Approximate Homomorphic Encryption and Privacy Preserving Machine Learning

Jung Hee Cheon (SNU, CryptoLab)

Thanks to YongSoo Song, Kiwoo Lee, Andrey KIM

Outline

1. Homomorphic Encryption
2. HEAAN
3. Bootstrapping of HEAAN
4. Toolkit for Homomorphic Computation

Homomorphic Encryption

❖ Integer-based HE scheme

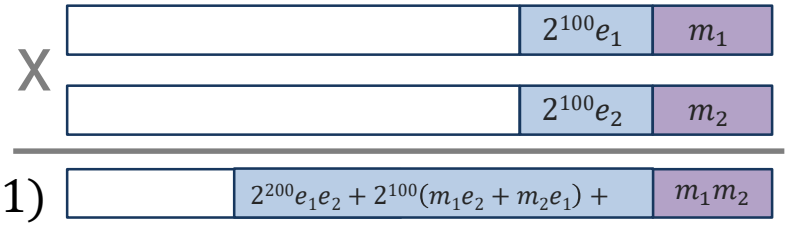
– RAD PH [1]

- (Secret Key, Operation Key) = (a large prime p, $n = pq_0$)
- Encryption: $Enc(m) = m + pq \ (mod\ n)$
- Decryption: $Enc(m)\ (mod\ p) = m$
- $Enc(m_1) + Enc(m_2) = (m_1 + pq_1) + (m_2 + pq_2)$
  $= (m_1 + m_2) + p(q_1 + q_2)$
  $= Enc(m_1 + m_2)$

But, INSECURE!

– DGHV HE scheme (on $\mathbb{Z}_2$) [2]

- $Enc(m \in Z_{2^{100}}) = m + 2100e + pq$
- SECURE against quantum computing
- Use a polynomial ring $R_q = \mathbb{Z}_q[x]/(xn + 1)$



Fully Homomorphic Encryption

- On Polynomials (RingLWE)
  - [Gen09] ideal lattice
  - NTRU: LTV12
  - Ring-LWE: BV11b, GHS13, BLLN13, HEAAN etc
- On Integers (AGCD)
  - [DGHV10] FHE over the Integers. Eurocrypt 2010
  - CMNT11, CNT12, CCKLLTY13, CLT14, etc
- On Matrices (LWE)
  - [BV11a] Efficient FHE from (Standard) LWE. FOCS11
  - Bra12, BGV12, GSW13

Summary of Progress in HE

1. **2009~2012: Plausibility and Scalable for Large Circuits**
   - [GH11] A single bit bootstrapping takes 30 minutes
   - [GHS12b] 120 blocks of AES-128 (30K gates) in 36 hours
2. **2012~2015: Depth-Linear Construction**
   - [BGV12] Modulus/Key Switching
   - [Bra12] Scale Invariant Scheme
   - [HS14] IBM's open-source library Helib: AES evaluation in 4 minutes
3. **2015~Today: Usability**
   - Various schemes with different advantages (HEAAN, TFHE)
   - Real-world tasks: Big data analysis, Machine learning
   - Competitions for Private Genome Computation (iDash, 2014~)
   - HE Standardization meetings (2017~)

Continued on next page

Standardization: HomomorphicEncryption.org

Mar 2018 in MIT

Jul 2017 in Microsoft, Redmond

Oct 2018 in Toronto

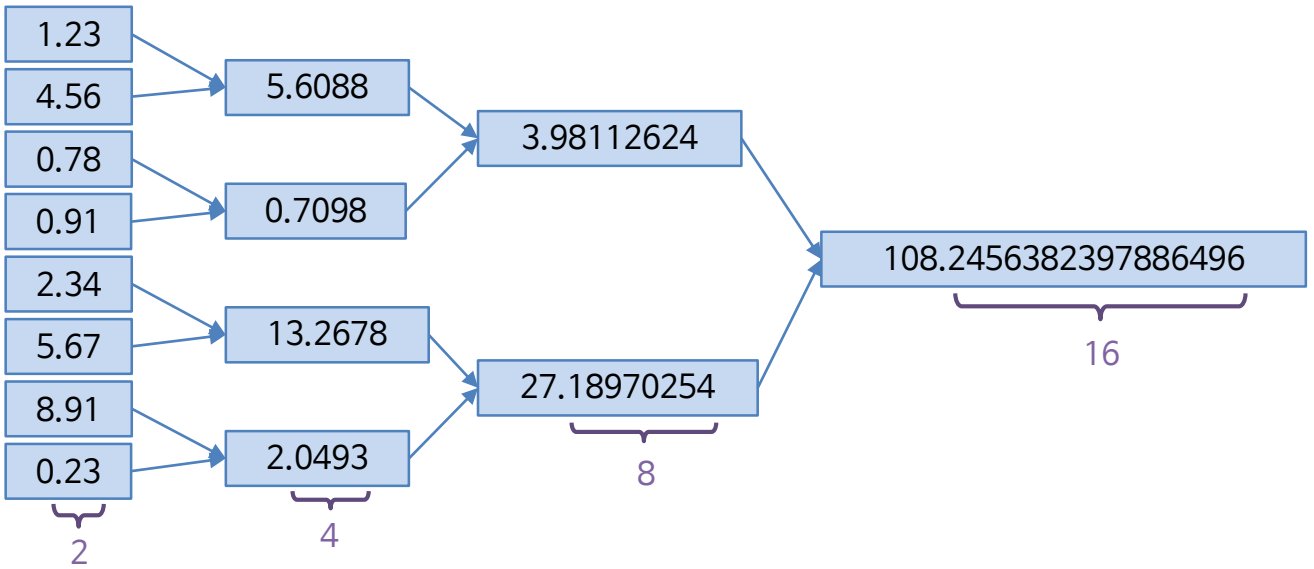
Homomorphic Encryption

❖ Best Performing HE Schemes

Type	Classical HE	Fast Bootstrapping	Approximate Computation
Scheme	[BGV12] BGV [Bra12, FV12] B/FV	[DM15] FHEW [CGGI16] TFHE	[CKKS17] HEAAN
Plaintext	Finite Field Packing	Binary string	Real/Complex numbers Packing
Operation	Addition, Multiplication	Look-up table & bootstrapping	Fixed-point Arithmetic
Library	HElib (IBM) SEAL (Microsoft Research) Palisade (Duality inc.)	TFHE (Inpher, Gemalto, etc.)	**HEAAN (SNU)**

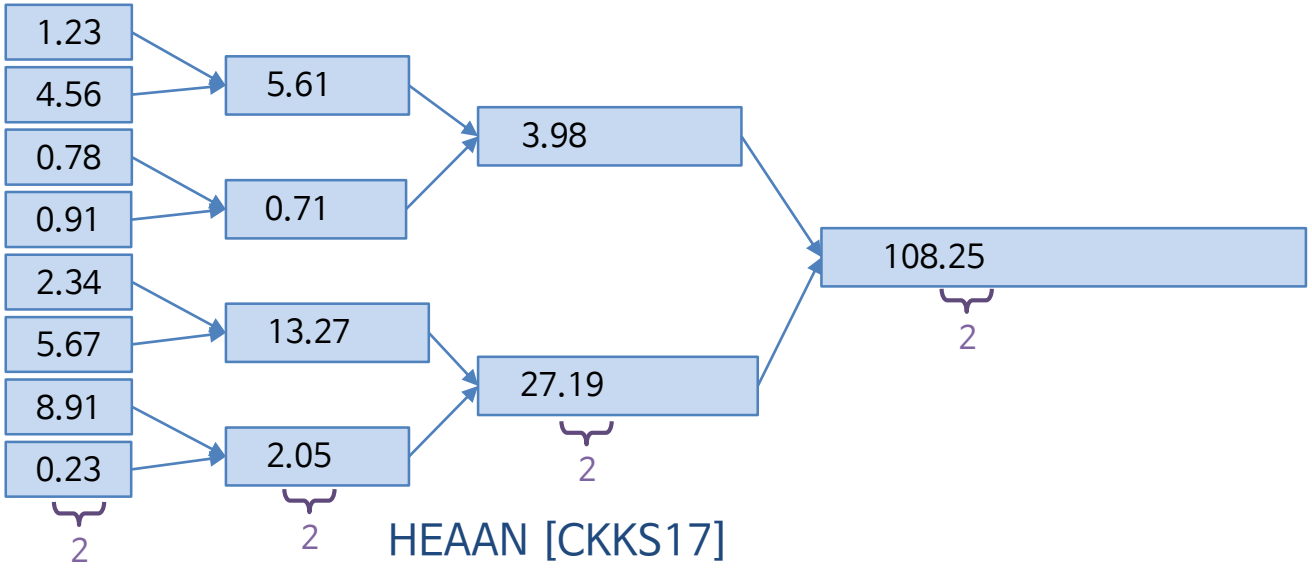
2. HEAAN: Approximate Homomorphic Encryption

Exact Multiplication



- The plaintext size is doubled after a multiplication.

Approximate Multiplication



HEAAN [CKKS17]

- Rescale after a multiplication
- Tracing # of significands
- Most data is processed approximately in Data analysis or ML

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

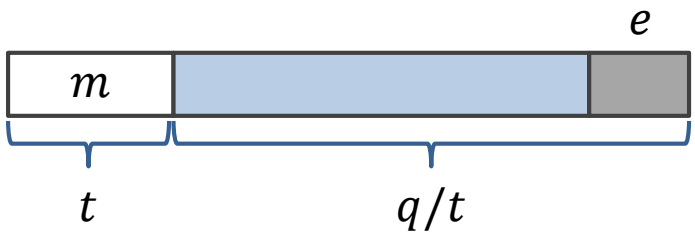
Approximate Computation

❖ 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)

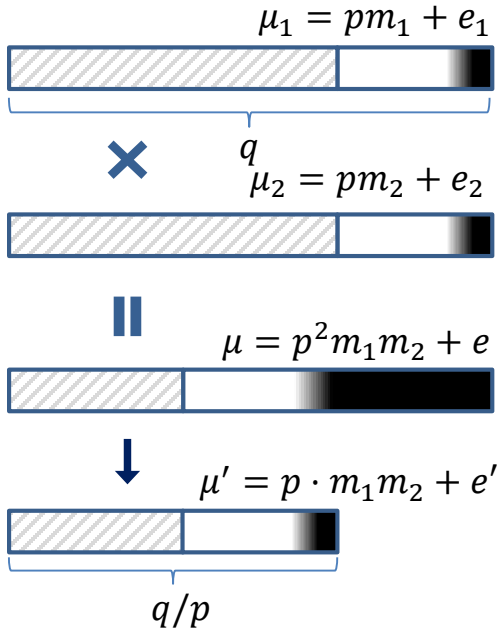
- $\mathrm{ct} = \mathrm{Enc}_{\mathrm{sk}}(m)$, $\langle \mathrm{ct}, \mathrm{sk} \rangle = \frac{q}{t} m + e \pmod{q}$
- Modulo t plaintext vs Rounding operation



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	$\mu_1 \approx pm_1, \mu_2 \approx pm_2$
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 : $q = p^L$

HEAAN Packed Ciphertext

- Construction over the ring
  - A single ctx can encrypt a vector of plaintext values $z = (z_1, z_2, \dots, z_\ell)$
  - Parallel computation in a SIMD manner $z \otimes w = (z_1 w_1, z_2 w_2, \dots, z_\ell w_\ell)$

Continued on next page

RLWE-based HEAAN

❖ Let $R = \mathbb{Z}[X]/(X^n + 1)$ and $R_q = (R \bmod q) = \mathbb{Z}_q[X]/(X^n + 1)$

- A ciphertext can encrypt a polynomial $m(X) \in R$
  - Note $(m_0+m_1X+\cdots)(m_0'+m_1'X+\cdots) = m_0m_0' + (m_0m_1' + m_0'm_1)X+\cdots$

- Decoding/Encoding function

$$R = \mathbb{Z}[X]/(X^n + 1) \subseteq \mathbb{R}[X]/(X^n + 1) \;\rightarrow\; \mathbb{C}^{n/2}$$

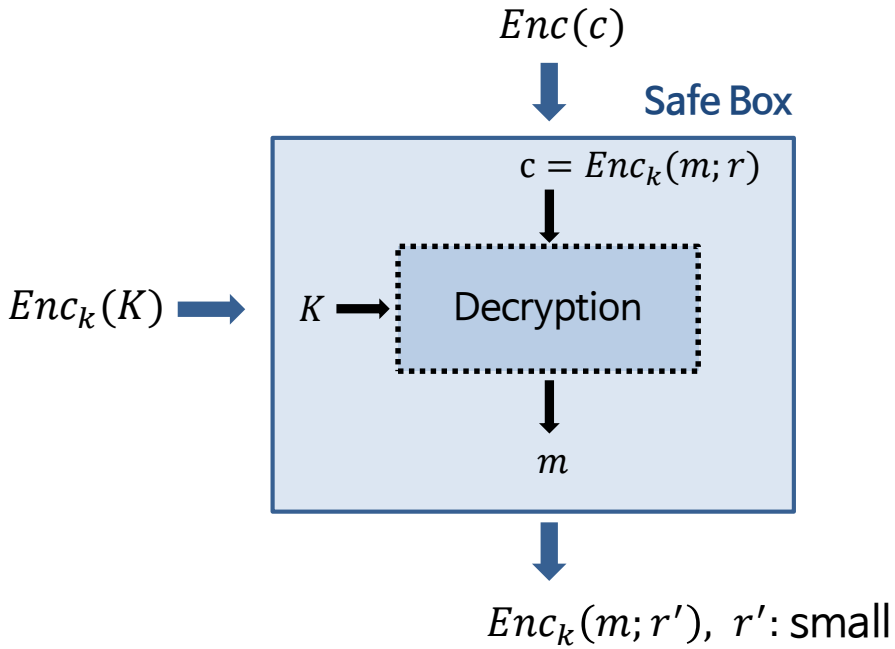
$$m(X) \;\mapsto\; z = (z_1, \ldots, z_{n/2}), \quad z_i = \mu(\zeta_i),$$

where $X^n + 1 = (X - \zeta_1)(X - \zeta_1^{-1})(X - \zeta_2)(X - \zeta_2^{-1}) \cdots (X - \zeta_{n/2})(X - \zeta_{n/2}^{-1})$

- Example: $n = 4, \zeta_1 = exp(\pi i/4), \zeta_2 = exp(5\pi i/4)$
  - $z = (1 - 2i, 3 + 4i) \mapsto m(X) = 2 - 2\sqrt{2}\,X + X^2 - \sqrt{2}\,X^3$
  - $\mapsto \mu(X) = 2000 - 2828X + 1000X^2 - 1414\,X^3$
  - $\mu(\zeta_1) \approx 1000.15 - 1999.55\,i, \;\; \mu(\zeta_2) \approx 2999.85 + 3999.55\,i$

3. Bootstrapping of HEAAN

Bootstrapping



Input

- Old Ciphertext with large noise
- Encrypted Secret Key

Process

- Evaluate Decrypt circuit

Output

- New ciphertext with small noise

[1] Cheon-Han-Kim-Kim-Song: Bootstrapping for Approximate Homomorphic Encryption. EUROCRYPT 2018

Bootstrapping

- Ciphertexts of a leveled HE have a limited lifespan
- Refresh a ciphertext $\mathrm{ct} = \mathrm{Enc}_{\mathrm{sk}}(m)$
  by **evaluating the decryption circuit homomorphically**
  : $\mathrm{Dec}_{\mathrm{sk}}(\mathrm{ct}) = m \iff F_{\mathrm{ct}}(\mathrm{sk}) = m$ where $F_{\mathrm{ct}}(*) = \mathrm{Dec}_{*}(\mathrm{ct})$
- Bootstrapping key $\mathrm{BK} = \mathrm{Enc}_{\mathrm{sk}}(\mathrm{sk})$
  : $F_{\mathrm{ct}}(\mathrm{BK}) = F_{\mathrm{ct}}\big(\mathrm{Enc}_{\mathrm{sk}}(\mathrm{sk})\big) = \mathrm{Enc}_{\mathrm{sk}}\big(F_{\mathrm{ct}}(\mathrm{sk})\big) = \mathrm{Enc}_{\mathrm{sk}}(m)$

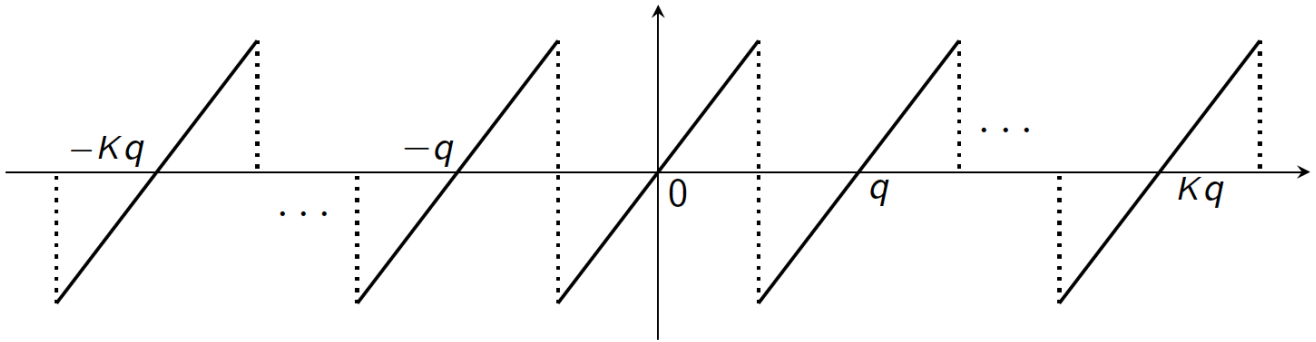
- Homomorphic operations introduce errors ➔ Fine
  : $F_{\mathrm{ct}}(\mathrm{BK}) = F_{\mathrm{ct}}\big(\mathrm{Enc}_{\mathrm{sk}}(\mathrm{sk})\big) = \mathrm{Enc}_{\mathrm{sk}}(F_{\mathrm{ct}}(\mathrm{sk}) + e) = \mathrm{Enc}_{\mathrm{sk}}(m + e)$

- How to evaluate the decryption circuit (efficiently)?
  : $\mathrm{Dec}_{\mathrm{sk}}(\mathrm{ct}) = \langle \mathrm{ct}, \mathrm{sk} \rangle \pmod{q}$

Approximate Decryption

$$Dec_{sk}(ct) \mapsto t = \langle ct, sk \rangle \mapsto [t]_q = \mu,$$
$$t = qI + \mu \text{ for some } |I| < K$$

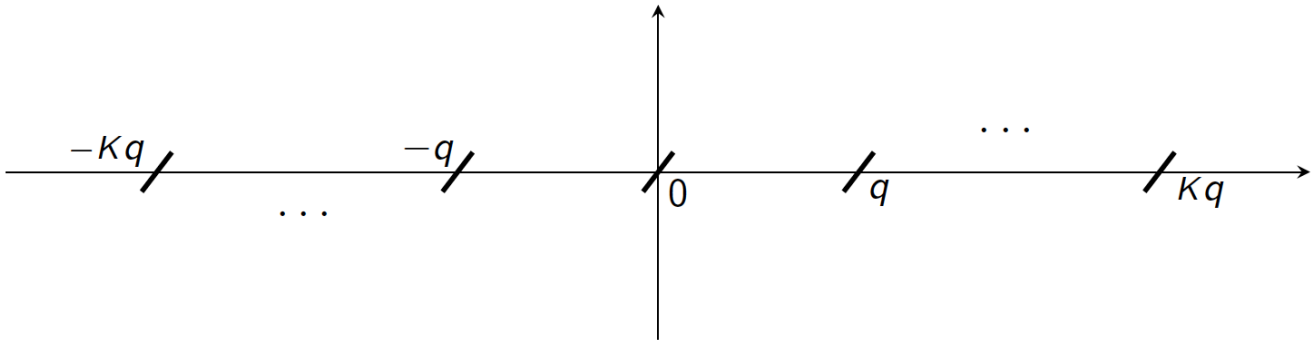
- Naïve solution: polynomial interpolation on $[-Kq, Kq]$
- Huge depth, complexity & inaccurate result



Approximate Decryption

$$Dec_{sk}(ct) \mapsto t = \langle ct, sk \rangle \mapsto [t]_q = \mu,$$
$$t = qI + \mu \text{ for some } |I| < K$$

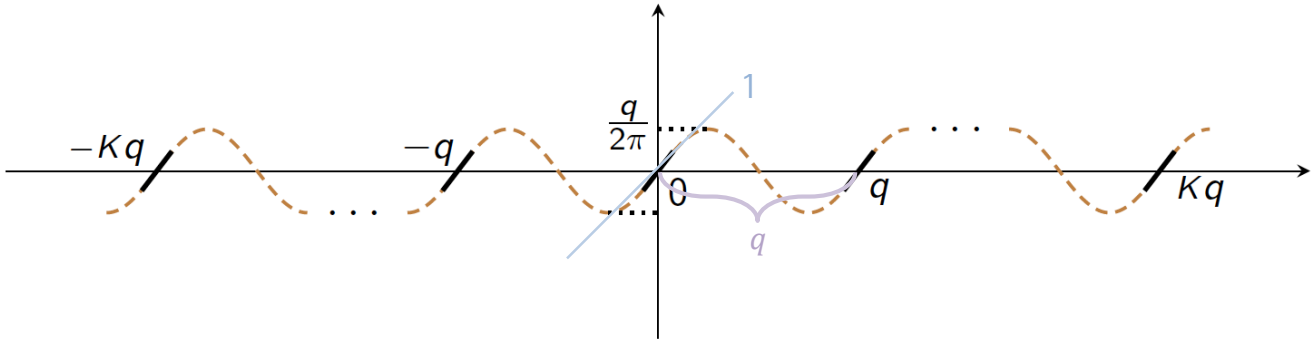
- **Idea1**: Restriction of domain $|\mu| \ll q$



Approximate Decryption

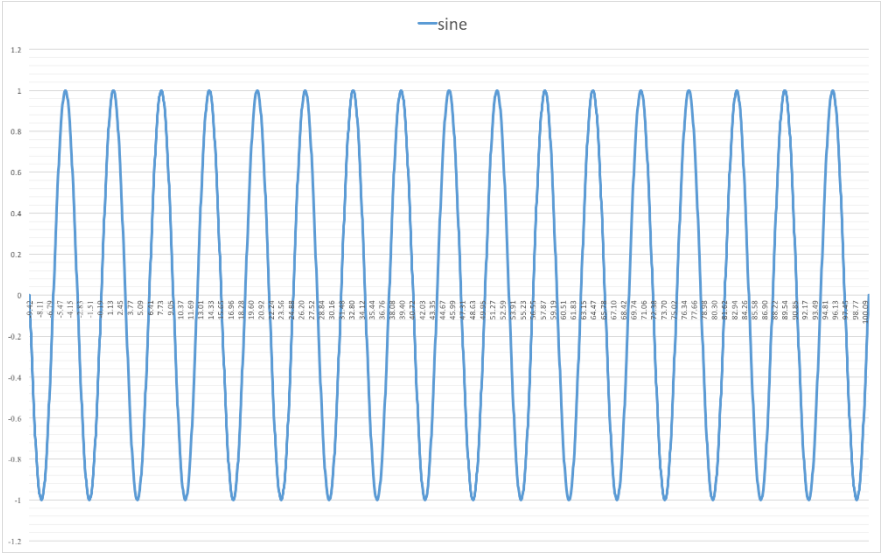
$$Dec_{sk}(ct) \mapsto t = \langle ct, sk \rangle \mapsto [t]_q = \mu,$$
$$t = qI + \mu \text{ for some } |I| < K$$

- **Idea1**: Restriction of domain $|\mu| \ll q$
- **Idea 2**: Sine approximation $\mu \approx \frac{q}{2\pi} \sin\theta$ for $\theta = \frac{2\pi}{q}\, t$ (period: q, slope at 0=1)



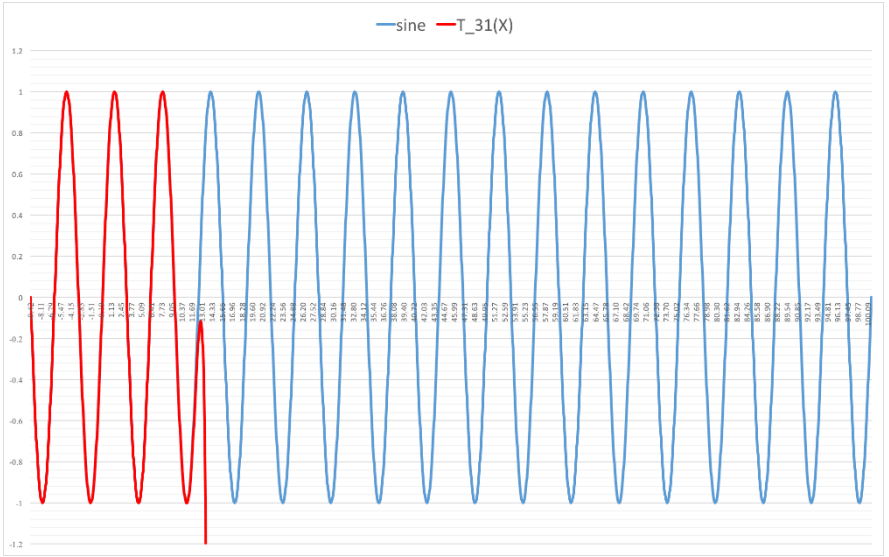
Bootstrapping of HEAAN

- Sine Evaluation

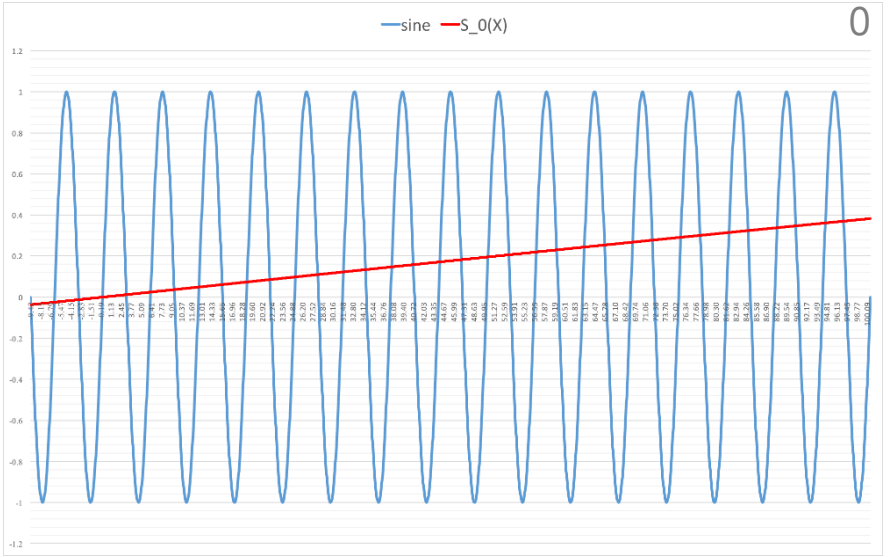


Bootstrapping of HEAAN

- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity

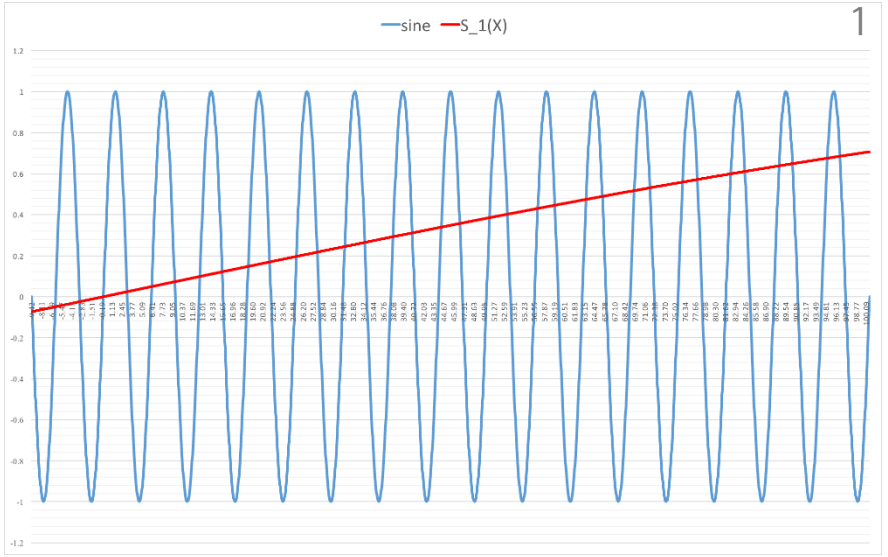


Bootstrapping of HEAAN



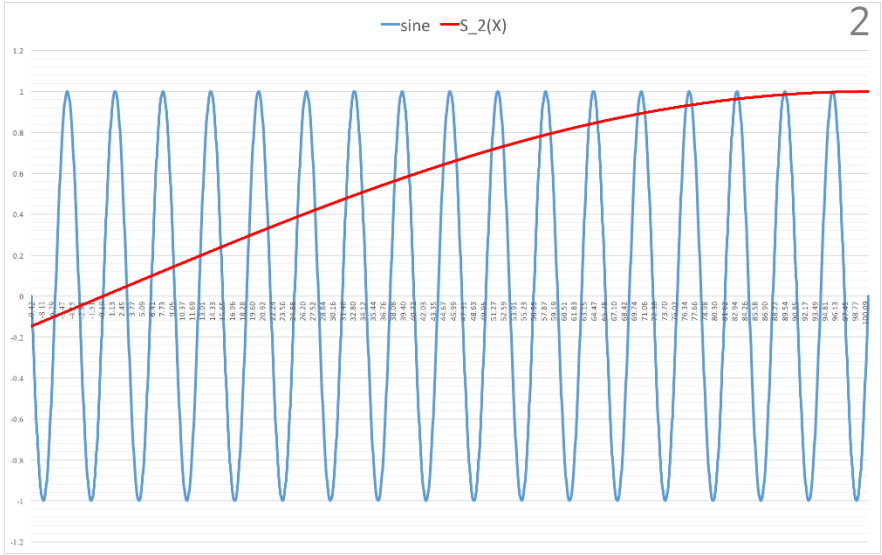
- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$

Bootstrapping of HEAAN



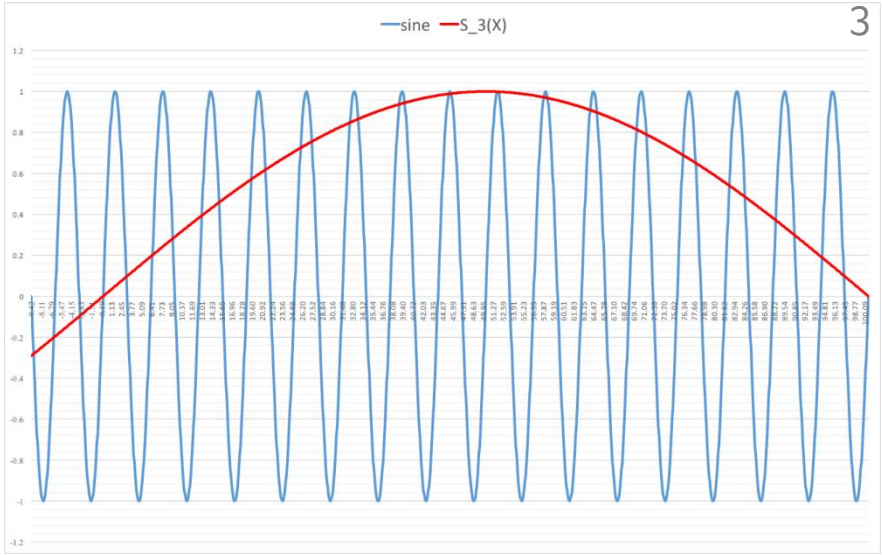
- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
  - **Idea 2:** Iterate by double-angle formula
    - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta), S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$

Bootstrapping of HEAAN



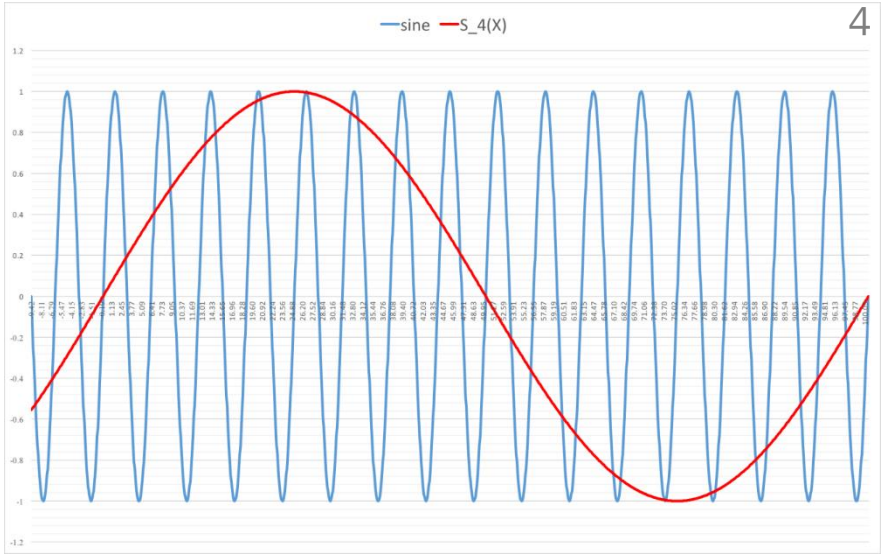
- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
  - **Idea 2:** Iterate by double-angle formula
    - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta)$, $S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$

Bootstrapping of HEAAN



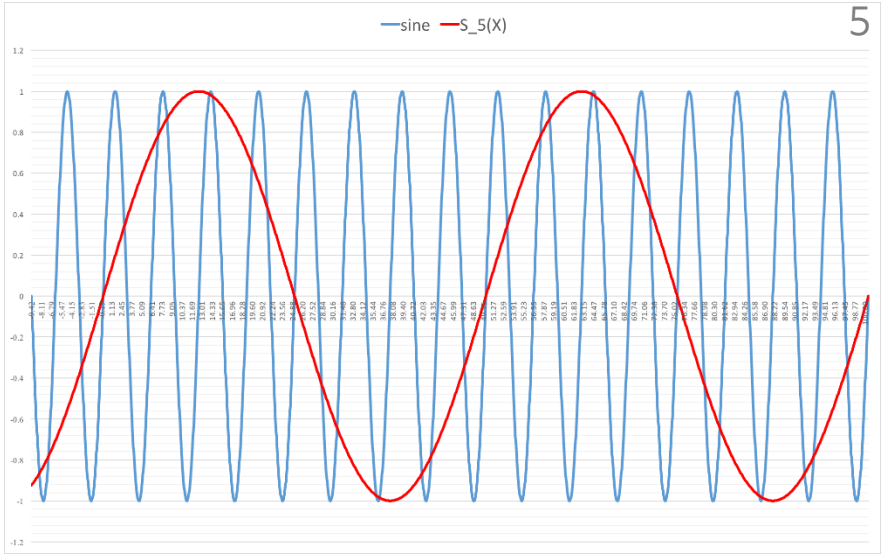
- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
  - **Idea 2:** Iterate by double-angle formula
    - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta), S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$

Bootstrapping of HEAAN



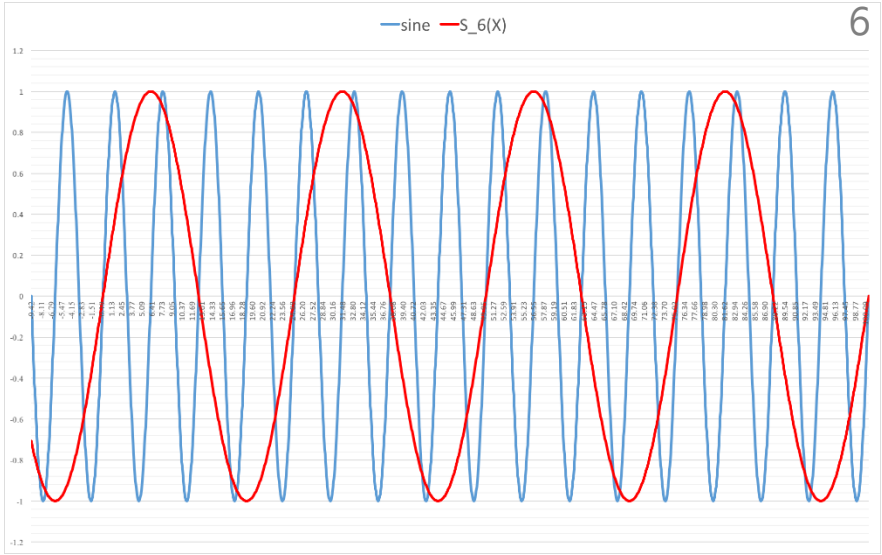
- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
  - **Idea 2:** Iterate by double-angle formula
    - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta), S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$

Bootstrapping of HEAAN



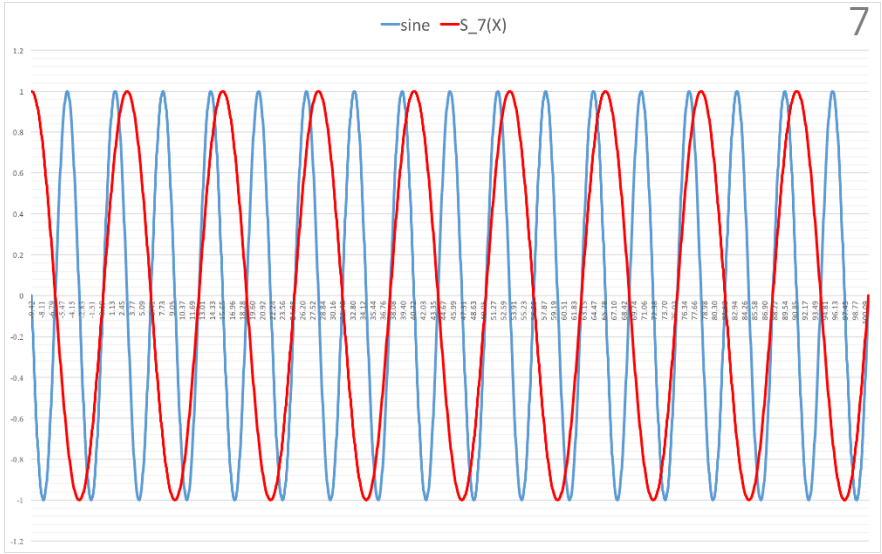
- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
  - **Idea 2:** Iterate by double-angle formula
    - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta), S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$

Bootstrapping of HEAAN



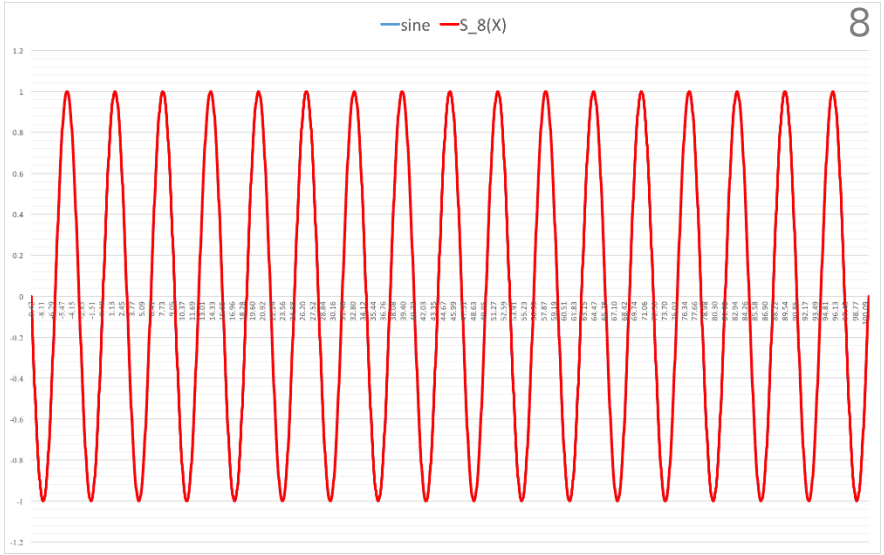
- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
  - **Idea 2:** Iterate by double-angle formula
    - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta), S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$

Bootstrapping of HEAAN



- Sine Evaluation
  - Direct Taylor approximation
    - Huge depth & complexity
  - **Idea 1**: Low-degree approx. near 0
    - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
    - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
  - **Idea 2:** Iterate by double-angle formula
    - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta), S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$

Bootstrapping of HEAAN



❖ Sine Evaluation

- Direct Taylor approximation
  - Huge depth & complexity
- **Idea 1**: Low-degree approx. near 0
  - $C_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k)!} (\theta/2^r)^{2k} \approx \cos(\theta/2^r)$
  - $S_0(\theta) = \sum_{k=0}^{d} \frac{(-1)^k}{(2k+1)!} (\theta/2^r)^{2k+1} \approx \sin(\theta/2^r)$
- **Idea 2:** Iterate by double-angle formula
  - $C_{k+1}(\theta) = C_k^2(\theta) - S_k^2(\theta), S_{k+1}(\theta) = 2S_k(\theta) \cdot C_k(\theta)$
  - **Numerically stable & Linear complexity**

$$S_r(\theta) \approx \sin\theta$$

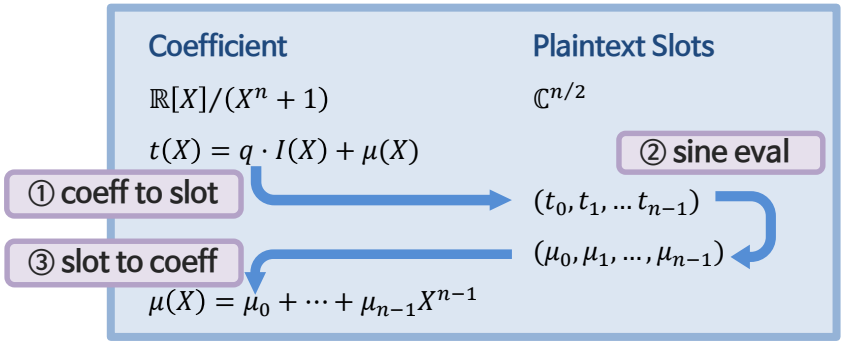
Bootstrapping of HEAAN

- Iteration vs Direct Computation
  - $S_r(\theta)$ is obtained from $S_0(\theta)$ and $C_0(\theta)$ by r iterations
    - One computation of Double – angle formula: 2 squarings + 1 addition
    - r iterations take $2r$ squarings + r additions
    - Degree of $S_r(\theta) \approx 2^r$
  - Direct Taylor Approximation
    - 2^r multiplications to get 2^r degree approximation T_{2^r} of sine function

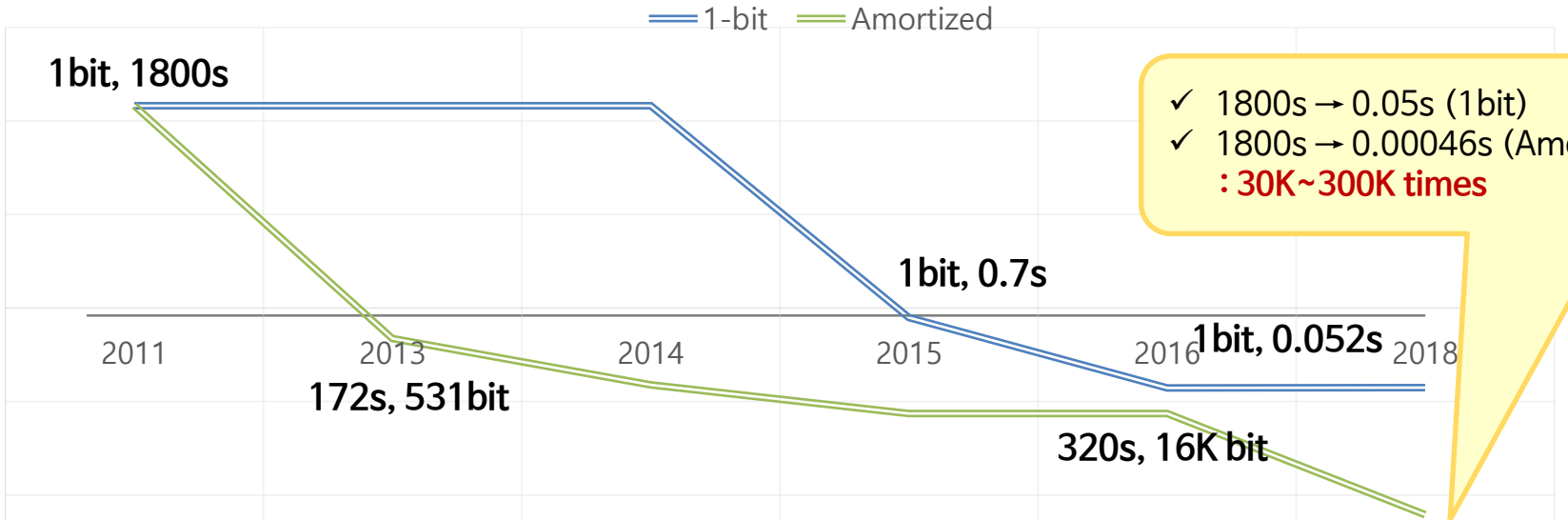
Bootstrapping of HEAAN

❖ Slot-Coefficient Switching

- Ring - based HEAAN
  - Homomorphic operations on plaintext slots, not on coefficients
  - We need to perform the modulo reduction on coefficients
- Pre/post computation before/after sine evaluation
  - Depth consumption: Sine evaluation
  - Complexity: Slot-Coefficient switchings (# of slots)
- Performance of Bootstrapping
- Experimental Results
  - 127 + 12 = 139 s / 128 slots × 12 bits
  - 456 + 68 = 524 s / 128 slots × 24 bits



Speed of FHE



[GH11] Implementing Gentry's Fully-Homomorphic Encryption Scheme, Eurocrypt 2011.
[CCK+13] Batch Fully Homomorphic Encryption over the Integers, Eurocrypt 2013.
[CLT14] Scale-Invariant Fully Homomorphic Encryption over the Integers, PKC 2014.
[HS15] Bootstrapping for Helib, Eurocrypt 2015
[DM15] FHEW: Boostrapping Homomorpic Encryption in Less Than a Second, Eurocrypt 2015.
[CGGI16] Faster Fully Homomorphic Encryption: Bootstrapping in less than 0.1 Seconds, Asiacrypt 2016.

* [CHH18] Faster Homomorphic Discrete Fourier Transforms and Improved FHE Bootstrapping, eprint, 1073, 2018/ Intel Xeon CPU E5-2620 2.10GHz, 64RAM

Secure Genome Analysis Competition

Hosted by

UC San Diego
School of Medicine

Sponsored by

NIH National Institutes of Health
Turning Discovery Into Health

AIM

Privacy Preserving Genom Analysis

2017 Track 3: Logistic Regression Training on Encrypted Data

TRACK 3: BEST-PERFORMING TEAMS

Evaluated on (three datasets of 1422 records for training/ 157 records for testing + 18 features)

Rank	Teams	AUC 0.7136	Encryption		Secure learning		Decryption		Overall time (mins)
			Size (MB)	Time (mins)	Time (mins)	Memory (MB)	Size (MB)	Time (mins)	
1	SNU	0.6934	537.667	**0.060**	**10.250**	2775.333	64.875	**0.050**	10.360
3	CEA LIST	0.6930	**53.000**	1.303	2206.057	**238.255**	**0.350**	**0.003**	2207.363
△	KU Leuven	0.6722	4904.000	4.304	155.695	7266.727	10.790	0.913	160.912
△	EPFL	0.6584	1011.750	1.633	15.089	1498.513	7.125	0.017	16.739
2	MSR	0.6574	1945.600	11.335	385.021	26299.344	76.000	0.033	396.390
X	Waseda*	0.7154	20.390	1.178	2.077	7635.600	20.390	2.077	5.332
X	Saarland	N/A	65536.000	1.633	48.356	29752.527	65536	7.355	57.344

* Interactive mechanism, no complete guarantee on 80-bit security at "analyst" side

** Program ends with errors

Slide Courtesy of Xiaoqian Jiang (△ = too small iteration → hard to adapt for other data)

2018 Track 2 : Secure Parallel Genome Wide Association Studies using HE

Team	Submission	Schemes	End to End Performance		Evaluation result (F1- Score) at different cutoffs							
			Running time (mins)	Peak Memory (M)	0.01		0.001		0.0001		0.00001	
					Gold	Semi	Gold	Semi	Gold	Semi	Gold	Semi
A*FHE	A*FHE -1 +	HEAAN	922.48	3,777	0.977	0.999	0.986	0.999	0.985	0.999	0.966	0.998
	A*FHE -2		1,632.97	4,093	0.882	0.905	0.863	0.877	0.827	0.843	0.792	0.826
Chimera	Version 1 +	TFHE & HEAAN (Chimera)	201.73	10,375	0.979	0.993	0.987	0.991	0.988	0.989	0.982	0.974
	Version 2		215.95	15,166	0.339	0.35	0.305	0.309	0.271	0.276	0.239	0.253
Delft Blue	Delft Blue	HEAAN	1,844.82	10,814	0.965	0.969	0.956	0.944	0.951	0.935	0.884	0.849
UC San Diego	Logistic Regr +	HEAAN	1.66	14,901	0.983	0.993	0.993	0.987	0.991	0.989	0.995	0.967
	Linear Regr		0.42	3,387	0.982	0.989	0.980	0.971	0.982	0.968	0.925	0.89
Duality Inc	Logistic Regr +	CKKS (Aka HEAAN), pkg: PALISADE	3.8	10,230	0.982	0.993	0.991	0.993	0.993	0.991	0.990	0.973
	Chi2 test		0.09	1,512	0.968	0.983	0.981	0.985	0.980	0.985	0.939	0.962
Seoul National University	SNU-1	HEAAN	52.49	15,204	0.975	0.984	0.976	0.973	0.975	0.969	0.932	0.905
	SNU-2		52.37	15,177	0.976	0.988	0.979	0.975	0.974	0.969	0.939	0.909
IBM	IBM-Complex	CKKS (Aka HEAAN), pkg: HElib	23.35	8,651	0.913	0.911	0.169	0.188	0.067	0.077	0.053	0.06
	IBM- Real		52.65	15,613	0.542	0.526	0.279	0.28	0.241	0.255	0.218	0.229

Slide Courtesy of Xiaoqian Jiang

4. Toolkit for Homomorphic Computation

How to Pack

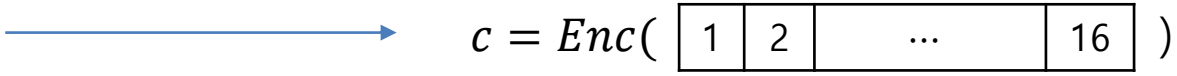
❖ Packing Method

- 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



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

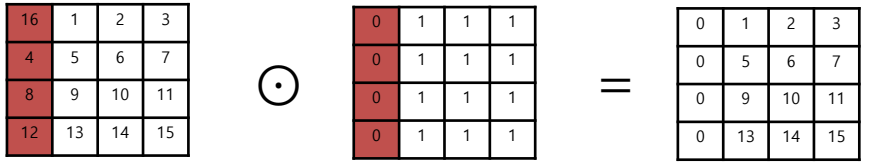
(wrong)

How to rotate

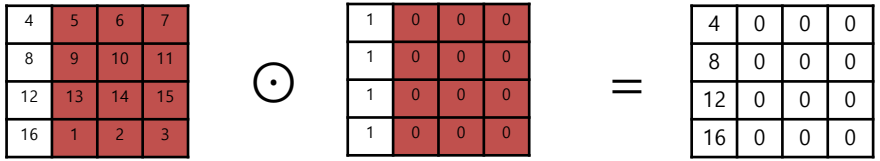
❖ Packing Method

- Solution : using masking vector

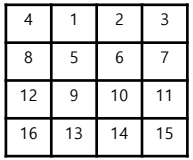
$$rot(c,1) \odot mask =$$



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



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



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 [HHCP]
- Need to evaluate sigmoid function : 1/(1+exp(-x))

- Real Financial Large Data (Provided by KCB)
- 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				

Comparison

- Idea: $Comp(a,b) = \lim_{k\to\infty} \frac{a^k}{a^k+b^k}$ for a,b>0

  1. Compute approximately (Take only the msb of the results)
  2. Choose k as a power of 2
  3. Use iteration 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**	**0.73 ms**	2^{16}	$< 2^{-8}$

Min/Max

- General Max

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

- 2nd $Max(a_1, \dots, at) = \lim_{k \to \infty} \frac{{a_1}^{k+1} + \cdots + {a_t}^{k+1} - max^{k+1}}{{a_1}^{k} + \cdots + {a_t}^{k} - max^{k}}$

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

Sorting on Encrypted Data

- Fast\Merge Sort: O(klogk)
  - 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 hour (previous work: impractical)

Toward Homomorphic Machine Learning

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

- Decision Tree: Packing, Comparison
- Boosting: Comparison and Gradient Decent

- Deep Neural Network: Fast matrix operations + Approximate
- Convolution Neural Network: + Comparison

HEAAN: Summary

- HEAAN natively supports for the (approximate) fixed point arithmetic
- Ciphertext modulus $\log q = L \log p$ grows linearly
- Useful when evaluating analytic functions approximately:
  - Polynomial
  - Multiplicative Inverse
  - Trigonometric Functions
  - Exponential Function (Logistic Function, Sigmoid Function)
- Packing technique based on DFT
  - SIMD (Single Instruction Multiple Data) operation
  - Rotation on plaintext slots:

$$z = (z_1, \dots, z_{n/2}) \mapsto z' = (z_2, \dots, z_{n/2}, z_1)$$

Thank you for your attention!

감사합니다.