An Introduction to the CKKS Approximate Homomorphic Encryption Scheme

Nathan Manohar

IBM T.J. Watson Research Center

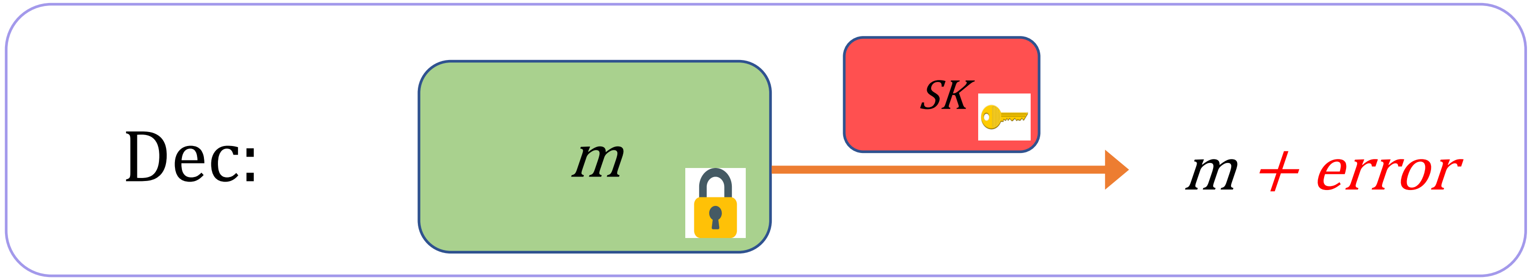
CKKS Approximate HE Scheme (2017)

- Supports arithmetic circuits over real/complex numbers
- Only for *approximate* arithmetic

Scheme	Circuit Type	Plaintext Type
BGV	Arithmetic	Mod p
BFV	Arithmetic	Mod p
GSW	Boolean	Several Bits
FHEW	Boolean	Several Bits
TFHE	Boolean	Several Bits
CKKS	Arithmetic	Real/Complex

CKKS Approximate HE Scheme (2017)

- Main insight: Treat error as part of approximate computation error
- Allows for much more efficient constructions!



Applications

- ❖ Machine Learning
- ❖ Secure Genome Analysis
- ❖ Big Data Analysis
- ❖ Secure Cloud Computing

and many more!

CKKS Overview

- Messages are vectors of up to $N/2$ complex numbers
- Message space is ring $R = \mathbb{Z}[X]/XN + 1$, for N a power of 2
- Vector of complex numbers encoded via inverse of canonical embedding $\sigma^{-1}\colon \mathbb{C}^{N/2} \to R$ up to some precision
- Ciphertexts are two ring elements in R_{Q_ℓ} for a modulus Q_ℓ and ciphertext level $\ell \leq L$.
- Homomorphic computation is SIMD

CKKS Message Encoding/Decoding

- $\mathbb{Z}[X]/X^N + 1$ can be embedded into $\mathbb{C}^{\frac{N}{2}}$ via "canonical embedding"

- Simply means evaluate $m(X) \in \mathbb{Z}[X]/X^N + 1$ at all N primitive $2N$th roots of unity

- In this case, these are $e^{\frac{2\pi i}{2N} * k}$ for k odd

- Embedding has redundancy since $-k$ and k are complex conjugates

CKKS Message Encoding

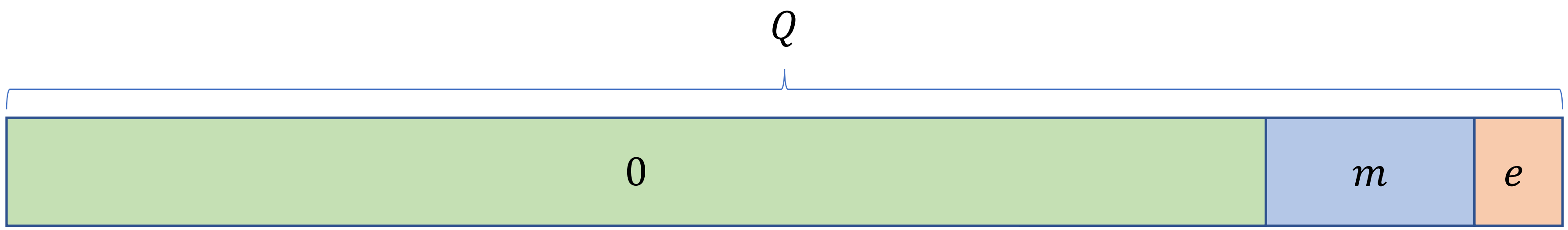
- Vector of $N/2$ complex numbers expanded to $\mathbb{C}^N$
- Multiply by scaling factor Δ
- Round to image of $\sigma(R)$ and apply σ^{-1}

CKKS Encryption/Decryption

- Notation: $R = \mathbb{Z}[X]/X^N + 1$
- $KeyGen(1^\lambda)$: Fix N, Q, χ. Sample sparse, ternary $s \in R$.
- $Enc(1^\lambda, s, m)$: Sample $a \leftarrow R_Q, e \leftarrow \chi$. Output $(a, a * s + e + m)$.
- $Dec(s, ct = (a, b))$: Output $b - a * s$.

Can easily be made public-key

Fresh Ciphertext



CKKS Encryption/Decryption

- Decryption is $m(X) + e(X) \approx m(X)$

- Since $||\zeta|| = 1$, $||e(\zeta)||$ relatively small

- Error introduced in each plaintext slot is small

Homomorphic Addition

- Ciphertexts:
  - $(a\ , a * s + e + m) = (ct_0, ct_1) \in R_q^2$
  - $(a', a' * s + e' + m') = (ct'_0, ct'_1) \in R_q^2$

- Add both components:
  - $\left(a + a', (a + a') * s + (e + e') + (m + m')\right)$

- Valid encryption of $m + m'$

Homomorphic Multiplication

- Ciphertexts:
  - $(a\ , a * s + e + m) = (ct_0, ct_1) \in R_q^2$
  - $(a', a' * s + e' + m') = (ct'_0, ct'_1) \in R_q^2$

- Multiply:
  - $(ct_1 - ct_0 * s)(ct_1' - ct_0' * s) \approx m * m'$
  - $ct_1 * ct_1' - (ct_0 * ct_1' + ct_0' * ct_1) * s + (ct_0 * ct_0') * s^2 \approx m\ * m'$
  - Ciphertext is 3 elements: $(ct_0 * ct_0',\ \ ct_0 * ct_1' + ct_0' * ct_1,\ \ ct_1 * ct_1')$

- Valid encryption of $m * m'$

- To decrypt, compute s^2 from s

Homomorphic Multiplication

How can we prevent the ciphertext from increasing in size?

Key-Switching

- Convert a 3 element ciphertext to a 2 element ciphertext
- Main idea: Encrypt s^2 under s
- $(k_0, k_1) = (a, a * s + e + Ps^2) \in R_{PQ}^2$

Key-Switching

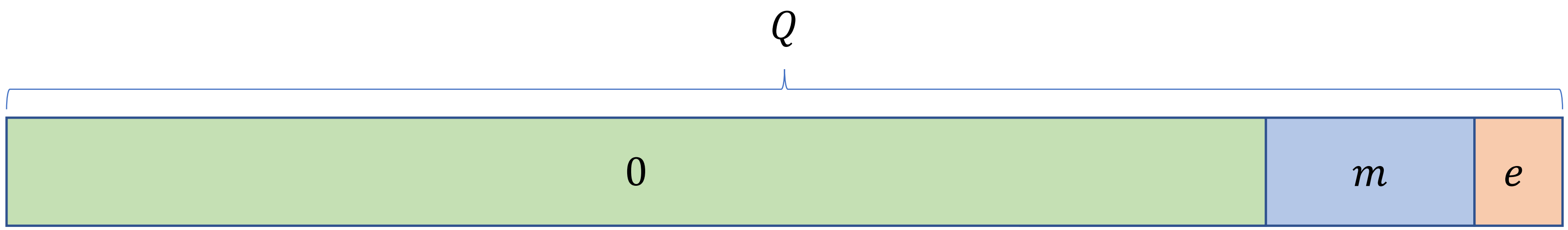
- Multiplied ciphertext: $(ct_0 * ct_0', \;\; ct_0 * ct_1' + ct_0' * ct_1, \;\; ct_1 * ct_1')$
  - Decrypts via $ct_1 * ct_1' - (ct_0 * ct_1' + ct_0' * ct_1) * s + (ct_0 * ct_0') * s^2 \approx m \; * m'$

- $(k_0, k_1) = (a, a * s + e + Ps^2) \in R_{PQ}^2$
  - Decrypts via $k_1 - k_0 * s \approx Ps^2$

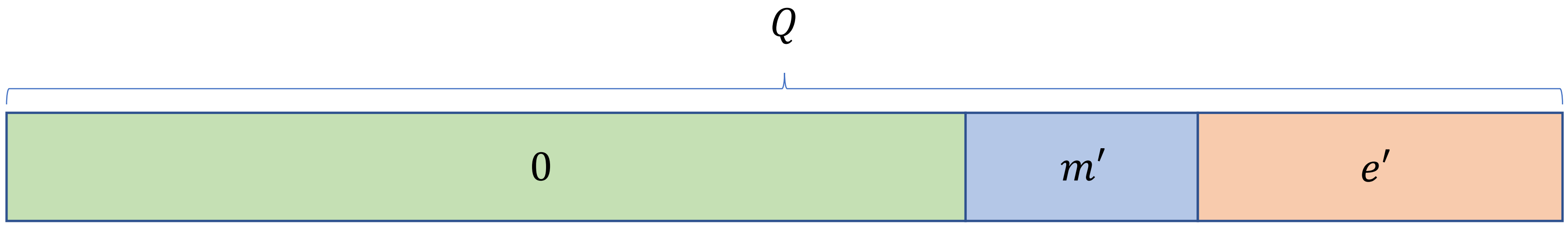
- $ct_1 * ct_1' - (ct_0 * ct_1' + ct_0' * ct_1) * s + (ct_0 * ct_0') * P^{-1}(k_1 - k_0 * s) \approx m \; * m'$

- Gives two element ciphertext: Add $\lfloor (ct_0 * ct_0') * P^{-1} k_1 \rceil$ to $ct_1 * ct_1'$, similarly for other term

Fresh Ciphertext



After Multiplication

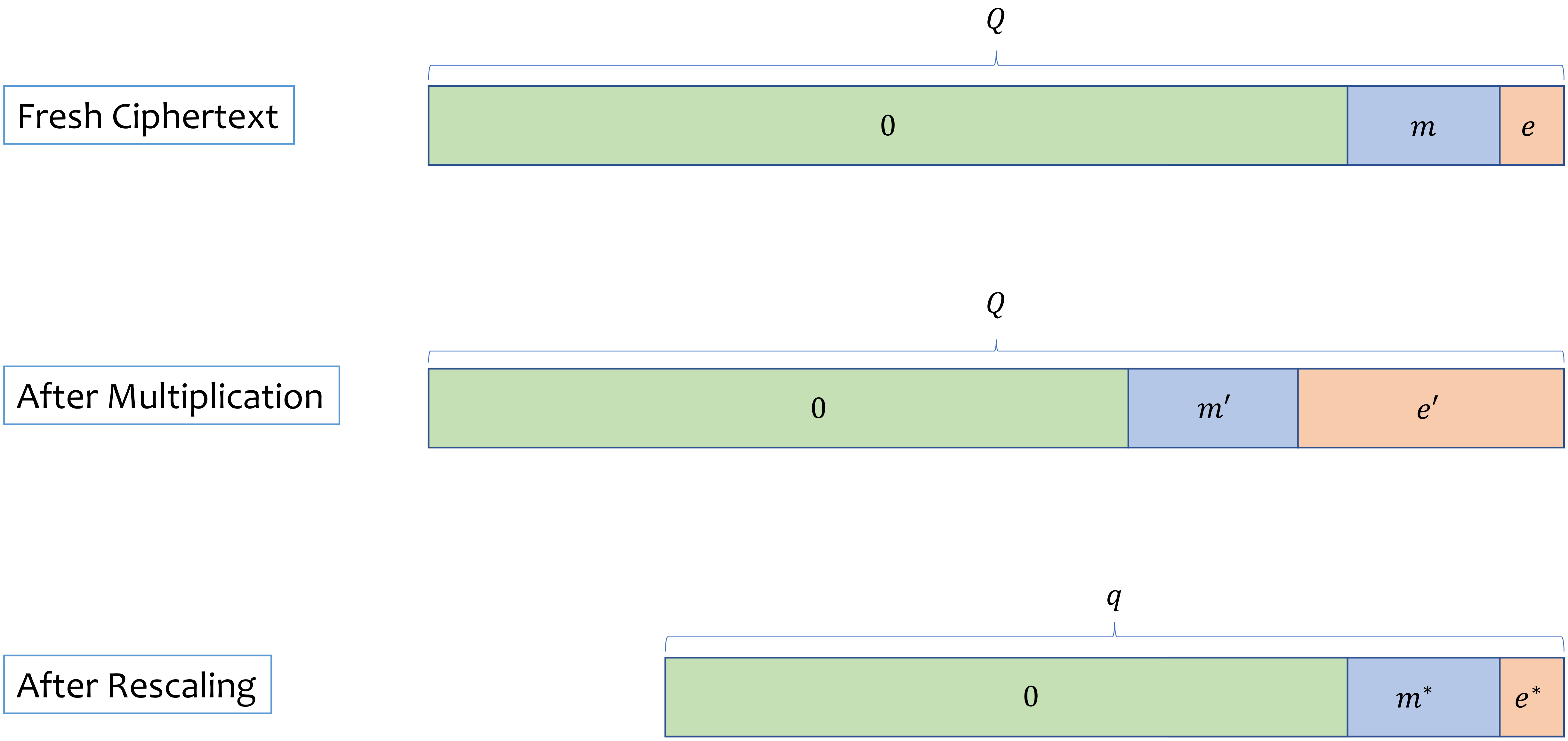


Rescaling

- Ciphertext: $(a\ , a * s + m + e) = (ct_0, ct_1) \in R_Q^2$
- Reduce ciphertext modulus to q and remove noisy LSBs of message
- $\left(\left\lfloor \frac{q}{Q} ct_0 \right\rceil, \left\lfloor \frac{q}{Q} ct_1 \right\rceil\right) \in R_q^2$
- Consider δ_0, δ_1 so that $ct_0 + \delta_0, ct_1 + \delta_1$ divisible by $\frac{Q}{q}$
- Decrypts to $m + e + (\delta_1 - \delta_0 * s)$

Rescaling

- Rescaled ct decrypts to $\frac{\mathrm{q}}{\mathrm{Q}}\left(m + e + (\delta_1 - \delta_0 * s)\right)$
- $\left(\frac{q}{Q}\right)\delta_0, \left(\frac{q}{Q}\right)\delta_1$ both polys with coefficients in $(-\frac{1}{2}, \frac{1}{2}]$
- s is sparse and ternary
- Overall rescaling error small


Fresh Ciphertext
Q
0
m
e
After Multiplication
Q
0
m'
e'
After Rescaling
q
0
m^*
e^*

Security

- Ciphertext $(a, a * s + m + e)$, decryption gives $m + e$
- IND-CPA follows from Ring-LWE
- Doesn’t consider public decryptions!
- [LM21] and [LMSS22] show attacks/fixes, introduce IND-CPA-D security

Plaintext Algebra

- For N a power of 2, Let ζ_{2N} be a primitive $2N$th root of unity (for example, $\zeta_{2N} = e^{\frac{2\pi i}{2N}}$)

- $\mathbb{Q}(\zeta_{2N})$ is a cyclotomic field

- $\mathbb{Q}(\zeta_{2N}) \cong \mathbb{Q}[X]/(X^N + 1)$

- $Gal(\mathbb{Q}(\zeta_{2N})/\mathbb{Q}) \cong \mathbb{Z}_{2N}^* \cong \mathbb{Z}_{\frac{N}{2}} \times \mathbb{Z}_2$

Plaintext Algebra

- $Gal(\mathbb{Q}(\zeta_{2N})/\mathbb{Q}) \cong \mathbb{Z}_{2N}^{*} \cong \mathbb{Z}_{\frac{N}{2}} \times \mathbb{Z}_2$
- $\mathbb{Z}_{2N}^{*}$ generated by 5 and -1.
- These correspond to the automorphisms $X \rightarrow X^5$ and $X \rightarrow X^{-1}$

Plaintext Algebra

- $R = \mathbb{Z}[X]/(X^N + 1)$ for N a power of 2.
- Decoding by evaluating $m(X)$ at primitive roots
- How should we order these roots?

Plaintext Algebra

- How should we order these roots?
- $\zeta, \zeta^5, \zeta^{5^2}, \dots, \zeta^{5^{\frac{N}{2}-1}}, \zeta^{-1}, \zeta^{-5}, \zeta^{-5^2}, \dots, \zeta^{-5^{\frac{N}{2}-1}}$
- Second half redundant since $\overline{m(\zeta)} = m(\bar{\zeta})$
- $m(X) \to [m(\zeta), m(\zeta^5), \dots, m\left(\zeta^{5^{\frac{N}{2}-1}}\right)]$

Plaintext Algebra

- $m(X) \rightarrow [m(\zeta), m(\zeta^5), \ldots, m\left(\zeta^{5^{\frac{N}{2}-1}}\right)]$

v_0	v_1	v_2	v_3	…	…	…	…	…	$v_{\frac{N}{2}-1}$

Plaintext Algebra

- $m(X) \to [m(\zeta), m(\zeta^5), \dots, m\left(\zeta^{5^{\frac{N}{2}-1}}\right)]$

Apply $\zeta \to \zeta^5$

v_1	v_2	v_3	…	…	…	…	…	$v_{\frac{N}{2}-1}$	v_0

Plaintext Algebra

- $m(X) \to [m(\zeta), m(\zeta^5), \ldots, m\left(\zeta^{5^{\frac{N}{2}-1}}\right)]$

Apply $\zeta \to \zeta^{-1}$

$\overline{v_0}$	$\overline{v_1}$	$\overline{v_2}$	$\overline{v_3}$	…	…	…	…	…	$\overline{v_{\frac{N}{2}-1}}$

Ciphertext Rotations

- Ciphertext is $(ct_0(X), ct_1(X)) \in R_q^2$ such that

$$ct_1(X) - ct_0(X) * s(X) = m(X) + e(X)$$

- Apply automorphism σ to get $(\sigma(ct_0(X)), \sigma(ct_1(X))) \in R_q^2$

$$\sigma\big(ct_1(X)\big) - \sigma\big(ct_0(X)\big) * \sigma\big(s(X)\big) = \sigma\big(m(X)\big) + \sigma\big(e(X)\big)$$

Ciphertext Rotations

- Apply automorphism σ to get $(\sigma(ct_0(X)), \sigma(ct_1(X))) \in R_q^2$

$$\sigma\big(ct_1(X)\big) - \sigma\big(ct_0(X)\big) * \sigma\big(s(X)\big) = \sigma\big(m(X)\big) + \sigma\big(e(X)\big)$$

- Decrypts to $\approx \sigma(m(X))$ but under key $\sigma(s(X))$

- Apply key-switching from $\sigma(s(X))$ to $s(X)$

Using CKKS

- Ciphertexts come with tagged info
- Scaling factor, upper bounds on message size and error
- Performance optimizations (Full-RNS etc.)
- Bootstrapping

Thank You!