CKKS Lecture Notes
Part 2: Operations

FEI

October 2026

1 Ciphertext operations

Assume ciphertexts $(a_1, b_1), (a_2, b_2)$ encode $f, g \in \mathbb{R}[X]/(X^N + 1)$, respectively, at some modulus $Q \in \mathbb{Z}$ and scale factor $\Delta \in \mathbb{R}$.

In this part, we will look at how to operate on (a_1, b_1) and (a_2, b_2) in ways that correspond to f and g. As stated in the previous handout, $N/2$ complex numbers are stored in the evaluations of f and g at the $2N$th roots of unity (the "slots"). Operations on f and g correspond to operations on the slots.

For some small errors e_1, e_2, we have

$$\begin{aligned} a_1 s + b_1 &\equiv \mathsf{round}(\Delta f) + e_1 \pmod{Q} \\ a_2 s + b_2 &\equiv \mathsf{round}(\Delta g) + e_2 \pmod{Q}. \end{aligned}$$

2 Addition

Ciphertext addition

$$(a_1, b_1) + (a_2, b_2) = (a_1 + a_2, b_1 + b_2).$$

We have

$$\begin{aligned} (a_1 + a_2)s + (b_1 + b_2) &\equiv \mathsf{round}(\Delta f) + \mathsf{round}(\Delta g) + e_1 + e_2 \pmod{Q} \\ &\equiv \mathsf{round}(\Delta(f + g)) \\ &\quad + \mathsf{round}(\Delta f) + \mathsf{round}(\Delta g) - \mathsf{round}(\Delta(f + g)) \\ &\quad + e_1 + e_2. \end{aligned}$$

Therefore, $(a_1 + a_2, b_1 + b_2)$ encodes $f + g$ with error

$$\mathsf{round}(\Delta f) + \mathsf{round}(\Delta g) - \mathsf{round}(\Delta(f + g)) + e_1 + e_2.$$

3 Keyswitching

Rotation, conjugation, and multiplication rely on *keyswitching*. The person who knows the secret key $s \in \mathbb{Z}[X]/(X^N+1)$ might have some other secret polynomial $s' \in \mathbb{Z}[X]/(X^N+1)$. This person can reveal a *keyswitching key*

$$\mathsf{ksk}_{s'}.$$

Anyone with the keyswitching key can *keyswitch*:

Keyswitching

For a polynomial $p \in \mathbb{Z}_Q[X]/(X^N+1)$,

$$\mathsf{keyswitch}(p, \mathsf{ksk}_{s'}) = (a, b)$$

is a ciphertext such that $as + b \equiv ps' + e \pmod{Q}$ for some small e.

Anyone who knows $\mathsf{ksk}_{s'}$ can do this. When the person with the secret key reveals $\mathsf{ksk}_{s' \to s}$, this does not reveal any useful information about s or s'.

4 Rotation

Reminder

Rotating $c \in \mathbb{C}^{N/2}$ by k slots is an automorphism of $\mathbb{C}^{N/2}$. This corresponds to the automorphism σ_{5^k} of $\mathbb{R}[X]/(X^N+1)$, which preserves $\mathbb{R}$ and sends X to X^{5^k}.

Ciphertext rotation

$$\mathsf{rotate}_k((a_1, b_1)) = \mathsf{keyswitch}(\sigma_{5^k}(a_1), \mathsf{ksk}_{\sigma_{5^k}(s)}) + (0, \sigma_{5^k}(b_1)).$$

We have

$$a_1 s + b_1 \equiv \mathsf{round}(\Delta f) + e_1$$
$$\sigma_{5^k}(a_1)\sigma_{5^k}(s) + \sigma_{5^k}(b_1) \equiv \mathsf{round}(\Delta \sigma_{5^k}(f)) + \sigma_{5^k}(e_1).$$

Let $\mathsf{keyswitch}(\sigma_{5^k}(a_1), \mathsf{ksk}_{\sigma_{5^k}(s)}) = (a', b')$. Then,

$$a's + b' = \sigma_{5^k}(a_1)\sigma_{5^k}(s) + e'$$

for some small error e'. Thus,

$$\mathsf{keyswitch}(\sigma_{5^k}(a_1), \mathsf{ksk}_{\sigma_{5^k}(s)}) + (0, \sigma_{5^k}(b_1)) = (a', b' + \sigma_{5^k}(b_1))$$

decrypts to

$$\begin{aligned} a's + (b' + \sigma_{5^k}(b_1)) &= \sigma_{5^k}(a_1)\sigma_{5^k}(s) + \sigma_{5^k}(b_1) + e' \\ &= \mathsf{round}(\Delta\sigma_{5^k}(f)) + \sigma_{5^k}(e_1) + e'. \end{aligned}$$

Conjugation works similarly.

Reminder

Conjugating $c \in \mathbb{C}^{N/2}$ is an automorphism of $\mathbb{C}^{N/2}$. This corresponds to the automorphism σ_{-1} of $\mathbb{R}[X]/(X^N + 1)$, which preserves $\mathbb{R}$ and sends X to X^{-1}.

Ciphertext conjugation

$$\mathsf{conjugate}((a_1, b_1)) = \mathsf{keyswitch}(\sigma_{-1}(a_1), \mathsf{ksk}_{\sigma_{-1}(s)}) + (0, \sigma_{-1}(b_1)).$$

5 Multiplication

Ciphertext multiplication

$$(a_1, b_1) \cdot (a_2, b_2) =$$
$$\mathsf{rescale}_{Q_\ell \to Q_{\ell-1}} \left(\mathsf{keyswitch}(a_1 a_2, \mathsf{ksk}_{s^2}) + (a_1 b_2 + b_1 a_2, b_1 b_2)\right)$$

Let

$$\epsilon = \mathsf{round}(\Delta f) e_2 + \mathsf{round}(\Delta g) e_1 + e_1 e_2.$$

We have

$$\begin{aligned} \mathsf{round}(\Delta f)\mathsf{round}(\Delta g) + \epsilon &= (a_1 s + b_1)(a_2 s + b_2) \\ &= a_1 a_2 s^2 + (b_1 a_2 + a_1 b_2) s + b_1 b_2. \end{aligned}$$

Let

$$(a', b') = \mathsf{keyswitch}(a_1 a_2, \mathsf{ksk}_{s^2}).$$

Then,

$$a' s + b' = a_1 a_2 s^2 + e'$$

for some small error e'. Therefore, $\mathsf{keyswitch}(a_1 a_2, \mathsf{ksk}_{s^2}) + (a_1 b_2 + b_1 a_2, b_1 b_2)$ encodes fg with scale factor Δ^2 and modulus Q. It also has much larger absolute error than the original ciphertexts.

6 Rescaling polynomials

As seen in the previous section, multiplication brings the scale factor from Δ to Δ^2. It also increases the error. We can correct for this by rescaling the result of multiplication.

Rescaling

Let $a \in \mathbb{Z}_Q[X]/(X^N + 1)$. Let $q \mid Q$.

- Round each coefficient of $\mathsf{rep}(a)$ to the nearest multiple of q.
- Divide by q.

This gives us an element $\mathsf{rescale}_{Q\to Q/q}(a) \in \mathbb{Z}_{Q/q}[X]/(X^N+1)$ such that

$$\mathsf{rep}(\mathsf{rescale}_{Q\to Q/q}(a)) \approx \mathsf{rep}(a)/q.$$

Therefore, $\mathsf{rescale}_{Q\to Q/q}(a_f a_g)$ is an approximation of fg with modulus Q/q and scale factor Δ^2/q. If $q \approx \Delta$, then the new scale factor is close to Δ. Also, the error is now on the order of $(\Delta/q)(ge_f + fe_g)$.

7 Rescaling ciphertexts

Recall that we can rescale a polynomial $p \in \mathbb{Z}_Q[X]/(X^N+1)$ by rounding its coefficients to the nearest multiple of q, then dividing by q. This results in a polynomial in $p' \in \mathbb{Z}_{Q/q}[X]/(X^N+1)$ such that $\mathsf{rep}(p') \approx \mathsf{rep}(p)/q$.

We can use this to rescale ciphertexts.

Rescaling ciphertexts

Let (a, b) be a ciphertext with modulus Q. Let $q \mid Q$. Then,

$$\mathsf{rescale}_{Q\to Q/q}((a,b)) = (\mathsf{rescale}_{Q\to Q/q}(a), \mathsf{rescale}_{Q\to Q/q}(b)).$$

If (a, b) encodes $h \in \mathbb{R}[X]/(X^N+1)$ with scale factor Δ, then $\mathsf{rescale}_{Q\to Q/q}((a, b))$ encodes h with scale factor Δ/q.

Recall that the secret key s has small coefficients.

Exercise

What would go wrong with rescaling if s did not have small coefficients?

Solution

Let a', a'' be polynomials such that $a = qa' + a''$ and the coefficients of a'' are in $[-q/2, q/2)$. Define b', b'' similarly. Then, we have

$$\mathsf{rescale}_{Q\to Q/q}((a,b)) = (a', b').$$

This decrypts to

$$a's + b' = \frac{as+b}{q} - \frac{(a''s+b'')}{q}.$$

The error introduced by rescaling is $\frac{(a''s+b'')}{q}$. This is only expected to be small if s is small.

Note that we can level-drop a ciphertext to decrease its level in a similar way to how we level-drop a plaintext.

8 Levels

In order to multiply repeatedly and rescale by a suitable factor each time, we use a set of *levels*. Each level ℓ has its own modulus Q_ℓ and scale factor Δ_ℓ.

The highest level has modulus $Q_n = q_0 q_1 \dots q_n$ and scale factor Δ_n. Level ℓ has modulus $Q_\ell = q_0 q_1 \dots q_\ell$ and scale Δ_ℓ. Each q_i is prime, and $q_i \approx \Delta_n$ for each $i > 0$.

Consider multiplying two plaintexts at level ℓ. Before rescaling, the result has modulus Q_ℓ and scale factor Δ_ℓ^2. After rescaling, the result has modulus $Q_{\ell-1}$ and scale factor Δ_ℓ^2/q_ℓ. Thus, we define

$$\Delta_{\ell-1} = \Delta_\ell^2/q_\ell.$$

So, the result is at level $\ell - 1$.

Because $q_i \approx \Delta_n$, we have

$$\Delta_0 \approx \Delta_1 \approx \dots \approx \Delta_n.$$

Level dropping

We can only add and multiply plaintexts at the same level. So, we sometimes need to decrease the level of a plaintext in order to do operations. This requires adjusting the scale.

Given a plaintext a at level ℓ, we can drop its level to level $\ell - 1$ by computing

$$\mathsf{rescale}_{Q_\ell \to Q_{\ell-1}}(\mathsf{round}(\Delta_\ell) \cdot a).$$

9 Key-switching

In this section, we discuss how to generate keyswitching keys and use them to keyswitch. The keyswitching key is generated for some modulus Q. With the keyswitching key, it's only possible to keyswitch ciphertexts whose modulus divides Q.

Attempt 1

Let $\mathsf{ksk}_{s'} = (a, b)$ such that $as + b = s'$. Then, for any polynomial p, (pa, pb) encodes ps'.

This is not secure.

Attempt 2

Let $\mathsf{ksk}_{s'} = (a, b)$ such that $as + b = s' + e$ for some error e. Then, for any polynomial p, (pa, pb) encodes ps' with error pe.

We want a ciphertext that encodes ps' with small error; in general, pe is not necessarily small. However, if p is small, then pe is small. For general p, we can express p in terms of small polynomials and apply attempt 2 to each of those:

Attempt 3

Choose a set $W = \{w_1, w_2, \ldots, w_k\} \subseteq \mathbb{Z}_Q$. (This is public.) Let $\mathsf{ksk}_{s'} = \{(a_i, b_i) : i \leq k\}$ such that $a_i s + b_i \equiv w_i s' + e_i \pmod{Q}$ for some small error e_i. Express p as

$$p = w_1 p_1 + \cdots + w_k p_k$$

for some small polynomials $p_1, \ldots, p_k \in \mathbb{Z}_Q[X]/(X^N+1)$. So, $(a_i p_i, b_i p_i)$ encodes $w_i p_i s'$ with error $p_i e_i$ for some small e_i. Therefore,

$$(a_1 p_1 + \cdots + a_k p_k, b_1 p_1 + \cdots + b_k p_k)$$

encodes $w_1 p_1 s' + \cdots + w_k p_k s' = ps'$ with error $p_1 e_1 + \cdots + p_k e_k$.

Because each p_i and e_i is small, $p_1 e_1 + \cdots + p_k e_k$ is small.

We can make it smaller by rescaling. We go from modulus Q to a larger modulus PQ, then scale away P to get back to modulus Q.

Attempt 4

Choose a set $W = \{w_1, w_2, \ldots, w_k\} \subseteq \mathbb{Z}_Q$. (This is public.) Let $\mathsf{ksk}_{s'} = \{(a_i, b_i) : i \leq k\}$ such that $a_i s + b_i \equiv w_i P s' + e_i \pmod{PQ}$ for some small error e_i. Express p as

$$p = w_1 p_1 + \cdots + w_k p_k$$

for some small polynomials $p_1, \ldots, p_k \in \mathbb{Z}_Q[X]/(X^N+1)$. So, $(a_i p_i, b_i p_i)$ encodes $w_i p_i P s'$ with error $p_i e_i$ for some small e_i. Therefore,

$$(a_1 p_1 + \cdots + a_k p_k, b_1 p_1 + \cdots + b_k p_k)$$

encodes $w_1 P p_1 s' + \cdots + w_k P p_k s' \equiv p P s' \pmod{PQ}$ with error $p_1 e_1 + \cdots + p_k e_k$. So,

$$(\mathsf{rescale}_{PQ \to Q}(a_1 p_1 + \cdots + a_k p_k), \mathsf{rescale}_{PQ \to Q}(b_1 p_1 + \cdots + b_k p_k))$$

encodes ps' with error $\frac{p_1 e_1 + \cdots + p_k e_k}{P} + e_{\text{rescale}}$.

In fact, we can choose $w_0, \ldots, w_k$ is based on the Chinese Remainder Theorem. Assume there are $n+1$ primes in the modulus chain. For simplicity, assume $n \equiv 2 \pmod 3$ and let $k = (n-2)/3$. Put the primes in the modulus chain into groups:

$$Q_n = (q_0 q_1 q_2)(q_3 q_4 q_5) \ldots (q_{n-2} q_{n-1} q_n) = D_0 D_1 \ldots D_k.$$

For $i = 0, 1, \ldots, k$, let $w_i \in \mathbb{Z}_{Q_n}$ such that:

- $w_i \equiv 1 \pmod{D_i}$
- $w_i \equiv 0 \pmod{D_j}$ for all $j \neq i$.

Choose the auxiliary modulus P such that $P > D_i/2$.

Let $p \in \mathbb{Z}_{Q_n}[X]/(X^N+1)$. Let $p_i = \mathsf{rep}(p \pmod{D_i})$. Then, each p_i has coefficients smaller than P and

$$p_0 w_0 + \cdots + p_k w_k \equiv p \pmod{Q_n}.$$

10 Ciphertext dot product

When we do CKKS operations on a computer, most of the time is spent keyswitching, so reducing the number of keyswitches is important.

One optimization is the "ciphertext dot product." This name can be confusing because it is not a dot product of the complex vectors that the ciphertexts encode. It is more like a dot product of vectors whose entries are ciphertexts.

Let

$$(a_1, b_1), \ldots, (a_k, b_k)$$

and

$$(\alpha_1, \beta_1), \ldots, (\alpha_k, \beta_k)$$

be two lists of ciphertexts. Assume they encode

$$f_1, \ldots, f_k \in \mathbb{R}[X]/(X^N + 1)$$

and

$$g_1, \ldots, g_k \in \mathbb{R}[X]/(X^N + 1),$$

respectively. Then,

$$\mathsf{rescale}_{Q \to Q/q} \left(\mathsf{keyswitch} \left(\sum_i a_i \alpha_i, \mathsf{ksk}_{s^2} \right) + \left(\sum_i (a_i \beta_i + b_i \alpha_i), \sum_i b_i \beta_i \right) \right)$$

encodes

$$f_1 g_1 + \cdots + f_k g_k.$$