Privacy-Preserving Approximation of Transcendental Functions

Aldrich Ellis C. Asuncion Brian Christopher T. Guadalupe

19th Philippine Computing Science Congress
March 28–30, 2019

Ateneo de Manila University

Big questions

- Cryptosystems allow us to **store and transmit** data securely.
  - To send a message securely, encrypt it into a secure form, which can then be decrypted by an authorized party.
- Can **computation** be performed securely?
  - If so, what kinds of computation are possible?

Secure computation

Given encrypted messages, can we perform operations on them *without revealing the underlying messages*?

Secure computation

Given encrypted messages, can we perform operations on them *without revealing the underlying messages*?

- Yes, using **fully homomorphic encryption (FHE)**.
  - First FHE scheme, using lattice-based cryptography, presented by Gentry in 2009.
  - FHE allows for arbitrary computation on encrypted data, but is very slow.

Secure computation

Given encrypted messages, can we perform operations on them *without revealing the underlying messages*?

- Yes, using **fully homomorphic encryption (FHE)**.
  - First FHE scheme, using lattice-based cryptography, presented by Gentry in 2009.
  - FHE allows for arbitrary computation on encrypted data, but is very slow.

Can we perform secure computation *efficiently*?

Secure computation

Given encrypted messages, can we perform operations on them *without revealing the underlying messages*?

- Yes, using **fully homomorphic encryption (FHE)**.
  - First FHE scheme, using lattice-based cryptography, presented by Gentry in 2009.
  - FHE allows for arbitrary computation on encrypted data, but is very slow.

Can we perform secure computation *efficiently*?

- To some extent, using **partially homomorphic encryption (PHE)**.
  - PHE only allows *some* operations to be performed securely.
  - PHE schemes are significantly less time-intensive than current FHE schemes.

The Paillier cryptosystem

The **Paillier cryptosystem** was published by Pascal Paillier in 1999, in the paper *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*.

Limitations of the Paillier cryptosystem:

1. Only allows for the encryption and manipulation of **integers**
2. A **partially homomorphic cryptosystem**, i.e. supports limited secure computation
   - A. Given two encrypted integers, can obtain encryption of their sum.
   - B. Given an encrypted integer and an unencrypted integer, can obtain encryption of their sum.
   - C. Given an encrypted integer and an unencrypted integer, can obtain encryption of their product.

Extensions to the Paillier cryptosystem

- There is a known method to extend Paillier to **floating-point numbers**, allowing secure addition/subtraction and secure plaintext multiplication.
- In 2017, Boukoros, Karvelas, and Katzenbeisser presented a protocol to perform secure **division** in a two-party system.
- In 2009, Erkin, et al. presented a protocol to perform secure **squaring** in a two-party system.
- We can use the squaring protocol to perform secure **multiplication** in a two-party system.
  - Suppose we have the encryptions of a and b.
  - We can compute the encryptions of $(a+b)^2$, a^2, and b^2.
  - Then, compute the encryption of $\frac{1}{2}((a+b)^2 - a^2 - b^2) = ab$.

What we know so far

Using the Paillier cryptosystem and known extensions, we are able to

- Add and subtract encrypted numbers;
- Multiply and divide encrypted numbers;
- Raise an encrypted number to an integer power.

But can we do more?

What we know so far

Using the Paillier cryptosystem and known extensions, we are able to

- ▷ Add and subtract encrypted numbers;
- ▷ Multiply and divide encrypted numbers;
- ▷ Raise an encrypted number to an integer power.

But can we do more? **Yes**.

- ▷ Compute logarithms of encrypted numbers
- ▷ Compute the inverse tangent of encrypted numbers

Conventional methods for logarithm approximation

Remarks on using conventional methods to approximate the logarithm of an encrypted number:

- Series approximation — the Maclaurin series representation of $\log(1+x)$ only converges for $-1 \leq x \leq 1$.
- Lookup tables — cannot be done securely as we cannot expose the value of the encrypted number.

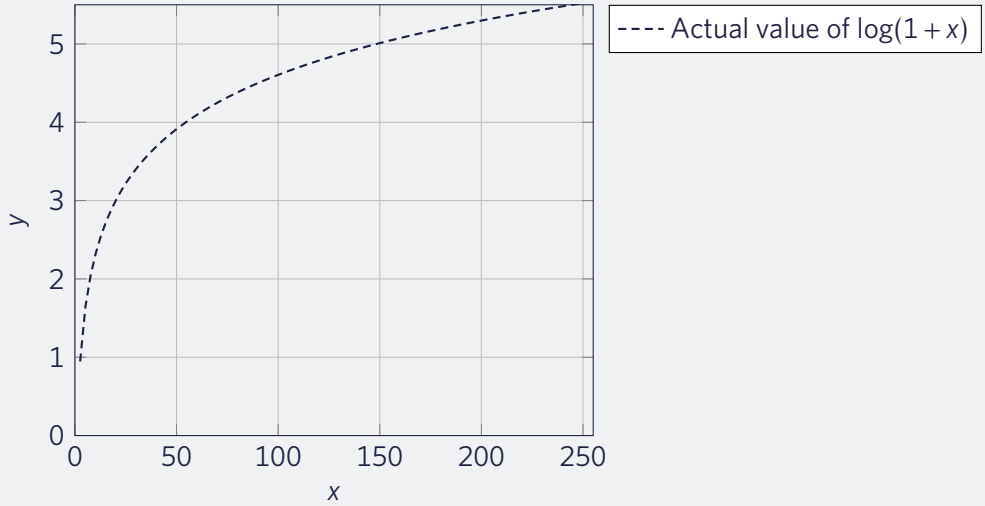
Known closed-form approximation to the logarithm

Khattri presented the following approximation for $\log(1+x)$:

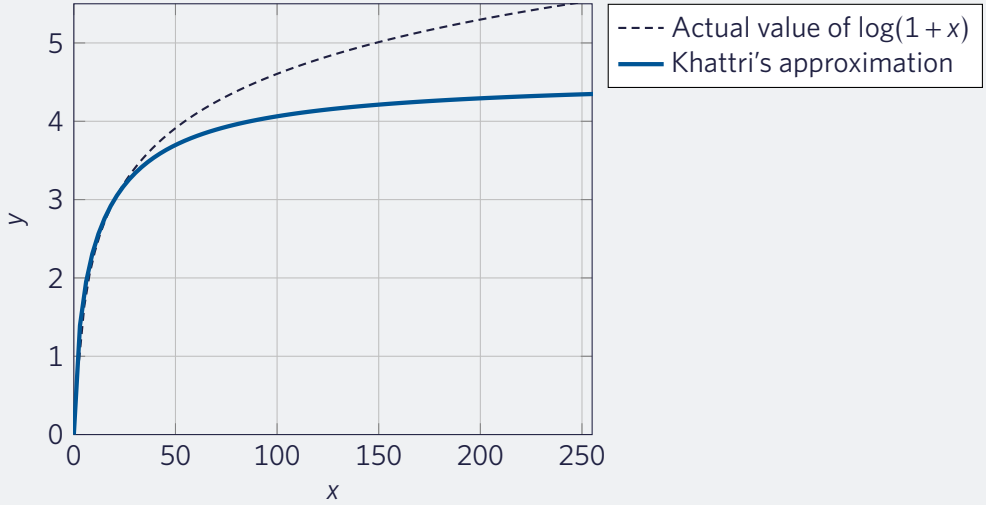
$$\log(1+x) \approx \frac{137x^5 + 2310x^4 + 9870x^3 + 15120x^2 + 7560x}{30x^5 + 900x^4 + 6300x^3 + 16800x^2 + 18900x + 7560}.$$

This was derived by approximating the integral $\int_n^{n+1} \frac{1}{t}\,\mathrm{d}t = \log\left(1 + \frac{1}{n}\right)$ using 5-point Gauss–Legendre quadrature.

Known closed-form approximation to the logarithm



Known closed-form approximation to the logarithm



New closed-form approximation to the logarithm

We present the following approximation:

New closed-form approximation to the logarithm

We present the following approximation:

$$\log(1+x) \approx \frac{a(x)}{b(x)} + \log 16,$$

where

$$a(x) = 137x^5 + 26685x^4 + 617370x^3 - 6498630x^2 - 121239315x - 257804775$$
$$b(x) = 30(x^5 + 405x^4 + 27210x^3 + 488810x^2 + 2536005x + 3122577).$$

Using Python 3.6.5, we implemented this approximation to compute the logarithm of numbers encrypted under the Paillier cryptosystem.

New closed-form approximation to the logarithm

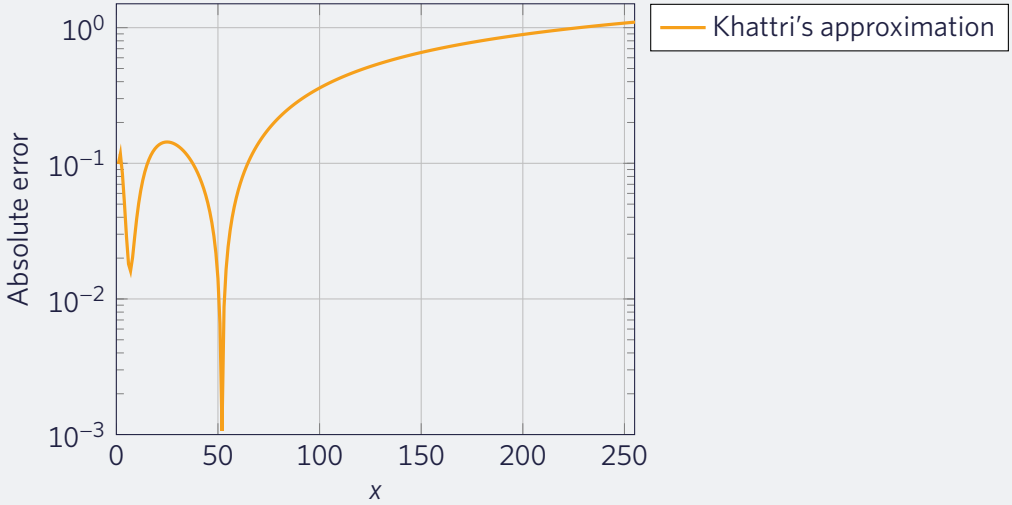
We derived this approximation by approximating the integral

$$\log\left(1+\frac{1}{n}\right) = \int_{n}^{\alpha n+\alpha} \frac{1}{t}\,\mathrm{d}t - \log\alpha,$$

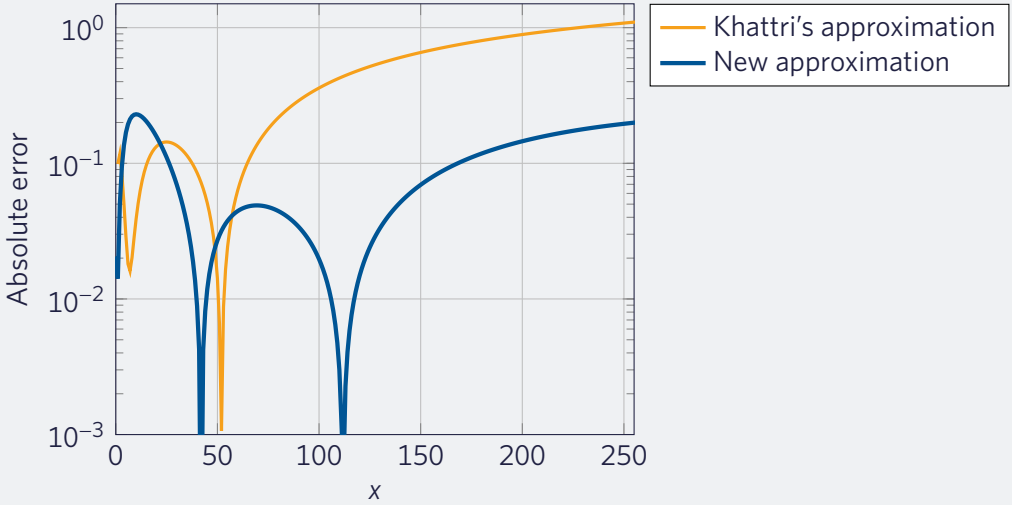
where $\alpha = 1/16$, using 5-point Gauss-Legendre quadrature.

It can be proven numerically that choosing $\alpha = 1/16$ minimizes the absolute error of the approximation from $\log(1+x)$ in the range $x \in [0, 255]$.

Comparison of logarithm approximations



Comparison of logarithm approximations



Comparison of logarithm approximations

Table 1: Comparison of logarithm approximation errors under Paillier for selected inputs

Input	Khattri's approximation	New approximation
$\log 1$	0	2.053729×10^{-2}
$\log 10$	2.833179×10^{-2}	2.280124×10^{-1}
$\log 50$	2.170779×10^{-2}	2.437148×10^{-2}
$\log 100$	3.524916×10^{-1}	2.122159×10^{-2}
$\log 200$	8.859869×10^{-1}	1.441611×10^{-1}
$\log 256$	1.098463×10^{0}	1.993136×10^{-1}

Conventional methods for inverse tangent approximation

Remarks on using conventional methods to approximate the inverse tangent of an encrypted number:

- Series approximation — there are two major series representations for $\arctan x$
  - The Maclaurin series expansion for $\arctan x$ only converges for $-1 \leq x \leq 1$.
  - In 1755, Euler discovered the following series for the inverse tangent,

$$\arctan x = \sum_{n=0}^{\infty} \frac{2^{2n}(n!)^2}{(2n+1)!} \frac{x^{2n+1}}{(1+x^2)^{n+1}},$$

    which converges for all values of x.
- Lookup tables — cannot be done securely as we cannot expose the value of the encrypted number.

New approximation for inverse tangent

We now present the approximation:

New approximation for inverse tangent

We now present the approximation:

$$\arctan x \approx \frac{4\left(225x^9 + 15925x^7 + 144753x^5 + 350595x^3 + 238140x\right)}{15\left(x^{10} + 480x^8 + 11760x^6 + 64120x^4 + 114660x^2 + 63504\right)}.$$

We derived this approximation by approximating the integral $\arctan x = \int_0^x \frac{1}{1+t^2}\,\mathrm{d}t$, using 5-point Gauss–Legendre quadrature.

New approximation for inverse tangent

We compared our new approximation to a partial sum consisting of the first five terms of Euler's series, as it requires similar number of floating-point operations to the new approximation.

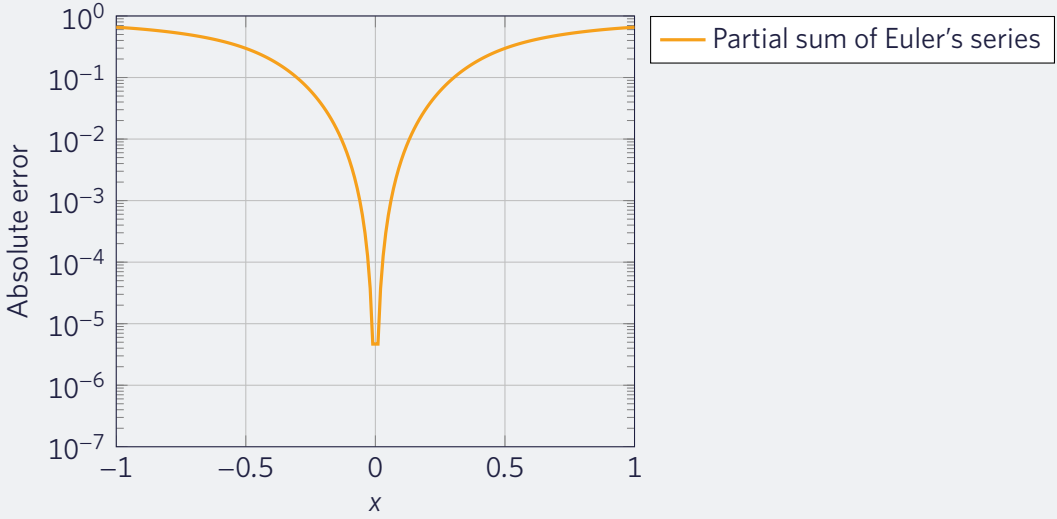
$$\sum_{n=0}^{4} \frac{2^{2n}(n!)^2}{(2n+1)!} \frac{x^{2n+1}}{(1+x^2)^{n+1}} = \frac{965x^9 + 2370x^7 + 2688x^5 + 1470x^3 + 315x}{315\left(x^{10} + 5x^8 + 10x^6 + 10x^4 + 5x^2 + 1\right)}$$

(Euler's series)

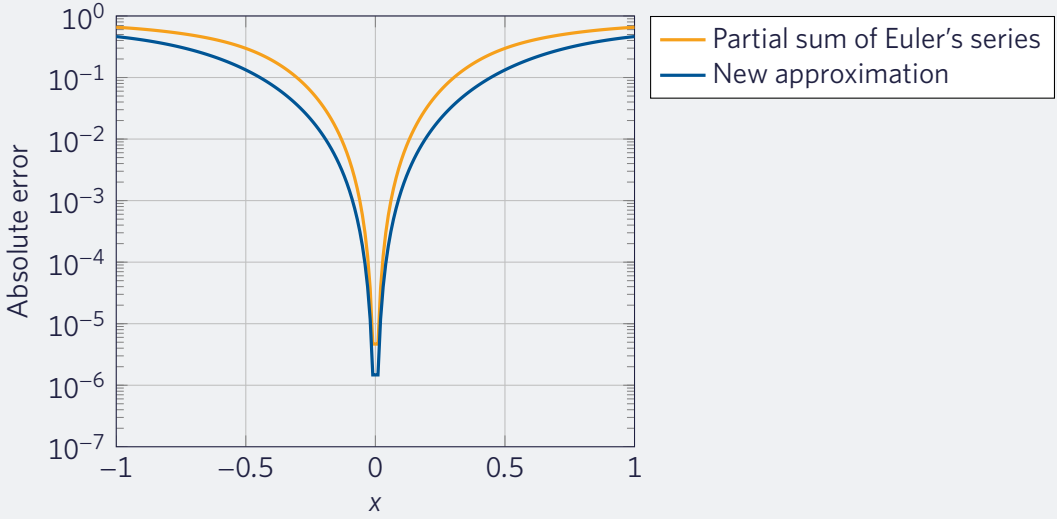
$$\arctan x \approx \frac{4\left(225x^9 + 15925x^7 + 144753x^5 + 350595x^3 + 238140x\right)}{15\left(x^{10} + 480x^8 + 11760x^6 + 64120x^4 + 114660x^2 + 63504\right)}.$$

(New approximation)

Comparison of inverse tangent approximations



Comparison of inverse tangent approximations



Comparison of inverse tangent approximations

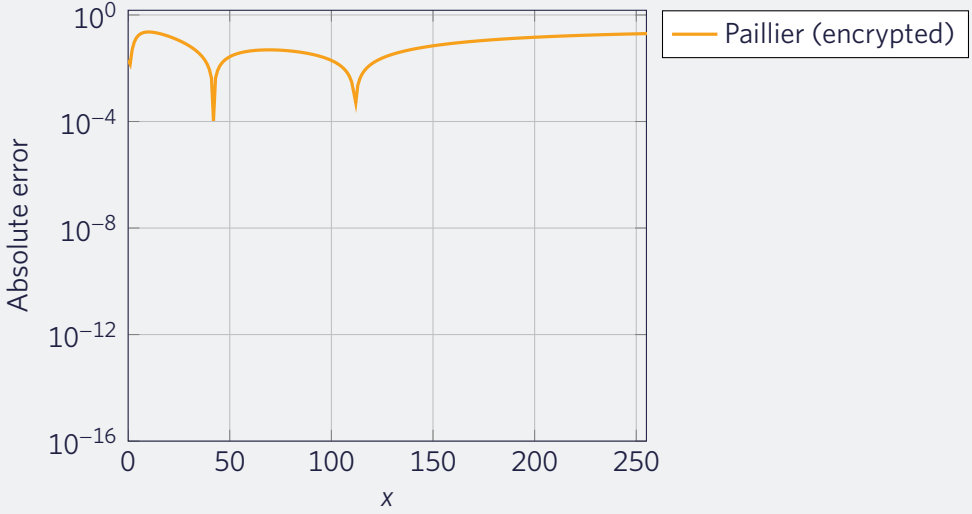
Table 2: Comparison of inverse tangent approximation errors under Paillier for selected inputs

Input	Partial sum of Euler's series	New approximation
$\arctan -1$	6.564089×10^{-1}	4.629557×10^{-1}
$\arctan -0.1$	4.523405×10^{-3}	1.451929×10^{-3}
$\arctan -0.01$	4.665260×10^{-6}	1.472027×10^{-6}
$\arctan 0$	0	0
$\arctan 0.01$	4.665228×10^{-6}	1.472020×10^{-6}
$\arctan 0.1$	4.523405×10^{-3}	1.451929×10^{-3}
$\arctan 1$	6.564089×10^{-1}	4.629557×10^{-1}

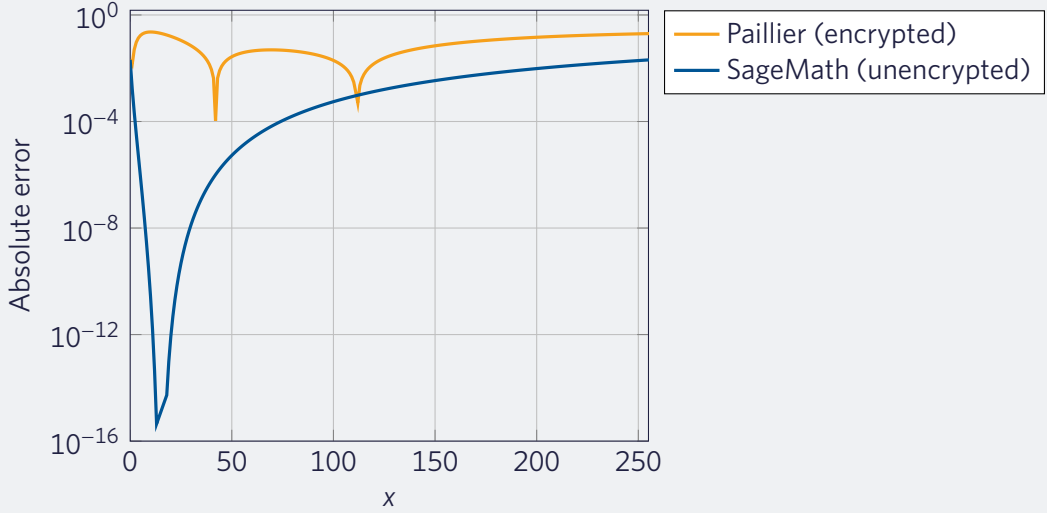
Remarks on new approximations

- Numerical integration can be used when infinite series representations can't be used.
- Approximation for $\log(1+x)$ was tailored for input range $0 \leq x \leq 255$. Different ranges can be accommodated by tweaking the approximation procedure.
- Approximation for $\arctan x$ was tested for the input range $-1 \leq x \leq 1$. Inputs outside this range can be accommodated using trigonometric identities.
- Noise was observed when the approximations were implemented using the Paillier cryptosystem.
  - This is likely due to errors in floating-point representation.
  - Approximations are significantly more accurate when evaluated using SageMath, i.e. unencrypted.

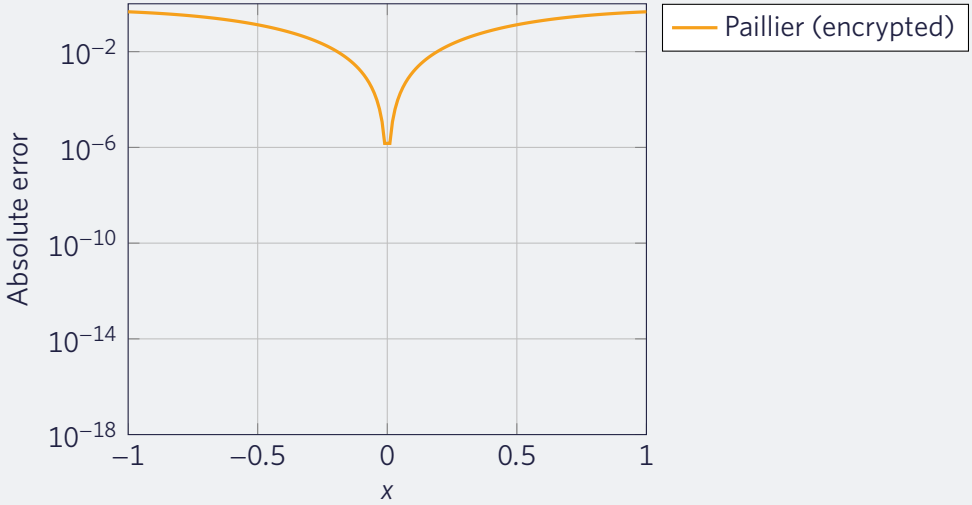
Experimental accuracy of new log approximation



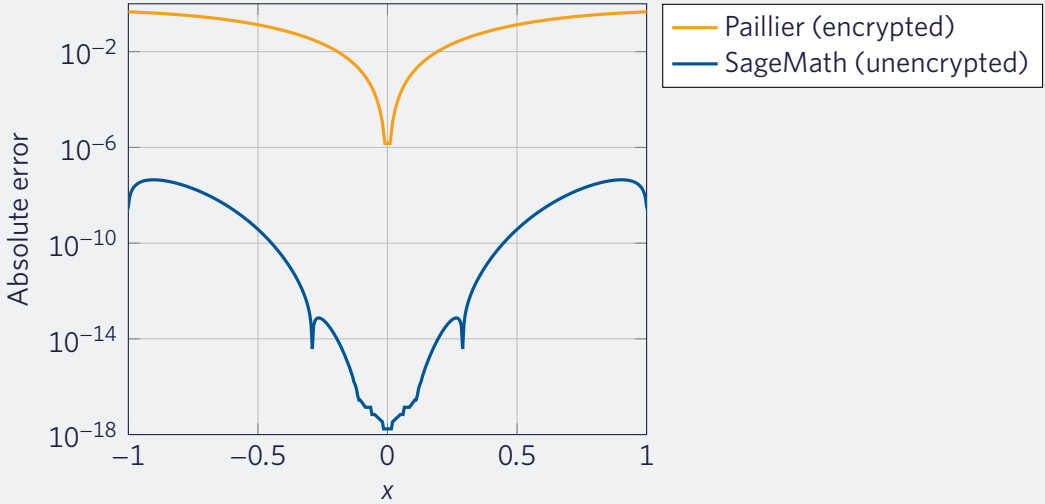
Experimental accuracy of new log approximation



Experimental accuracy of new inverse tangent approximation



Experimental accuracy of new inverse tangent approximation



Conclusion and recommendations

- We have shown new closed-form approximations to the logarithm and inverse tangent functions.
- We have demonstrated how these can be used with the Paillier cryptosystem to allow relatively accurate privacy-preserving transcendental function calculation.
- For further research:
  - Explore other numerical integration methods
  - Explore other homomorphic cryptosystems

Thank you!

Gauss–Legendre quadrature

We first convert the integral to an integral over the interval $[-1, 1]$ using the following transformation:

$$\int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)\mathrm{d}x.$$

Gauss–Legendre quadrature

Then, we approximate the integral using the following summation:

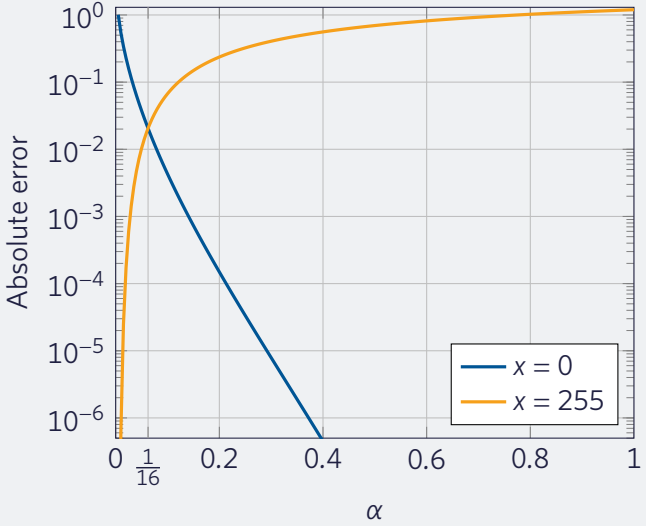
$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{5} w_i f(x_i),$$

where

$$w_1 = 0,$$
$$w_2 = \frac{1}{21}\sqrt{245 - 14\sqrt{70}},$$
$$w_3 = -\frac{1}{21}\sqrt{245 - 14\sqrt{70}},$$
$$w_4 = \frac{1}{21}\sqrt{245 + 14\sqrt{70}},$$
$$w_5 = -\frac{1}{21}\sqrt{245 + 14\sqrt{70}},$$

$$x_1 = \frac{128}{225},$$
$$x_2 = \frac{1}{900}\left(322 + 13\sqrt{70}\right),$$
$$x_3 = \frac{1}{900}\left(322 + 13\sqrt{70}\right),$$
$$x_4 = \frac{1}{900}\left(322 - 13\sqrt{70}\right),$$
$$x_5 = \frac{1}{900}\left(322 - 13\sqrt{70}\right).$$

Optimizing the logarithm approximation



Optimizing the logarithm approximation