Introduction to HEAAN (aka CKKS)

Yongsoo Song, Microsoft Research

Lattices: From Theory to Practice

Simons Workshop, April 30

Definition

- [Cheon-Kim-Kim-Song '17] Homomorphic Encryption for Arithmetic of Approximate Numbers

- HEAAN is NOT a homomorphic encryption scheme
  - $Dec\left(Enc(\boldsymbol{m})\right) \neq \boldsymbol{m}$
  - $Dec\left(\boldsymbol{ct}_1 * \boldsymbol{ct}_2\right) \neq Dec(\boldsymbol{ct}_1) * Dec(\boldsymbol{ct}_2)$

$$
\begin{tikzcd}
\boxed{ct_i} \arrow[r, "\text{Decrypt}"] \arrow[d, "F"'] \arrow[dr, phantom, "\circlearrowright"] & \boxed{m_i} \arrow[d, "F"'] \\
\boxed{ct = F(ct_i)} \arrow[r, "\text{Decrypt}"] & \boxed{F(m_i)}
\end{tikzcd}
$$

Definition

- [Cheon-Kim-Kim-Song ’17] Homomorphic Encryption for Arithmetic of Approximate Numbers

- HEAAN is an approximate homomorphic encryption scheme
  - $Dec\left(Enc(\boldsymbol{m})\right) \approx \boldsymbol{m}$
  - $Dec\left(\boldsymbol{ct}_1 * \boldsymbol{ct}_2\right) \approx Dec(\boldsymbol{ct}_1) * Dec(\boldsymbol{ct}_2)$
  - Noise bounds are determined by the parameter set

$$\begin{tikzcd} \boxed{ct_i} \arrow[r, "\text{Decrypt}"] \arrow[d, "F"'] \arrow[dr, phantom, "\circlearrowright"] & \boxed{m_i} \arrow[d, "F"'] \\ \boxed{ct = F(ct_i)} \arrow[r, "\text{Decrypt}"] & \boxed{F(m_i)} \end{tikzcd}$$

- This talk:
  - Construction (Leveled & Bootstrapping)
  - Pros and cons
  - Implementation & optimization
  - Subsequent works

Motivation

- Floating point representation

$$\pi \approx \underbrace{314}_{\text{significand}} * \underbrace{10^{-2}}_{\text{scaling factor (base}^{\text{exponent}}\text{)}}$$

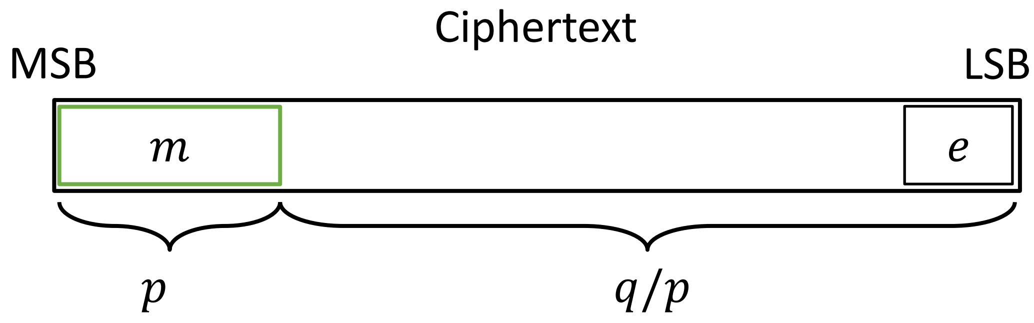
- Approximate arithmetic

$$(314 \ * 10^{-2}) * (314 \ * 10^{-2}) = 98596 \ * 10^{-4} \approx 986 \ * 10^{-2}$$

- The **rounding-off** operation makes a trade-off between accuracy and efficiency
  - Not represented as a low-degree polynomial

Learning with Errors

- Homomorphic Encryption candidate
    - $\mathrm{Dec}: \mathbb{Z}_q^{n+1} \to \mathbb{Z}_q, \quad \boldsymbol{c} \mapsto \langle \boldsymbol{c}, \boldsymbol{s} \rangle = m$



- LWE-based scheme
    - $\mathrm{Dec}: \mathbb{Z}_q^{n+1} \to \mathbb{Z}_q \to \mathbb{Z}_p$

      $\boldsymbol{c} \mapsto \langle \boldsymbol{c}, \boldsymbol{s} \rangle = \frac{q}{p} m + e \mapsto m$
    - Dec is approximately homomorphic
    - Exact computation over a discrete space modulo p

- **Main Idea:**
    - Consider the LWE noise as a part of numerical error in approximate computation
    - Support homomorphic rounding-off

Algorithms in HEAAN

- n : Ring dimension (power of two)
- $K = \mathbb{Q}[X]/(X^n + 1)$, $R = \mathbb{Z}[X]/(X^n + 1)$, $R_q = \mathbb{Z}_q[X]/(X^n + 1)$

- Homomorphic operations
  - Addition & Multiplication (relinearization)
  - Rescaling
  - Rotation
  - Complex conjugation

$$\underset{\mathbb{C}^{n/2}}{\boxed{\text{Message}}} \quad \underset{\text{decoding}}{\overset{\text{encoding}}{\rightleftarrows}} \quad \underset{R}{\boxed{\text{Plaintext}}} \quad \underset{\text{decryption}}{\overset{\text{encryption}}{\rightleftarrows}} \quad \boxed{\text{Ciphertext}}$$

Homomorphic operations
(encrypted computation)

Encoding & Decoding

- Canonical embedding

$$\sigma : K = \mathbb{Q}[X]/(X^n + 1) \to \mathbb{C}^n,\ \ \sigma(a) = \big(a(\zeta), a(\zeta^3), \dots, a(\zeta^{2n-1})\big)\ \text{ where } \zeta = \exp(\pi i/n).$$

$$\tau\ : K = \mathbb{Q}[X]/(X^n + 1) \to \mathbb{C}^{n/2},\ \ \tau(a) = \big(a(\zeta), a(\zeta^5), \dots, a(\zeta^{2n-3})\big).$$

- The precision of encoding is determined by the scaling factor $\Delta > 0$.

$\mathbb{C}^{n/2}$

$\boxed{\boldsymbol{m} = (m_0, m_1, \dots, m_{n/2-1})}$

Message vector

encoding = $\lfloor \Delta \cdot \tau^{-1}(\cdot) \rceil$ $\longrightarrow$

$\boxed{\Delta > 0}$ Scaling factor

$\longleftarrow$ decoding = $\Delta^{-1} \cdot \tau(\cdot)$

$R = \mathbb{Z}[X]/(X^n + 1)$

$\boxed{\mu(X)}$ such that $\tau(\mu) \approx \Delta \cdot \boldsymbol{m}$

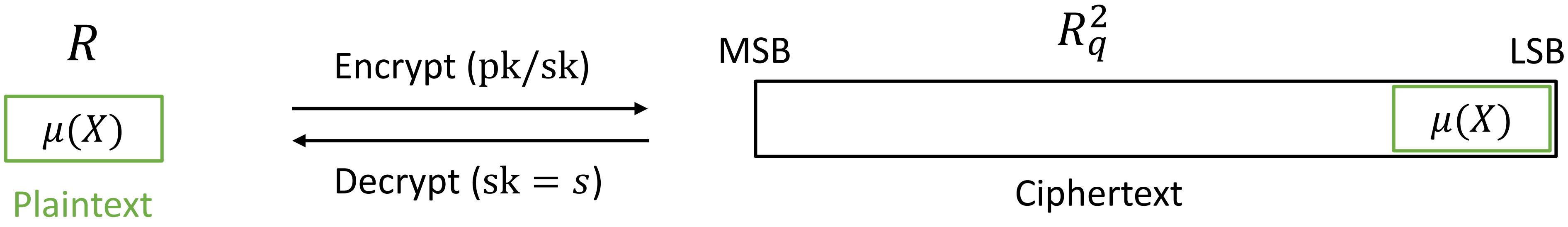
Plaintext

\# Toy example: $n = 4,\ \ \Delta = 10^2$

$$\boldsymbol{m} = (1 + 4i, 5 - 2i)\ \ \mapsto\ \ 3 + \frac{1}{\sqrt{2}}X + X^2 + \frac{5}{\sqrt{2}}X^3\ \ \mapsto\ \ \mu(X) = 300 + 71X + 100X^2 + 354X^3$$

$$\tau(\mu) = \big(\mu(\zeta), \mu(\zeta^5)\big) = (99.89.. + i * 400.52..,\ 500.11.. - i * 200.52..) \approx \Delta \cdot \boldsymbol{m}$$

Encrypt & Decrypt



- Enc: $\mu(X) \mapsto ct = (b + \mu, a) \in R_q^2$ for a random RLWE instance (b, a) s.t. $b + as = e$
  - Dec: $ct = (c_0, c_1) \ \mapsto \ c_0 + c_1 \cdot s \pmod{q}$
  - (Approx) Correctness: $\mathrm{Dec}\big(\mathrm{Enc}(\mu)\big) = \mu + e$ if $\|\mu + e\| < q/2$

- Notation: $ct(S) = c_0 + c_1 \cdot S \in R_q[S]$
  - $\mathrm{Dec}(ct) = ct(s) \pmod{q}$

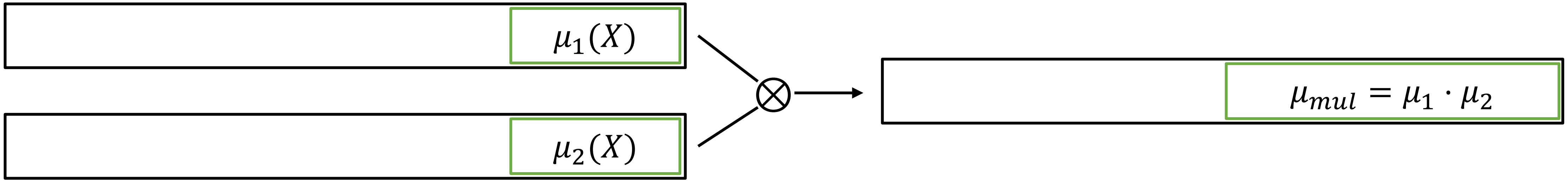
Arithmetic Operations

Given ct_i such that $ct_i(s) \approx \mu_i \pmod{q}$ and $\tau(\mu_i) \approx \Delta_i \cdot \boldsymbol{m}_i$

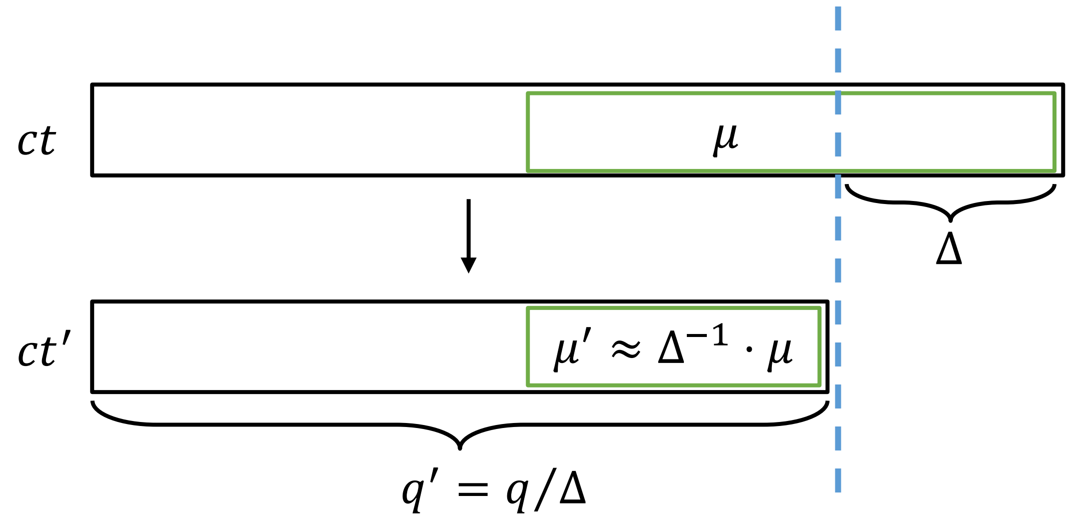
- $ct_{add} = ct_1 + ct_2 \pmod{q}$
  - Input should have the same scale $\Delta = \Delta_i$ to get a meaningful result

- $ct_{mul} = ct_1 \cdot ct_2 \pmod{q}$
  - The scaling factor is set to be $\Delta_{mul} = \Delta_1 \cdot \Delta_2$ so that $ct_{mul} \xmapsto{\text{decrypt}} \mu_1 \cdot \mu_2 \xmapsto{\text{decode w/ } \Delta_{mul}} \boldsymbol{m}_1 \odot \boldsymbol{m}_2$
  - $ct_{mul} = c_0 + c_1 S + c_2 S^2$ is quadratic

    Replace S^2 by relinearization key $\mathrm{rlk}(S) = k_0 + k_1 \cdot S$ such that $\mathrm{rlk}(s) \approx s^2$
  - Scaling factor increases rapidly during homomorphic evaluation



Rescaling



- Homomorphic 'rounding-off'
  - Usually performed after multiplication

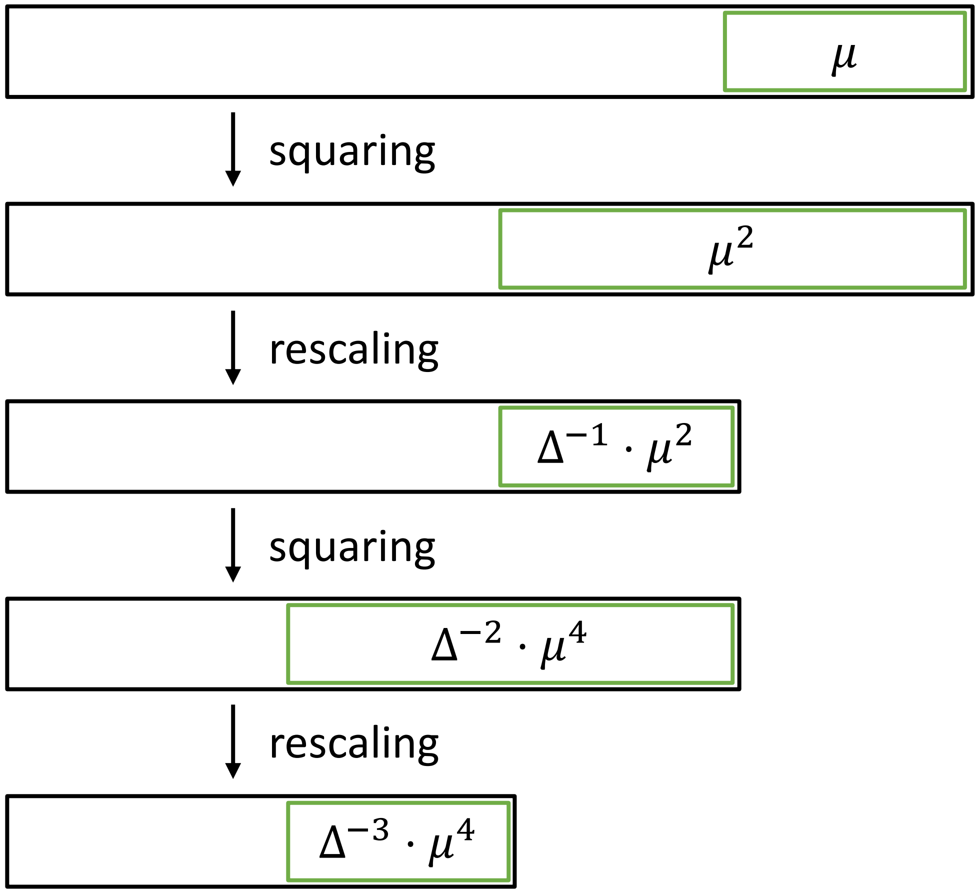
- Given $ct = c_0 + c_1 S \in R_q[S]$ of scale Δ^2,

  compute $ct' = \lfloor \Delta^{-1} \cdot ct \rceil \in R_{q'}[S]$ for $q' = q/\Delta$ and set its scale as Δ

- The underlying plaintext is (approximately) divided by Δ
  - $ct'(s) = \lfloor \Delta^{-1} \cdot c_0 \rceil + \lfloor \Delta^{-1} \cdot c_1 \rceil \cdot s \approx \Delta^{-1} \cdot (c_0 + c_1 \cdot s)$

- Plaintexts μ, μ' are encodings of the **same message** with different scaling factors
  - $\Delta^{-2} \cdot \tau(\mu) \approx \boldsymbol{m} \approx \Delta^{-1} \cdot \tau(\mu')$

Example: $\mathrm{F(x) = x^4}$



Ciphertext Modulus	Plaintext	Scaling Factor	Message
q	μ	Δ	$\boldsymbol{m}$
q	μ^2	Δ^2	$\boldsymbol{m}^2$
$q' = \Delta^{-1} \cdot q$	$\Delta^{-1} \cdot \mu^2$	Δ	$\boldsymbol{m}^2$
q'	$\Delta^{-2} \cdot \mu^4$	Δ^2	$\boldsymbol{m}^4$
$q'' = \Delta^{-1} \cdot q'$	$\Delta^{-3} \cdot \mu^4$	Δ	$\boldsymbol{m}^4$

Leveled HE

- Ciphertext modulus $q = p_0 \cdot \Delta^L$
  - Base modulus p_0 ($\gg \Delta$), $q_\ell = p_0 \cdot \Delta^\ell$ for $0 \le \ell \le L$
  - Ciphertext level is ℓ = Ciphertext modulus is q_ℓ

- Support a **fixed-point** style computation

- Other operations
  - Based on the evaluation of automorphism $X \mapsto X^k$ in $Gal(K/\mathbb{Q}) \approx \mathbb{Z}_{2n}^\times = \langle 5, -1 \rangle$

$$\tau\left(\mu\left(X^k\right)\right) = \left(\mu\left(\zeta^k\right), \mu\left(\zeta^{5k}\right), \dots, \mu\left(\zeta^{(2n-3)k}\right)\right)$$

  - If $c_0(X) + c_1(X) \cdot s(X) = \mu(X)$, then $c_0\left(X^k\right) + c_1\left(X^k\right) \cdot s\left(X^k\right) = \mu\left(X^k\right)$
  - $k = 5$: rotation on $\left\langle \zeta^5 \right\rangle = \{\zeta, \zeta^5, \dots, \zeta^{2n-3}\}$ (as well as plaintext slots)
  - $k = -1$: complex conjugate

From theory to practice

- First proof-of-concept implementation : the HEAAN library (Seoul National Univ.)
  - Modular q operation is expensive (NTL for high-precision arithmetic)

- [CHKKS18b] RNS-friendly parameter setting, inspired by [BEHZ16] Full RNS variant of FV
  - $q = p_0 \cdot p_1 p_2 \dots p_L$, for distinct primes $p_1, \dots, p_L$ and use the CRT representation
    The chain of ciphertext moduli determines the functionality of rescaling
  - 'Approximate basis' : find prime integers such that $p_\ell \approx \Delta$

- More than 5 libraries which are much of a muchness from theoretic perspective
  - Different choices of gadget decomposition for key-switching (relinearization)

	HEAAN	RNS-HEAAN	SEAL	Lattigo
Institute	SNU	SNU	Microsoft	EPFL
Decomposition	Trivial	Trivial	Prime	Hybrid
RNS friendly?	No	Yes	Yes	Yes

- Standardization in progress

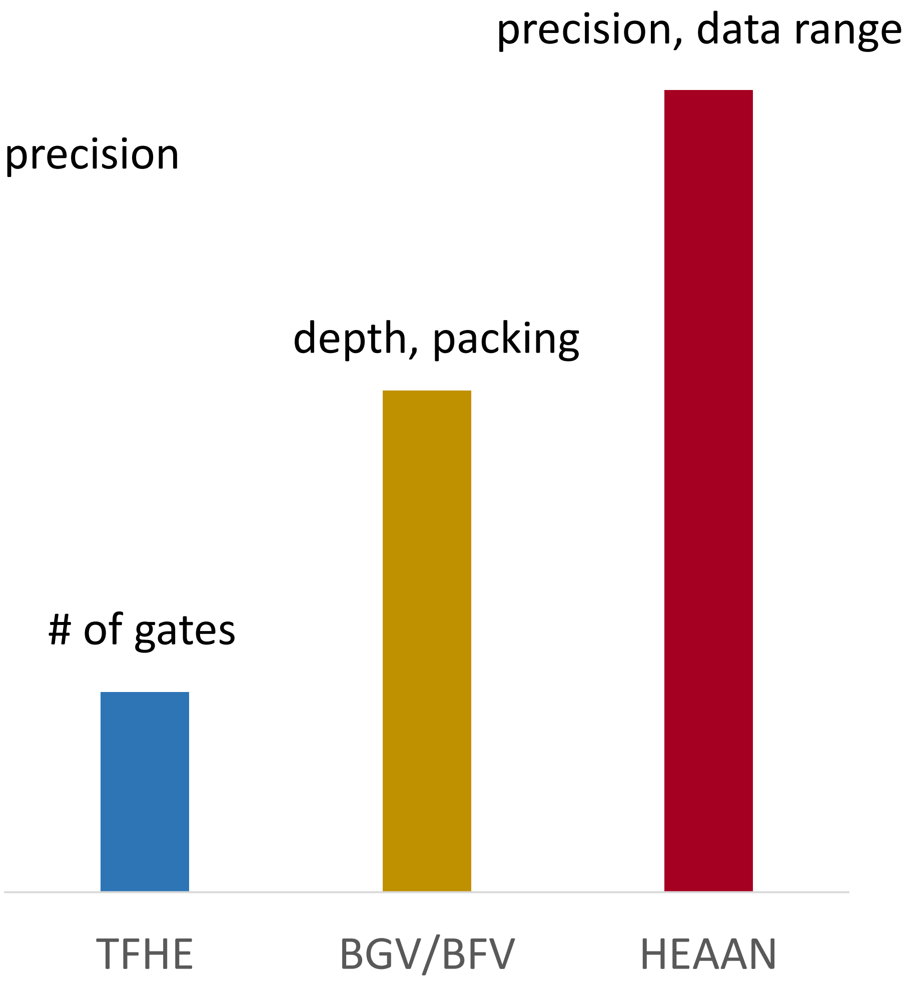
[Bajard-Eynard-Hasan-Zucca '16] A Full RNS Variant of FV like Somewhat Homomorphic Encryption Schemes
[Cheon-Han-Kim-Kim-Song '18] A Full RNS Variant of Approximate Homomorphic Encryption

Two sides of HEAAN

- Best known solution for encrypted real number arithmetic
  - $\log q = \log p_0 + L \cdot \log \Delta$ grows linearly with the depth and precision
  - Wide real-world applications

- Evaluation of analytic functions
  - Multiplicative inverse, sigmoid, etc.

× Difficult-to-learn, hard-to-optimize
  - Security, scaling factor, precision, depth, packing, data size, …
  - Polynomial approximation of a target function
  - Huge performance gap between
    fully/poorly optimized implementation



Definition and necessity [CHKKS18a]

- Bootstrapping of HE
  - Given ct such that $\mathrm{Dec}_{sk}(ct) = \boldsymbol{m}$, let $F(x) = \mathrm{Dec}_x(ct)$
  - $ct' \coloneqq F\big(\mathrm{Enc}(\mathrm{sk})\big) = \mathrm{Enc}\big(F(\mathrm{sk})\big) = \mathrm{Enc}(\boldsymbol{m})$ refreshes the (noise) level

- Q1. What is bootstrapping of approximate HE?
  - $ct' \coloneqq F\big(\mathrm{Enc}(\mathrm{sk})\big) \approx \mathrm{Enc}\big(F(\mathrm{sk})\big) = \mathrm{Enc}(\boldsymbol{m})$
  - Adding a sufficiently small error is acceptable

- Q2. Why do we need approximate bootstrapping?
  - Numerically stable circuits
  - e.g. negative feedback in control systems, convergence property of ML training algorithms

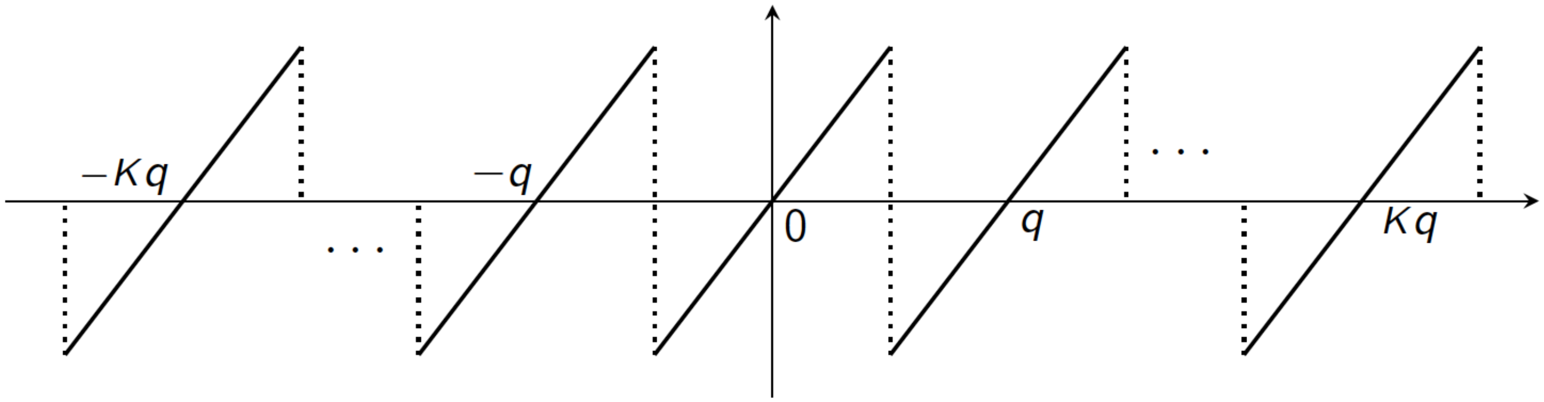
[Cheon-Han-Kim-Kim-Song '18] Bootstrapping for approximate homomorphic encryption
[Chen-Chillotti-Song '19] Improved bootstrapping for approximate homomorphic encryption
[Han-Ki '20] Better bootstrapping for approximate homomorphic encryption

Main Idea

- Dec: $ct \mapsto t = c_0 + c_1 \cdot s \mapsto [t]_q = \mu$
  - $t = qI + \mu$ for some small $\|I\| < K$

- Step 1 : Raise the modulus up to $Q \gg q$
  - $\mathrm{Dec}(ct) = [c_0 + c_1 \cdot s]_Q = t$

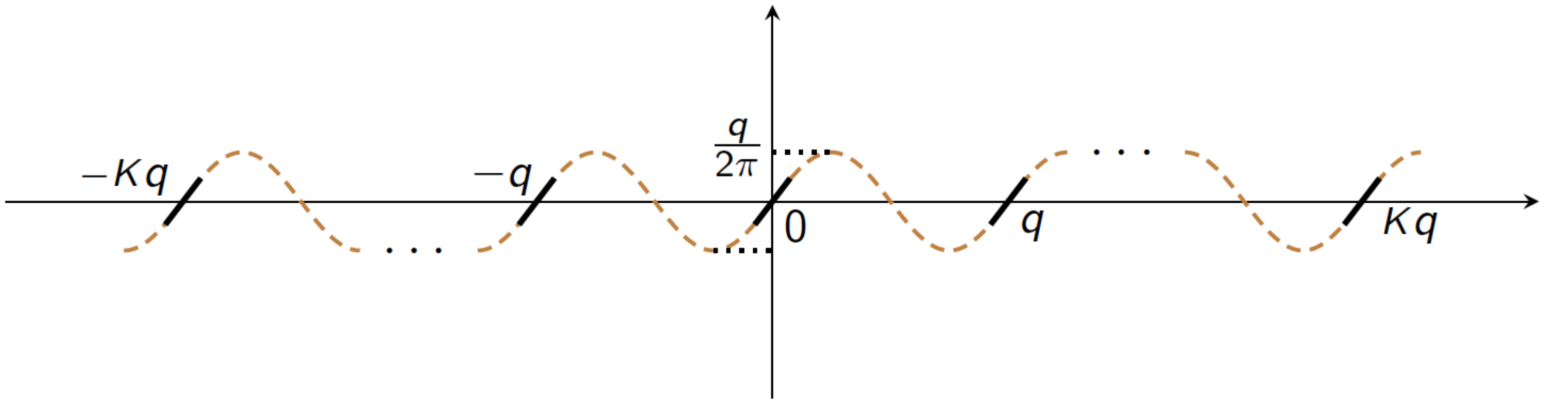
- Step 2: Homomorphically evaluate the reduction modulo q function



Step 2: Modular reduction

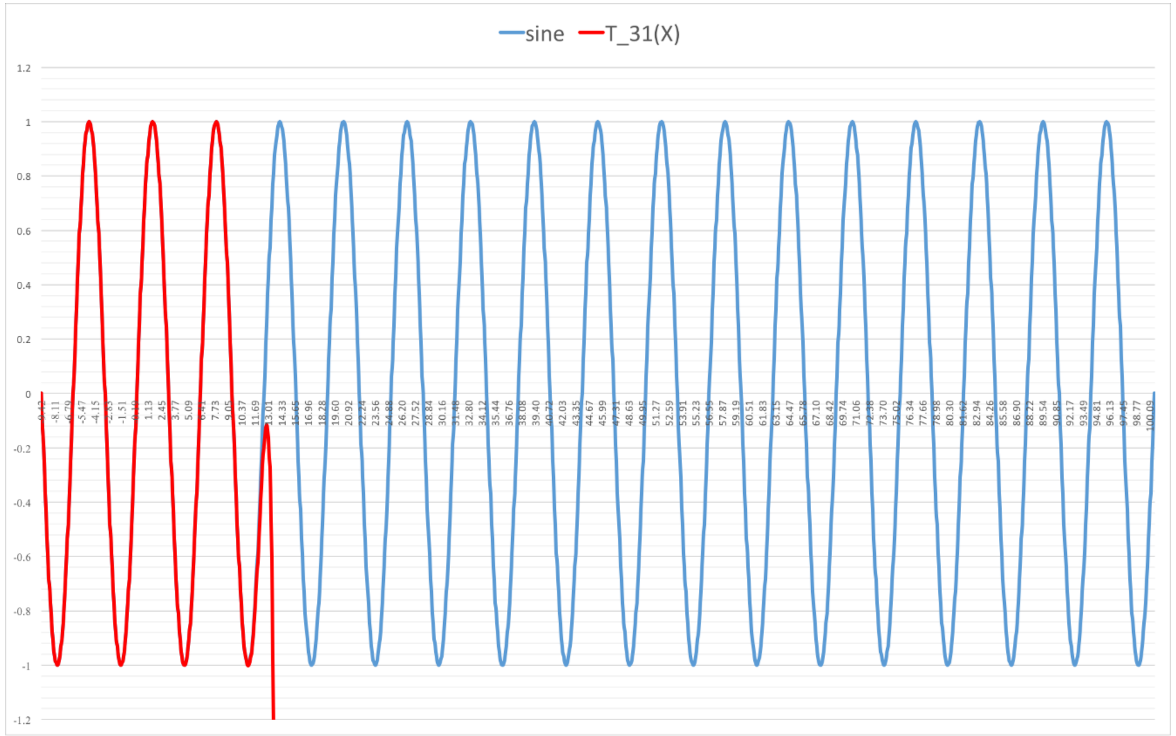
- $t \mapsto [t]_q$ is not continuous
  - Cannot be approximated by a polynomial

- Assume that $t = qI + \mu$ for some $|\mu| < B \ll q$
  - Restrict the domain of the function to $\cup_{|k| \le K}(qk - B, qk + B)$
  - Precisely approximated by the sine function $[t]_q \approx \frac{q}{2\pi} \sin \theta$ for $\theta = 2\pi t/q$



Step 2: sine evaluation

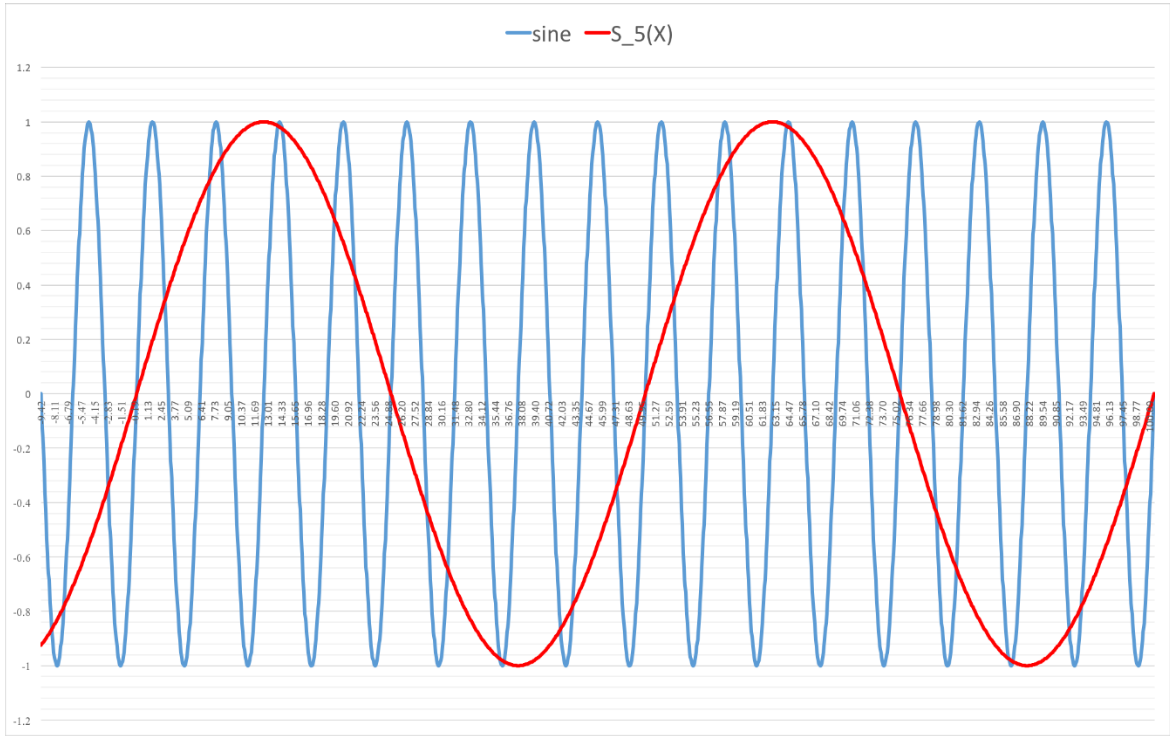
- Naïve approach: Taylor expansion
  - Require a large degree
  - Numerically unstable power representation



Step 2: sine evaluation

- Naïve approach: Taylor expansion
  - Require a large degree
  - Numerically unstable power representation

- [CHKKS18a] Double-angle formula
  - $\exp(i\theta/2^r) = \cos(\theta/2^r) + \sin(\theta/2^r)$ for $r > 0$
    (Small degree approximation is available)
  - Repeat squaring r times to obtain $\exp(i\theta)$
  - Extract its imaginary part



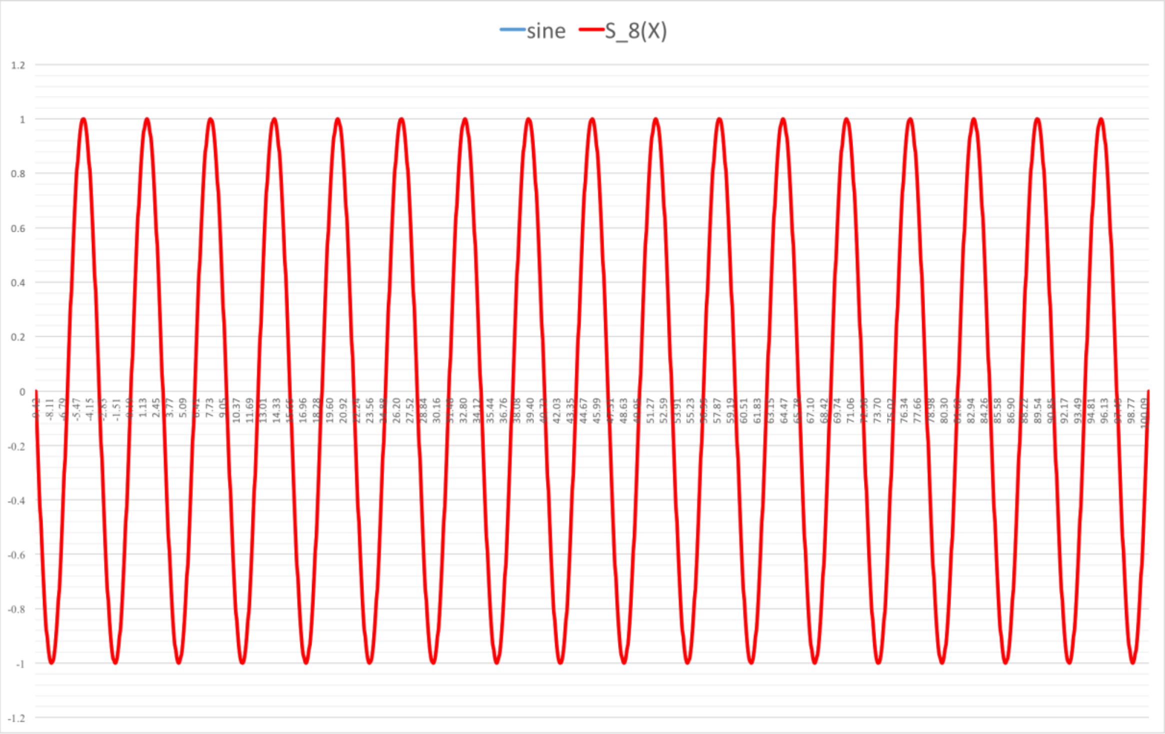
Step 2: sine evaluation

- Naïve approach: Taylor expansion
  - Require a large degree
  - Numerically unstable power representation

- [CHKKS18a] Double-angle formula
  - $\exp(i\theta/2^r) = \cos(\theta/2^r) + \sin(\theta/2^r)$ for $r > 0$

    (Small degree approximation is available)
  - Repeat squaring r times to obtain $\exp(i\theta)$
  - Extract its imaginary part

- [CCS19] Chebyshev approximation method
  - Almost optimal depth consumption
  - Efficient & numerically stable evaluation algorithm



Pre- and post-processing

Step 1 : Raise the modulus up to $Q \gg q$, $\quad \mathrm{Dec}(ct) = [c_0 + c_1 \cdot s]_Q = t$

Step 1.5: Move the coefficients $t_i = qI_i + \mu_i$ into the plaintext slots

Step 2: Homomorphically evaluate $t = qI + \mu \mapsto [t]_q = \mu$

Step 2.5 : Bring the values μ_i back to the coefficients

- Step 1.5 and 2.5 are homomorphic evaluation of encoding/decoding function (τ and τ^{-1})

- [CHKKS18] General BSGS method for linear transformation
  - Optimal in terms of depth, but expensive

- [CCS19] FFT-style algorithm using the property of τ
  - Fine trade-off between complexity and depth (3~4 are enough in practice)

Conclusion

- Defined and designed approximate HE and its bootstrapping
  - Asymptotic/practical performance improvement

- Numerical analysis + cryptographic knowledge for optimization
  - Need more studies on efficient polynomial approximation and evaluation
  - Higher-level API to provide better usability for general engineers

- Open questions
  - Build cryptographic protocol on the top of HEAAN
  - Previous techniques (e.g. noise flooding, circuit privacy) for HE do not apply