CKKS Lecture Notes

Part 3: Bootstrapping

FEI

October 2026

1 Bootstrapping

Recall that when we multiply ciphertexts, the result has a lower level than the inputs. Ciphertexts at level 0 cannot be multiplied.

When a ciphertext reaches level 0, we can increase its level by *bootstrapping*. Bootstrapping is an operation that takes in a level-0 ciphertext and produces a higher-level ciphertext encoding the same message.

Let (a, b) be a level-0 ciphertext encoding $f \in \mathbb{R}[X]/(X^N + 1)$.

Let

$$f(X) = r_0 + r_1 X + \cdots + r_{N-1} X^{N-1}.$$

Then, the slots of the ciphertext (a, b) encode

$$\left(f(\zeta), f(\zeta^5), \ldots, f(\zeta^{5^{N/2-1}})\right) = (c_0, c_1, \ldots, c_{N/2-1}) \in \mathbb{C}^{N/2}.$$

Recall that the operations (addition, multiplication, rotation, conjugation) apply to the slots of the ciphertext.

For some small error e,

$$as + b \equiv \mathsf{round}(\Delta_0 f) + e \pmod{Q_0}.$$

Lift a and b to a higher modulus Q. Then,

$$as + b \equiv \mathsf{round}(\Delta_0 f) + e + q_0 I \pmod{Q}$$

for some integer polynomial I.

This encodes

$$f + \frac{q_0}{\Delta_0} I.$$

The main idea of bootstrapping is to perform a sequence of operations that removes the $\frac{q_0}{\Delta_0} I$ term.

2 Extending the modulus chain

The computation to remove $\frac{q_0}{\Delta_0} I$ takes about 15 levels. So, we extend the modulus chain:

$$Q_{n'} = q_0 \dots q_n q_{n+1} \dots q_{n'}.$$

Bootstrapping is prone to error, so it needs to be done at a higher level of precision. Thus,

$$q_{n+1} \approx \dots \approx q_{n'} \approx \Delta_{n+1} \approx \dots \approx \Delta_{n'} \gg q_1 \approx \dots \approx q_n \approx \Delta_0 \approx \dots \approx \Delta_n.$$

Note that $\frac{\Delta_{n+1}^2}{q_{n+1}} \neq \Delta_n$ and $\Delta_{n'} \neq \Delta_0$.

Sidenote about security

Higher modulus Q is less secure and higher ring dimension N is more secure. Therefore, the higher N is, the higher we can make Q without breaking security. $N = 65536$ is the lowest ring dimension where Q can be high enough to fit the bootstrapping levels.

3 Sparse secret

This computation is easier when I has small coefficients. So, instead of directly lifting (a, b) to a higher modulus as described in the first section, we introduce an extra step that makes I smaller.

Since lower modulus is more secure, at level 0, we can switch to a *sparse secret* without losing security. The sparse secret is a secret $s' \in \mathbb{Z}[X]/(X^N + 1)$ such that s' has only 32 nonzero coefficients, each of which is ± 1.

The person with the secret key reveals a dense-to-sparse keyswitching key $\mathsf{ksk}_{s \to s'}$ at level 0 and a sparse-to-dense keyswitching key $\mathsf{ksk}_{s' \to s} = \mathsf{ksk}_{s'}$ at level n'.

Let (a', b') be the result of switching (a, b) from the dense key to the sparse key at level 0. Then,

$$a's' + b' \equiv \mathsf{round}(\Delta_0 f) + q_0 I + e \pmod{Q_{n'}}.$$

Each coefficient of $a's' + b'$ is the combination of 32 coefficients of a' and a coefficient of b'. The coefficients of a' and b' lie in $[-q_0/2, q_0/2)$. Therefore, the coefficients of $a's' + b' = \mathsf{round}(\Delta_0 f) + q_0 I \pmod{Q_{n'}}$ lie in $[-33q_0/2, 33q_0/2)$. Therefore, the coefficients of I lie in $[-16, 16]$.

At level n', we keyswitch (a', b') back to the dense secret. This produces a ciphertext encoding $f + \frac{q_0}{\Delta_0} I$ at level n' under the dense key, which we can then do operations on.

Let

$$I = j_0 + j_1 X + \dots + j_{N-1} X^{N-1}.$$

Then,

$$f + \frac{q_0}{\Delta_0} I = (r_0 + \frac{q_0}{\Delta_0} j_0) + (r_1 + \frac{q_0}{\Delta_0} j_1)X + \cdots + (r_{N-1} + \frac{q_0}{\Delta_0} j_{N-1})X^{N-1}.$$

Removing $\frac{q_0}{\Delta_0} I$ involves the following steps:

- **Coefficients to slots.** Compute ciphertexts whose slots contain $r_i + \frac{q_0}{\Delta_0} j_i$.
- **Mod reduction.** Apply a polynomial to these ciphertexts that approximately reduces them modulo $\frac{q_0}{\Delta_0}$. This produces ciphertexts whose slots contain the r_i's.
- **Slots to coefficients.** Use the ciphertexts whose slots contain the r_i's to compute a ciphertext encoding the polynomial

  $$r_0 + r_1 X + \cdots + r_{N-1} X^{N-1} = f \in \mathbb{R}[X]/(X^N + 1).$$

  Slots to coefficients and coefficients to slots are both linear transformations.

4 Slots to coefficients matrix

We have

$$\begin{bmatrix} 1 & \zeta^{5^0} & \cdots & \zeta^{(N-1)5^0} \\ 1 & \zeta^{5^1} & \cdots & \zeta^{(N-1)5^1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{5^{N/2-1}} & \cdots & \zeta^{(N-1)5^{N/2-1}} \end{bmatrix} \begin{bmatrix} r_0 \\ \vdots \\ r_{N-1} \end{bmatrix} = \begin{bmatrix} c_0 \\ \vdots \\ c_{N/2-1} \end{bmatrix}.$$

Since there are N coefficients and $N/2$ slots, this is a $N/2 \times N$ matrix. The right half is the left half multiplied by i, so we can write it as a $N/2 \times N/2$ matrix:

$$\begin{bmatrix} 1 & \zeta^{5^0} & \cdots & \zeta^{(N/2-1)5^0} \\ 1 & \zeta^{5^1} & \cdots & \zeta^{(N/2-1)5^1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{5^{N/2-1}} & \cdots & \zeta^{(N/2-1)5^{N/2-1}} \end{bmatrix} \begin{bmatrix} r_0 + r_{N/2} i \\ \vdots \\ r_{N/2-1} + r_{N-1} i \end{bmatrix} = \begin{bmatrix} c_0 \\ \vdots \\ c_{N/2-1} \end{bmatrix}.$$

5 Coefficients to slots matrix

The coefficients-to-slots matrix is the inverse of the slots-to-coefficients matrix. The columns of the slots-to-coefficients matrix are orthogonal and have squared norm $N/2$. So, the inverse of the slots-to-coefficients matrix is its conjugate transpose scaled by $\frac{1}{N/2}$:

$$\frac{1}{N/2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \zeta^{-5^0} & \zeta^{-5^1} & \cdots & \zeta^{-5^{N/2-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta^{-(N/2-1)5^0} & \zeta^{-(N/2-1)5^1} & \cdots & \zeta^{-(N/2-1)5^{N/2-1}} \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_{N/2-1} \end{bmatrix} = \begin{bmatrix} r_0 + r_{N/2} i \\ \vdots \\ r_{N/2-1} + r_{N-1} i \end{bmatrix}.$$

6 Matrix multiplication

Under CKKS, we can efficiently multiply a ciphertext by a diagonal-sparse matrix, i.e. a matrix with few nonzero diagonals.

Let M be a diagonal-sparse matrix. Assume that diagonals $d_1, \ldots, d_k$ of M are nonzero.

Diagonal-sparse matrix multiplication

Let c be a ciphertext whose slots encode an $N/2$-dimensional complex vector $\mathbf{v}$.

Then, the ciphertext given by

$$\begin{aligned} &\mathsf{rotate}_{d_1}(c) \odot \begin{bmatrix} M_{1,1+d_1} & \ldots & M_{N/2,N/2+d_1} \end{bmatrix} \\ &+ \mathsf{rotate}_{d_2}(c) \odot \begin{bmatrix} M_{1,1+d_2} & \ldots & M_{N/2,N/2+d_2} \end{bmatrix} \\ &\vdots \\ &+ \mathsf{rotate}_{d_k}(c) \odot \begin{bmatrix} M_{1,1+d_k} & \ldots & M_{N/2,N/2+d_k} \end{bmatrix} \end{aligned}$$

encodes $M\mathbf{v}$.

If the positions of the nonzero diagonals are in an arithmetic progression, then we can do even better. For simplicity, assume that M has nonzero diagonals at positions $0, \ldots, d^2 - 1$.

Baby-step giant-step

Construct matrices $M^{(0)}, \ldots, M^{(d-1)}$ such that:

- Each $M^{(i)}$ has nonzero diagonals at positions $0, \ldots, d-1$.
- The jth diagonal of $M^{(i)}$ is equal to the $di+j$th diagonal of M, cyclically shifted by di positions. Equivalently,

$$M^{(i)}_{k,k+j} = M_{k-di,k+j}.$$

Then, compute rotations $\mathsf{rotate}_0(c), \ldots, \mathsf{rotate}_{d-1}(c)$. Use these to compute ciphertexts encoding $M^{(0)}\mathbf{v}, \ldots, M^{(d-1)}\mathbf{v}$. Then, $M\mathbf{v} = \sum_i \mathsf{rotate}_{di}(M^{(i)}\mathbf{v})$.

Rotations take a keyswitch and plaintext-ciphertext multiplication does not, so rotations are the main cost of diagonal-sparse matrix multiplication. With baby-step giant-step, we only need $O(d)$ rotations to apply a matrix with d^2 nonzero diagonals.

7 Matrix factorization

By writing the slots-to-coefficients and coefficients-to-slots matrices in a different order, we can split them into diagonal-sparse factors.

Let $\mathsf{br}(n)$ denote the integer obtained by writing n in binary with $\log_2(N)-1$ digits (possibly with leading zeroes) and reversing the digits.

The bit-reversed coefficients-to-slots and slots-to-coefficients matrices are the following:

$$\begin{bmatrix} 1 & \zeta^{\mathsf{br}(1)5^0} & \cdots & \zeta^{\mathsf{br}(N/2-1)5^0} \\ 1 & \zeta^{\mathsf{br}(1)5^1} & \cdots & \zeta^{\mathsf{br}(N/2-1)5^1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{\mathsf{br}(1)5^{N/2-1}} & \cdots & \zeta^{\mathsf{br}(N/2-1)5^{N/2-1}} \end{bmatrix}$$

$$\frac{1}{N/2}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \zeta^{-\mathsf{br}(1)5^0} & \zeta^{-\mathsf{br}(1)5^1} & \cdots & \zeta^{-\mathsf{br}(1)5^{N/2-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta^{-\mathsf{br}(N/2-1)5^0} & \zeta^{-\mathsf{br}(N/2-1)5^1} & \cdots & \zeta^{-\mathsf{br}(N/2-1)5^{N/2-1}} \end{bmatrix}$$

Each of these can be split into $\log_2(N/2)$ factors.

Slots-to-coefficients factorization

Start with

$$\begin{bmatrix} r_{\mathsf{br}(0)} + r_{\mathsf{br}(0)+N/2^i} \\ \vdots \\ r_{\mathsf{br}(N/2-1)} + r_{\mathsf{br}(N/2-1)+N/2^i} \end{bmatrix}.$$

At step k (starting with 0), for each j with $\lfloor j/2^k \rfloor$ even, apply the 2×2 matrix

$$\begin{bmatrix} 1 & \zeta^{5^j \cdot (N/2^{k+2})} \\ 1 & \zeta^{5^{j+2^k} \cdot (N/2^{k+2})} \end{bmatrix}$$

to the 2-dimensional vector formed by the jth and $j+2^k$th slots of the current vector.

Each matrix multiplication takes 1 level. To decrease the level consumption, we combine groups of steps into matrices. If we use three equally-sized groups of factors, the factors have 63, 63, and 32 nonzero diagonals, and applying the factros consumes 3 levels.

A similar factorization works for the coefficients-to-slots matrix. In particular, it factors into the inverses of the factors of the slots-to-coefficients matrix.

8 Mod reduction

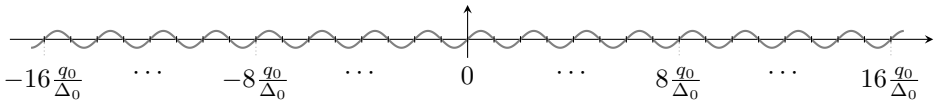
After applying coefficients-to-slots, we have a ciphertext whose slots contain

$$((r_0+\frac{q_0}{\Delta_0}j_0)+(r_{N/2}+\frac{q_0}{\Delta_0}j_{N/2})i,\ldots,(r_{N/2-1}+\frac{q_0}{\Delta_0}j_{N/2-1})+(r_{N-1}+\frac{q_0}{\Delta_0}j_{N-1})i).$$

(These are in bit-reversed order). We can use complex conjugation to split this into two ciphertexts, which encode the real and imaginary parts:

$$\begin{aligned}\mathfrak{c}_{\mathrm{re}} \text{ encodes } &\left(r_0+\frac{q_0}{\Delta_0}j_0,\ldots,r_{N/2-1}+\frac{q_0}{\Delta_0}j_{N/2-1}\right)\\ \mathfrak{c}_{\mathrm{im}} \text{ encodes } &\left(r_{N/2}+\frac{q_0}{\Delta_0}j_{N/2},\ldots,r_{N-1}+\frac{q_0}{\Delta_0}j_{N-1}\right)\end{aligned}$$

Each r_i is much smaller than $\frac{q_0}{\Delta_0}$. Therefore, $r_i+\frac{q_0}{\Delta_0}j_i$ is close to a multiple of $\frac{q_0}{\Delta_0}$, and we must reduce it modulo $\frac{q_0}{\Delta_0}$. Moreover, we know $j_i\in[-16,16]$. Let ψ be a polynomial that approximates reduction modulo $\frac{q_0}{\Delta_0}$:



Then,

$$\begin{aligned}\psi(\mathfrak{c}_{\mathrm{re}}) \text{ encodes } &\left(r_0,\ldots,r_{N/2-1}\right)\\ \psi(\mathfrak{c}_{\mathrm{im}}) \text{ encodes } &\left(r_{N/2},\ldots,r_{N-1}\right)\\ \psi(\mathfrak{c}_{\mathrm{re}})+i\psi(\mathfrak{c}_{\mathrm{im}}) \text{ encodes } &\left(r_0+r_{N/2}i,\ldots,r_{N/2-1}+r_{N-1}i\right)\end{aligned}$$

Finally, apply slots-to-coefficients to $\psi(\mathfrak{c}_{\mathrm{re}})+i\psi(\mathfrak{c}_{\mathrm{im}})$ to get a ciphertext encoding f at level n.