Numerical Method for Comparison on Homomorphically Encrypted Numbers

Jung Hee Cheon, Dongwoo Kim, **Duhyeong Kim**, Hun Hee Lee, and Keewoo Lee

Seoul National University

ASIACRYPT 2019
Kobe, Japan

Homomorphic Encryption

Homomorphic Encryption (HE)

- **Homomorphic Encryption** is a cryptosystem which allows computations on ciphertexts.

Homomorphic Encryption (HE)

- **Homomorphic Encryption** is a cryptosystem which allows computations on ciphertexts.

$$
\begin{tikzcd}
\boldsymbol{m} \arrow[rr, "\boldsymbol{f}(\ )"] \arrow[d, shift right, "\boldsymbol{Enc}(\ )"'] \arrow[drr, phantom, "\circlearrowleft"] & & \boldsymbol{f}(\boldsymbol{m}) \arrow[d, shift right, "\boldsymbol{Enc}(\ )"'] \\
\boldsymbol{Enc}(\boldsymbol{m}) \arrow[rr, "\tilde{\boldsymbol{f}}(\ )"'] \arrow[u, shift right, "\boldsymbol{Dec}(\ )"'] & & \begin{array}{c}\boldsymbol{Enc}\big(\boldsymbol{f}(\boldsymbol{m})\big) \\ = \tilde{\boldsymbol{f}}\big(\boldsymbol{Enc}(\boldsymbol{m})\big)\end{array} \arrow[u, shift right, "\boldsymbol{Dec}(\ )"']
\end{tikzcd}
$$

Homomorphic Encryption (HE)

- Encryption methods: "bit-wise" encryption and "word-wise" encryption

Encryption	Plaintext space	Basic operation
Bit-wise	$\{0,1\}$	Logical gate, Look-up Table
Word-wise	$\mathbb{C}$ or $\mathbb{Z}_p$	Addition, Multiplication

Homomorphic Encryption (HE)

- Encryption methods: "bit-wise" encryption and "word-wise" encryption

Encryption	Plaintext space	Basic operation
Bit-wise	$\{0,1\}$	Logical gate, Look-up Table
Word-wise	$\mathbb{C}$ or $\mathbb{Z}_p$	Addition, Multiplication

- Bit-wise Encryption
  - Pros: High efficiency in gate operations and max/comparison operations
    - Comparison of ℓ-bit integers: $\Theta(\ell)$ complexity and $\log \ell$ depth
  - Cons: Polynomial evaluation (addition & multiplication) is very inefficient!

Homomorphic Encryption (HE)

- Encryption methods: "bit-wise" encryption and "word-wise" encryption

Encryption	Plaintext space	Basic operation
Bit-wise	$\{0,1\}$	Logical gate, Look-up Table
Word-wise	$\mathbb{C}$ or $\mathbb{Z}_p$	Addition, Multiplication

- Bit-wise Encryption
  - Pros: High efficiency in gate operations and max/comparison operations
    - Comparison of ℓ-bit integers: $\Theta(\ell)$ complexity and $\log \ell$ depth
  - Cons: Polynomial evaluation (addition & multiplication) is very inefficient!

"Word-wise" Encryption can be more suitable in many of real-world applications such as privacy-preserving machine learning

Homomorphic Encryption (HE)

- Word-wise Encryption
  - Pros: Much better performance for polynomial evaluations (e.g., addition almost for free)
  - Cons: Not easy to evaluate non-polynomial functions
    - Use polynomial approximation (e.g., Taylor, LSA, minimax, etc.)
    - Substitute $\max(a_1, \ldots, a_n)$ by the summation $a_1 + \cdots + a_n$ ($\Leftarrow$ application-dependent solution)

Homomorphic Encryption (HE)

- Word-wise Encryption
  - Pros: Much better performance for polynomial evaluations (e.g., addition almost for free)
  - Cons: Not easy to evaluate non-polynomial functions
    - Use polynomial approximation (e.g., Taylor, LSA, minimax, etc.)
    - Substitute $\max(a_1, \ldots, a_n)$ by the summation $a_1 + \cdots + a_n$ ($\Leftarrow$ application-dependent solution)

Q. Can we find an efficient polynomial approximation for Min/Max and Comparison?

Our Contribution

Our Contribution

Propose efficient approximate algorithms of Min/Max & Comparison operations for word-wise HEs with concrete error analysis.

Our Contribution

Propose efficient approximate algorithms of Min/Max & Comparison operations for word-wise HEs with concrete error analysis.

- **Efficiency**
  - Our Method $\geq$ General Polynomial Approximation (asymptotically)
  - Our Method achieves (quasi-)optimality in asymptotic complexity
  - Our Method $\approx$ Bit-wise Comparison in encrypted state (amortized running time)

Our Contribution

Propose efficient approximate algorithms of Min/Max & Comparison operations for word-wise HEs with concrete error analysis.

- **Efficiency**
  - Our Method $\geq$ General Polynomial Approximation (asymptotically)
  - Our Method achieves (quasi-)optimality in asymptotic complexity
  - Our Method $\approx$ Bit-wise Comparison in encrypted state (amortized running time)
- **Applications**
  - Top-k-max, and Threshold counting (+ Sorting, Clustering,…)

Key Idea

Use Iterative Algorithms !

Key Idea

Use Iterative Algorithms !

- General polynomial evaluation requires $\Theta(\sqrt{deg})$ multiplications [PS'73]
- However, in iterative algorithms, Complexity = Θ(Depth) = $\Theta(\log deg)$ = Θ(#iterations)
- If we use a structured polynomial which can be evaluated by iterative algorithms, then we may achieve much smaller complexity

Min / Max & Comparison

One-variable expression of Max & Comparison

- Max$(a, b) = \frac{a+b}{2} + \frac{|a-b|}{2}$ $\quad\Leftrightarrow\quad$ $|a| = Max(a, 0) = Max(0, a)$
- Comp$(a, b) = \chi_{[0,\infty)}(a - b)$ $\quad\Leftrightarrow\quad$ $\chi_{[0,\infty)}(a) =$ Comp$(a, 0) =$ Comp$(0, -a)$

One-variable expression of Max & Comparison

- $\mathrm{Max}(a,b) = \frac{a+b}{2} + \frac{|a-b|}{2}$ $\Leftrightarrow$ $|a| = Max(a,0) = Max(0,a)$
- $\mathrm{Comp}(a,b) = \chi_{[0,\infty)}(a-b)$ $\Leftrightarrow$ $\chi_{[0,\infty)}(a) = \mathrm{Comp}(a,0) = \mathrm{Comp}(0,-a)$

Max = Absolute function & Comp = Step function!

One-variable expression of Max & Comparison

- Max$(a,b) = \frac{a+b}{2} + \frac{|a-b|}{2}$ $\Leftrightarrow$ $|a| = Max(a,0) = Max(0,a)$
- Comp$(a,b) = \chi_{[0,\infty)}(a-b)$ $\Leftrightarrow$ $\chi_{[0,\infty)}(a) =$ Comp$(a,0) =$ Comp$(0,-a)$

Max = Absolute function & Comp = Step function!

Approximately compute absolute and step functions by iterative algorithms

Min / Max

- $\text{Max}(a, b) = \frac{a+b}{2} + \frac{|a-b|}{2} = \frac{a+b}{2} + \frac{\sqrt{(a-b)^2}}{2}$

Min / Max

- $\text{Max}(a,b) = \frac{a+b}{2} + \frac{|a-b|}{2} = \frac{a+b}{2} + \frac{\sqrt{(a-b)^2}}{2}$

Algorithm 2 Sqrt$(x; d)$

Input: $0 \le x \le 1$, $d \in \mathbb{N}$

Output: an approximate value of $\sqrt{x}$ (refer Lemma 2)

1: $a_0 \leftarrow x$
2: $b_0 \leftarrow x - 1$
3: **for** $n \leftarrow 0$ **to** $d-1$ **do**
4: $\quad a_{n+1} \leftarrow a_n \left(1 - \frac{b_n}{2}\right)$
5: $\quad b_{n+1} \leftarrow b_n^2 \left(\frac{b_n - 3}{4}\right)$
6: **end for**
7: **return** a_d

1. Given $a, b \in [0, 2^\ell)$
2. Scale down $(a, b) \leftarrow \left(\frac{a}{2^\ell}, \frac{b}{2^\ell}\right)$
3. Use Algorithm 2 for input $(a-b)^2 \in [0,1)$
4. Scale up the result by 2^ℓ

Min / Max

- $\mathrm{Max}(a,b) = \frac{a+b}{2} + \frac{|a-b|}{2} = \frac{a+b}{2} + \frac{\sqrt{(a-b)^2}}{2}$

Algorithm 2 $\mathtt{Sqrt}(x;d)$

Input: $0 \le x \le 1$, $d \in \mathbb{N}$
Output: an approximate value of $\sqrt{x}$ (refer Lemma 2)

1: $a_0 \leftarrow x$
2: $b_0 \leftarrow x - 1$
3: **for** $n \leftarrow 0$ **to** $d-1$ **do**
4: $\quad a_{n+1} \leftarrow a_n \left(1 - \frac{b_n}{2}\right)$
5: $\quad b_{n+1} \leftarrow b_n^2 \left(\frac{b_n - 3}{4}\right)$
6: **end for**
7: **return** a_d

Theorem 1 (Informal). If $\boxed{d = \theta(\alpha)}$ (resp. $d = \theta(\log \alpha)$), the error of $\mathrm{Max}(a,b;d)$ from the true value $\mathrm{Max}(a,b)$ is bounded by $2^{-\alpha}$ for any $a, b \in [0,1)$ (resp. unif. randomly chosen a, b with high prob.)

1. Given $a, b \in [0, 2^{\ell})$
2. Scale down $(a,b) \leftarrow \left(\frac{a}{2^{\ell}}, \frac{b}{2^{\ell}}\right)$
3. Use Algorithm 2 for input $(a-b)^2 \in [0,1)$
4. Scale up the result by 2^{ℓ}

Select the parameter d based on Theorem 1

Comparison

Motivation

Sigmoid approximation of the Step function $\chi_{(0,\infty)}$

Comparison

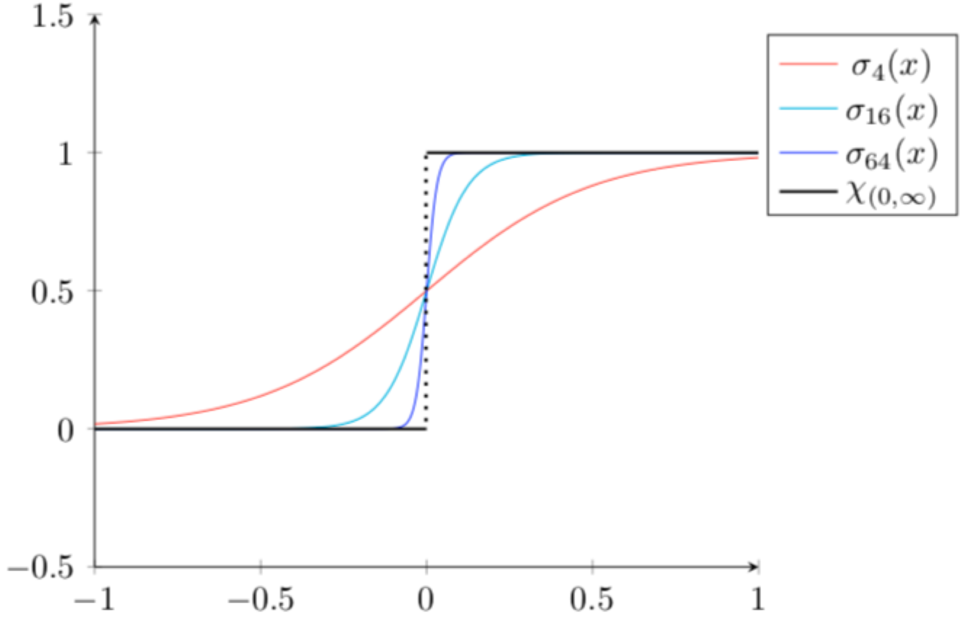

Fig. 1. Approximation of the step function $\chi_{(0,\infty)}$ by scaled sigmoid functions

Comparison

Motivation

Sigmoid approximation of the Step function $\chi_{(0,\infty)}$

$$Comp(a,b) = \chi_{(0,\infty)}(f(a) - f(b))$$

$$\approx \sigma_k(f(a) - f(b)) := \frac{1}{1+e^{-k(f(a)-f(b))}} = \frac{e^{kf(a)}}{e^{kf(a)}+e^{kf(b)}} \text{ for any strictly increasing function } f$$

Comparison

Motivation

Sigmoid approximation of the Step function $\chi_{(0,\infty)}$

$$
\begin{aligned}
Comp(a,b) &= \chi_{(0,\infty)}(f(a)-f(b)) \\
&\approx \sigma_k(f(a)-f(b)) \coloneqq \frac{1}{1+e^{-k(f(a)-f(b))}} = \frac{e^{kf(a)}}{e^{kf(a)}+e^{kf(b)}} \text{ for any strictly increasing function } f
\end{aligned}
$$

Take $\boldsymbol{f(x) = \log x}$, then exponential function $\boldsymbol{e^{kf(a)}} \Longrightarrow$ polynomial $\boldsymbol{a^k}$!

Comparison

Motivation

Sigmoid approximation of the Step function $\chi_{(0,\infty)}$

$$Comp(a,b) = \chi_{(0,\infty)}(f(a) - f(b))$$

$$\approx \sigma_k(f(a) - f(b)) \coloneqq \frac{1}{1+e^{-k(f(a)-f(b))}} = \frac{e^{kf(a)}}{e^{kf(a)}+e^{kf(b)}}$$ for any strictly increasing function f

Take $\boldsymbol{f(x) = \log x}$, then exponential function $\boldsymbol{e^{kf(a)}} \Longrightarrow$ polynomial $\boldsymbol{a^k}$!

$$\lim_{k\to\infty} \frac{a^k}{a^k + b^k} = Comp(a,b)$$

Comparison

Motivation

Sigmoid approximation of the Step function $\chi_{(0,\infty)}$

$$Comp(a,b) = \chi_{(0,\infty)}(f(a) - f(b))$$

$$\approx \sigma_k(f(a) - f(b)) := \frac{1}{1+e^{-k(f(a)-f(b))}} = \frac{e^{kf(a)}}{e^{kf(a)}+e^{kf(b)}}$$ for any strictly increasing function f

Take $\boldsymbol{f(x) = \log x}$, then exponential function $\boldsymbol{e^{kf(a)}} \Longrightarrow$ polynomial $\boldsymbol{a^k}$!

$$\lim_{k\to\infty} \frac{a^k}{a^k + b^k} = Comp(a,b)$$

We need an iterative algorithm for inversion!

Comparison

Goldschmidt's Inverse Algorithm

Algorithm 1 $\texttt{Inv}(x; d)$

Input: $0 < x < 2$, $d \in \mathbb{N}$
Output: an approximate value of $1/x$ (refer Lemma 1)

1: $a_0 \leftarrow 2 - x$
2: $b_0 \leftarrow 1 - x$
3: **for** $n \leftarrow 0$ **to** $d - 1$ **do**
4: $\quad b_{n+1} \leftarrow b_n^2$
5: $\quad a_{n+1} \leftarrow a_n \cdot (1 + b_{n+1})$
6: **end for**
7: **return** a_d

Convergence Rate:

$$\left| a_d - \frac{1}{x} \right| \le (1 - x)^{2^{d+1}}$$

For $\frac{1}{2} \le x \le \frac{3}{2}$,

$$\left| a_d - \frac{1}{x} \right| \le 2^{-2^{d+1}}$$

$$\frac{\mathbf{1}}{\boldsymbol{x}} = \frac{\mathbf{1}}{\mathbf{1} - (\mathbf{1} - \boldsymbol{x})} \approx \left(\mathbf{1} + (\mathbf{1} - \boldsymbol{x})\right)\left(\mathbf{1} + (\mathbf{1} - \boldsymbol{x})^{\mathbf{2}}\right)\left(\mathbf{1} + (\mathbf{1} - \boldsymbol{x})^{\mathbf{4}}\right) \cdots \left(\mathbf{1} + (\mathbf{1} - \boldsymbol{x})^{\mathbf{2}^{\boldsymbol{d}}}\right)$$

Comparison

- Main Obstacle of computing $\frac{a^k}{a^k + b^k}$?

Comparison

- Main Obstacle of computing $\frac{a^k}{a^k+b^k}$?

$\Longrightarrow$ Normalization: Inverse operation of $(a^k + b^k)$

Comparison

- Main Obstacle of computing $\frac{a^k}{a^k + b^k}$?

$\Longrightarrow$ Normalization: Inverse operation of $(a^k + b^k)$

$\Longrightarrow$ Hard to control the size of $(a^k + b^k)$ if k is too large

Comparison

- Main Obstacle of computing $\frac{a^k}{a^k + b^k}$?

$\Longrightarrow$ Normalization: Inverse operation of $(a^k + b^k)$

$\Longrightarrow$ Hard to control the size of $(a^k + b^k)$ if k is too large

$\Longrightarrow$ Iterative Normalization with much smaller k

Iterative Normalization

1. Scale down $a, b \in (0, 2^{\ell})$ into $\left(\frac{1}{2}, \frac{3}{2}\right)$ via mapping $x \mapsto \frac{x+2^{\ell-1}}{2^{\ell}}$

2. $(a, b) \leftarrow \left(\frac{a}{2} \cdot Inv\left(\frac{a+b}{2}; d\right), \frac{b}{2} \cdot Inv\left(\frac{a+b}{2}; d\right)\right)$ (Initial Normalization)

Repeat the following for t times

3. $(a, b) \leftarrow \left(a^2 \cdot Inv(a^2 + b^2; d), b^2 \cdot Inv(a^2 + b^2; d)\right)$ (Iterative Normalization)

Iterative Normalization

1. Scale down $a, b \in (0, 2^\ell)$ into $\left(\frac{1}{2}, \frac{3}{2}\right)$ via mapping $x \mapsto \frac{x+2^{\ell-1}}{2^\ell}$

2. $(a, b) \leftarrow \left(\frac{a}{2} \cdot Inv\left(\frac{a+b}{2}; d\right), \frac{b}{2} \cdot Inv\left(\frac{a+b}{2}; d\right)\right)$ (Initial Normalization)

Repeat the following for t times

3. $(a, b) \leftarrow \left(a^2 \cdot Inv(a^2 + b^2; d), b^2 \cdot Inv(a^2 + b^2; d)\right)$ (Iterative Normalization)

The output $\approx \left(\frac{a^{2^t}}{a^{2^t} + b^{2^t}}, \frac{b^{2^t}}{a^{2^t} + b^{2^t}}\right) \approx (a = \max(a, b)\,?\,, b = \max(a, b)\,?\,)$

Iterative Normalization

1. Scale down $a, b \in (0,2^\ell)$ into $\left(\frac{1}{2}, \frac{3}{2}\right)$ via mapping $x \mapsto \frac{x+2^{\ell-1}}{2^\ell}$

2. $(a, b) \leftarrow \left(\frac{a}{2} \cdot Inv\left(\frac{a+b}{2}; d\right), \frac{b}{2} \cdot Inv\left(\frac{a+b}{2}; d\right)\right)$ (Initial Normalization)

Repeat the following fo

3. $(a, b) \leftarrow (a^2 \cdot Inv$

Theorem 2 (Informal). For $t = \Theta(\alpha)$ and $d = \Theta(\log \alpha)$, the error rate of $Comp(a, b; d, t)$ compared to the true $comp(a, b)$ is bounded by $2^{-\alpha}$ for $a, b \in \left[\frac{1}{2}, \frac{3}{2}\right]$ satisfying $\frac{\max(a,b)}{\min(a,b)} \geq 1 + 2^{-\alpha}$.

The output $\approx \left(\frac{a^{2^t}}{a^{2^t}+b^{2^t}}, \frac{b^{2^t}}{a^{2^t}+b^{2^t}}\right) \approx (a = \max(a, b)\,?\,, b = \max(a, b)\,?\,)$

Asymptotic Optimality of our Method

Approximation Theorems for Min/Max and Comp

Theoretical results on the polynomial approximation of $|x|$ and $\chi_{(0,\infty)}$:

Theorem 3 [Ber'14]

$$\lim_{k\to\infty} k \cdot \| |x| - p_k \|_{\infty,[-1,1]} = \beta \text{ for some constant } \beta \approx 0.28$$

Theorem 4 [EY'07]

$$\lim_{k\to\infty} \sqrt{\frac{k-1}{2}} \cdot \left(\frac{1+\epsilon}{1-\epsilon}\right)^{\frac{k-1}{2}} \cdot \left\| \chi_{(0,\infty)} - q_{k,\epsilon} \right\|_{\infty,[-1,-\epsilon]\cup[\epsilon,1]} = \frac{1-\epsilon}{2\sqrt{\pi\epsilon}}$$

p_k: degree-k minimax approx. poly. of $|x|$ over the interval $[-1,1]$

$q_{k,\epsilon}$: degree-k minimax approx. poly. of $\chi_{(0,\infty)}$ over the interval $[-1,-\epsilon] \cup [\epsilon, 1]$

Approximation Theorems for Min/Max and Comp

Theoretical results on the polynomial approximation of $|x|$ and $\chi_{(0,\infty)}$:

Theorem 3 [Ber'14]

$$\lim_{k\to\infty} k \cdot \| |x| - p_k \|_{\infty,[-1,1]} = \beta \text{ for some constant } \beta \approx 0.28$$

Theorem 4 [EY'07]

$$\lim_{k\to\infty} \sqrt{\frac{k-1}{2}} \cdot \left(\frac{1+\epsilon}{1-\epsilon}\right)^{\frac{k-1}{2}} \cdot \left\| \chi_{(0,\infty)} - q_{k,\epsilon} \right\|_{\infty,[-1,-\epsilon]\cup[\epsilon,1]} = \frac{1-\epsilon}{2\sqrt{\pi\epsilon}}$$

p_k: degree-k minimax approx. poly. of $|x|$ over the interval $[-1,1]$

$q_{k,\epsilon}$: degree-k minimax approx. poly. of $\chi_{(0,\infty)}$ over the interval $[-1,-\epsilon] \cup [\epsilon,1]$

To obtain $O(2^{-\alpha})$ error:

$\boxed{k \geq \Theta(2^{\alpha})}$

$\boxed{k \geq \Theta(\alpha)}$ **for const ϵ**

$\boxed{k \geq \Theta(\alpha \cdot 2^{\alpha})}$ **for $\epsilon = 2^{-\alpha}$**

Our Method vs. Minimax Approx.

Overall, to achieve $O(2^{-\alpha})$ error :

		Minimax Approx.	**Our Method**
min/max		$\Theta(2^{\alpha/2})$	$\boldsymbol{\Theta(\alpha)}$
comparison	$\epsilon = \omega(1)$	$\Theta(\sqrt{\alpha})$	$\boldsymbol{\Theta\left(\log^2 \alpha\right)}$
	$\epsilon = 2^{-\alpha}$	$\Theta\left(\sqrt{\alpha} \cdot 2^{\alpha/2}\right)$	$\boldsymbol{\Theta\left(\alpha \log \alpha\right)}$

Table 1. Complexity of our methods and minimax approximation method

Our Method vs. Minimax Approx.

Overall, to achieve $O(2^{-\alpha})$ error :

		Minimax Approx.	**Our Method**
min/max		$\Theta(2^{\alpha/2})$	$\boldsymbol{\Theta(\alpha)}$
comparison	$\epsilon = \omega(1)$	$\Theta(\sqrt{\alpha})$	$\boldsymbol{\Theta\left(\log^2 \alpha\right)}$
	$\epsilon = 2^{-\alpha}$	$\Theta\left(\sqrt{\alpha} \cdot 2^{\alpha/2}\right)$	$\boldsymbol{\Theta\left(\alpha \log \alpha\right)}$

Table 1. Complexity of our methods and minimax approximation method

Complexity $\geq$ Θ(Depth) = Θ(log Degree)

Approximation Theorems for Min/Max and Comp

Theoretical results on the polynomial approximation of $|x|$ and $\chi_{(0,\infty)}$:

Theorem 3 [Ber'14]

$$\lim_{k\to\infty} k \cdot \| |x| - p_k \|_{\infty,[-1,1]} = \beta \text{ for some constant } \beta \approx 0.28$$

Theorem 4 [EY'07]

$$\lim_{k\to\infty} \sqrt{\frac{k-1}{2}} \cdot \left(\frac{1+\epsilon}{1-\epsilon}\right)^{\frac{k-1}{2}} \cdot \left\| \chi_{(0,\infty)} - q_{k,\epsilon} \right\|_{\infty,[-1,-\epsilon]\cup[\epsilon,1]} = \frac{1-\epsilon}{2\sqrt{\pi\epsilon}}$$

p_k: degree-k minimax approx. poly. of $|x|$ over the interval $[-1,1]$

$q_{k,\epsilon}$: degree-k minimax approx. poly. of $\chi_{(0,\infty)}$ over the interval $[-1,-\epsilon] \cup [\epsilon,1]$

To obtain $O(2^{-\alpha})$ error:

$\boldsymbol{k \geq \Theta(2^{\alpha})}$

$\boldsymbol{k \geq \Theta(\alpha)}$ **for const ϵ**

$\boldsymbol{k \geq \Theta(\alpha \cdot 2^{\alpha})}$ **for $\epsilon = 2^{-\alpha}$**

Our Method vs. Minimax Approx.

Overall, to achieve $O(2^{-\alpha})$ error :

		Minimax Approx.	**Our Method**
min/max		$\Theta(2^{\alpha/2})$	$\boldsymbol{\Theta(\alpha)}$
comparison	$\epsilon = \omega(1)$	$\Theta(\sqrt{\alpha})$	$\boldsymbol{\Theta\left(\log^2 \alpha\right)}$
	$\epsilon = 2^{-\alpha}$	$\Theta\left(\sqrt{\alpha} \cdot 2^{\alpha/2}\right)$	$\boldsymbol{\Theta\left(\alpha \log \alpha\right)}$

Optimal

Quasi-Optimal

Table 1. Complexity of our methods and minimax approximation method

Complexity $\geq$ Θ(Depth) = Θ(log Degree)

Implementation Results

Max

- Input: two 8-bit integers a, b (or fractional numbers in [0, 1] w/ difference > 2^{-8})
- Output: max(a, b) with 8-bit precision
- Performance

Work	Method	Scheme	Amortized Running time	# of pairs	Error
[CGH+18]	Bit-wise	HElib	≈ 1 ms	1800	0
[CGGI17]		TFHE	≈ 1 ms	-	
Ours	**Word-wise**	**HEAAN**	**0.73 ms**	2^{16}	$< 2^{-8}$

- Ours: on Intel Xeon CPU E5-2620 v4 at 2.10GHz processor with multi-threading (8 threads) turned on.
- HEAAN parameter (with 128-bit security): $\log N = 17,\ \log Q = 930, \lambda = 192.2$
- Theoretical (# iterations): 13
- Practical (# iterations): 11

Comparison

- Input: two 8-bit integers a, b (or fractional numbers in [0, 1] w/ difference > 2^{-8})
- Output: comp(a, b) with 8-bit precision
- Performance

Work	Method	Amortized Running time	# of pairs	Error
Ours	**Word-wise**	**4.72 ms**	2^{16}	$< 2^{-7}$

- on Intel Xeon CPU E5-2620 v4 at 2.10GHz processor with multi-threading (8 threads) turned on.
- HEAAN parameters
  - ✓ Comp: $\log N = 17,\ \log Q = 1870,\ \log p = 30,\ \lambda = 108.9$
- # Iterations: $(d, t) = (5{,}6)$

Follow-up Study

- Can we design a new framework for the polynomial approximation of min/max and comparison operations?

Follow-up Study

- Can we design a new framework for the polynomial approximation of min/max and comparison operations?
- Can we achieve the optimal complexity for the comparison algorithm?

Follow-up Study

- Can we design a new framework for the polynomial approximation of min/max and comparison operations?
- Can we achieve the optimal complexity for the comparison algorithm?
- Can we make a trade-off between depth and complexity?

Follow-up Study

- Can we design a new framework for the polynomial approximation of min/max and comparison operations? ✓
- Can we achieve the optimal complexity for the comparison algorithm? ✓
- Can we make a trade-off between depth and complexity? ✓

YES!

If interested, welcome to **ia.cr/2019/1234**

thank you!