Toolkits of HEAAN and Privacy-Preserving Machine Learning

Jung Hee Cheon (SNU, CryptoLab)

Thanks to YongSoo Song, Kiwoo Lee, Andrey Kim, Duhyeong Kim

HEAAN = 慧眼 = Insightful Minds

Homomorphic Encryption for Arithmetic of Approximate Numbers

Jung Hee Cheon[1], Andrey Kim[1], Miran Kim[2], and Yongsoo Song[1]

[1] Seoul National University, Republic of Korea
{jhcheon, kimandrik, lucius05}@snu.ac.kr
[2] University of California, San Diego
mrkim@ucsd.edu

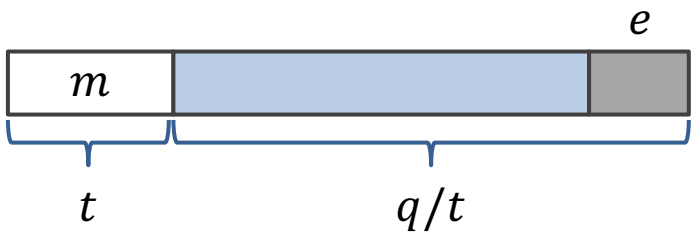
Abstract. We suggest a method to construct a homomorphic encryption scheme for approximate arithmetic. It supports an approximate addition and multiplication of encrypted messages, together with a new *rescaling* procedure for managing the magnitude of plaintext. This procedure truncates a ciphertext into a smaller modulus, which leads to rounding of plaintext. The main idea is to add a noise following significant figures which contain a main message. This noise is originally added to the plaintext for security, but considered to be a part of error occurring during approximate computations that is reduced along with plaintext by rescaling. As a result, our decryption structure outputs an approximate value of plaintext with a predetermined precision.

[CKKS, AC17] Homomorphic Encryption for Arithmetic of Approximate Numbers
https://eprint.iacr.org/2016/421.pdf

Approximate Computation

- Numerical Representation
  - Encode m into an integer $m \approx px$ for a scaling factor p : $\sqrt{2} \mapsto 1412 \approx \sqrt{2} \cdot 10^3$
  - Fixed - Point Multiplication
    - Compute $m = m_1 m_2$ and extract its significant digits $m' \approx p^{-1} \cdot m$
      : $1.234 \times 5.678 = (1234 \cdot 10^{-3}) \times (5678 \cdot 10^{-3}) = 7006652 \cdot 10^{-6} \mapsto 7007 \cdot 10^{-3} = 7.007$

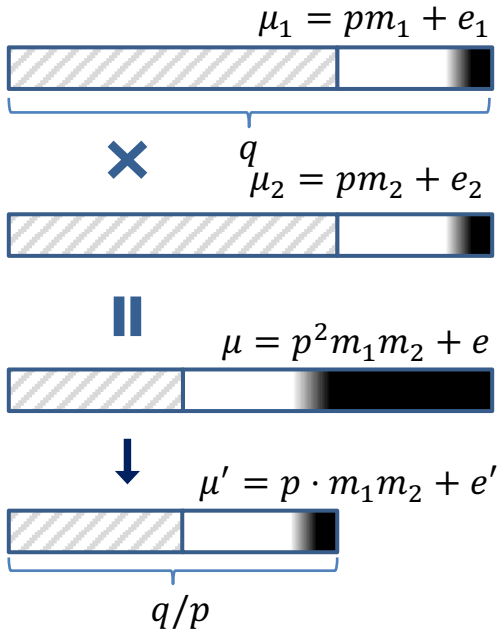
- Previous HE on LWE problem (Regev, 2005)
  - $\mathrm{ct} = \mathrm{Enc}_{\mathrm{sk}}(m)$, $\langle \mathrm{ct}, \mathrm{sk} \rangle = \frac{q}{t} m + e \pmod{q}$
  - Modulo t plaintext vs Rounding operation



HEAAN

- A New Message Encoding
  - $\mathrm{ct} = \mathrm{Enc}_{\mathrm{sk}}(m), \langle \mathrm{ct}, \mathrm{sk} \rangle = pm + e \pmod{q}$
  - Consider e as part of approximation error
- Homomorphic Operations

Input	$\mu_1 \approx p \cdot m_1$, $\mu_2 \approx p \cdot m_2$
Addition	$\mu_1 + \mu_2 \approx p \cdot (m_1 + m_2)$
Multiplication	$\mu = \mu_1 \mu_2 \approx p^2 \cdot m_1 m_2$
Rounding	$\mu' \approx p^{-1} \cdot \mu \approx p \cdot m_1 m_2$



- Support for the (approximate) fixed-point arithmetic
  - **Leveled** HE : $q = p^L$

Toolkit for Homomorphic Computation

How to Pack a Matrix

❖ Packing Method

- HEAAN supports vector operations
- How can we compute matrix operations for ciphertexts?

- Matrix Encoding method

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Encode

$c = Enc($ | 1 | 2 | ⋯ | 16 | $)$

How to rotate

- Packing Method
  - Matrix addition : trivial
  - Matrix multiplication : non-trivial (exercise)
  - Row/column rotation?

$rot(c, 1) = Enc($ | 16 | 1 | … | 15 | $) =$

16	1	2	3
4	5	6	7
18	9	10	11
12	13	14	15

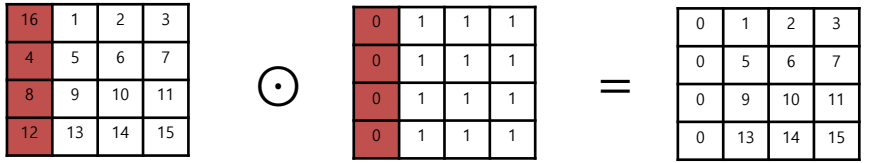
(wrong)

How to rotate

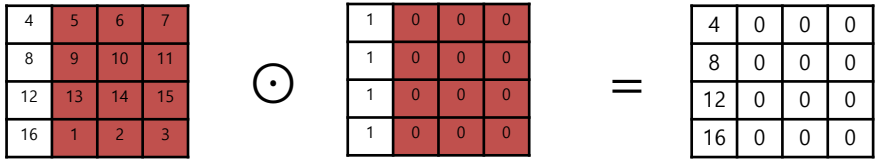
❖ Packing Method

- Solution : using masking vector

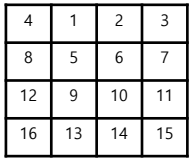
$$rot(c,1) \odot mask =$$



$$rot(c,-3) \odot mask' =$$



$$\therefore ColRot\ (c) = rot(c,1) \odot mask + rot(c,-3) \odot mask' =$$



Polynomial Approximation

❖ Message Space

- Bit-wise HE: $\mathbb{Z}_2$={0,1} with logical gates
  - Good at gate operations but slow at arithmetic.
- Word-wise HE: $\mathbb{C}$ or $\mathbb{Z}_p$ with add. & mult.
  - Good at poly evaluation but hard to evaluate non-poly function.

❖ Idea: Use Polynomial Approximation for Non-Poly Functions!

- Don't use naïve Taylor Approx. It is local i.e. approx at a point
- Minimize errors on an interval
- Methods: Least Square, Chevyshev, Minimax …

Polynomial Approximation

Application : Homomorphic Logistic Regression [HHCP19] (IAAI'19)
- Need to evaluate sigmoid function : 1/(1+exp(-x))

- Real Financial Large Data (Provided by Korea Credit Bureau)
- About 400,000 samples with 200 features.
- Target value = "credit score"

# Iterations	200	Accuracy	80%	Public Key	Encrypted Block	Total	Time / Iter
Learning Rate	0.01	AUROC	0.8	≈ 2 GB	4.87 MB	1060 min	5.3 min
Mini-batch Block Size	512	K-S value	50.84				

Matrix by Vector Multiplication

❖ Application : Homomorphic DNN & RNN Classification
- Need to evaluate matrix & vector multiplication

❖ Idea : Use a matrix-variant of HEAAN with different ring structure

- DNN Classification with FC (Fully-Connected) layers
- RNN Classification with GRU (Gated Recurrent Unit) and FC layers

Structure	Scheme	Running time	# FC layers	# GRU layers
DNN	MHEAAN	3.6 min	5	
RNN	MHEAAN	142 min	2	40

Comparison

❖ Idea: $Comp(a,b) = \lim_{k\to\infty} \frac{a^k}{a^k+b^k}$ for $a, b > 0$ (To Appear in Asiacrypt'19)

1. Compute approximately (Take only MSBs of the results)
2. Choose k as a power of 2
3. Use iterative algorithms for division

Method	Scheme	(Amortized) Running time	# of pairs	Error
Bit-wise	HElib	≈ 1 ms	1800	0
	TFHE	≈ 1 ms	-	
Ours (Word-wise)	**HEAAN**	**0.45 ms**	2^{16}	$< 2^{-8}$

Max

- General Max Algorithms

$$Maxidx(a_1, \dots, a_t) = \lim_{k \to \infty} \left(\frac{a_1^k}{{a_1}^k + \dots + {a_t}^k}, \dots, \frac{a_t^k}{{a_1}^k + \dots + {a_t}^k} \right) \text{ for } a_i > 0$$

$$Max(a_1, \dots, a_t) = \lim_{k \to \infty} \frac{{a_1}^{k+1} + \dots + {a_t}^{k+1}}{{a_1}^k + \dots + {a_t}^k} \text{ for } a_i > 0$$

- 2nd $Max(a_1, \dots, a_t) = Max\left((a_1, \dots, a_t) \odot \left((1, \dots, 1) - Maxidx(a_1, \dots, a_t)\right)\right)$

- Applications: Top-k-max, threshold counting, clustering, …

Sorting on Encrypted Data

- Fast Merge Sort: $O(k \log k)$
  - Comparison-based algorithm doesn't work on HE
  - We cannot check min-max condition

- Sorting Network: $O(k \log^2 k)$
  - Comparison network that always sort their inputs
  - Data-independent algorithm

- Results
  - 64 slots : about 12 min. (previous work : 42 min. [EGN+15])
  - 32,768 slots : about 10.5 hours (previous work: impractical)

Homomorphic Machine Learning (HEML)

ML Type	Task	Name	Publication	HE	Remarks
Supervised Learning	Neural Network (Prediction)	CryptoNets	ICML'16	SEAL	- First Privacy-Preserving ML based on HE - CNN on MNIST (99.0% accuracy) / 570 sec
		Gazelle	USENIX'18	*Additive HE*	- Hybrid of Additive HE & MPC - Fast but Requires Communication
		[BMMP18]	CRYPTO'18	TFHE	- DiNN on MNIST : Deep Discretized Neural Network - 96.3% accuracy / 1.6 sec
		[JKLS18]	CCS'18	**HEAAN**	- CNN on MNIST (98.1% accuracy) - 30 sec (Amortized : 0.45 sec)
	Logistic Regression (Training)	[KSK+18]	BMC Journal	**HEAAN**	- iDASH'17 Winner, Genomic data - 1579 samples / 18 features / 7 min
		[HHCP19]	IAAI'19	**HEAAN**	- Large-scale Financial data with Bootstrapping - 422108 samples / 200 features / 18 hours
Unsupervised Learning	Clustering	[JA18]	SAC'18	TFHE	- K-means Algorithm
		[CKP19]	SAC'19	**HEAAN**	- Mean-Shift Algorithm (400 times faster than the above)

Ongoing Works on HEML

- Basic Tools:
  - Packing, Matrix Operation, Comparison, Approximate inv/sigmoid,

- Decision Tree (Random Forest): Packing, Comparison
- Gradient Boosting: Comparison, Gradient Decent

- Deep Neural Network: Fast matrix operations + Approximate
- Convolution Neural Network: + Comparison (Max-pool)
- Recurrent Neural Network: Fast Bootstrapping (A number of iterations)

Thank you for your attention!

감사합니다.