Discretization Error Reduction for High Precision Torus Fully Homomorphic Encryption

Kang Hoon Lee, Ji Won Yoon
PKC 2023

Korea University, School of CyberSecurity

Table of contents

1. Introduction

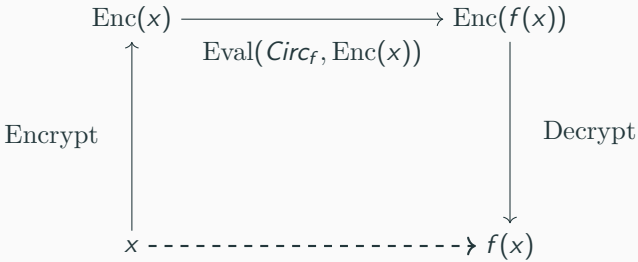
2. Extended BootStrapping for TFHE

3. Experimental Results

Introduction

Fully Homomorphic Encryption

- Encryption scheme that allows to operate with encrypted data



- Usually comes with a *recryption* procedure, called **bootstrapping**.

Notations

- $n \in \mathbb{Z}$: TLWE dimension
- N : power of 2. (Ring Dimension)
- $\mathbb{T} = \mathbb{R}/\mathbb{Z} = \left[-\frac{1}{2}, \frac{1}{2}\right)$
- $\mathbb{B} = \{0, 1\}$
- $\mathbb{Z}_N[X] = \mathbb{Z}[X]/\langle X^N + 1\rangle$
- $\mathbb{T}_N[X] = \mathbb{T}[X]/\langle X^N + 1\rangle$
- $\mathbf{s} \in \mathbb{B}^n$
- $\mathcal{K} \in \mathbb{B}_N[X]$
- ν : Extension Parameter

TFHE

FHE scheme introduced in [CGGI20][1], based on the LWE (Learning With Errors) problem.

Works with three ciphertext spaces:

- TLWE : $\mathrm{TLWE}_{\mathbf{s}}(m) = (\mathbf{a}, b = \mathbf{a} \cdot \mathbf{s} + m + e) \in \mathbb{T}^{n+1}$
- TRLWE : $\mathrm{TRLWE}_{\mathcal{K}}(m(X)) = (a(X), b(X) = a(X)\mathcal{K}(X) + m(X) + e(X)) \in \mathbb{T}_N[X]^2$
- TRGSW :

$$\mathrm{TRGSW}_{\mathcal{K}}(z(X)) = \left[\begin{array}{c} \mathrm{TRLWE}_{\mathcal{K}}(z(X)/B) \\ \vdots \\ \mathrm{TRLWE}_{\mathcal{K}}(z(X)/B^d) \\ \hline \mathrm{TRLWE}_{\mathcal{K}}(z(X)\cdot(-\mathcal{K})/B) \\ \vdots \\ \mathrm{TRLWE}_{\mathcal{K}}(z(X)\cdot(-\mathcal{K})/B^d) \end{array}\right], \; z(X) \in \mathbb{Z}_N[X].$$

[1] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: fast fully homomorphic encryption over the torus. In: *Journal of Cryptology* 33.1 (2020), pp. 34–91

TFHE (continued)

TFHE supports *external product* $\boxdot$, a multiplication between TRGSW and TRLWE ciphertext:

$$\mathrm{TRGSW}_{\mathcal{K}}(z(X)) \boxdot \mathrm{TRLWE}_{\mathcal{K}}(m(X)) = \mathrm{TRLWE}_{\mathcal{K}}(z(X)m(X)).$$

Bootstrapping homomorphically decrypts the TLWE ciphertext by calculating **CMux** circuit for $i \in [\![0, n-1]\!]$:

CMux : Homomorphic Selection

$$\mathbf{CMux}\left(\mathrm{BSK}_i = \mathrm{TRGSW}_{\mathcal{K}}(\mathbf{s}_i), X^{\bar{\mathbf{a}}_i}\mathrm{ACC}, \mathrm{ACC}\right) = \mathrm{TRGSW}_{\mathcal{K}}(\mathbf{s}_i) \boxdot ((X^{\bar{\mathbf{a}}_i} - 1) \cdot \mathrm{ACC}) + \mathrm{ACC}$$

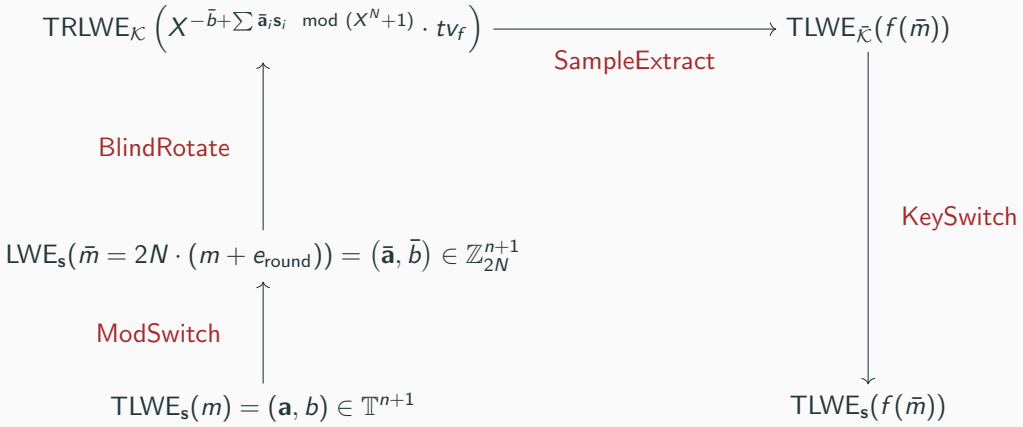
which gives

$$X^{-\bar{b}+\sum \bar{\mathbf{a}}_i\mathbf{s}_i \mod (X^N+1)} \cdot \mathrm{ACC} = \mathrm{TRLWE}_{\mathcal{K}}\left(X^{-\bar{b}+\sum \bar{\mathbf{a}}_i\mathbf{s}_i \mod (X^N+1)} \cdot m(X)\right).$$

Extended BootStrapping for TFHE

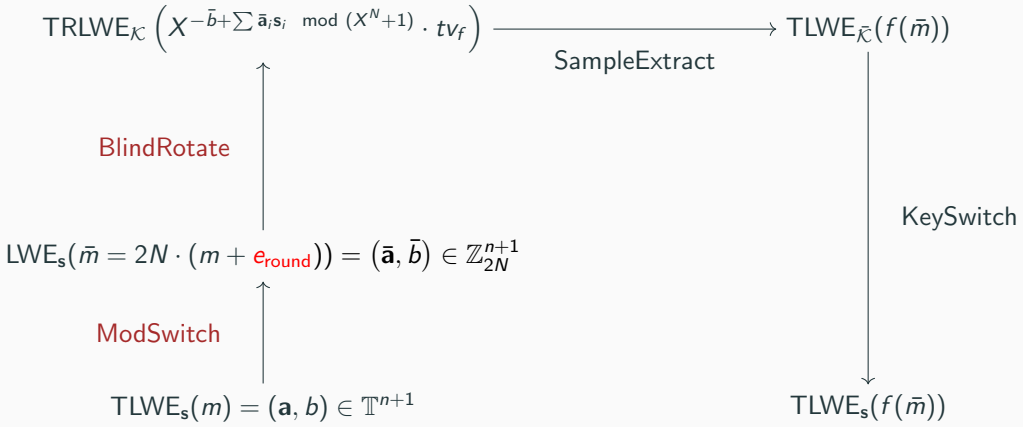
TFHE (Functional) Bootstrapping

Given a function $f : \mathbb{T} \to \mathbb{T}$, the (functional) TFHE bootstrapping is a series of algorithms:

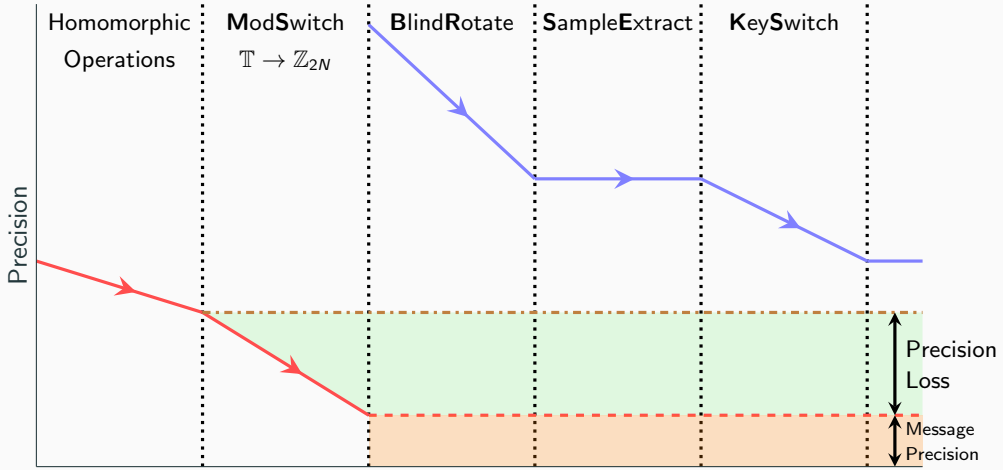


TFHE (Functional) Bootstrapping

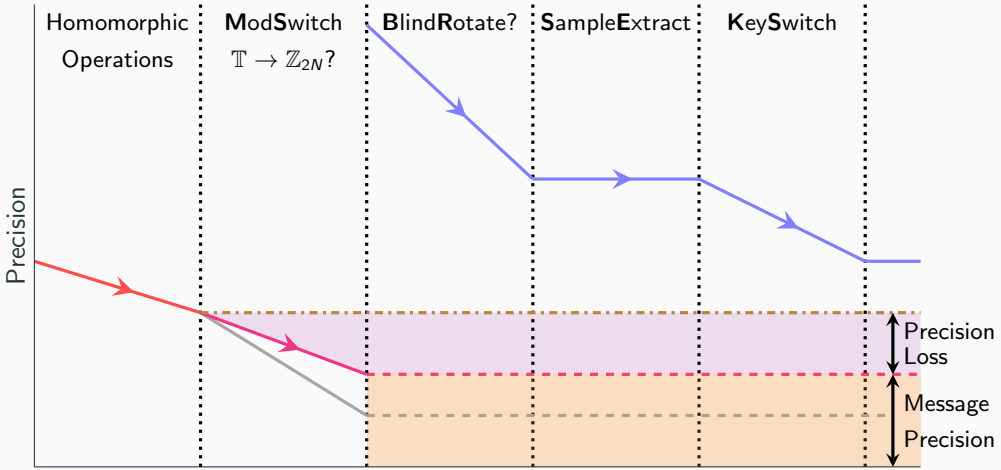
Given a function $f : \mathbb{T} \to \mathbb{T}$, the (functional) TFHE bootstrapping is a series of algorithms:



Precision in TFHE Bootstrapping

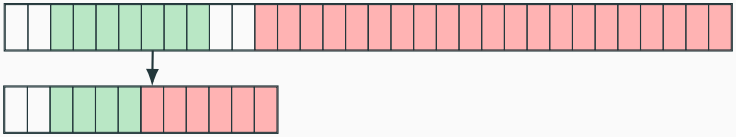


High Precision TFHE : Reduce the ModSwitch (Discretization) Error



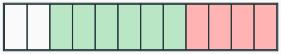
High Precision TFHE : Reduce the ModSwitch (Discretization) Error

ModSwitch to $\mathbb{Z}_{2N}$ adds error:



To reduce this, there are some approaches:

- Reduce Hamming weight of secret key $\mathbf{s}$, Ham($\mathbf{s}$).



- **Increase** $N \to 2^k N$

Enlarging $N \to N' = 2^k N$

Advantages

- Can use small TRLWE/TRGSW error $\Rightarrow$ BlindRotate noise decreases
- Error from ModSwitch decreases $\Rightarrow$ precision increases by k bits

Disadvantages

- Quasilinear growth for polynomial multiplication complexity : $O(N' \log N')$
- (Possibly) Larger keyswitch noise
- (Possibly) Larger public key / ciphertext size

Our Work 1 - Ciphertext Extension

Module homomorphism $\iota : \mathbb{T}_N[X] \to \mathbb{T}_{2^\nu N}[X]$

$$\iota : \mathbb{T}_N[X] \longrightarrow \mathbb{T}_{2^\nu N}[X],$$
$$p(x) = \sum_{i=0}^{N-1} p_i X^i \longmapsto p_{\mathsf{ext}}(X) = \sum_{i=0}^{N-1} p_i X^{2^\nu i}$$

Toy Example: $\mathbb{T}_2[X] \to \mathbb{T}_8[X]$

$$\iota(0.5X + 0.7) = 0.5X^4 + 0.7$$

Ciphertext Extension by Zero Padding

$$\iota\left(\mathsf{TRLWE}^N_{\mathcal{K}}(p(X))\right) = \mathsf{TRLWE}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(p_{\mathsf{ext}}(X)) \text{ for } p(X) \in \mathbb{T}_N[X],$$
$$\iota\left(\mathsf{TRGSW}^N_{\mathcal{K}}(q(X))\right) = \mathsf{TRGSW}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(q_{\mathsf{ext}}(X)) \text{ for } q(X) \in \mathbb{Z}_N[X],$$

Our Work 1 - Extended BlindRotate

The external product $\boxdot$ on ring dimension $2^\nu N$ naturally follows:

$$\mathsf{TRGSW}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(q_{\mathsf{ext}}(X)) \boxdot \mathsf{TRLWE}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(p(X)) = \mathsf{TRLWE}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(p(X) \cdot q_{\mathsf{ext}}(X)).$$

Which enables to evaluate the **CMux** circuit on ring dimension $2^\nu N$ while original ring dimension stays in N:

$$\mathsf{ACC} \leftarrow \mathsf{TRGSW}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(\mathbf{s}_i) \boxdot \left((X^{\bar{\mathbf{a}}_i} - 1) \cdot \mathsf{ACC}\right) + \mathsf{ACC}.$$

Thus, we **ModSwitch** to $\mathbb{Z}_{2^{\nu+1}N}$, gaining additional ν bits of precision.

Our Work 2 - Parallel External Product

Module isomorphism $\tau : \mathbb{T}_{2^\nu N}[X] \to \mathbb{T}_N^{2^\nu}[X]$

$$p(x) = \sum_{i=0}^{2^\nu N-1} p_i X^i \longmapsto \left(p^{(0)}(X), \cdots, p^{(2^\nu-1)}(X)\right)$$
$$= \left(\sum_{i=0}^{N-1} p_{2^\nu i} X^i, \cdots, \sum_{i=0}^{N-1} p_{2^\nu i+2^\nu-1} X^i\right).$$

Toy Example: $\mathbb{T}_8[X] \to \mathbb{T}_2^4[X]$

$$\tau(0.2X^7 + 0.1X^6 + 0.5X^4 + 0.3X + 0.7) = (0.7 + 0.5X, 0.3, 0.1X, 0.2X)$$

Ciphertext Folding

$$\tau\left(\mathsf{TRLWE}_{\mathcal{K}_{\text{ext}}}^{2^\nu N}(m)\right) = \left(\mathsf{TRLWE}_{\mathcal{K}}^{N}(m_0), \cdots, \mathsf{TRLWE}_{\mathcal{K}}^{N}(m_{2^\nu-1})\right) \text{ for } m(X) \in \mathbb{T}_N[X].$$

Our Work 2 - Parallel External Product

For special case, we can parallelize the external product over dimension $2^\nu N$ into 2^ν external products in dimension N:

Special case of External Product $\boxdot$

- **C1.** $\mathcal{K}_{ext}$ should be an extended key of $\mathcal{K}$.
- **C2.** $\mathsf{TRGSW}^{2^\nu N}_{\mathcal{K}_{ext}}(z)$ is an extension of $\mathsf{TRGSW}^{N}_{\mathcal{K}}(z)$, and $z \in \mathbb{Z}$.

$$\mathsf{TRGSW}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(z) \boxdot \mathsf{TRLWE}^{2^\nu N}_{\mathcal{K}_{\mathsf{ext}}}(m) \cong \begin{bmatrix} \mathsf{TRGSW}^{N}_{\mathcal{K}}(z) \boxdot \mathsf{TRLWE}^{N}_{\mathcal{K}}(m_0) \\ \mathsf{TRGSW}^{N}_{\mathcal{K}}(z) \boxdot \mathsf{TRLWE}^{N}_{\mathcal{K}}(m_1) \\ \vdots \\ \mathsf{TRGSW}^{N}_{\mathcal{K}}(z) \boxdot \mathsf{TRLWE}^{N}_{\mathcal{K}}(m_{2^\nu-1}) \end{bmatrix}$$

$$\cong \left(\mathsf{TRLWE}^{N}_{\mathcal{K}}(z \cdot m_0), \cdots, \mathsf{TRLWE}^{N}_{\mathcal{K}}(z \cdot m_{2^\nu-1})\right)$$

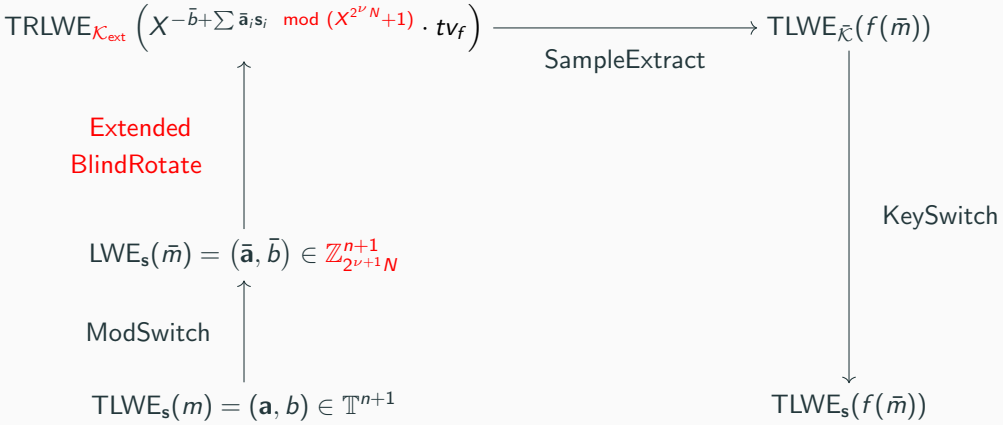
Our Work 2 - Parallel CMux / BlindRotate

With the parallel external product, we can build parallel CMux circuit:

$$\overrightarrow{\mathsf{ACC}} \leftarrow \begin{bmatrix} \mathsf{BSK}_i \boxdot \left(\tau\left(X^{\bar{a}_i} \cdot \tau^{-1}\left(\overrightarrow{\mathsf{ACC}}\right)\right)_0 - \overrightarrow{\mathsf{ACC}}_0\right) + \overrightarrow{\mathsf{ACC}}_0 \\ \mathsf{BSK}_i \boxdot \left(\tau\left(X^{\bar{a}_i} \cdot \tau^{-1}\left(\overrightarrow{\mathsf{ACC}}\right)\right)_1 - \overrightarrow{\mathsf{ACC}}_1\right) + \overrightarrow{\mathsf{ACC}}_1 \\ \vdots \\ \mathsf{BSK}_i \boxdot \left(\tau\left(X^{\bar{a}_i} \cdot \tau^{-1}\left(\overrightarrow{\mathsf{ACC}}\right)\right)_{2^\nu-1} - \overrightarrow{\mathsf{ACC}}_{2^\nu-1}\right) + \overrightarrow{\mathsf{ACC}}_{2^\nu-1} \end{bmatrix}$$

Equivalent to running 2^ν CMux circuits!

Summary : Extended BootStrapping



- Gains ν bit of precision
- Able to keep small ring dimension N
- Parallelized bootstrapping

Experimental Results

Implementation

We implemented our **E**xtended **B**oot**S**trapping on top of the TFHE library[2]

We also implemented three state-of-the-art full-domain bootstrapping algorithms ([KS21], [YXSCZ21], [CZBSG22]), which all uses TFHE bootstrapping as sub-algorithm. The codes are all available at our github page[3].

Experimental Environment

- CPU : Intel i9-13900K 5.8GHz 24 core, 32 threads
- RAM : DDR5 128GB
- Ubuntu 22.04

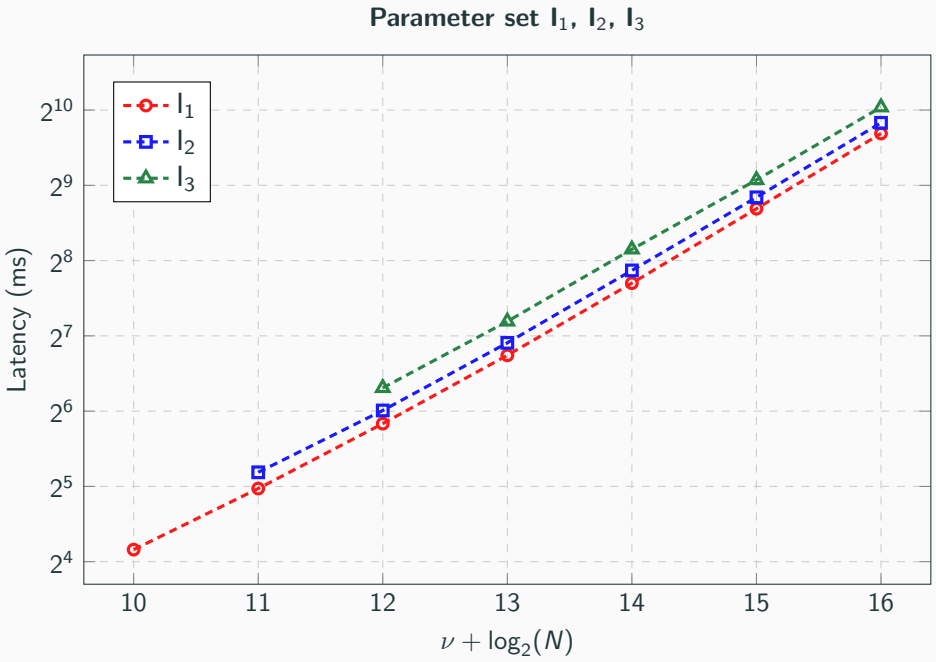
[2] https://tfhe.github.io/tfhe/
[3] https://github.com/Stirling75/Extended-BootStrapping

Parameters

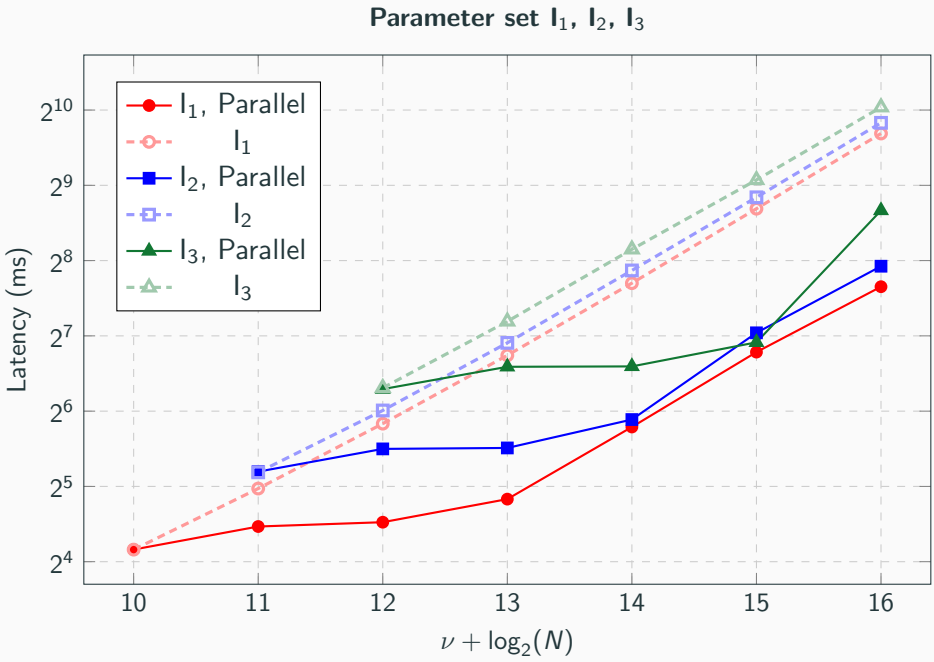
Table 1: TFHE parameter sets. λ indicates the security level of given parameter set.

Param Set	λ	TLWE		TRLWE			KSK		BSK	
		n	σ_{TLWE} $(\log_2)$	N	k	σ_{TRLWE} $(\log_2)$	ℓ_{KS}	B_{KS}	ℓ_{BS}	B_{BS}
I_1	80	750	-21.2	1024	1	-29.3	3	2^8	7	2^4
I_2	80	750	-21.2	2048	1	-32	3	2^8	7	2^4
I_3	80	750	-21.2	4096	1	-32	3	2^8	7	2^4
II	80	900	-25.7	2048	1	-32	5	2^6	7	2^4
III_1	128	670	-12.4	1024	1	-20.1	3	2^5	8	2^3
III_2	128	670	-12.4	2048	1	-32	3	2^5	8	2^3
III_3	128	670	-12.4	4096	1	-32	3	2^5	8	2^3
IV	128	1300	-26.1	2048	1	-32	5	2^6	7	2^4

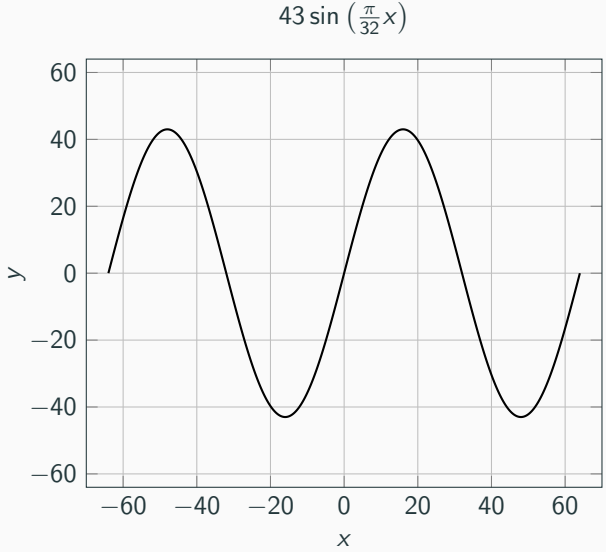
EBS : Performance (Non-Parallelized)



EBS : Performance (Parallelized)

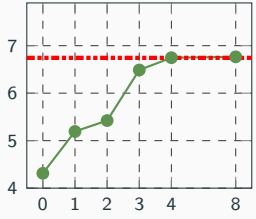
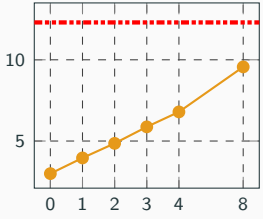
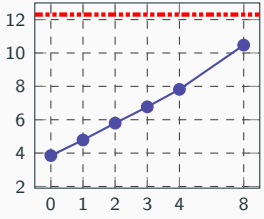
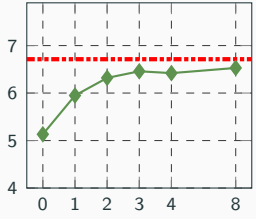
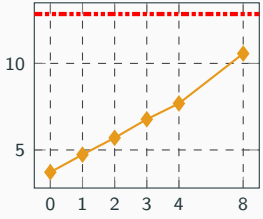
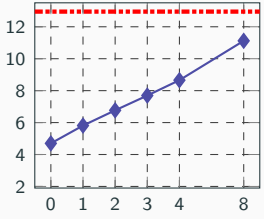


EBS : Function Evaluation



Homomorphic evaluation of function $f(x) = 43 \sin\left(\frac{\pi}{32}x\right)$ on domain $[-64, 64)$.

EBS : Function Evaluation

Parameter	FDFB-EBS [KS21]	TOTA-EBS [YXSCZ21]	Comp-EBS [CZBSG22]
I_1			
II			

Thank You

References i

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: fast fully homomorphic encryption over the torus. In: *Journal of Cryptology* 33.1 (2020), pp. 34–91 (cit. on p. 6).

[CZBSG22] Pierre-Emmanuel Clet, Martin Zuber, Aymen Boudguiga, Renaud Sirdey, and Cédric Gouy-Pailler. Putting up the swiss army knife of homomorphic calculations by means of TFHE functional bootstrapping. In: *Cryptology ePrint Archive* (2022) (cit. on pp. 22, 27).

[KS21] Kamil Kluczniak and Leonard Schild. FDFB: Full domain functional bootstrapping towards practical fully homomorphic encryption. In: *arXiv preprint arXiv:2109.02731* (2021) (cit. on pp. 22, 27).

[YXSCZ21] Zhaomin Yang, Xiang Xie, Huajie Shen, Shiying Chen, and Jun Zhou. TOTA: Fully Homomorphic Encryption with Smaller Parameters and Stronger Security. In: *Cryptology ePrint Archive* (2021) (cit. on pp. 22, 27).