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
  - $Dec(Enc(m)) \neq m$
  - $Dec(ct_1 \cdot ct_2) \neq Dec(ct_1) \cdot Dec(ct_2)$
Definition

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

- **HEAAN** is an approximate homomorphic encryption scheme
  - \( \text{Dec} (\text{Enc}(m)) \approx m \)
  - \( \text{Dec}(ct_1 \times ct_2) \approx \text{Dec}(ct_1) \times \text{Dec}(ct_2) \)
  - Noise bounds are determined by the parameter set

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

- Floating point representation
  \[ \pi \approx 314 \times 10^{-2} \]
  significand scaling factor (base\(^{\text{exponent}}\))

- Approximate arithmetic
  \[ (314 \times 10^{-2}) \times (314 \times 10^{-2}) = 98596 \times 10^{-4} \approx 986 \times 10^{-2} \]

- 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
  - Dec : \( \mathbb{Z}_q^{n+1} \rightarrow \mathbb{Z}_q, \ c \mapsto \langle c, s \rangle = m \)

- LWE-based scheme
  - Dec : \( \mathbb{Z}_q^{n+1} \rightarrow \mathbb{Z}_q \rightarrow \mathbb{Z}_p \)
    - \( c \mapsto \langle c, s \rangle = \frac{q}{p} m + e \mapsto m \)
  - 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

Message $\mathbb{C}^{n/2}$ encoding decoding Plaintext $R$ encryption decryption Ciphertext

Homomorphic operations (encrypted computation)
## Encoding & Decoding

- **Canonical embedding**
  \[\sigma : K = \mathbb{Q}[X]/(X^n + 1) \to \mathbb{C}^n, \quad \sigma(a) = (a(\zeta), a(\zeta^3), \ldots, a(\zeta^{2n-1})) \quad \text{where} \quad \zeta = \exp(\pi i/n).\]
  \[\tau : K = \mathbb{Q}[X]/(X^n + 1) \to \mathbb{C}^{n/2}, \quad \tau(a) = (a(\zeta), a(\zeta^5), \ldots, a(\zeta^{2n-3})).\]

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

### Toy example:

\[n = 4, \quad \Delta = 10^2\]

\[m = (1 + 4i, 5 - 2i) \leadsto 3 + \frac{1}{\sqrt{2}}X + X^2 + \frac{5}{\sqrt{2}}X^3 \quad \leadsto \quad \mu(X) = 300 + 71X + 100X^2 + 354X^3\]

\[\tau(\mu) = (\mu(\zeta), \mu(\zeta^5)) = (99.89.. + i \times 400.52.., 500.11.. - i \times 200.52..) \approx \Delta \cdot m\]
Encrypt & Decrypt

- **Enc**: $\mu(X) \rightarrow ct = (b + \mu, a) \in R^2_q$ 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: $\text{Dec}(\text{Enc}(\mu)) = \mu + e$ if $||\mu + e|| < q/2$

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

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

- $ct_{add} = ct_1 + ct_2 \pmod{q}$
  - 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 $ct_{mul} \mapsto \mu_1 \cdot \mu_2 \mapsto m_1 \odot m_2$
  - $ct_{mul} = c_0 + c_1 S + c_2 S^2$ is quadratic
    - Replace $S^2$ by relinearization key $rlk(S) = k_0 + k_1 \cdot S$ such that $rlk(s) \approx s^2$
  - Scaling factor increases rapidly during homomorphic evaluation
Rescaling

- Homomorphic ‘rounding-off’
  - Usually performed after multiplication

- Given \( ct = c_0 + c_1S \in R_q[S] \) of scale \( \Delta^2 \),
  compute \( ct' = [\Delta^{-1} \cdot ct] \in R_{q'}[S] \) for \( q' = q/\Delta \) and set its scale as \( \Delta \)

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

- Plaintexts \( \mu, \mu' \) are encodings of the same message with different scaling factors
  \( \Delta^{-2} \cdot \tau(\mu) \approx m \approx \Delta^{-1} \cdot \tau(\mu') \)
Example: $F(x) = x^4$

<table>
<thead>
<tr>
<th>Ciphertext Modulus</th>
<th>Plaintext</th>
<th>Scaling Factor</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>$\mu$</td>
<td>$\Delta$</td>
<td>$m$</td>
</tr>
<tr>
<td>$q^2 = \mu^2$</td>
<td>$\mu^2$</td>
<td>$\Delta^2$</td>
<td>$m^2$</td>
</tr>
<tr>
<td>$q' = \Delta^{-1} \cdot q$</td>
<td>$\Delta^{-1} \cdot \mu^2$</td>
<td>$\Delta$</td>
<td>$m^2$</td>
</tr>
<tr>
<td>$q'' = \Delta^{-2} \cdot \mu^4$</td>
<td>$\Delta^{-2} \cdot \mu^4$</td>
<td>$\Delta^2$</td>
<td>$m^4$</td>
</tr>
<tr>
<td>$q''' = \Delta^{-3} \cdot \mu^4$</td>
<td>$\Delta^{-3} \cdot \mu^4$</td>
<td>$\Delta$</td>
<td>$m^4$</td>
</tr>
</tbody>
</table>
Leveled HE

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

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

- Other operations
  - Based on the evaluation of automorphism $X \mapsto X^k$ in $Gal(K/\mathbb{Q}) \approx \mathbb{Z}_{2n}^\times = \langle 5, -1 \rangle$
    
    $$\tau(\mu(X^k)) = (\mu(\zeta^k), \mu(\zeta^{5k}), ..., \mu(\zeta^{(2n-3)k}))$$

  - If $c_0(X) + c_1(X) \cdot s(X) = \mu(X)$, then $c_0(X^k) + c_1(X^k) \cdot s(X^k) = \mu(X^k)$

  - $k = 5$ : rotation on $\langle \zeta^5 \rangle = \{\zeta, \zeta^5, ..., \zeta^{2n-3}\}$ (as well as plaintext slots)
  - $k = -1$ : complex conjugate
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
  - $q = p_0 \cdot p_1 p_2 \ldots p_L$, for distinct primes $p_1, \ldots, p_L$ 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)

Standardization in progress

<table>
<thead>
<tr>
<th>Institute</th>
<th>HEAAN</th>
<th>RNS-HEAAN</th>
<th>SEAL</th>
<th>Lattigo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decomposition</td>
<td>Trivial</td>
<td>Trivial</td>
<td>Prime</td>
<td>Hybrid</td>
</tr>
<tr>
<td>RNS friendly?</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

[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 $\text{Dec}_{sk}(ct) = m$, let $F(x) = \text{Dec}_x(ct)$
  - $ct’ := F(\text{Enc}(sk)) = \text{Enc}(F(sk)) = \text{Enc}(m)$ refreshes the (noise) level

- Q1. What is bootstrapping of approximate HE?
  - $ct’ := F(\text{Enc}(sk)) \approx \text{Enc}(F(sk)) = \text{Enc}(m)$
  - 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: \( ct \mapsto t = c_0 + c_1 \cdot s \mapsto [t]_q = \mu \)
  - \( t = ql + \mu \) for some small \( \|I\| < K \)

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

- Step 2: Homomorphically evaluate the reduction modulo \( q \) function

\[
\begin{align*}
-\kappa q & \quad \cdots \quad -q \\
\cdots & \quad 0 \quad q \quad \cdots \\
& \quad \kappa q
\end{align*}
\]
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 $\bigcup_{|k| \leq K} (qk - B, qk + B)$
  - 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 $Q \gg q$, \[ \text{Dec}(ct) = [c_0 + c_1 \cdot s]_Q = t \]

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 $\mu_i$ back to the coefficients

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

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

- [CCS19] FFT-style algorithm using the property of $\tau$
  - 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