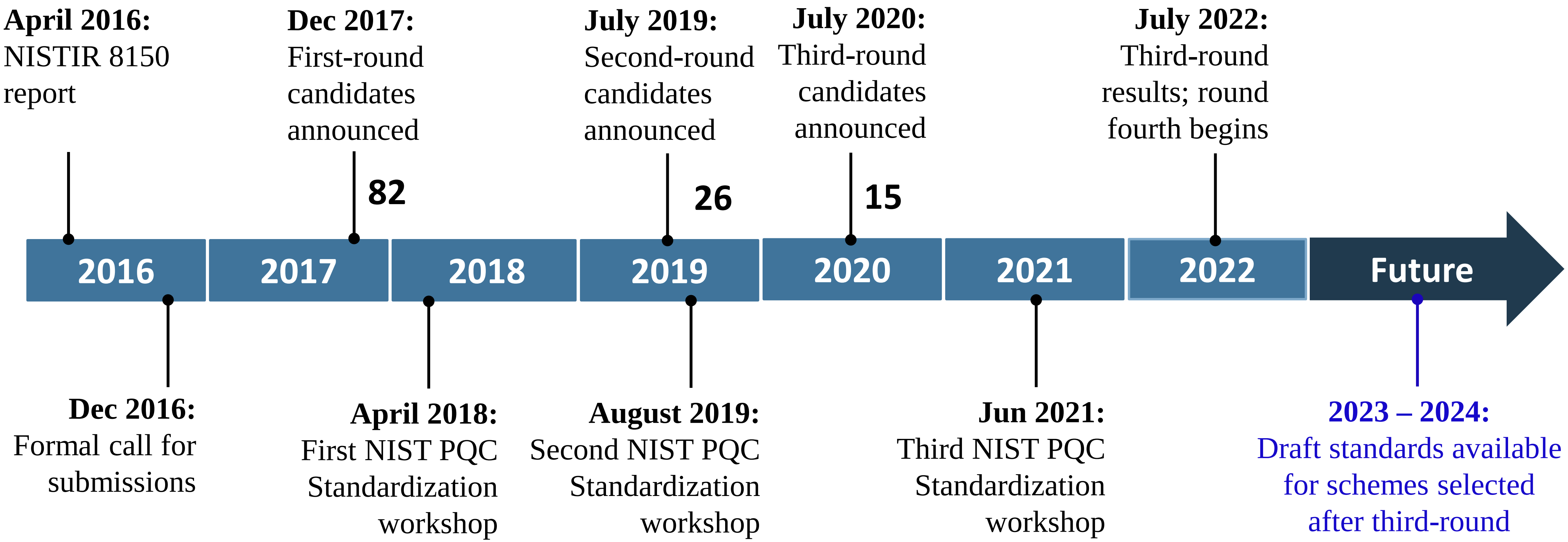
The Lattice-based Post-quantum Cryptography The case study of CRYSTAL-Kyber

Outline

1. Post-quantum Cryptography Competition
2. Lattice and it’s hard-problems
3. The case study of CRYSTAL-Kyber
4. Accelerating by Number Theoretic Transform
5. Discussion

1. Post-quantum Cryptography Competition

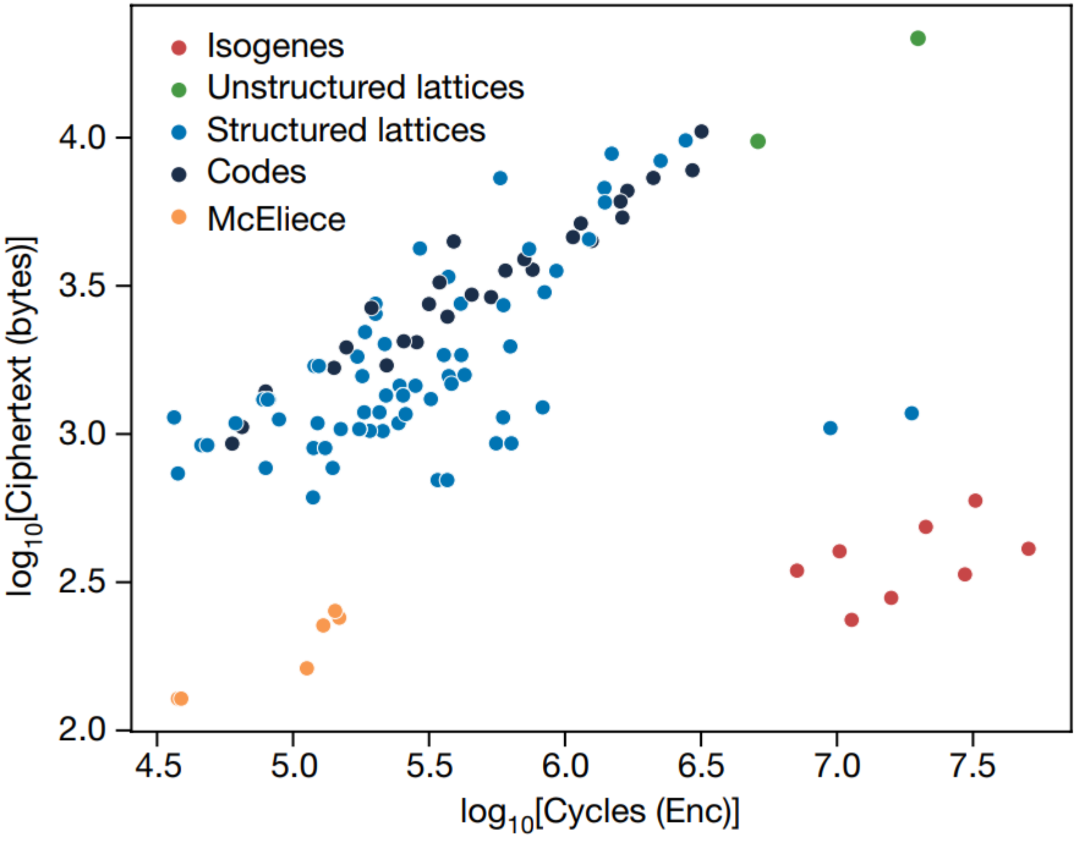
Post-quantum cryptography (PQC) process timeline by NIST*

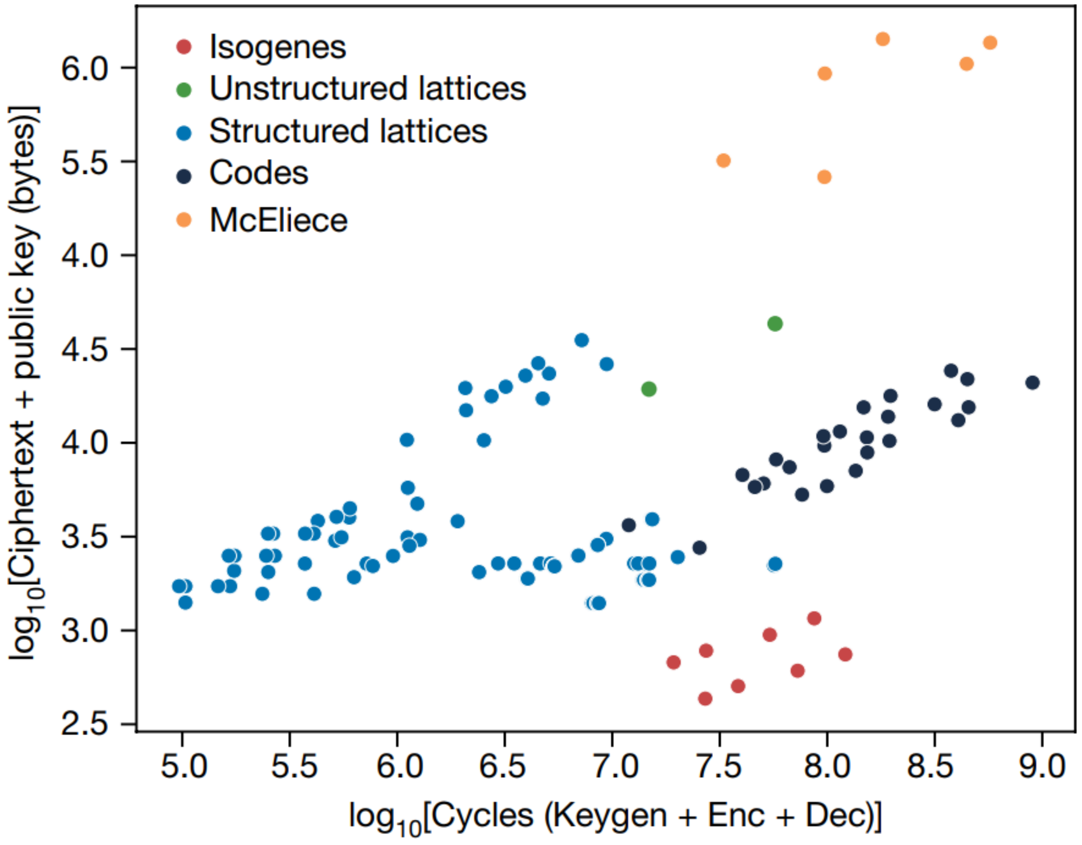


***NIST**: National Institute of Standards and Technology

1. Post-quantum Cryptography Competition

NIST Post-quantum cryptography algorithm performance [1]





[1] Joseph, David, et al. **"Transitioning organizations to post-quantum cryptography."** Nature 605.7909 (2022): 237-243.

1. Post-quantum Cryptography Competition

The initial PQC algorithms to be standardized

- The NIST round 3 was concluded on July 5, 2022

Public-Key Encryption/KEMs	
Finalists	*Alternates*
CRYSTAL-Kyber(1)	BIKE(2) Classic McEliece(2) HQC(2) SIKE(4)

Digital Signatures	
Finalists	*Alternates*
CRYSTAL-Dilithium(1) FALCON(1) SPHINCS+(3)	

(1) Lattice-based (2) Code-based (3) Hash-based (4) Isogeny-based

1. Post-quantum Cryptography Competition

The initial PQC algorithms to be standardized

- The NIST round 3 was concluded on July 5, 2022

Public-Key Encryption/KEMs	
Finalists	*Alternates*
CRYSTAL-Kyber(1)	BIKE(2) Classic McEliece(2) HQC(2) ~~SIKE(4)~~

Digital Signatures	
Finalists	*Alternates*
CRYSTAL-Dilithium(1) **FALCON(1)** SPHINCS+(3)	

(1) Lattice-based (2) Code-based (3) Hash-based (4) Isogeny-based

2. Lattice and It's Hard-problems

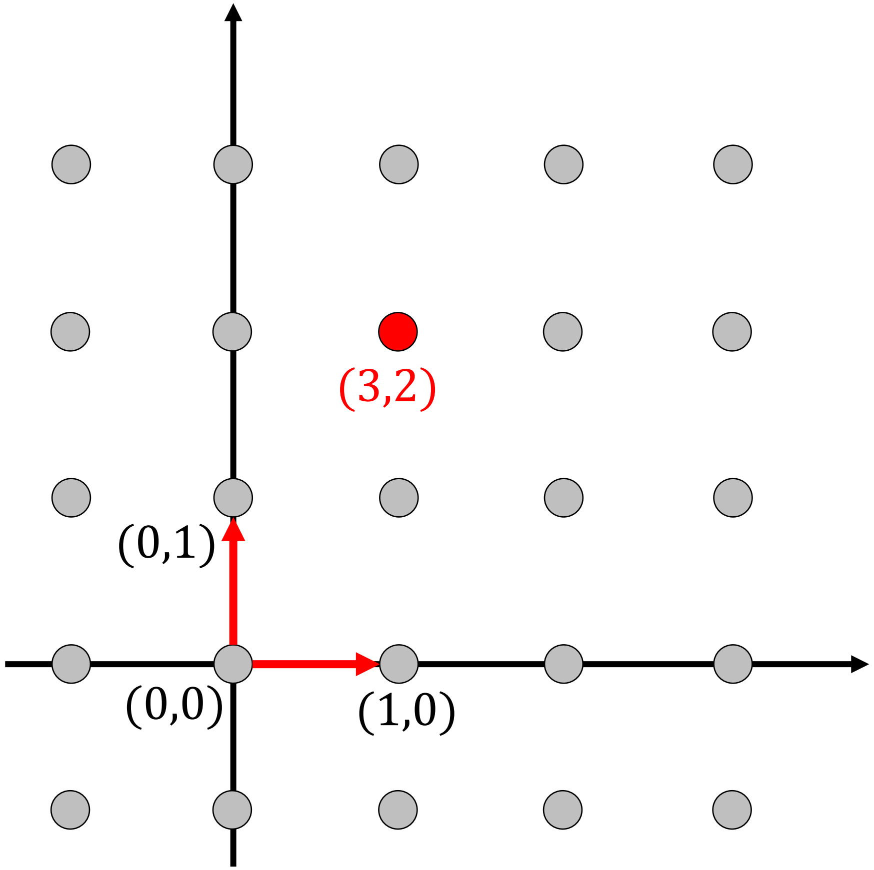
What is a lattice ?

- Lattices are basically a regular-spaced grid of a set of points that are infinite in number.
- The "basis" vectors are used to present any point in the lattice grid that forms a lattice.

$$B = \{b_o, b_1\} = \left\{\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right\} \in Z^2$$

$$L = \{a_0 b_0 + a_1 b_1\}$$

Example: $3\begin{pmatrix}1\\0\end{pmatrix} + 2\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}3\\2\end{pmatrix}$

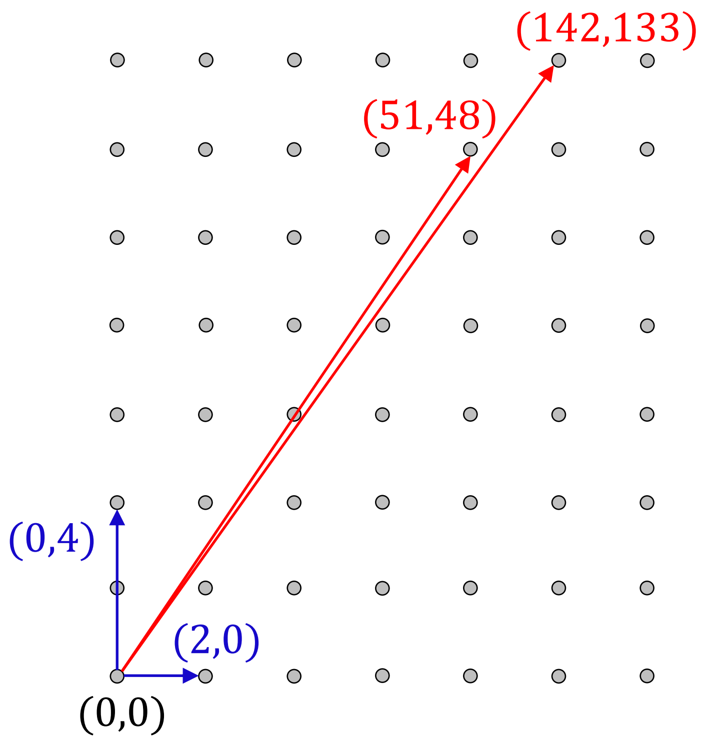


2. Lattice and It's Hard-problems

“Good” & “Bad” basis

- A Good basis
  - The basis consists of short length of vectors
  - The vectors are being orthogonal to each other.

- A Bad basis
  - The basis consists of long length of vectors
  - The vectors are being non-orthogonal to each other.



2. Lattice and It's Hard-problems

What hard-problem is?

Closest Vector Problem (CVP):

Given a lattice and a randomly chosen point P, the CVP asks to find the closest lattice point to challenge P.

Assume $P = \binom{29}{12}$; $L_{bad} = \{a_0\binom{51}{48} + a_1\binom{142}{133}\}$; $L_{good} = \{a_0\binom{3}{0} + a_1\binom{0}{3}\}$

- Good basis case

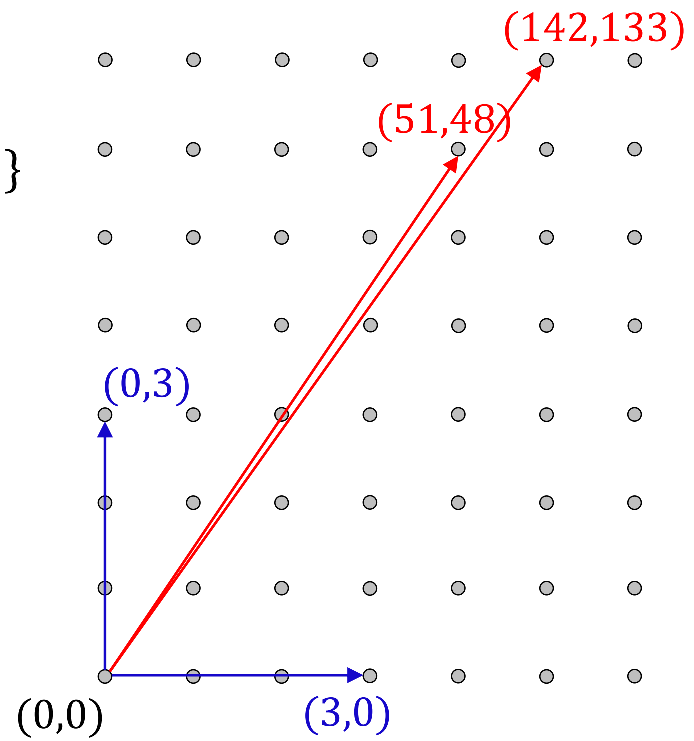
$\begin{cases} 3a_0 + 0a_1 = 29 \\ 0a_0 + 3a_1 = 12 \end{cases} \rightarrow \begin{cases} a_0 = 9.6 \\ a_1 = 4 \end{cases} \rightarrow (a_0, a_1) = (10,4)$

Calculates P= $10\binom{3}{0} + 4\binom{0}{3} = \binom{30}{12}$ → near $\binom{29}{12}$!

- Bad basis case:

$\begin{cases} 51a_0 + 142a_1 = 29 \\ 48a_0 + 133a_1 = 12 \end{cases} \rightarrow \begin{cases} a_0 = -65.24 \\ a_1 = 23.64 \end{cases} \rightarrow (a_0, a_1) = (-65,24)$

Calculates P= $-65\binom{51}{48} + 24\binom{142}{133} = \binom{93}{72}$ → incorrect!



2. Lattice and It's Hard-problems

The basic idea behind lattice-based cryptosystem

- An asymmetric key encryption with a public key for encryption and a private key for decryption

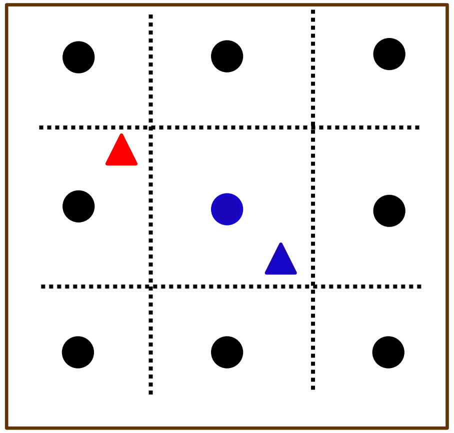
  - **Keys**

    - Public keys: $B_{bad} = \left\{ \begin{pmatrix} 51 \\ 48 \end{pmatrix}, \begin{pmatrix} 142 \\ 133 \end{pmatrix} \right\}$; Private keys: $B_{good} = \left\{ \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 3 \end{pmatrix} \right\}$

  - **Encryption**

    - Message: "HI" (H=43, I = -15) ➔ Step 1: $43 \begin{pmatrix} 51 \\ 48 \end{pmatrix} - 15 \begin{pmatrix} 142 \\ 133 \end{pmatrix} = \begin{pmatrix} 63 \\ 69 \end{pmatrix}$

    Step 2: $\begin{pmatrix} 63 \\ 69 \end{pmatrix} + \begin{pmatrix} -0.4 \\ 0.2 \end{pmatrix} = \begin{pmatrix} 62.6 \\ 69.2 \end{pmatrix}$



  - **Decryption**

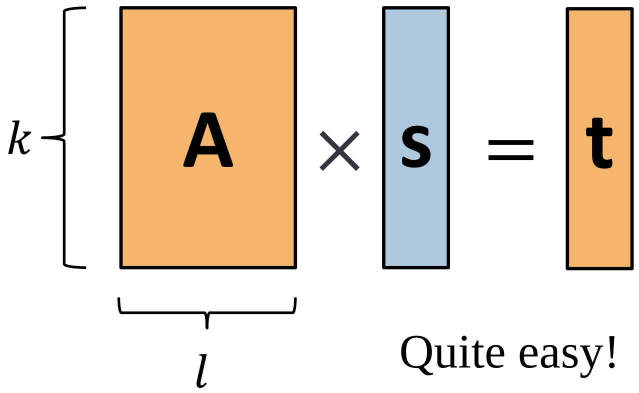
    Step 1: $a_0 \begin{pmatrix} 3 \\ 0 \end{pmatrix} + a_1 \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 62.4 \\ 69.2 \end{pmatrix}$ ➔ $(a_0, a_1) = (20.8, 23.07) \approx (21, 23)$

    Step 2: $21 \begin{pmatrix} 3 \\ 0 \end{pmatrix} + 23 \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 63 \\ 69 \end{pmatrix}$

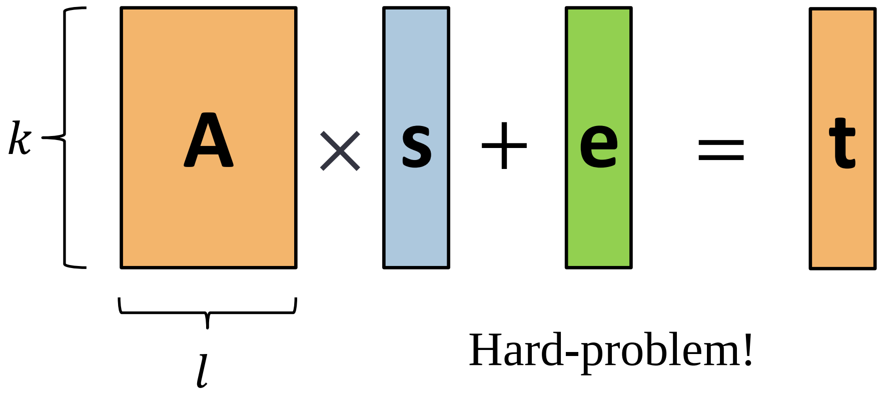
    Step 3: $a_0 \begin{pmatrix} 51 \\ 48 \end{pmatrix} + a_1 \begin{pmatrix} 142 \\ 133 \end{pmatrix} = \begin{pmatrix} 63 \\ 69 \end{pmatrix}$ ➔ $(a_0, a_1) = (43, -15)$ "HI"

Learning with errors (LWE)

- Given uniform matrix $A \in Z_q^{k \times l}$
- Given "noise distribution" X
- Given samples $A \times s + e$, with vector $e \leftarrow X$
- Require find s



Quite easy!

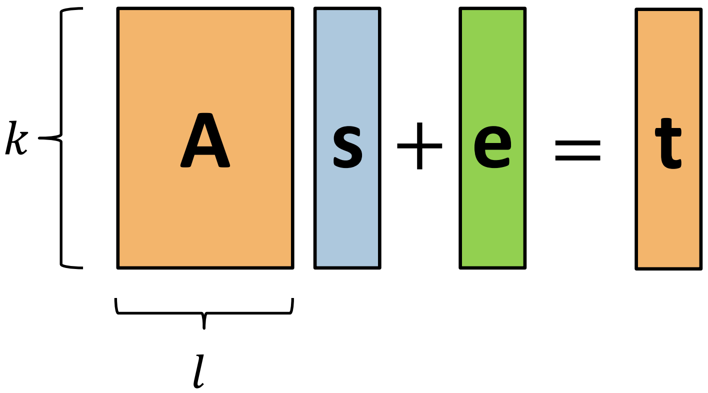


Hard-problem!

3. The case study of CRYSTAL-Kyber

LWE Encryption

Keys

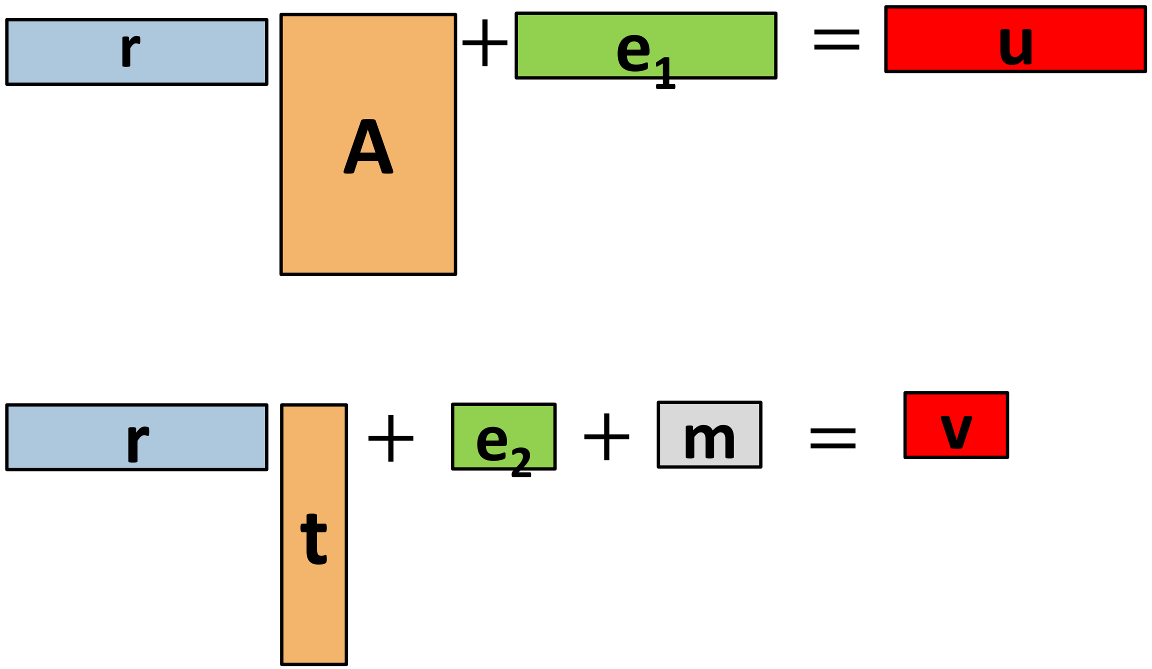


Random matrix A, small noises (s, e)

Public key ← (A, t)

Secret key ← s

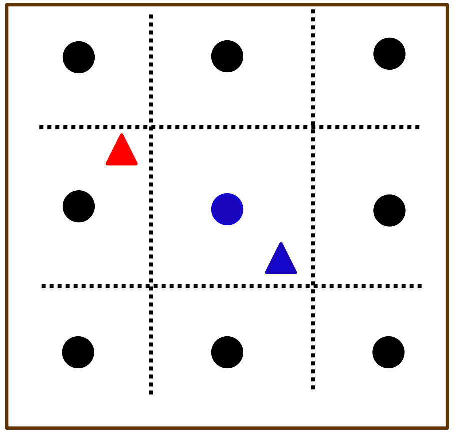
Encryption



LWE Encryption

Decryption

$$v - u\,s = m + \square$$



$$v = r\,(A\,s + e) + e_2 + m = u\,s + \square + m$$

3. The case study of CRYSTAL-Kyber

LWE and its variants

$$\begin{pmatrix} a_{00} & \cdots & a_{0n} \\ \vdots & \ddots & \vdots \\ a_{n0} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} s_0 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_0 \\ \vdots \\ e_n \end{pmatrix} = \begin{pmatrix} t_0 \\ \vdots \\ t_n \end{pmatrix}$$

- Learning with error
  - Storage: $O(n^2)$
  - Computation: $O(n^2)$

$$\begin{pmatrix} a_0 & a_n & \dots & a_1 \\ a_1 & a_0 & \dots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n-1} & \dots & a_0 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{pmatrix} = \begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_n \end{pmatrix}$$

- Ring-Learning with error
  - Storage: $O(n)$
  - Computation: $O(nlogn)$

$$\begin{pmatrix} A_{00}(X) & \cdots & A_{0k}(X) \\ \vdots & \ddots & \vdots \\ A_{k0}(X) & \cdots & A_{kk}(X) \end{pmatrix} \begin{pmatrix} s_0(X) \\ \vdots \\ s_k(X) \end{pmatrix} + \begin{pmatrix} e_0(X) \\ \vdots \\ e_k(X) \end{pmatrix} = \begin{pmatrix} t_0(X) \\ \vdots \\ t_k(X) \end{pmatrix}$$

- Module-Learning with error
  - Storage: $O(k^2n)$
  - Computation: $O(k^2nlogn)$

3. The case study of CRYSTAL-Kyber

CRYSTAL-Kyber

- Built on the difficulty of the M-LWE problem.

  Step 1: Chose a random matrix $\boldsymbol{A} \in R_q^{n\times k}$, a random small vector $\boldsymbol{s} \in R_q^k$, and a random small error $\boldsymbol{e} \in R_q^n$

  Step 2: Define $\boldsymbol{b} = \boldsymbol{A} \times \boldsymbol{s} + \boldsymbol{e}$

- All operation over the ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$, where $X^n + 1$ is the n-th cyclotomic polynomial.

Parameters for Round 3 – Kyber submission.

	Sec. Level	n	k	q	(η_1, η_2)	(d_u, d_v)	pk (B)	sk (B)	ct (B)
Kyber512	1	256	2	3329	(3,2)	(10,4)	800	1632	768
Kyber786	3	256	3	3329	(2,2)	(10,4)	1184	2400	1088
Kyber1024	5	256	4	3329	(2,2)	(11,5)	1568	3168	1568

[2] CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.02)
https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf

3. The case study of CRYSTAL-Kyber

CRYSTAL-Kyber

Algorithm 1. Kyber CPA Key Generation

Input: Random $d \in \{0,1\}^{256}$
$(\rho, \sigma) \leftarrow \text{SHA3-512}(d)$
$\hat{\boldsymbol{A}} \in R_q^{k \times k} \leftarrow \text{RejectionSampler}(\rho)$
$\boldsymbol{s} \in R_q^k \leftarrow \text{CBDSampler}_{\eta_1}(\sigma, 0)$
$\boldsymbol{e} \in R_q^k \leftarrow \text{CBDSampler}_{\eta_1}(\sigma, k)$
$\hat{\boldsymbol{s}} \leftarrow \text{NTT}(\boldsymbol{s})$
$\hat{\boldsymbol{e}} \leftarrow \text{NTT}(\boldsymbol{e})$
$\hat{\boldsymbol{t}} \leftarrow \hat{\boldsymbol{A}} \circ \hat{\boldsymbol{s}} + \hat{\boldsymbol{e}}$
return $(pk = (\rho, \text{Encode}_{12}(\hat{\boldsymbol{t}})),\ sk = \text{Encode}_{12}(\hat{\boldsymbol{s}}))$

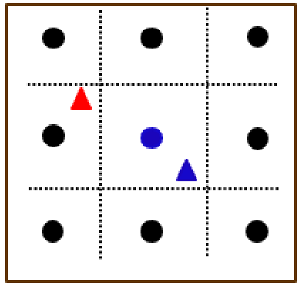
Algorithm 2. Kyber CPA Encryption

Input: $pk = (\rho, t_{enc})$, message $m \in \{0,1\}^{256}$, random $r \in \{0,1\}^{256}$
$\hat{\boldsymbol{t}} \leftarrow \text{Decode}_{12}(t_{enc})$
$\hat{\boldsymbol{A}} \in R_q^{k \times k} \leftarrow \text{RejectionSampler}(\rho)$
$\boldsymbol{r} \in R_q^k \leftarrow \text{CBDSampler}_{\eta_1}(r, 0)$
$\boldsymbol{e}_1 \in R_q^k \leftarrow \text{CBDSampler}_{\eta_2}(r, k)$
$e_2 \in R_q \leftarrow \text{CBDSampler}_{\eta_2}(r, 2k)$
$\hat{\boldsymbol{r}} \leftarrow \text{NTT}(\boldsymbol{r})$
$\boldsymbol{u} \leftarrow \text{NTT}^{-1}(\hat{\boldsymbol{A}}^T \circ \hat{\boldsymbol{r}}) + \boldsymbol{e}_1$
$v \leftarrow \text{NTT}^{-1}(\hat{\boldsymbol{t}}^T \circ \hat{\boldsymbol{r}}) + e_2 + \text{Decompress}_q(\text{Decode}_1(m), 1)$
return $c = (\text{Encode}_{d_u}(\text{Compress}_q(u, d_u)), \text{Encode}_{d_v}(\text{Compress}_q(v, d_v)))$

Algorithm 3. Kyber CPA Decryption

Input: $sk = (\hat{\boldsymbol{s}})$, ciphertext $c = (c_1, c_2)$
$\boldsymbol{u} \leftarrow \text{Decompress}_q(\text{Decode}_{d_u}(c_1), d_u)$
$v \leftarrow \text{Decompress}_q(\text{Decode}_{d_v}(c_2), d_v)$
$\hat{\boldsymbol{s}} \leftarrow \text{Decode}_{12}(sk)$
$m \in \{0,1\}^{256} \leftarrow \text{Encode}_1(\text{Compress}_q(v - \text{NTT}^{-1}(\hat{\boldsymbol{s}}^T \circ \text{NTT}(\boldsymbol{u})), 1))$
return m

$$
\begin{aligned}
& v - s^T \times u \\
&= (t^T \times r + e_2 + m') - \left(s^T \times (A^T \times r + e_1)\right) \\
&= \left((A \times s + e)^T \times r + e_2 + m'\right) - \left(s^T \times (A^T \times r + e_1)\right) \\
&= (A \times s)^T \times r + e^T \times r + e_2 + m' - s^T \times A^T \times r - s^T \times e_1 \\
&= m' + (\boldsymbol{e^T} \times \boldsymbol{r} - \boldsymbol{s^T} \times \boldsymbol{e_1} + \boldsymbol{e_2})
\end{aligned}
$$



3. The case study of CRYSTAL-Kyber

»» "Small" Kyber

- ❑ Assuming $R_{q=17} = \mathbb{Z}_{17}[X]/(X^4+1)$.

Keys

$$\boldsymbol{s} = (-x^3 - x^2 + x, -x^3 - x), \qquad \boldsymbol{e} = (x^2, x^2 - x)$$

$$\boldsymbol{A}_{\mathbf{2}\times\mathbf{2}} = \begin{pmatrix} 6x^3 + 16x^2 + 16x + 11 & 9x^3 + 4x^2 + 6x + 3 \\ 5x^3 + 3x^2 + 10x + 1 & 6x^3 + 1x^2 + 9x + 15 \end{pmatrix}$$

$$\boldsymbol{t} = \boldsymbol{As} + \boldsymbol{e} = (16x^3 + 15x^2 + 7 \qquad 10x^3 + 12x^2 + 11x + 16)$$

- Private key: **s**
- Public key: **(A,t)**

“Small” Kyber

- Assuming $R_{q=17} = \mathbb{Z}_{17}[X]/(X^4+1)$.

Encryption

$$\boldsymbol{r} = (-x^3 + x^2 + x \qquad x^3 + x^2 - 1),\ e_1 = (x^2 + x \qquad x^2),\ \ e_2 = -x^3 - x^2$$

$$Message: m_b = (1011)_2 = x^3 + x + 1$$

$$Up - scale: m = \left\lfloor \frac{q}{2} \right\rceil m_b = 9x^3 + 9x + 9$$

$$\begin{cases} \boldsymbol{u} = A^T r + e_1 = (11x^3 + 11x^2 + 10x + 3 \qquad 4x^3 + 4x^2 + 13x + 11) \\ \boldsymbol{v} = t^T r + e_2 + m = 7x^3 + 6x^2 + 8x + 5 \end{cases}$$

3. The case study of CRYSTAL-Kyber

“Small” Kyber

- Assuming $R_{q=17} = \mathbb{Z}_{17}[X]/(X^4 + 1)$.

Decryption

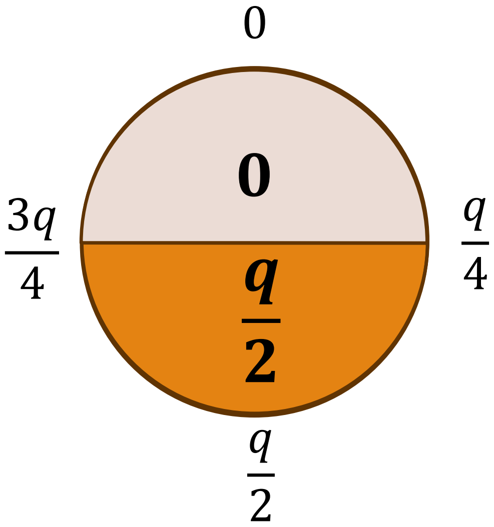
$$\boldsymbol{r} = (-x^3 + x^2 + x, x^3 + x^2 - 1),\ \ e_1 = (x^2 + x, x^2),\ \ e_2 = -x^3 - x^2$$

Calculate: $m' = v - s^T \times u = m + (\boldsymbol{e^T} \times \boldsymbol{r} - \boldsymbol{s^T} \times \boldsymbol{e_1} + \boldsymbol{e_2})$

$$= 7x^3 + 14x^2 + 7x + 5$$

$$\approx 9x^3 + 0x^2 + 9x + 9$$

Down-scale: $m = \frac{1}{9}m' = x^3 + 0x^2 + x + 1 = (1011)_2$



3. The case study of CRYSTAL-Kyber

Hardware implementation

Algorithm 1. Kyber CPA Key Generation

Input: Random $d \in \{0,1\}^{256}$
$(\rho, \sigma) \leftarrow \text{SHA3-512}(d)$
$\hat{\boldsymbol{A}} \in R_q^{k \times k} \leftarrow \text{RejectionSampler}(\rho)$
$\boldsymbol{s} \in R_q^k \leftarrow \text{CBDSampler}_{\eta_1}(\sigma, 0)$
$\boldsymbol{e} \in R_q^k \leftarrow \text{CBDSampler}_{\eta_1}(\sigma, k)$
$\hat{\boldsymbol{s}} \leftarrow \text{NTT}(\boldsymbol{s})$
$\hat{\boldsymbol{e}} \leftarrow \text{NTT}(\boldsymbol{e})$
$\hat{\boldsymbol{t}} \leftarrow \hat{\boldsymbol{A}} \circ \hat{\boldsymbol{s}} + \hat{\boldsymbol{e}}$
return $(pk{=}(\rho, \text{Encode}_{12}(\hat{\boldsymbol{t}})),\ sk{=}\text{Encode}_{12}(\hat{\boldsymbol{s}}))$

Algorithm 2. Kyber CPA Encryption

Input: $pk = (\rho, t_{enc})$, message $m \in \{0,1\}^{256}$, random $r \in \{0,1\}^{256}$
$\hat{\boldsymbol{t}} \leftarrow \text{Decode}_{12}(t_{enc})$
$\hat{\boldsymbol{A}} \in R_q^{k \times k} \leftarrow \text{RejectionSampler}(\rho)$
$\boldsymbol{r} \in R_q^k \leftarrow \text{CBDSampler}_{\eta_1}(r, 0)$
$\boldsymbol{e_1} \in R_q^k \leftarrow \text{CBDSampler}_{\eta_2}(r, k)$
$e_2 \in R_q \leftarrow \text{CBDSampler}_{\eta_2}(r, 2k)$
$\hat{\boldsymbol{r}} \leftarrow \text{NTT}(\boldsymbol{r})$
$\boldsymbol{u} \leftarrow \text{NTT}^{-1}(\hat{\boldsymbol{A}}^T \circ \hat{\boldsymbol{r}}) + \boldsymbol{e_1}$
$v \leftarrow \text{NTT}^{-1}(\hat{\boldsymbol{t}}^T \circ \hat{\boldsymbol{r}}) + e_2 + \text{Decompress}_q(\text{Decode}_1(m), 1)$
return $c = (\text{Encode}_{d_u}(\text{Compress}_q(u, d_u)), \text{Encode}_{d_v}(\text{Compress}_q(v, d_v)))$

Algorithm 3. Kyber CPA Decryption

Input: $sk = (\hat{\boldsymbol{s}})$, ciphertext $c = (c_1, c_2)$
$\boldsymbol{u} \leftarrow \text{Decompress}_q(\text{Decode}_{d_u}(c_1), d_u)$
$v \leftarrow \text{Decompress}_q(\text{Decode}_{d_v}(c_2), d_v)$
$\hat{\boldsymbol{s}} \leftarrow \text{Decode}_{12}(sk)$
$m \in \{0,1\}^{256} \leftarrow \text{Encode}_1(\text{Compress}_q(v - \text{NTT}^{-1}(\hat{\boldsymbol{s}}^T \circ \text{NTT}(\boldsymbol{u})), 1))$
return m

- ❑ Randomness generation (Kekkak-SHA-3)
- ❑ **Polynomial operations: multiplication**

NTT-based polynomial multiplication

- Number Theoretic Transform (NTT)

$$\boldsymbol{h} = \boldsymbol{f} \cdot \boldsymbol{g} = \boldsymbol{NTT}^{-1}(\boldsymbol{NTT}(\boldsymbol{f}) \circ \boldsymbol{NTT}(\boldsymbol{g}))$$

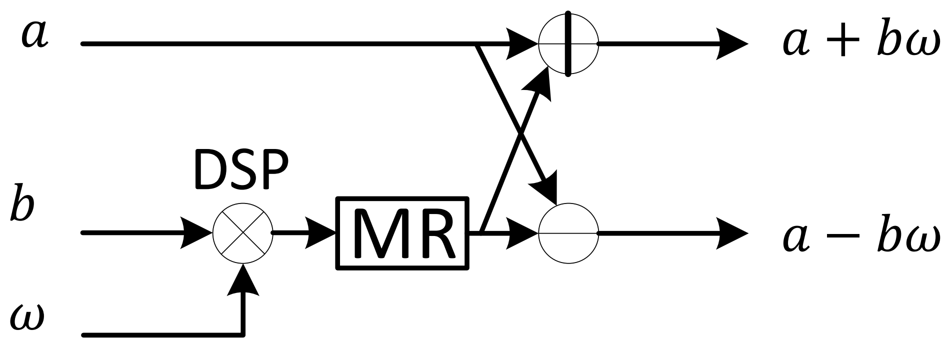
$$\hat{f}_i = \sum_{j=0}^{n-1} f_j \omega^{ij} \ (mod\ q)$$

$$\hat{g}_i = \sum_{j=0}^{n-1} g_j \omega^{ij} \ (mod\ q)$$

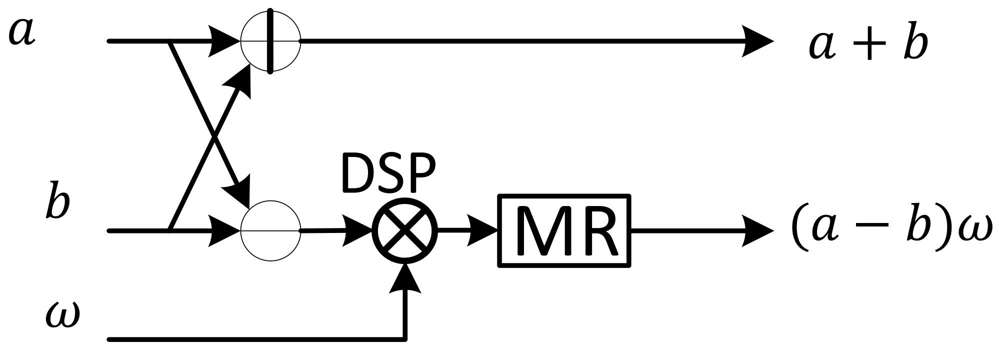
PWM

$$h_i = n^{-1} \sum_{j=0}^{n-1} \hat{f}_j \omega^{-ij} \ (mod\ q)$$

- Fast NTT algorithms



(a) Cooley-Tukey (CT)



(b) Gentlemen-Sande (GS)

4. Accelerating by Number Theoretic Transform

All operands are modular arithmetic

❑ Start point

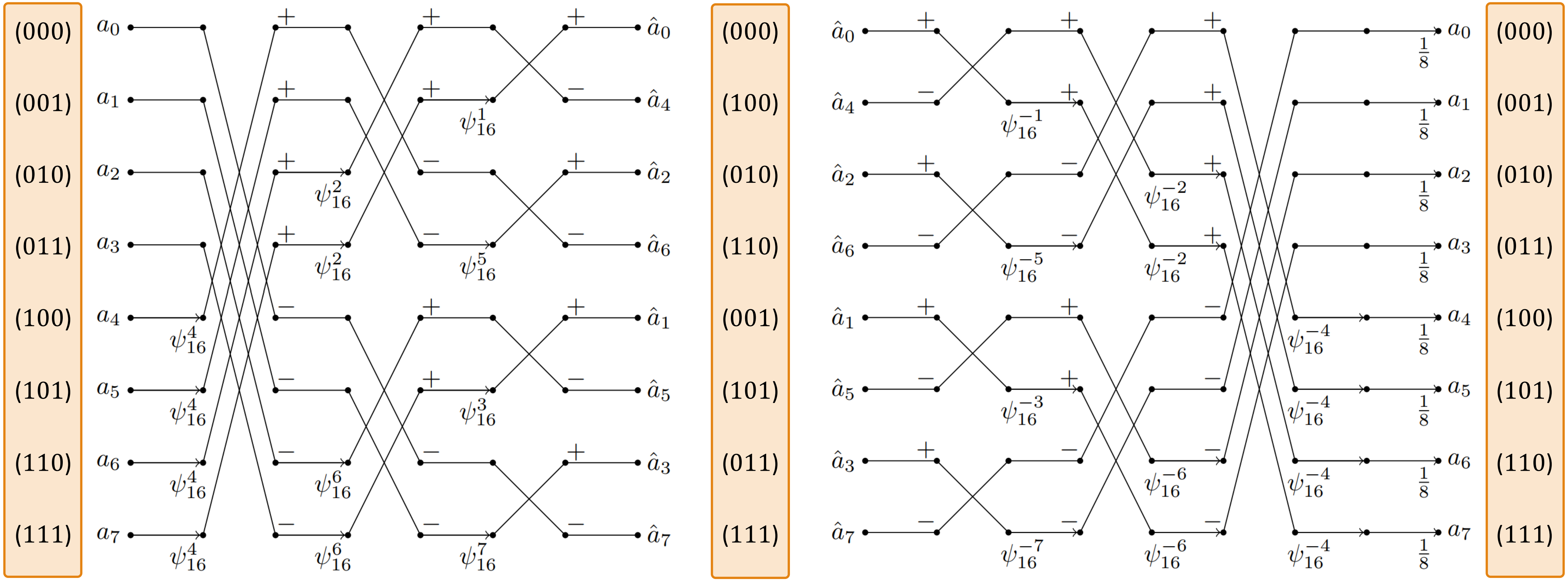
Multiplication modulus q expression

$$a \times b = c \equiv x \bmod q \qquad (0 \le a, b, x < q;\ 0 \le c < q^2)$$

Methods	Configuration	Notes
Barret reduction	Soft-hardware	Requires more multiplications
Montgomery reduction	Soft-hardware	Montgomery domain, requires more multiplications
K-Red reduction	Full hardware	Fast & low hardware resources, limited in NTT operations
The special form of $q = 2^{12} - 2^9 - 2^8 + 1$ $\Rightarrow\ 2^{12} \equiv 2^9 - 2^8 + 1\ (mod\ q)$	Full hardware	Combinational logics, complex design

4. Accelerating by Number Theoretic Transform

NTT transformations



(a) $NTT^{CT}_{No\to Bo}$ (b) $INTT^{GS}_{Bo\to No}$

4. Accelerating by Number Theoretic Transform

Iterative NTT accelerator

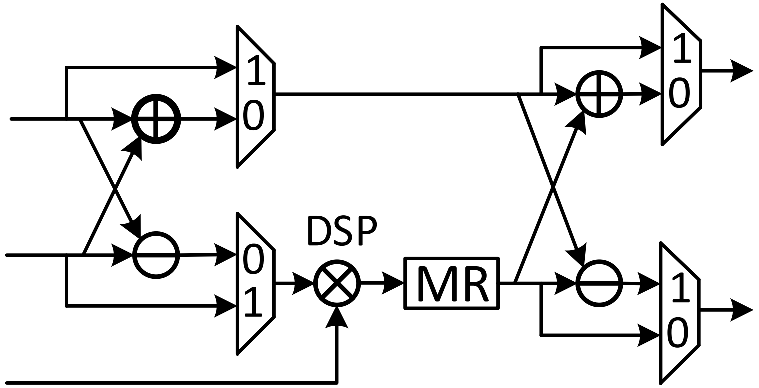
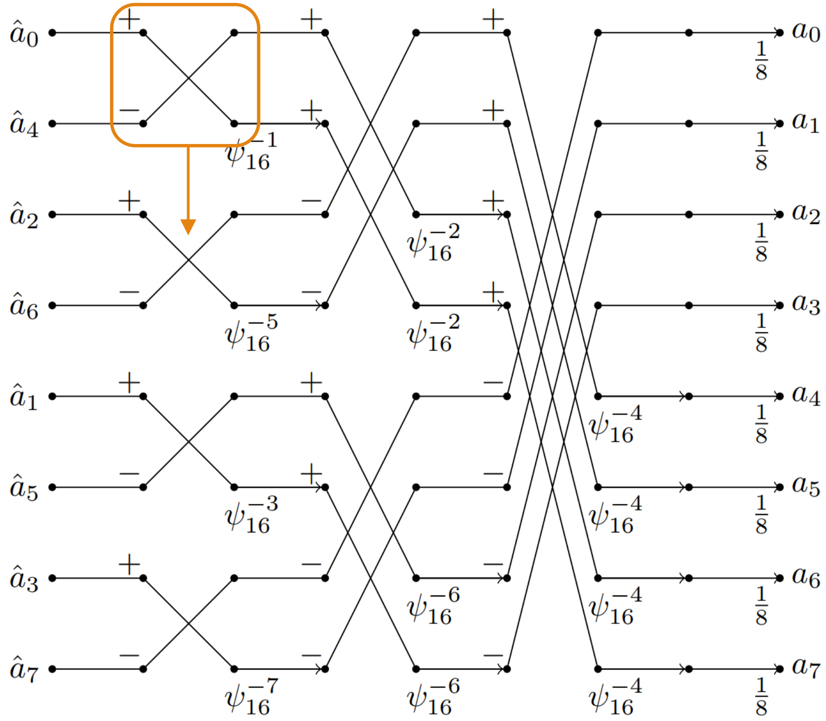


Fig. Unified butterfly Configuration (CT+GS)

- Avoid zero padding in NTT processes, [3]
- Merging pre- & post-processing [4],[5]



[3] Lyubashevsky, et. al. "**A Modest Proposal for FFT Hashing**". In Proceedings of the Fast Software Encryption (FSE), Switzerland, 10–13 February 2008; pp. 54–72.
[4] Roy, S.S. et al. "**Compact Ring-LWE Cryptoprocessor**". (CHES), Busan, South Korea, 23–26 September 2014; pp. 371–39
[5] Pöppelmann, T et. al. "**High-Performance Ideal Lattice-Based Cryptography on 8-Bit ATxmega Microcontrollers**." In Proceedings of the Progress in Cryptology (LATINCRYPT), Guadalajara, Mexico, 23–26 August 2015; pp. 346–365

Iterative NTT accelerator

- ❑ **Lightweight**: single butterfly core [6]
- ❑ **Balance**: 2x1, 2x2 butterfly cores [7][8]
- ❑ **High-performance**: 16, 32 butterfly core [9] []

<u>The drawback</u>

- ❑ Requires temporary memory.
- ❑ Complex memory access patten.

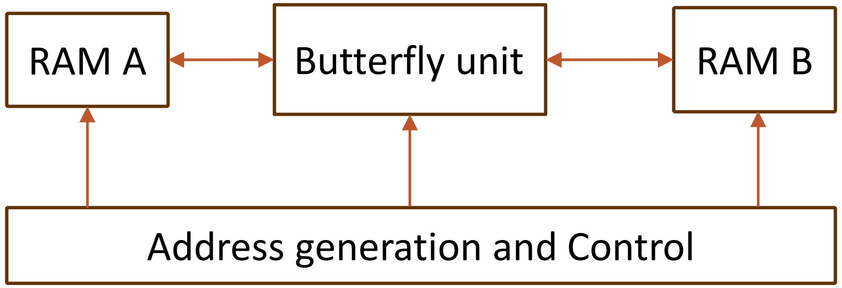
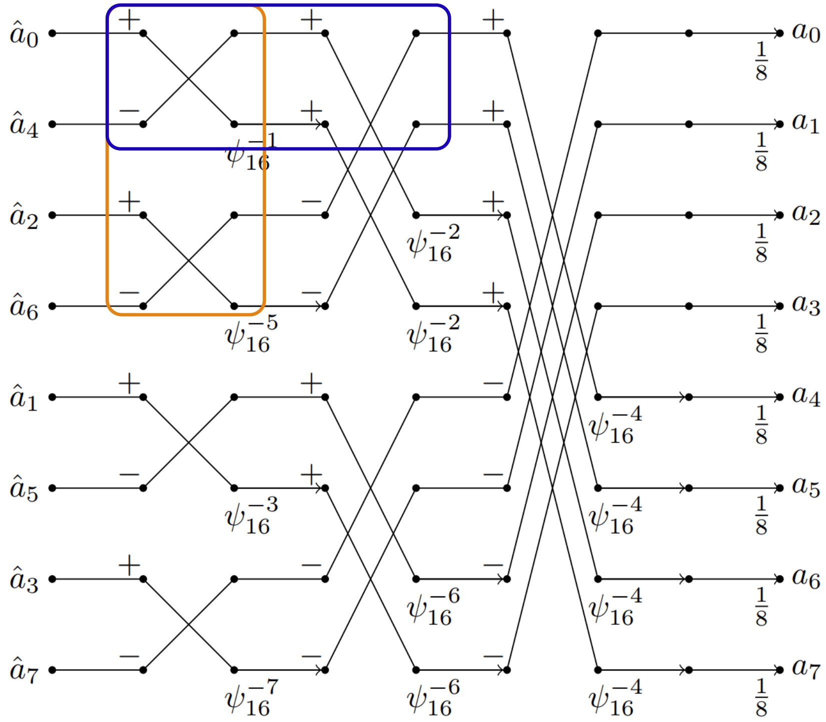


Fig. Ping-pong memory scheme [6]



[11] N. Gupta, et. al. **"Lightweight hardware accelerator for post-quantum digital signature crystals-dilithium,"** IEEE TCAS I: Regular Papers, 2023.

Iterative NTT accelerator

[6] Y. Xing et. al., "**A compact hardware implementation of cca-secure key exchange mechanism crystals-kyber on fpga,"** IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 328–356, 2021.

[7] M. Bisheh-Niasar, et. al., **"Instruction-set accelerated implementation of crystals-kyber,"** IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 11,pp. 4648–4659, 2021.

[8] V. B. Dang, et al., **"High-speed hardware architectures and fpga benchmarking of crystals-kyber, ntru, and saber,"** IEEE Transactions on Computers, vol. 72, no. 2, pp. 306–320, 2022.

[9] F. Yaman, et al., **"A hardware accelerator for polynomial multiplication operation of crystals-kyber pqc scheme,"** in 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2021, pp. 1020–1025.

[10] Y. Geng, et al., **"Rethinking Parallel Memory Access Pattern in Number Theoretic Transform Design,"** in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1689-1693, May 2023, doi: 10.1109/TCSII.2023.3260811.

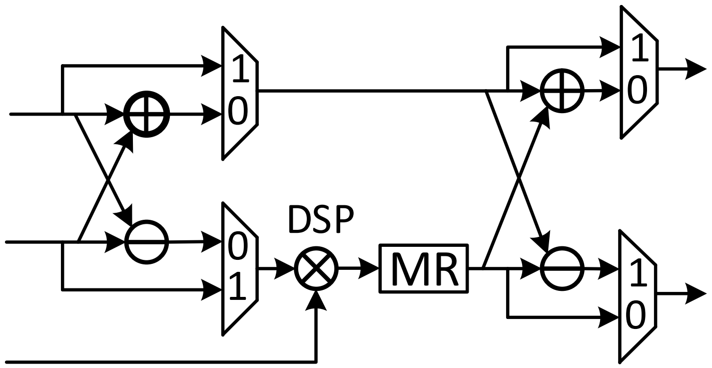
4. Accelerating by Number Theoretic Transform

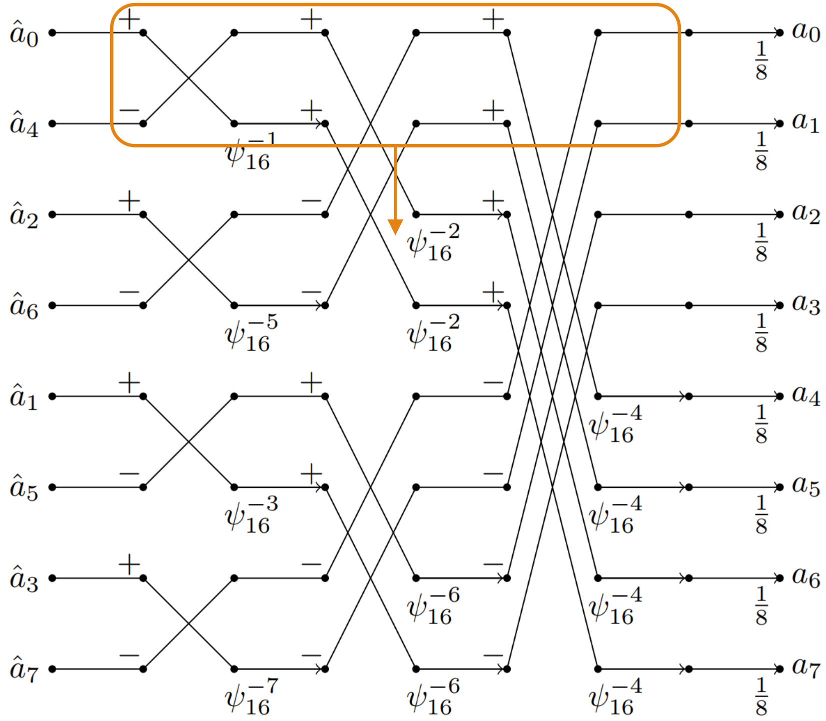
Pipelined NTT accelerator

- ❑ The straight forward control pattern
- ❑ High-performance
- ❑ Free temporary memory

The drawback

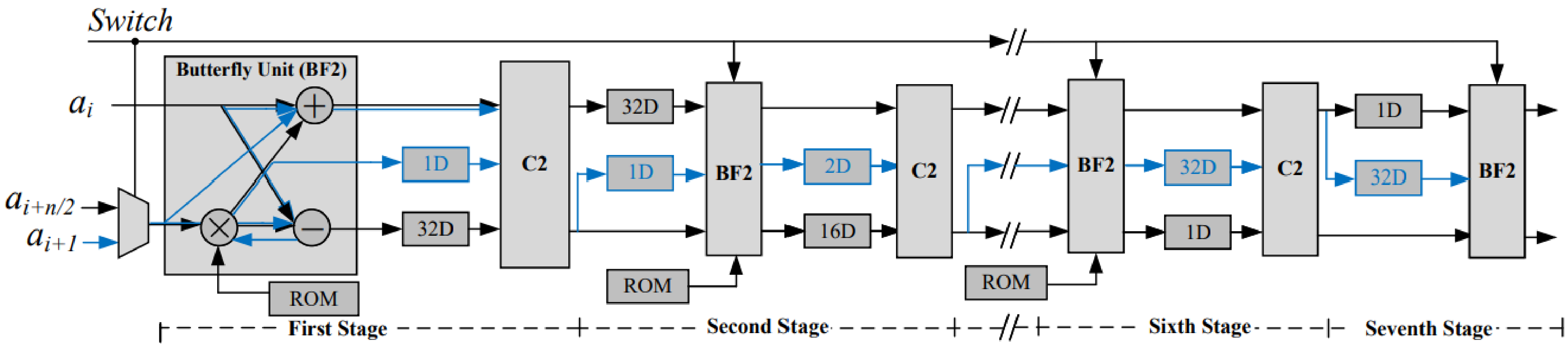
- ❑ Requires $\log n$ butterfly cores for n-degree polynomial
- ❑ Double the number of re-order unit.





4. Accelerating by Number Theoretic Transform

Pipelined NTT Architectures



The radix-2 Multipath Delay Commutator NTT/INTT pipelined architecture for CRYSTAL-Kyber, [7]

[12] Z. Ni, et. al., **"HPKA: A High-Performance CRYSTALS-Kyber Accelerator Exploring Efficient Pipelining,"** in *IEEE Transactions on Computers 2023*, doi: 10.1109/TC.2023.3296899.

5. Discussion

- Introduction about lattice-based cryptography
- Lattice hard problems
- CRYSTAL-Kyber case study
- NTT-based Polynomial Multiplication Hardware implementations

Thank you for your listening

NGUYEN TRONG HUNG[D1]
(グエン　チョン　フン)