CKKS PERFORMANCE

Damien Stehlé

damien.stehle@cryptolab.co.kr

Rennes, November 19, 2024

HEAAN CRYPTO LAB

POST-QUANTUM CRYPTO

Integer Factorization

Given $N = p \cdot q$, find p and q

Discrete Logarithm

Given g and $k * g$, find k
(g lives in some algebraic group)

Lattices
Codes
Systems of quad. Equations
Isogenies
Hash functions

POST-QUANTUM CRYPTO

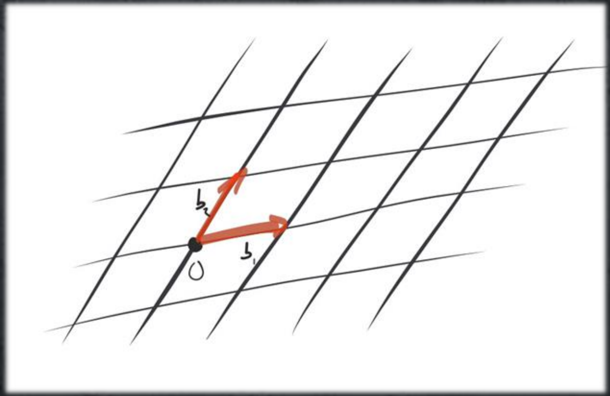
~~Integer Factorization~~

~~Given $N = p \cdot q$, find p and q~~

~~Discrete Logarithm~~

~~Given g and $k * g$, find k~~
~~(g lives in some algebraic group)~~

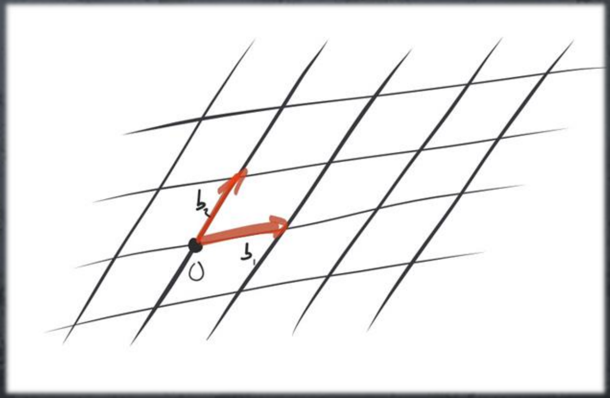
Lattices
Codes
Systems of quad. Equations
Isogenies
Hash functions



HEAAN CRYPTO LAB

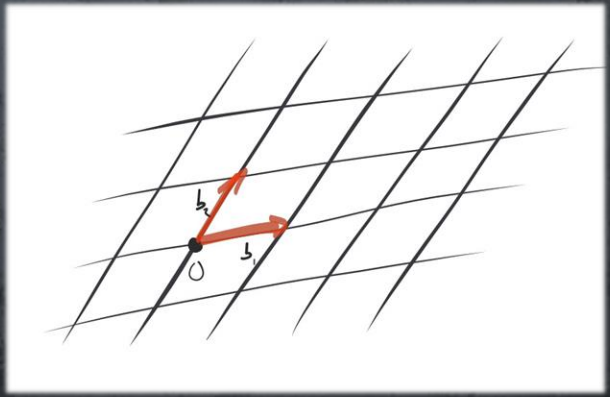
POST-QUANTUM CRYPTO: BEYOND BASIC PRIMITIVES

Lattices
Codes
Systems of quad. Equations
Isogenies
Hash functions



POST-QUANTUM CRYPTO: BEYOND BASIC PRIMITIVES

Lattices
Codes
Systems of quad. Equations
Isogenies
Hash functions

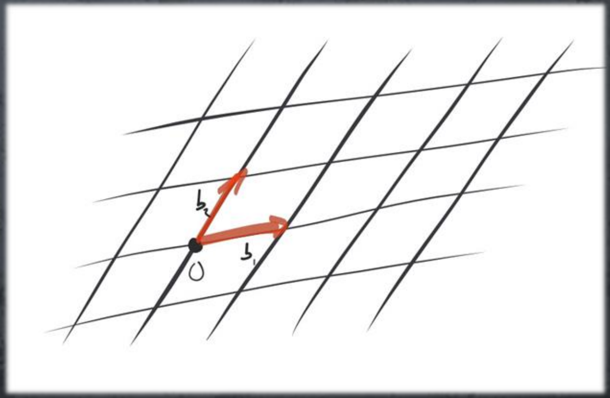


Public key encr.: Kyber/ML-KEM

Signatures: Dilithium/ML-SIG & Falcon

POST-QUANTUM CRYPTO: BEYOND BASIC PRIMITIVES

Lattices
Codes
Systems of quad. Equations
Isogenies
Hash functions



Public key encr.: Kyber/ML-KEM

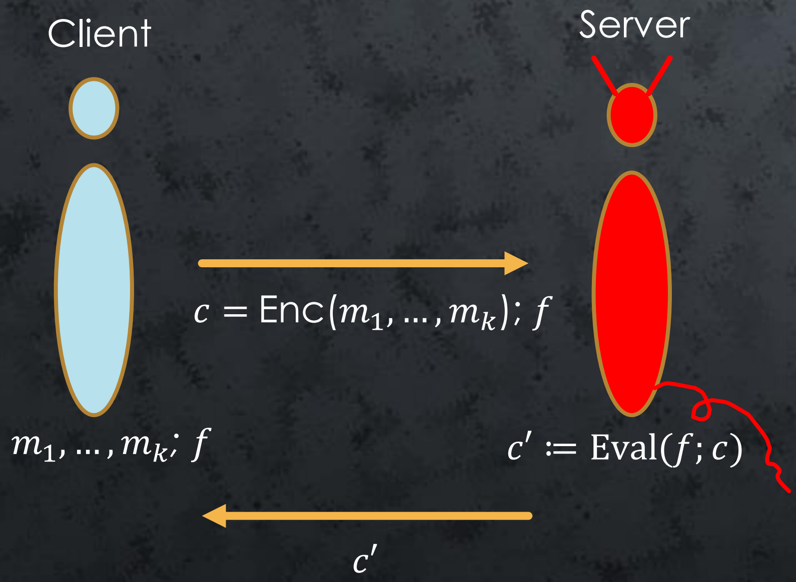
Signatures: Dilithium/ML-SIG & Falcon

New cryptographic functionality

Fully homomorphic encryption (FHE)

- Only possible with lattices
- Post-quantum security

FULLY HOMOMORPHIC ENCRYPTION



Fully homomorphic encryption (FHE) enables arbitrary computations on encrypted data

An FHE has four algorithms:

1. KeyGen: $\cdot \mapsto \mathrm{pk}, \mathrm{sk}, \mathrm{evk}$
2. Enc: $\mathrm{pk}, m \mapsto c$
3. Dec: $\mathrm{sk}, c \mapsto m$
4. Eval: $\mathrm{evk},\ (c_i)_{i=1..k}\ , f \mapsto c$

Correctness:

For any circuit f, for any $m_1, \ldots, m_k$ (with k the arity of f)

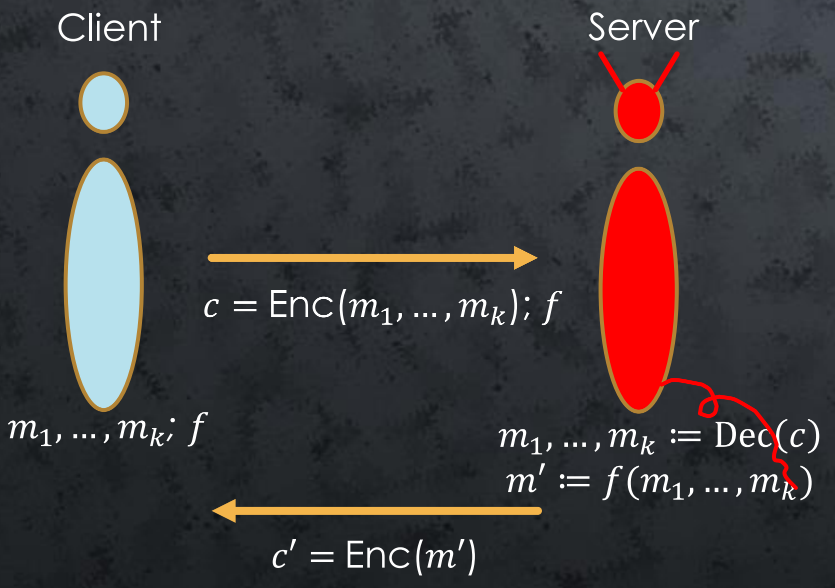
$$\mathbf{Dec}\left(\mathbf{Eval}\left(\left(\mathbf{Enc}(m_i)\right)_{i=1..k} ; f\right)\right)$$

$$= f(m_1, \ldots, m_k)$$

HEAAN CRYPTO LAB

OUTSOURCING COMPUTATIONS

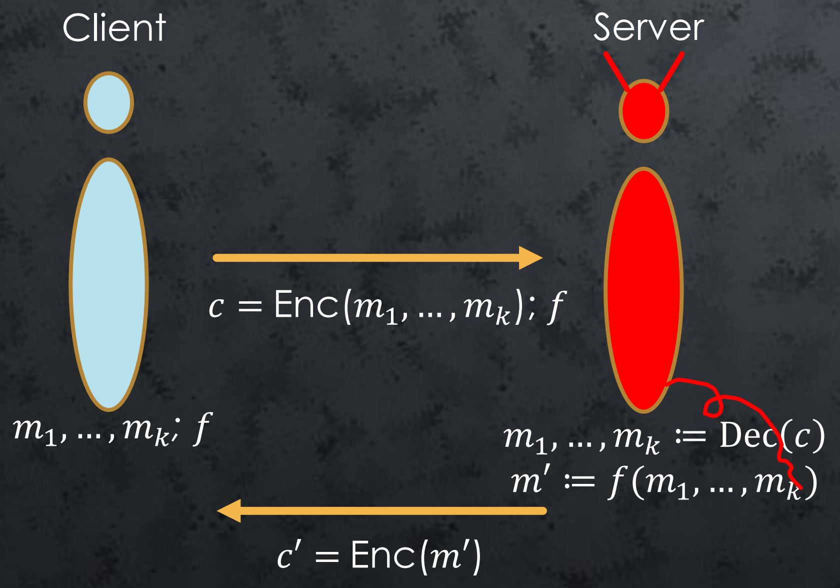
"Trust-based" solution



Server sees the data

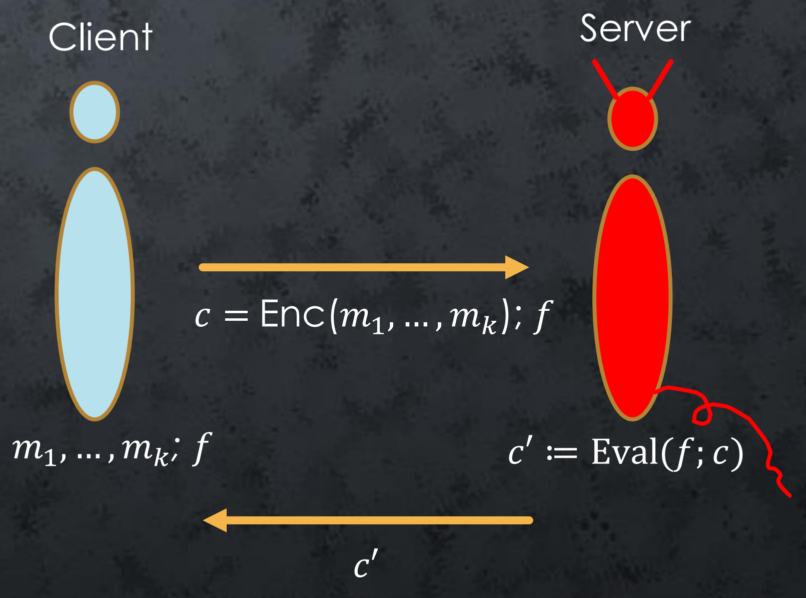
OUTSOURCING COMPUTATIONS

"Trust-based" solution



Server sees the data

FHE-based solution



Server does not see the data

KEY PROPERTIES OF FHE

Computations are as **exact** encrypted as in plaintexts
[Unlike differential privacy or digital twins]

No interactions
[Unlike MPC]

Ciphertexts can be reused
[Unlike MPC]

Security holds under **well-established hardness assumptions**
[Unlike "Confidential Computing"]

KEY USES OF FHE

Externalization of computations while protecting confidentiality

- Heavy computations
- Complex computations
- Proprietary computations

Protection of **databases** that need processing

$\Rightarrow$ Hacking the database server becomes useless

$\Rightarrow$ Sensitive databases can be externalized

Collaboration between entities wishing to protect their own input data

(Requires Threshold-FHE)

Protection of **internal data** against insiders/hackers

FHE IS KICKING OFF

2009: first solution (Gentry) → totally impractical

Now: many libraries → practical for many computations

Soon: hardware accelerations

Next year(?): ISO/IEC standard

- Privacy-preserving medical services
- Privacy-preserving database queries
- Privacy-preserving smart contracts on blockchain
- And more to be found

MAIN FHE SCHEMES

	Plaintext space	Native operations
BFV/BGV (2012)	$\left(\mathrm{F}_{p^k}\right)^{N/k}$	Add, Mult in // Coord. rotation
DM/CGGI (2015)	$\{0,1\}$	Binary gates
CKKS (2017)	$\mathbb{C}^{N/2}$	Add, Mult, Conj in // Coord. rotation

First realization by Gentry (2009)

First efficient libraries: SEAL, HELIB (~2015)

CKKS PLAINTEXTS AND OPS

CKKS is a fully homomorphic encryption scheme:

$$\forall f, m_1, \dots, m_k : \quad \mathrm{Dec}\left(\mathrm{Eval}\left(f;\ \mathrm{Enc}(m_1), \dots, \mathrm{Enc}(m_k)\right)\right) \approx f(m_1, \dots, m_k)$$

Plaintext space: vectors of $\mathbb{C}^{N/2}$ (up to some precision)

- add in //
- multiply in //
- conjugate in //
- rotate the coordinates

CKKS is **level-based**

- mult — consumes 1 level
- add, conj & rot — consume 0 level
- bootstrapping (BTS) — regains level

BOOTSTRAPPING EFFICIENCY

$N = 2^{16}$ Precision ≈ 22 bits Remaining levels: 13	**CPU** Single-thread, AVX512 Intel Xeon Gold 6342 @2.8GHz
Real-BTS ($N/2$ real numbers)	**5.3 s**
Complex-BTS ($N/2$ complex numbers or N real numbers)	**6.9 s**

HEaaN library, binaries available at heaan.it

BOOTSTRAPPING EFFICIENCY

$N = 2^{16}$ Precision ≈ 22 bits Remaining levels: 13	**CPU** Single-thread, AVX512 Intel Xeon Gold 6342 @2.8GHz	**GPU** NVIDIA GeForce RTX 4090
Real-BTS ($N/2$ real numbers)	**5.3 s**	**49 ms**
Complex-BTS ($N/2$ complex numbers or N real numbers)	**6.9 s**	**61 ms**

HEaaN library, binaries available at heaan.it

AMORTIZED COMPUTATIONAL COST

Cost of a ct-ct multiplication, amortized over:

- slots
- a full BTS loop iteration
- several ciphertexts

$$73.6\ ns$$

(GPU, NVIDIA GeForce RTX 4090)

LLAMA2-7B… HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

LLAMA2-7B… HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm dim = 2^7 2^{12} in //

LLAMA2-7B... HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12}	in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3	in //

HEAAN CRYPTO LAB

LLAMA2-7B… HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //

LLAMA2-7B… HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //

HEAAN CRYPTO LAB

LLAMA2-7B... HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //

HEAAN CRYPTO LAB

LLAMA2-7B… HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once

HEAAN CRYPTO LAB

LLAMA2-7B... HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once
RMSNorm	dim = 2^7	2^{12} in //

LLAMA2-7B... HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once
RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{13.4} \times 2^{12} \times 2^7$	2 in //

LLAMA2-7B... HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once
RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{13.4} \times 2^{12} \times 2^7$	2 in //
SILU		$2^{20.4}$ in //

LLAMA2-7B… HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once
RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{13.4} \times 2^{12} \times 2^7$	2 in //
SILU		$2^{20.4}$ in //
pt-ct matrix mult	$2^{12} \times 2^{13.4} \times 2^7$	once

HEAAN CRYPTO LAB

LLAMA2-7B… HOMOMORPHICALLY!

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once
RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{13.4} \times 2^{12} \times 2^7$	2 in //
SILU		$2^{20.4}$ in //
pt-ct matrix mult	$2^{12} \times 2^{13.4} \times 2^7$	once

Repeat 32 times (!#?!!)
And more of the same for each generated token

LLAMA2-7B… HOMOMORPHICALLY!

1 token $\approx 2^{42}$ bit ops

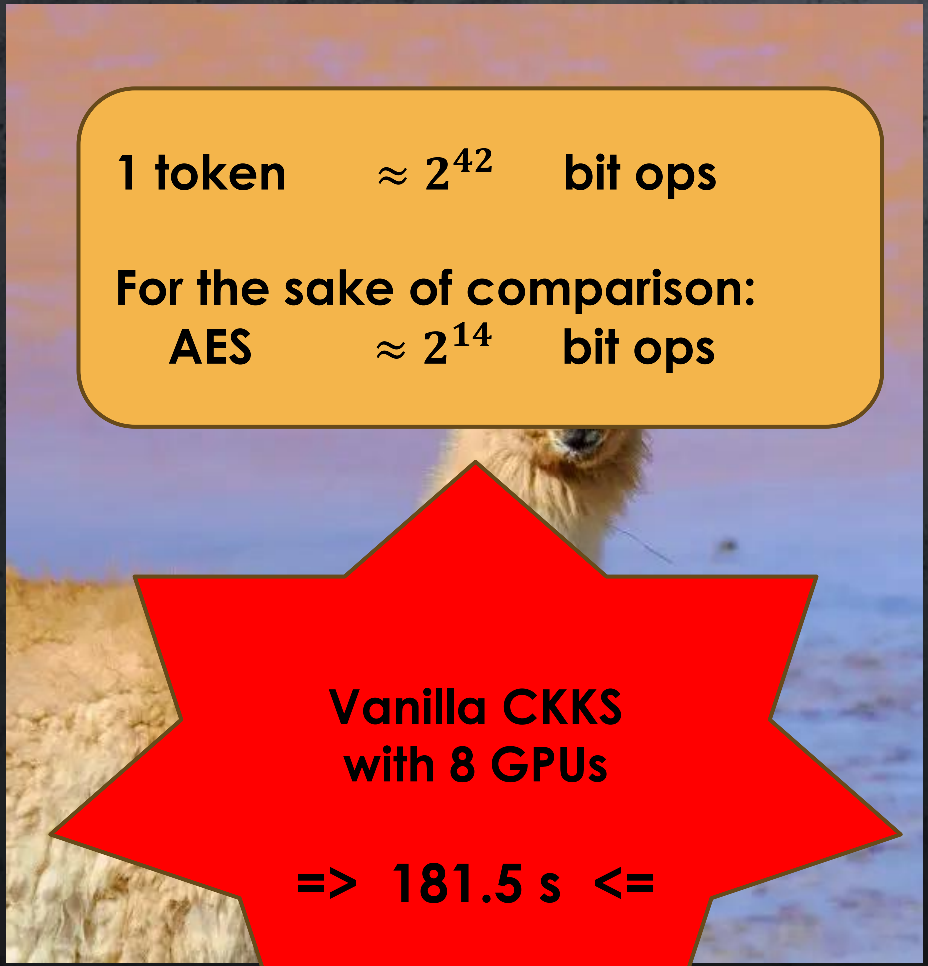
For the sake of comparison:
AES $\approx 2^{14}$ bit ops

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once
RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{13.4} \times 2^{12} \times 2^7$	2 in //
SILU		$2^{20.4}$ in //
pt-ct matrix mult	$2^{12} \times 2^{13.4} \times 2^7$	once

Repeat 32 times (!#?!!)
And more of the same for each generated token

LLAMA2-7B... HOMOMORPHICALLY!



1 token $\approx 2^{42}$ bit ops

For the sake of comparison:
AES $\approx 2^{14}$ bit ops

Vanilla CKKS with 8 GPUs

=> 181.5 s <=

One of Meta's transformer-based LLMs (with 2^7 tokens)

RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	3 in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
Softmax	dim = 2^7	2^{12} in //
ct-ct matrix mult	$2^7 \times 2^7 \times 2^7$	2^5 in //
pt-ct matrix mult	$2^{12} \times 2^{12} \times 2^7$	once
RMSNorm	dim = 2^7	2^{12} in //
pt-ct matrix mult	$2^{13.4} \times 2^{12} \times 2^7$	2 in //
SILU		$2^{20.4}$ in //
pt-ct matrix mult	$2^{12} \times 2^{13.4} \times 2^7$	once

Repeat 32 times (!#?!!)
And more of the same for each generated token

[DMPS24] N. Drucker, G. Moshkowich, T. Pelleg, and H. Shaul. BLEACH: Cleaning errors in discrete computations over CKKS. J. Cryptol., 2024
[BCKS24] Y. Bae, J. H. Cheon, J. Kim, and D. Stehlé. Bootstrapping bits with CKKS. Eurocrypt 2024.
[BKSS24] Y. Bae, J. Kim, D. Stehlé, and E. Suvanto. Bootstrapping integers with CKKS. Asiacrypt 2024.

BINARY CIRCUITS

CKKS is usually thought of as designed for **real/complex numbers**
But it can be used for **binary** computations! [DMPS24]

Bootstrapping can be optimized for such plaintext formats [BCKS24,BKSS24]

[DMPS24] N. Drucker, G. Moshkowich, T. Pelleg, and H. Shaul. BLEACH: Cleaning errors in discrete computations over CKKS. J. Cryptol., 2024
[BCKS24] Y. Bae, J. H. Cheon, J. Kim, and D. Stehlé. Bootstrapping bits with CKKS. Eurocrypt 2024.
[BKSS24] Y. Bae, J. Kim, D. Stehlé, and E. Suvanto. Bootstrapping integers with CKKS. Asiacrypt 2024.

BINARY CIRCUITS

CKKS is usually thought of as designed for **real/complex numbers**
But it can be used for **binary** computations! [DMPS24]

Bootstrapping can be optimized for such plaintext formats [BCKS24,BKSS24]

	CGGI	[DMPS24]	[BCKS24]	[BKSS24]	In a few months?
Throughput (amortized time / binary gate) **single-thread CPU**	~10ms	92.6μs	17.6μs	7.39μs	3μs

(with GPU, it's 100x faster!)

WRAPPING UP

Usecase performance cheatsheet (GPU)

Facial recognition of 1 out of 250K: ~0.2s

Simple CNN (Resnet18, Facenet): ~1s

Training for image classification: a few mins
(demo available at https://autofhe.com/)

LLM inference (Llama2-7B): 3min per token

HEAAN CRYPTO LAB

WRAPPING UP

Usecase performance cheatsheet (GPU)

Facial recognition of 1 out of 250K: ~0.2s

Simple CNN (Resnet18, Facenet): ~1s

Training for image classification: a few mins
(demo available at https://autofhe.com/)

LLM inference (Llama2-7B): 3min per token

Software

LattiGo
OpenFHE
HEaaN

WRAPPING UP

Usecase performance cheatsheet (GPU)

Facial recognition of 1 out of 250K: ~0.2s

Simple CNN (Resnet18, Facenet): ~1s

Training for image classification: a few mins
(demo available at https://autofhe.com/)

LLM inference (Llama2-7B): 3min per token

Software

LattiGo
OpenFHE
HEaaN

HEaaN CPU&GPU binaries available at https://heaan.it/

WRAPPING UP

Usecase performance cheatsheet (GPU)

Facial recognition of 1 out of 250K: ~0.2s

Simple CNN (Resnet18, Facenet): ~1s

Training for image classification: a few mins
(demo available at https://autofhe.com/)

LLM inference (Llama2-7B): 3min per token

Software

LattiGo
OpenFHE
HEaaN

HEaaN CPU&GPU binaries available at https://heaan.it/

ANY QUESTIONS?

damien.stehle@cryptolab.co.kr

HEAAN CRYPTO LAB