Ciphertext-Ciphertext Matrix Multiplication: Fast for Large Matrices

Jai Hyun Park

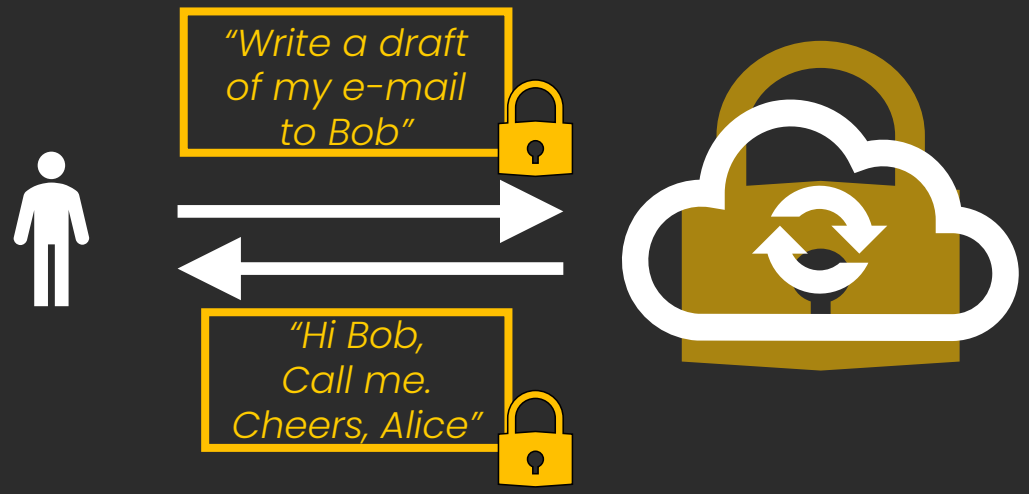
jaihyunp@gmail.com

Madrid, Spain – May 6, 2025

Summary

- Fast ciphertext-ciphertext matrix multiplication (CCMM)
  - 85.2 s for CCMM of 4096×4096 matrices in a single thread CPU
  - How? Reduce CCMM to plaintext matrix multiplications
- Fast ciphertext matrix transpose (CMT)
  - 0.76 s for CMT of a 2048×2048 matrices in a single thread CPU
- Lightweight CCMM and CMT algorithms with smaller key sizes

Matrix Multiplication on Encrypted Data



- PPMM: plaintext-plaintext matrix multiplication
- PCMM: plaintext-ciphertext matrix multiplication
- CCMM: ciphertext-ciphertext matrix multiplication

- PCMMs and CCMMs with diverse dimensions
  - e.g., PCMM of dimension 128 ~ 16384 for GPT-3.5

PPMM vs. PCMM vs. CCMM

- PPMM – BLAS libraries
  - highly optimized open libraries
  - Can be 30x faster than a naïve implementation

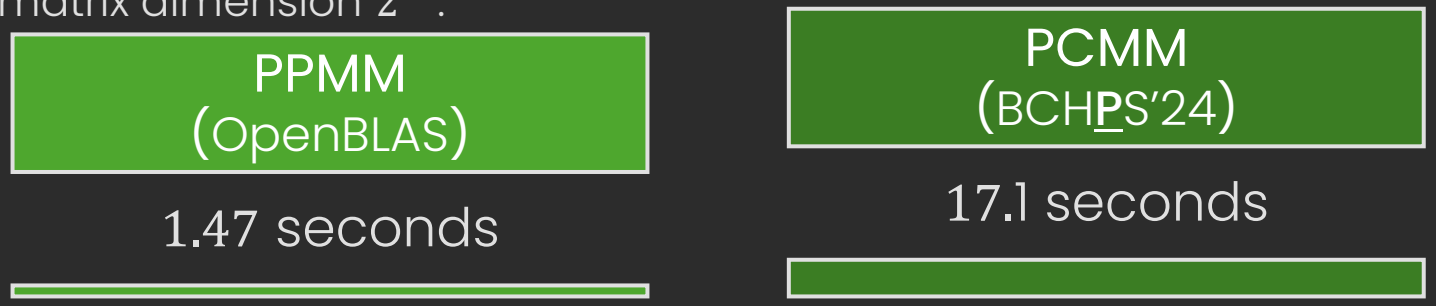
For matrix dimension 2^{12}:

PPMM (OpenBLAS)
1.47 seconds

PPMM vs. PCMM vs. CCMM

- PPMM – BLAS libraries
  - highly optimized open libraries
  - Can be 30x faster than a naïve implementation
- PCMM – BCHPS'24
  - Reduction from PCMM to PPMM
  - Optimizations with shared-a, truncation, and others

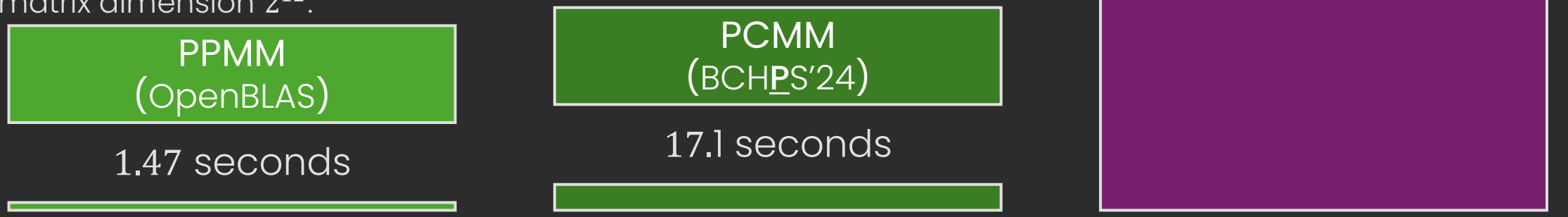
For matrix dimension 2^{12}:



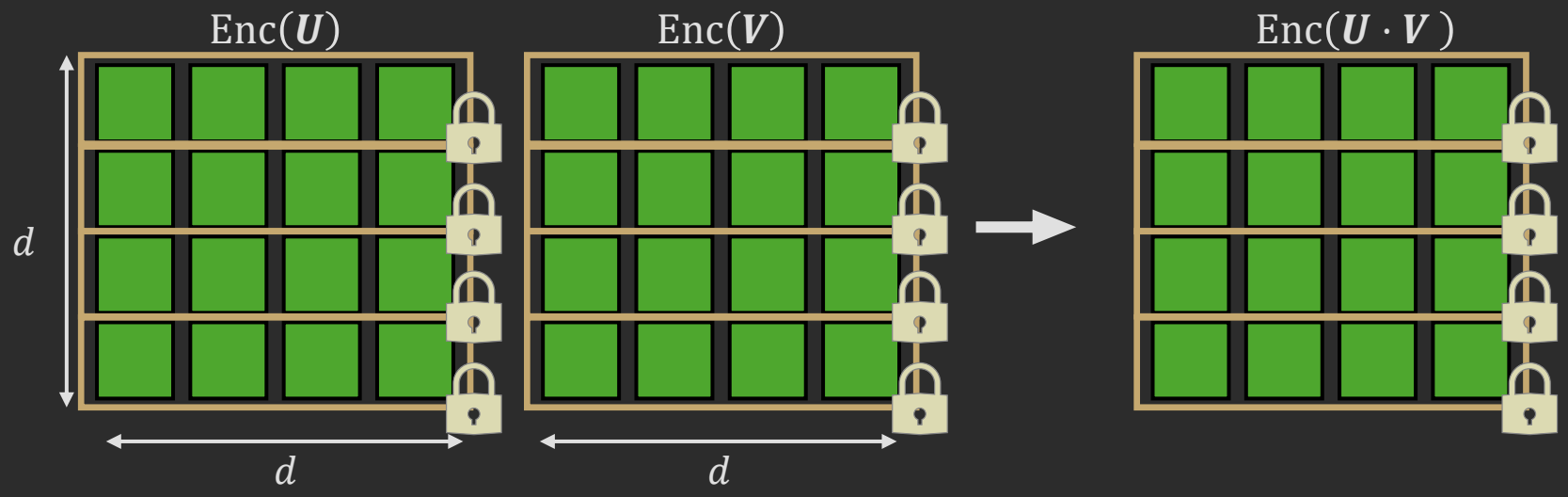
PPMM vs. PCMM vs. CCMM

- PPMM – BLAS libraries
  - highly optimized open libraries
  - Can be 30x faster than a naïve implementation
- PCMM – BCHPS'24
  - Reduction from PCMM to PPMM
  - Optimizations with shared-a, truncation, and others
- CCMM – JKLS'18
  - cubic bit complexity
  - 0.6 seconds for matrix dimension 64

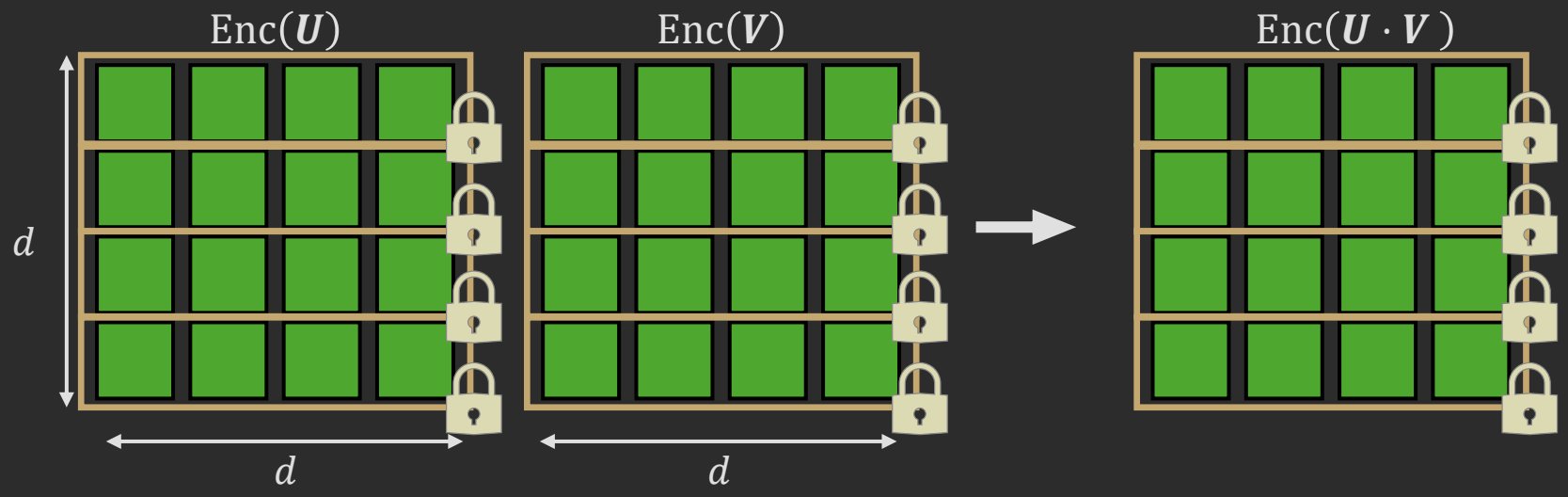
For matrix dimension 2^{12}:



Ciphertext-Ciphertext Matrix Multiplication (CCMM)



Ciphertext-Ciphertext Matrix Multiplication (CCMM)



- CCMM with RLWE-based (fully) homomorphic encryption schemes
  - Compatibility with the other machine learning tasks
  - High efficiency

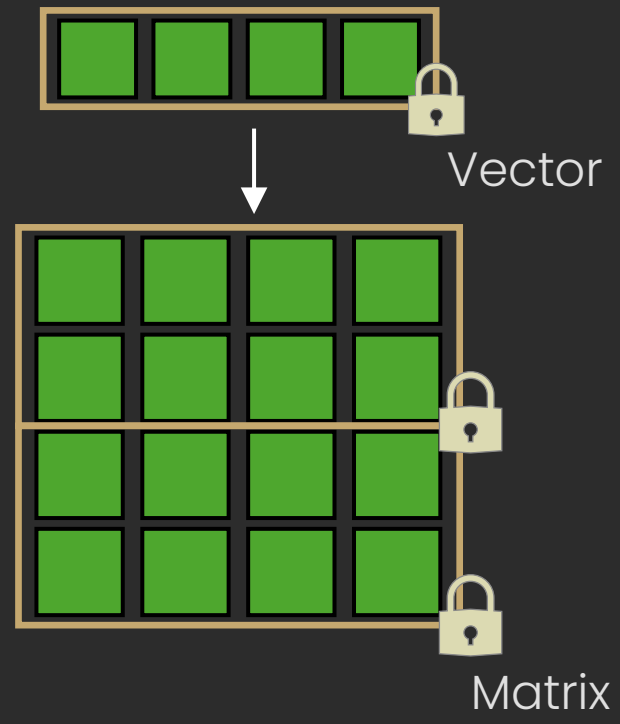
Encrypted Matrix Multiplication with CKKS

- CKKS
  - Plaintext: vector of real numbers
  - Native operations: // add, // mult, and rotate

Vector

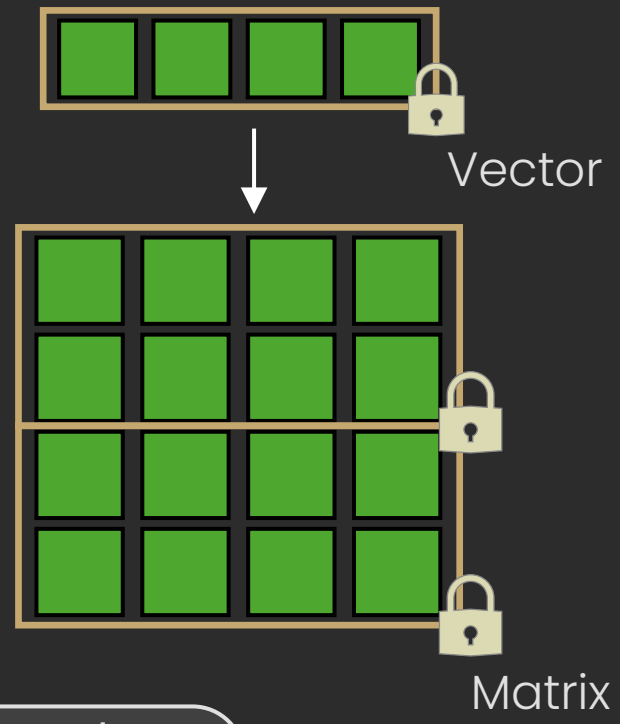
Encrypted Matrix Multiplication with CKKS

- CKKS
  - Plaintext: vector of real numbers
  - Native operations: // add, // mult, and rotate
- With the native operations, PCMM/CCMM requires lots of rotates
  - For example, [JKLS'18] has a cubic bit complexity, but is orders of magnitude slower than PPMM



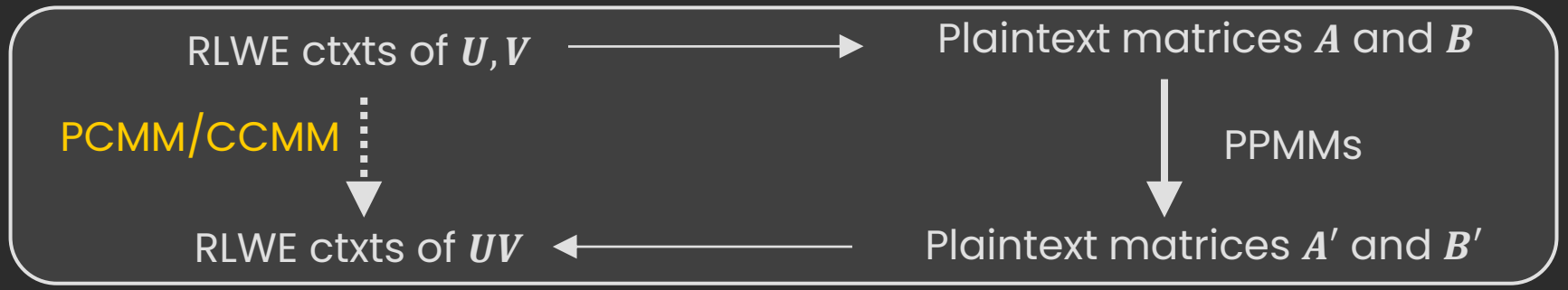
Encrypted Matrix Multiplication with CKKS

- CKKS
  - Plaintext: vector of real numbers
  - Native operations: // add, // mult, and rotate
- With the native operations, PCMM/CCMM requires lots of rotates
  - For example, [JKLS'18] has a cubic bit complexity, but is orders of magnitude slower than PPMM



Q. How to utilize PPMM BLAS libraries?

A. Reduction from PCMM/CCMM to PPMM



RLWE-based Encryption of Matrices

- In the ring $\mathbb{Z}_Q[X]/(X^N+1)$, an RLWE ciphertext $(a, b = -a \cdot s + m)$ is:

$$[a_0 \quad a_1 \quad \cdots \quad a_{N-1}] \begin{bmatrix} s_0 & s_1 & \cdots & s_{N-1} \\ -s_{N-1} & s_0 & \cdots & s_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ -s_1 & -s_2 & \cdots & s_0 \end{bmatrix} + [b_0 \quad b_1 \quad \cdots \quad b_{N-1}] = [m_0 \quad m_1 \quad \cdots \quad m_{N-1}]$$

$\mathrm{Toep}(s)$

✓ a_i, b_i, s_i, m_i are coeffs of a, b, s, m

RLWE-based Encryption of Matrices

- In the ring $\mathbb{Z}_Q[X]/(X^N+1)$, an RLWE ciphertext $(a, b = -a \cdot s + m)$ is:

$$[a_0 \quad a_1 \quad \cdots \quad a_{N-1}] \begin{bmatrix} s_0 & s_1 & \cdots & s_{N-1} \\ -s_{N-1} & s_0 & \cdots & s_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ -s_1 & -s_2 & \cdots & s_0 \end{bmatrix} + [b_0 \quad b_1 \quad \cdots \quad b_{N-1}] = [m_0 \quad m_1 \quad \cdots \quad m_{N-1}]$$

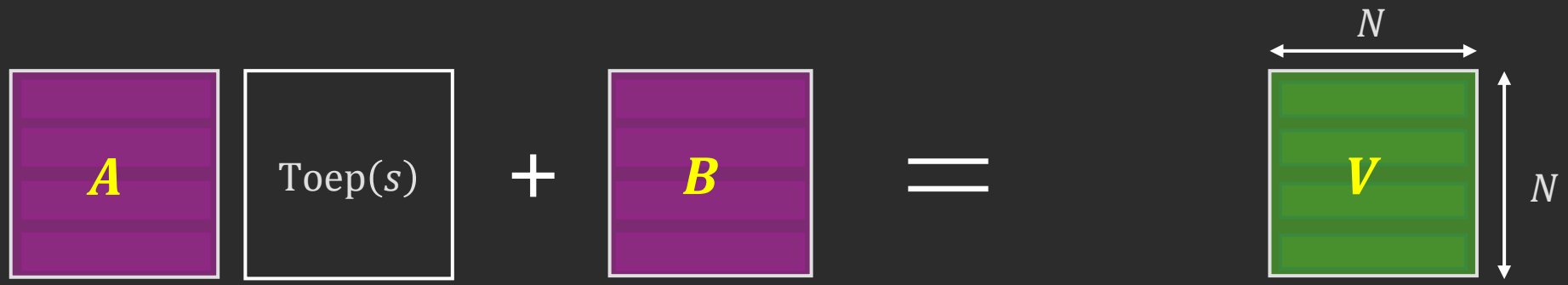
$\mathrm{Toep}(s)$

✓ a_i, b_i, s_i, m_i are coeffs of a, b, s, m

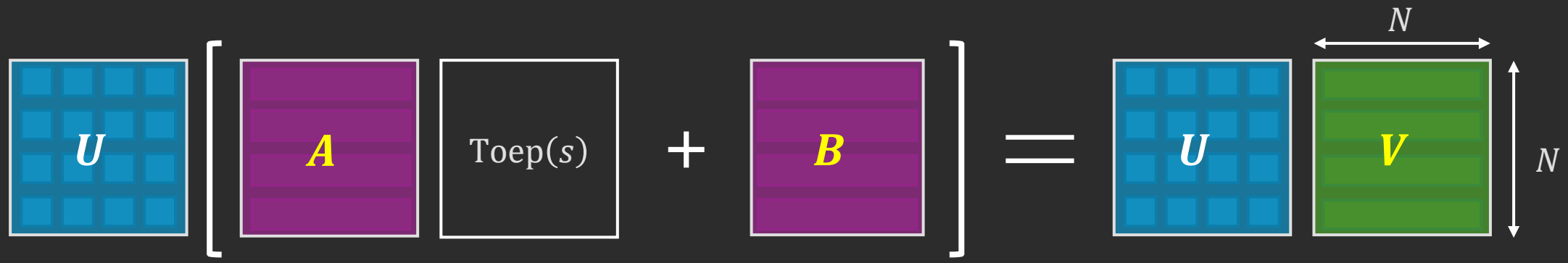
- N RLWE ciphertexts are:

$$\begin{bmatrix} \text{—} & a_1^t & \text{—} \\ \text{—} & a_2^t & \text{—} \\ & \boldsymbol{A} & \\ & \vdots & \\ \text{—} & a_N^t & \text{—} \end{bmatrix} \begin{bmatrix} s_0 & s_1 & \cdots & s_{N-1} \\ -s_{N-1} & s_0 & \cdots & s_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ -s_1 & -s_2 & \cdots & s_0 \end{bmatrix} + \begin{bmatrix} \text{—} & b_1^t & \text{—} \\ \text{—} & b_2^t & \text{—} \\ & \boldsymbol{B} & \\ & \vdots & \\ \text{—} & b_N^t & \text{—} \end{bmatrix} = \begin{bmatrix} \text{—} & m_1^t & \text{—} \\ \text{—} & m_2^t & \text{—} \\ & \boldsymbol{M} & \\ & \vdots & \\ \text{—} & m_N^t & \text{—} \end{bmatrix}$$

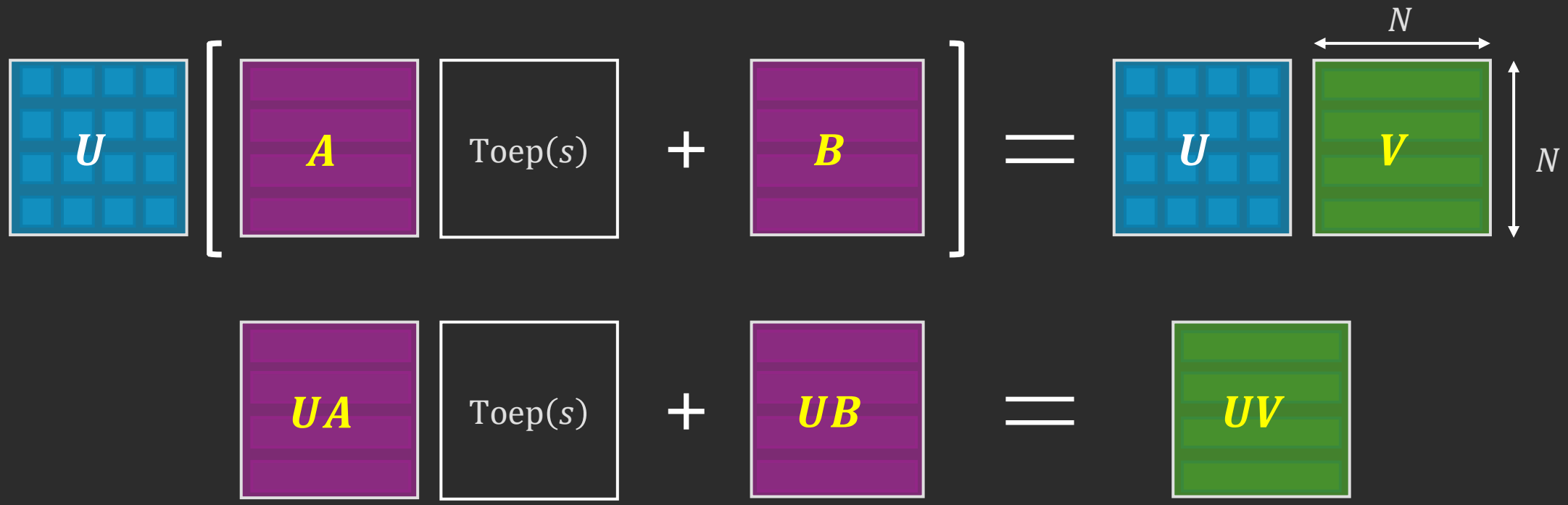
PCMM ≤ PPMMs (BCHPS'24, LZ'22)



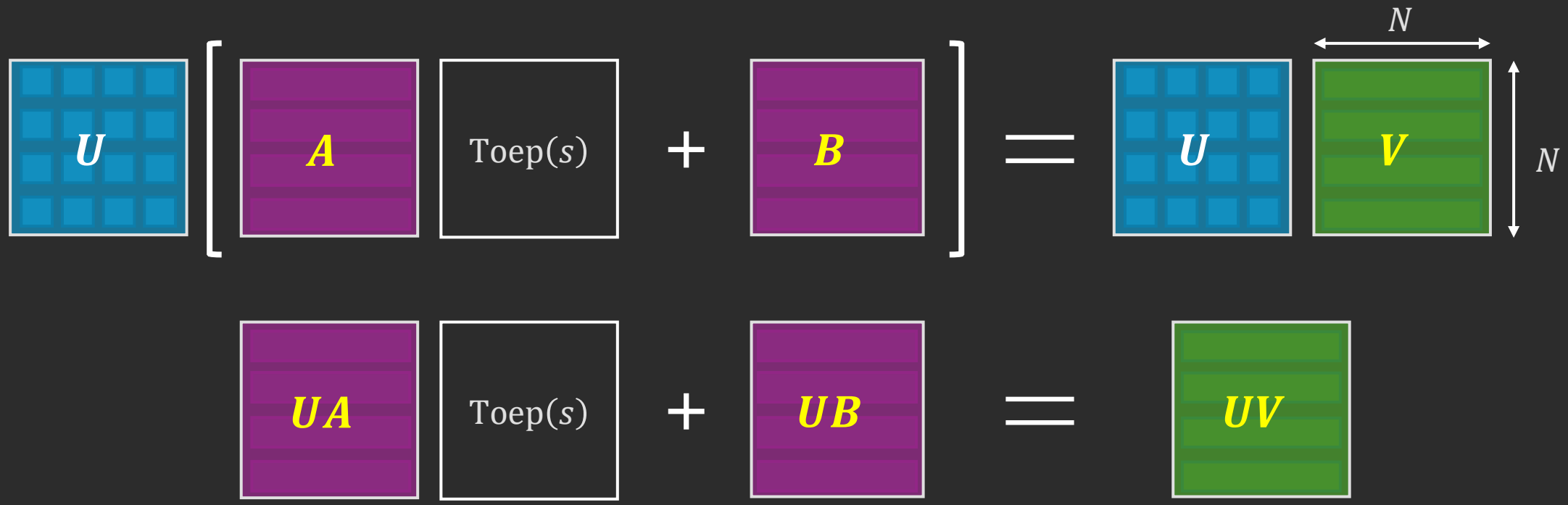
PCMM ≤ PPMMs (BCHPS'24, LZ'22)



PCMM ≤ PPMMs (BCHPS'24, LZ'22)

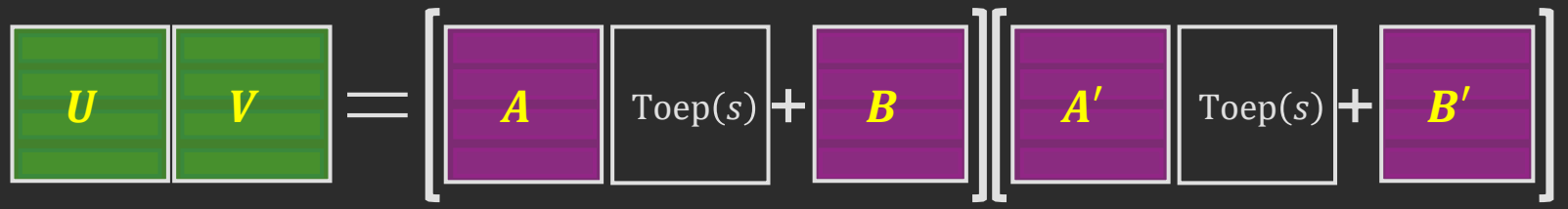


PCMM ≤ PPMMs (BCHPS'24, LZ'22)

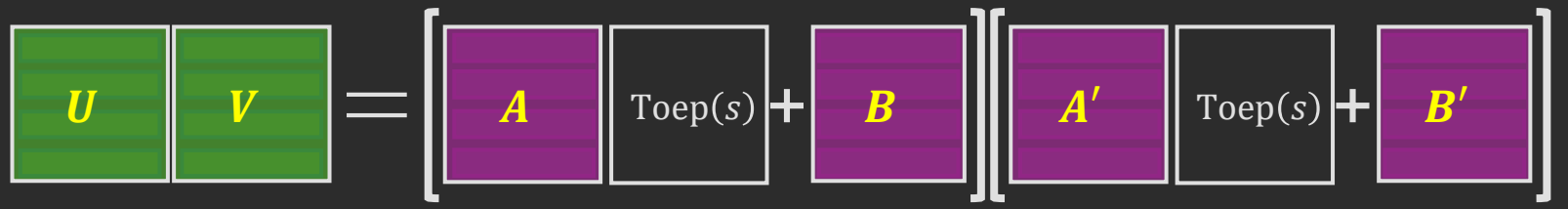


- $N \times N \times N$ PCMM ≤ *two* $N \times N \times N$ PPMMs *modulo* Q
- We use fast PPMM BLAS libraries for $N \times N \times N$ PCMM
- For PCMMs with other dimensions, see BCHPS'24

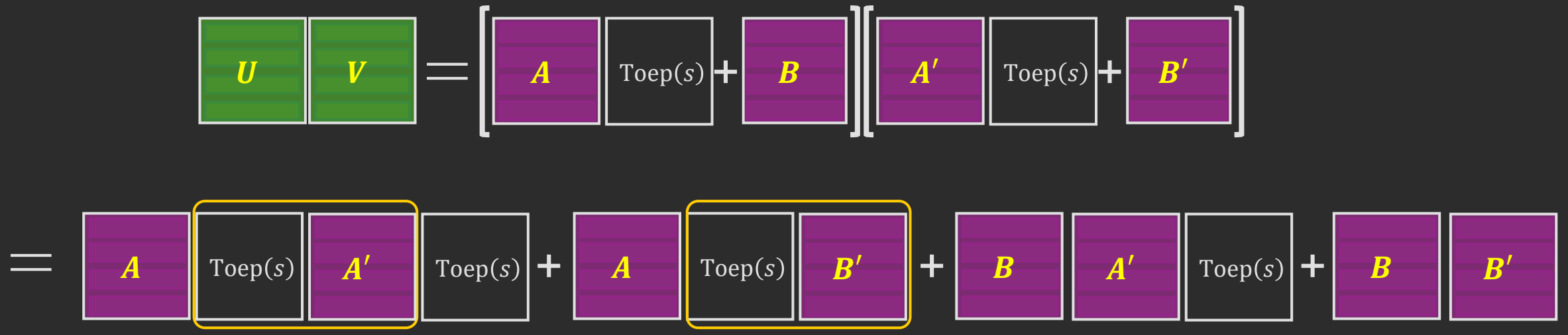
CCMM?



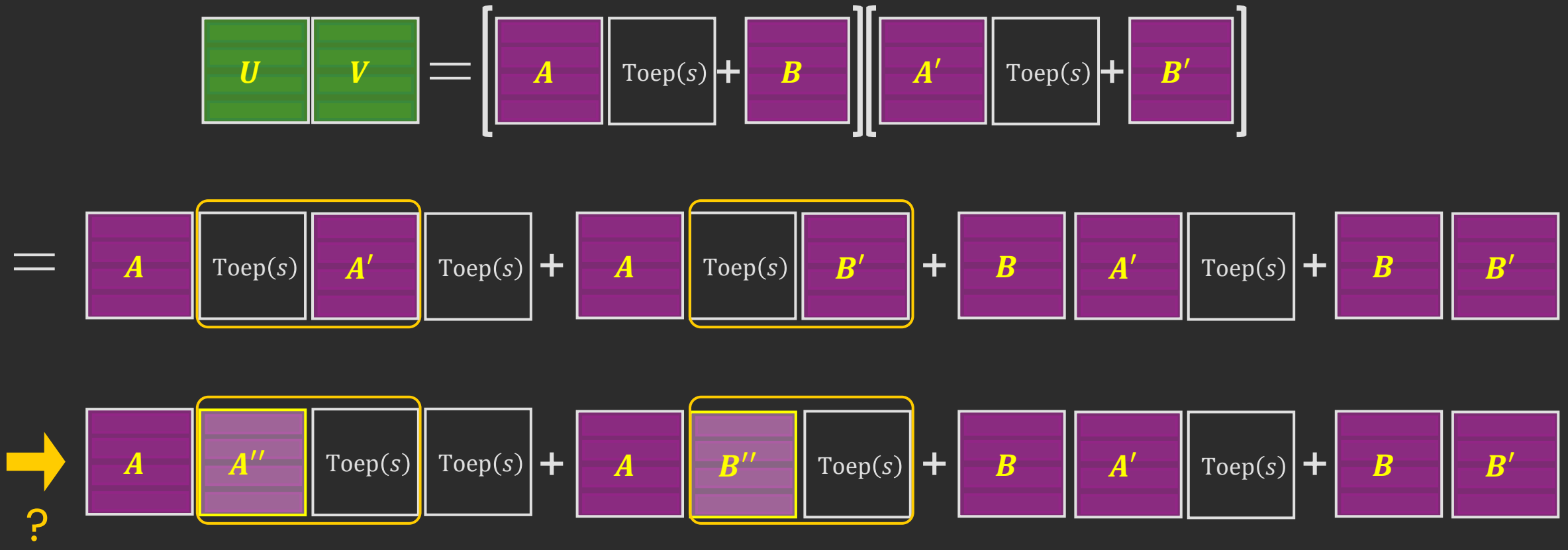
CCMM?



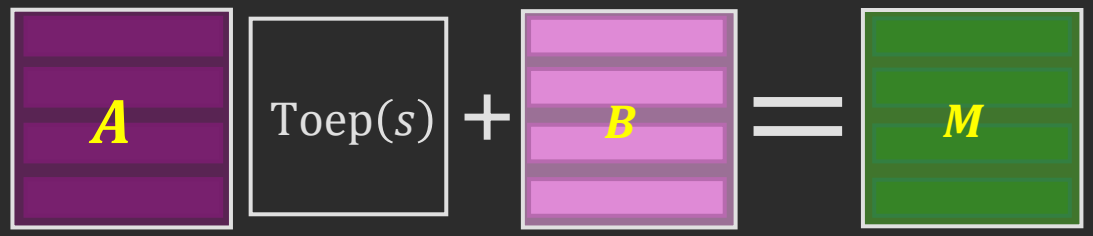
CCMM?



CCMM?

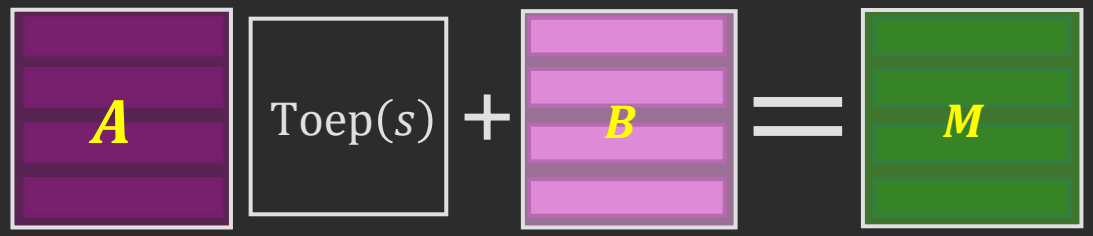

U V = [A Toep(s) + B][A′ Toep(s) + B′]
= A Toep(s) A′ Toep(s) + A Toep(s) B′ + B A′ Toep(s) + B B′
? A A″ Toep(s) Toep(s) + A B″ Toep(s) + B A′ Toep(s) + B B′

Ciphertext Matrix Transpose (CMT)

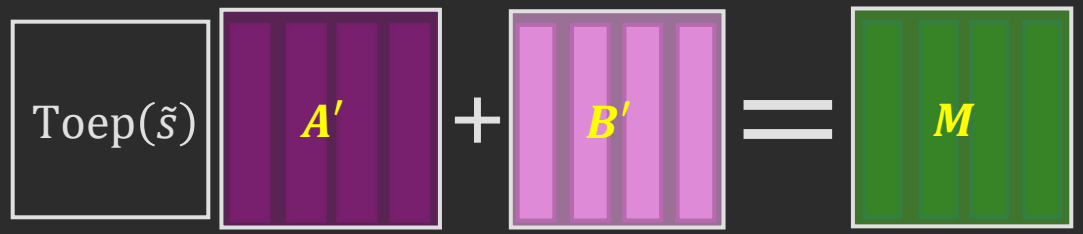


Encrypting each row: $a_i s + b_i = m_i$

Ciphertext Matrix Transpose (CMT)

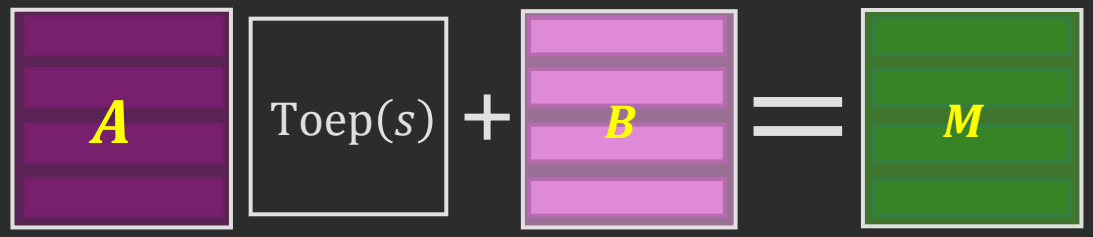


Encrypting each row: $a_i s + b_i = m_i$

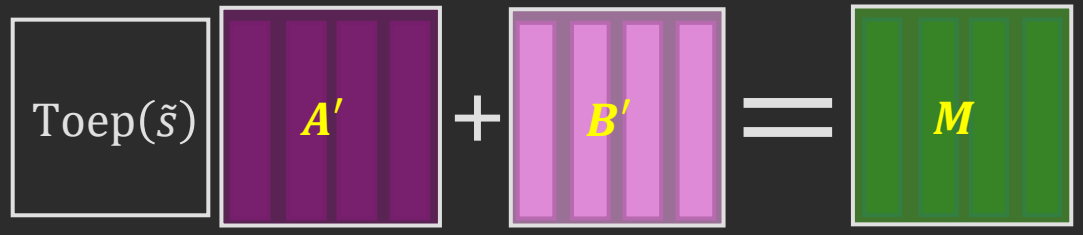


Encrypting each column: $a_j s + b_j = m'_j$

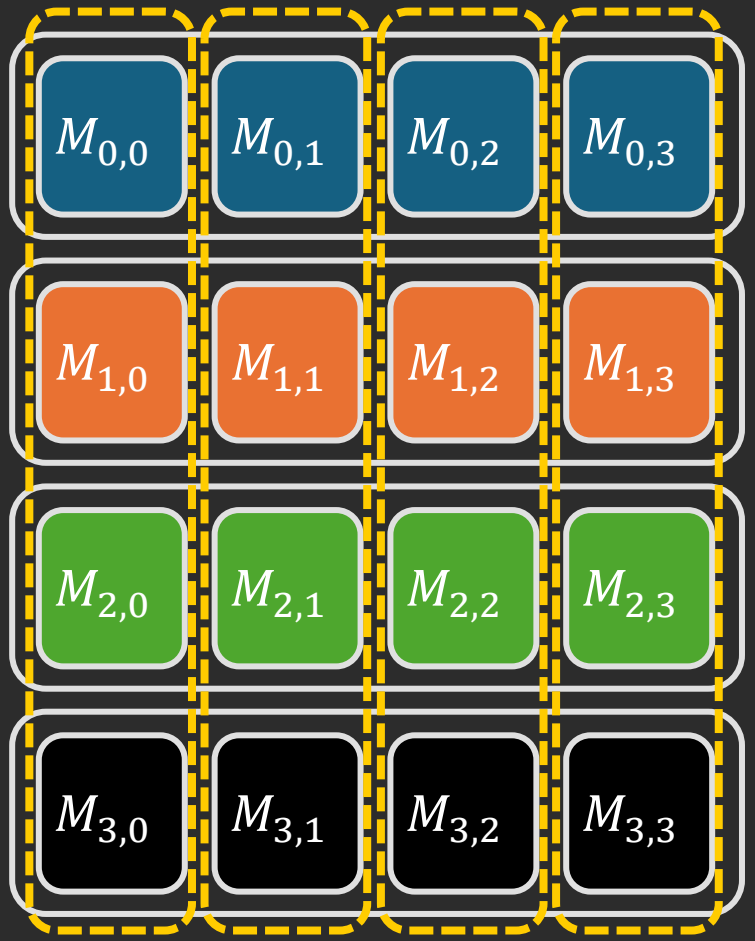
Ciphertext Matrix Transpose (CMT)



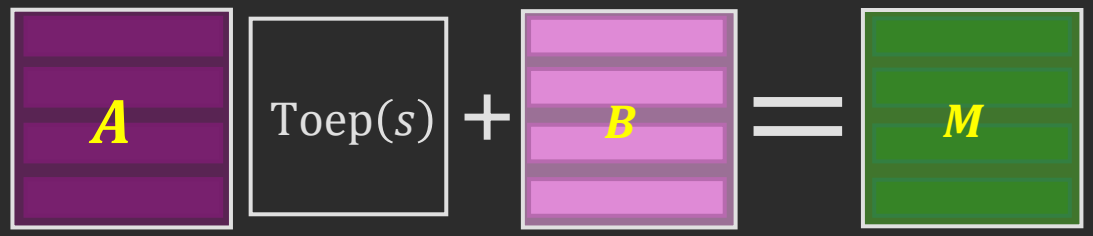
Encrypting each row: $a_i s + b_i = m_i$



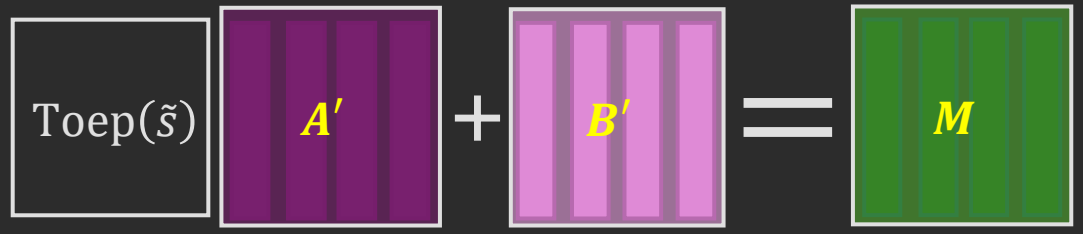
Encrypting each column: $a_j s + b_j = m'_j$



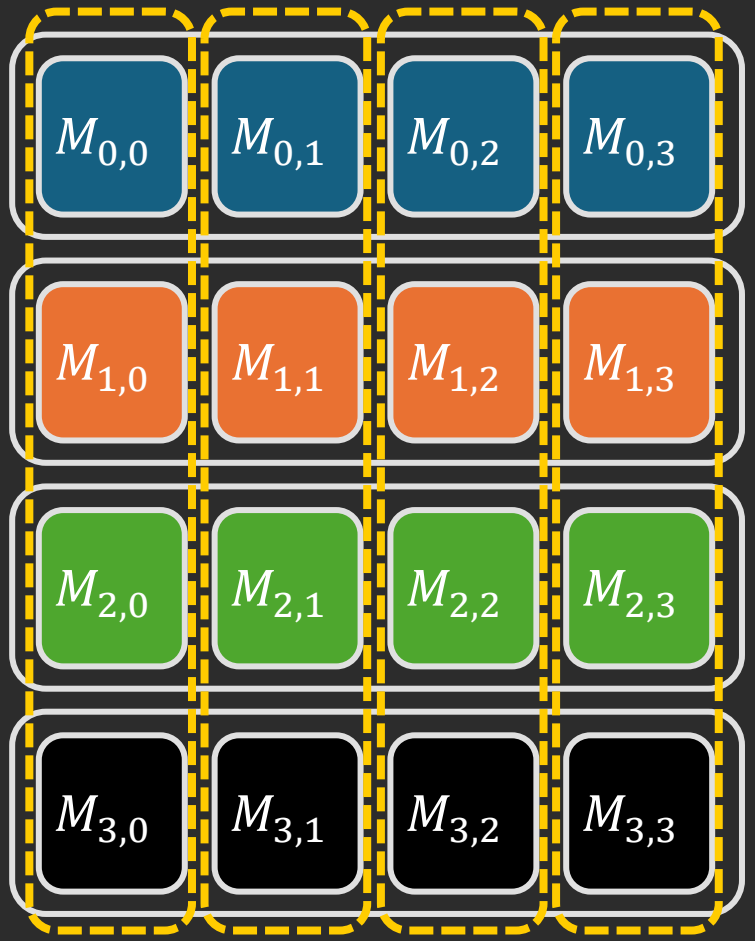
Ciphertext Matrix Transpose (CMT)



Encrypting each row: $a_i s + b_i = m_i$



Encrypting each column: $a_j s + b_j = m'_j$



N ciphertexts

$$m_i(X) = \sum_{j \in [N]} M_{i,j} X^j$$

CMT

$$m'_j(X) = \sum_{i \in [N]} M_{i,j} X^i$$

N ciphertexts

CMT with N keyswitchings

Trace

$$\forall i,j, \quad M_{i,j} = N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(m_i(X) \cdot X^{-j}\right)$$

N ciphertexts

$$m_i(X) = \sum_{j \in [N]} M_{i,j}\, X^j$$

CMT

$$m'_j(X) = \sum_{i \in [N]} M_{i,j}\, X^i$$

N ciphertexts

CMT with N keyswitchings

Trace

$$\forall i,j, \quad M_{i,j} = N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(m_i(X) \cdot X^{-j}\right)$$

N ciphertexts

$$m_i(X) = \sum_{j \in [N]} M_{i,j} X^j$$

CMT

$$m'_j(X) = \sum_{i \in [N]} M_{i,j} X^i$$

N ciphertexts

CMT with N keyswitchings

Trace

$$\forall i,j, \quad M_{i,j} = N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(m_i(X) \cdot X^{-j}\right)$$

Sharing automorphisms

$$\forall j, \quad m'_j(X) = N^{-1} \cdot \sum_{i \in]N]} \sum_{\sigma \in \mathrm{Aut}} \sigma\left(X^{-j} \cdot m_i(X)\right) \cdot X^i$$

N ciphertexts

$$m_i(X) = \sum_{j \in [N]} M_{i,j}\, X^j$$

CMT

$$m'_j(X) = \sum_{i \in [N]} M_{i,j}\, X^i$$

N ciphertexts

CMT with N keyswitchings

Trace

$$\forall i,j, \quad M_{i,j} = N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(m_i(X) \cdot X^{-j}\right)$$

Sharing automorphisms

$$\forall j, \quad m'_j(X) = N^{-1} \cdot \sum_{i \in]N]} \sum_{\sigma \in \mathrm{Aut}} \sigma\left(X^{-j} \cdot m_i(X)\right) \cdot X^i$$

$$= N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sum_{i \in [N]} \sigma\left(X^{-j}\right) \cdot \sigma\left(m_i(X)\right) \cdot X^i$$

$$= N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(X^{-j}\right) \cdot \sigma\left(\sum_{i \in [N]} m_i(X) \cdot \sigma^{-1}\left(X^i\right)\right)$$

N ciphertexts

$$m_i(X) = \sum_{j \in [N]} M_{i,j}\, X^j$$

CMT

$$m'_j(X) = \sum_{i \in [N]} M_{i,j}\, X^i$$

N ciphertexts

CMT with N keyswitchings

Trace

$$\forall i,j, \quad M_{i,j} = N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(m_i(X) \cdot X^{-j}\right)$$

Sharing automorphisms

$$\forall j, \quad m'_j(X) = N^{-1} \cdot \sum_{i \in]N]} \sum_{\sigma \in \mathrm{Aut}} \sigma\left(X^{-j} \cdot m_i(X)\right) \cdot X^i$$

$$= N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sum_{i \in [N]} \sigma\left(X^{-j}\right) \cdot \sigma\left(m_i(X)\right) \cdot X^i$$

$$= N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(X^{-j}\right) \cdot \sigma\left(\sum_{i \in [N]} m_i(X) \cdot \sigma^{-1}\left(X^i\right)\right)$$

$\widetilde{m_\sigma}(X)$

N ciphertexts

$$m_i(X) = \sum_{j \in [N]} M_{i,j}\, X^j$$

CMT

$$m'_j(X) = \sum_{i \in [N]} M_{i,j}\, X^i$$

N ciphertexts

CMT with N keyswitchings

Trace

$$\forall i,j, \quad M_{i,j} = N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(m_i(X) \cdot X^{-j}\right)$$

Sharing automorphisms

$$\forall j, \quad m'_j(X) = N^{-1} \cdot \sum_{i \in]N]} \sum_{\sigma \in \mathrm{Aut}} \sigma\left(X^{-j} \cdot m_i(X)\right) \cdot X^i$$

$$= N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sum_{i \in [N]} \sigma\left(X^{-j}\right) \cdot \sigma\left(m_i(X)\right) \cdot X^i$$

$$= N^{-1} \cdot \sum_{\sigma \in \mathrm{Aut}} \sigma\left(X^{-j}\right) \cdot \sigma\left(\sum_{i \in [N]} m_i(X) \cdot \sigma^{-1}\left(X^i\right)\right)$$

$\widetilde{m_\sigma}(X)$

N ciphertexts

$$m_i(X) = \sum_{j \in [N]} M_{i,j}\, X^j$$

CMT

$$m'_j(X) = \sum_{i \in [N]} M_{i,j}\, X^i$$

N ciphertexts

$\{m_i\}_i$ → (N^2 add) → $\{\widetilde{m_\sigma}\}_\sigma$ → (N key-switchings) → $\{\sigma(\widetilde{m_\sigma})\}_\sigma$ → (N^2 add) → $\{m'_j\}_j$

Tweak Algorithm

- ✓ $\widetilde{m_\sigma}(X) = \sum_i \sigma^{-1}(X^i) \cdot m_i$
- ✓ $m'_j(X) = \sum_\sigma \sigma(X^{-j}) \cdot \sigma(\widetilde{m_\sigma})$

Tweak Algorithm

- ✓ $\widetilde{m_\sigma}(X) = \sum_i \sigma^{-1}(X^i) \cdot m_i$
- ✓ $m'_j(X) = \sum_\sigma \sigma(X^{-j}) \cdot \sigma(\widetilde{m_\sigma})$

$$\text{Tweak}(\{m_i\}_{i\in[n]}) \mapsto \left\{ \sum_{i\in[n]} X^{\frac{2N}{n}ij} \cdot m_i \right\}_{j\in[n]}$$

N is the ring degree of RLWE

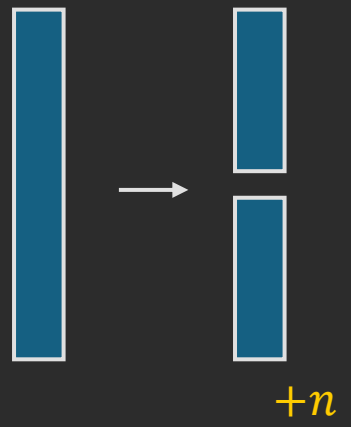
Tweak Algorithm

- ✓ $\widetilde{m_\sigma}(X) = \sum_i \sigma^{-1}(X^i) \cdot m_i$
- ✓ $m'_j(X) = \sum_\sigma \sigma(X^{-j}) \cdot \sigma(\widetilde{m_\sigma})$

$$\text{Tweak}(\{m_i\}_{i\in[n]}) \mapsto \left\{\sum_{i\in[n]} X^{\frac{2N}{n}ij} \cdot m_i\right\}_{j\in[n]}$$

N is the ring degree of RLWE

- $\text{Tweak}(\{m_i\}_{i\in[n]})$ can be done with
  - $\text{Tweak}(\{m_{2i}\}_{i\in[n/2]})$
  - $\text{Tweak}(\{m_{2i+1}\}_{i\in[n/2]})$
  - n ring additions



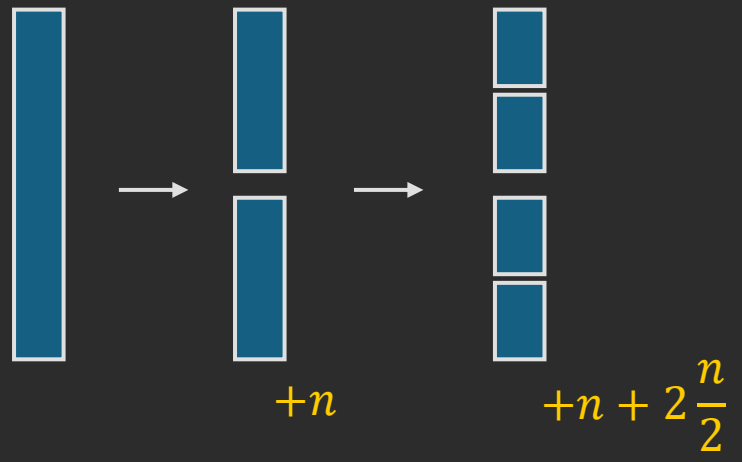
Tweak Algorithm

- ✓ $\widetilde{m_\sigma}(X) = \sum_i \sigma^{-1}(X^i) \cdot m_i$
- ✓ $m'_j(X) = \sum_\sigma \sigma(X^{-j}) \cdot \sigma(\widetilde{m_\sigma})$

$$\text{Tweak}(\{m_i\}_{i\in[n]}) \mapsto \left\{\sum_{i\in[n]} X^{\frac{2N}{n}ij} \cdot m_i\right\}_{j\in[n]}$$

N is the ring degree of RLWE

- $\text{Tweak}(\{m_i\}_{i\in[n]})$ can be done with
  - $\text{Tweak}(\{m_{2i}\}_{i\in[n/2]})$
  - $\text{Tweak}(\{m_{2i+1}\}_{i\in[n/2]})$
  - n ring additions



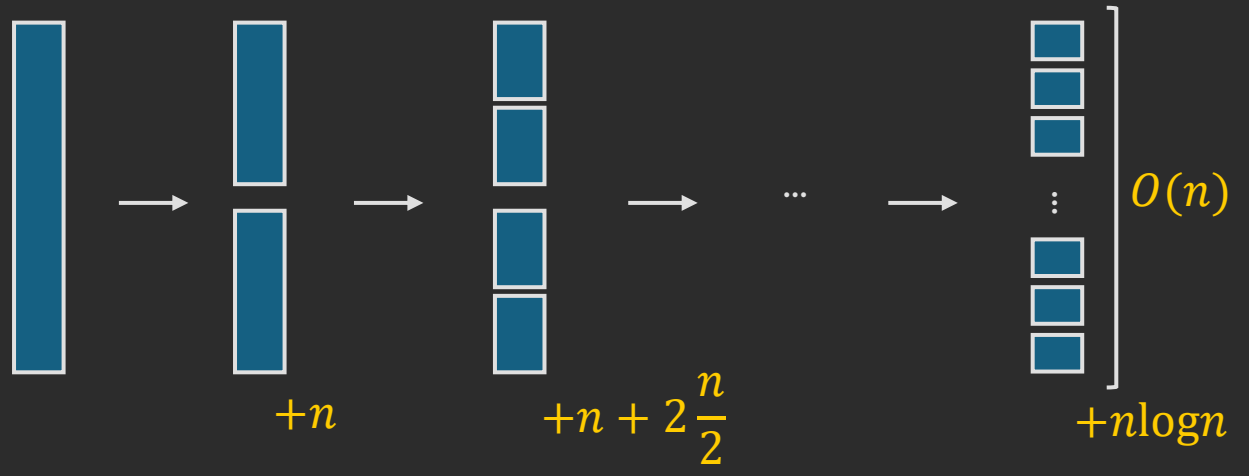
Tweak Algorithm

- ✓ $\widetilde{m_\sigma}(X) = \sum_i \sigma^{-1}(X^i) \cdot m_i$
- ✓ $m'_j(X) = \sum_\sigma \sigma(X^{-j}) \cdot \sigma(\widetilde{m_\sigma})$

$$\text{Tweak}(\{m_i\}_{i\in[n]}) \mapsto \left\{ \sum_{i\in[n]} X^{\frac{2N}{n}ij} \cdot m_i \right\}_{j\in[n]}$$

N is the ring degree of RLWE

- $\text{Tweak}(\{m_i\}_{i\in[n]})$ can be done with
  - $\text{Tweak}(\{m_{2i}\}_{i\in[n/2]})$
  - $\text{Tweak}(\{m_{2i+1}\}_{i\in[n/2]})$
  - n ring additions



Tweak Algorithm

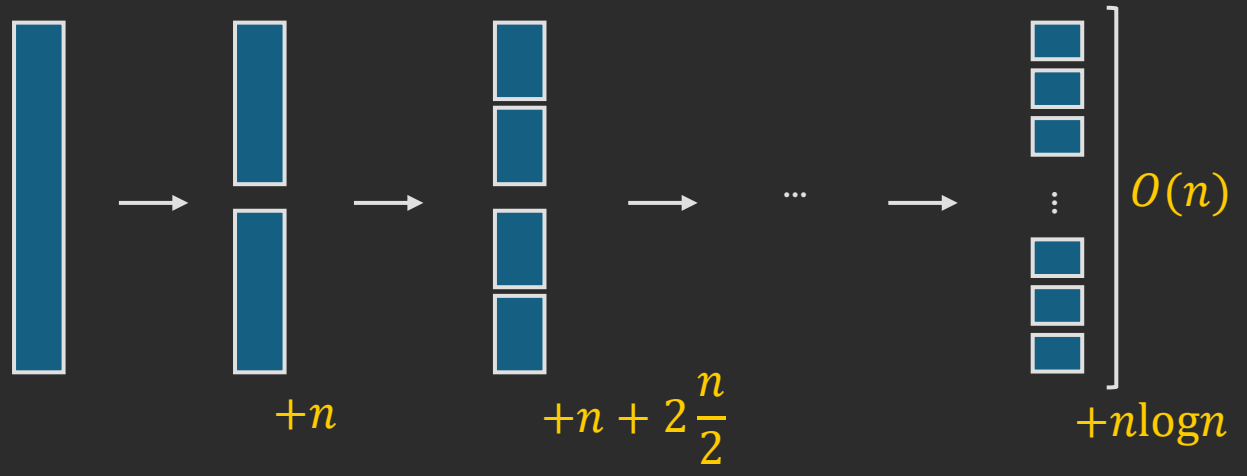
- ✓ $\widetilde{m_\sigma}(X) = \sum_i \sigma^{-1}(X^i) \cdot m_i$
- ✓ $m'_j(X) = \sum_\sigma \sigma(X^{-j}) \cdot \sigma(\widetilde{m_\sigma})$

$$\text{Tweak}(\{m_i\}_{i\in[n]}) \mapsto \left\{\sum_{i\in[n]} X^{\frac{2N}{n}ij} \cdot m_i\right\}_{j\in[n]}$$

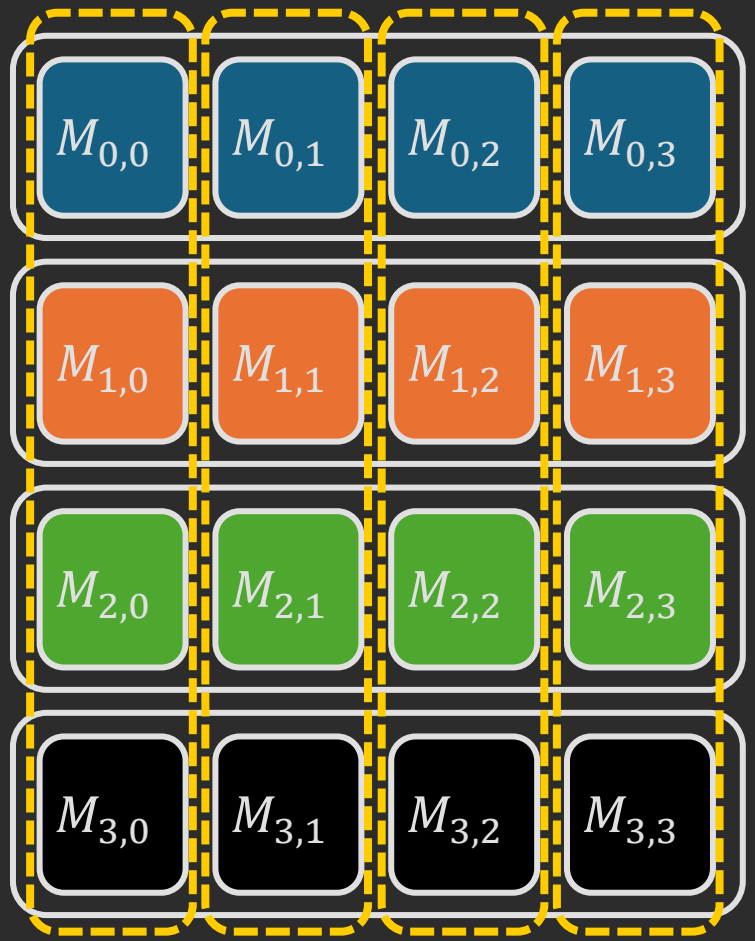
N is the ring degree of RLWE

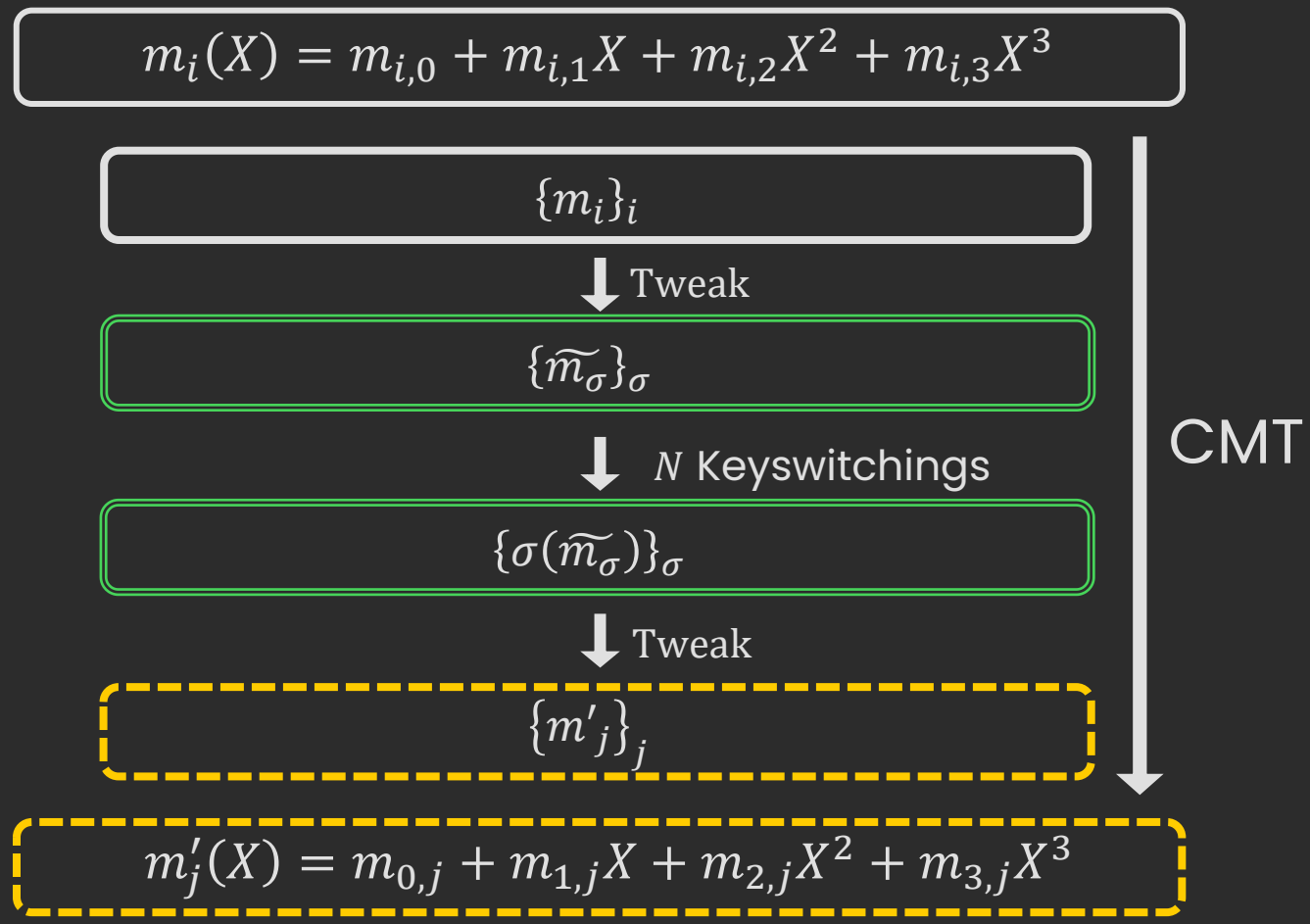
- $\text{Tweak}(\{m_i\}_{i\in[n]})$ can be done with
  - $\text{Tweak}(\{m_{2i}\}_{i\in[n/2]})$
  - $\text{Tweak}(\{m_{2i+1}\}_{i\in[n/2]})$
  - n ring additions

❖ The cost of $\text{Tweak}(\{m_i\}_{i\in[n]})$ is $Nn\log n$



CMT Algorithm

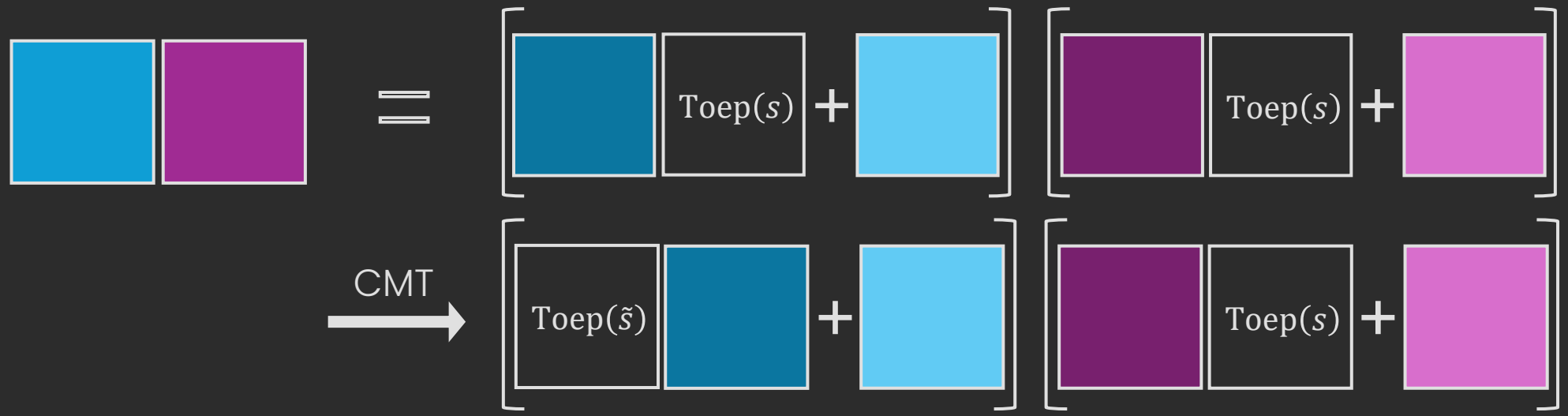




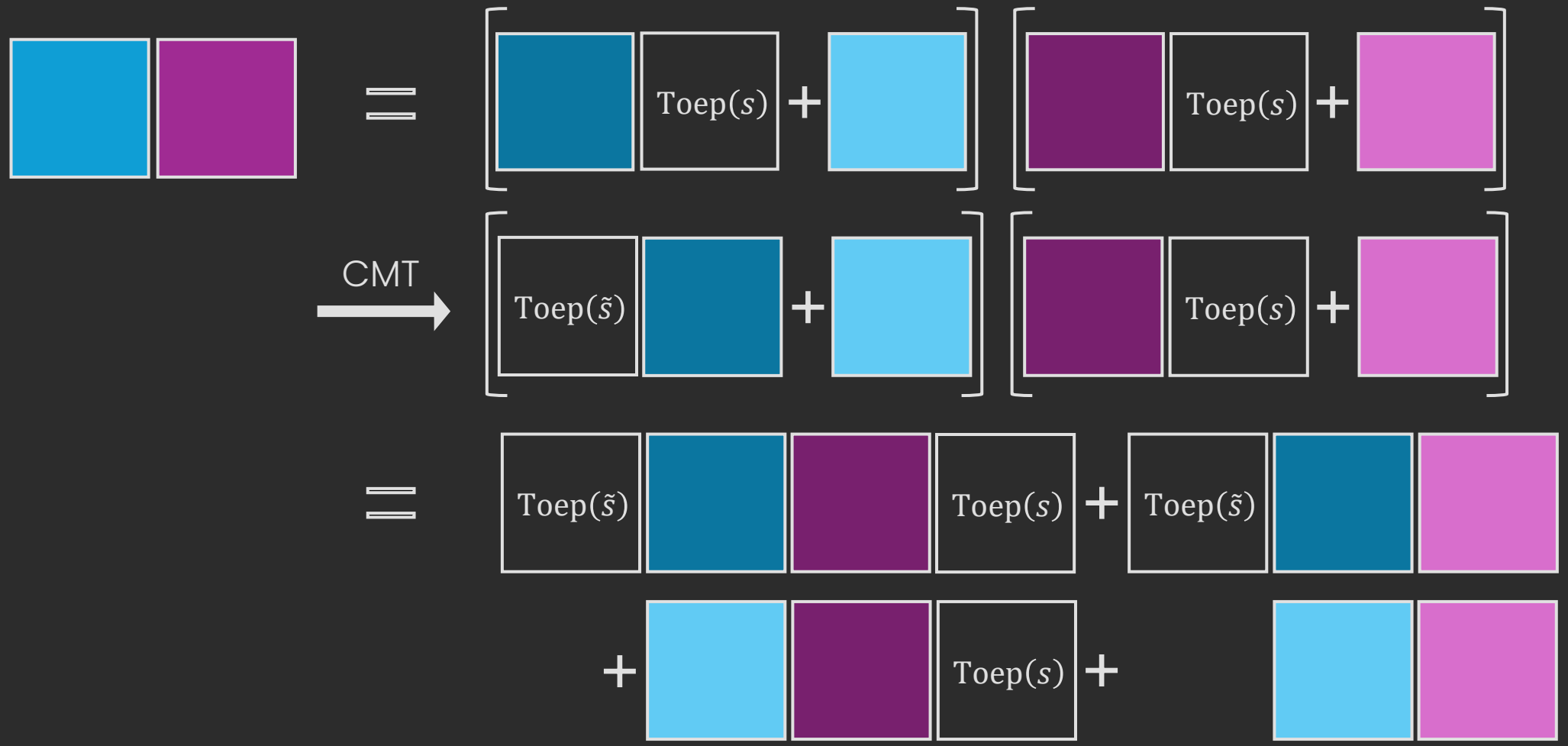
❖$\tilde{O}(N^2)$ operations

Reduction from CCMM to PPMM (1/2)

Reduction from CCMM to PPMM (1/2)

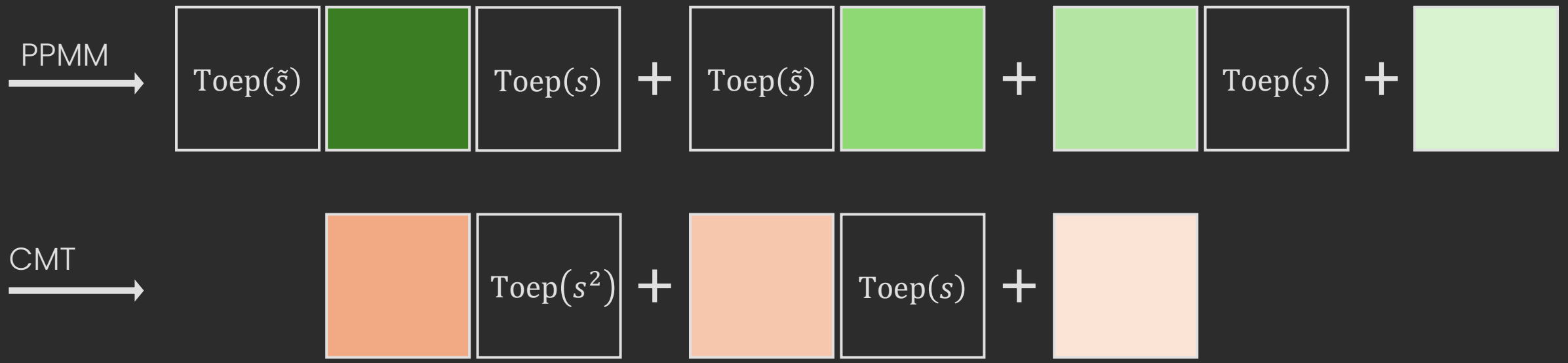


Reduction from CCMM to PPMM (1/2)

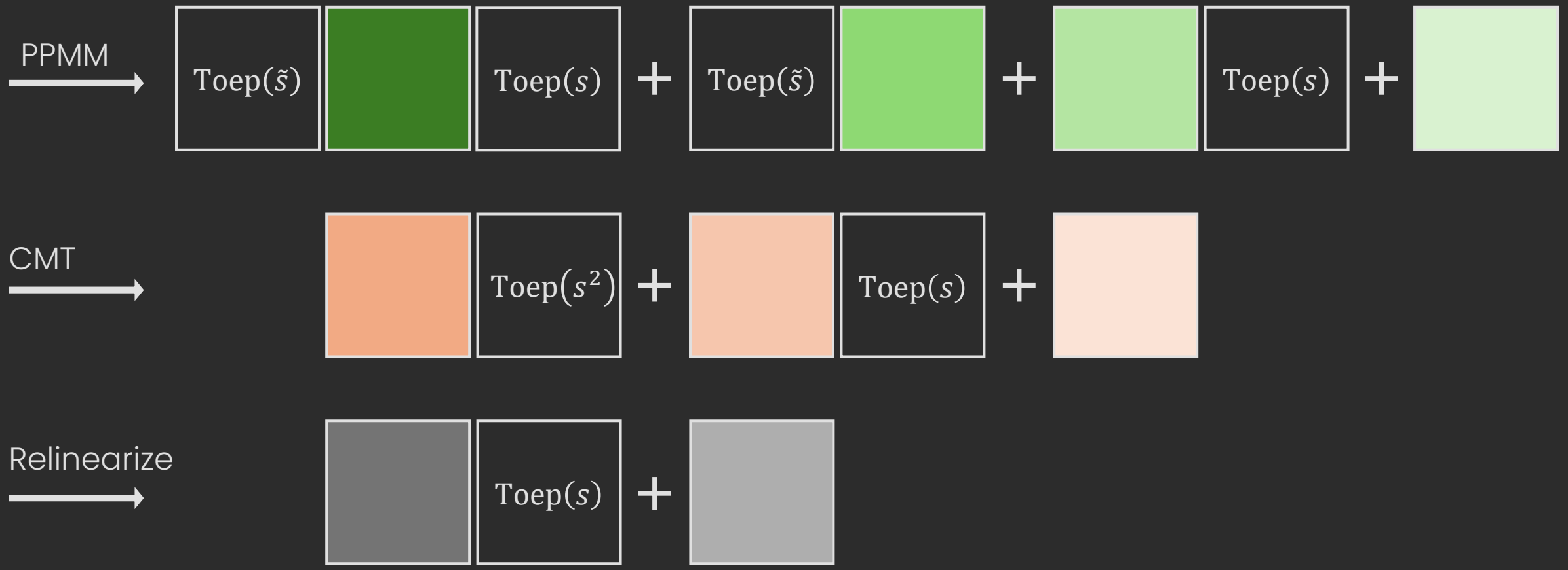


Reduction from CCMM to PPMM (2/2)

Reduction from CCMM to PPMM (2/2)



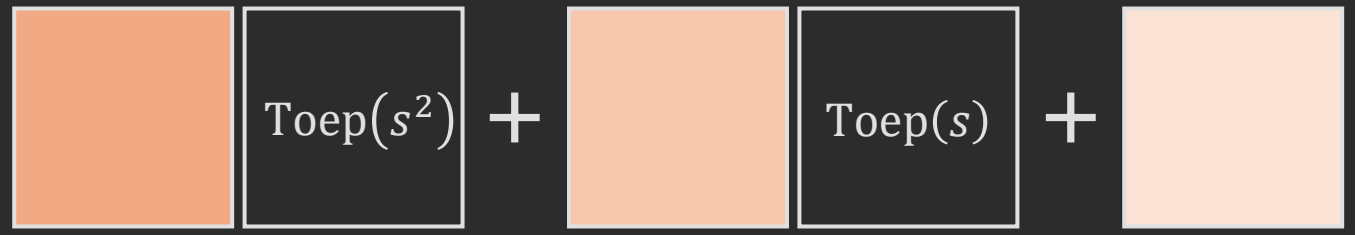
Reduction from CCMM to PPMM (2/2)



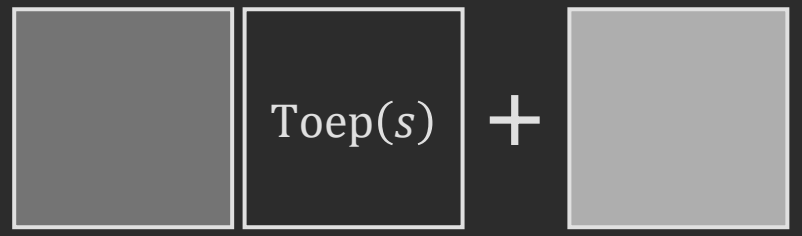
Reduction from CCMM to PPMM (2/2)

PPMM →

CMT →



Relinearize →



- $N \times N \times N$ CCMM by 4 PPMMs and 3 CMTs
- For PPMM, we use BLAS libraries.
- The others than PPMM are subcubic.

Experimental Results on CCMM

Algorithm	Matrix Dimension	$(\log N, \log Q)$	CMTs	PPMMs	Relin. & Resc.	Total (s)	Prec. (bit)	Key size (MB)
Basic	4096	$(12, 36+28)$	25.5	57.1	2.58	85.2	18.7	436
Basic	8192	$(13, 38+28)$	104	481	11.8	596	18.5	1960
Lightweight	8192	$(13, 38+28)$	186	474	11.8	672	18.5	1.57

All experiments are measured on Intel® Xeon® Gold 6242 CPU at 2.80GHz with a single-thread

All parameters are 128-bit secure

HEaaN library for HE, FLINT library (based on OpenBLAS) for PPMM

Experimental Results on CMT

Algorithm	Matrix Dimension	$(\log N, \log Q)$	Latency (s)	Prec. (bit)	Key size (MB)
Basic	2048	$(11, 26)$	0.764	10.7	27.3
Basic	4096	$(12, 28)$	3.04	16.3	134
Lightweight	4096	$(12, 28)$	4.92	14.2	0.246

All experiments are measured on Intel® Xeon® Gold 6242 CPU at 2.80GHz with a single-thread

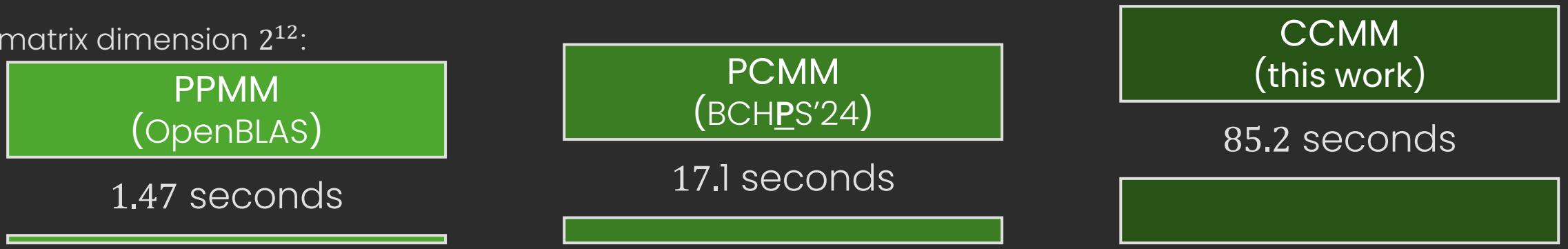
All parameters are 128-bit secure

HEaaN library for HE

Wrapping up!

- Fast CCMM
  - Leverage efficiency of BLAS libraries
- Fast CMT
  - Useful beyond being as a tool for CCMM
- Lightweight algorithms
  - CCMM with keys less than 2 MB

For matrix dimension 2^{12}:



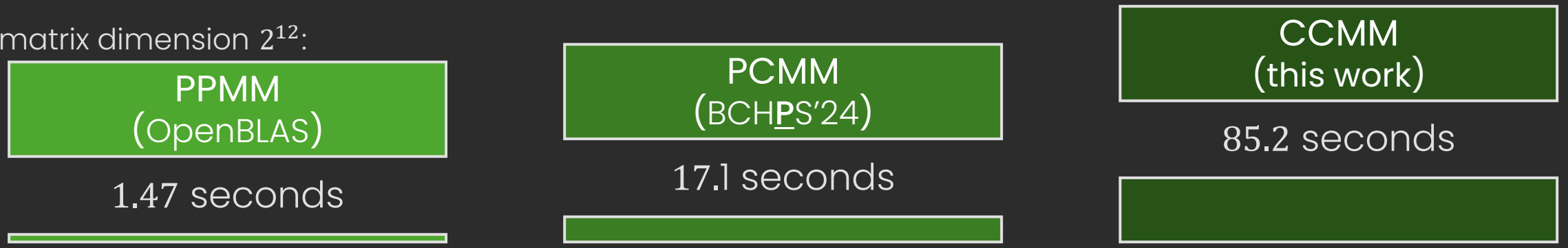
Wrapping up!

- Fast CCMM
  - Leverage efficiency of BLAS libraries
- Fast CMT
  - Useful beyond being as a tool for CCMM
- Lightweight algorithms
  - CCMM with keys less than 2 MB

eprint: 2025/448

Thank you!

For matrix dimension 2^{12}:



References

[BCHPS'24] Y. Bae, J. H. Cheon, G. Hanrot, J. H. Park, D. Stehlé. *"Plaintext-Ciphertext Matrix Multiplication and FHE Bootstrapping: Fast and Fused."* Crypto 2024

[JKLS'18] X. Jiang, M. Kim, K. Lauter, Y. Song. *"Secure Outsourced Matrix Computation and Application to Neural Networks."* CCS 2018

[LZ'22] J. Liu, L. F. Zhang. *"Privacy-preserving and publicly verifiable matrix multiplication."* IEEE Trans. on Services Computing, 2022