ZAMA

FHE.org 2026 • Tapei • March 8, 2026

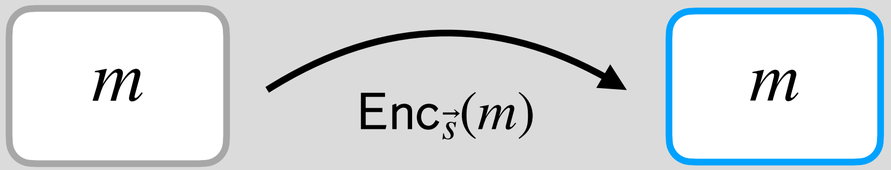
Accelerating TFHE With Sorted Bootstrapping Techniques

Loris Bergerat[1,2], Jean-Baptiste Orfila[1], Adeline Roux-Langlois[2], Samuel Tap[1]

[1]Zama, France [2]CNRS, Greyc, Caen, France

Introduction To (T)FHE

LWE Ciphertext



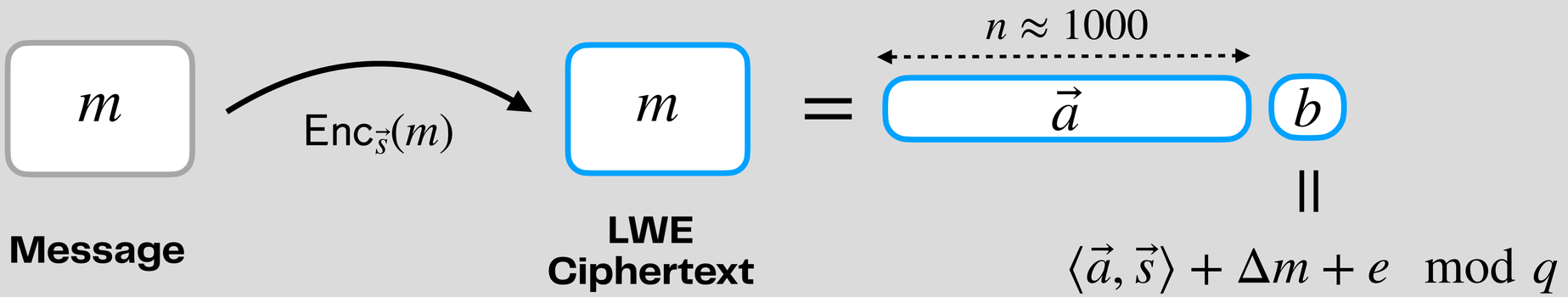
Message

LWE Ciphertext

Secret key	$\vec{s} \leftarrow \mathcal{U}(\{0,1\})^n$
Mask	$\vec{a} \leftarrow \mathcal{U}(\mathbb{Z}_q)^n$
Error (a.k.a. noise)	$e \leftarrow \mathcal{N}(0,\sigma^2)$

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In STOC 2005. ACM, 2005.

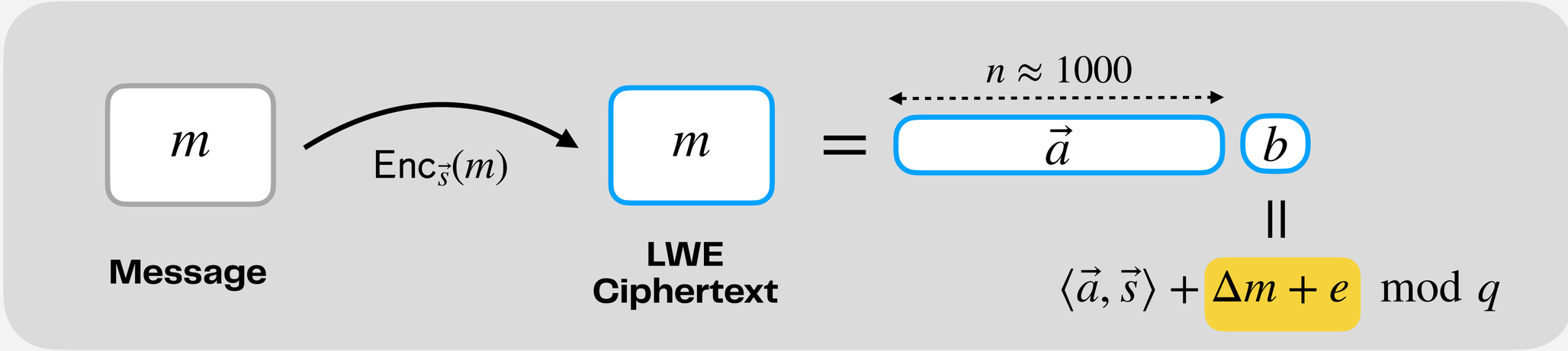
LWE Ciphertext



Secret key	$\vec{s} \leftarrow \mathcal{U}(\{0,1\})^n$
Mask	$\vec{a} \leftarrow \mathcal{U}(\mathbb{Z}_q)^n$
Error (a.k.a. noise)	$e \leftarrow \mathcal{N}(0,\sigma^2)$

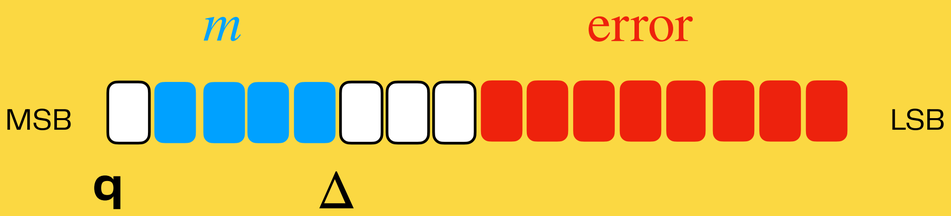
[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In STOC 2005. ACM, 2005.

LWE Ciphertext



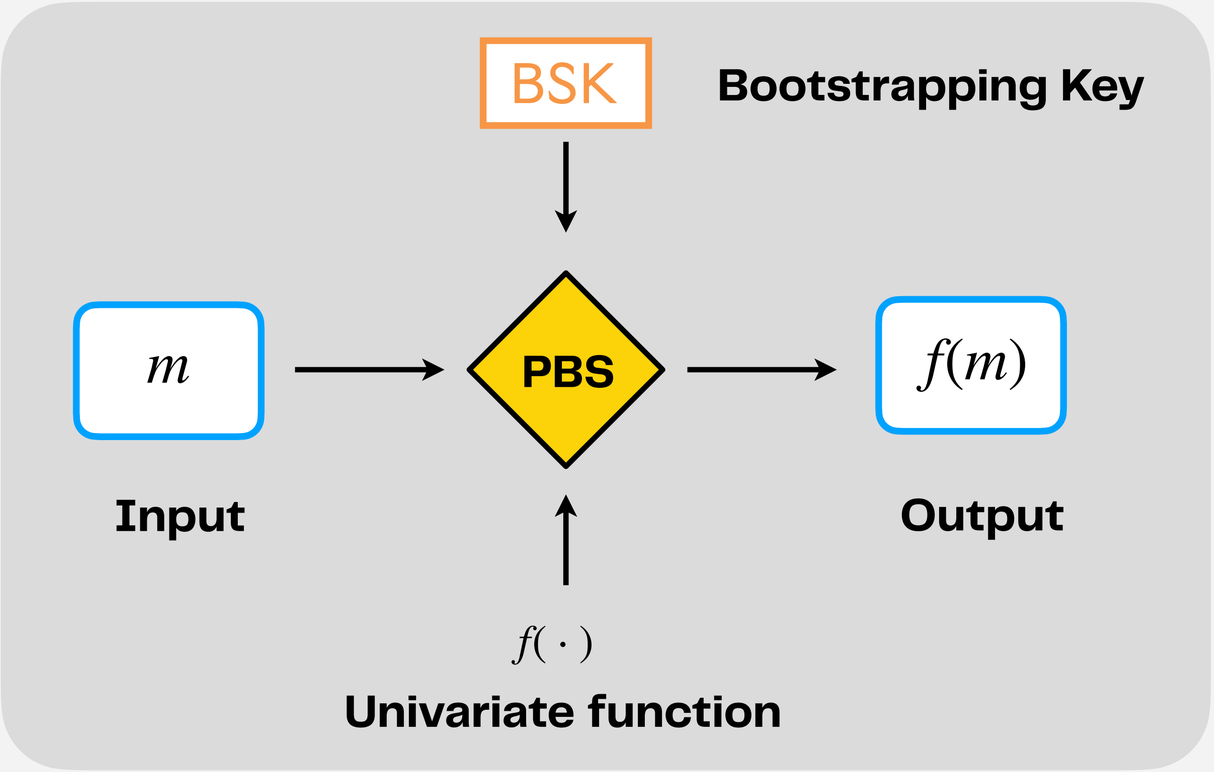
Secret key	$\vec{s} \leftarrow \mathcal{U}(\{0,1\})^n$
Mask	$\vec{a} \leftarrow \mathcal{U}(\mathbb{Z}_q)^n$
Error (a.k.a. noise)	$e \leftarrow \mathcal{N}(0,\sigma^2)$

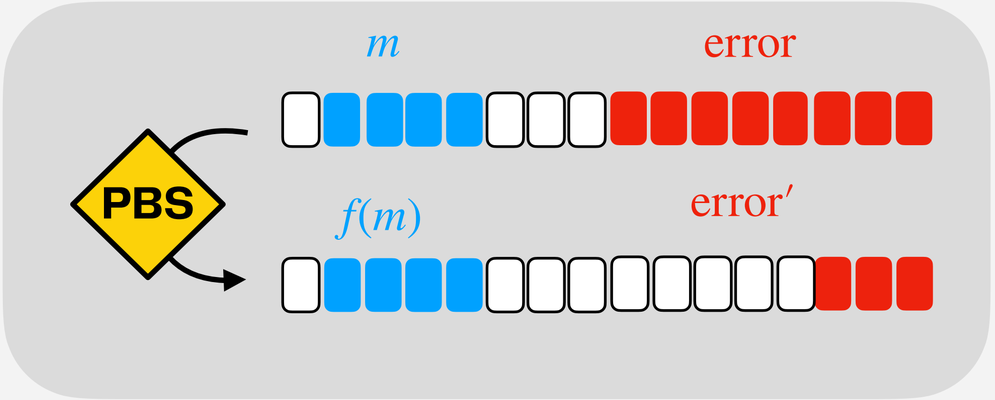
Plaintext Representation



[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In STOC 2005. ACM, 2005.

TFHE Programmable Bootstrapping (PBS)





Small messages only:
$|m| \leq 8$ bits

Noise reduction &

Homomorphic evaluation of any $f(\cdot)$

[Gen09]Craig Gentry. Fully homomorphic encryption using ideal lattices. In Proceedings the 41st Annual ACM Symposium on Theory of Computing, STOC 2009
[CGGI20]Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. TFHE:fast fully homomorphic encryption over the torus. Journal of Cryptology, 2020

Challenge

PBSs account for about 90% of the total execution time of a program.

5

Challenge

PBSs account for about 90% of the total execution time of a program.

Limitations

- Composed of costly polynomial operations
- Sequential Operation

Challenge

PBSs account for about 90% of the total execution time of a program.

Limitations

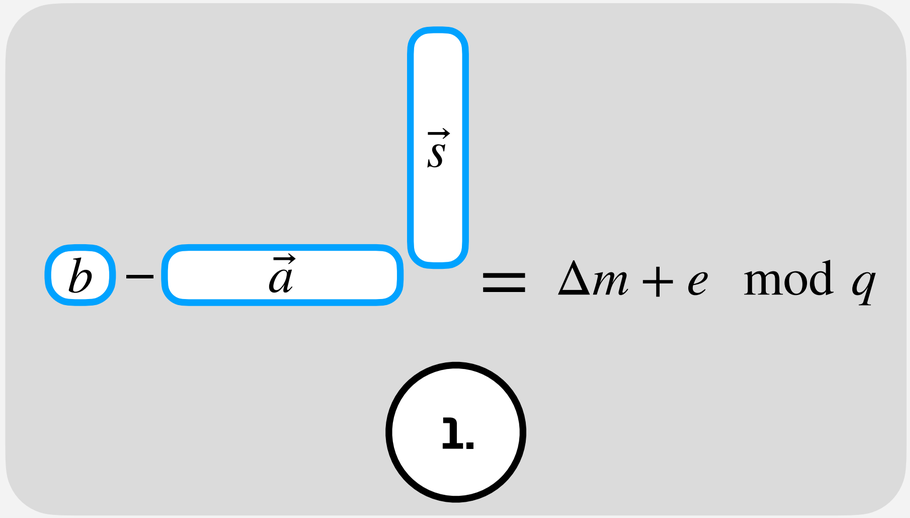
Composed of costly polynomial operations

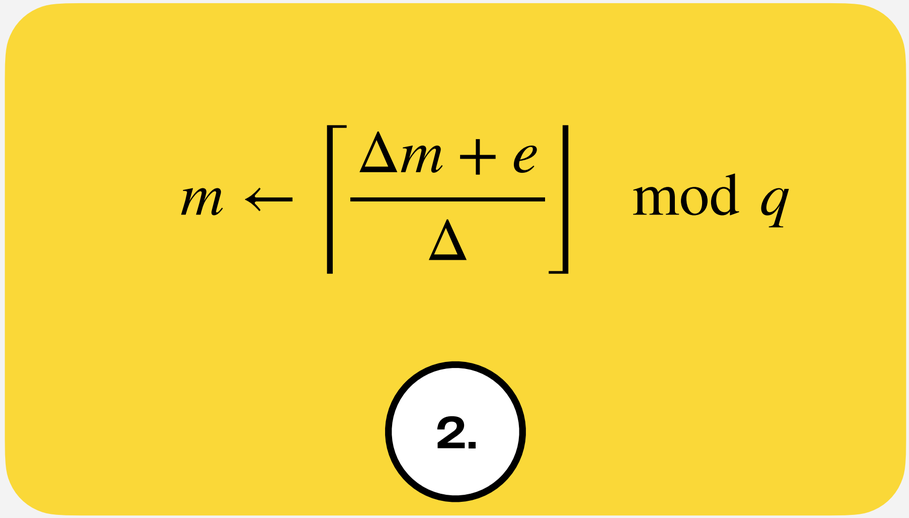
Sequential Operation

How to reduce the cost of the TFHE Bootstrapping ?

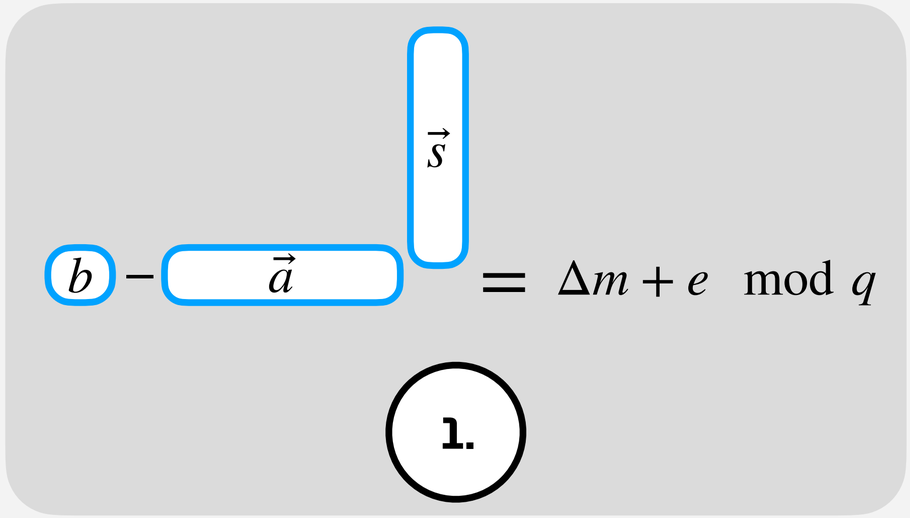
TFHE Bootstrapping

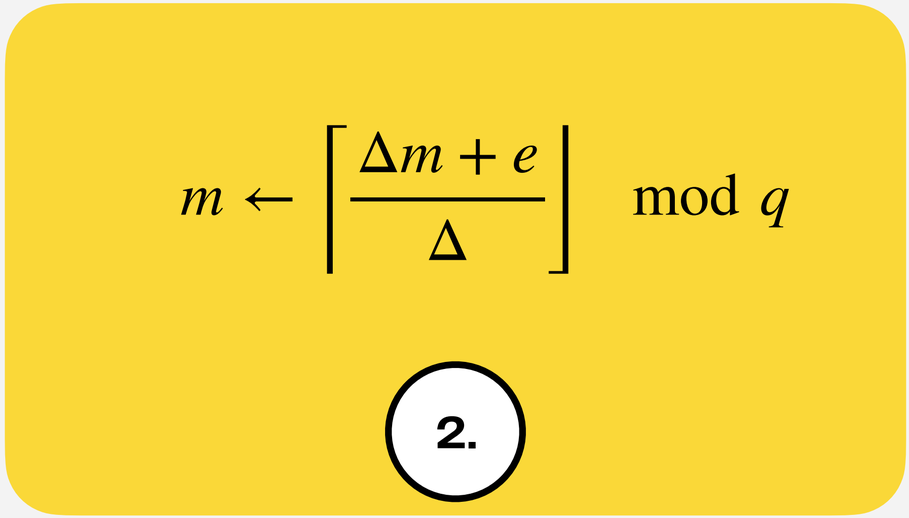
Bootstrapping





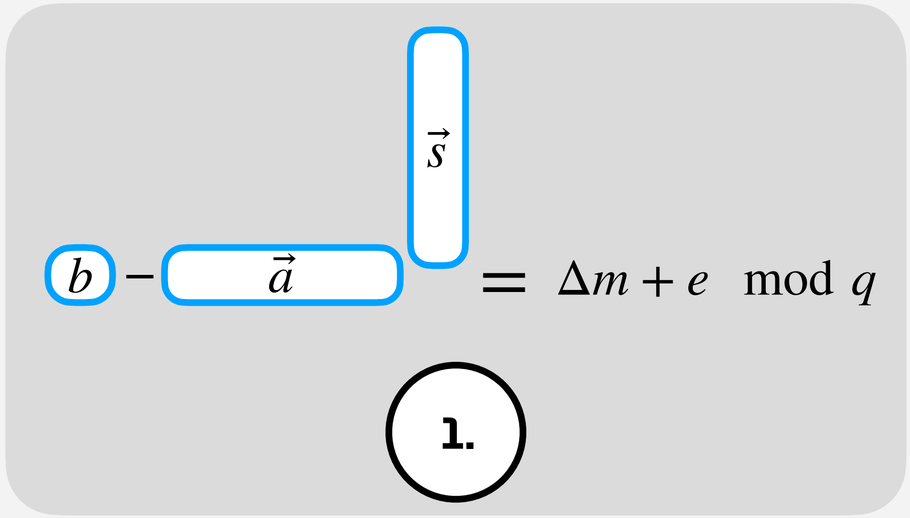
Bootstrapping

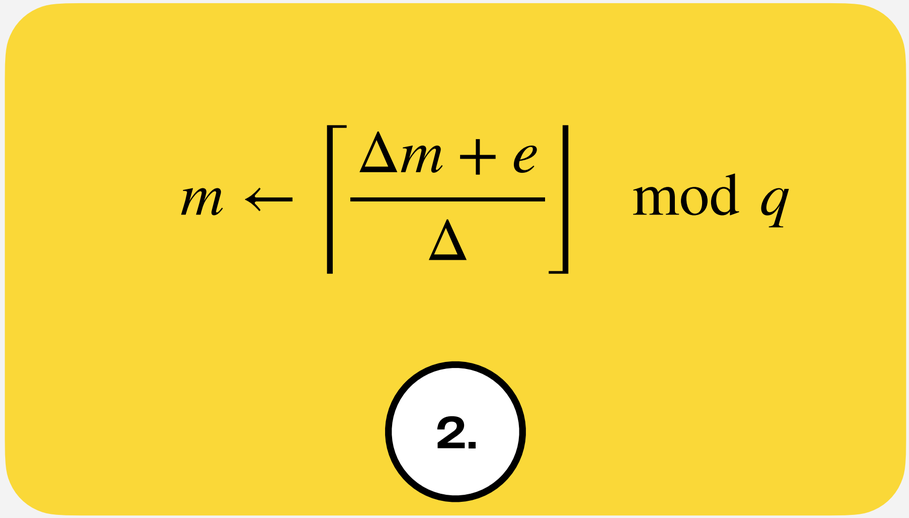




$$X^{-b+\langle a,s \rangle} \cdot \mathrm{LUT} = m + mX + \cdots + \in \mathbb{Z}_q[X]/(X^N + 1)$$

Bootstrapping





$$X^{-(\Delta m+e)} \cdot \mathrm{LUT} = m + mX + \cdots + \in \mathbb{Z}_q[X]/(X^N+1)$$

Bootstrapping

$$b - \vec{a} \cdot \vec{s} = \Delta m + e \mod q$$

1.

Encrypt the **input message** m in the **first coefficient**

$$m \leftarrow \left\lceil \frac{\Delta m + e}{\Delta} \right\rfloor \mod q$$

2.

$$X^{-(\Delta m + e)} \cdot \mathrm{LUT} = m + mX + \cdots + \in \mathbb{Z}_q[X]/(X^N + 1)$$

Bootstrapping

$$b - \vec{a} \cdot \vec{s} = \Delta m + e \mod q$$

1.

Encrypt the **input message** m in the **first coefficient**

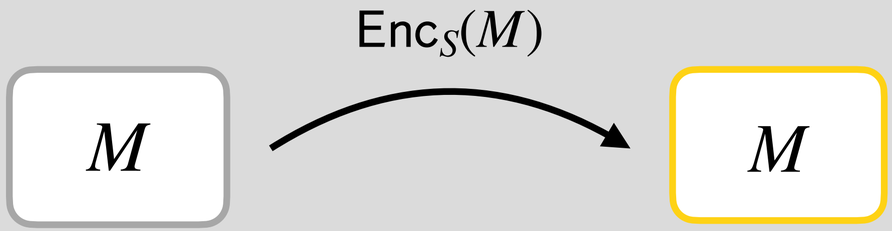
$$m \leftarrow \left\lceil \frac{\Delta m + e}{\Delta} \right\rfloor \mod q$$

2.

$$X^{-(\Delta m + e)} \cdot \mathrm{LUT} = m + mX + \cdots + \in \mathbb{Z}_q[X]/(X^N + 1)$$

Homomorphically perform a rotation of a redundant lookup table (LUT) represented as polynomial

Ring LWE ciphertext

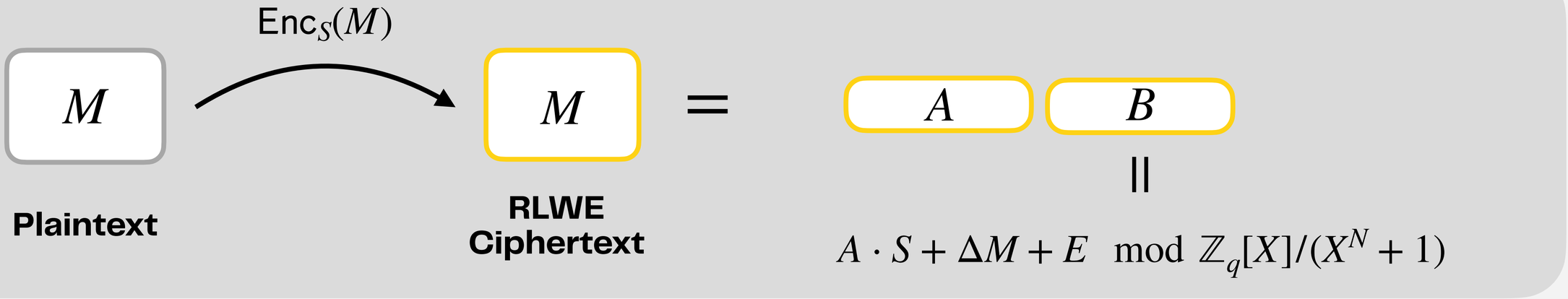


Secret key	$S \leftarrow \mathcal{U}(\{0,1\})^N$
Mask	$A \leftarrow \mathcal{U}(\mathbb{Z}_q[X]/(X^N+1))$
Error (a.k.a. noise)	$E \leftarrow \mathcal{N}(0,\sigma^2)^N$

Based on the **RLWE** assumption

[SSTX09] Damien Stehle, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In ASIACRYPT 2009. Springer, 2009.
[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In EUROCRYPT 2010. Springer, 2010.

Ring LWE ciphertext

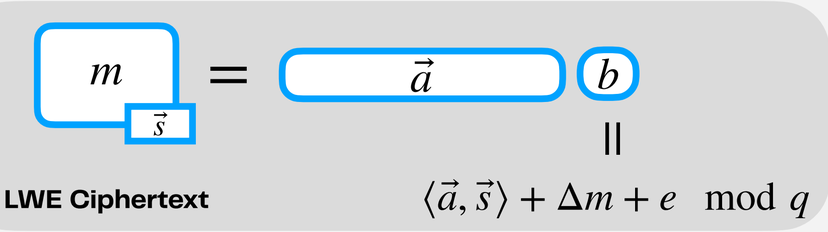


Secret key	$S \leftarrow \mathcal{U}(\{0,1\})^N$
Mask	$A \leftarrow \mathcal{U}(\mathbb{Z}_q[X]/(X^N + 1))$
Error (a.k.a. noise)	$E \leftarrow \mathcal{N}(0,\sigma^2)^N$

Based on the **RLWE** assumption

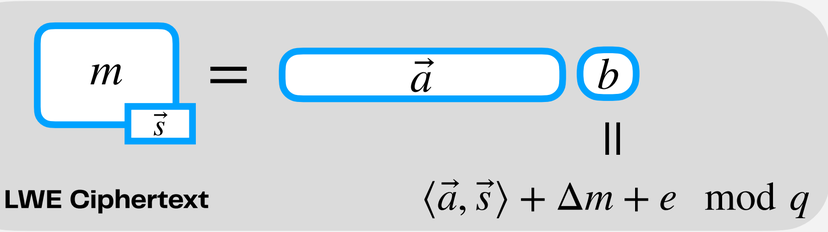
[SSTX09] Damien Stehle, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In ASIACRYPT 2009. Springer, 2009.
[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In EUROCRYPT 2010. Springer, 2010.

Bootstrapping



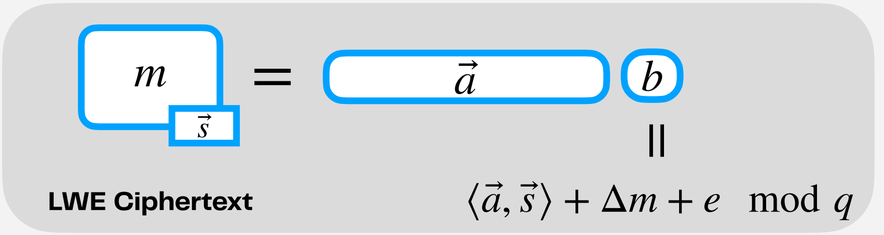
$$X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \text{LUT} = m + mX + \cdots +$$

Bootstrapping

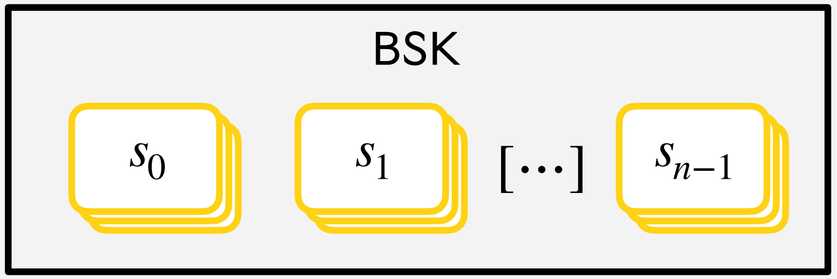


$$X^{-b} \cdot \mathsf{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathsf{LUT} = m + mX + \cdots +$$

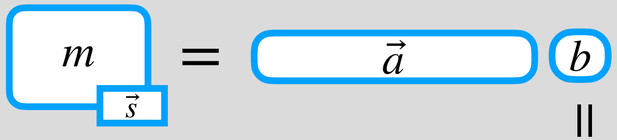
Bootstrapping



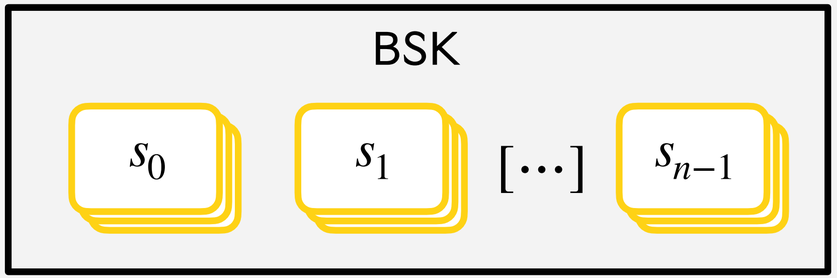
$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$



Bootstrapping



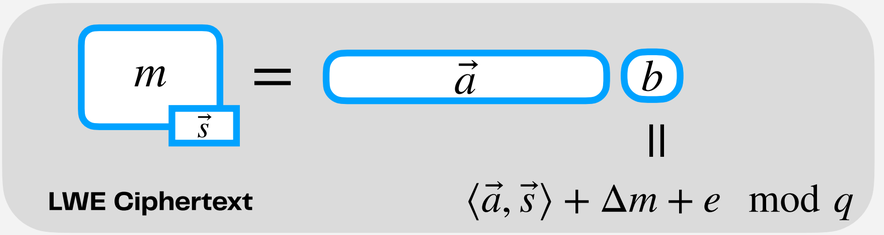
$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$



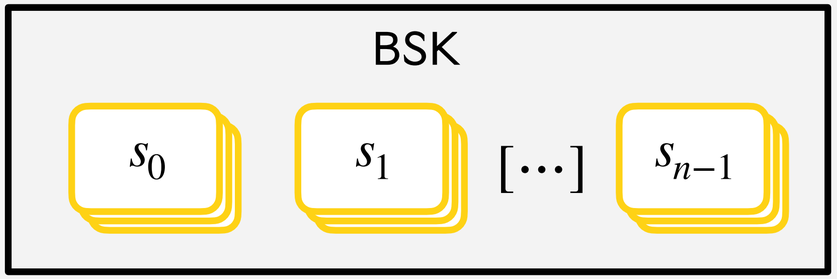
$\mathrm{LUT} \cdot X^{-b}$

$\mathrm{LUT} \cdot X^{-b} \cdot X^{a_0}$

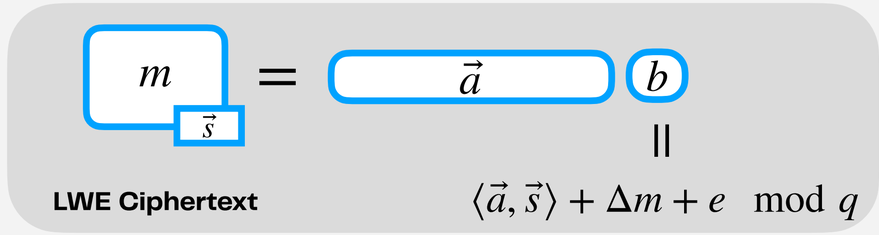
Bootstrapping



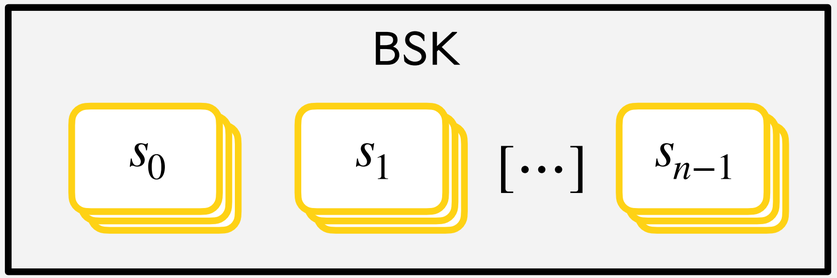
$$\underline{X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0}} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$



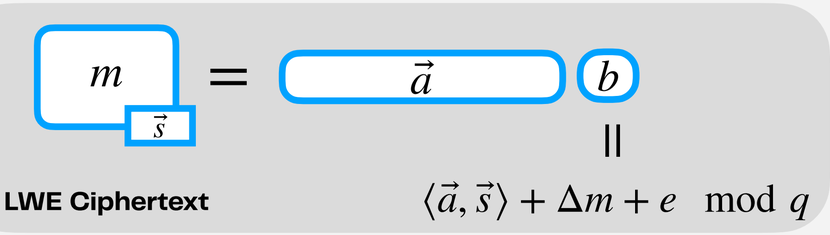
Bootstrapping



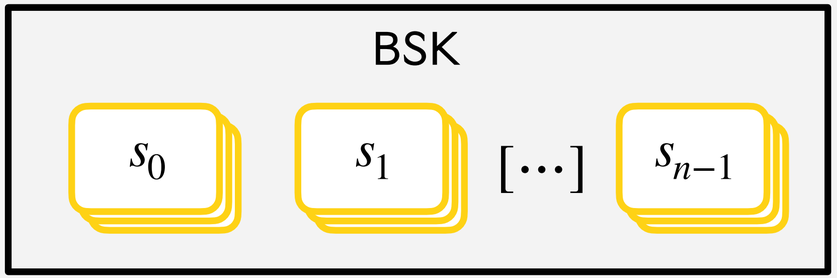
$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$



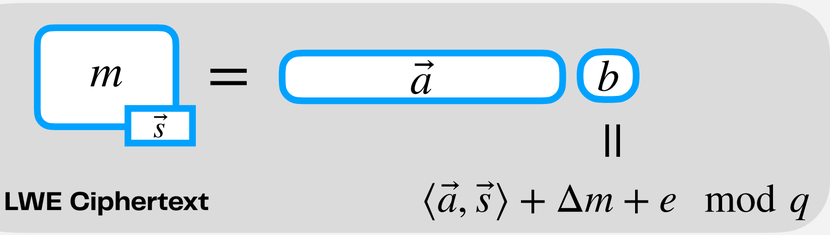
Bootstrapping



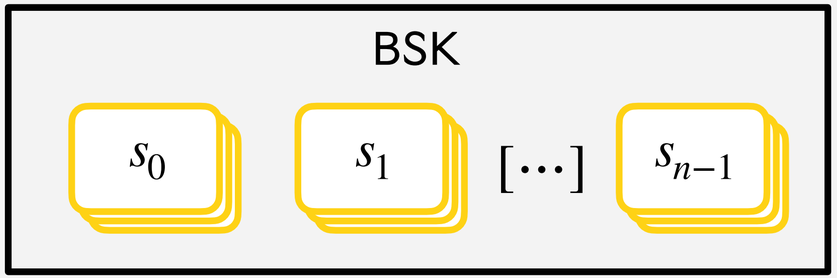
$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$

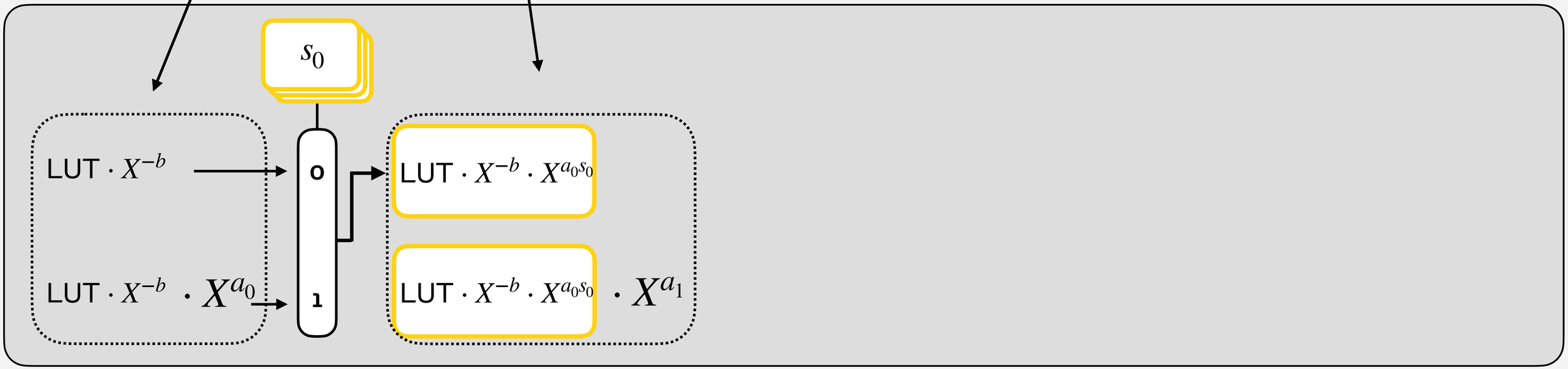


Bootstrapping

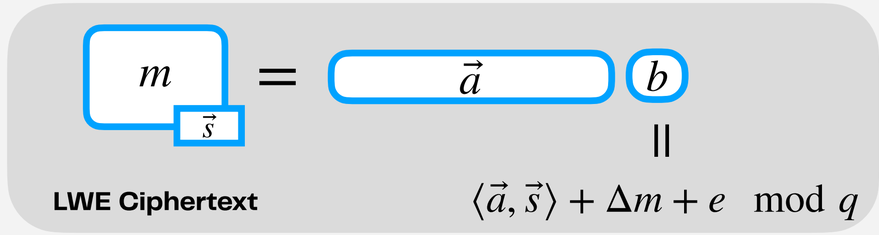


$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$

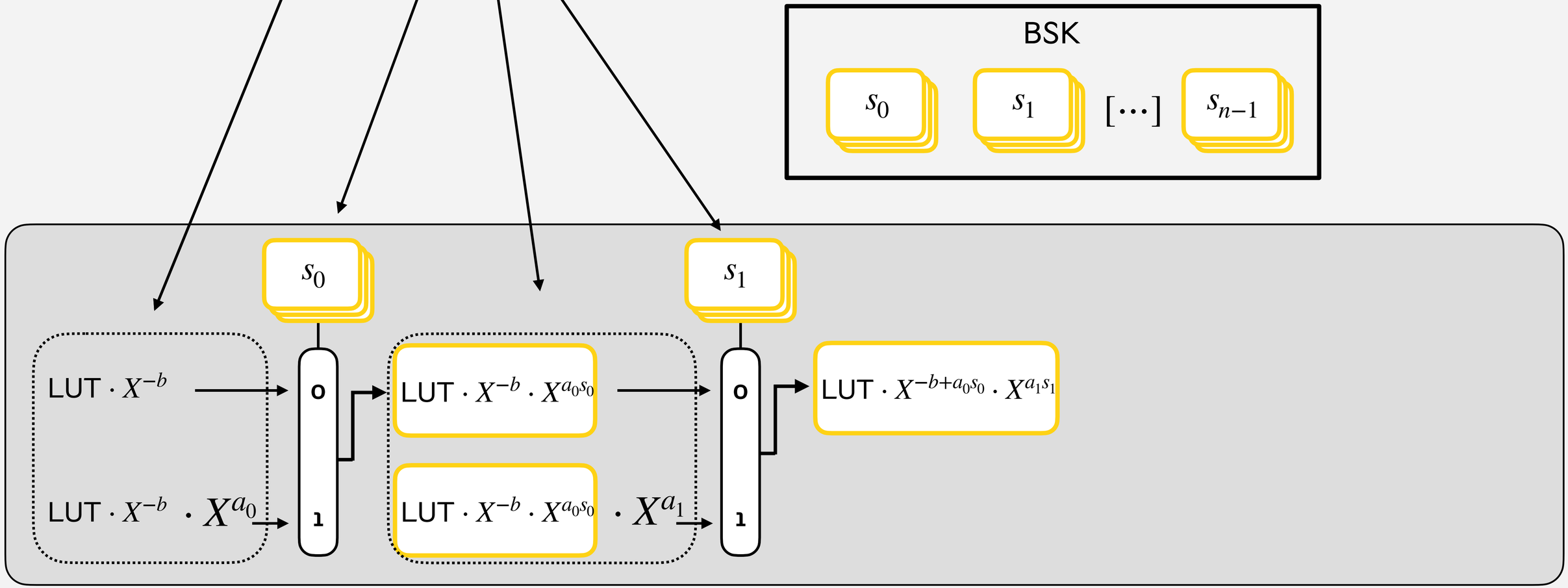




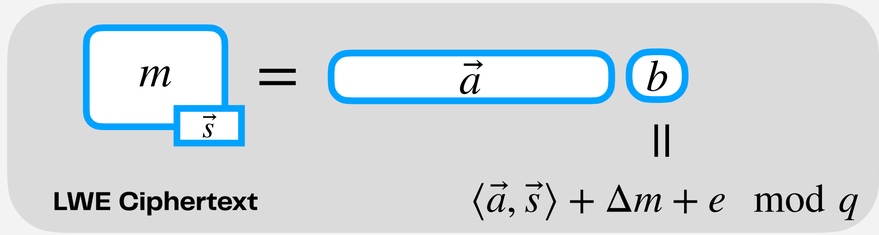
Bootstrapping



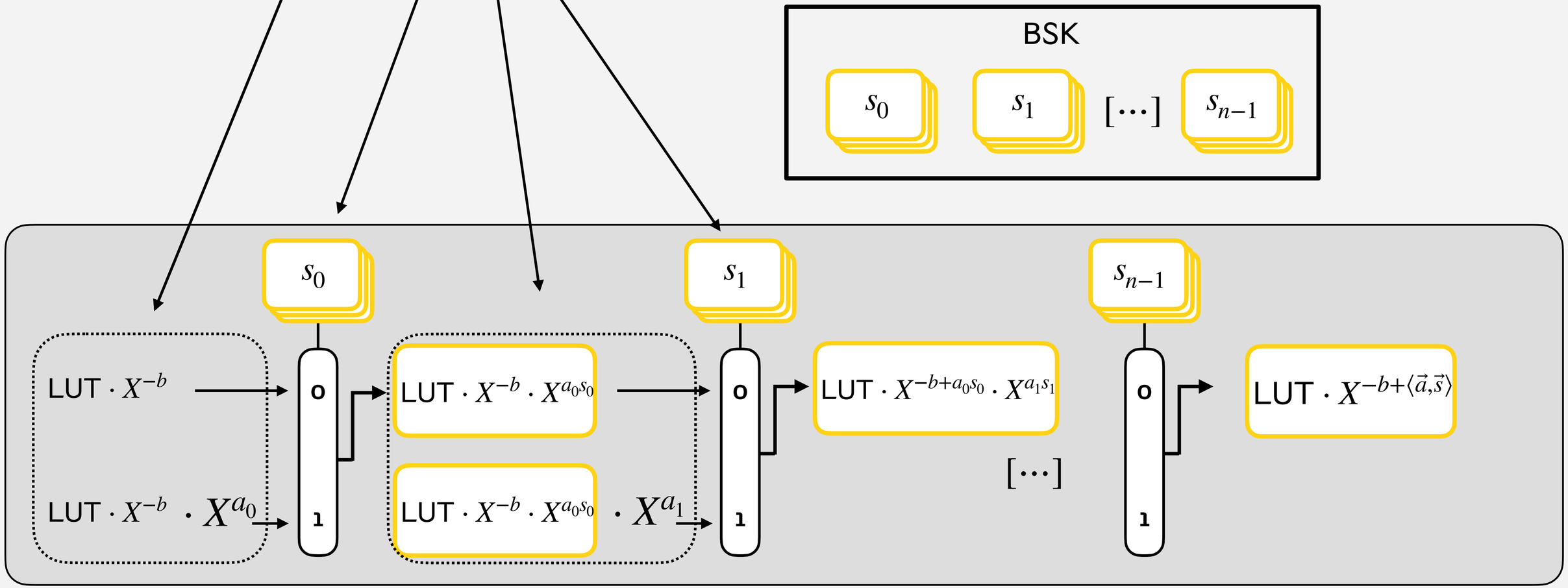
$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$



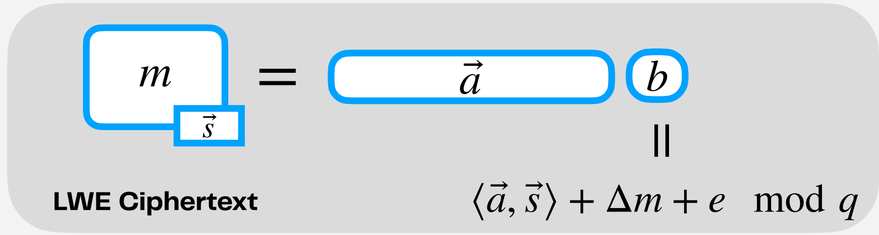
Bootstrapping



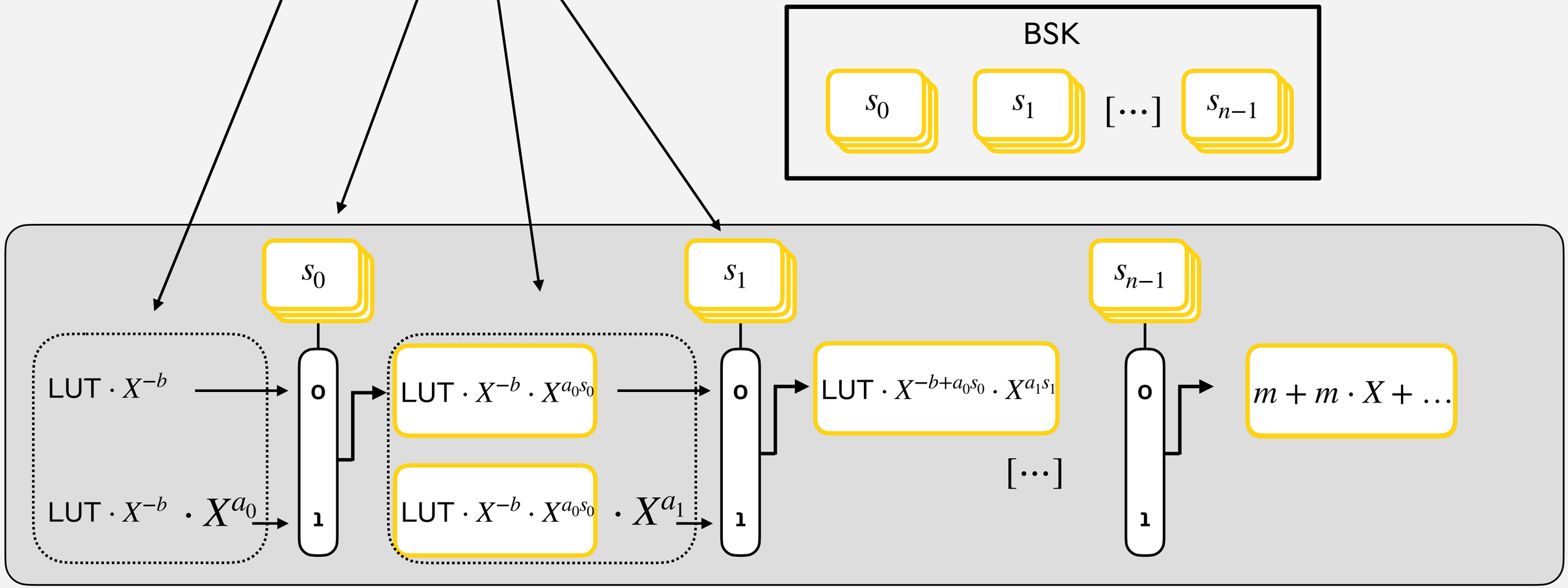
$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$



Bootstrapping



$$X^{-b} \cdot \mathrm{LUT} \cdot X^{a_0 s_0} \cdot X^{a_1 s_1} \cdots X^{a_{n-1} s_{n-1}} = X^{-b+\langle \vec{a}, \vec{s} \rangle} \cdot \mathrm{LUT} = m + mX + \cdots +$$

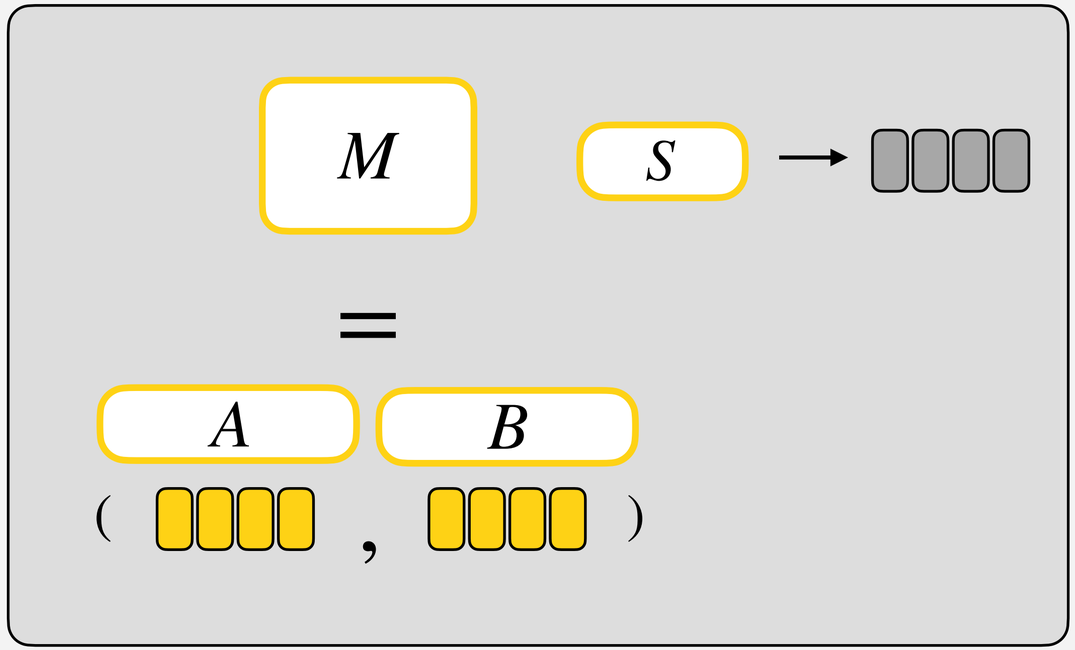


Representation

$$P = p_0 + p_1X + p_2X^2 + p_3X^3 \longrightarrow$$ 

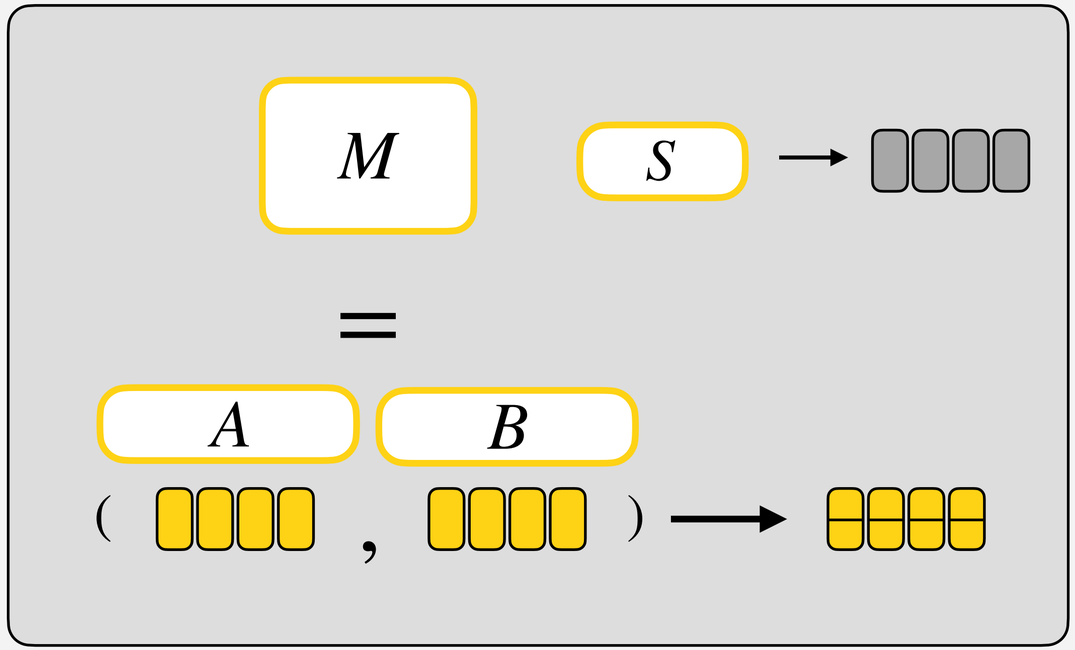
Representation

$$P = p_0 + p_1X + p_2X^2 + p_3X^3 \longrightarrow$$ 



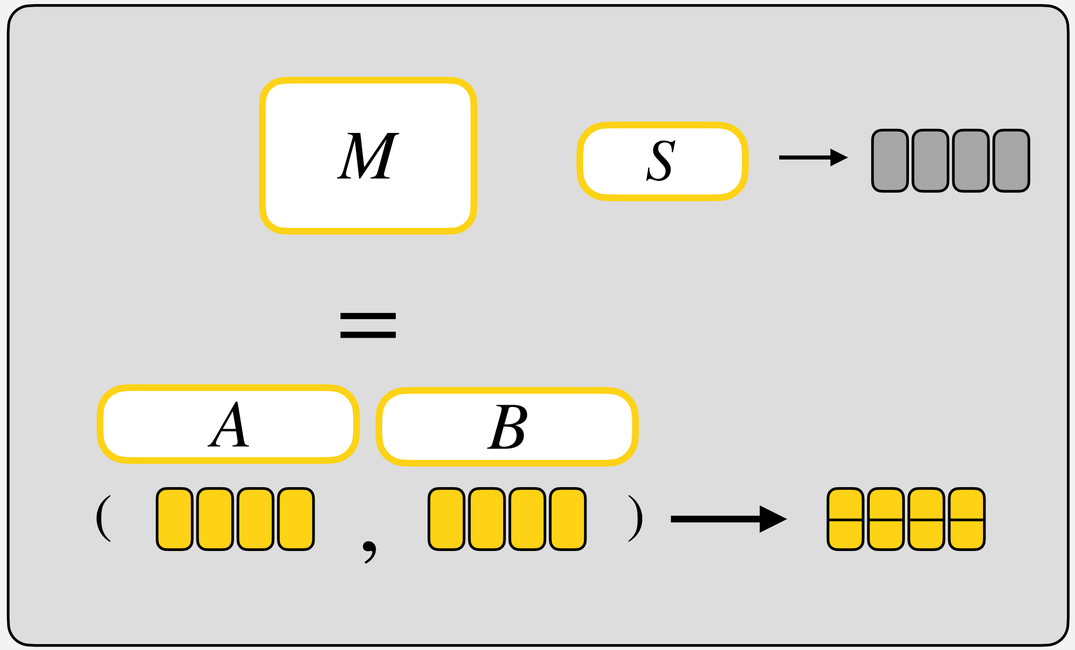
Representation

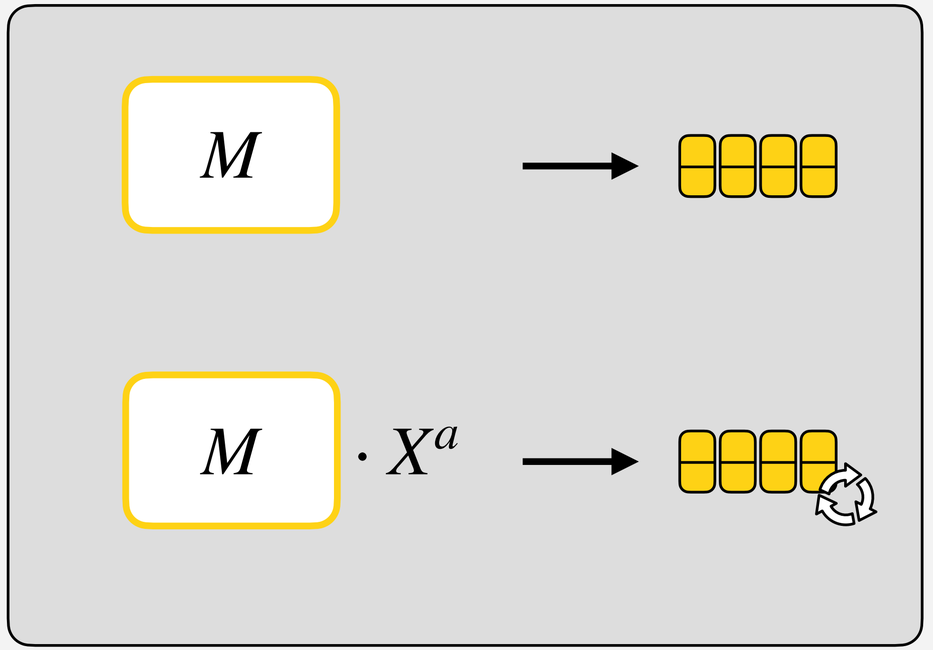
$$P = p_0 + p_1X + p_2X^2 + p_3X^3 \longrightarrow$$



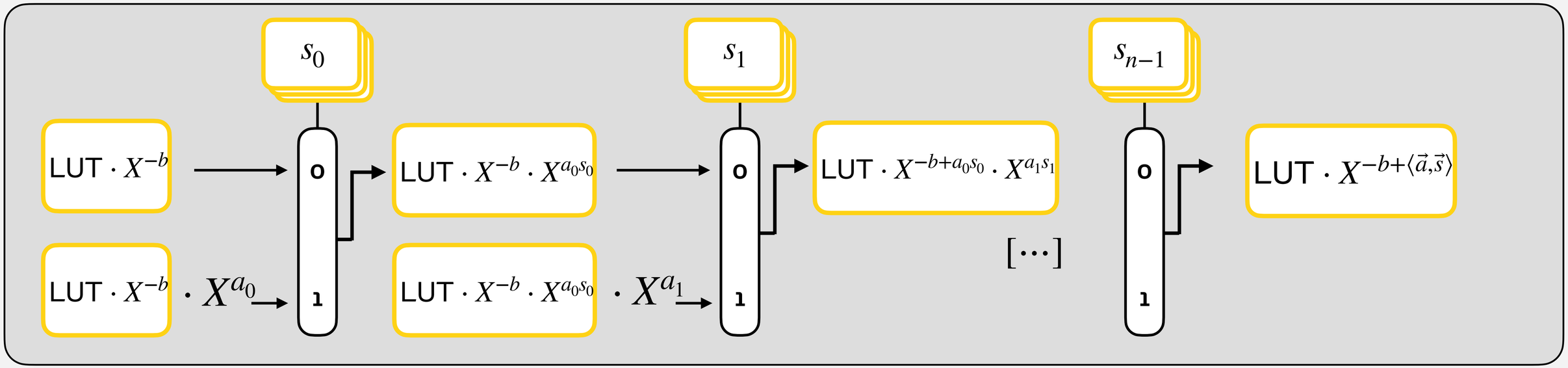
Representation

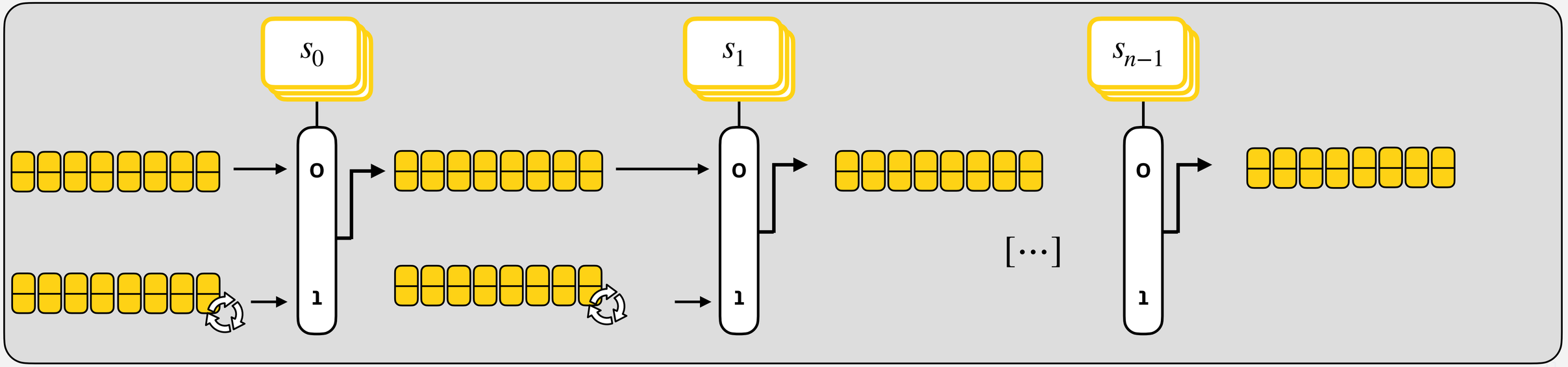
$$P = p_0 + p_1X + p_2X^2 + p_3X^3 \longrightarrow$$





Bootstrapping

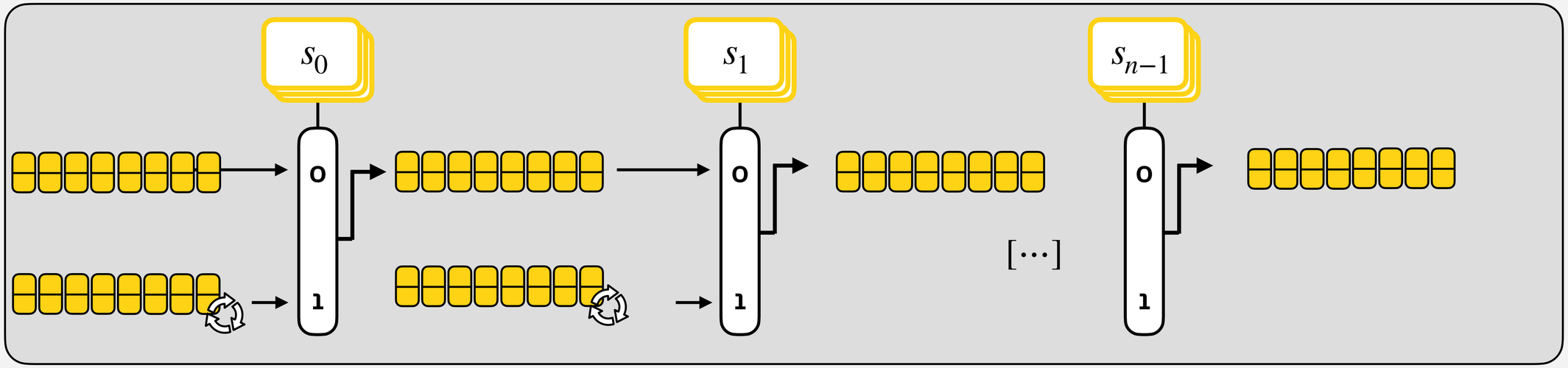




PBS with [LY23]

[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

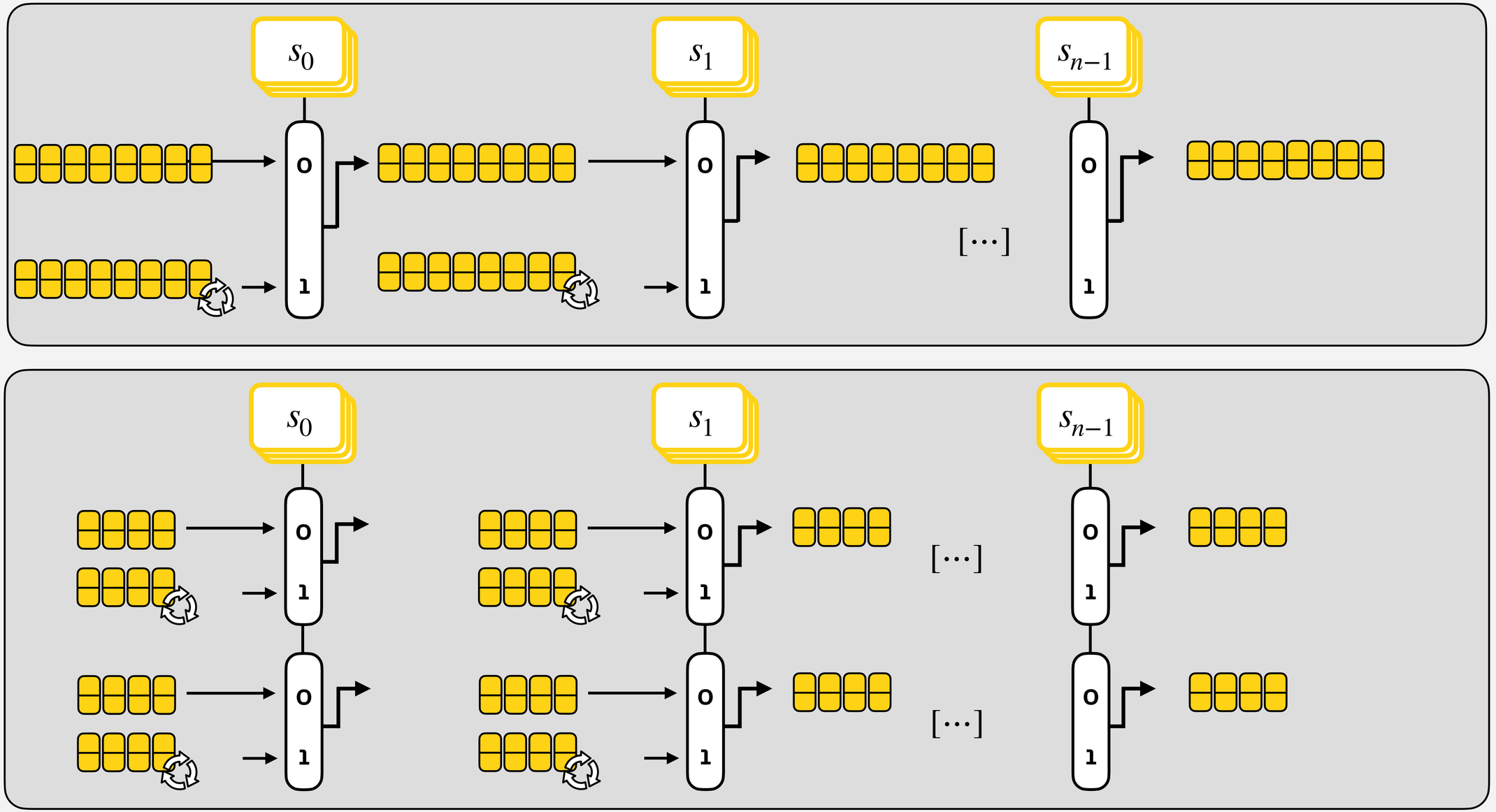
Extended Bootstrapping



[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

13

Extended Bootstrapping

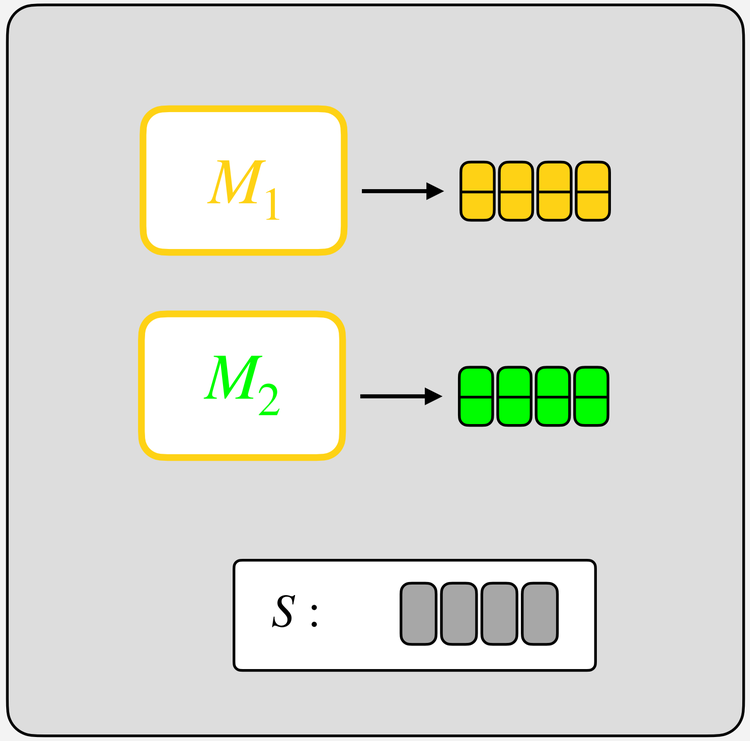


[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

13

Context [LY23]

$$\mathcal{R}_{q,N} = Z_q[X]/X^N + 1$$

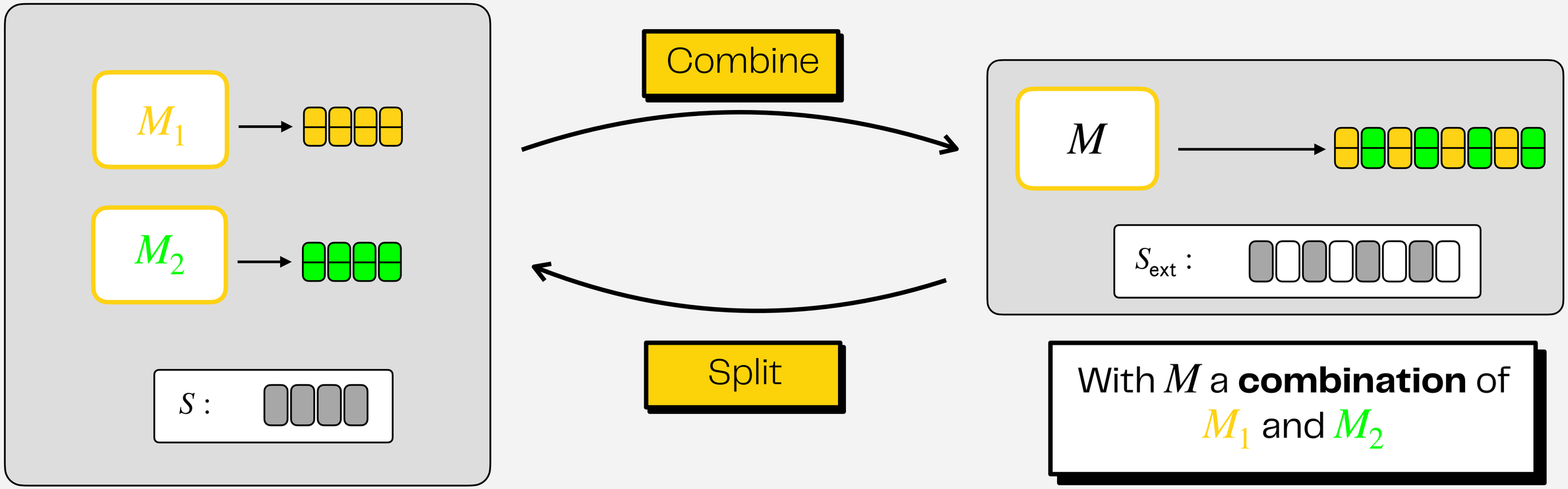


[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

14

Context [LY23]

$$\mathcal{R}_{q,N} = Z_q[X]/X^N + 1$$



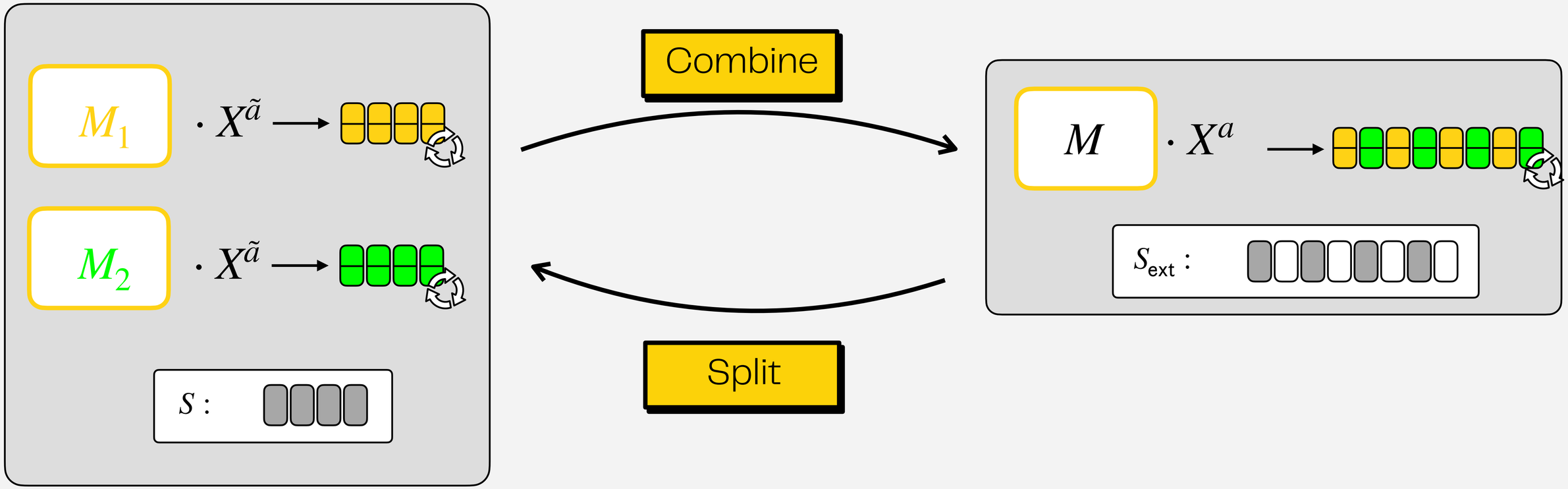
With M a **combination** of M_1 and M_2

2^η **ciphertexts** in $\mathcal{R}_{q,N}^{k+1}$ can be combined in **one ciphertext** in $\mathcal{R}_{q,2^\eta N}^{k+1}$ with a secret key composed of $2^\eta - 1$ known zeros between two secret coefficients.

[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

Context [LY23]

$$\mathcal{R}_{q,N} = Z_q[X]/X^N + 1$$



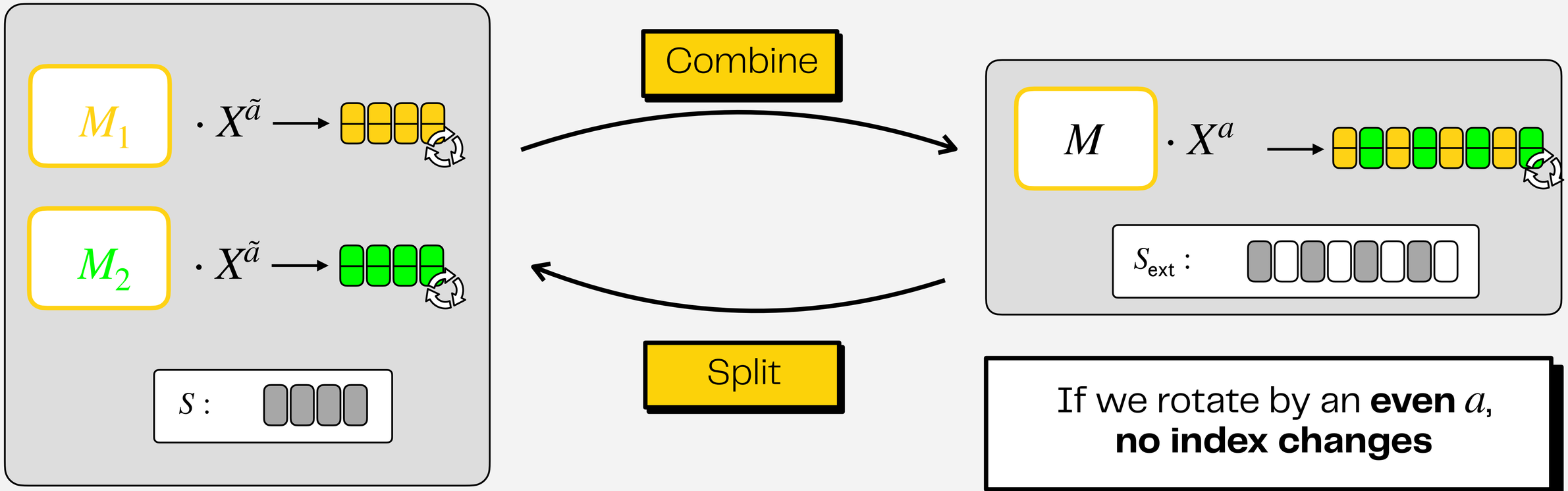
A rotation of the extended RLWE ciphertext corresponds to **changing the index** of the small RLWE ciphertexts **plus an inner rotation.**

[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

15

Context [LY23]

$$\mathcal{R}_{q,N} = Z_q[X]/X^N + 1$$



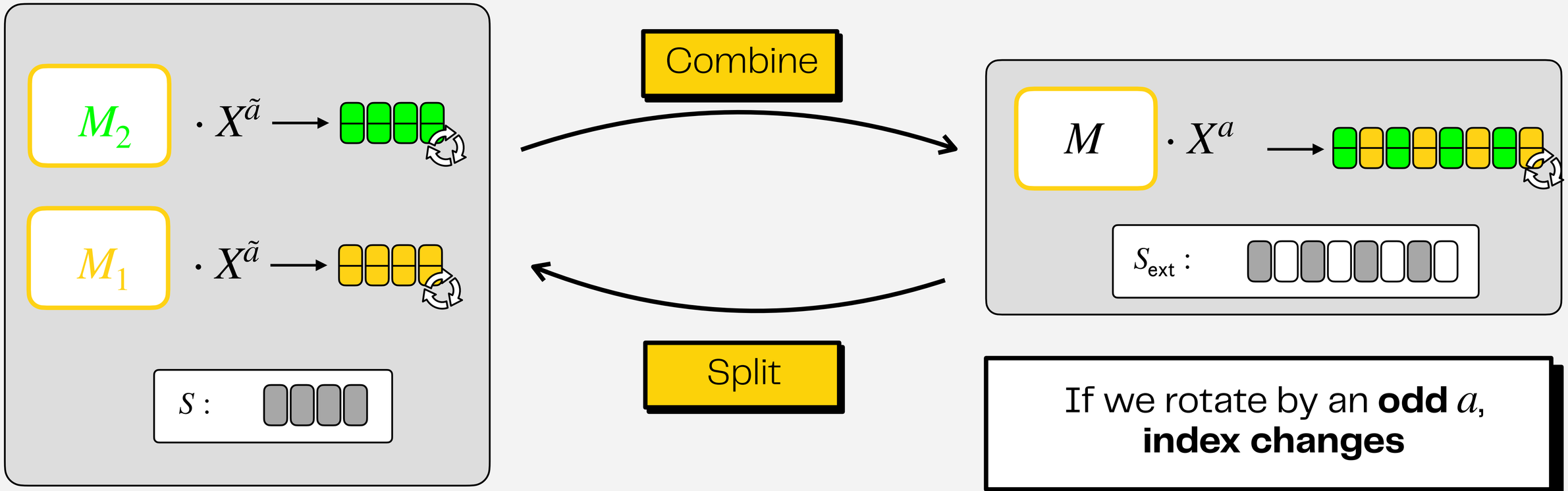
If we rotate by an **even** a, **no index changes**

A rotation of the extended RLWE ciphertext corresponds to **changing the index** of the small RLWE ciphertexts **plus an inner rotation.**

[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

Context [LY23]

$$\mathcal{R}_{q,N} = Z_q[X]/X^N + 1$$



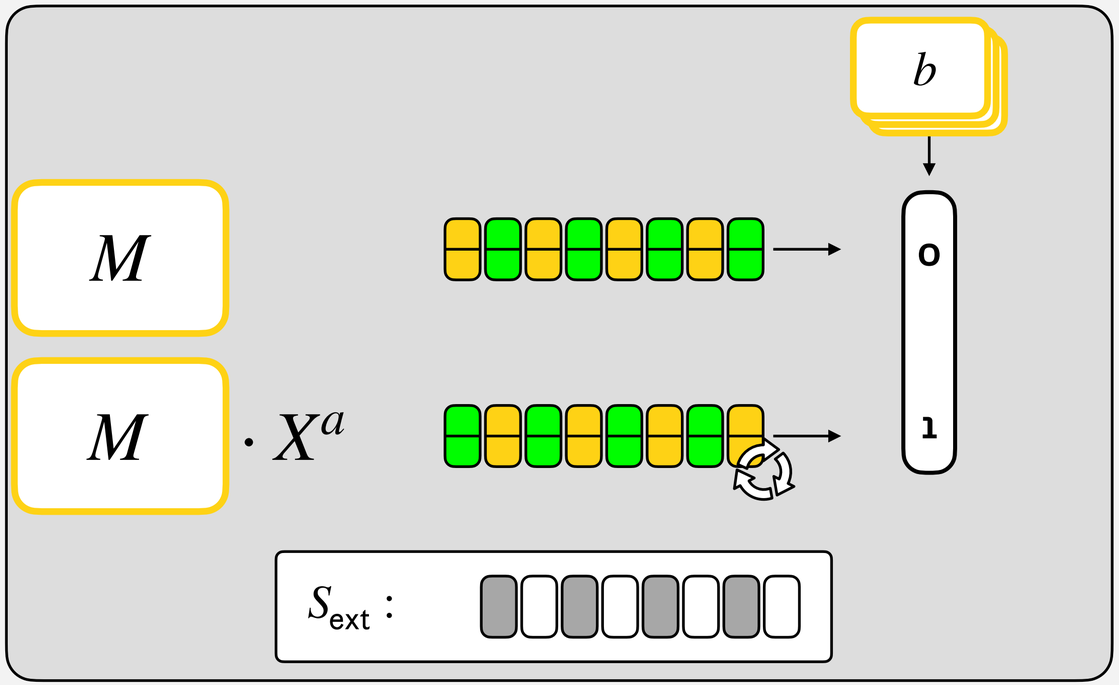
If we rotate by an **odd** a, **index changes**

A rotation of the extended RLWE ciphertext corresponds to **changing the index** of the small RLWE ciphertexts **plus an inner rotation.**

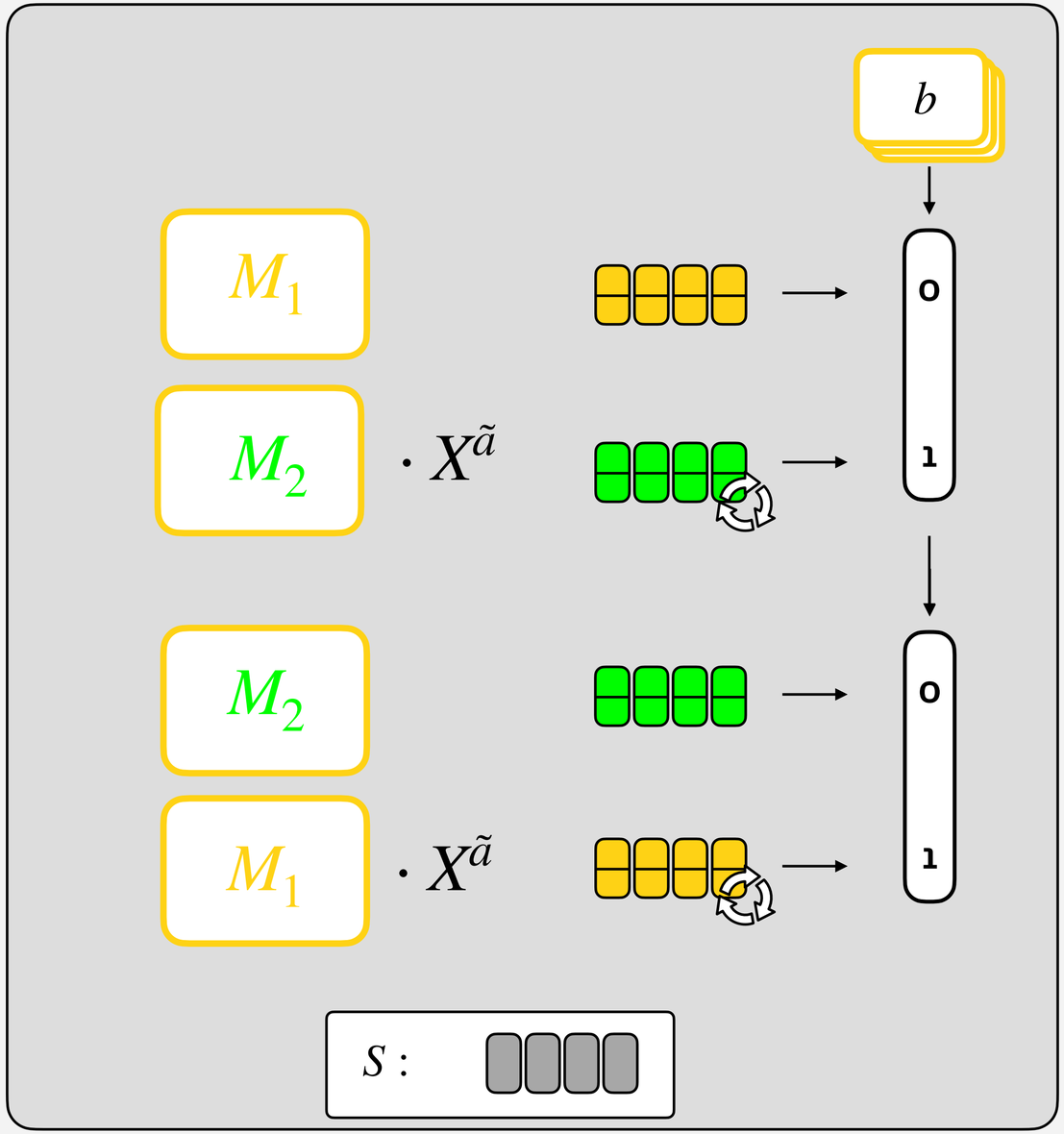
[LY23]Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography,

Context [LY23]

$$\mathscr{R}_{q,N} = Z_q[X]/X^N + 1$$

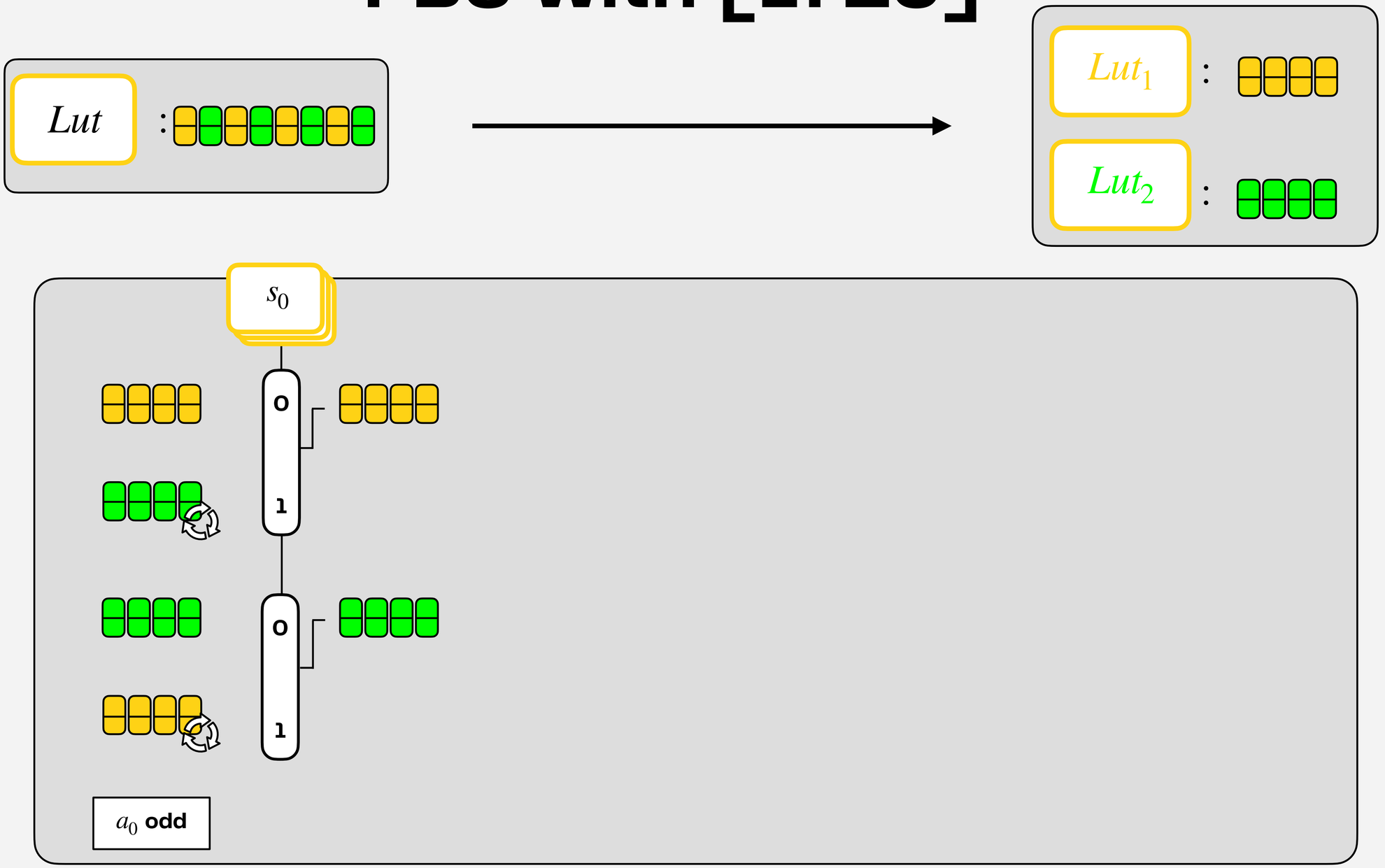


CMuxes in $\mathscr{R}^{k+1}_{q,2^\eta N}$ (with an extended secret key) are **equivalent** to 2^η CMuxes in $\mathscr{R}^{k+1}_{q,N}$.

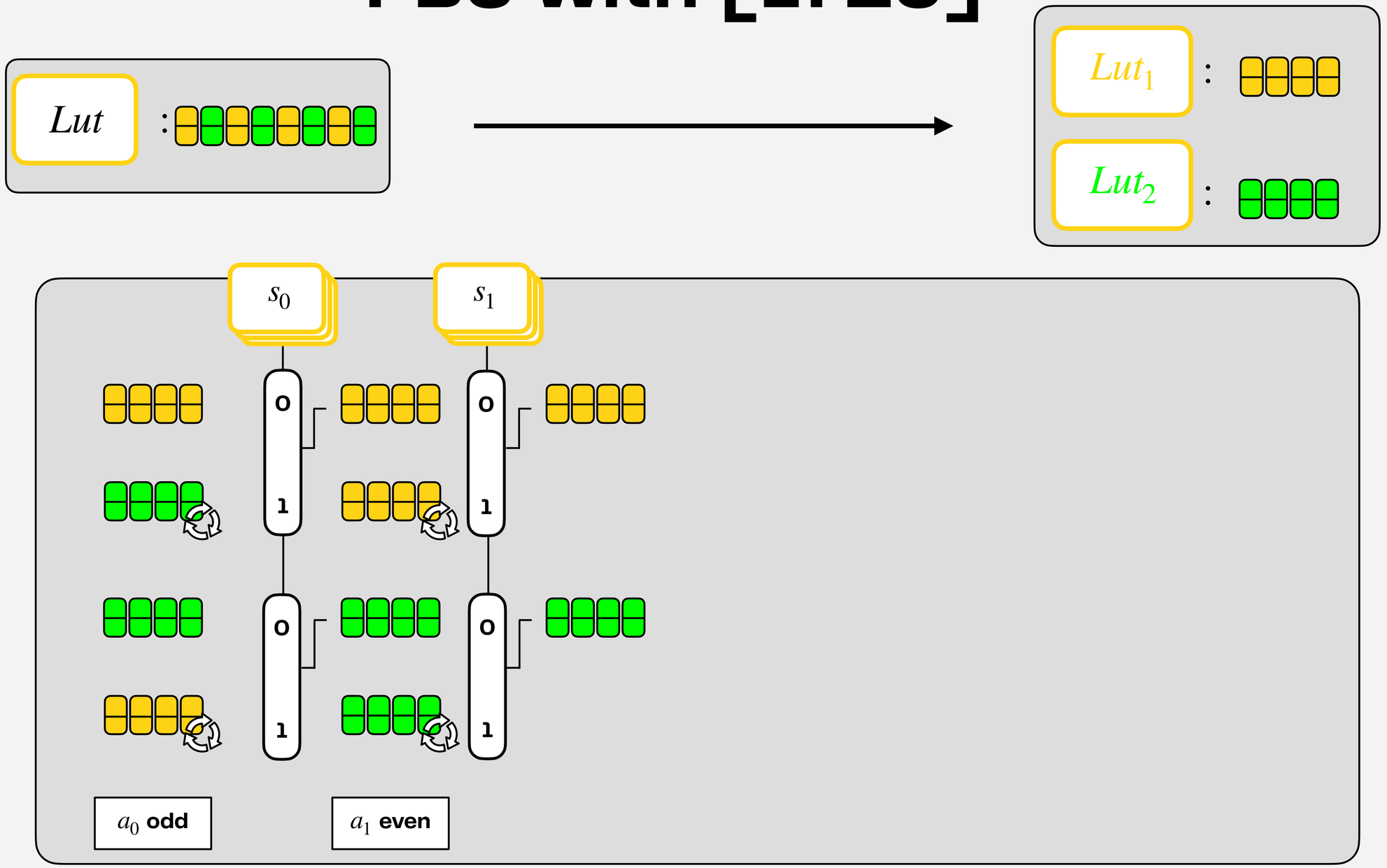


PBS with [LY23]

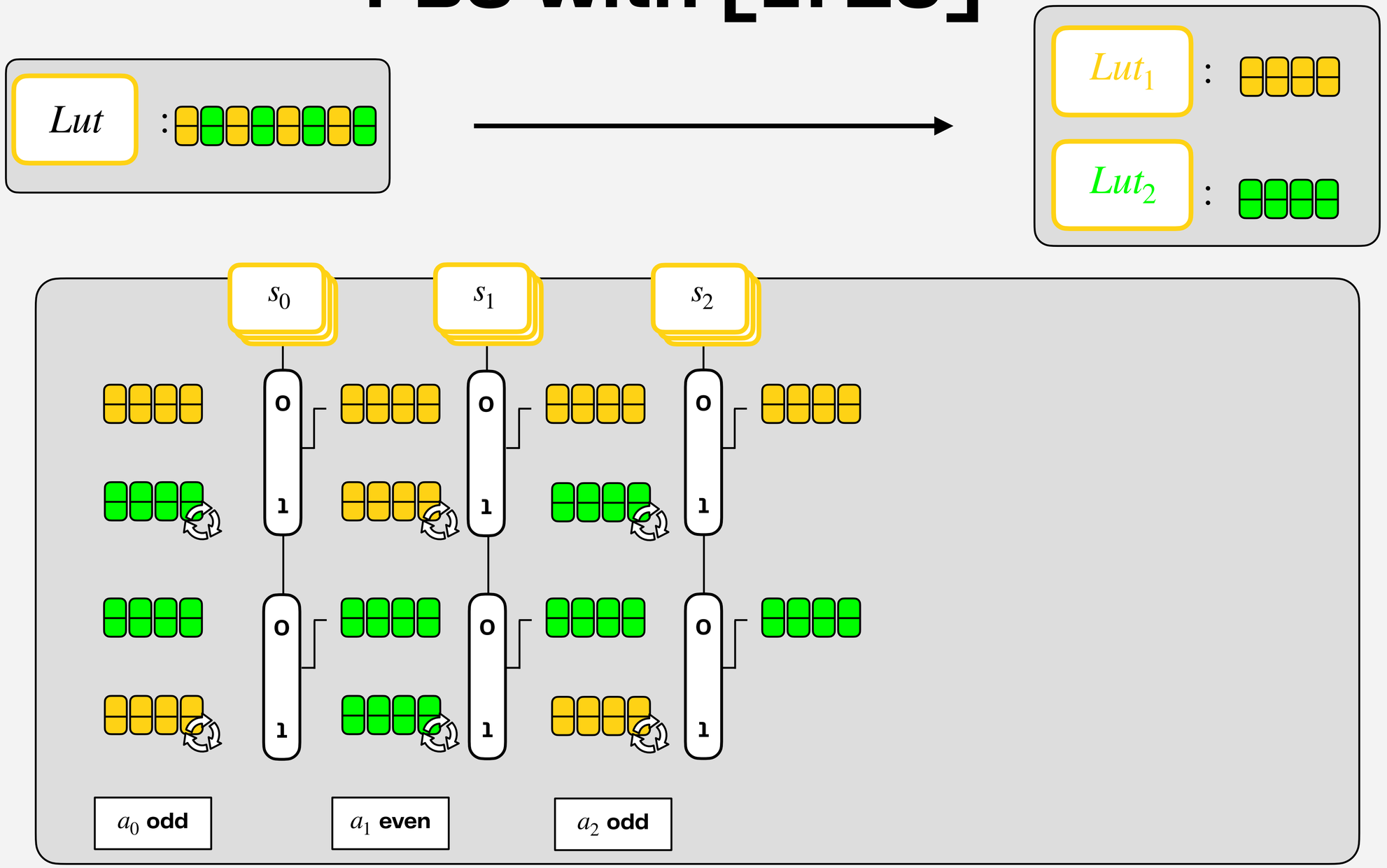
PBS with [LY23]



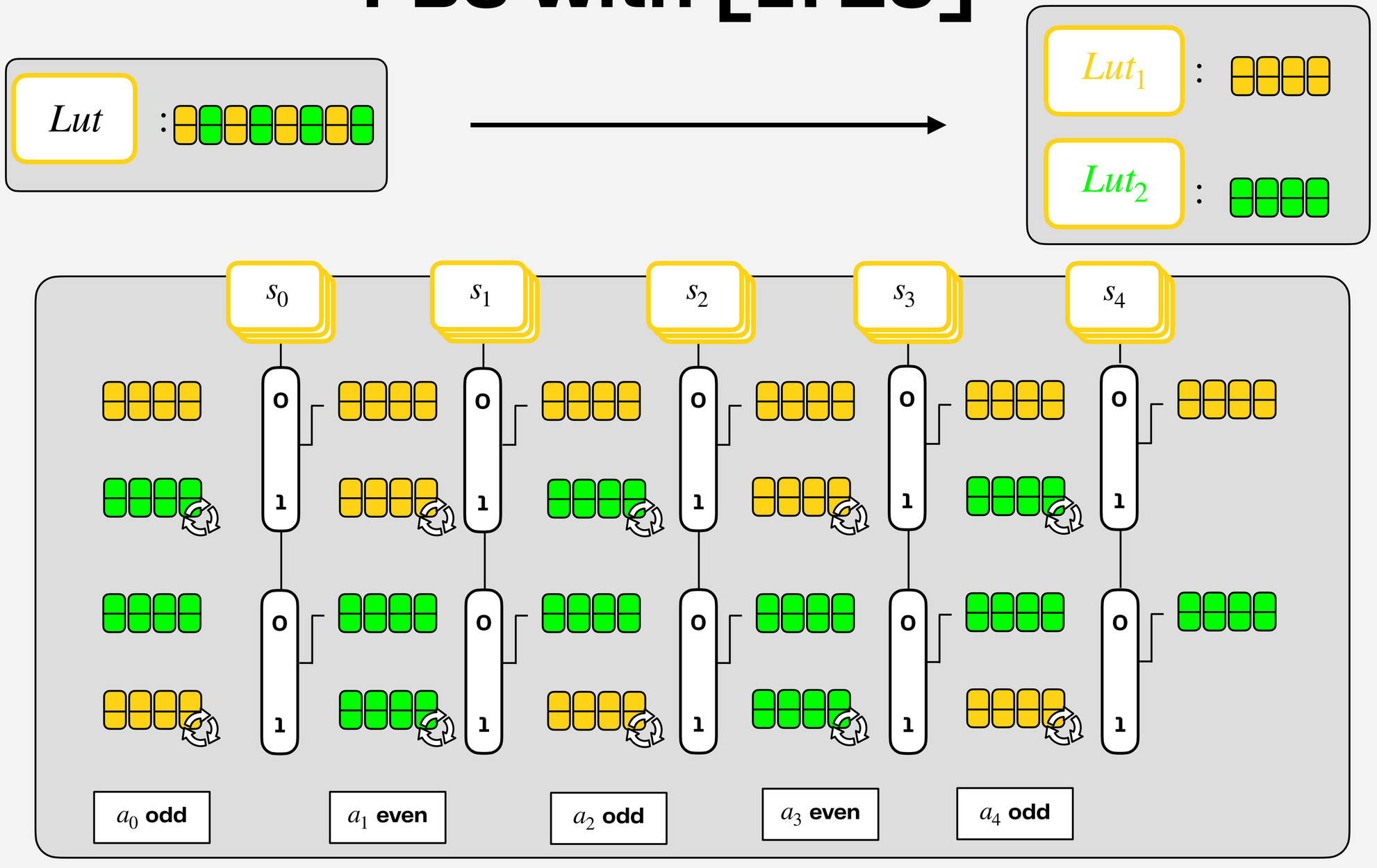
PBS with [LY23]



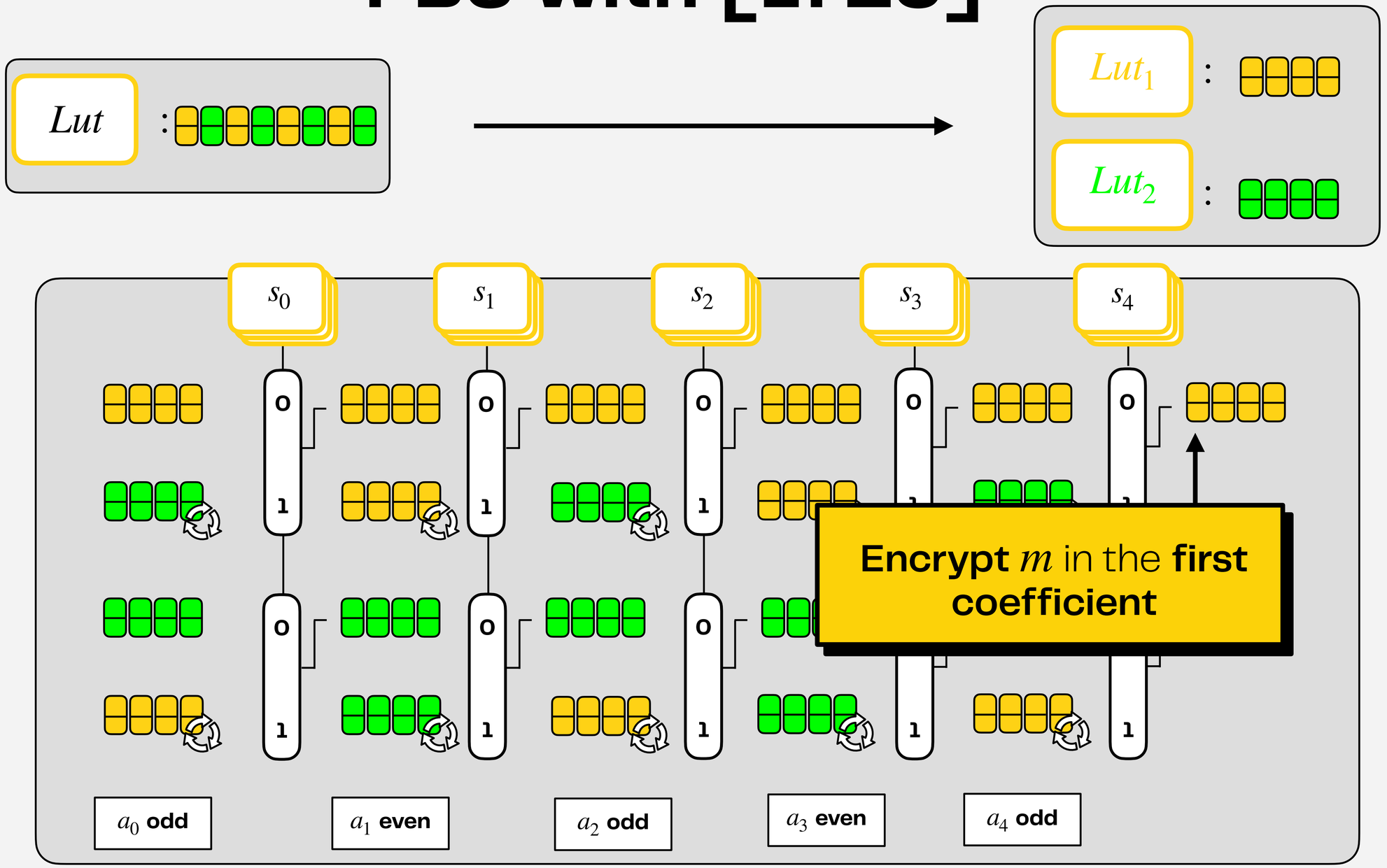
PBS with [LY23]



PBS with [LY23]

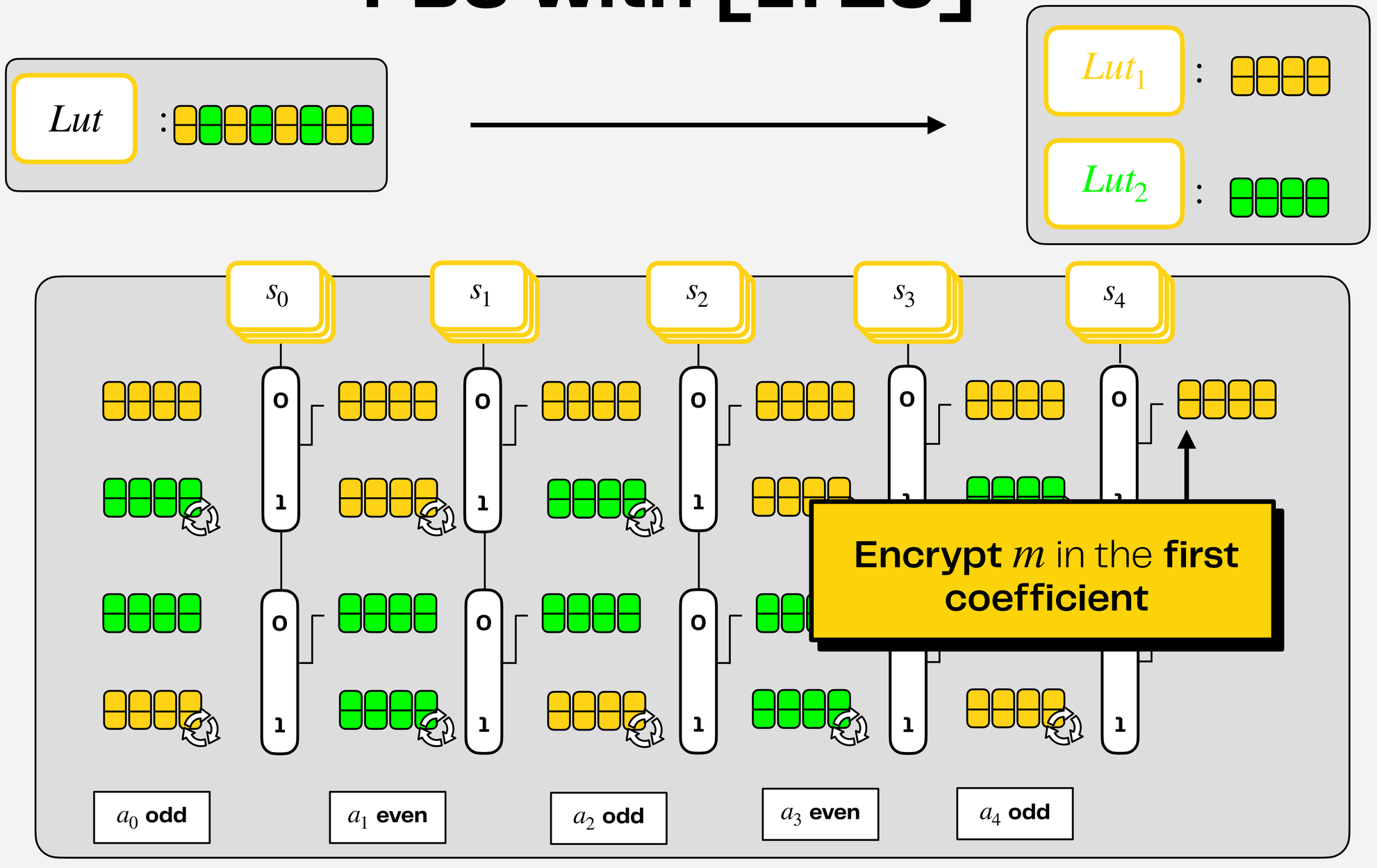


PBS with [LY23]

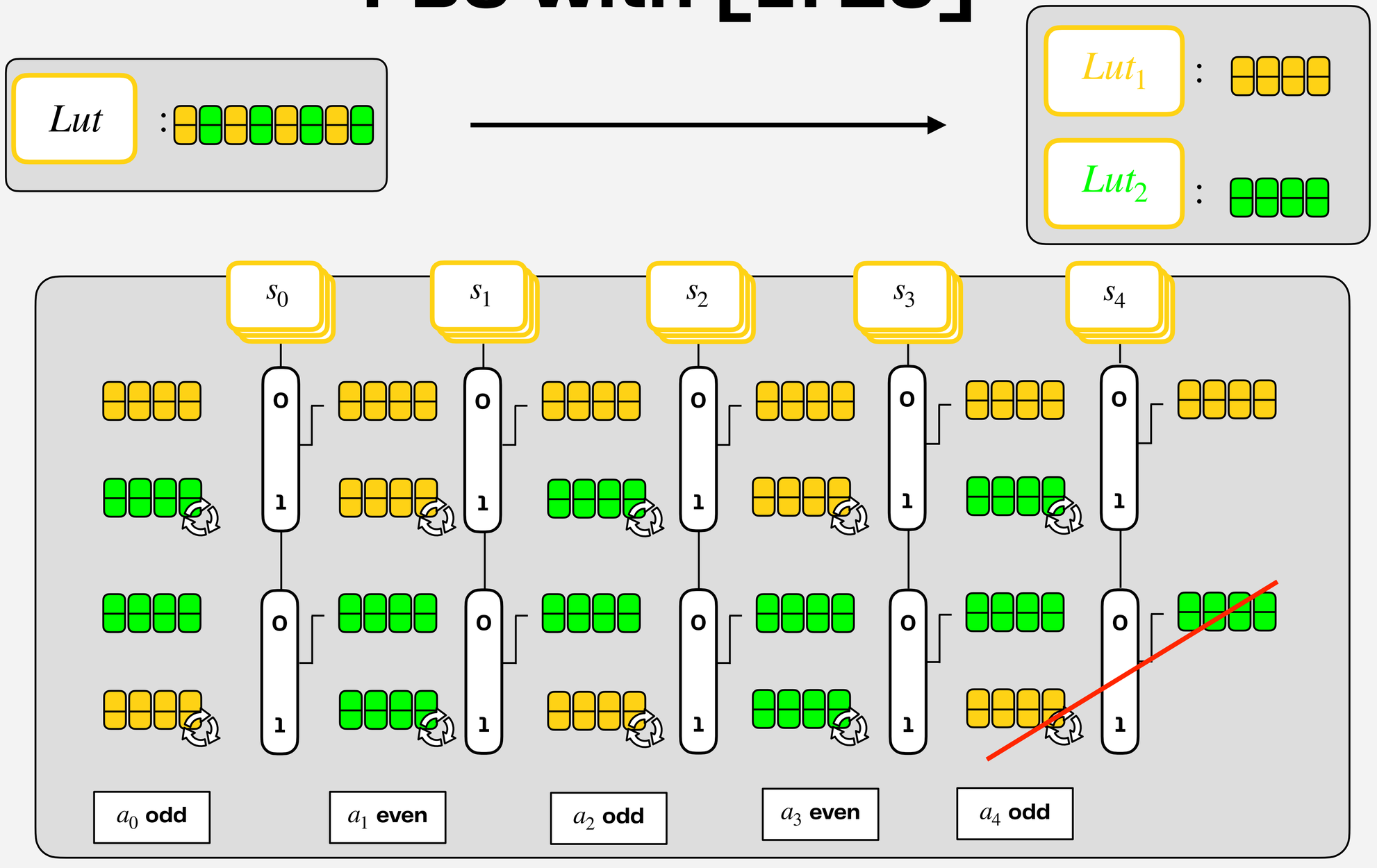


Improving Bootstrapping With the Sorted Technique

PBS with [LY23]

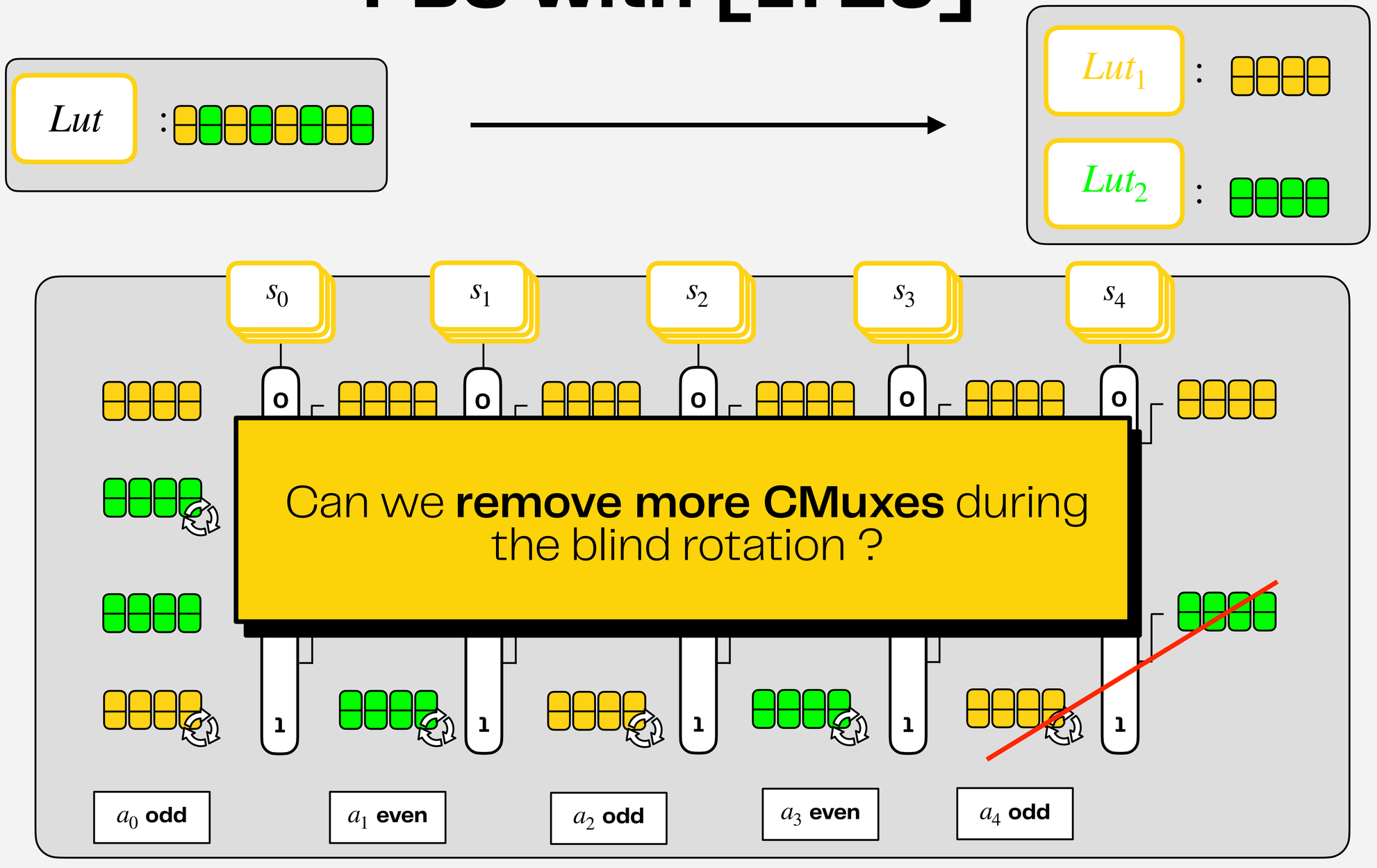


PBS with [LY23]



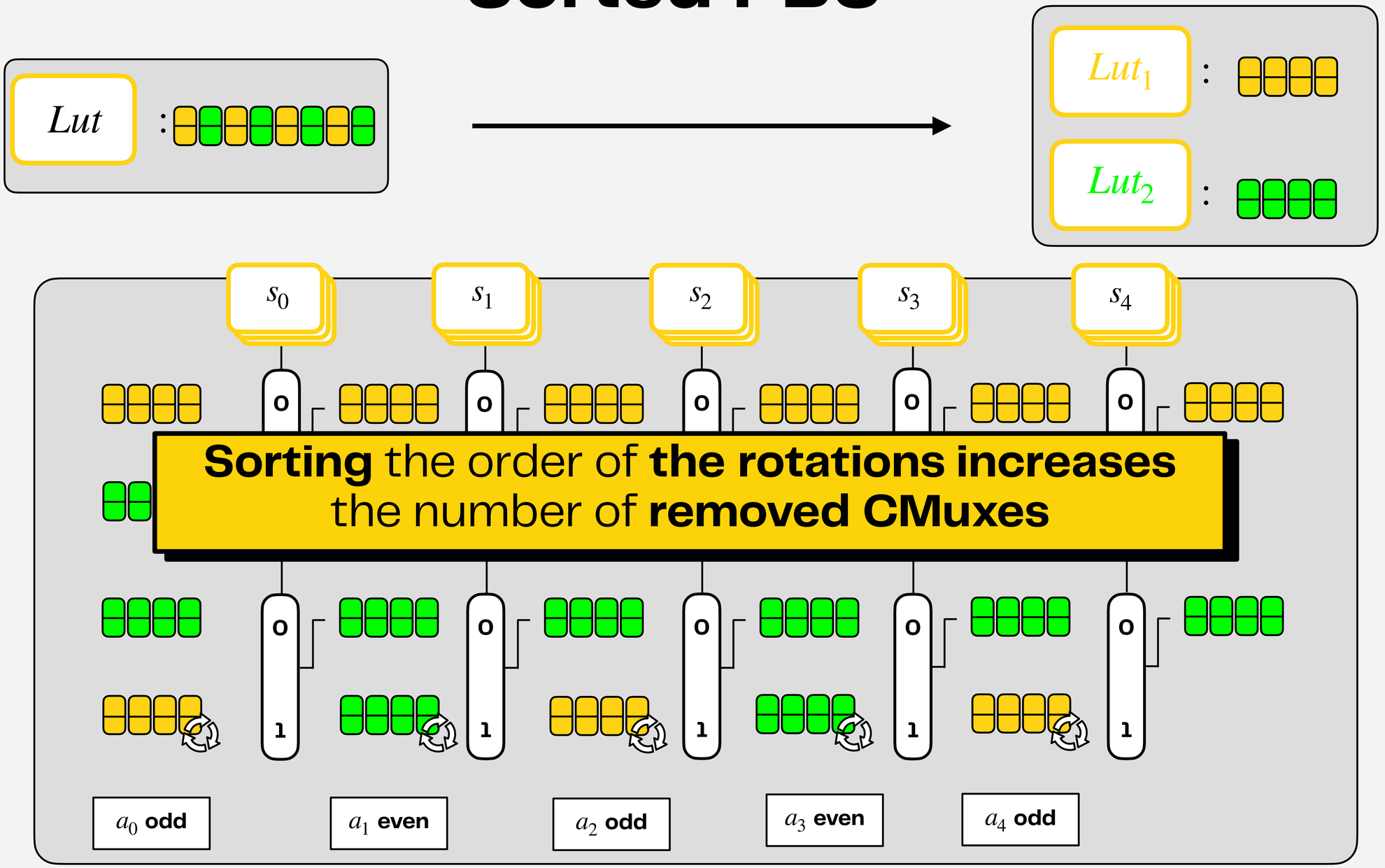
21

PBS with [LY23]

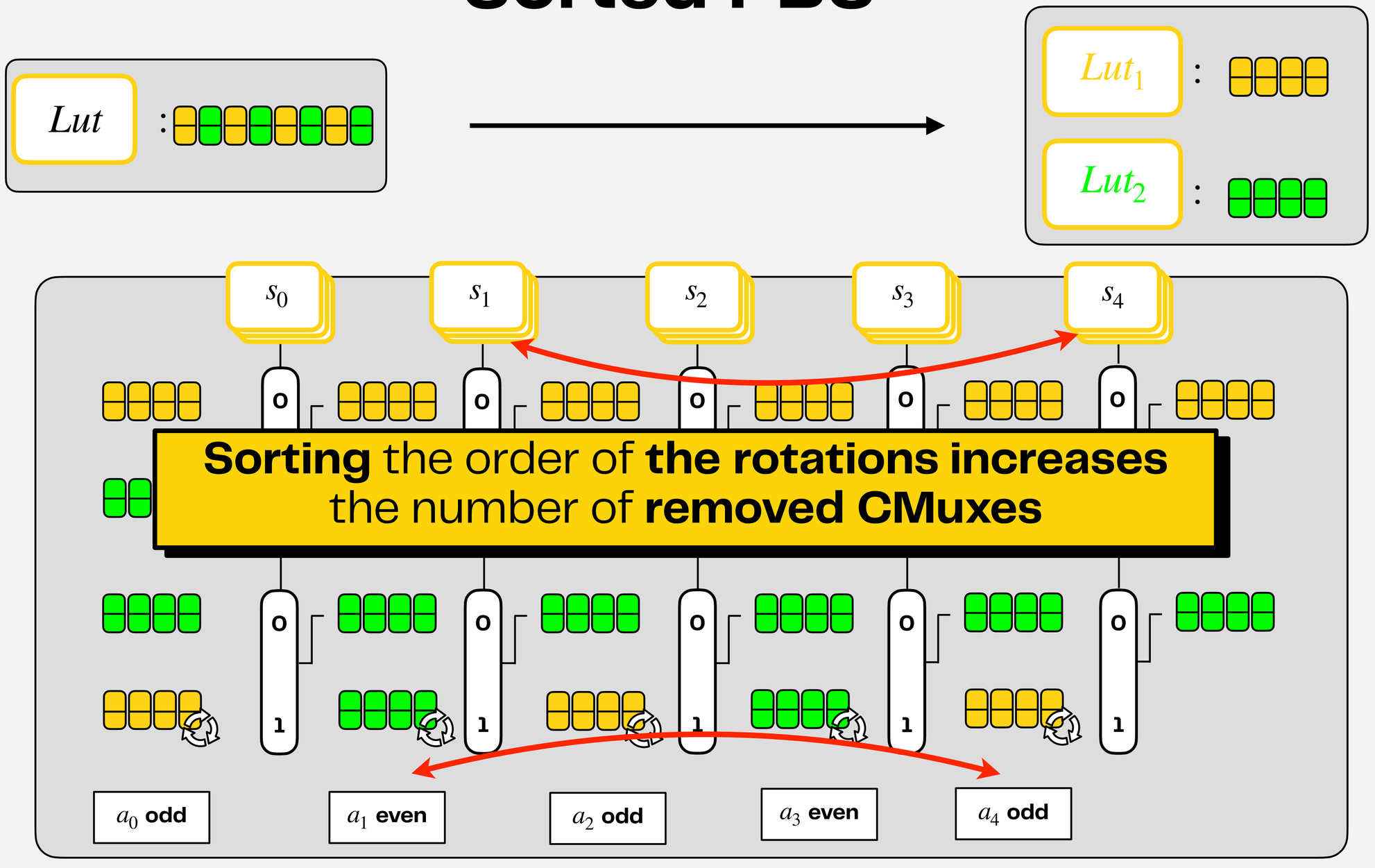


21

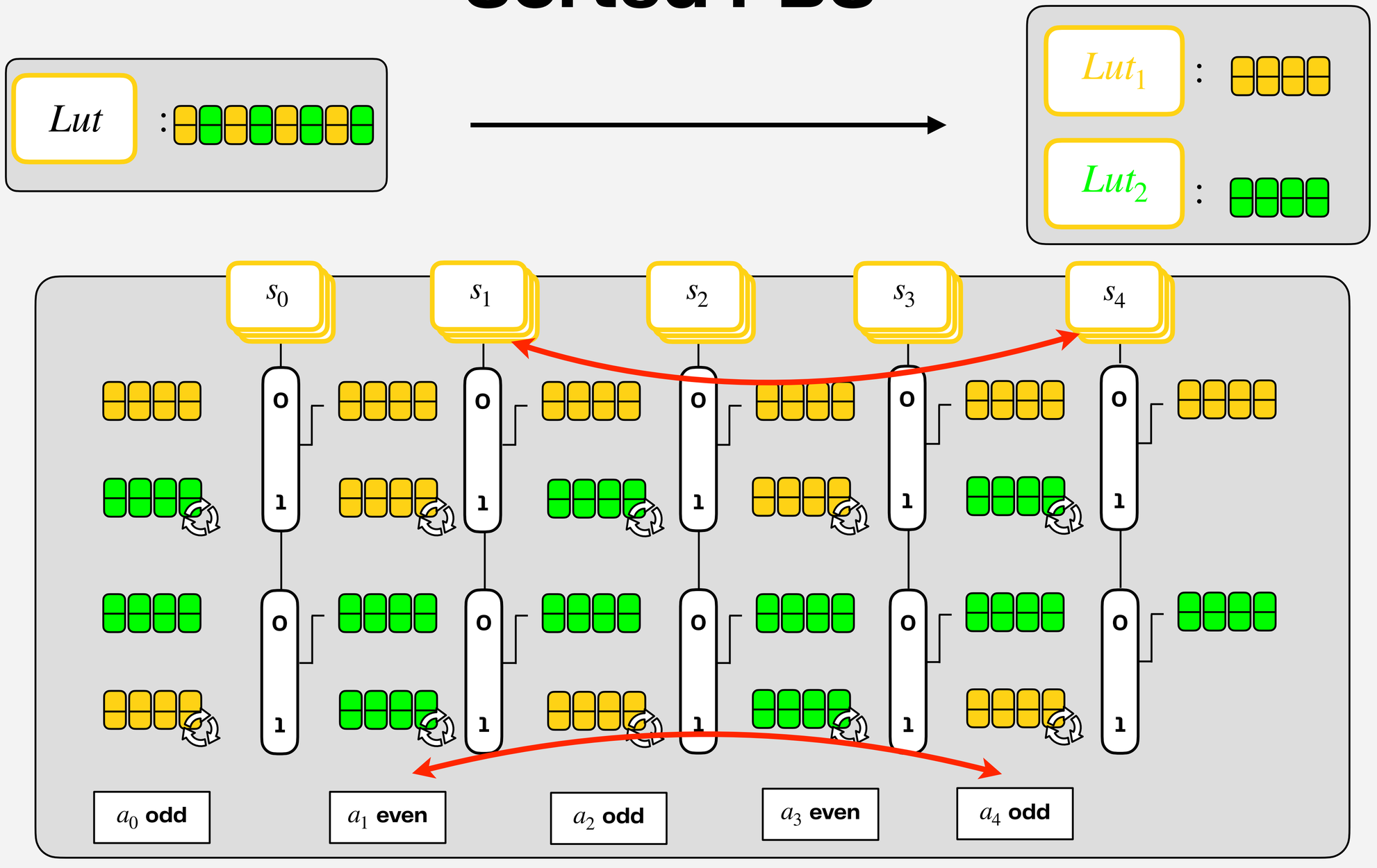
Sorted PBS


Lut
Lut_1
Lut_2
s_0
s_1
s_2
s_3
s_4
0
1
Sorting the order of the rotations increases the number of removed CMuxes
a_0 odd
a_1 even
a_2 odd
a_3 even
a_4 odd

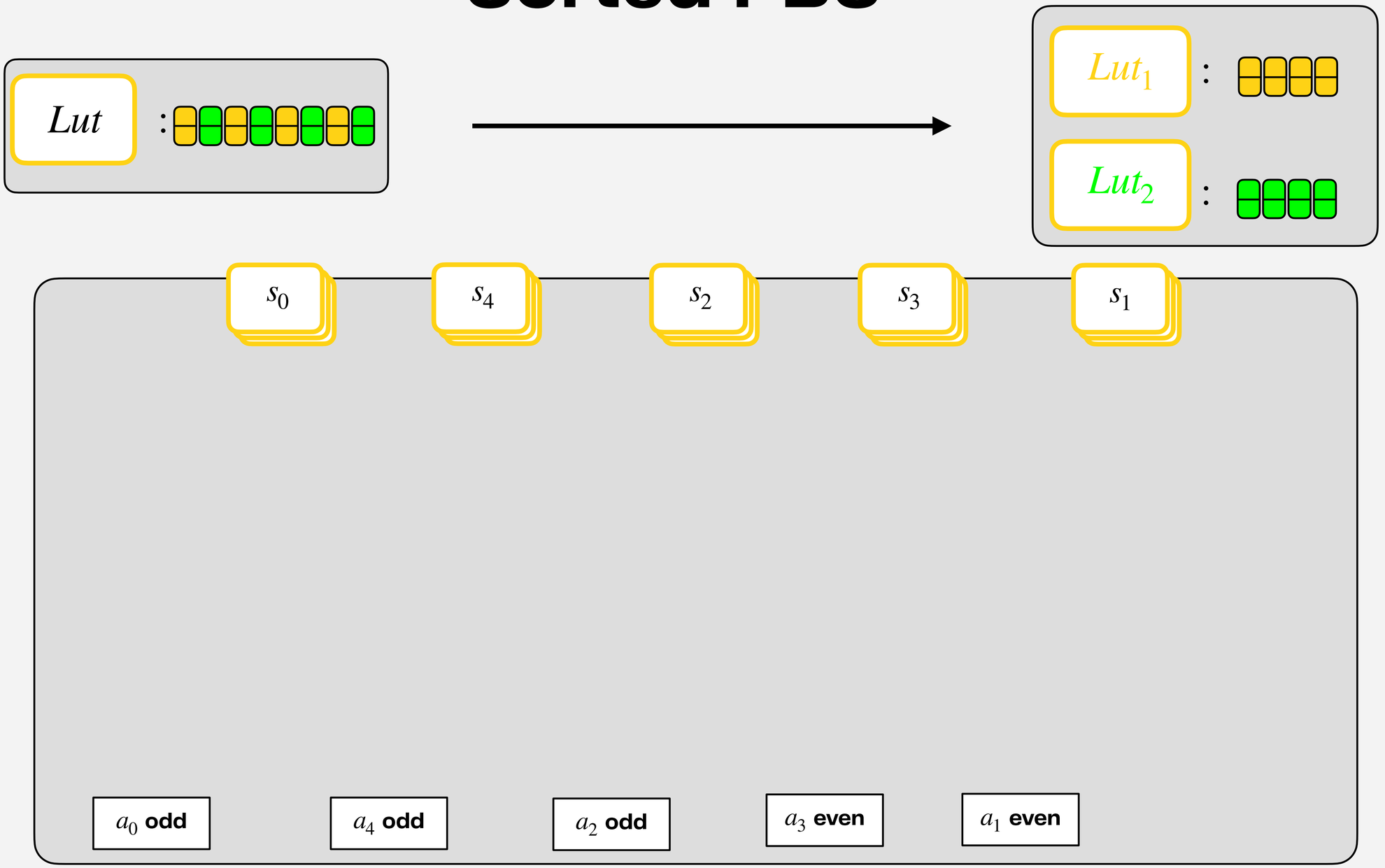
Sorted PBS


Lut
Lut_1
Lut_2
s_0
s_1
s_2
s_3
s_4
0
1
Sorting the order of the rotations increases the number of removed CMuxes
a_0 odd
a_1 even
a_2 odd
a_3 even
a_4 odd

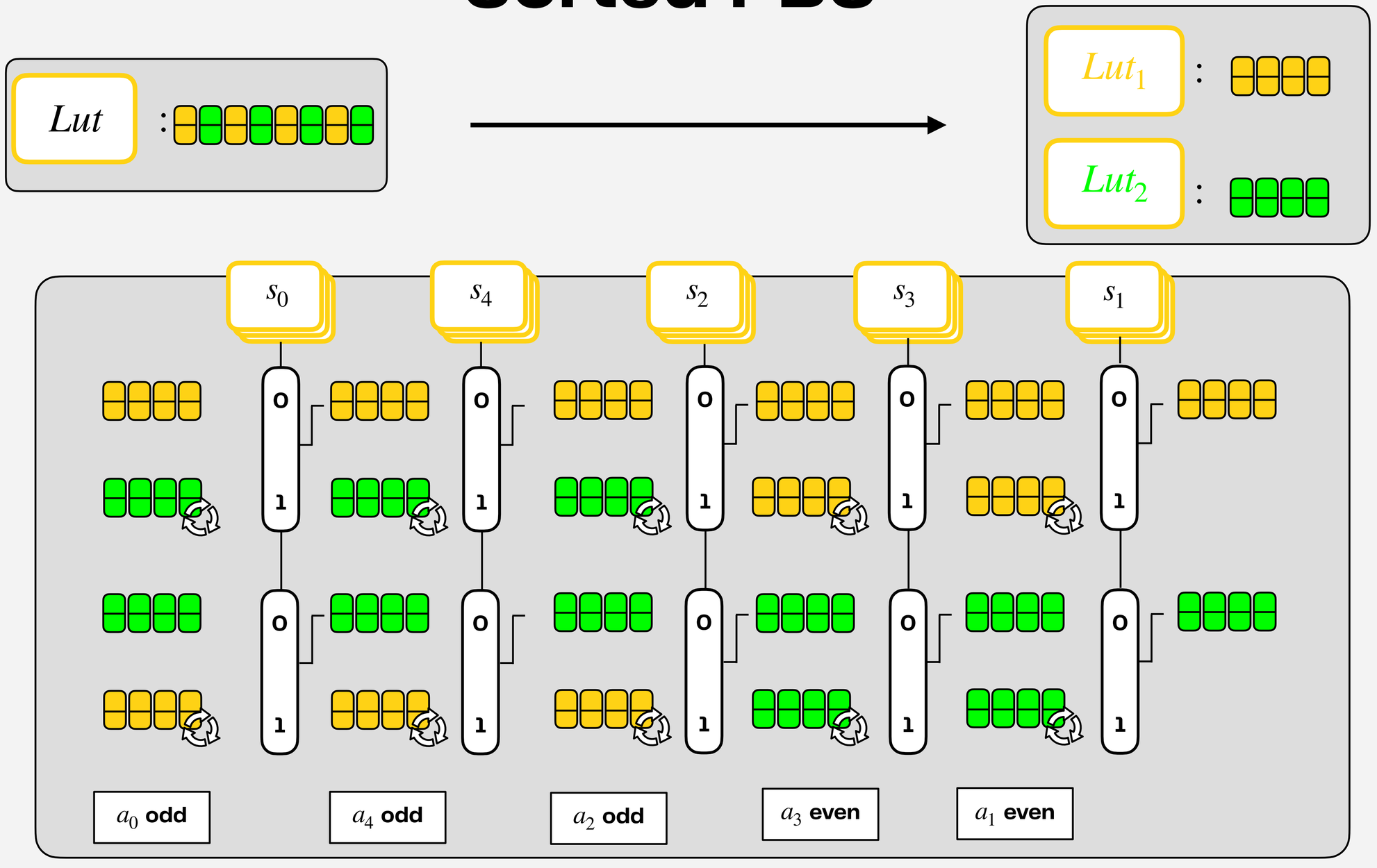
Sorted PBS


Lut
Lut1
Lut2
s0
s1
s2
s3
s4
0
1
a0 odd
a1 even
a2 odd
a3 even
a4 odd

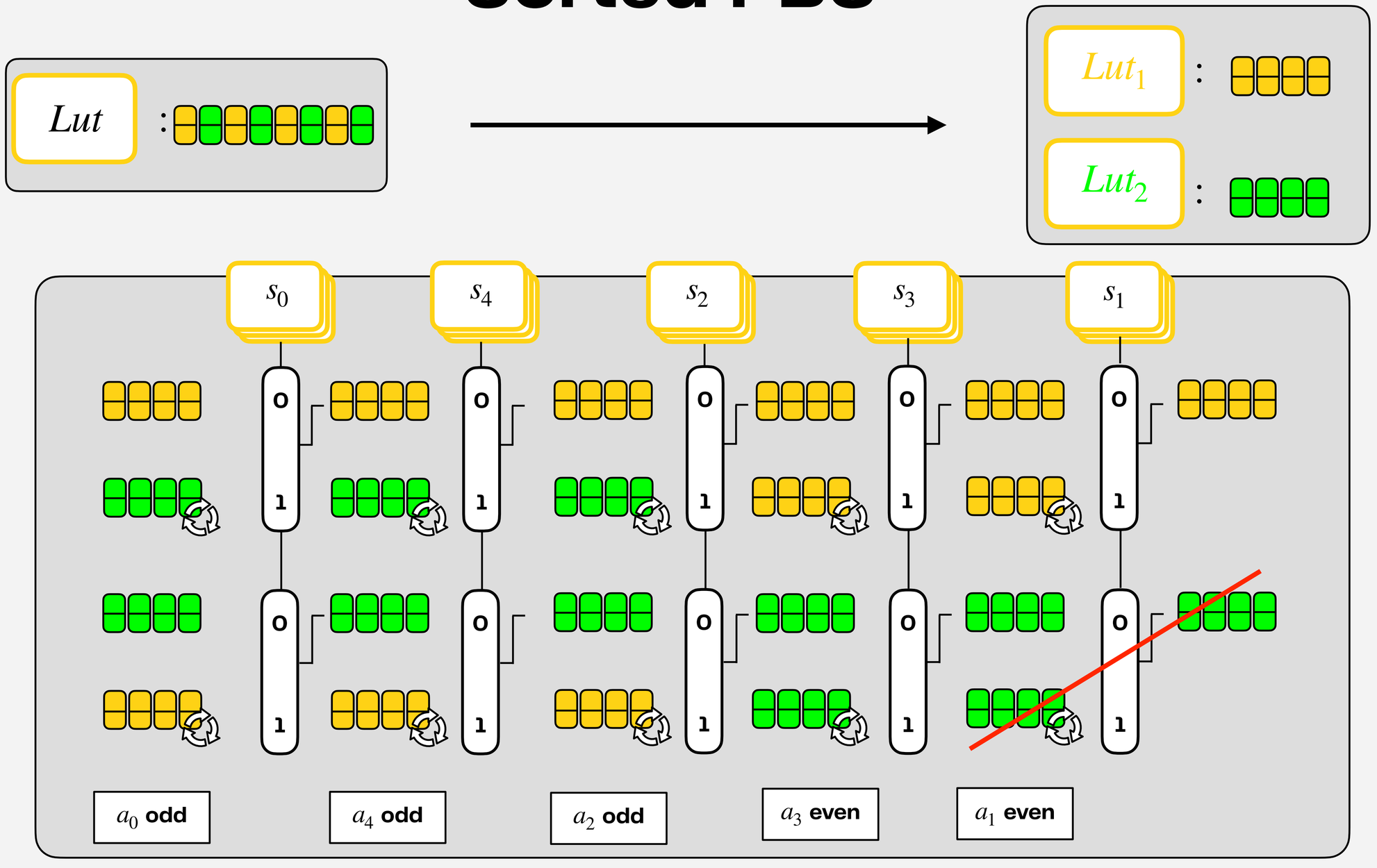
Sorted PBS



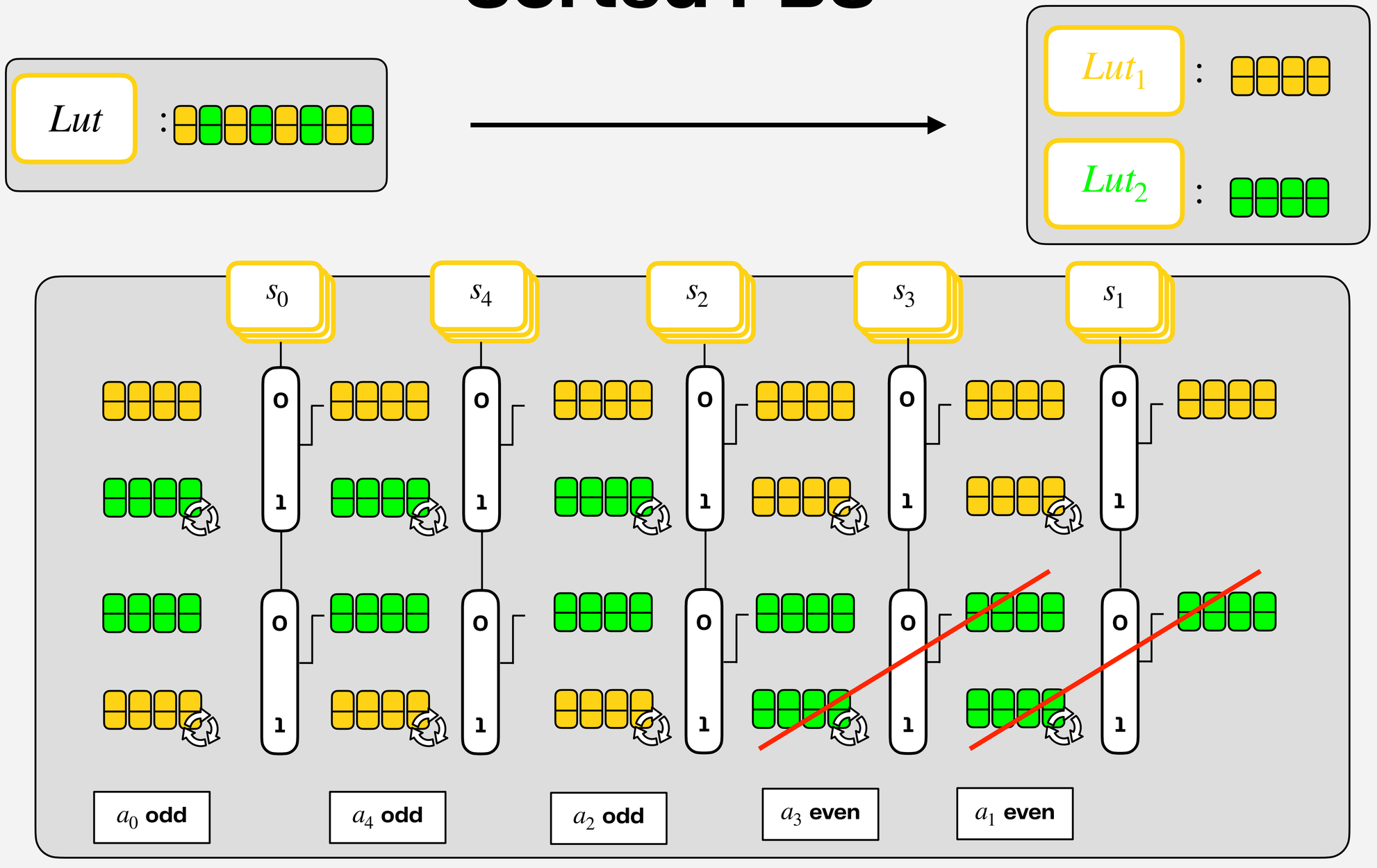
Sorted PBS


Lut
Lut1
Lut2
s0
s4
s2
s3
s1
0
1
a0 odd
a4 odd
a2 odd
a3 even
a1 even

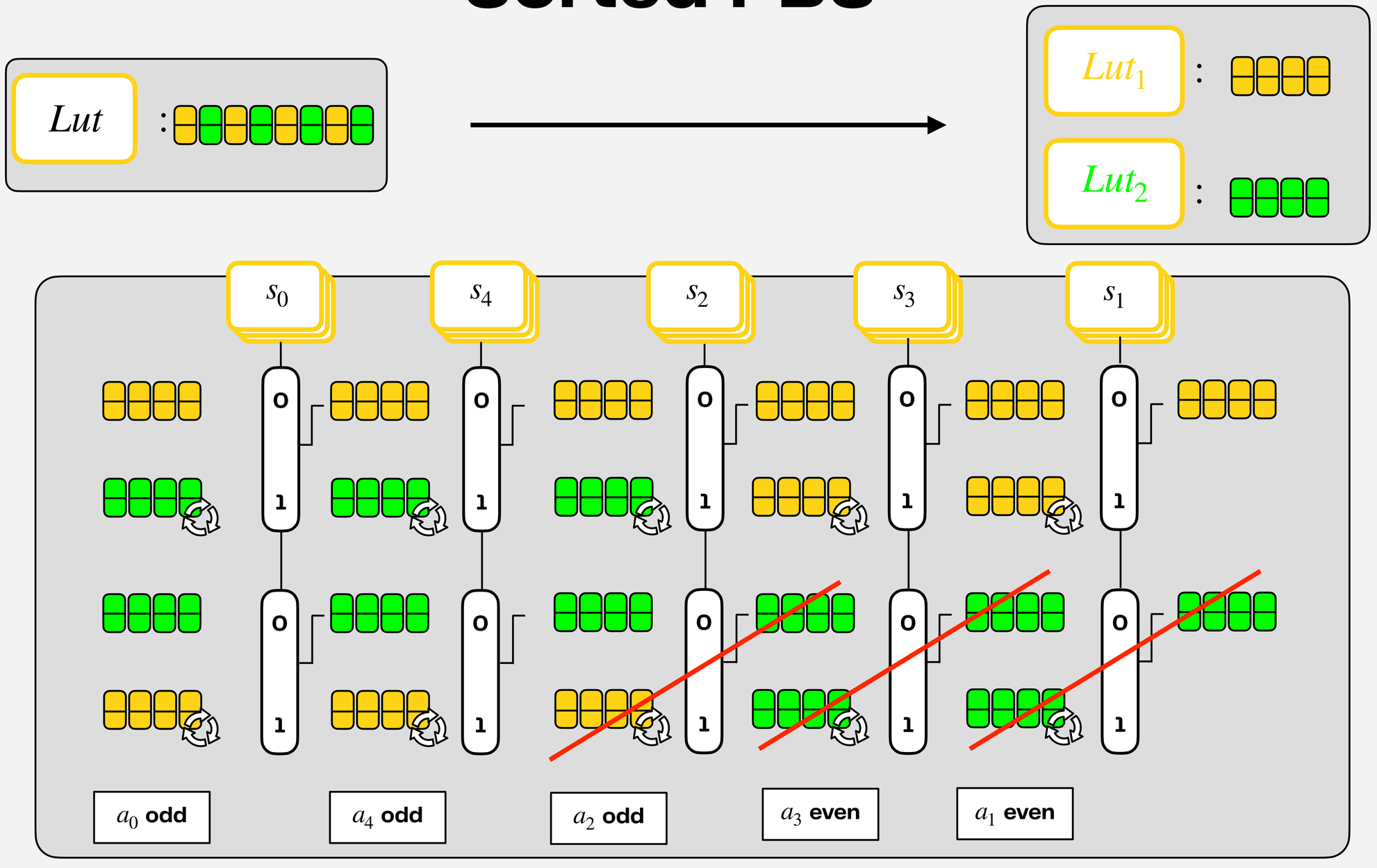
Sorted PBS


Lut
Lut_1
Lut_2
s_0
s_4
s_2
s_3
s_1
0
1
a_0 odd
a_4 odd
a_2 odd
a_3 even
a_1 even

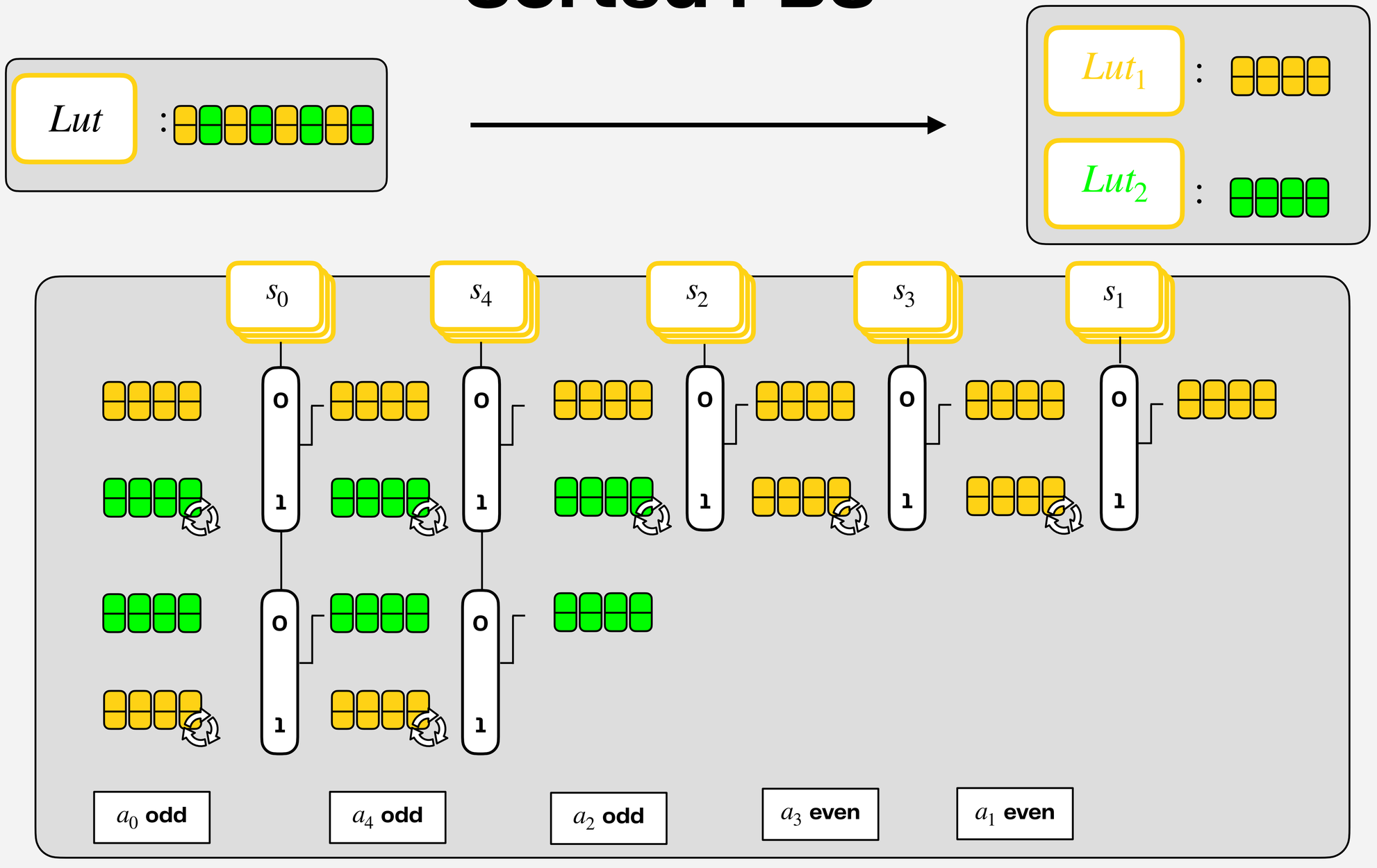
Sorted PBS


Lut
Lut_1
Lut_2
s_0
s_4
s_2
s_3
s_1
0
1
a_0 odd
a_4 odd
a_2 odd
a_3 even
a_1 even

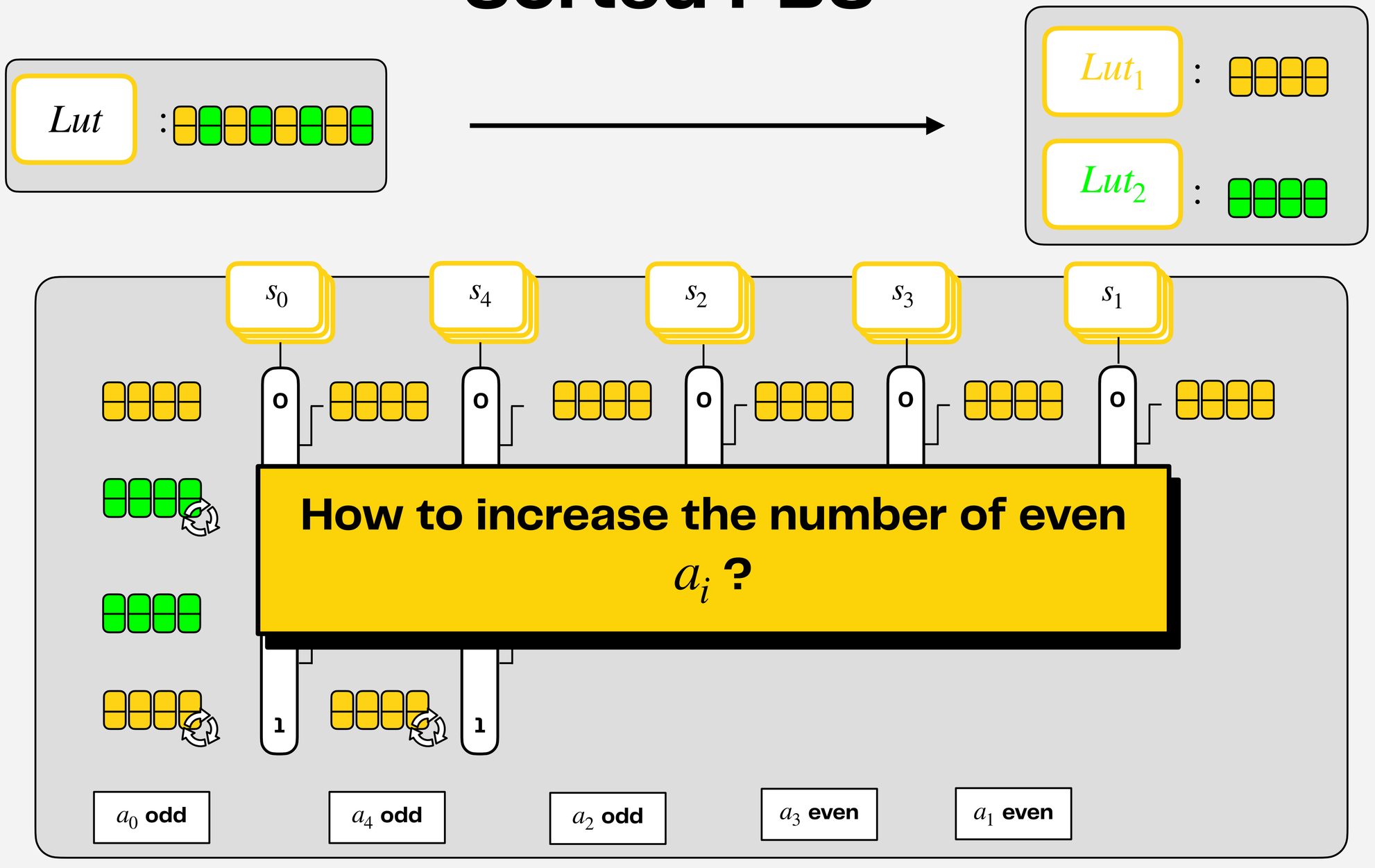
Sorted PBS


Lut
Lut_1
Lut_2
s_0
s_4
s_2
s_3
s_1
0
1
a_0 odd
a_4 odd
a_2 odd
a_3 even
a_1 even

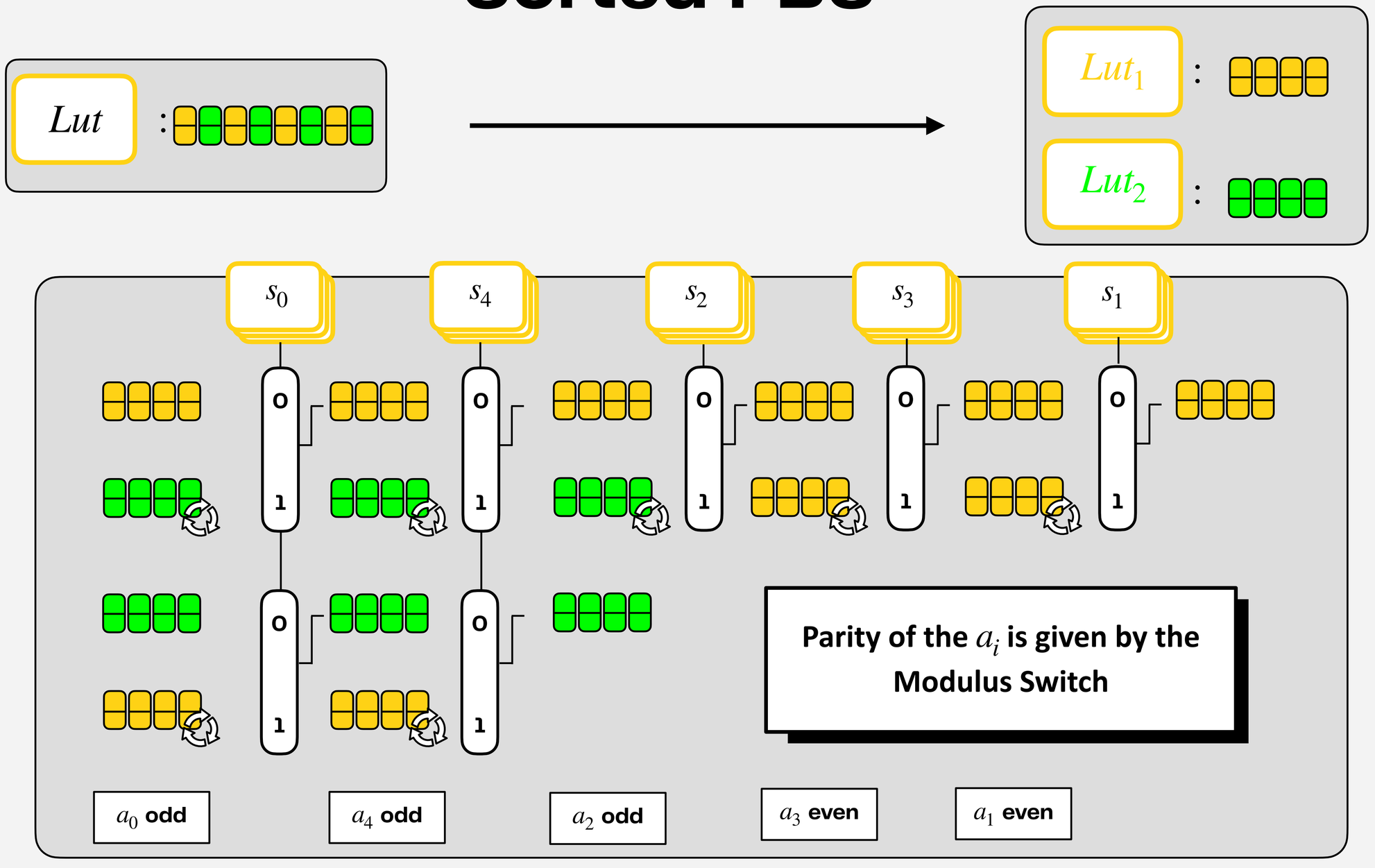
Sorted PBS


Lut
Lut1
Lut2
s0
s4
s2
s3
s1
0
1
a0 odd
a4 odd
a2 odd
a3 even
a1 even

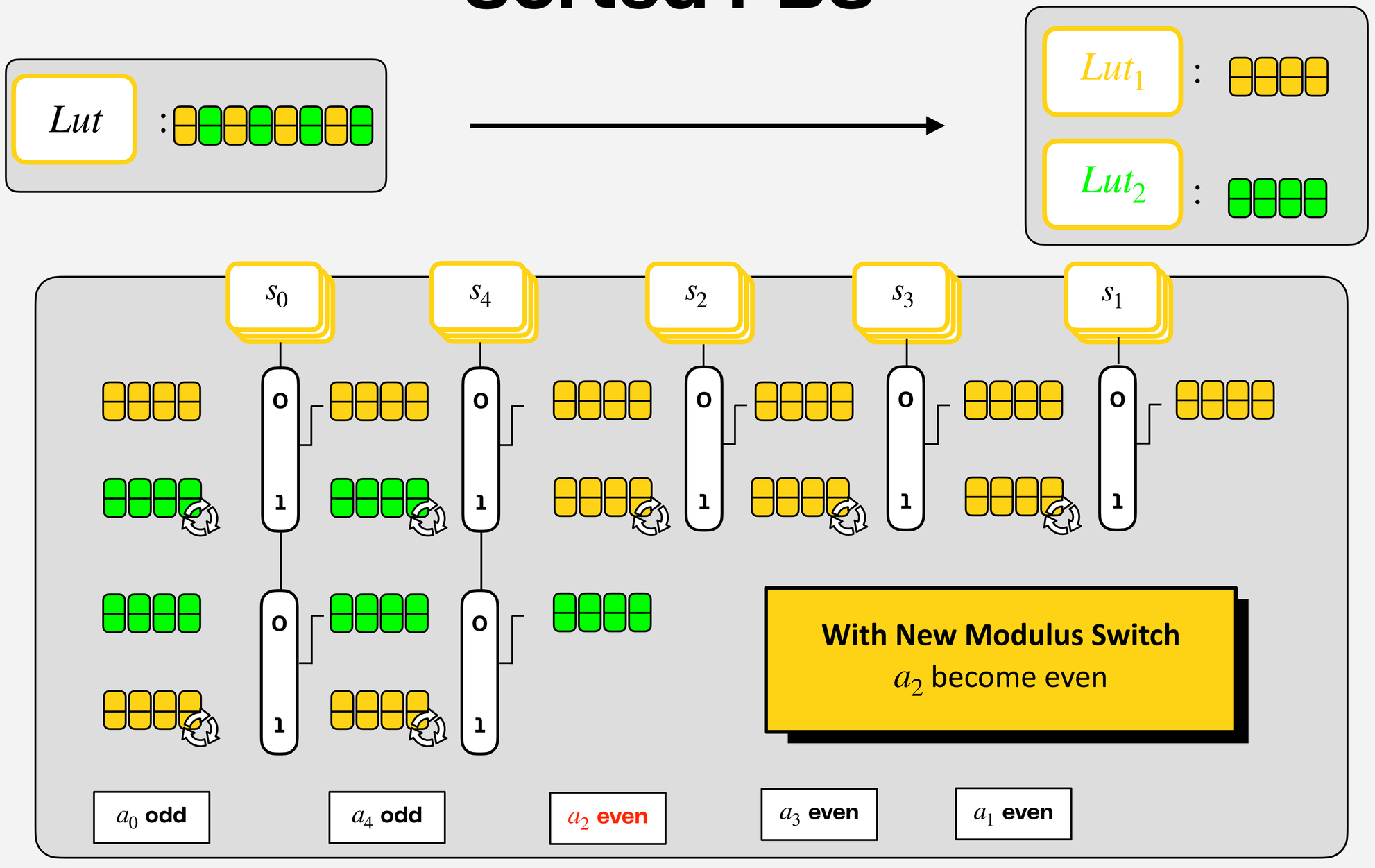
Sorted PBS


Lut
Lut_1
Lut_2
s_0
s_4
s_2
s_3
s_1
0
1
How to increase the number of even a_i ?
a_0 odd
a_4 odd
a_2 odd
a_3 even
a_1 even

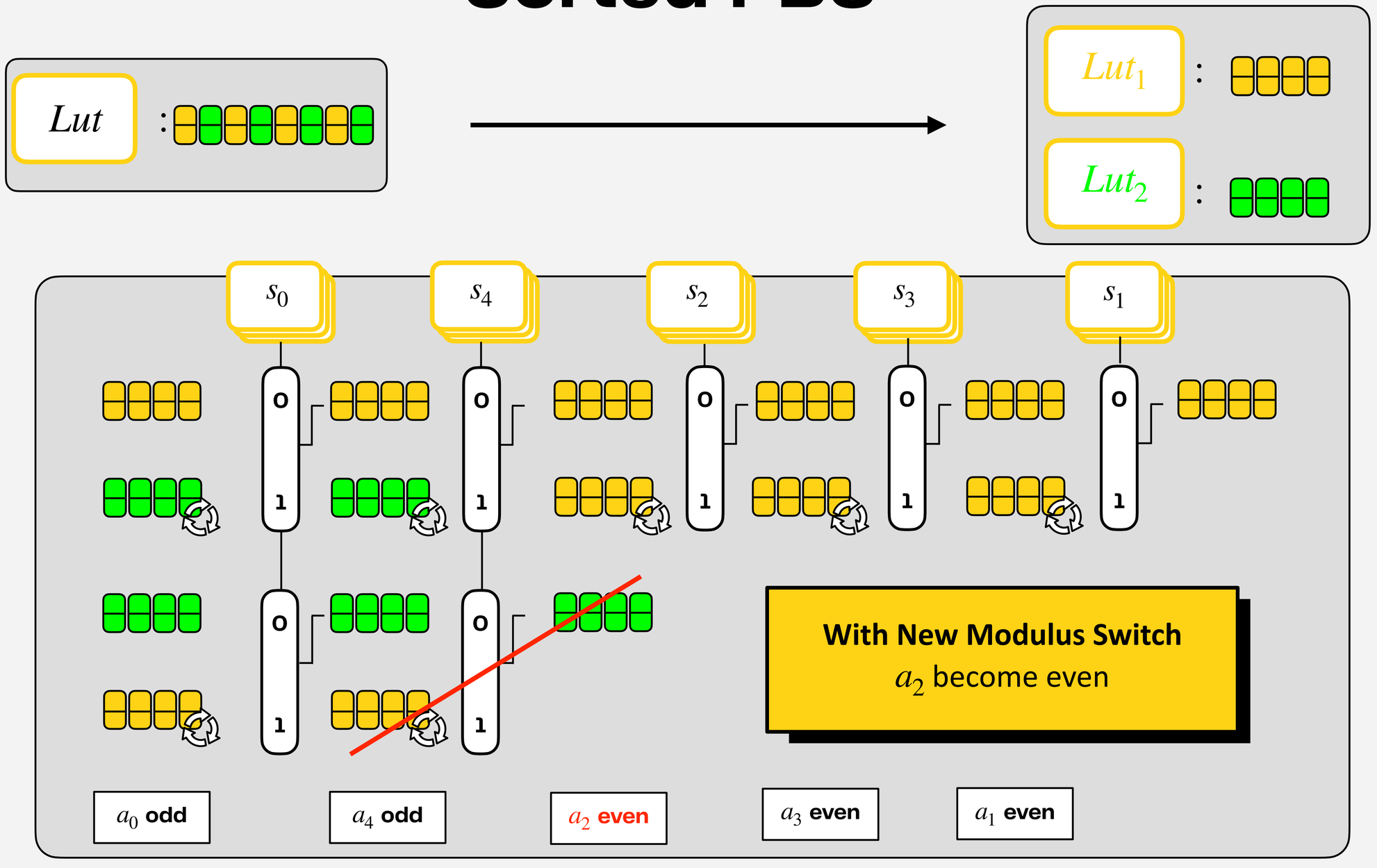
Sorted PBS


Lut
Lut1
Lut2
s0
s4
s2
s3
s1
0
1
Parity of the a_i is given by the Modulus Switch
a_0 odd
a_4 odd
a_2 odd
a_3 even
a_1 even

Sorted PBS


Lut
Lut1
Lut2
s0
s4
s2
s3
s1
0
1
With New Modulus Switch
a2 become even
a0 odd
a4 odd
a2 even
a3 even
a1 even

Sorted PBS


Lut
Lut1
Lut2
s0
s4
s2
s3
s1
0
1
With New Modulus Switch
a2 become even
a0 odd
a4 odd
a2 even
a3 even
a1 even

Benchmarks

Input precision	[LY23]	Sorted Bootstrapping	Gain
5 bits	46.7 ms	29.9 ms	1.56 X
6 bits	136 ms	67.4 ms	2.02 X
7 bits	226 ms	128 ms	2.07 X
8 bits	542 ms	256 ms	2.11 X
9 bits	**1118 ms**	**521 ms**	**2.14 X**

with $P_{fail} = 2^{-80}$
using **TFHE-rs**

Speed-ups between
1.56 and **2.14**

AWS hpc7a.96xlarge
AMD EPYC 9R14 CPU
@ 2.60GHz

Conclusion

New sorted Bootstrapping

Improve previous work of **[LY23]**

Other improvements

New **modulus switch**

More **parallelism**

Speed-ups

Between **1.56** and **2.14** compared to **[LY23]**

Between **3.3** and **6.7** compared to the **PBS**

[LY23] Kang Hoon Lee and Ji Won Yoon. Discretization error reduction for high precision torus fully homomorphic encryption. In IACR International Conference on Public-Key Cryptography

Thank you

Full paper at ePrint 2025/2214

ZAMA

Contact & Links

Loris.bergerat@zama.ai

zama.ai

github.com/zama-ai

zama.ai/community