CKKS Lecture Notes
Part 1: CKKS

FEI

October 2026

1 Introduction

In CKKS, we encrypt data by expressing it in terms of a secret key and adding a small amount of noise. Only the person with the secret key can decrypt this data. However, other people can perform operations this encrypted data, which produce data encrypted under the same secret key.

2 CKKS

Let $N = 65536 = 2^{16}$. A CKKS ciphertext encodes an element of the ring $\mathbb{C}^{N/2}$. We can perform the following operations on encrypted data:

- **Addition.** Given encryptions of $(x_0, \ldots, x_{32767}), (y_0, \ldots, y_{32767}) \in \mathbb{C}^{N/2}$, we can compute an encryption of

$$(x_0 + y_0, \ldots, x_{32767} + y_{32767}).$$

- **Multiplication.** Given encryptions of $(x_0, \ldots, x_{32767}), (y_0, \ldots, y_{32767}) \in \mathbb{C}^{N/2}$, we can compute an encryption of

$$(x_0 y_0, \ldots, x_{32767} y_{32767}).$$

- **Automorphism.**
  - **Rotation.** Given an encryption of $(x_0, \ldots, x_{32767}) \in \mathbb{C}^{N/2}$, and an integer k, we can compute an encryption of

$$(x_k, x_{k+1} \ldots, x_{32767}, x_0, \ldots, x_{k-1}).$$

  - **Conjugation.** Given an encryption of $(x_0, \ldots, x_{32767}) \in \mathbb{C}^{N/2}$, we can compute an encryption of

$$(\overline{x_0}, \ldots, \overline{x_{32767}}) \in \mathbb{C}^{N/2}.$$

3 Polynomial interpolation

Before we encrypt data, we express it in terms of a polynomial. Let $N = 65536 = 2^{16}$. Let $\mathbf{x} = (x_0, \ldots, x_{N/2-1}) \in \mathbb{C}^{N/2}$. Let $\zeta = e^{\frac{i\pi}{N}}$.

There is exactly one complex polynomial f_x of degree at most $N-1$ such that

$$f(\zeta^{5^k}) = x_k$$
$$f(\zeta^{-5^k}) = \overline{x_k}$$

for all $0 \leq k \leq N/2 - 1$.

In fact, this polynomial has real coefficients. To see this, let g be the polynomial whose coefficients are the conjugates of the coefficients of f_x. Notice that f_x and g are both polynomials of degree at most $N-1$, and they agree at N points. Therefore, they are equal.

Polynomial ring

Let $\mathbb{R}[X]$ denote the ring of polynomials with real coefficients. Then, the polynomial f_x is in $\mathbb{R}[X]$.

Since we only care about the evaluation of f_x at points $X \in \mathbb{C}$ where $X^N + 1 = 0$, we can view f_x as an element of $\mathbb{R}[X]/(X^N+1)$ (the ring of real polynomials where $X^N + 1$ can be reduced to 0).

CKKS operations in terms of polynomials

Let $(x_0, \ldots, x_{N/2-1}), (y_0, \ldots, y_{N/2-1}) \in \mathbb{C}^{N/2}$. Define f_x as above. Define $f_y, f_{x+y}, f_{xy}, f_{\mathsf{rotate}_k(x)}, f_{\overline{x}}$ analogously. Then, operations on these polynomials correspond to operations on the underlying data as follows:

- **Addition.** $f_x + f_y = f_{x+y}$.
- **Multiplication.** $f_x f_y = f_{xy}$.
- **Automorphism.** Let m be odd. Define the function $\sigma_m : \mathbb{R}[X]/(X^N+1) \to \mathbb{R}[X]/(X^N+1)$ such that

  $$\sigma_m(f)(X) = f(X^m)$$

  for all $f \in \mathbb{R}[X]/(X^N+1)$. Note that σ_m is an automorphism of $\mathbb{R}[X]/(X^N+1)$.

  - **Rotation.** $f_{\mathsf{rotate}_k(x)} = \sigma_{5^k}(f_x)$.

– **Conjugation.** $f_{\overline{x}} = \sigma_{2N-1}(f_x)$.

So, we can encrypt elements of $\mathbb{C}^{N/2}$ by encrypting these polynomials. As long as we can add, multiply, and apply σ_m to the encryptions of these polynomials, we can perform addition, multiplication, rotation, and conjugation on the underlying elements of $\mathbb{C}^{N/2}$.

4 Finite ring approximation

Choose a modulus $Q \in \mathbb{Z}$ and a scale factor Δ. Typically, Δ is on the order of 2^{50} and Q is much larger. To encrypt a polynomial $f \in \mathbb{R}[X]/(X^N + 1)$, we first approximate it as an element of the ring

$$\mathbb{Z}_Q[X]/(X^N + 1).$$

This finite ring element is a *plaintext*.

First, we approximate f as an element of $\mathbb{Z}[X]/(X^N + 1)$. We scale f up by a factor of Δ, then round each coefficient to the nearest integer to get

$$\mathsf{round}(\Delta f) \in \mathbb{Z}[X]/(X^N + 1).$$

Example

Let $\Delta = 2^{50}$. Let $f = 0.0001 + 0.0002X^2$. Then, we can approximate f as

$$\mathsf{round}(2^{50}(0.0001 - 0.0002X^2)) = 112589990684 - 225179981369X^2.$$

Integers in the range $[-Q/2, Q/2)$ can be represented as elements of $\mathbb{Z}_Q$. So, if all coefficients of $\mathsf{round}(\Delta f)$ are in this range, then $\mathsf{round}(\Delta f)$ can be represented as an element of $\mathbb{Z}_Q[X]/(X^N + 1)$.

In the other direction, each element of $\mathbb{Z}_Q$ has an integer representative in the range $[-Q/2, Q/2)$:

Integer representatives

For $k \in \mathbb{Z}_Q$, define its integer representative $\mathsf{rep}(k) \in \mathbb{Z}$ as the unique integer such that:

- $\mathsf{rep}(k) \equiv k \pmod{Q}$.
- $-Q/2 \leq \mathsf{rep}(k) < Q/2$.

Similarly, we can take rep of an element of $\mathbb{Z}_Q[X]/(X^N + 1)$ by taking

the representative of each of its coefficients.

We also allow these approximations to have a small amount of noise.

Approximation

CKKS is a scheme for approximate arithmetic, so we allow a small amount of noise. Say that $a \in \mathbb{Z}_Q[X]/(X^N+1)$ "approximates" $f \in \mathbb{R}[X]/(X^N+1)$ if

$$\mathsf{rep}(a) = \mathsf{round}(\Delta f) + e$$

for some error $e \in \mathbb{Z}[X]/(X^N+1)$ with small coefficients.

5 Encryption

Now, we have a finite ring element $\mathsf{round}(\Delta f) \in \mathbb{Z}_Q[X]/(X^N+1)$. We can encrypt this by adding a small amount of noise and expressing it in terms of a secret key.

The secret key is an integer polynomial $s \in \mathbb{Z}[X]/(X^N+1)$ with small coefficients. A ciphertext is a pair (a, b) of polynomials in $\mathbb{Z}_Q[X]/(X^N+1)$ such that

$$as + b = \mathsf{round}(\Delta f) + e$$

for some error $e \in \mathbb{Z}_Q[X]l(X^N+1)$ with small coefficients.

Public-key encryption

Let m be a plaintext. Randomly generate $u, e_0, e_1 \in \mathbb{Z}_Q[X]/(X^N+1)$ with small coefficients. Then, compute the ciphertext $(ua_{\mathsf{pk}}+e_0, m+ub_{\mathsf{pk}}+e_1)$. Then,

$$(ua_{\mathsf{pk}} + e_0)s + (m + ub_{\mathsf{pk}} + e_1) = u(a_{\mathsf{pk}}s + b_{\mathsf{pk}}) + e_0 s + e_1 + m \approx m.$$

Thus, $(ua_{\mathsf{pk}} + e_0, m + ub_{\mathsf{pk}} + e_1)$ is an encryption of m.

The security of CKKS relies on the following assumption:

Ring learning with errors (rough statement)

Alice knows a secret $s \in \mathbb{Z}_Q[X]/(X^N+1)$ with small coefficients. She also has some polynomial $m \in \mathbb{Z}[X]/(X^N+1)$. Alice generates $a \in \mathbb{Z}_Q[X]/(X^N+1)$ uniformly at random, and randomly generates

$e \in \mathbb{Z}_Q[X]/(X^N + 1)$ with small coefficients. She chooses b such that

$$as + b = m + e.$$

- If Alice tells Bob a, b, and m, this does not give Bob any useful information about s or e.
- If Alice tells Bob a and b, this does not give Bob any useful information about m, s, or e.

This guarantees the following:

- Given an encryption of an unknown $x \in \mathbb{C}^{N/2}$, an attacker cannot efficiently recover x or the secret key.
- Given a known $x \in \mathbb{C}^{N/2}$ and an encryption of x, an attacker cannot efficiently recover the secret key.