Simpler and Faster BFV Bootstrapping for Arbitrary Plaintext Modulus from CKKS

Jaehyung Kim[1] **Jinyeong Seo**[2] Yongsoo Song[2]

[1]CryptoLab Inc.
[2]Seoul National University

Homomorphic Encryption

Crypto schemes that enable computation in encrypted state.

- **BGV & BFV**: Support modular operations on integers.
- **CKKS**: Supports approximate operations on complex numbers.
- **TFHE**: Supports bitwise operations.

Common issue: the number of homomoprhic operations is bounded.

Bootstrapping

Special operations that refresh the number of remaining operations.

- **BGV & BFV**: Digit extraction [HS15]
- **CKKS**: Approximate modular reduction [Che+18]
- **TFHE**: Blind rotation [Chi+16]

Motivation

The known BFV bootstrapping method so far has some limitations.

- The efficiency highly depends on the plaintext modulus.
- Only a specific form of plaintext modulus is usable.

This Work

We design a novel BFV bootstrapping method.

- Utilizes CKKS bootstrapping as a subroutine.
- Supports arbitrary plaintext modulus.
- Provides flexible performance depending on the bootstrapping quality.

Overview

1. Review of BFV bootstrapping
2. Review of CKKS bootstrapping
3. Our new BFV bootstrapping
4. Experimental results

Notations

- $\Phi_M(X)$: the M-th cyclotomic polynomial
- $R = \mathbb{Z}[X]/\Phi_M(X)$
- q: ciphertext modulus
- t: plaintext modulus
- $R_q = R/qR$, a quotient ring of R

BFV Bootstrapping

BFV

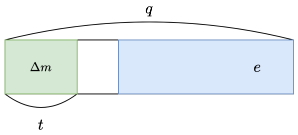
Plaintext Space. Supports homomorphic operations over $R_t = R/tR$

- With the packing method, it supports SIMD arithmetics over $\mathbb{Z}_t$.

Ciphertext Structure. A BFV ciphertext $\mathsf{ct} = (c_0, c_1) \in R_q^2$ satisfies

$$c_0 + c_1 \cdot s = \Delta m + e \pmod{q}$$

where $m \in R_t$, $\Delta = \lfloor q/t \rceil$ and $e \in R$ with $\|e\|_\infty < \Delta/2$.

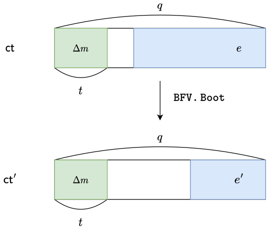


BFV Bootstrapping

After each homomorphic operation, the size of noise $\|e\|_\infty$ grows.

- For further operations, we need to decrease $\|e\|_\infty$ while preserving m

$\underline{\texttt{BFV.Boot}(\mathrm{ct}) \to \mathrm{ct}'}$: Given a ciphertext $\mathrm{ct} = (c_0, c_1) \in R_q^2$ with $c_0 + c_1 \cdot s = \Delta m + e \pmod q$, it outputs a ciphertext $\mathrm{ct}' = (c_0', c_1') \in R_q^2$ with $c_0' + c_1' \cdot s = \Delta m + e' \pmod q$ where $\|e'\|_\infty \ll \|e\|_\infty$.

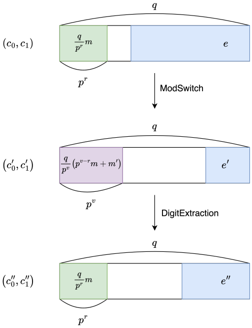


- Its functionality can be parametrized as $(q, t, B_{\text{in}}, B_{\text{out}})$ where $\|e\|_\infty < B_{\text{in}}$ and $\|e'\|_\infty < B_{\text{out}}$.
- We call the ratio $B_{\text{in}}/B_{\text{out}}$ the denoising factor.

Digit Extraction

The previous framework for the BFV bootstrapping [HS15].

- Plaintext modulus is set to $t = p^r$ for some prime p.



Key step: homomorphic evaluation of the map $x \mapsto \lfloor x/p^r \rceil \pmod{p^v}$.

- Recent studies [CH18; Gee+23; OPP23] focused on improving this.
- Provides optimal performance when p is **a small prime**.

CKKS Bootstrapping

CKKS

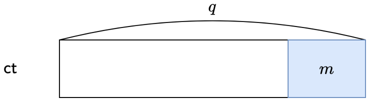
Plaintext Space. Supports approximate homomorphic operations over R

- With the packing method, it supports approx SIMD operations over $\mathbb{C}$.

Ciphertext Structure. A CKKS ciphertext $(c_0, c_1) \in R_q^2$ satisfies the followings for $m \in R$.

$$c_0 + c_1 \cdot s = m \pmod q$$

Note: There is no strict distinction between plaintext and noise in CKKS.

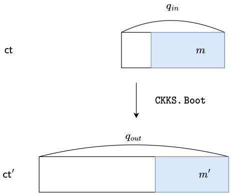


CKKS Bootstrapping

After each homomorphic operation, the ciphertext modulus q decreases.

- For further operations, we need to increase q while almost preserving m

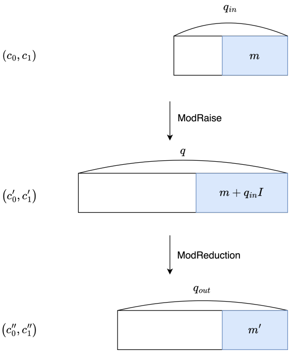
$\underline{\texttt{CKKS.Boot}(\mathsf{ct}) \to \mathsf{ct}'}$: Given a ciphertext $\mathsf{ct} = (c_0, c_1) \in R_{q_{\mathsf{in}}}^2$ with $c_0 + c_1 \cdot s = m \pmod{q_{\mathsf{in}}}$, it outputs a ciphertext $\mathsf{ct}' = (c_0', c_1') \in R_{q_{\mathsf{out}}}^2$ with $c_0' + c_1' \cdot s = m' \pmod{q_{\mathsf{out}}}$ such that $m' \approx m$ and $q_{\mathsf{out}} \gg q_{\mathsf{in}}$.



- Its functionality can be parametrized as $(q_{\mathsf{in}}, q_{\mathsf{out}}, B_{\mathsf{in}}, B_{\mathsf{out}})$ where $\|m\|_\infty < B_{\mathsf{in}}$ and $\|m - m'\|_\infty < B_{\mathsf{out}}$.
- We call the ratio $B_{\mathsf{in}}/B_{\mathsf{out}}$ the precision.

Approximate Modular Reduction

The basic framework for the CKKS bootstrapping [Che+18].



Key step: approximate evaluation of the map $x \mapsto [x]_{q_{\text{in}}}$.

- Recent studies [JM22; Lee+21; Lee+22] focused on improving this.
- The performance depends on the **precision**.

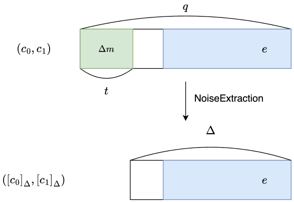
Our Method

Noise Extraction

Key observation: the noise e can be extracted if $\Delta | q$ (i.e., $t | q$).

$$c_0 + c_1 \cdot s = \Delta m + e \pmod{q}$$

$$[c_0]_\Delta + [c_1]_\Delta \cdot s = e \pmod{\Delta}$$



Note: the ciphertext modulus decreases to Δ.

Approximate Lifting

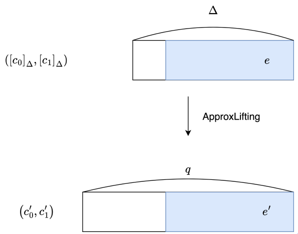
We can regard $([c_0]_\Delta, [c_1]_\Delta)$ as a **CKKS ciphertext** encrypting the noise e.

Idea: we can apply CKKS.Boot of functionality $(\Delta, q, B_{\text{in}}, B_{\text{out}})$ on $([c_0]_\Delta, [c_1]_\Delta)$ to raise the ciphertext modulus from Δ to q.

For $(c_0', c_1') \leftarrow \texttt{CKKS.Boot}([c_0]_\Delta, [c_1]_\Delta)$, it holds that:

$$c_0' + c_1' \cdot s = e' \approx e \pmod{q}$$

where $\|e\|_\infty < B_{\text{in}}$ and $\|e - e'\|_\infty < B_{\text{out}}$.



Subtraction.

Finally, subtracting (c'_0, c'_1) from (c_0, c_1) finishes BFV bootstrapping of functionality $(q, t, B_{\text{in}}, B_{\text{out}})$

$$(c_0 - c'_0) + (c_1 - c'_1) \cdot s = \Delta m + (e - e') \pmod{q}$$

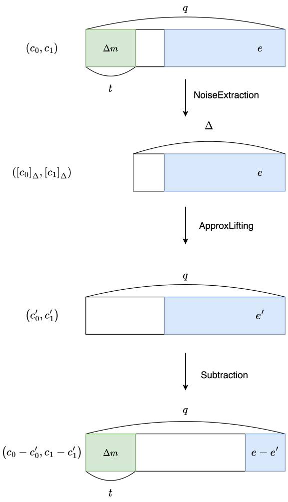
<u>Note</u>: the noise decrease from $\|e\|_\infty < B_{\text{in}}$ to $\|e - e'\|_\infty < B_{\text{out}}$.

Theorem

Let $\Delta | q$. Given a `CKKS.Boot` *with functionality $(\Delta, q, B_{in}, B_{out})$, one can instantiate* `BFV.Boot` *of with functionality (q, t, B_{in}, B_{out}).*

<u>Note</u>: the precision of `CKKS.Boot` directly translate to the denoising factor of `BFV.Boot`

Overall Pipeline



META-BTS

Issue: We need a high-precision CKKS bootstrapping.

- Ordinary CKKS bootstrapping only supports **dozens of bits** precision.
- In BFV bootstrapping, the denoising factor is usually **hundreds of bits**.

Solution: We employ META-BTS method [Bae+22] for `CKKS.Boot`.

- It provides **arbitrary precision** CKKS bootstrapping by iterating low-precision CKKS boostrapping.
  - To attain kn-bits precision, it iterates n-bits CKKS bootstrapping k times.
  - Precision can be easily adjusted by modifying the iteration count k.
- It provides **asymptotically faster** time complexity in achieving high-precision CKKS boostrapping.

Unexpected feature: We can adjust the denoising factor in `BFV.Boot` by changing the iteration count in `CKKS.Boot`.

Experimental Results

Experimental Results

Performance comparison with digit extraction

- We compare the performance with the state-of-the-art digit extraction method [Gee+23] for 51-bits sized plaintext modulus
- Denoising factors are set to 121-bits.
- [Gee+23] sets $t = 2^{51}$, a power of small prime, for the efficiency.

Table: BFV bootstrapping functionality used in the benchmark

	q	t	B_{in}	B_{out}
[Gee+23]	1200 bits	51 bits	1137 bits	1006 bits
Ours	1200 bits	51 bits	1077 bits	949 bits

Table: Full bootstrapping performance.

	Plaintext modulus	Ring dimension	Boot time (sec)	Amortized boot time (ms/coeff)
[Gee+23]	2^{51}	42336	**1344+**	**31.7+**
Ours	$\approx 2^{51}$	32768	**35.5**	**1.08**

Experimental Results

Effect of plaintext modulus in our method

Table: Bootstrapping performance for various plaintext moduli.

q	t	B_{in}	B_{out}	Boot time
791 bits	**54** bits	376 bits	16 bits	**392** sec
	144 bits			
	234 bits			

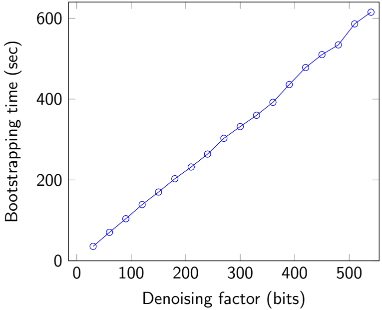
The bootstrapping times are identical since it internally runs the same CKKS bootstrapping.

- The precision of the underlying CKKS bootstrappings is identical.

Experimental Results

Effect of denoising factor in our method

Figure: Bootstrapping time with respect to the denoising factor.



Bootstrapping time grows linearly with denoising factor.

- The iteration count in META-BTS grows linearly with precision.

Conclusion

We design a novel BFV bootstrapping method

- It does not inherit previous limitations of digit extraction.
- One can freely set plaintext modulus as one's need.
- It can adjust bootstrapping quality depending on situations.

We incorporate CKKS bootstrapping as a subroutine.

- Performance advance in CKKS bootstrapping leads to BFV boostrapping.
- Hardware acceleration of CKKS bootstrapping is directly applicable.

Thank you !

https://eprint.iacr.org/2024/109

References I

[Bae+22] Youngjin Bae et al. “Meta-bts: Bootstrapping precision beyond the limit”. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2022, pp. 223–234 (cit. on p. 19).

[CH18] Hao Chen and Kyoohyung Han. “Homomorphic lower digits removal and improved FHE bootstrapping”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2018, pp. 315–337 (cit. on p. 9).

[Che+18] Jung Hee Cheon et al. “A full RNS variant of approximate homomorphic encryption”. In: *International Conference on Selected Areas in Cryptography*. Springer. 2018, pp. 347–368 (cit. on pp. 2, 13).

References II

[Chi+16] Ilaria Chillotti et al. "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds". In: *international conference on the theory and application of cryptology and information security*. Springer. 2016, pp. 3–33 (cit. on p. 2).

[Gee+23] Robin Geelen et al. "On polynomial functions modulo pe and faster bootstrapping for homomorphic encryption". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 257–286 (cit. on pp. 9, 21).

[HS15] Shai Halevi and Victor Shoup. "Bootstrapping for HElib". In: *Advances in Cryptology–EUROCRYPT 2015* (2015), pp. 641–670 (cit. on pp. 2, 9).

References III

[JM22] Charanjit S Jutla and Nathan Manohar. "Sine series approximation of the mod function for bootstrapping of approximate HE". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2022, pp. 491–520 (cit. on p. 13).

[Lee+21] Joon-Woo Lee et al. "High-precision bootstrapping of RNS-CKKS homomorphic encryption using optimal minimax polynomial approximation and inverse sine function". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2021, pp. 618–647 (cit. on p. 13).

References IV

[Lee+22] Yongwoo Lee et al. "High-Precision Bootstrapping for Approximate Homomorphic Encryption by Error Variance Minimization". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2022, pp. 551–580 (cit. on p. 13).

[OPP23] Hiroki Okada, Rachel Player, and Simon Pohmann. "Homomorphic polynomial evaluation using Galois structure and applications to BFV bootstrapping". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2023, pp. 69–100 (cit. on p. 9).