ICICS 2024 PRESENTS

Amortized Functional Bootstrapping for Homomorphic Evaluation of Encrypted Functions

Yan Xu, Li-Ping Wang*, Huaxiong Wang

August 28, 2024, Greece

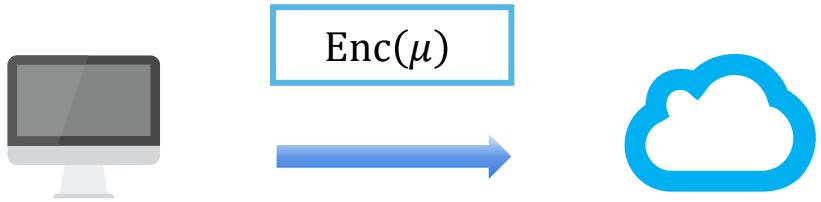
INSTITUTE OF INFORMATION ENGINEERING, CAS

Contents

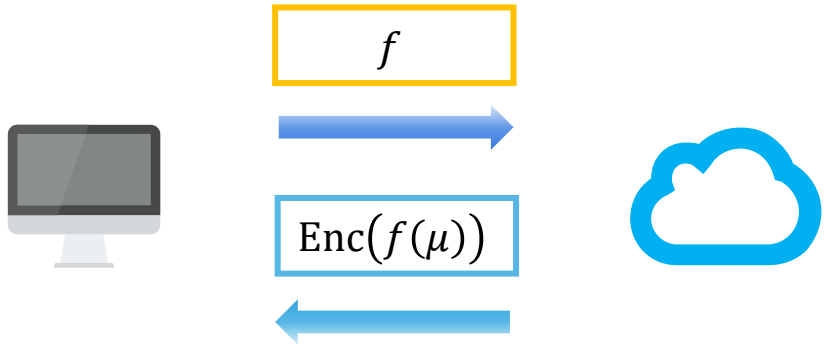
1 Background and Motivation

Background

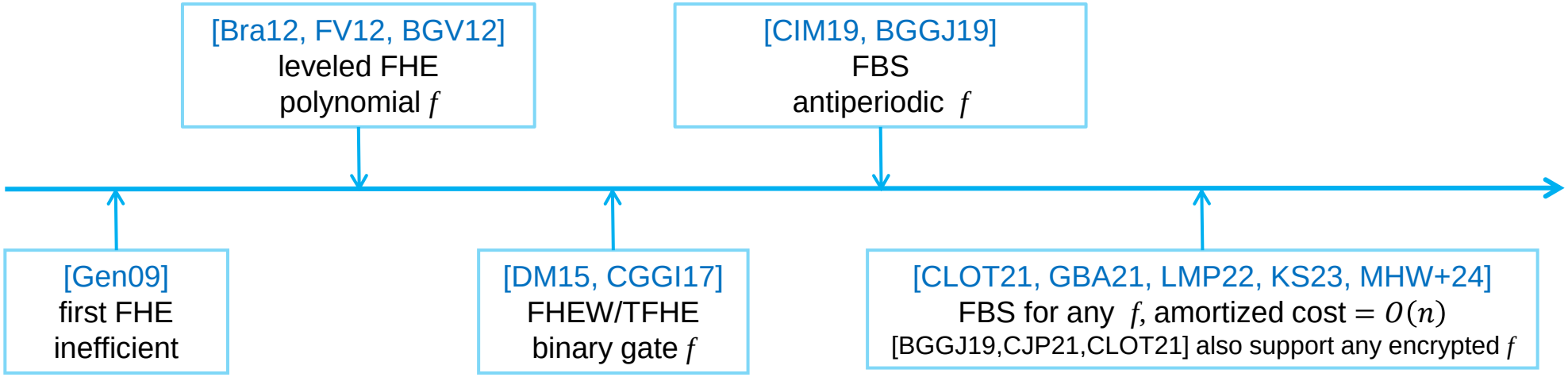
Phase 1: Upload data



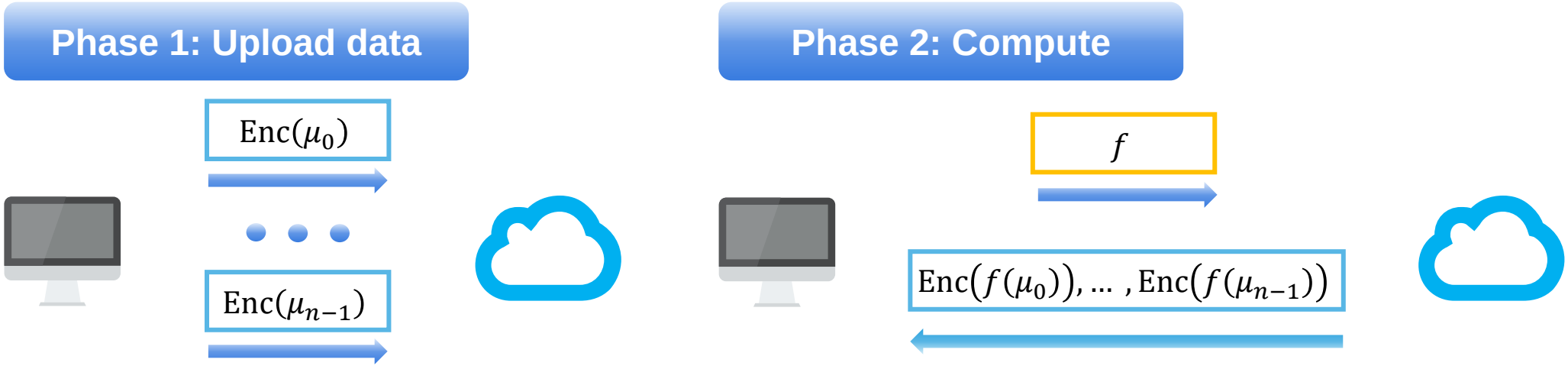
Phase 2: Compute



Related works



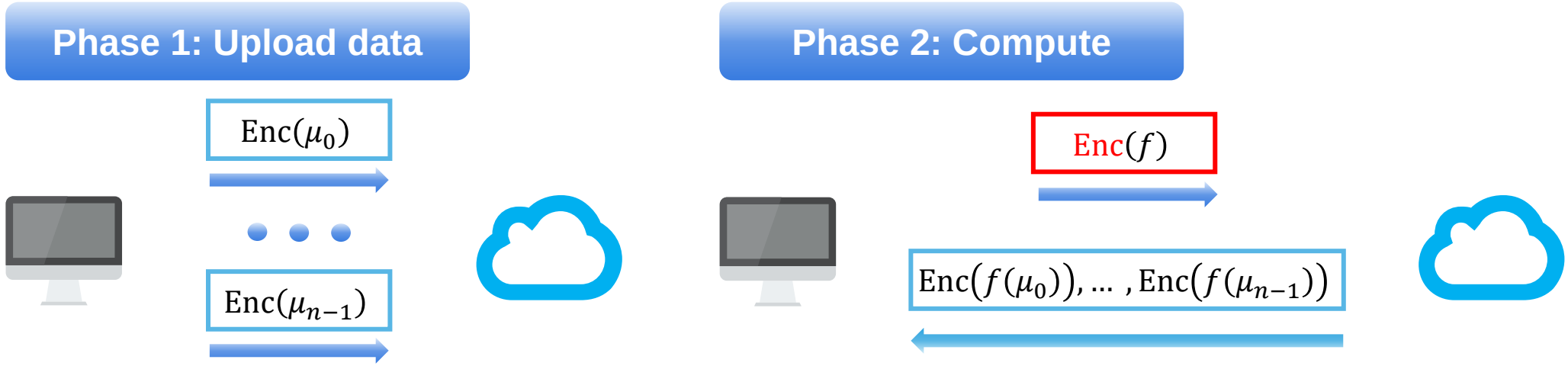
Background



Related works

Year	2018	2023			
Algorithm	[MS18]	[GPL23, MKMS23]	[LW23a]	[LW23b]	[LW23c]
Function	binary gate f	any f	binary gate f	binary gate f	any f
Amortized cost	$\tilde{O}(3^{\rho} n^{1/\rho})$	$\tilde{O}(\rho n^{1/\rho})$	$\tilde{O}(n^{0.75})$	$\tilde{O}(1)$	$\tilde{O}(1)$
Type	FHEW-like FBS				BFV/BGV-based FBS

Motivation



Gap No amortized FBS can be applied to this case!

Problem How to construct an amortized FBS to compute an encrypted f?

- Any function f is supported;
- Amortized computational cost is $\tilde{O}(1)$;
- Output errors are independent of f;

2 Preliminaries

- LWE-based FHE
- Amortized FHEW-like bootstrapping

LWE-based FHE

LWE-based Ciphertexts

LWE $\quad \mathrm{LWE}_{\boldsymbol{s}}^{Q,\Delta}(\mu) = \left(\boldsymbol{a}, b = [\langle \boldsymbol{a}, \boldsymbol{s} \rangle + e + \Delta\mu]_Q\right) \in \mathbb{Z}_Q^{n+1}$

RLWE $\quad \mathrm{RLWE}_{z,\mathcal{R}}^{Q,\Delta}(\mu) = \left(a, b = [a \cdot z + e + \Delta\mu]_Q\right) \in \mathcal{R}_Q^2$

REG $\quad \mathrm{REG}_{z,\mathcal{R}}^{Q}(\mu) = \left(\mathrm{RLWE}_{z,\mathcal{R}}^{Q,1}(\mu), \mathrm{RLWE}_{z,\mathcal{R}}^{Q,B}(\mu), \dots, \mathrm{RLWE}_{z,\mathcal{R}}^{Q,B^{\ell-1}}(\mu)\right) \in \mathcal{R}_Q^{2\ell}$

RGSW $\quad \mathrm{RGSW}_{z,\mathcal{R}}^{Q}(\mu) = \left[\mathrm{REG}_{z,\mathcal{R}}^{Q}(-z \cdot \mu), \mathrm{REG}_{z,\mathcal{R}}^{Q}(\mu)\right] \in \mathcal{R}_Q^{2\ell \times 2}$

Note: $B \ll Q, \ell = \lceil \log_B Q \rceil$

LWE-based FHE

Basic Homomorphic Operations

$\mu_0 + \mu_1$ Arithmetic addition of ciphertexts with the same type

$\mu_0 \cdot \mu_1$

$\mathrm{REG} \boxtimes \mathcal{R}$: $\boldsymbol{C} \boxtimes \mu_1 = \boldsymbol{C}^T \mathrm{g}^{-1}(\mu_1) \bmod Q$

$\mathrm{RGSW} \boxtimes \mathrm{RLWE}$: $\boldsymbol{C} \boxtimes \boldsymbol{d} = \boldsymbol{C}^T \mathrm{g}^{-1}(\boldsymbol{d}) \bmod Q$

$\theta(\mu)$

$$(a, b) = \mathrm{RLWE}_{z,\mathcal{R}}^{Q,\Delta}(\mu) \to \big(\theta(a), \theta(b)\big)$$

Automorphism (Aut)

$$\to \Big(\big(0, \theta(b)\big) - \mathrm{REG}_{z,\mathcal{R}}^{Q}\big(\theta(z)\big) \boxtimes \theta(a)\Big) \bmod Q$$

Key Switching (KS)
$\theta(z) \to z$

Note: θ is an automorphism of $\mathcal{R}$

gadget vector $\boldsymbol{g} = \big(1, B, \dots, B^{\ell-1}\big)$, $\langle \mathrm{g}^{-1}(\mu_1), \boldsymbol{g} \rangle = \mu_1 \pmod Q$

LWE-based FHE

Advanced Homomorphic Operations

Pack LWEs

$$\mathrm{LWE}_{\boldsymbol{s}_{\mathrm{in}}}^{q,\Delta}(\mu_0), \mathrm{LWE}_{\boldsymbol{s}_{\mathrm{in}}}^{q,\Delta}(\mu_1), \dots, \mathrm{LWE}_{\boldsymbol{s}_{\mathrm{in}}}^{q,\Delta}(\mu_{N-1}) \rightarrow \mathrm{RLWE}_{s,\hat{\mathcal{R}}}^{q,\Delta}\left(\sum_{i=0}^{N-1} \mu_i X^i\right)$$

Coefficient encoding and KS

Extract

$$(c_0, c_1) = \mathrm{RLWE}_{z,\mathcal{R}}^{Q,1}\left(\mu = \sum_{i=0}^{q-2} \mu_i X^i\right) \rightarrow \mathrm{LWE}_{\boldsymbol{z}}^{Q,1}(\mu_0)$$

Extracting coefficients of c_0, c_1

If $\mu = vX^{-\varphi}, v = \sum_{i=0}^{q-2} \sum_{j=0}^{i} F_j X^i, \varphi \in [q-1]: \mu_0 = F_\varphi$

Scheme Switch

$$\mathrm{REG}_{z,\mathcal{R}}^{Q}(\mu) \rightarrow \mathrm{RGSW}_{z,\mathcal{R}}^{Q}(\mu)$$

Similar to KS

Note: $\hat{\mathcal{R}} = \mathbb{Z}[X]/(X^N + 1), \mathcal{R} = \mathbb{Z}[X]/\left(\sum_{i=0}^{q-1} X^i\right)$, N is a power of two, odd prime $q = 1 (\mathrm{mod}\ 2N)$

FHEW-like FBS

$\boldsymbol{c} = (\boldsymbol{a}, b) = \mathrm{LWE}_{\boldsymbol{s}}^{q,\Delta}(\mu)$

High noise

→ Blind rotate →

$\boldsymbol{c}_{BR} = \mathrm{RLWE}_{z,\mathcal{R}}^{Q,1}\left(X^{\langle \boldsymbol{a},\boldsymbol{s}\rangle - b}\right)$

by RGSW ⊠ RLWE and

Aut ψ_k: $\mathcal{R} \to \mathcal{R}$ s.t. $\psi_k(m(X)) \to \psi_k\left(m\left(X^k\right)\right)$, $k \in \mathbb{Z}_q^*$

→ Multiply with v and extract →

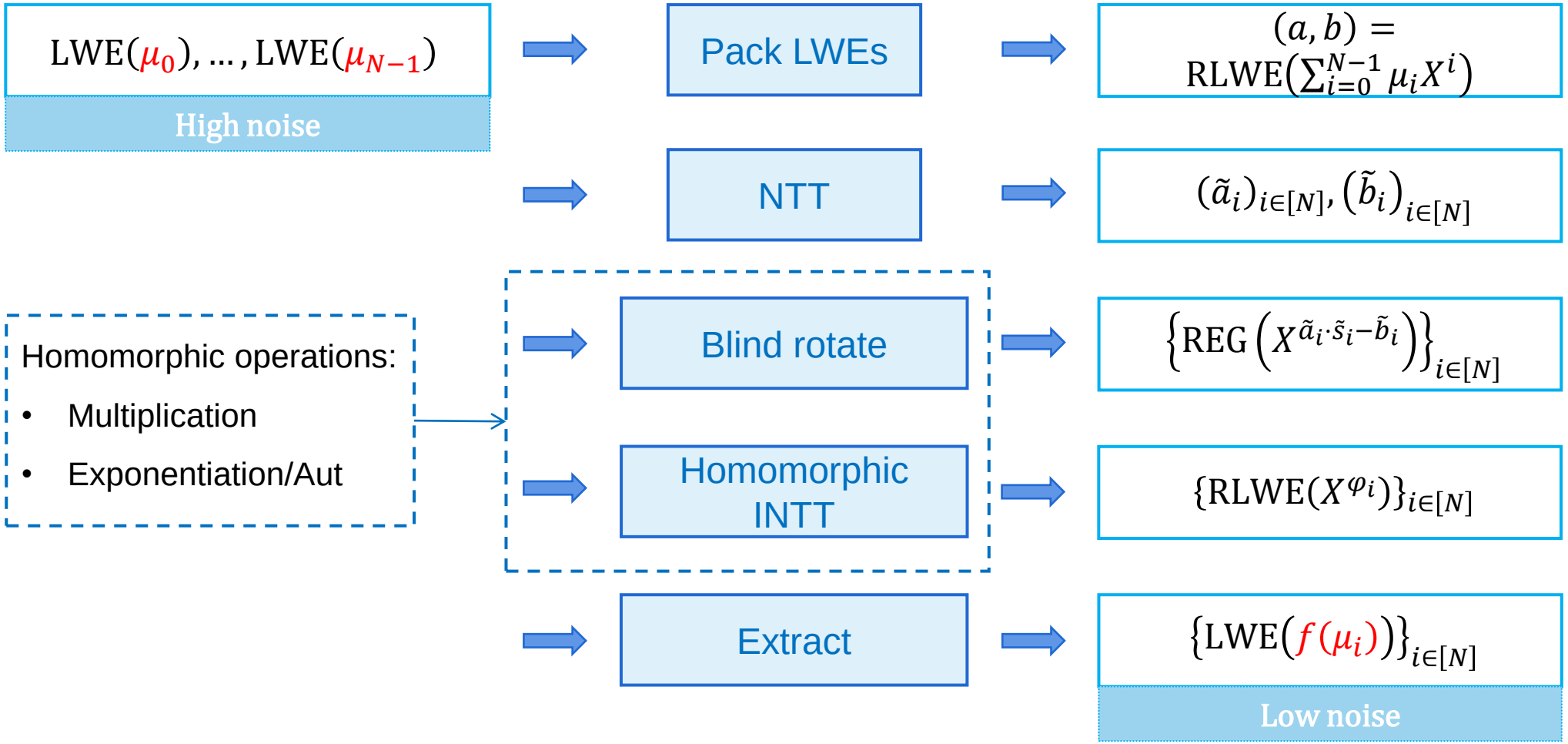
$\boldsymbol{c}_{out} = \mathrm{LWE}_{\boldsymbol{z}}^{Q,\Delta}(f(\mu))$

Low noise

$v = \sum_{i=0}^{q-2} \sum_{j=0}^{i} F_j X^i$

$F_\varphi = f\left((b - \langle \boldsymbol{a}, \boldsymbol{s}\rangle) \bmod q\right)$

Amortized FHEW-like FBS



Note: $(\tilde{s}_i)_{i\in[N]} = \mathrm{NTT}(s), \tilde{\varphi}_i = \tilde{a}_i \cdot \tilde{s}_i - \tilde{b}_i \pmod q$,

$\mathrm{INTT}\left((\tilde{\varphi}_i)_{i\in[N]}\right) = \sum_{i=0}^{N-1} \varphi_i X^i = (a \cdot s - b) \bmod q = \sum_{i=0}^{N-1} (\mu_i + e_i) X^i$

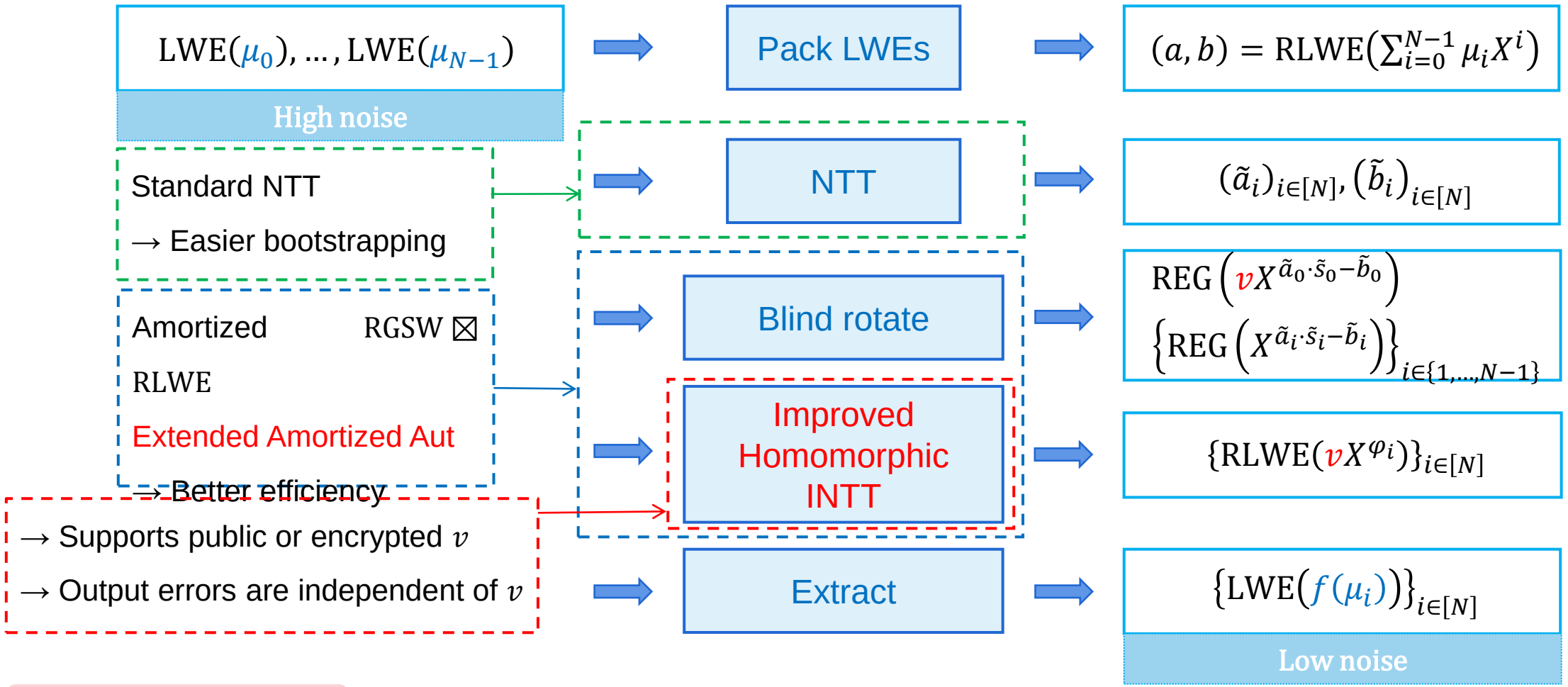
3

Our Work

—— Amortized Functional Bootstrapping

- Extended amortized homomorphic automorphism
- Improved homomorphic inverse NTT
- Amortized functional bootstrapping for encrypted functions

Framework



Our amortized FBS

Any public or encrypted function f is supported

Amortized cost is $\tilde{O}(1)$

Output errors are independent of f

Extended Amortized Aut

Tensor Rings

Different odd primes	q, p_0, p_1, p_2
Linearly disjoint fields	$K = \mathbb{Q}[\xi_q], K_0 = \mathbb{Q}\left[\xi_{q_0^{'}}\right], K_1 = \mathbb{Q}\left[\xi_{q_1^{'}}\right], K_2 = \mathbb{Q}\left[\xi_{q_2^{'}}\right]$
Rings of integers	$\mathcal{R}, \mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2$
Base of $\mathcal{R}_k$	$B_k = \{r_{k,0}, r_{k,1}, \dots, r_{k,q_k-1}\}$
Base of dual ring $\mathcal{R}_k^{\vee}$	$B_k^{\vee} = \{r_{k,0}^{\vee}, r_{k,1}^{\vee}, \dots, r_{k,q_k-1}^{\vee}\}$
Tensor rings	$\tilde{\mathcal{R}} = \mathcal{R} \otimes \mathcal{R}_0 \otimes \mathcal{R}_1, \tilde{\mathcal{R}}_0 = \mathcal{R} \otimes \mathcal{R}_0, \tilde{\mathcal{R}}_1 = \mathcal{R} \otimes \mathcal{R}_1,$ $\tilde{\mathcal{R}}_2 = \mathcal{R} \otimes \mathcal{R}_0 \otimes \mathcal{R}_1 \otimes \mathcal{R}_2$

Note: $q_k^{'} = p_k^{d_k}, q_k = \phi(q_k^{'})$

Extended Amortized Aut

Packing Based on Tensor Rings

Pack

$$x_0, x_1, \dots, x_{u-1} \in \mathcal{R}, \text{mode } M \rightarrow \begin{cases} \sum_{i=0}^{u-1} x_i r_{0,i}, \text{if } M = 0 \\ \sum_{i=0}^{u-1} x_i r_{1,i}, \text{if } M = 1 \\ \sum_{i=0}^{u-1} x_i r_{0,i}^{\vee} r_{1,i}, \text{if } M = 2 \\ \sum_{i=0}^{u-1} x_i r_{0,i}^{\vee} r_{1,i}, \text{if } M = 3 \end{cases}$$

Add

$$\sum_{i=0}^{u-1} x_i r_{0,i} + \sum_{i=0}^{u-1} y_i r_{0,i} = \sum_{i=0}^{u-1} (x_i + y_i) r_{0,i}$$

Mult

$$\text{Tr}_{\tilde{K}/\tilde{K}_1}\left(r_{0,i}^{\vee} \cdot r_{0,j}\right) = \begin{cases} 0, \text{if } i \neq j \\ 1, \text{if } i = j \end{cases}$$

$$\text{Tr}_{\tilde{K}/\tilde{K}_1}\left(\left(\sum_{i=0}^{u-1} x_i r_{0,i}^{\vee} r_{1,i}\right) \cdot \left(\sum_{i=0}^{u-1} y_i r_{0,i}\right)\right) = \sum_{i=0}^{u-1} x_i y_i r_{1,i}$$

Note: $\tilde{K} = K \otimes K_0 \otimes K_1, \tilde{K}_1 = K \otimes K_1, \text{Tr}_{\tilde{K}/\tilde{K}_1}(x) = \sum_{\theta \in \text{Gal}(\tilde{K}/\tilde{K}_1)} \theta(x), u = \min_{k \in [3]} \{q_k\}$

Extended Amortized Aut

Amortized Aut

Aut

$$\mathrm{RLWE}_{z,\mathcal{R}}^{Q,\Delta}(\mu) = (a,b) \xrightarrow{\theta} \big(\theta(a),\theta(b)\big)$$
$$\xrightarrow{\mathrm{KS}} \Big(\big(0,\theta(b)\big) - \mathrm{REG}_{z,\mathcal{R}}^{Q}\big(\theta(z)\big) \boxtimes \theta(a)\Big) \bmod Q$$
$$= \mathrm{RLWE}_{z,\mathcal{R}}^{Q,\Delta}\big(\theta(\mu)\big)$$

Amortized Aut

$$\big\{\mathrm{RLWE}_{z,\mathcal{R}}^{Q,\Delta}(\mu_i) = (a_i,b_i)\big\}_{i\in[u]} \xrightarrow{\theta} \{\theta(a_i),\theta(b_i)\}_{i\in[u]}$$
$$\xrightarrow{\mathrm{Pack}} \alpha = \textstyle\sum_{i=0}^{u-1} \theta(a_i) r_{0,i}^{\vee} r_{1,i}\,,\ \beta = \sum_{i=0}^{u-1} \theta(b_i) r_{1,i}$$
$$\xrightarrow{\mathrm{KS}} \Big((0,\beta) - \mathrm{Trace}_{\tilde{K}/\tilde{K}_1}(K \boxtimes \alpha)\Big) \bmod Q$$
$$= \mathrm{RLWE}_{z,\tilde{\mathcal{R}}_1}^{Q,\Delta}\Big(\textstyle\sum_{i=0}^{u-1} \theta(\mu_i) r_{1,i}\Big)$$

Note: Automorphism $\theta\colon \mathcal{R} \to \mathcal{R}$, Aut key $K = \sum_{i=0}^{u-1} \mathrm{REG}_{z,\mathcal{R}}^{Q}\big(\theta(z_i)\big) r_{0,i}$

Extended Amortized Aut

Extended Amortized Aut

Computes different automorphisms

$$\left\{\mathrm{RLWE}_{z,\mathcal{R}}^{Q,\Delta}(\mu_i) = (a_i, b_i)\right\}_{i\in[u]} \xrightarrow{\{\theta_i\}_{i\in[u]}} \{\theta_i(a_i), \theta_i(b_i)\}_{i\in[u]}$$

$$\xrightarrow{\text{Pack}} \alpha = \sum_{i=0}^{u-1} \theta_i(a_i) r_{0,i}^{\vee} r_{1,i}\,, \beta = \sum_{i=0}^{u-1} \theta_i(b_i) r_{1,i}$$

$$\xrightarrow{\text{KS}} \left((0,\beta) - \mathrm{Trace}_{\widetilde{K}/\widetilde{K}_1}(K \boxtimes \alpha)\right) \bmod Q = \mathrm{RLWE}_{z,\tilde{\mathcal{R}}_1}^{Q,\Delta}\left(\sum_{i=0}^{u-1} \theta_i(\mu_i) r_{1,i}\right)$$

Note: Automorphism θ_i: $\mathcal{R} \to \mathcal{R}$, Aut key $K = \sum_{i=0}^{u-1} \mathrm{REG}_{z,\mathcal{R}}^{Q}\left(\theta_i(z_i)\right) r_{0,i}$

Extended Amortized Aut

Extended Amortized Aut

Computes different automorphisms

Supports more output modes

$$\left\{\mathrm{RLWE}_{z,\mathcal{R}}^{Q,\Delta}(\mu_i)=(a_i,b_i)\right\}_{i\in[u]} \xrightarrow{\{\theta_i\}_{i\in[u]}} \{\theta_i(a_i),\theta_i(b_i)\}_{i\in[u]}$$

$$\xrightarrow{\text{Pack}} \alpha=\sum_{i=0}^{u-1}\theta_i(a_i)r_{0,i}^{\vee}\cdot r_{2,i}\,,\beta=\sum_{i=0}^{u-1}\theta_i(b_i)r_{0,i}^{\vee}r_{1,i}\,,$$

$$\xrightarrow{\text{KS}} \left((0,\beta)-\mathrm{Trace}_{\tilde{K}_2/\tilde{K}}(K\boxtimes\alpha)\right) \bmod Q=\mathrm{RLWE}_{z,\tilde{\mathcal{R}}_1}^{Q,\Delta}\left(\sum_{i=0}^{u-1}\theta_i(\mu_i)r_{0,i}^{\vee}r_{1,i}\right)$$

Note: $\tilde{K}=K\otimes K_0\otimes K_1, \tilde{K}_2=K\otimes K_0\otimes K_1\otimes K_2$, Aut key $K=\sum_{i=0}^{u-1}\mathrm{REG}_{z,\mathcal{R}}^{Q}\left(\theta_i(z_i)\right)r_{1,i}\cdot r_{2,i}^{\vee}$

Improved HomINTT

NTT

Given $b = \sum_{i=0}^{N-1} b_i X^i \in \mathbb{Z}_q[X]/(X^N + 1)$

NTT $\boldsymbol{a} = (a_i)_{i\in[N]} = \mathrm{NTT}(b)$ where $a_i = \sum_{j=0}^{N-1} b_j \omega_{2N}^{(2i+1)j} \mod q$

INTT $c = \sum_{k=0}^{N-1} c_k X^i = \mathrm{INTT}(\boldsymbol{a})$ where $c_k = N^{-1} \sum_{i=0}^{N-1} a_i \omega_{2N}^{-(2i+1)k} \mod q$

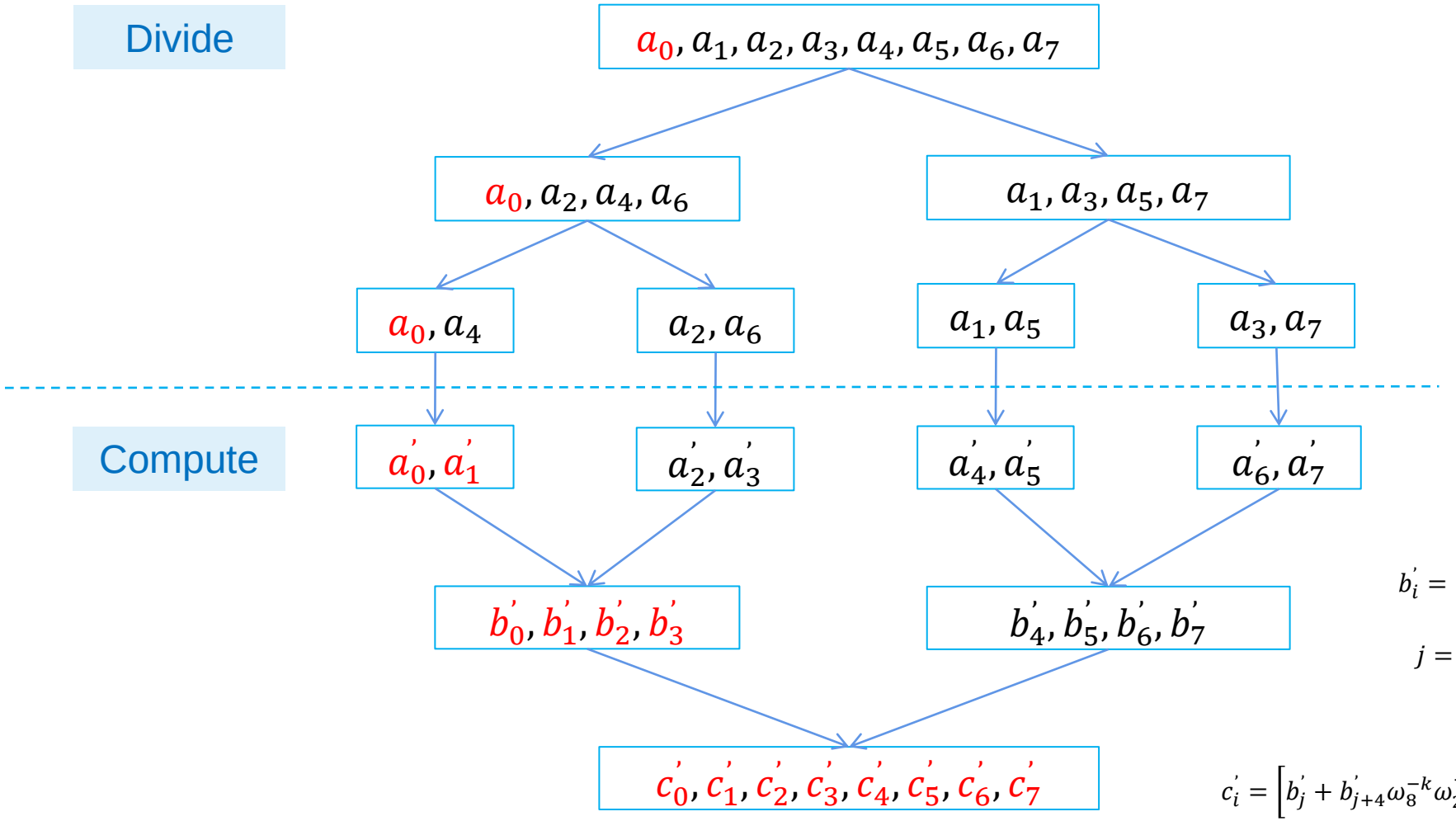
Note: N is a power of two, q is an odd prime, $q = 1 \pmod{2N}$

Improved HomINTT

Radix-r INTT E.g. $N = 2^3, r = 2$

Divide

Compute



$$a_i' = \left[a_j + a_{j+4}\omega_2^{-[i]_2^+}\right]_q$$
$$j = 2\left\lfloor\frac{[i]_4^+}{2}\right\rfloor + \left\lfloor\frac{i}{4}\right\rfloor$$

$$b_i' = \left[a_j' + a_{j+2}'\omega_4^{-k}\omega_2^{-[i]_2^+}\right]_q$$
$$j = k + 4\left\lfloor\frac{i}{4}\right\rfloor, k = \left\lfloor\frac{[i]_4^+}{2}\right\rfloor$$

$$c_i' = \left[b_j' + b_{j+4}'\omega_8^{-k}\omega_2^{-[i]_2^+}\right]_q, j = \left\lfloor\frac{i}{2}\right\rfloor$$

Note: Computational cost $= O(N \log N)$

Improved HomINTT

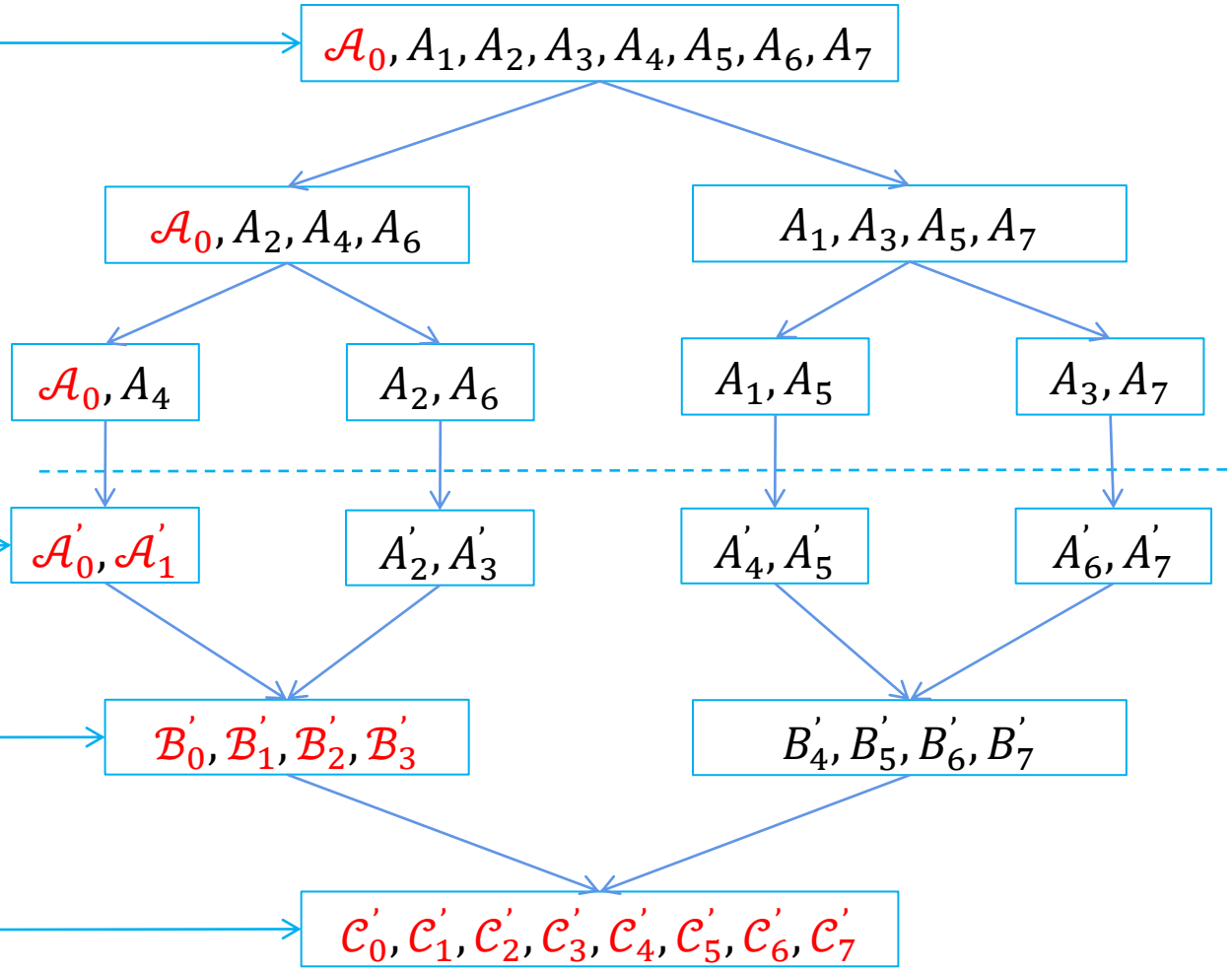
Homomorphic Radix-r INTT

$\mathcal{A}_0 = \mathrm{REG}^{Q}_{z,\mathcal{R}}(vX^{a_0}), A_i = \mathrm{REG}^{Q}_{z,\mathcal{R}}(X^{a_i})$

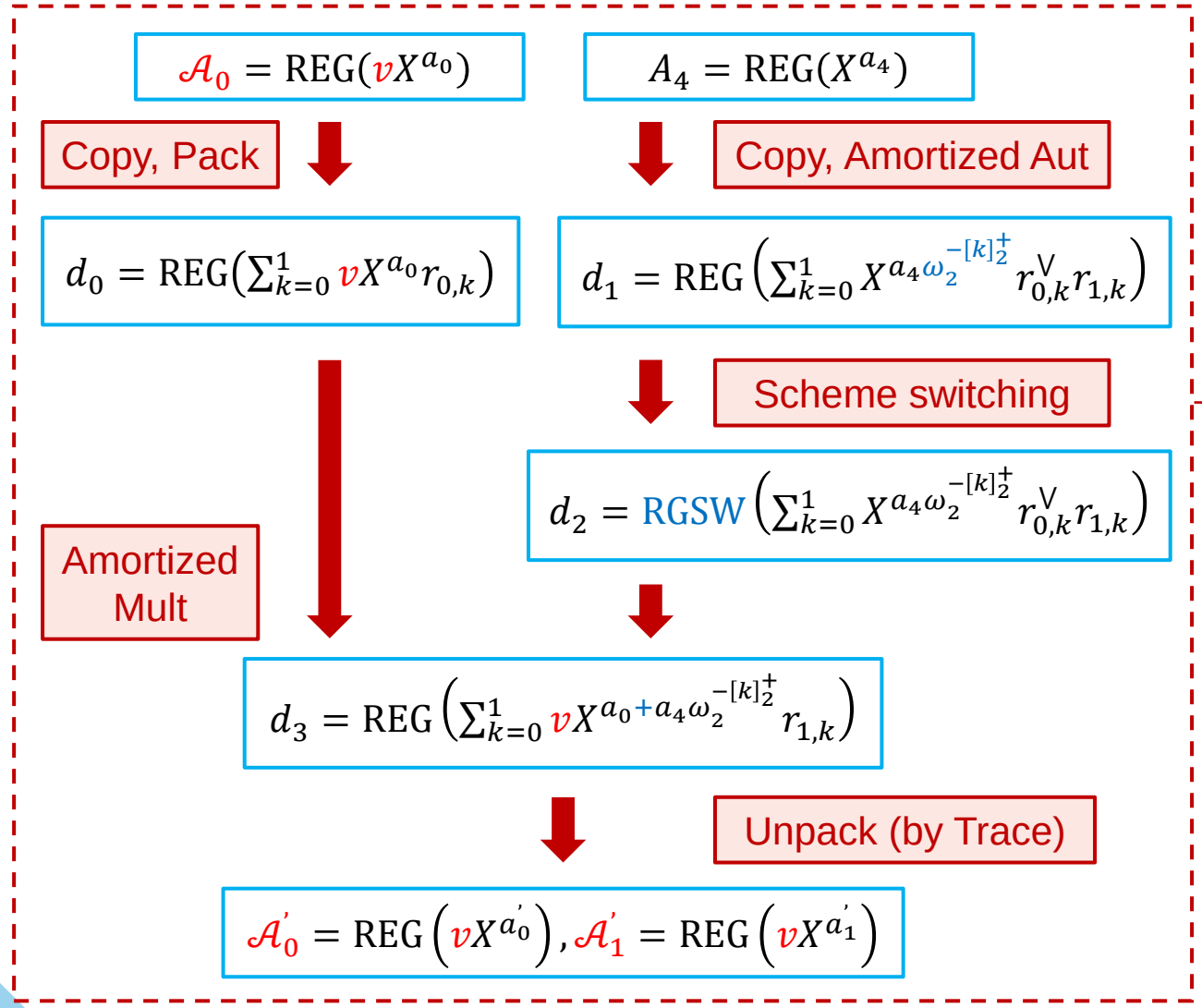
$\mathcal{A}_i^{'} = \mathrm{REG}^{Q}_{z,\mathcal{R}}\left(vX^{a_i^{'}}\right), A_i^{'} = \mathrm{REG}^{Q}_{z,\mathcal{R}}\left(X^{a_i^{'}}\right)$

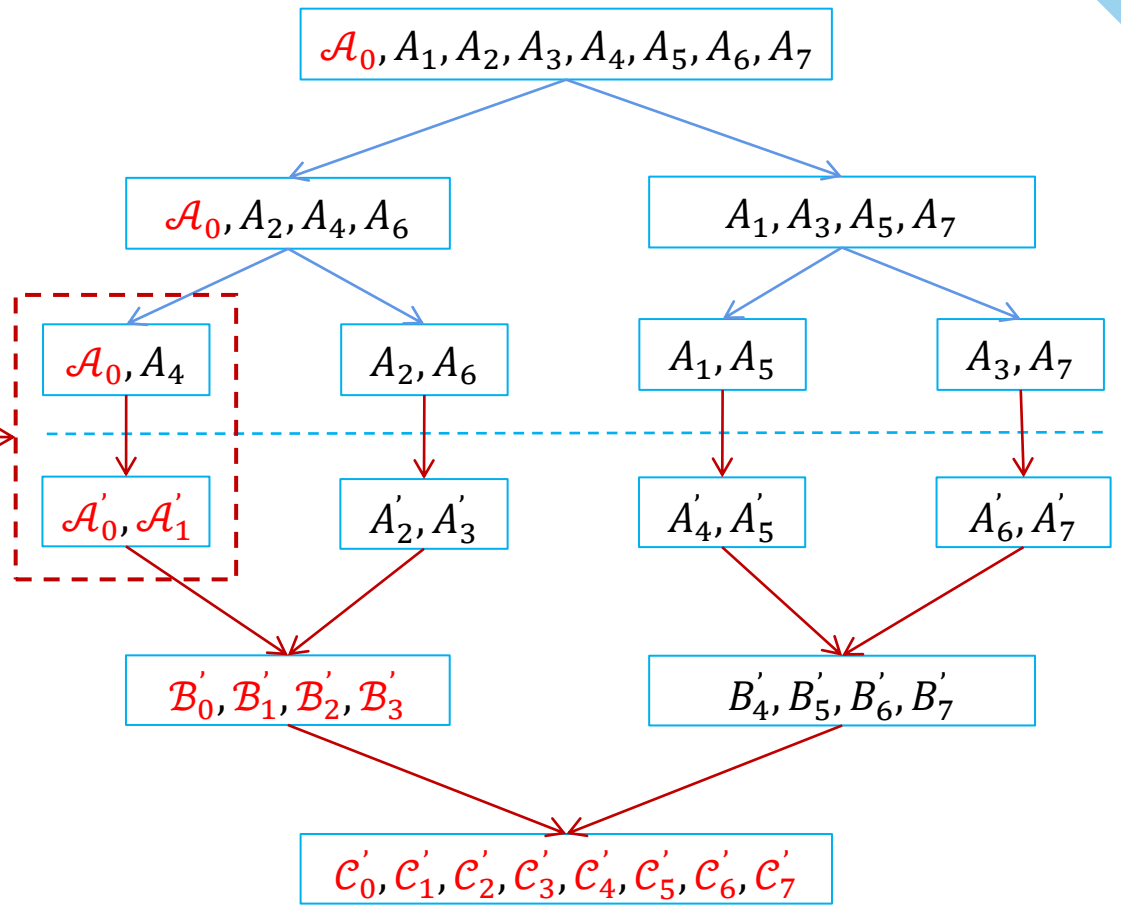
$\mathcal{B}_i^{'} = \mathrm{REG}^{Q}_{z,\mathcal{R}}\left(vX^{b_i^{'}}\right), B_i^{'} = \mathrm{REG}^{Q}_{z,\mathcal{R}}\left(X^{b_i^{'}}\right)$

$\mathcal{C}_i^{'} = \mathrm{REG}^{Q}_{z,\mathcal{R}}\left(vX^{c_i^{'}}\right)$



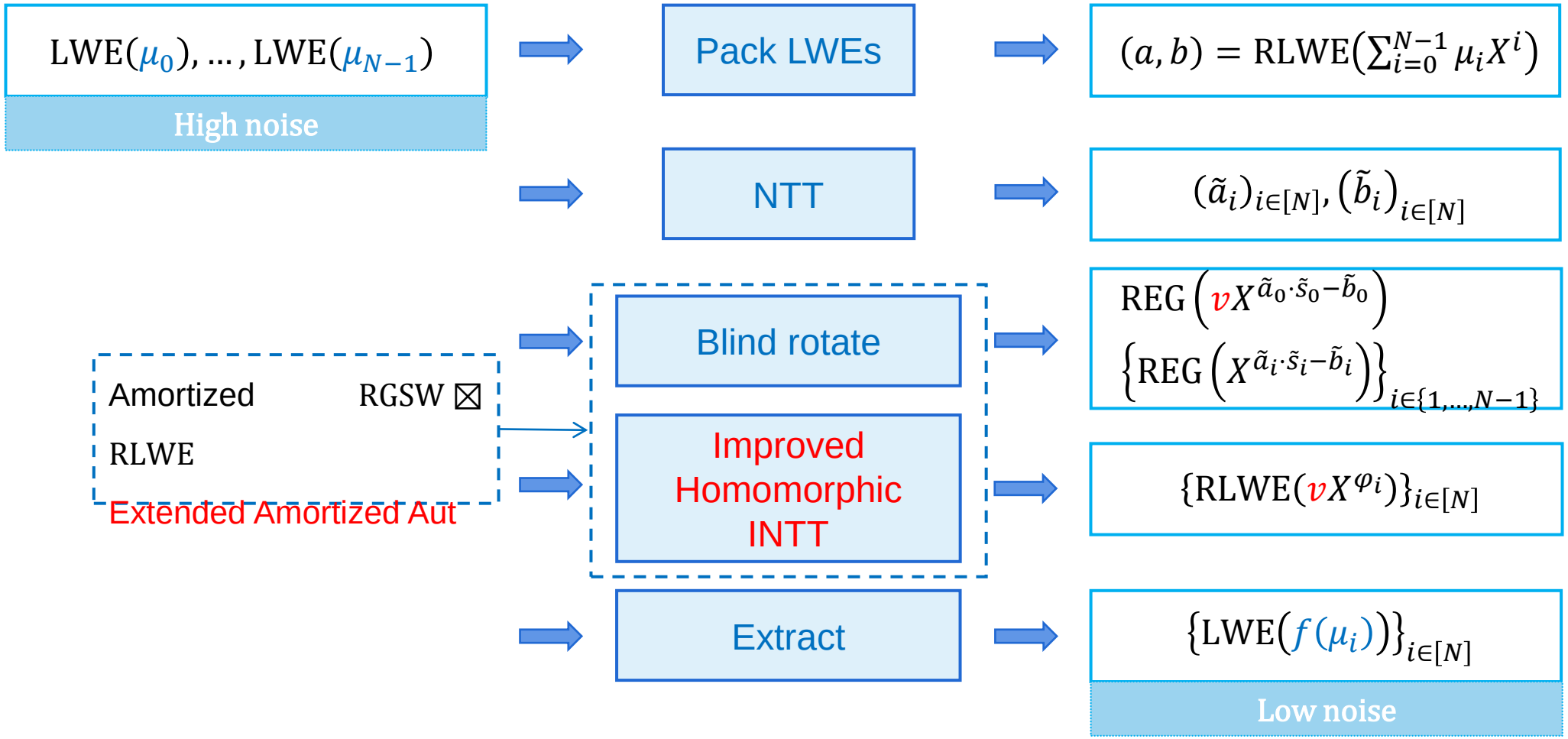
Improved HomINTT





Note: $a_0' = a_0 + a_4, a_1' = a_0 + a_4\omega_2^{-1}$

Amortized FBS for Encrypted Functions



Performance

Algorithm	Our work	[DM15, CGGI17]	[LMP22, MHW+24]	[MS18]	[GPL23, MKMS23]	[LW23a]	[LW23b]
Amortized cost	$\tilde{O}(1)$	$O(n)$	$O(n)$	$\tilde{O}(3^{\rho}n^{1/\rho})$	$\tilde{O}(\rho n^{1/\rho})$	$\tilde{O}(n^{0.75})$	$\tilde{O}(1)$
Multi-bit plaintext	√	×	√	×	√	×	×
Encrypted function	√	×	√	×	×	×	×

Algorithm	Our work	[LW23b]	[LW23c]
Function	Arbitrary	Boolean	Arbitrary
Sub-Gaussian parameter of error	$\tilde{O}(N^{9.375}E_{\mathrm{ks}})$	$\tilde{O}(N^{11.5}E_{\mathrm{ks}})$	$\tilde{O}(\lvert f(x)\rvert N^{38})$
Increasing factor	/	$\tilde{O}(N^{2.125})$	$\tilde{O}(\lvert f(x)\rvert N^{28.625})$

Conclusion

Our contributions

- Extended amortized Aut
  - Different automorphisms
  - More packing modes
- Improved HomINTT
  - Efficient
  - Allows inputting $\mathrm{REG}(vX^{\varphi})$
- Amortized FBS
  - Amortized cost = $\tilde{O}(1)$
  - Output errors are independent of f
  - Supports any public or encrypted function f

Future works

- Optimization
  - Key size
  - Computational costs
  - Output errors
- Implementation
  - Efficient
  - Practicable

Thank you!

Email address: xuyan1@iie.ac.cn

References

[Gen09] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009. pp. 169–178. ACM (2009)

[Bra12] Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical gapsvp. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012)

[FV12] Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, p. 144 (2012)

[BGV12] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. Innovations in Theoretical Computer Science 2012, PP. 309-325. ACM (2012)

[DM15] Ducas, L., Micciancio, D.: FHEW: bootstrapping homomorphic encryption in less than a second. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 617–640. Springer, Heidelberg (2015)

[CGGI17] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 377–408. Springer, Cham (2017)

[CIM19] Carpov, S., Izabachène, M., Mollimard, V.: New techniques for multi-value input homomorphic evaluation and applications. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 106–126. Springer, Cham (2019)

[BGGJ19] Boura, C., Gama, N., Georgieva, M., Jetchev, D.: Simulating homomorphic evaluation of deep learning predictions. In: Dolev, S., Hendler, D., Lodha, S., Yung, M. (eds.) CSCML 2019. LNCS, vol. 11527, pp. 212–230. Springer, Cham (2019)

[CJP21] Chillotti, I., Joye, M., Paillier, P.: Programmable bootstrapping enables efficient homomorphic inference of deep neural networks. In: Dolev, S., Margalit, O., Pinkas, B., Schwarzmann, A.A. (eds.) CSCML 2021. LNCS, vol. 12716, pp. 1–19. Springer, Cham (2021)

[CLOT21] Chillotti, I., Ligier, D., Orfila, J., Tap, S.: Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for TFHE. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021. LNCS, vol. 13092, pp. 670–699. Springer, Cham (2021)

[GBA21] Guimarães, A., Borin, E., Aranha, D.F.: Revisiting the functional bootstrap in TFHE. IACR Transactions on Cryptographic Hardware and Embedded Systems 2021(2), 229–253 (2021)

References

[LMP22] Liu, Z., Micciancio, D., Polyakov, Y.: Large-precision homomorphic sign evaluation using FHEW/TFHE bootstrapping. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022. LNCS, vol. 13792, pp. 130–160. Springer, Cham (2022)

[KS23] Kluczniak, K., Schild, L.: FDFB: full domain functional bootstrapping towards practical fully homomorphic encryption. IACR Transactions on Cryptographic Hardware and Embedded Systems 2023(1), 501–537 (2023)

[MHW+24] Ma, S., Huang, T., Wang, A., Zhou, Q., Wang, X.: Fast and accurate: Efficient full-domain functional bootstrap and digit decomposition for homomorphic computation. IACR Transactions on Cryptographic Hardware and Embedded Systems 2024(1), 592–616 (2024)

[MS18] Micciancio, D., Sorrell, J.: Ring packing and amortized FHEW bootstrapping. In: 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018. LIPIcs, vol. 107, pp. 100:1–100:14. Schloss Dagstuhl - Leibniz Zentrum für Informatik (2018)

[GPL23] Guimarães, A., Pereira, H.V.L., Leeuwen, B.V.: Amortized bootstrapping revisited: Simpler, asymptotically-faster, implemented. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023. LNCS, vol. 14443, pp. 3–35. Springer, Singapore (2023)

[MKMS23] Micheli, G.D., Kim, D., Micciancio, D., Suhl, A.: Faster amortized FHEW bootstrapping using ring automorphisms. IACR Cryptology ePrint Archive, p. 112 (2023)

[LW23a] Liu, F., Wang, H.: Batch bootstrapping I: - A new framework for SIMD bootstrapping in polynomial modulus. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023. LNCS, vol. 14006, pp. 321–352. Springer, Cham (2023)

[LW23b] Liu, F., Wang, H.: Batch bootstrapping II: - bootstrapping in polynomial modulus only requires o(1) FHE multiplications in amortization. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023. LNCS, vol. 14006, pp. 353–384. Springer, Cham (2023)

[LW23c] Liu, Z., Wang, Y.: Amortized functional bootstrapping in less than 7 ms, with õ(1) polynomial multiplications. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023. LNCS, vol. 14443, pp. 101–132. Springer, Singapore (2023)