

Worst-case Precision Accuracy Bounds in Approximate Homomorphic Encryption

(and, an Introduction to Post-Quantum Cryptography)

Kyle Yates

Clemson University

March 8, 2025

This talk will consist of 2 parts:

Part 1: An Introduction to Post-Quantum Cryptography (~25 mins)

Part 2: Worst-case Precision Accuracy Bounds in Approximate Homomorphic Encryption (~20 mins)

Part 1:

An Introduction to Post-Quantum Cryptography

What is Quantum Computing?

Quantum computers are computers that use quantum bits (qubits) rather than classical bits.

The fundamental structure of qubits is different from classical bits and allows for some calculations to be performed exponentially faster than modern computers.

Quantum computers are still in their infancy, but do have potential.

Timeline

A brief quantum computer timeline

Richard Feynman

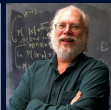
proposes to build a quantum computer for simulation



1982

Peter Shor

Algorithm for prime factorization of large integers



1994

Lov Kumar Grover

shows how to search in \sqrt{N}



1997

Google

Quantum supremacy announced

Google Research

2019

IBM

Unveils breakthrough 433-Qubit Quantum Processor



2023

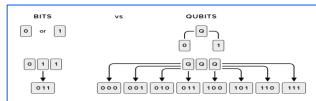
Most Recently: Microsoft Majorana 1 chip February 19, 2025 (Still lots of skepticism).

Courtesy: Nokia-Bell Labs

Timeline

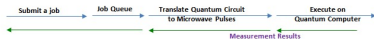
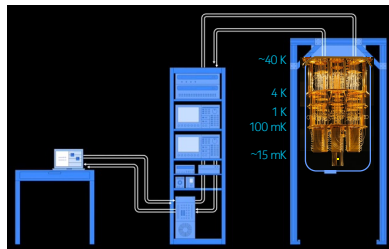
- **Quantum Supremacy:** Google/NASA, Oct 2019
 - 200 seconds vs. 10,000 years (for random sampling)
 - (but Zhang et al, Aug 2022...)

- **Power Source:** entangled qubits (superposition)



- **Threat:** can solve problems once thought mathematically intractable
 - Includes most crypto-systems employed today (e.g., elliptic curve cryptography)
 - **Attack:** Store-now / Crack-later

IBM Quantum Computer (source: IBM)



NOKIA

Courtesy: Nokia-Bell Labs

How do Quantum Computers Impact Modern Cryptographic Systems?

What Can a Quantum Computer Do Efficiently?

- Integer Factorization i.e. find p from pq
- Discrete Logarithm i.e. find x from $g, g^x \pmod p$
- Discrete Logarithm on Elliptic Curves
- Finding Generators of a Principal Ideal

What Can a Quantum Computer NOT Do Efficiently?

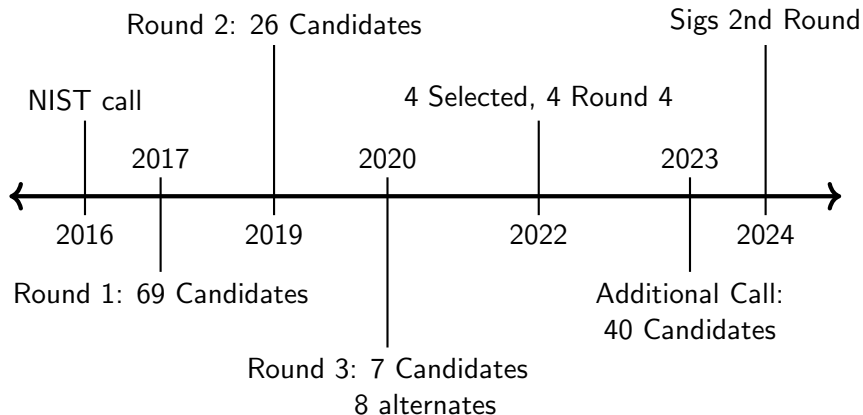
- *We don't know yet. But, we suspect...*
 - ▶ Decode random linear codes.
 - ▶ Solve multivariate quadratic equations over finite fields.
 - ▶ Solve lattice problems.
 - ▶ Find isogenies between elliptic curves.

Post-Quantum Cryptography

Cryptography that can be implemented on a classical computer but will (hopefully) be secure against attacks completed on either a classical or a quantum computer.

There has been a push to move towards Post-Quantum Cryptography with the possible threat of quantum computers.

NIST Standardization Process



We'll briefly discuss 4 popular approaches for post-quantum cryptography:

- 1 **Code-based** cryptography
- 2 **Multivariate** cryptography
- 3 **Lattice-based** cryptography
- 4 **Isogeny-based** cryptography

- An $[n, k]$ -**linear code** \mathcal{C} is a k -dimensional subspace of \mathbb{F}_q^n . We call n the length of the code, and k its dimension. An element $\mathbf{x} \in \mathbb{F}_q^n$ is called a **codeword** if $\mathbf{x} \in \mathcal{C}$.
- The number of nonzero elements in \mathbf{x} is called the **Hamming weight** of \mathbf{x} and we denote it as $\text{wt}(\mathbf{x})$.
- We define a basis matrix \mathbf{G} of \mathcal{C} to be the **generator matrix** of the code. A **parity check matrix** of \mathcal{C} is $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ such that the right kernel of \mathbf{H} is the code \mathcal{C} . The subspace spanned by the rows of \mathbf{H} is called the **dual code** of \mathcal{C} .

Code-Based Cryptography

Hard Problems

Problem (Decoding a random code)

Given a random linear code \mathcal{C} and a vector $\mathbf{y} = \mathbf{x} + \mathbf{e}$ ($\text{wt}(\mathbf{e}) \leq t$, $\mathbf{x} \in \mathcal{C}$), find \mathbf{x} .

Problem (SD: Syndrome Decoding)

Given a parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ and a vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$ (called the syndrome), find a vector $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{H}\mathbf{e}^\top = \mathbf{s}^\top$ and $\text{wt}(\mathbf{e}) \leq t$.

Multivariate Cryptography

- Setting: finite field \mathbb{F}_q
- Multivariate equations: equations in many variables

- ▶ Linear equations:

- ★ Example: 3 equations in 3 variables over \mathbb{F}_5

$$x_1 + 4x_2 + 2x_3 = 2$$

$$x_1 + 3x_2 = 3$$

$$x_1 + x_2 + 4x_3 = 3$$

- ★ Ways to solve: Gaussian Elimination

- ▶ Quadratic Equations:

- ★ Example: 3 equations in 3 variables over \mathbb{F}_5

$$x_1^2 + 4x_2x_1 + 2x_3x_2 = 2$$

$$x_1x_3 + 3x_2^2 + x_3 = 3$$

$$x_1x_2 + x_2 + 4x_3 = 3$$

- ★ Ways to solve: Polynomial Solvers, Gröbner basis algorithms

Multivariate Cryptography

Hard Problem

Problem (MQ: Multivariate Quadratic)

Given a system of multivariate quadratic equations over a finite field, find a solution.

Lattices

- Vector Spaces

- ▶ Let $\mathcal{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n) \in \mathbb{R}^{m \times n}$ be a matrix of rank n . Consider the set

$$V = \left\{ \sum_{i=1}^m c_i \mathbf{b}_i : c_i \in \mathbb{R} \right\}.$$

- ▶ Then V is a vector space, and we call \mathcal{B} a basis of V .

- Lattices

- ▶ Let $\mathcal{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n) \in \mathbb{R}^{m \times n}$ be a matrix of rank n . Consider the set

$$\mathcal{L} = \mathcal{L}(\mathcal{B}) = \left\{ \sum_{i=1}^m c_i \mathbf{b}_i : c_i \in \mathbb{Z} \right\}.$$

- ▶ Then \mathcal{L} is a discrete additive subgroup of \mathbb{R}^m , called a **lattice** of rank n , and \mathcal{B} is a lattice basis of \mathcal{L} .

Lattice-Based Cryptography

Hard Problems

Problem (SVP: Shortest Vector Problem)

Given a basis \mathcal{B} of \mathcal{L} , find the shortest nonzero vector. i.e., find nonzero $v \in \mathcal{L}$ such that $\|v\|_2$ is minimized.

Problem (LWE: Learning With Errors)

Let $\mathbf{s} \in \mathbb{Z}_q^n$ fixed. Given many pairs $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where \mathbf{a}_i is uniform random and b_i is computed as

$$b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \pmod{q}$$

for small random $e_i \in \mathbb{Z}$ (from some discrete subgaussian distribution), find \mathbf{s} .

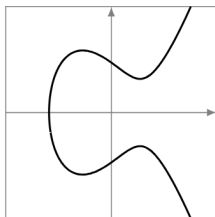
[Regev 2005] LWE problem is as hard as worst-case δ -SVP on a quantum computer for $\delta = \tilde{O}(n^{1.5})$.

Isogeny-Based Cryptography

An elliptic curve E over a field \mathbb{F} is the set of solutions $(x, y) \in \mathbb{F}^2$ to

$$y^2 = x^3 + ax + b$$

for suitable fixed $a, b \in \mathbb{F}$, plus a 'point at infinity' \mathcal{O} .



An isogeny ϕ between elliptic curves E_1 and E_2 is a rational map $\phi : E_1 \rightarrow E_2$ that preserves addition (i.e., is a group homomorphism).

Isogeny-Based Cryptography

Problem

Given two elliptic curves E_1, E_2 find an isogeny ϕ (if it exists) such that

$$\phi : E_1 \rightarrow E_2$$

Part 2:

Worst-case Precision Accuracy Bounds in Approximate Homomorphic Encryption

Homomorphic encryption allows for **addition** and **multiplication** to be performed on ciphertexts without decrypting or possessing the secret key.

Furthermore, ciphertext computations result in the same output as plaintext computations. That is, for messages m_0, m_1 and key k ,

$$\text{Enc}(m_0, k) + \text{Enc}(m_1, k) = \text{Enc}(m_0 + m_1, k),$$

$$\text{Enc}(m_0, k) \times \text{Enc}(m_1, k) = \text{Enc}(m_0 \cdot m_1, k).$$

Notation

For positive q , let $\mathbb{Z}_q := \mathbb{Z} \cap [-q/2, q/2)$.

For n a power of two, define the rings R_n and $R_{n,q}$ via

$$R_n := \mathbb{Z}[x]/(x^n + 1),$$

$$R_{n,q} := \mathbb{Z}_q[x]/(x^n + 1).$$

CKKS Structure

We'll discuss the Cheon-Kim-Kim-Song (CKKS) scheme ([1], 2016), which allows for homomorphic encryption with approximate arithmetic of numbers.

Our message space in CKKS is $\mathbb{C}^{n/2}$. The general structure of CKKS is as follows.

- 1 **Encode** a message $z \in \mathbb{C}^{n/2}$.
- 2 **Encrypt** an encoded message from R_n into a ciphertext in $R_{n,q}^2$.
- 3 Do ciphertext computation in $R_{n,q}^2$.
- 4 **Decrypt** a ciphertext from $R_{n,q}^2$ to R_n using the secret key.
- 5 **Decode** back to $\mathbb{C}^{n/2}$.

CKKS Preliminaries

For the encoding procedure, we need to define the following spaces and mappings:

$$\mathbb{H} = \{z \in \mathbb{C}^n : z_j = \overline{z_{n-j}}\}.$$

$\pi : \mathbb{H} \rightarrow \mathbb{C}^{n/2}$ is the projection of \mathbb{H} onto $\mathbb{C}^{n/2}$.

$\sigma : \mathbb{C}[x]/(x^n + 1) \rightarrow \mathbb{C}^n$ is the **canonical embedding** map. Let $\zeta_1, \zeta_2, \dots, \zeta_n$ be the n roots of $\Phi(x) = x^n + 1$, which are all $2n$ th primitive roots of unity. Given a polynomial $w \in \mathbb{C}[x]/(x^n + 1)$, σ is defined via

$$\sigma(w) = (w(\zeta_1), w(\zeta_2), \dots, w(\zeta_n)) \in \mathbb{C}^n$$

CKKS Mappings

The mappings

$$\pi : \mathbb{H} \rightarrow \mathbb{C}^{n/2}, \quad \sigma : \mathbb{C}[x]/(x^n + 1) \rightarrow \mathbb{C}^n$$

are both isomorphisms. Furthermore, σ establishes an isomorphism between $\mathbb{R}[x]/(x^n + 1)$ and \mathbb{H} . Thus, we have the isomorphisms

$$\begin{aligned} \pi \circ \sigma &: \mathbb{R}[x]/(x^n + 1) \rightarrow \mathbb{C}^{n/2}, \\ \sigma^{-1} \circ \pi^{-1} &: \mathbb{C}^{n/2} \rightarrow \mathbb{R}[x]/(x^n + 1). \end{aligned}$$

Note that $\pi \circ \sigma$ is a scaled isometry in the 2-norm:

$$\|(\pi \circ \sigma)(w)\|_2 = \sqrt{\frac{n}{2}} \|w\|_2.$$

CKKS Encoding and Decoding

For $z \in \mathbb{C}^{n/2}$ and $\Delta \in \mathbb{N}$, **CKKS encoding** is

$$\text{Ecd}(z, \Delta) = \lfloor \sigma^{-1}(\Delta \pi^{-1}(z)) \rfloor \in \mathbb{Z}[x]/(x^n + 1).$$

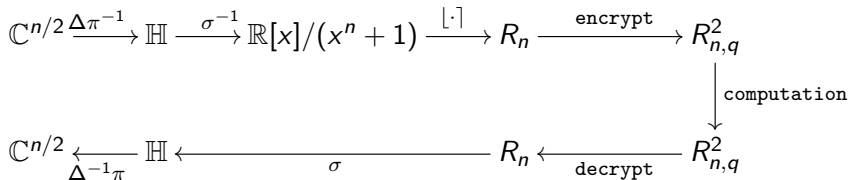
For $m \in \mathbb{Z}[x]/(x^n + 1)$ and $\Delta \in \mathbb{N}$, **CKKS decoding** is

$$\text{Dcd}(m, \Delta) = \pi(\sigma(\Delta^{-1} m)) \in \mathbb{C}^{n/2}.$$

An observation: the rounding step in encoding adds some error to our message.

Recall that $R_n = \mathbb{Z}[x]/(x^n + 1)$ and $R_{n,q} = \mathbb{Z}_q[x]/(x^n + 1)$.

CKKS Mappings



CKKS Encryption

Once an encoded message $m = \text{Ecd}(z, \Delta) \in R_n$ is established, a secret polynomial s with coefficients in $\{0, \pm 1\}$ is chosen.

Encryption	
Step 1.	Sample a uniform randomly from $R_{n,q}$.
Step 2.	Sample e randomly from R_n with small coefficients.
Step 3.	Compute $b = -as + m + e \pmod{(x^n + 1, q)}$.
Step 4.	Return $\text{ct} = (a, b) \in R_{n,q}^2$.

Observe that $b + as \equiv m + e \pmod{(x^n + 1, q)}$.

CKKS Decryption

Suppose we are given s and $ct = (a, b) \in R_{n,q}^2$ satisfying

$$b + as \equiv m + e \pmod{(x^n + 1, q)}.$$

Decryption	
Step 1.	Compute $m' = b + as \pmod{(x^n + 1, q)}$.
Step 2.	Compute $z' = \text{Dcd}(m', \Delta)$.
Step 3.	Return z' .

Since $m' = m + e$, note that z' will be different from the original message $z \in \mathbb{C}^{n/2}$. Furthermore, the error term e expands with ciphertext operations. This makes $\|z - z'\|$ larger.

Precision Loss in CKKS

Definition (Precision Loss, [2])

Consider a normed space $(\mathcal{M}, \|\cdot\|)$, messages $m_1, \dots, m_N \in \mathcal{M}$, and a circuit $C : \mathcal{M}^N \rightarrow \mathcal{M}$. Then we define the precision loss associated with calculating the circuit C homomorphically as the distance $\|\tilde{m} - m\|$, where \tilde{m} is the output of the homomorphic evaluation of $C(m_1, \dots, m_N)$, and m is the true value of the circuit.

We wish to determine worst-case precision accuracy bounds based on allowed amount of computations in CKKS.

The topic of precision accuracy is surprisingly understudied from the theoretical perspective (Costache et al. 2023 [2]).

The analyses of noise and precision is usually in the **canonical embedding norm**, which is the infinity norm of the canonical embedding. i.e., for $w \in \mathbb{C}[x]/(x^n + 1)$,

$$\|w\|_{\infty}^{\text{can}} = \|\sigma(w)\|_{\infty}.$$

This allows us to discuss sizes of error polynomials $e \in R_n$ directly in the message space $\mathbb{C}^{n/2}$.

We consider the total precision loss as

Total precision loss = Precision loss from encoding error
+ Precision loss from encryption error.

Precision Loss from Encoding

CKKS encoding and decoding is implemented with fast Fourier transforms (FFTs), which also contributes to additional error.

Lemma

Let m be the true CKKS encoding of message $z \in \mathbb{C}^{n/2}$, and \hat{m} the CKKS encoding of z obtained from using IEEE 754 double floating-point arithmetic. Suppose that $\|z\|_\infty \leq 2^{49}/\Delta$. Then, $\|\hat{m} - m\|_\infty \leq 1$.

Corollary

For $z \in \mathbb{C}^{n/2}$, suppose that $\hat{m} \in R_n$ is the CKKS encoding of z computed in IEEE 754 double floating-point arithmetic under scale factor Δ . If $\|z\|_\infty \leq 2^{49}/\Delta$, then $\|\hat{m}\|_\infty^{\text{can}} \leq \Delta \|z\|_\infty + \frac{3n}{2\sqrt{2}}$.

Costache et al. 2023 show $\|m\|_\infty \leq \Delta \|z\|_\infty + \frac{1}{2}$.

Precision Loss From Encryption Error

Let h be the hamming weight of s . For each operation, suppose the input errors are bounded by E and the messages are bounded by $t/2$ (in the canonical embedding norm).

Operation	Bound from [2]	Our Bound
Encryption	$\sigma\sqrt{\frac{n^2}{2}} + hn + n \cdot H_{\mathbb{C}}(\alpha, n)$	nh
Add	$2E$	$2E$
Multiply	$Et + E^2$	$Et + E^2$
Relinearize	$E + \sqrt{n} \cdot \eta_{ks} \cdot H_{\mathbb{C}}(\alpha, n)$	$E + \frac{n^3}{24} + \frac{n}{\sqrt{2}}h$
Rescale	$\Delta^{-1}E + \sqrt{\frac{n}{12}(h+1)} \cdot H_{\mathbb{C}}(\alpha, n)$	$\Delta^{-1}E + \frac{n}{2\sqrt{2}}(h+2)$

Table: Comparison of Canonical Embedding Bounds

Scheme	CKKS	CKKS	CKKS
λ	128	196	256
$\log_2(\Delta)$	43	42	42
$\log_2(n)$	15	16	17
L	8	6	5
$\log_2(Q)$	393	297	255
$\log_2(P)$	134	100	88
$\log_2(PQ)$	527	397	343
precision	13.22	13.31	13.36

Table 7.9: Secure CKKS Parameters with $\text{wt}(s) = 64$

Scheme	CKKS	CKKS	CKKS
λ	128	196	256
$\log_2(\Delta)$	45	43	40
$\log_2(n)$	14	15	15
L	6	8	6
$\log_2(Q)$	321	393	285
$\log_2(P)$	109	136	100
$\log_2(PQ)$	430	529	385
precision	15.985	11.22	10.31

Table 7.11: Secure CKKS Parameters with $\text{wt}(s) = 256$

Scheme	CKKS	CKKS	CKKS
λ	128	196	256
$\log_2(\Delta)$	39	42	44
$\log_2(n)$	14	15	16
L	6	6	8
$\log_2(Q)$	279	303	402
$\log_2(P)$	96	103	137
$\log_2(PQ)$	375	406	540
precision	10.495	12.985	12.22

Table 7.10: Secure CKKS Parameters with $\text{wt}(s) = 128$

Scheme	CKKS	CKKS	CKKS
λ	128	196	256
$\log_2(\Delta)$	46	49	49
$\log_2(n)$	14	15	15
L	6	8	6
$\log_2(Q)$	327	447	354
$\log_2(P)$	113	153	121
$\log_2(PQ)$	440	600	475
precision	11.72	10.64	12.73



Table 7.12: Secure CKKS Parameters with $s \leftarrow U(R_{n,s})$

We repeatedly predict a worst-case accuracy of about 10 to 13 bits of precision.

This is about what we should expect. The predictions and observed accuracy in [2] give 20 to 26 bits of precision for similar multiplicative depths.

This is due to the difference of a square root factor the earlier encryption error bounds.

References

-  Jung Hee Cheon, Andrey Kim, Miran Kim and Yongsoo Song. *Homomorphic Encryption for Arithmetic of Approximate Numbers.*
-  Anamaria Costache and Benjamin R. Curtis and Erin Hales and Sean Murphy and Tabitha Ogilvie and Rachel Player. *On the Precision Loss in Approximate Homomorphic encryption.*