KU LEUVEN

Amortized Bootstrapping Revisited

Simpler, Asymptotically-faster, Implemented

Antonio Guimarães, Hilder V. L. Pereira, **Barry van Leeuwen**

Institute of Computing, University of Campinas, Brazil;
COSIC, KU Leuven, Belgium

1 Outline

KU LEUVEN

1 Principles of FHE

1 Principles of FHE

$\mathrm{Enc}_{\mathbf{z}}(\mathrm{m}_1)$

$\mathrm{Enc}_{\mathbf{z}}(\mathrm{m}_2)$

1 Principles of FHE

$$\mathrm{Enc}_{\mathbf{z}}(\mathrm{m}_1) \qquad\qquad \mathrm{Enc}_{\mathbf{z}}(\mathrm{m}_2)$$

$$\mathrm{Enc}_{\mathbf{z}}(\mathrm{m}_1) * \mathrm{Enc}_{\mathbf{z}}(\mathrm{m}_2) = \mathrm{Enc}_{\mathbf{z}}(\mathrm{m}_1 * \mathrm{m}_2)$$

1 Learning with Errors

- Encryption:

$$\mathsf{Enc}_{\mathsf{sk}} m = (a, b = a \cdot s + \Delta \cdot m + e) = c$$

- Decryption:

$$\mathsf{Dec}_{sk}(c) = \left\lfloor \frac{b - a \cdot s}{\Delta} \right\rceil$$

- $\mathsf{Dec}_{sk}(\mathsf{Enc}_{sk}(m)) = m$ **iff** $\left\lfloor \frac{e}{\Delta} \right\rceil = 0$

1 Learning with Errors

Let $c_i = (a_i, b_i)$ be an encryption of m_i under a single secret key sk for every i. Then $c_1 + c_2 = (a_1 + a_2, b_1 + b_2)$ where

$$\begin{aligned} b_1 + b_2 &= (a_1 \cdot s + \Delta \cdot m_1 + e_1) + (a_2 \cdot s + \Delta \cdot m_2 + e_2) \\ &= (a_1 + a_1) \cdot s + \Delta(m_1 + m_2) + \mathbf{e_1} + \mathbf{e_2} \end{aligned}$$

Addition: Linear error growth, manageable
Multiplication: Exponential error growth, unmanageable

1 Learning with Errors

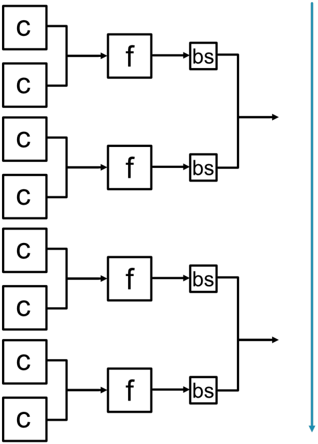
Let $c_i = (a_i, b_i)$ be an encryption of m_i under a single secret key sk for every i. Then $c_1 + c_2 = (a_1 + a_2, b_1 + b_2)$ where

$$\begin{aligned} b_1 + b_2 &= (a_1 \cdot s + \Delta \cdot m_1 + e_1) + (a_2 \cdot s + \Delta \cdot m_2 + e_2) \\ &= (a_1 + a_1) \cdot s + \Delta(m_1 + m_2) + \mathbf{e_1} + \mathbf{e_2} \end{aligned}$$

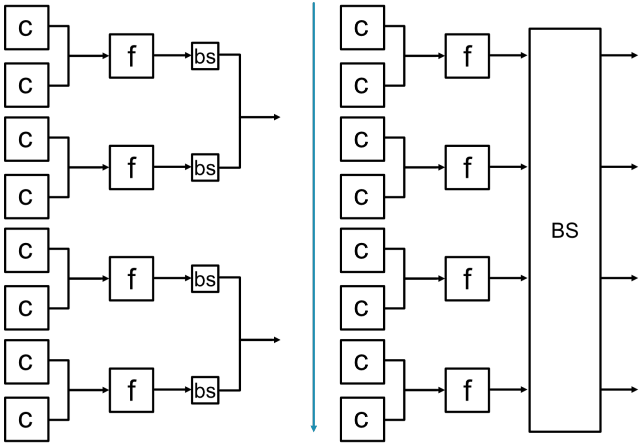
Addition: Linear error growth, manageable
Multiplication: Exponential error growth, unmanageable

Bootstrapping: An algorithm to reduce the noise in a ciphertext

1 Types of Bootstrapping

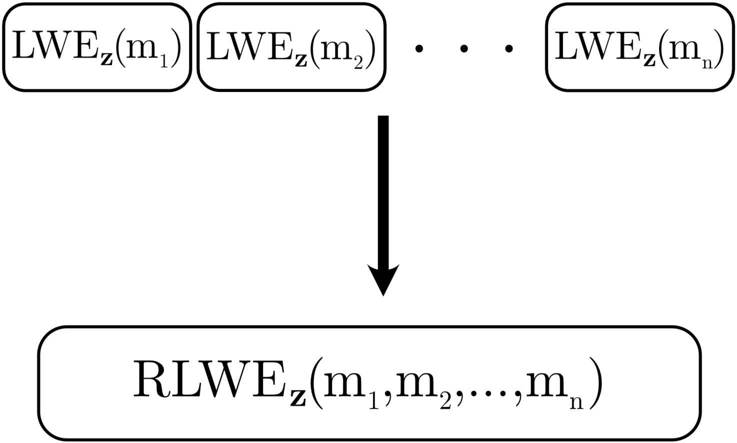


1 Types of Bootstrapping



2 Outline

2 RLWE Packing



2 Ring GSW

Ring LWE:

- LWE, but over Rings.

2 Ring GSW

Ring LWE:

- LWE, but over Rings.
- Allows encryption of $a(X) \in \mathcal{R}_Q$, where

$$\mathcal{R}_Q = \mathbb{Z}_Q[X]/\langle f(X) \rangle,$$

for some polynomial $f(X)$

2 Ring GSW

Ring LWE:

- LWE, but over Rings.
- Allows encryption of $a(X) \in \mathcal{R}_Q$, where

$$\mathcal{R}_Q = \mathbb{Z}_Q[X]/\langle f(X) \rangle,$$

for some polynomial $f(X)$

Encryption: For a message m and secret key $\mathsf{sk} = s$

$$\mathsf{Enc}_{\mathsf{sk}}^{\mathsf{RGSW}}(m) = [a \mid a \cdot s + e] + m \cdot G_\alpha$$

where $G = I_2 \otimes \mathbf{g}$, $\mathbf{g} = (B^0, \dots, B^{d-1})$ for $B \in \mathbb{N}$, $d = \log_B(Q)$.

2 Accumulator

Let $c = (b, a) = \mathsf{Enc}_z^{\mathsf{LWE}}(m)$

$$\mathsf{ACC} = \mathsf{Enc}_s^{\mathsf{GSW}}(T(x) \cdot X^b)$$

$$\mathsf{ACC} = \mathsf{ACC} \cdot \mathsf{Enc}_s^{\mathsf{GSW}}(X^{-a_i s_i})$$

2 Accumulator

Let $c = (b, a) = \mathsf{Enc}_z^{\mathsf{LWE}}(m)$

$$\mathsf{ACC} = \mathsf{Enc}_s^{\mathsf{GSW}}(T(x) \cdot X^b)$$

$$\mathsf{ACC} = \mathsf{ACC} \cdot \mathsf{Enc}_s^{\mathsf{GSW}}(X^{-a_i s_i})$$

Once done for all i:

$$\mathsf{ACC} = \mathsf{Enc}_{\hat{s}}^{\mathsf{GSW}}(T(X) \cdot X^{\Delta \cdot m + e})$$

2 Accumulator

Let $c = (b, a) = \mathsf{Enc}_z^{\mathsf{LWE}}(m)$

$$\mathsf{ACC} = \mathsf{Enc}_s^{\mathsf{GSW}}(T(x) \cdot X^b)$$

$$\mathsf{ACC} = \mathsf{ACC} \cdot \mathsf{Enc}_s^{\mathsf{GSW}}(X^{-a_i s_i})$$

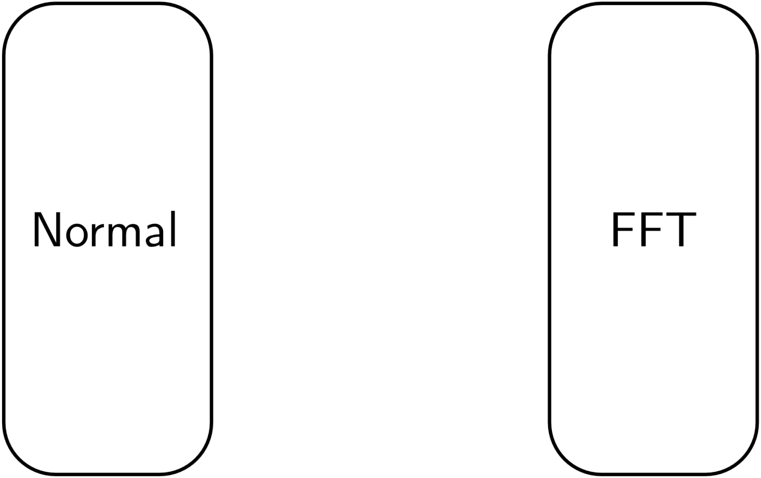
Once done for all i:

$$\mathsf{ACC} = \mathsf{Enc}_{\hat{s}}^{\mathsf{GSW}}(T(X) \cdot X^{\Delta \cdot m + e})$$

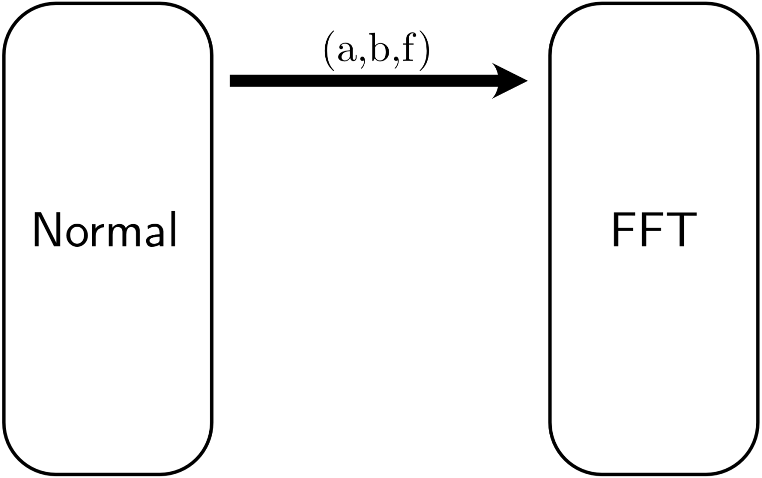
Well chosen test vector, $T(X)$, allows us to obtain:

$$\mathsf{ACC} = \mathsf{Enc}_{\hat{s}}^{\mathsf{GSW}}(\Delta \cdot m + e)$$

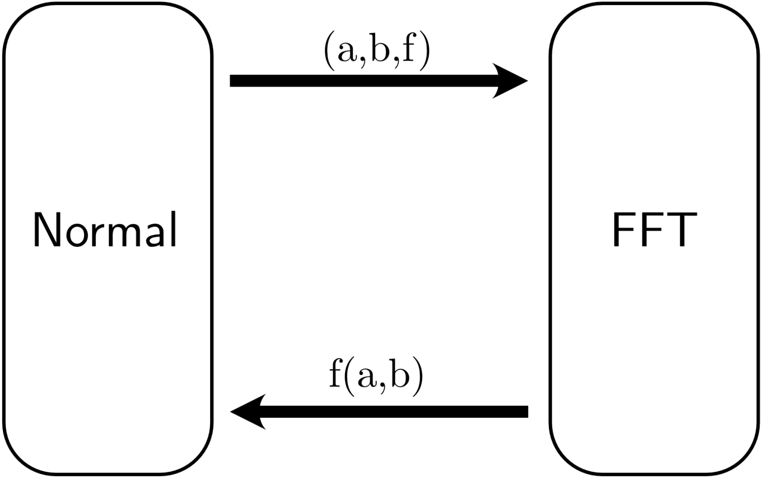
2 FFT/NTT Space



2 FFT/NTT Space



2 FFT/NTT Space



2 Double-CRT Form

- RGSW over $\mathcal{R}_Q = \mathbb{Z}_Q/\langle X^p + 1\rangle$

2 Double-CRT Form

- RGSW over $\mathcal{R}_Q = \mathbb{Z}_Q / \langle X^p + 1 \rangle$, $p \equiv 1 \mod 2N$

2 Double-CRT Form

- RGSW over $\mathcal{R}_Q = \mathbb{Z}_Q / \langle X^p + 1 \rangle$, $p \equiv 1 \mod 2N$

Double-CRT Form:

Theorem (Decomposition Theorem)

Let $Q = \prod_{i=1}^{l} q_i$ be a decomposition into primes, then

$$\mathcal{R}_Q = \mathbb{Z}_Q[X] / \langle f(X) \rangle = \prod_{i=1}^{l} \mathbb{Z}_{q_i} / \langle f(X) \rangle$$

2 Double-CRT Form

Let $a(X), b(X) \in \mathcal{R}_Q$, then for and $a(X) = (a_0, \ldots, a_{n-1})$

$$\texttt{Mat}(a(X)) = \begin{pmatrix} \mathsf{NTT}_{q_1}(a(X) \mod q_1) \\ \vdots \\ \mathsf{NTT}_{q_l}(a(X) \mod q_l) \end{pmatrix} = \begin{pmatrix} a_{1,0} & \cdots & a_{1,n-1} \\ \vdots & \ddots & \vdots \\ a_{l,0} & \cdots & a_{l,n-1} \end{pmatrix}$$

2 Double-CRT Form

Let $a(X), b(X) \in \mathcal{R}_Q$, then for and $a(X) = (a_0, \dots, a_{n-1})$

$$\texttt{Mat}(a(X)) = \begin{pmatrix} \mathsf{NTT}_{q_1}(a(X) \mod q_1) \\ \vdots \\ \mathsf{NTT}_{q_l}(a(X) \mod q_l) \end{pmatrix} = \begin{pmatrix} a_{1,0} & \cdots & a_{1,n-1} \\ \vdots & \ddots & \vdots \\ a_{l,0} & \cdots & a_{l,n-1} \end{pmatrix}$$

Addition: $\mathsf{Mat}(a(X)) + \mathsf{Mat}(b(X)) = \mathsf{Mat}(a(X) + b(X))$

Multiplication: $\mathsf{Mat}(a(X)) \odot \mathsf{Mat}(b(X)) = \mathsf{Mat}(a(X) \cdot b(X))$

2 Double-CRT RGSW

Definition

Let $Q = \prod_{i=1}^{l}(q_i)$ as in the Decomp. Theorem. Let $d \in \mathbb{N}$ be the "number of digits". Assume that $d \mid l$ and let $k = l/d$. Then, for $1 \leq i \leq d$, define the i'th "CRT Digit" as $D_j := \prod_{j=(i-1)\cdot k+1}^{i\cdot k} q_j$ such that D_j is a product of k consecutive primes. Let $Q_i = Q/D_i$ and let $\hat{Q}_i = Q_i^{-1} \mod D_i$. Then the scaled gadget matrix, $\mathbf{G}_\alpha$, is given by

$$\mathbf{G}_\alpha = I_2 \otimes (Q_1 \cdot \hat{Q}_i \cdot \alpha_1, \ldots, Q_d \cdot \hat{Q}_d \cdot \alpha_d)$$

where $\alpha_i = \alpha \mod D_i$.
Moreover, for any two polynomials, $a(X), b(X)$, let

$$G_\alpha^{-1}(a(X), b(X)) = \left(\mathsf{CRT}^{-1}_{D_1,\ldots,D_d}(a), \mathsf{CRT}^{-1}_{D_1,\ldots,D_d}(b)\right)$$

then it follows that $G_\alpha^{-1}(a(X), b(X) \cdot G_\alpha = (a(X), b(X))$.

2 Shrinking

Algorithm 2: Shrink ciphertext

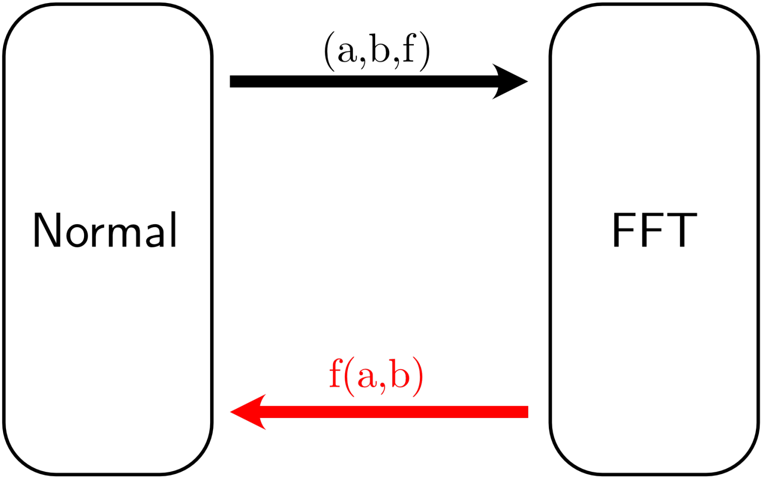
Input: $\mathbf{C} \in \tilde{\mathcal{R}}_Q^{2d\times 2}$ in double-CRT form, scaling factor α, CRT digits $D_1, \ldots, D_d$, and $k \in \mathbb{Z}$ such that $1 \leq k < d$.
Output: $\mathbf{C}' \in \tilde{\mathcal{R}}_{Q'}^{2(d-k)\times 2}$ and new correction factor $\alpha' \in \mathbb{Z}$.
Complexity: $4 \cdot (d-k) \cdot \ell$ NTTs and $O(k \cdot \ell^2 \cdot p)$ multiplications on $\mathbb{Z}_{q_i}$.
Noise growth: $E \mapsto O(E/D^{(k)} + \sqrt{p} \cdot S)$

```
D^(k) := D_1 · ... · D_k
Q' := Q / D^(k)
C̄ := π_k(C) ∈ R̃_{Q'}^{2(d−k)×2}
for 1 ≤ i ≤ 2·(d − k) do
    c_i := ModSwt_{Q→Q'}(row_i(C̄))
Define C' such that row_i(C') = c_i.
β := CRT_{D_{k+1},...,D_d}(D^(k), ..., D^(k))^{−1} mod Q'.
α' := α · β mod Q'
return C', α'
```


KU LEUVEN

3 Outline

3 The Problem:



3 Homomorphic Inverse NTT

- $a(X), b(X) \in \mathbb{Z}_Q[X]/\langle X^p + 1\rangle, p \equiv 1 \mod 2N$

3 Homomorphic Inverse NTT

- $a(X), b(X) \in \mathbb{Z}_Q[X]/\langle X^p + 1\rangle, p \equiv 1 \mod 2N$
- ψ be the $2N$'th root of unity, $\boldsymbol{\psi} = (\psi^0, \ldots, \psi^{2N-1})$

3 Homomorphic Inverse NTT

- $a(X), b(X) \in \mathbb{Z}_Q[X]/\langle X^p + 1\rangle, p \equiv 1 \mod 2N$
- ψ be the $2N$'th root of unity, $\boldsymbol{\psi} = (\psi^0, \ldots, \psi^{2N-1})$

Normally:

$$\mathsf{NTT}^{-1}(\mathsf{NTT}(a(X) \odot \mathsf{NTT}(b(X))) \equiv N \cdot a(X) \cdot b(X) \mod \langle X^N - 1, q\rangle$$

KU LEUVEN

3 Homomorphic Inverse NTT

- $a(X), b(X) \in \mathbb{Z}_Q[X]/\langle X^p + 1\rangle, p \equiv 1 \mod 2N$
- ψ be the $2N$'th root of unity, $\boldsymbol{\psi} = (\psi^0, \ldots, \psi^{2N-1})$

Normally:

$$\mathsf{NTT}^{-1}(\mathsf{NTT}(a(X) \odot \mathsf{NTT}(b(X))) \equiv N \cdot a(X) \cdot b(X) \mod \langle X^N - 1, q\rangle$$

Negacyclic Polynomial:

$$\boldsymbol{\psi}^{-1} \odot \mathsf{NTT}^{-1}(\mathsf{NTT}(\boldsymbol{\psi} \odot a(X)) \odot \mathsf{NTT}(\boldsymbol{\psi} \odot b(X)) \equiv N \cdot a(X) \cdot b(X) \mod \langle X^p + 1, q\rangle$$

3 Bootstrapping Algorithm

- **Input:** $c_i = \mathsf{Enc}_{\mathsf{sk}}^{\mathsf{LWE}}(m_i)$ for $i \in \{0, \ldots, N-1\}$

3 Bootstrapping Algorithm

- **Input:** $c_i = \mathsf{Enc}_{\mathsf{sk}}^{\mathsf{LWE}}(m_i)$ for $i \in \{0, \ldots, N-1\}$
- Standard ciphertext packing method to obtain $\mathbf{c} = \mathsf{Enc}_{\mathbf{z}}^{\mathsf{RLWE}}(a(X), b(X))$

3 Bootstrapping Algorithm

Algorithm 13: NTTDec, the homomorphic partial decryption

Input: Encryption $\mathbf{c} \in \hat{\mathcal{R}}_p\mathsf{LWE}_z(m, E^{(in)})$. Bootstrapping keys $\mathbf{K}_i \in \tilde{\mathcal{R}}_Q\mathsf{GSW}_s^d(1 \cdot X^{-\bar{z}_i}, E)$, where $(\bar{z}_0, \ldots, \bar{z}_{N-1}) := \mathsf{NTT}(\boldsymbol{\psi} \odot z) \in \mathbb{Z}_p^N$, and key-switching keys for all the Galois automorphims $\eta_a : X \mapsto X^a$. Vectors with powers of $2N$-th root of unity ψ in $\mathbb{Z}_p$, i.e., $\boldsymbol{\psi} = (\psi^0, \ldots, \psi^{N-1})$ and $\boldsymbol{\psi}^{-1} = (\psi^0, \ldots, \psi^{-(N-1)})$

Output: $\bar{\mathbf{C}}_i \in \tilde{\mathcal{R}}_Q\mathsf{GSW}_s^d(\alpha \cdot X^{e_i + \Delta \cdot m_i}, E'')$ for $0 \leq i < N$

Complexity: $O\left(N^{1+\frac{1}{\rho}} \cdot \rho \cdot d^2 \cdot \ell\right)$ NTTs and $O\left(N^{1+\frac{1}{\rho}} \cdot \rho \cdot d \cdot \ell^2 \cdot p\right)$ products over $\mathbb{Z}_{q_i}$

Noise growth: $(E, E_k) \mapsto E'' = O\left((\sqrt{d} \cdot D \cdot p \cdot \|s\|)^\rho \cdot \sqrt{n \cdot p} \cdot (E \cdot \|s\| + E_k \cdot \sqrt{d} \cdot D)\right)$

```
Parse c as (a, b) ∈ R̂_p^2 where R̂_p = Z_p[X]/⟨X^N + 1⟩
(ā_0, ..., ā_{N-1}) ← ψ ⊙ NTT(a) ;                 ▷ NTT(a) ∈ Z_p^N
(b̄_0, ..., b̄_{N-1}) ← ψ ⊙ NTT(b) ;                 ▷ NTT(b) ∈ Z_p^N
for i ∈ {1, ..., n} do
    K̄_i = η_{ā_i}(K_i) ;                           ▷ K̄_i ∈ R̃_Q GSW^d_{η_{ā_i}(s)}(1 · X^{-ā_i·z̄_i}, E)
    KS_{η_{ā_i}(s)→s}(K̄_i) ;                       ▷ K̄_i ∈ R̃_Q GSW^d_s(1 · X^{-ā_i·z̄_i}, E')
    K̃_i = X^{b̄_i} · K̄_i ;                          ▷ K̃_i ∈ R̃_Q GSW^d_s(1 · X^{b̄_i - ā_i·z̄_i}, E')
(C̄_0, ..., C̄_{N-1}) = NTT^{-1}(K̃_0, ..., K̃_{N-1}) ; ▷ C̄_i ∈ R̃_Q GSW^d_s(1 · X^{e_i + Δ·m_i}, E'')
                                                   ▷ E'' = O((√d · D · p · ‖s‖)^ρ · √N · (E' + E_k))
```

3 Bootstrapping Algorithm

- Message Extraction by standard means

3 Bootstrapping Algorithm

- Message Extraction by standard means
- **Output:** $\mathbf{c}_1, \ldots, \mathbf{c}_n$ under secret key sk.

4 Outline

KU LEUVEN

4 Relative Comparison

Scheme	Total cost	Messages	Amortized cost	Noise overhead
2014/816	$\tilde{O}(n)$	1	$\tilde{O}(n)$	$\tilde{O}(n^{1.5})$
2016/870	$O(n)$	1	$O(n)$	$\tilde{O}(n)$
2018/532	$\tilde{O}(3^{\rho} \cdot n^{1+1/\rho})$	$O(n)$	$\tilde{O}(3^{\rho} \cdot n^{1/\rho})$	$\tilde{O}(n^{2+3\cdot\rho})$
This work	$O(\rho \cdot n^{1+1/\rho})$	$O(n)$	$O(\rho \cdot n^{1/\rho})$	$\tilde{O}(n^{1+\rho})$

Table: *Comparison of number of homomorphic operations and noise growth of bootstrapping algorithms of different schemes based on worst-case lattice problems with polynomial approximation factor. The notation $\tilde{O}$ hides polylogarithmic factors in n.*

4 Relative Comparison

p	N	Without shrinking		With shrinking		Shrinking Speedup
		Exec. Time	Amortized Time	Exec. Time	Amortized Time	
12289	512	1,078,033	2,106	695,761	1,359	1.5
12289	1024	3,546,423	3,463	2,132,504	2,083	1.7

Table: *Execution time, in milliseconds, for the INTT for $\ell = 4$ and $\rho = 2$.*

4 Relative Comparison

p	N	Without shrinking		With shrinking		Shrinking Speedup
		Exec. Time	Amortized Time	Exec. Time	Amortized Time	
12289	512	1,078,033	2,106	695,761	1,359	1.5
12289	1024	3,546,423	3,463	2,132,504	2,083	1.7

Table: *Execution time, in milliseconds, for the INTT for $\ell = 4$ and $\rho = 2$.*

p	n	$\ell = d$	Total Time	Amortized Time	Speedup
12289	1024	3	871,827	851	3.4
		4	1,540,075	1,504	1.7

Table: *Execution time, in milliseconds, for the amortized bootstrapping. Speedup is over the fastest parameter of the non-amortized bootstrapping.*

Questions?

Find the paper on eprint: 2023/014

Code available on: https://github.com/antoniocgj/Amortized-Bootstrapping