FHE ring packing - affordable and convenient

A trip to find out efficient ring packing method : HERMES

Youngjin Bae, Jung Hee Cheon, Jaehyung Kim,
Jai Hyun Park, Damien Stehlé

Slides originally made by Jai Hyun Park. Modified by Jaehyung Kim.

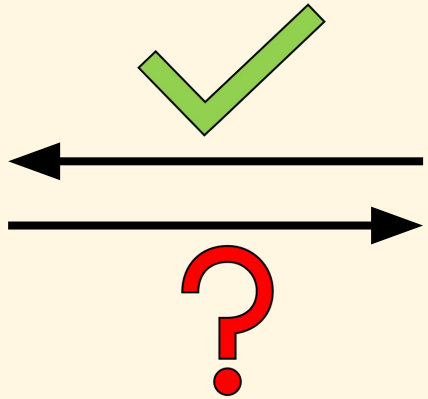
This Work

- We consider FHE ring packing problem
- We suggest generic acceleration tools for FHE RP.
- We propose a new FHE RP method, HERMES.
  - **40x** higher throughput compared to state-of-the-art.
- Application to transciphering.

Ring Packing

LWE format

- TFHE / FHEW
- Granularity and fast latency
- Efficient computations on individual bits



Ring packing

Ring LWE format (RLWE)

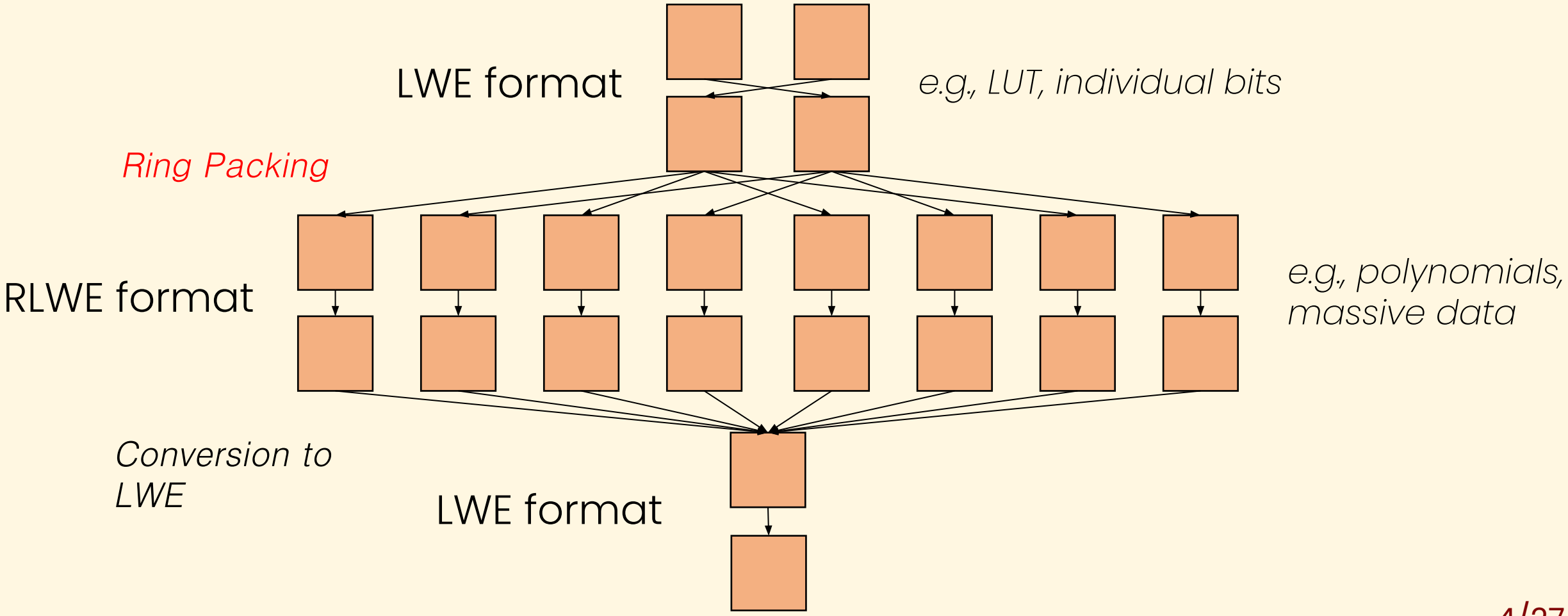
- BGV / BFV / CKKS
- Scalability and high throughput
- Efficient + and x on small integers

[CGGI17, MS18, BGGJ20, CDKS21, LHH+21, ..]

Application 1: Scheme switching

[BGGJ20, LHH+21]

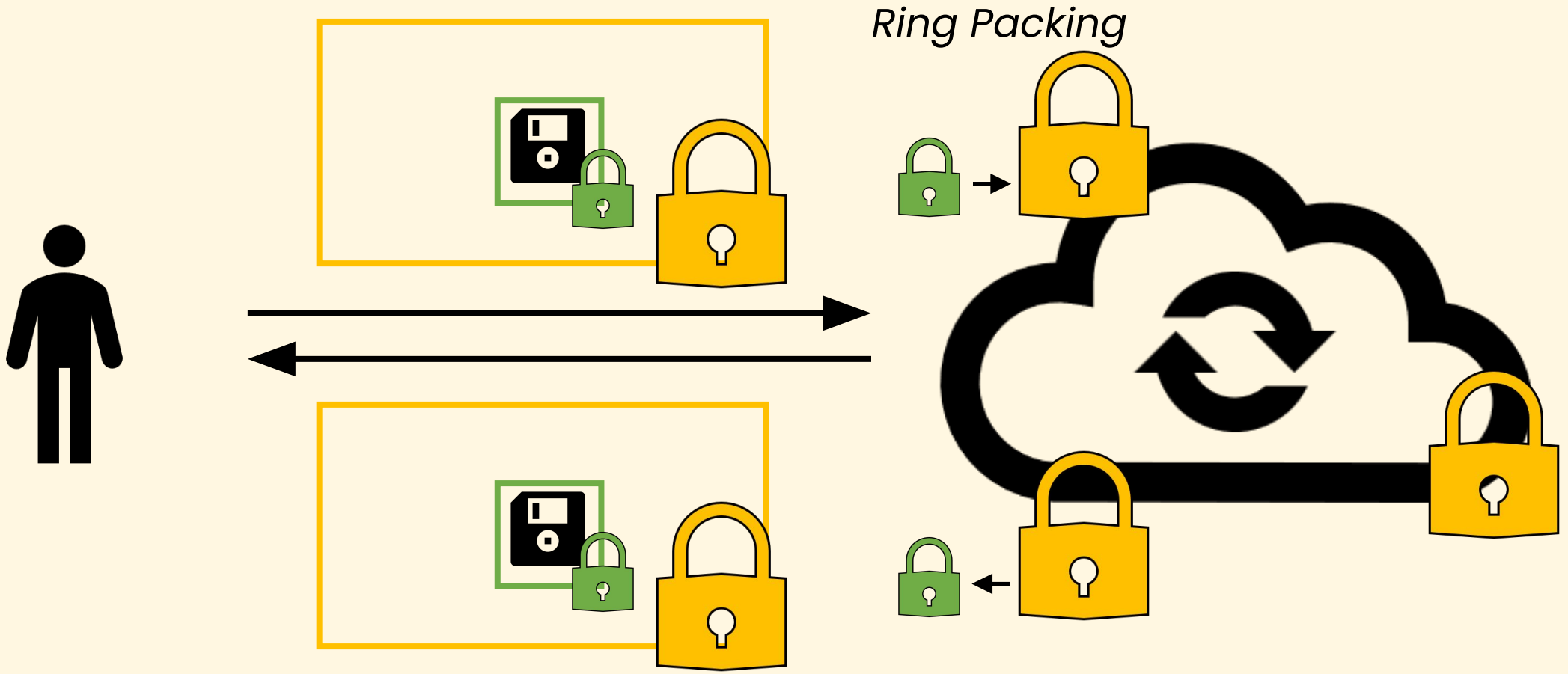
- Heterogeneous operation types and computational widths.



Application 2: Transciphering

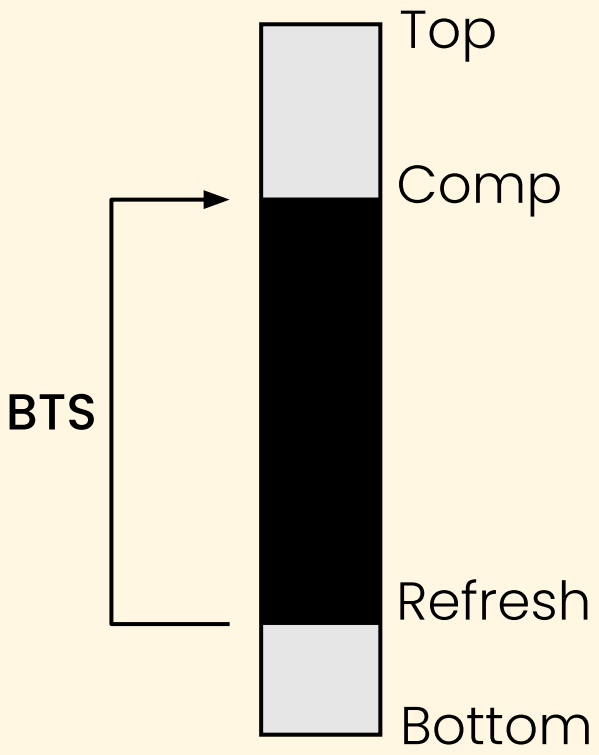
[CGGI17, CDKS21]

- Sending RLWE-based FHE ciphertexts with high granularity



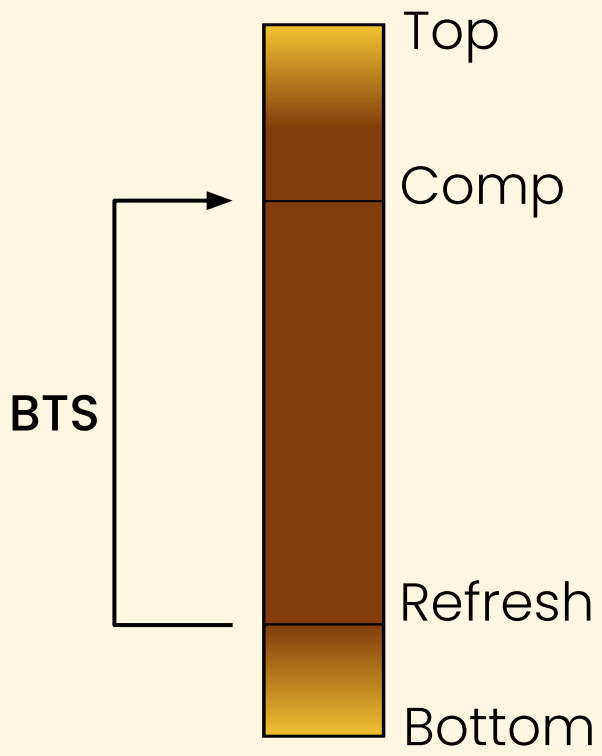
RLWE-based FHE schemes

- Moduli chain
  - RLWE schemes are *leveled homomorphic encryptions*.
  - Level $m \leftrightarrow Q_m = q_0 q_1 \cdots q_m$
- Life of a ciphertext
  - Black region : homomorphic computations
  - Bootstrapping (**BTS**): Refresh → Comp
  - Grey region : reserved for bootstrapping



RLWE-based FHE schemes

- Encoding structure
  - Slots-Encoding / Coefficients-Encoding
  - DFT / NTT Slots for SIMD
  - $\mathbb{C}^{N/2} \leftrightarrow \mathbb{Z}_q[X]/(X^N+1)$
- How state changes
  - Coefficients-to-Slots and Slots-to-Coefficients
  - At the very bottom and the very top

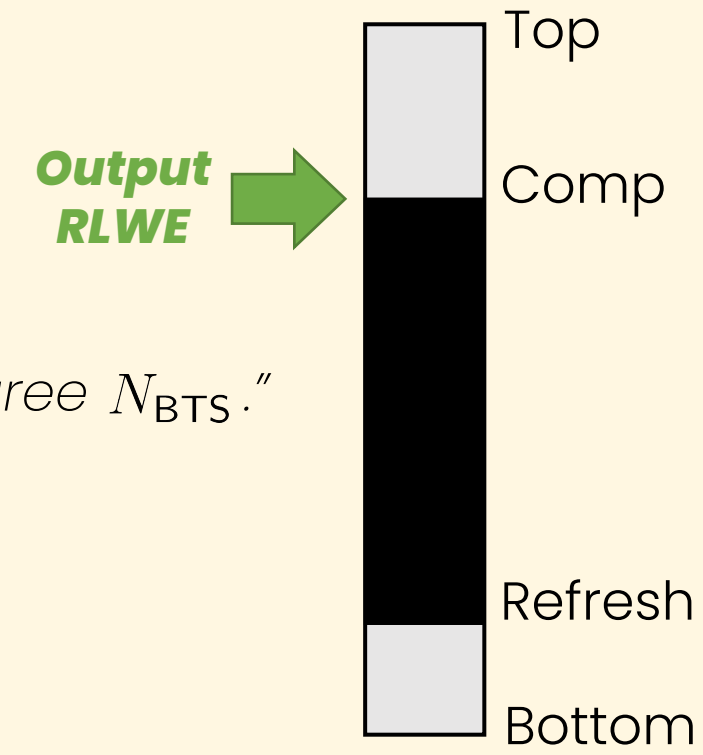


FHE Ring Packing

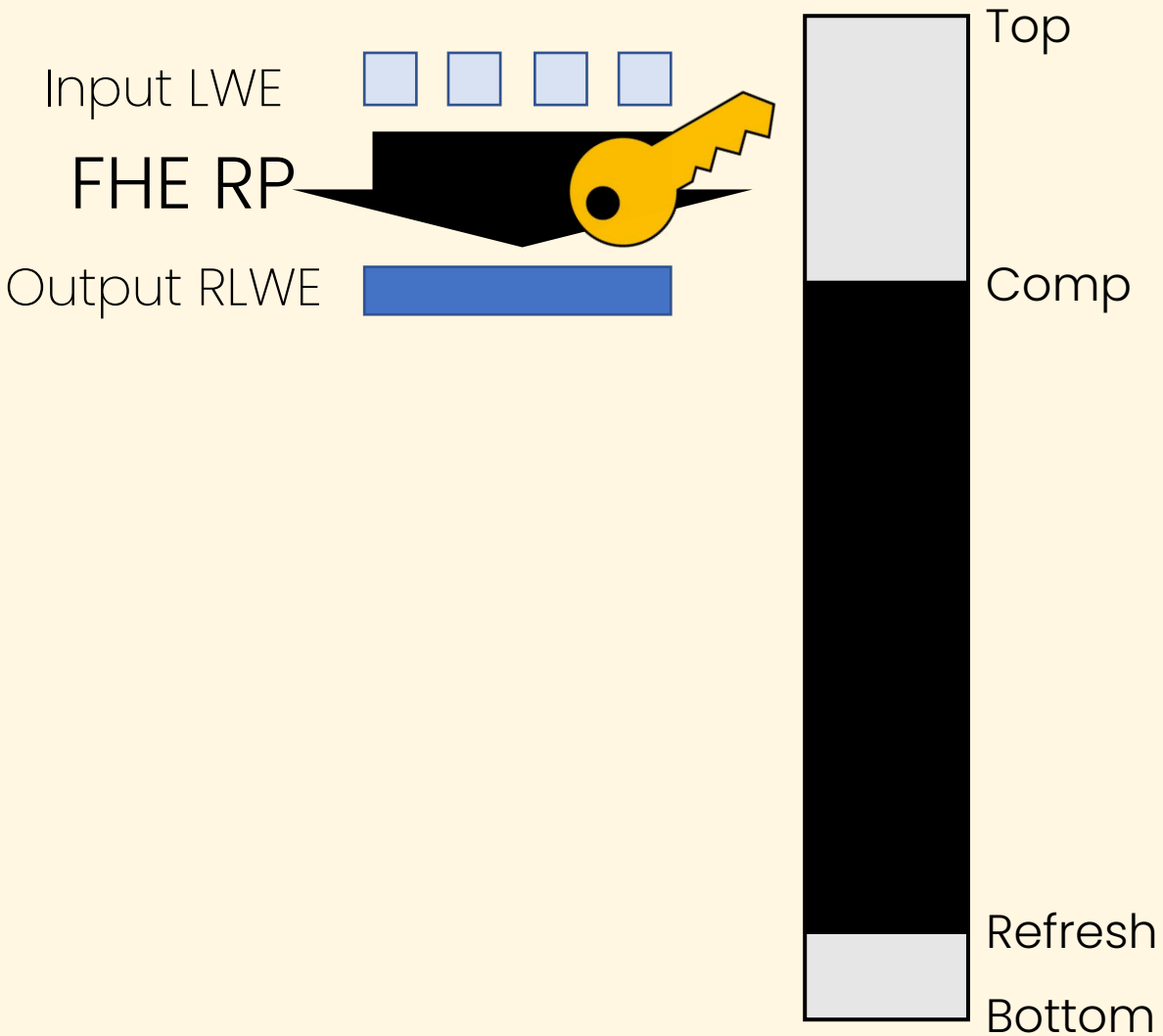
- Which ring packing?
  - Parameters: Modulus and Ring degree
  - Encoding: Slots-encoding / Coefficients-encoding

- FHE Ring Packing (FHE RP)

"Packing into slots-encoded RLWE of modulus Q_{comp} and degree N_{BTS}."



Large parameters for Naive FHE RP

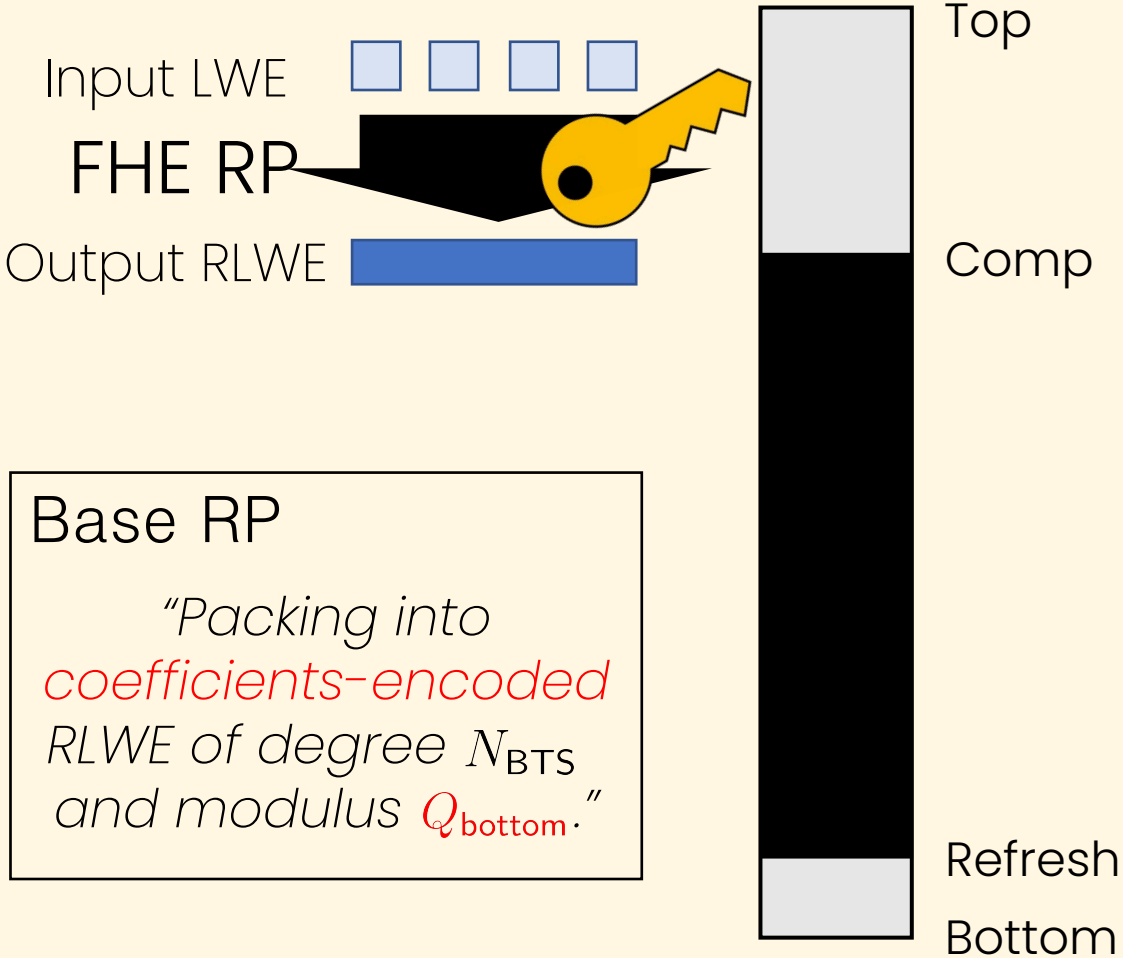


- FHE RP means outputting RLWE with large parameters.
- Unsatisfactory runtime and key size.
  - Computation in higher modulus and degree is slow.
  - Requires large evaluation keys.

Accelerating FHE RP

RLWE Modulus Optimization

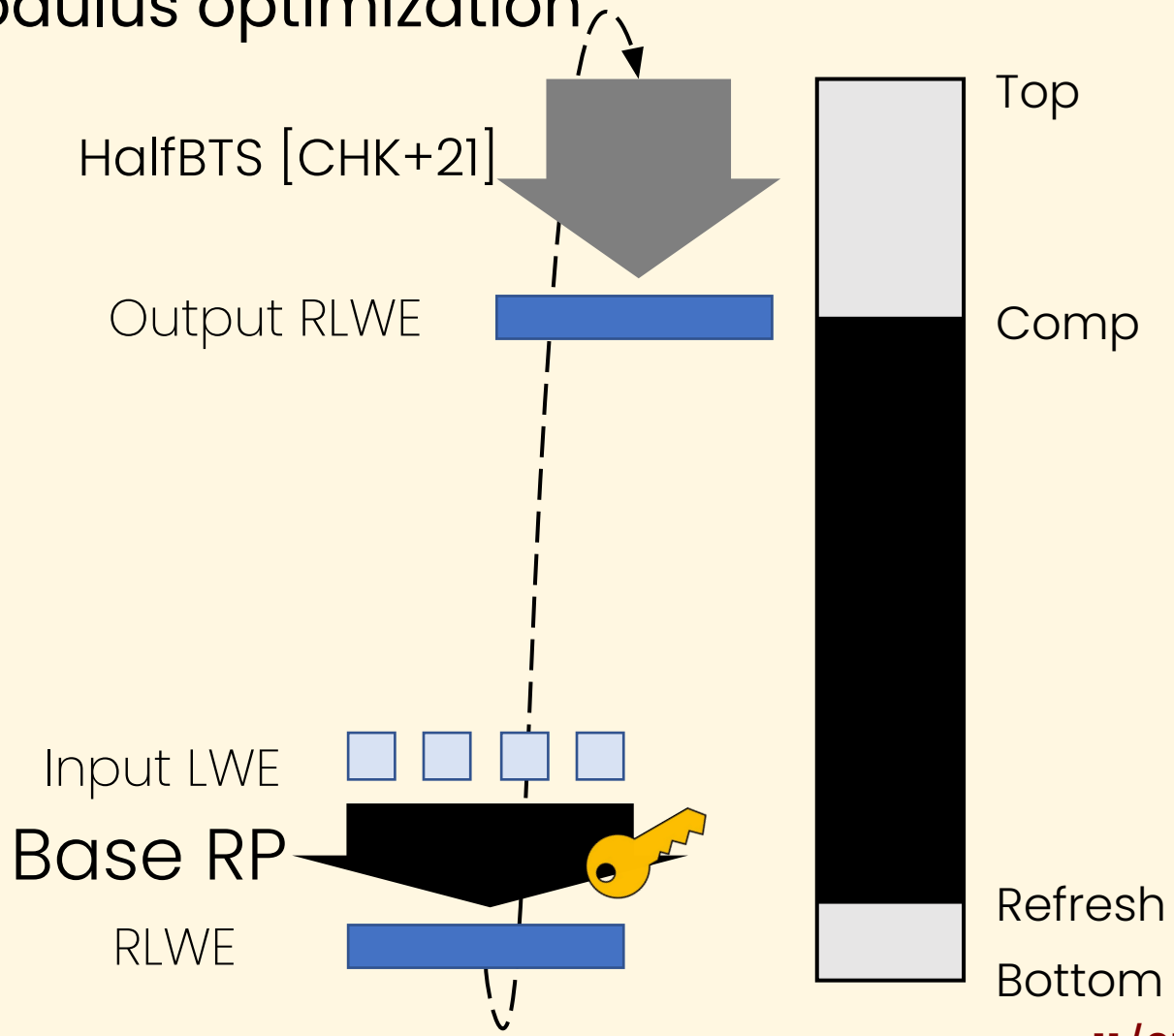
Conventional approach



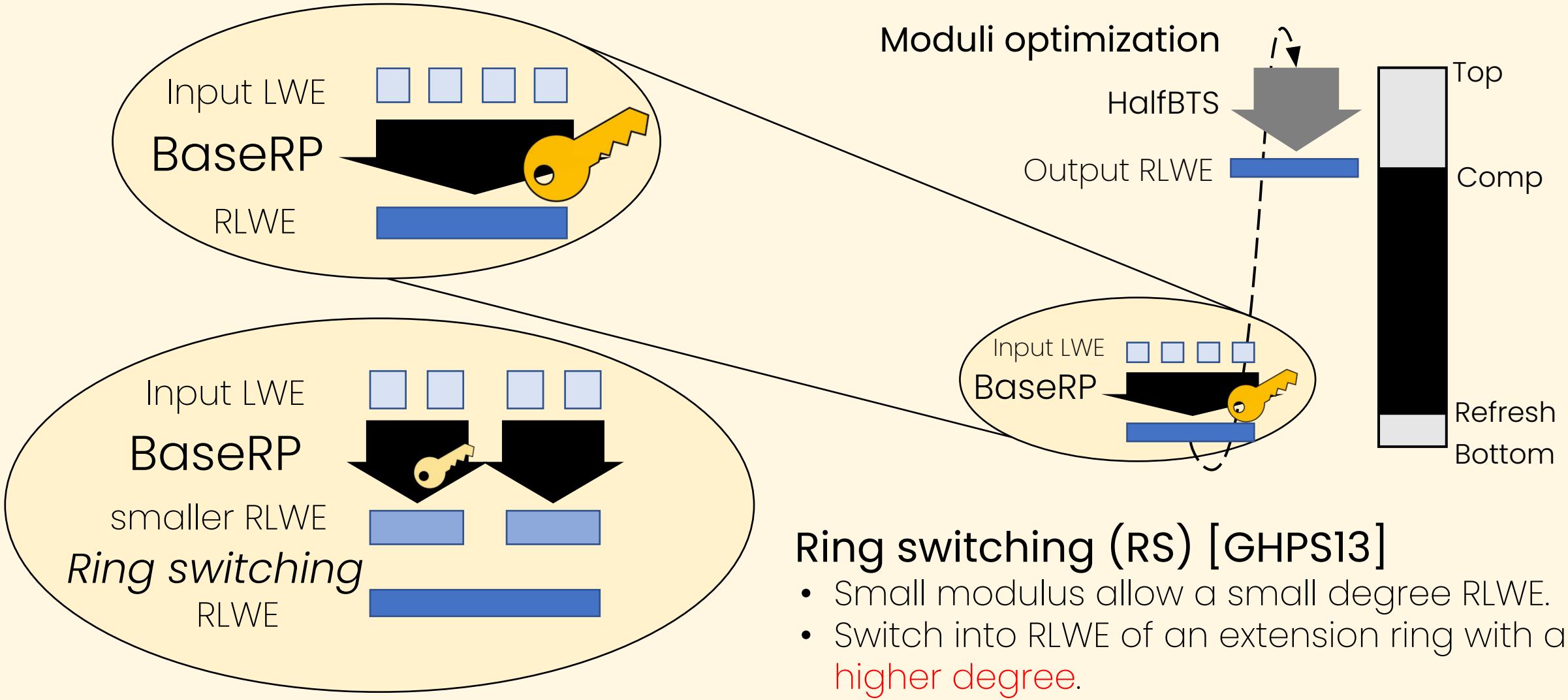
Base RP

"Packing into coefficients-encoded RLWE of degree N_{BTS} and modulus Q_{bottom}."

Modulus optimization



RLWE Degree Optimization

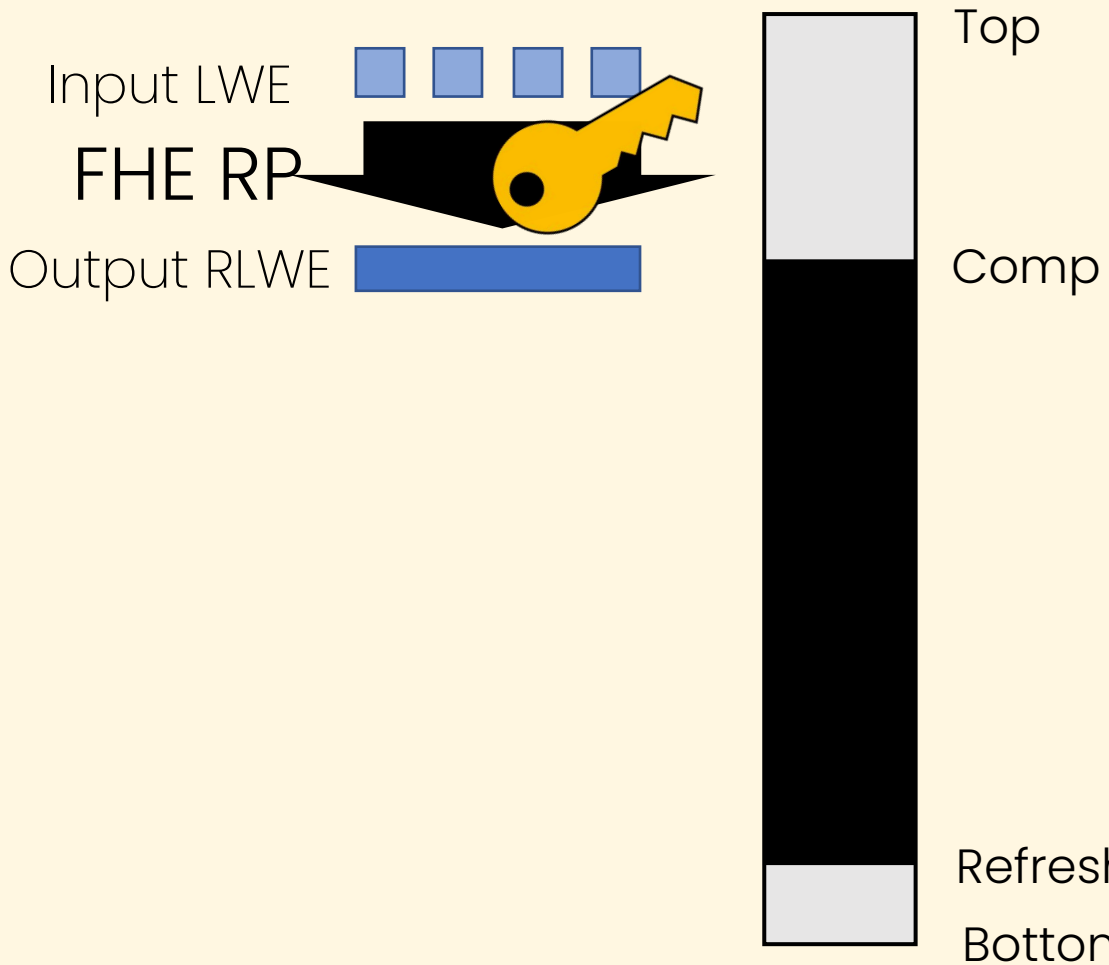


Ring switching (RS) [GHPS13]

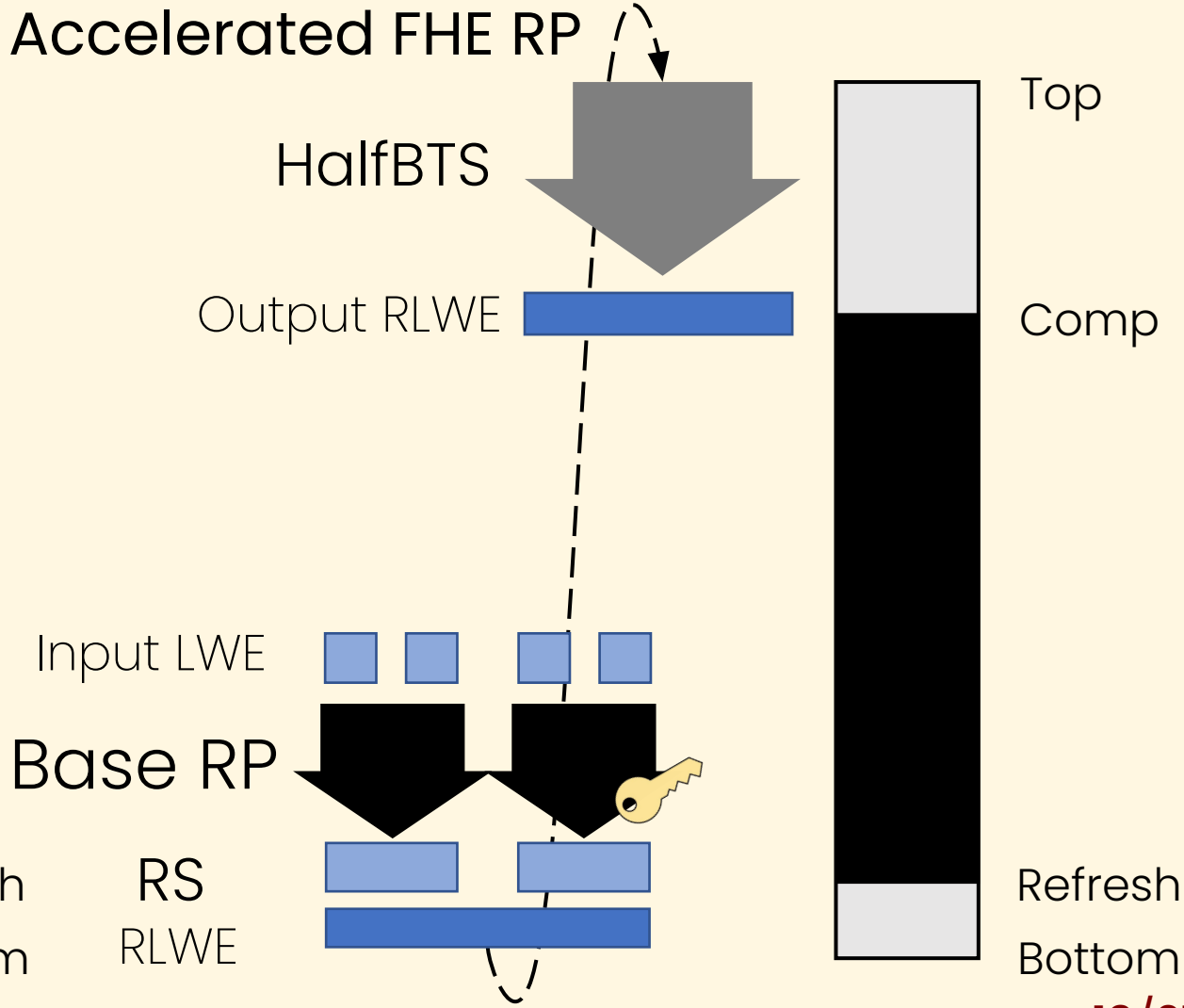
- Small modulus allow a small degree RLWE.
- Switch into RLWE of an extension ring with a higher degree.

Improved FHE RP

Conventional FHE RP



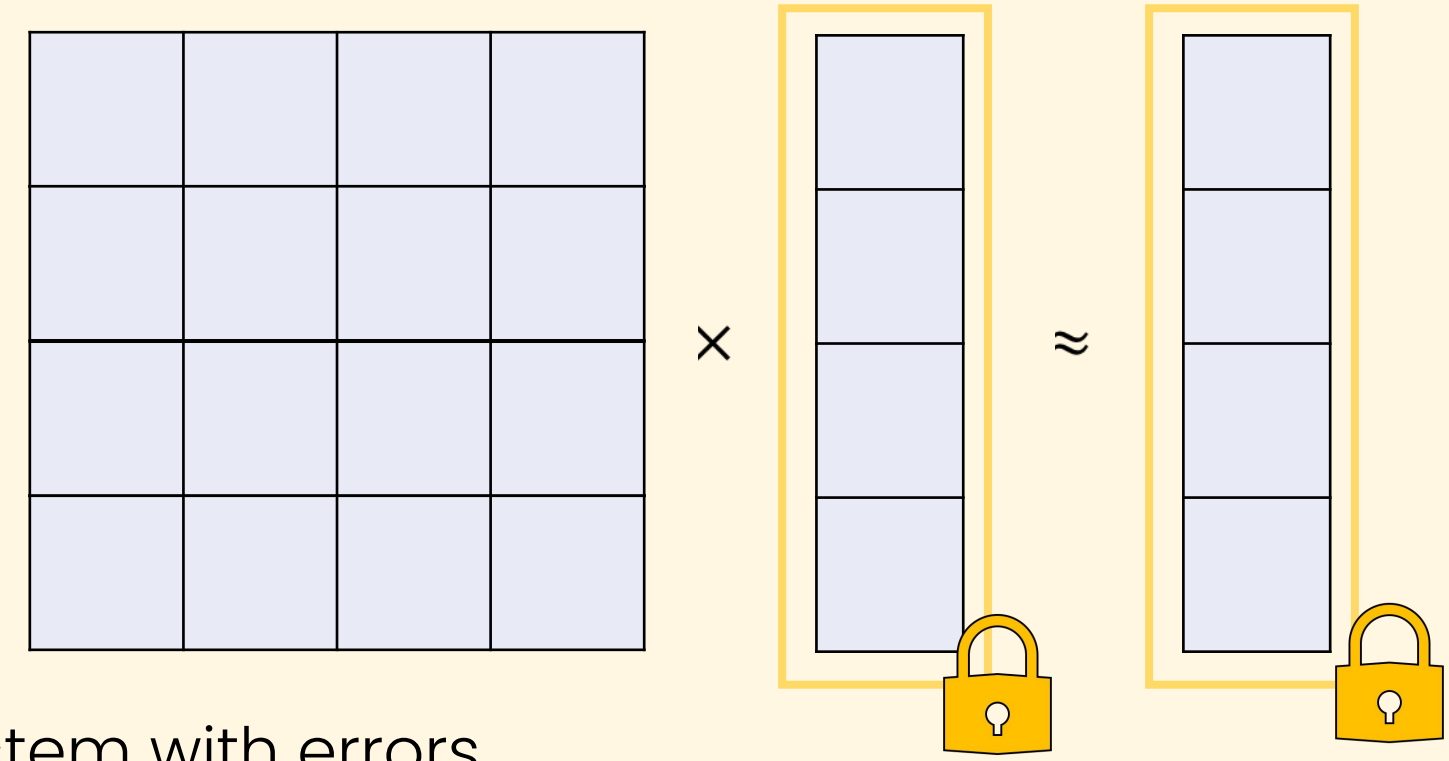
Accelerated FHE RP



Existing Approaches

RP as a Matrix-Vector Multiplication

LWE ciphertexts $\mathbf{ct}_i$: $c_{i\,1}s_1 + c_{i\,2}s_2 + \cdots + s_{i\,K}s_K \approx m_i$ for each i



... is a linear system with errors.

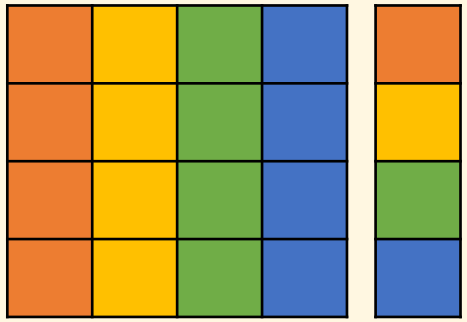
RP is (plaintext) matrix – (ciphertext) vector multiplication in RLWE formats.

Existing Approaches

- Three approaches to encode the plaintext matrix.

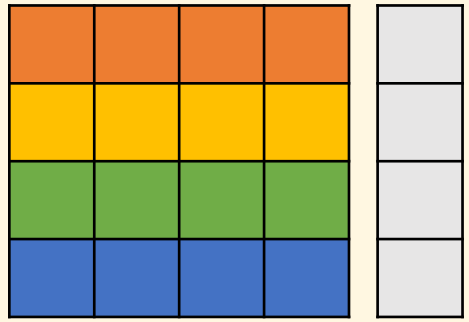
Column method

- CGGI17, BGGJ20
- K key switchings
- K keys
- Consumes $\mathbf{0}$ level



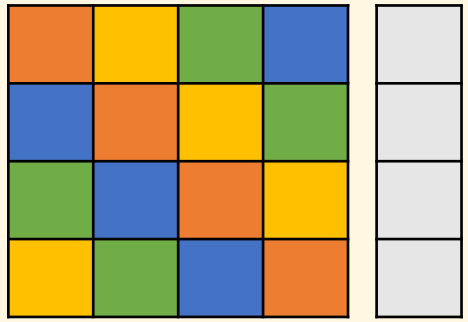
Row method

- CDKS21
- K key switchings
- $\log K$ keys
- Consumes $\mathbf{1}$ level



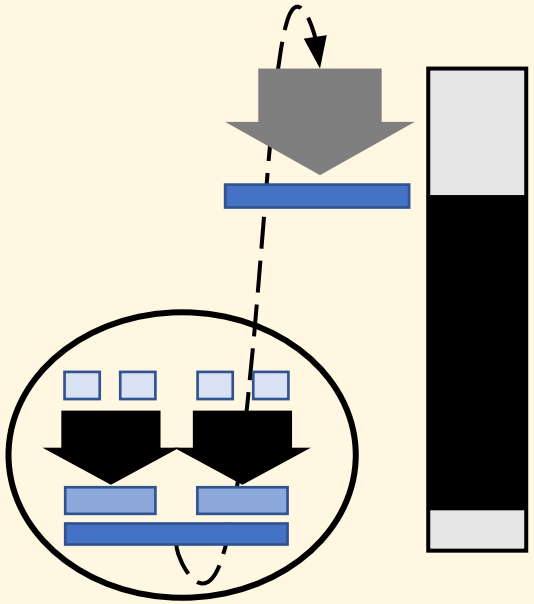
Diagonal method

- HS14, LHH+21
- $2\sqrt{K}$ key switchings
- $2\sqrt{K}$ keys
- Consumes $\geq \mathbf{4}$ levels



K: LWE dimension

Base RP with Existing Approaches



- The less modulus consumption is better.
- Column method is the most effective after optimizations.

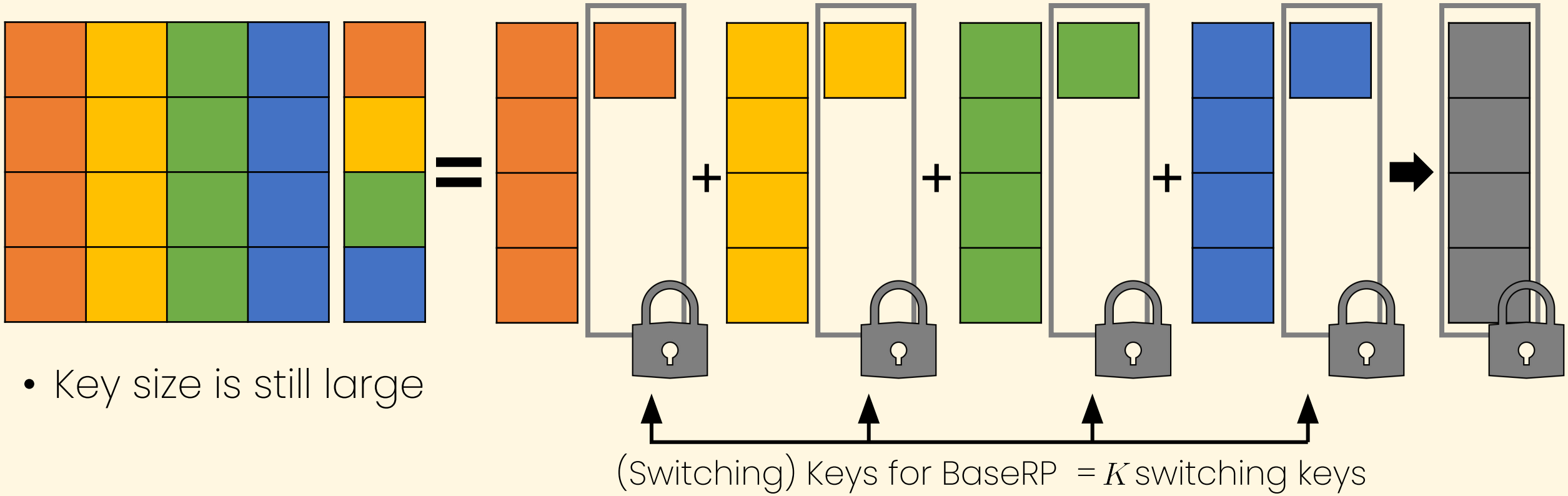
HERMES

HERMES^0: Column method

HERMES^0: the column method with our optimizations

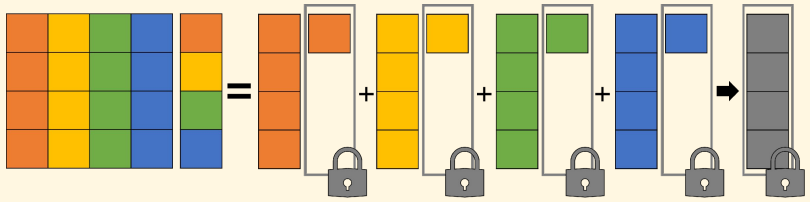
- Practically fast.

K : LWE dimension



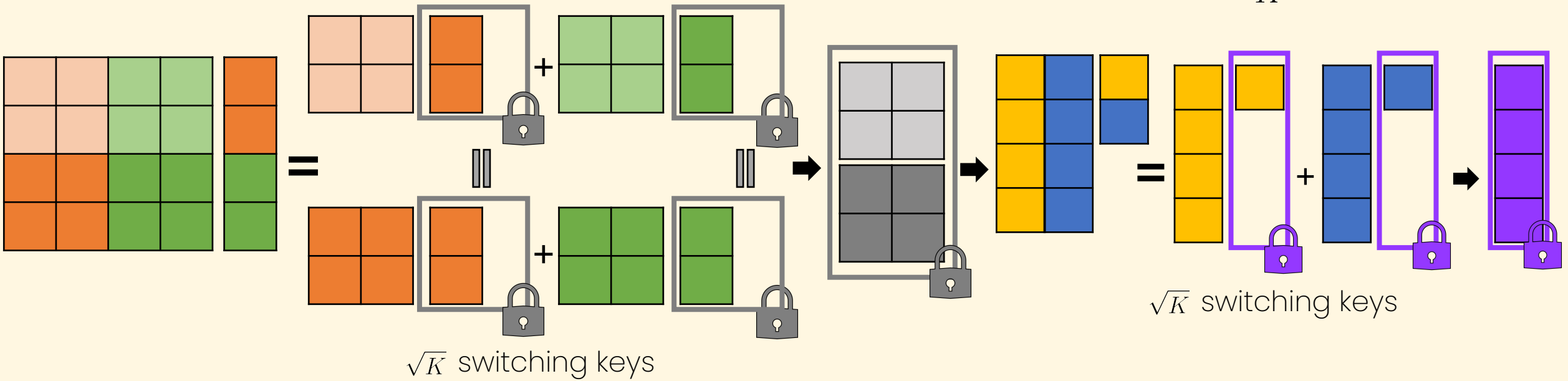
- Key size is still large

HERMES[1]: Block method



HERMES[1]: the *block method* with our optimizations

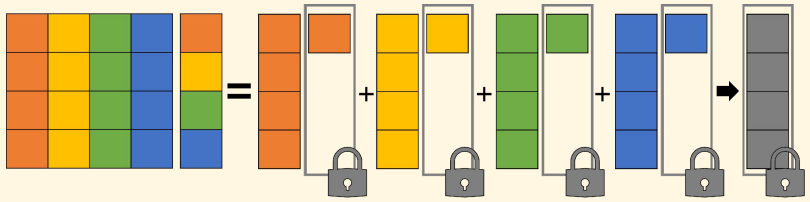
K : LWE dimension



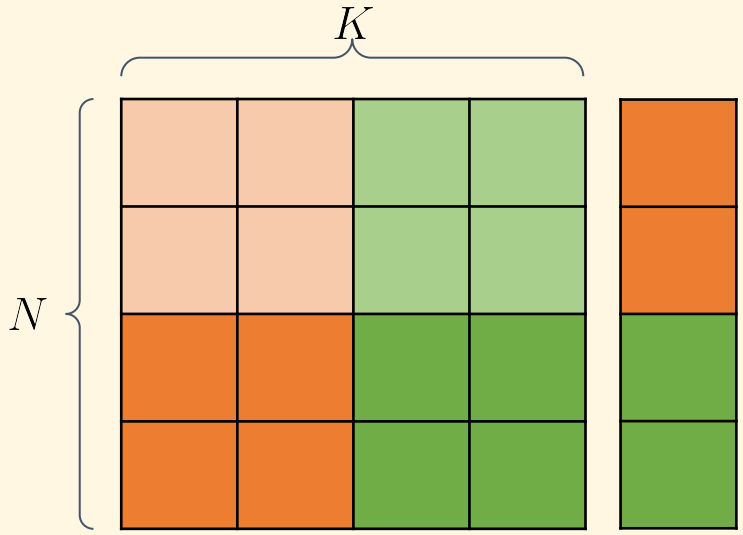
Cost : $K \rightarrow K + \sqrt{K}$ (Mod Up)

Key : $K \rightarrow 2\sqrt{K}$ (Switching Key)

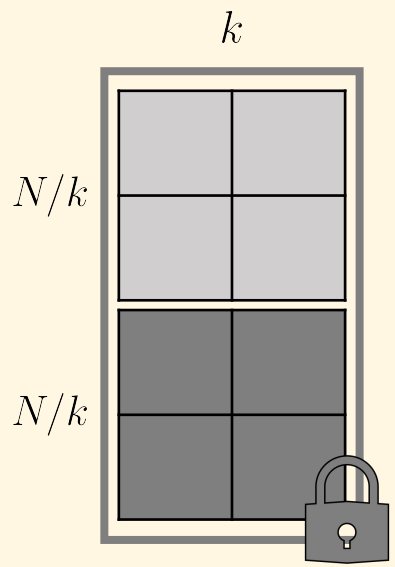
HERMES[1]: Block method



- How to encode the blocks? Module LWE (MLWE)



$$(a_0, a_1, a_2, a_3) \cdot (s_0, s_1, s_2, s_3) + b = m + e \in \mathbb{Z}$$



$$(a_{00} + a_{10}X, a_{01} + a_{11}X) \cdot (s_{00} + s_{10}X, s_{01} + s_{11}X) + (b_0 + b_1X) = m(x) + e(x) \in \mathbb{Z}_q[X]/(X^2 + 1)$$

- How to squeeze the blocks? MLWE key switching / MLWE ring switching

Experimental Results

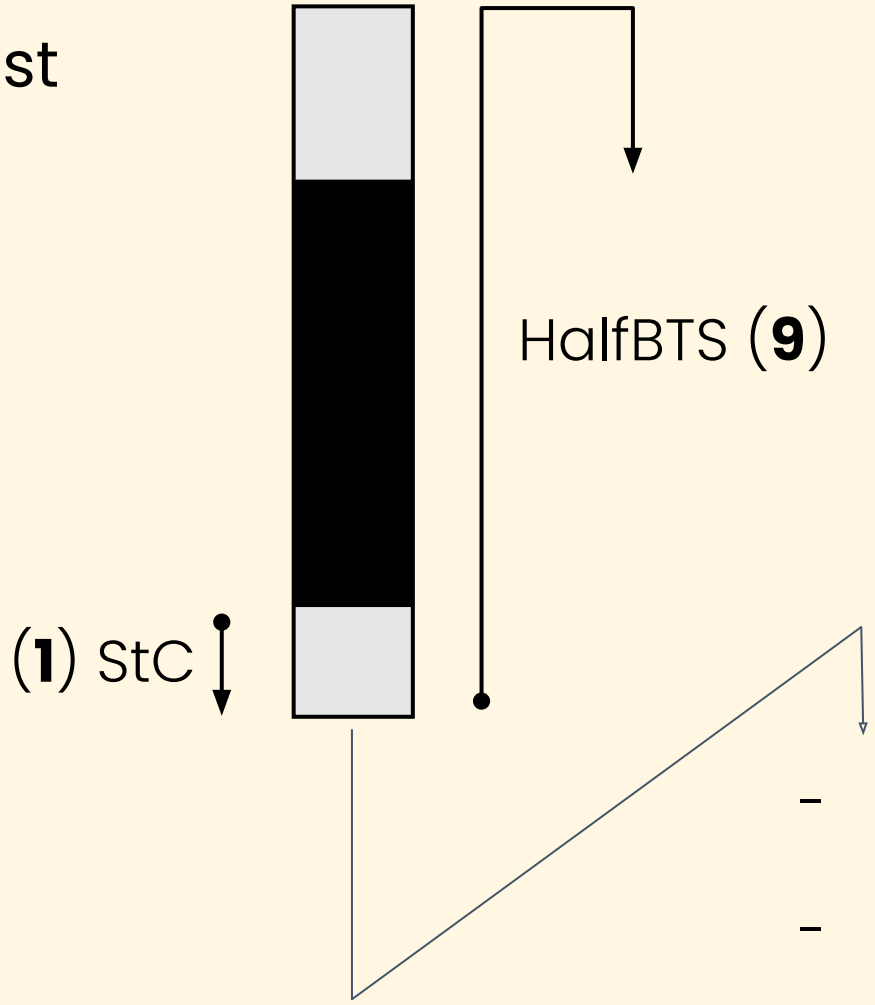
Ring Packing

FHE RP Method		$\log(Q_{comp})$	RLWE degree	# of input LWE	Amortized Time (ms/slot)	Key size (MB)	
						Base RP	Half-BTS
Pegasus [LHH+21]	Diagonal	270	2^{16}	2^{12}	12.62	3540	
[CDKS21] with Optimizations	Row	562	2^{16}	2^{16}	0.83	1	667
HERMES^0	Column	562	2^{16}	2^{16}	0.44	114	667
HERMES^1	Block	562	2^{16}	2^{16}	0.47	6	667
		270	2^{15}	2^{15}	0.31	5	542

All experiments are measured on AMD® Ryzen 7 3700x 8-core processor with a single-threaded CPU.
Pegasus figures are borrowed from [LHH+21]; measured on single-threaded Intel Xeon Platinum 8269CY CPU (20-cores) at 2.50GHz.

Application in CKKS

Expected cost



BTS
= StC + HalfBTS
= **10**

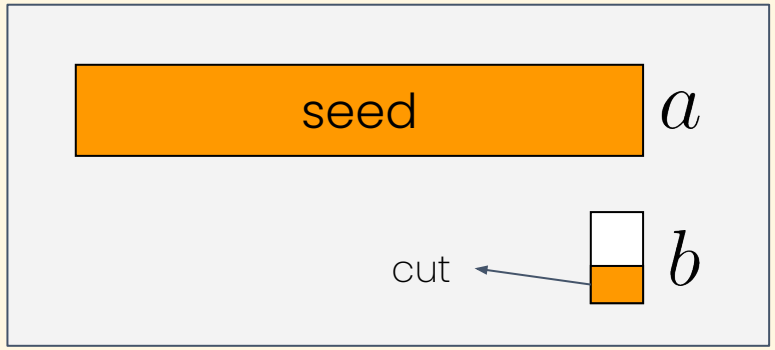
Reordering

= **15** = 1.5 * BTS

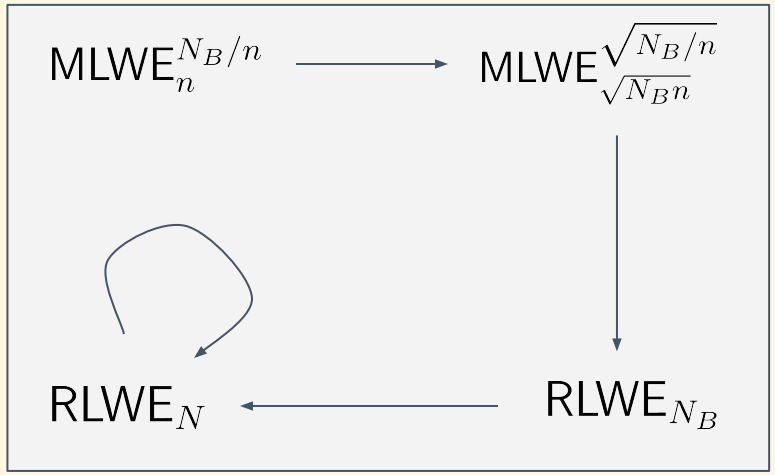
- Whatever LWE operations (**?**)
- BaseHERMES (**5**)

Transciphering

- Input (M)LWE ciphertexts
  - Encrypt (scaled up) real numbers
  - Use XOF on a public seed
  - Cut least significant bits



- Output RLWE ciphertext
  - Slot-encoded CKKS ciphertext
  - Use HERMES[1] with a midpoint at the middle



Transciphering

Scheme	RLWE degree	Granularity	Latency (s)	Expansion Ratio	Mean Precision
HERA [CHK+21]	2^{16}	**16**	141.6	1.24	19.1
Rubato [HKL+22]		64	106.4	1.26	18.9
		16	71.1	1.31	18.8
HERMES		64	25.8	1.58	23.0
		16	25.7	1.58	23.0
		1	30.9	1.58	23.0

All experiments are measured on AMD® Ryzen 7 3700x 8-core processor with a single-threaded CPU.
HERA and Rubato figures are borrowed from [CHK+21] and [HKL+22]; measured on AMD Ryzen 7 2700X @ 3.70 GHz single-threaded CPU.

Wrapping up!

- Optimize the existing RP methods for FHE RP.
- Propose a new efficient FHE RP method, HERMES.
  - Compared to state-of-the-art, 40x faster in terms of throughput
- Application to tranciphering

eprint 2023/1244

Thank you!

References

[BGGJ20] C. Boura, N. Gama, M. Georgieva, and D. Jetchev. CHIMERA: combining ring-LWE-based fully homomorphic encryption schemes. J. Math. Cryptol., 2020.

[CDKS21] H. Chen, W. Dai, M. Kim, and Y. Song. Efficient homomorphic conversion between (ring) LWE ciphertexts. In ACNS, 2021.

[CGGI17] I. Chillotti, N. Gama, M. Georgieva, and M. Izabach`ene. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In ASIACRYPT, 2017

[CHK+21] J. Cho, J. Ha, S. Kim, B. Lee, J. Lee, J. Lee, D. Moon, and H. Yoon. Transciphering framework for approximate homomorphic encryption. In ASIACRYPT, 2021.

[GHPS13] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart. Field switching in BGV-style homomorphic encryption. Journal of Computer Security, 2013.

[HKL+22] J. Ha, S. Kim, B. Lee, J. Lee, and M. Son. Rubato: Noisy ciphers for approximate homomorphic encryption. In EUROCRYPT, 2022.

[HS14] S. Halevi and V. Shoup. Algorithms in HElib. In CRYPTO, 2014.

[LHH+21] W.-J. Lu, Z. Huang, C. Hong, Y. Ma, and H. Qu. PEGASUS: bridging polynomial and non-polynomial evaluations in homomorphic encryption. In S&P, 2021.

[MS18] D. Micciancio and J. Sorrell. Ring packing and amortized FHEW bootstrapping. In ICALP, 2018.