Reducing Bootstrapping Cost by One Level

Eric Crockett

May 13, 2025

1 Introduction

CKKS homomorphic encryption requires *rescaling* a ciphertext after performing a multiplication. This effectively reduces the scale factor on the message from Δ^2 to Δ, but "costs" a factor of Δ in the modulus (or equivalently, reduces the ciphertext *level* from ℓ to $\ell - 1$, or equivalently, removes on RNS modulus). You can only do this so many times before the modulus becomes too small to do further multiplications/rescaling (i.e., you reach level 0, or have only one RNS modulus remaining). Bootstrapping is a process to raise the modulus to a much larger value, enabling more multiplications and rescalings.

[KPP20] introduces a proposal to reduce the approximation error in CKKS by rescaling *before* multiplications rather than after.

In the bootstrapping procedure, the first step (ModRaise) is to raise the modulus of the ciphertext to Q_L, the modulus of a fresh encryption. This produces a ciphertext with the correct modulus, but wrong message. The rest of the steps of bootstrapping restore the correct message. Raising the modulus doesn't require any "secret" information; the evaluator can do it without any assistance.

The next step in bootstrapping is to apply a linear transformation. This step is known as `CoeffsToSlots`. There are many different techniques for achieving this step, but at least some of them involve immediately rescaling the output of the ModRaise step (e.g., to control noise).

However, this is wasteful: for a given parameter set, a fresh encryption (and ModRaise) output a level L ciphertext. In bootstrapping, however, we raise the ciphertext to level L, then immediately scale down to level $L - 1$.

We can avoid this by introducing a special "bootstrapping modulus" p_b. In the ModRaise step, we now raise the ciphertext modulus $Q_L \cdot p_b$. Since this step doesn't involve any secret information,

- the evaluator is free to choose p_b on its own; this doesn't require any coordination with the encryptor
- this doesn't affect security.

Rescaling *also* doesn't require any secret information, so the evaluator can rescale from level $L + 1$ to level L after ModRaise, and then start ModRaise at one level higher than before. As a result, the output of bootstrapping is also one level higher, which effectively reduces the depth of bootstrapping by one. The depth of bootstrapping is directly related to the efficiency of the overall scheme; we can now do one additional multiplication before bootstrapping again.

2 Security

Typically, adding additional moduli would reduce security because there also need to be key-switch keys for the same (larger) modulus. However, that only applies if the encryptor creates a ciphertext with more moduli. In this case, the evaluator is doing a "public" operation (i.e., one that doesn't require the secret key), so by definition this can't reduce security.

References

[KPP20] Andrey Kim, Antonis Papadimitriou, and Yuriy Polyakov. *Approximate Homomorphic Encryption with Reduced Approximation Error*. Cryptology ePrint Archive, Paper 2020/1118. 2020. URL: `https://eprint.iacr.org/2020/1118`.