More Efficient Functional Bootstrapping for General Functions in Polynomial Modulus

Han Xia[1,2], Feng-Hao Liu[3], and Han Wang[1,2]

[1] Institute of Information Engineering, CAS
[2] University of Chinese Academy of Sciences
[3] Washington State University

December 6, TCC 2024

Background

Fully Homomorphic Encryption (FHE)

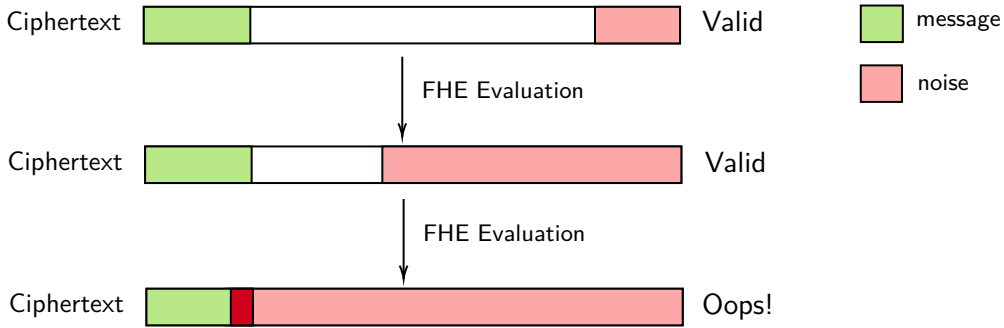
FHE allows arbitrary computations on encrypted data.

FHE ciphertexts are “noisy” (mostly based on the LWE family):

Background

Fully Homomorphic Encryption (FHE)

FHE allows arbitrary computations on encrypted data.
FHE ciphertexts are "noisy" (mostly based on the LWE family):



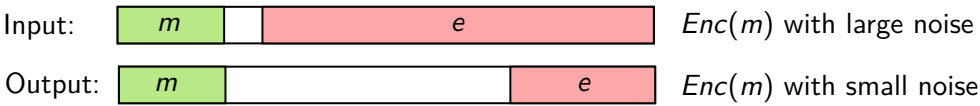
Too large noise $\Longrightarrow$ Incorrect decryption result.

Background

Bootstrapping

Bootstrapping [Gentry'09]: Refresh the noise before it becomes too large.
Idea: Homomorphically evaluate the decryption circuit.

(Regular) Bootstrapping



Background

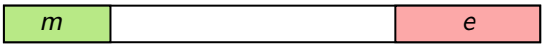
Bootstrapping

Bootstrapping [Gentry'09]: Refresh the noise before it becomes too large.
Idea: Homomorphically evaluate the decryption circuit.

(Regular) Bootstrapping

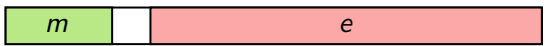
Input: 

$Enc(m)$ with large noise

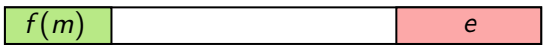
Output: 

$Enc(m)$ with small noise

Functional Bootstrapping

Input: 

$Enc(m)$ with large noise

Output: 

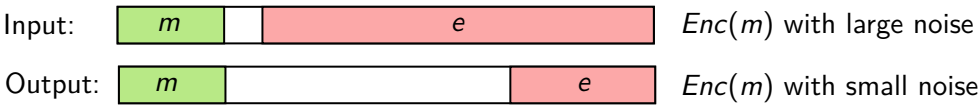
$Enc(f(m))$ with small noise

Background

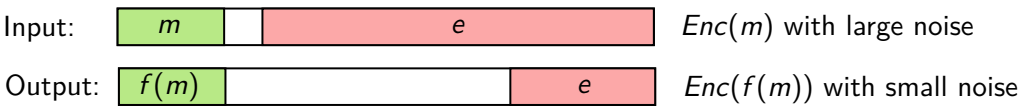
Bootstrapping

Bootstrapping [Gentry'09]: Refresh the noise before it becomes too large.
Idea: Homomorphically evaluate the decryption circuit.

(Regular) Bootstrapping



Functional Bootstrapping



If f is the identity map, they are equivalent.

Hope: They have similar efficiency even for general functions!

Bootstrapping in Polynomial Modulus

The FHEW/TFHE (Regular) Gate-Bootstrapping
[Ducas-Micciancio'15, Chillotti-Gama-Georgieva-Izabachène'16]

Ring: The power-of-2 cyclotomic ring $\mathbb{Z}[\zeta_{2N}] \cong \mathbb{Z}[X]/(X^N + 1)$.

Extracted problem:
Given $\mathsf{Enc}(v \cdot X^{-\alpha})$ where $\alpha \in [0, 2N - 1]$
How to evaluate the MSB (most significant bit) of α?

Bootstrapping in Polynomial Modulus

The FHEW/TFHE (Regular) Gate-Bootstrapping
[Ducas-Micciancio'15, Chillotti-Gama-Georgieva-Izabachène'16]

Ring: The power-of-2 cyclotomic ring $\mathbb{Z}[\zeta_{2N}] \cong \mathbb{Z}[X]/(X^N+1)$.

Extracted problem:
Given $\mathsf{Enc}(v \cdot X^{-\alpha})$ where $\alpha \in [0, 2N-1]$
How to evaluate the MSB (most significant bit) of α?

Solution: Set

$$v = 1 + X + X^2 + \cdots + X^{N-1}$$

The constant term of $v \cdot X^{-\alpha}$ is

$$\mathsf{Const}(v \cdot X^{-\alpha}) = \begin{cases} 1 & \text{if} \quad 0 \le \alpha \le N-1 \\ -1 & \text{if} \quad N \le \alpha \le 2N-1 \end{cases}$$

This is sufficient for **(homomorphically) distinguishing the MSB** of α.

Functional Bootstrapping in Polynomial Modulus

Generalized problem:
Given $\mathrm{Enc}(v \cdot X^{-\alpha})$ where $\alpha \in [0, 2N-1]$
How to evaluate $f(\alpha)$ for $f : \mathbb{Z}_{2N} \to \mathbb{Z}_h$?

Functional Bootstrapping in Polynomial Modulus

Generalized problem:
Given $\mathrm{Enc}(v \cdot X^{-\alpha})$ where $\alpha \in [0, 2N-1]$
How to evaluate $f(\alpha)$ for $f : \mathbb{Z}_{2N} \to \mathbb{Z}_h$?

Solution: ([Boura-Gama-Georgieva-Jetchev'19]) Set

$$v = f(0) + f(1) \cdot X + f(2) \cdot X^2 + \cdots + f(N-1) \cdot X^{N-1}$$

The constant term of $v \cdot X^{-\alpha}$ is

$$\mathrm{Const}(v \cdot X^{-\alpha}) = \begin{cases} f(\alpha) & \text{if} \quad 0 \le \alpha \le N-1 \\ -f(\alpha - N) & \text{if} \quad N \le \alpha \le 2N-1 \end{cases}$$

Nothing else to do if f is "negacyclic".

Functional Bootstrapping in Polynomial Modulus

Generalized problem:
Given $\mathsf{Enc}(v \cdot X^{-\alpha})$ where $\alpha \in [0, 2N-1]$
How to evaluate $f(\alpha)$ for $f : \mathbb{Z}_{2N} \to \mathbb{Z}_h$?

Solution: ([Boura-Gama-Georgieva-Jetchev'19]) Set

$$v = f(0) + f(1) \cdot X + f(2) \cdot X^2 + \cdots + f(N-1) \cdot X^{N-1}$$

The constant term of $v \cdot X^{-\alpha}$ is

$$\mathsf{Const}(v \cdot X^{-\alpha}) = \begin{cases} f(\alpha) & \text{if} \quad 0 \leq \alpha \leq N-1 \\ -f(\alpha - N) & \text{if} \quad N \leq \alpha \leq 2N-1 \end{cases}$$

Nothing else to do if f is "negacyclic".

Conclusion: Functional bootstrapping for negacyclic functions is as efficient as regular bootstrapping over power-of-2 cyclotomic rings using Constant Sample Extraction.

Functional Bootstrapping in Polynomial Modulus

Attempts to Support General Functions

To bootstrap LWE ciphertexts with modulus q:

	# of Regular Bootstrapping	Minimal Cyclotomic Index	Type of Cyclotomic Index	Without Additional Assumptions
[Chillotti-Ligier-Orfila-Tap'21] (ASIACRYPT 2021)	2	$2q$	Power-of-2	Yes
	3	q	Power-of-2	Yes
[Kluczniak-Schild'23] (TCHES 2023)	> 2	q	Power-of-2	Yes
[Liu-Micciancio-Polyakov'22] (ASIACRYPT 2022)	2	$2q$	Power-of-2	No
	3	$4q$	Power-of-2	Yes
[Bergerat-Boudi-Bourgerie...'23] (JoC 2023)	> 2	$O(q)$	Power-of-2	Yes/No

Functional Bootstrapping in Polynomial Modulus

Attempts to Support General Functions

To bootstrap LWE ciphertexts with modulus q:

	# of Regular Bootstrapping	Minimal Cyclotomic Index	Type of Cyclotomic Index	Without Additional Assumptions
[Chillotti-Ligier-Orfila-Tap'21] (ASIACRYPT 2021)	2	$2q$	Power-of-2	Yes
	3	q	Power-of-2	Yes
[Kluczniak-Schild'23] (TCHES 2023)	> 2	q	Power-of-2	Yes
[Liu-Micciancio-Polyakov'22] (ASIACRYPT 2022)	2	$2q$	Power-of-2	No
	3	$4q$	Power-of-2	Yes
[Bergerat-Boudi-Bourgerie...'23] (JoC 2023)	> 2	$O(q)$	Power-of-2	Yes/No
Ours	$1 + o(1)$	q	General	Yes

Our Framework

To support general functions,

$$\cancel{\text{Constant Sample Extraction}} \longrightarrow \textbf{EvalFunc}$$

EvalFunc:

Input: $\mathrm{Enc}(\zeta_q^{\alpha})$ and arbitrary $f : \mathbb{Z}_q \to \mathbb{Z}_h$

Output: $\mathrm{Enc}(f(\alpha))$

Complexity: $O(\log n)$ **HomMul** (n is the lattice dimension)

Our Framework

To support general functions,

$$\cancel{\text{Constant Sample Extraction}} \longrightarrow \textbf{EvalFunc}$$

EvalFunc:

Input: $\mathrm{Enc}(\zeta_q^\alpha)$ and arbitrary $f : \mathbb{Z}_q \to \mathbb{Z}_h$

Output: $\mathrm{Enc}(f(\alpha))$

Complexity: $O(\log n)$ **HomMul** (n is the lattice dimension)

New Workflow

1. (≈Regular bootstrapping) Use n **HomMul** to obtain $\mathrm{Enc}(\zeta_q^\alpha)$
2. (Post-processing) Use **EvalFunc** to obtain $\mathrm{Enc}(f(\alpha))$

Our Framework

To support general functions,

$$\text{~~Constant Sample Extraction~~} \longrightarrow \textbf{EvalFunc}$$

EvalFunc:

Input: $\mathrm{Enc}(\zeta_q^\alpha)$ and arbitrary $f : \mathbb{Z}_q \to \mathbb{Z}_h$

Output: $\mathrm{Enc}(f(\alpha))$

Complexity: $O(\log n)$ **HomMul** (n is the lattice dimension)

New Workflow

1. ($\approx$Regular bootstrapping) Use n **HomMul** to obtain $\mathrm{Enc}(\zeta_q^\alpha)$
2. (Post-processing) Use **EvalFunc** to obtain $\mathrm{Enc}(f(\alpha))$

Efficiency ratio: $\dfrac{\text{Our functional bootstrapping}}{\text{Regular bootstrapping}} = \dfrac{n + O(\log n)}{n} = 1 + o(1)$

Construct **EvalFunc**

To construct **EvalFunc**, we adopt the technique of

Homomorphic Equality Test

[Alperin-Sheriff-Peikert'14] (CRYPTO 2014) gave a method over plain-LWE.

Construct **EvalFunc**

To construct **EvalFunc**, we adopt the technique of

Homomorphic Equality Test

[Alperin-Sheriff-Peikert'14] (CRYPTO 2014) gave a method over plain-LWE.

Homomorphic Equality Test over Rings

Define the equality test for the exponent of ζ_q as

$$\mathsf{EqT}(\zeta_q^{\alpha}, \beta) = \begin{cases} 1 & \text{if} \quad \alpha = \beta \mod q \\ 0 & \text{if} \quad \alpha \neq \beta \mod q \end{cases},$$

Construct **EvalFunc**

To construct **EvalFunc**, we adopt the technique of

Homomorphic Equality Test

[Alperin-Sheriff-Peikert'14] (CRYPTO 2014) gave a method over plain-LWE.

Homomorphic Equality Test over Rings

Define the equality test for the exponent of ζ_q as

$$\mathsf{EqT}(\zeta_q^{\alpha}, \beta) = \begin{cases} 1 & \text{if} \quad \alpha = \beta \mod q \\ 0 & \text{if} \quad \alpha \neq \beta \mod q \end{cases},$$

then for any $f : \mathbb{Z}_q \to \mathbb{Z}_h$ and $\alpha \in \mathbb{Z}_q$, we have

$$f(\alpha) = \sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \mathsf{EqT}(\zeta_q^{\alpha}, \beta).$$

Any discrete function can be represented as a linear combination of EqT**.**

Instantiate EqT

Prime Cyclotomic Rings - A Warm-Up

For a prime q, the trace is close to the equality test:

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_q^{\alpha-\beta}\right) = \begin{cases} q-1 & \text{if} \quad \alpha = \beta \mod q \\ -1 & \text{otherwise} \end{cases}.$$

Instantiate EqT

Prime Cyclotomic Rings - A Warm-Up

For a prime q, the trace is close to the equality test:

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_q^{\alpha-\beta}\right)+1=\begin{cases} q & \text{if} \quad \alpha=\beta \mod q \\ 0 & \text{otherwise} \end{cases}.$$

Instantiate EqT

Prime Cyclotomic Rings - A Warm-Up

For a prime q, the trace is close to the equality test:

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_q^{\alpha-\beta}\right)+1=\begin{cases} q & \text{if} \quad \alpha=\beta \mod q \\ 0 & \text{otherwise} \end{cases}.$$

For any $f:\mathbb{Z}_q \leftarrow \mathbb{Z}_h$ and $\alpha \in \mathbb{Z}_q$, we have

$$\begin{aligned} & \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\zeta_q^{\alpha-\beta}\right)+\sum_{\beta\in\mathbb{Z}_q} f(\beta) \\ =& \sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_q^{\alpha-\beta}\right)+1\right) \\ =& \; q\cdot f(\alpha). \qquad \text{(we can easily remove the factor } q\text{)} \end{aligned}$$

Instantiate EqT

Prime Cyclotomic Rings - A Warm-Up

For a prime q, the trace is close to the equality test:

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_q^{\alpha-\beta}\right)+1=\begin{cases} q & \text{if} \quad \alpha=\beta \mod q \\ 0 & \text{otherwise} \end{cases}.$$

For any $f:\mathbb{Z}_q \leftarrow \mathbb{Z}_h$ and $\alpha \in \mathbb{Z}_q$, we have

$$\begin{aligned} & \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\zeta_q^{\alpha-\beta}\right)+\sum_{\beta\in\mathbb{Z}_q} f(\beta) \\ =& \sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_q^{\alpha-\beta}\right)+1\right) \\ =& \; q\cdot f(\alpha). \qquad \text{(we can easily remove the factor } q\text{)} \end{aligned}$$

- ✅ Thanks to the $\mathbb{Q}$-linearity of trace, all equality tests are perfromed **in parallel**.
- ✅ **EvalFunc** only requires 1 **EvalTr**.

Instantiate EqT

Attempt to More General Cases

Not for general q...

Example: For prime-power $q = p^r$:

$$\mathrm{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^i) = \begin{cases} \varphi(q) & \text{if } \ i = 0 \bmod q \\ -p^{r-1} & \text{if } \ i = 0 \bmod p^{r-1} \ \text{ and } \ i \neq 0 \bmod q \\ 0 & \text{otherwise.} \end{cases}$$

☹ There are three branches.

Instantiate EqT

Attempt to More General Cases

Not for general q...

Example: For prime-power $q = p^r$:

$$\mathrm{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^i) = \begin{cases} \varphi(q) & \text{if } \ i = 0 \bmod q \\ -p^{r-1} & \text{if } \ i = 0 \bmod p^{r-1} \ \text{ and } \ i \neq 0 \bmod q \\ 0 & \text{otherwise.} \end{cases}$$

☹ There are three branches.

Consider the equality test in [Abla-**Liu-Wang**-Wang'21] (TCC 2021):

$$\sum_{i=0}^{q-1} \zeta_q^{(\alpha-\beta)\cdot i} = 1 + \zeta_q^{\alpha-\beta} + \cdots + \zeta_q^{(q-1)(\alpha-\beta)} = \begin{cases} q & \text{if } \ \alpha = \beta \bmod q \\ 0 & \text{if } \ \alpha \neq \beta \bmod q \end{cases}.$$

Instantiate EqT

Attempt to More General Cases

Not for general q...

Example: For prime-power $q = p^r$:

$$\mathrm{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^i) = \begin{cases} \varphi(q) & \text{if } \ i = 0 \bmod q \\ -p^{r-1} & \text{if } \ i = 0 \bmod p^{r-1} \ \text{ and } \ i \neq 0 \bmod q \\ 0 & \text{otherwise.} \end{cases}$$

☹ There are three branches.

Consider the equality test in [Abla-**Liu-Wang**-Wang'21] (TCC 2021):

$$\sum_{i=0}^{q-1} \zeta_q^{(\alpha-\beta)\cdot i} = 1 + \zeta_q^{\alpha-\beta} + \cdots + \zeta_q^{(q-1)(\alpha-\beta)} = \begin{cases} q & \text{if } \ \alpha = \beta \bmod q \\ 0 & \text{if } \ \alpha \neq \beta \bmod q \end{cases}.$$

This equation works for any q but

- ☹ Each equality test requires q **HomMul**
- ☹ The required q equality tests cannot be computed in parallel

Instantiate EqT

Attempt to More General Cases

Fix the representative set

$$\mathbb{Z}_q = \{0, 1, 2, \ldots, q-1\}$$
$$\mathbb{Z}_q^* = \{x \in \mathbb{Z}_q \mid \gcd(x, q) = 1\}$$

Instantiate EqT

Attempt to More General Cases

Fix the representative set

$$\mathbb{Z}_q = \{0, 1, 2, \ldots, q-1\}$$
$$\mathbb{Z}_q^* = \{x \in \mathbb{Z}_q \mid \gcd(x, q) = 1\}$$

Then consider the equality test in [Abla-**Liu**-**Wang**-Wang'21] and the algebraic trace:

$$\mathsf{EqT}(\zeta_q^{\alpha}, \beta) = \sum_{i \in \mathbb{Z}_q} \zeta_q^{(\alpha-\beta)\cdot i}$$
$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^{\alpha-\beta}) = \sum_{i \in \mathbb{Z}_q^*} \zeta_q^{(\alpha-\beta)\cdot i}$$

🤔 Maybe we can bridge this gap by finding relations between $\mathbb{Z}_q$ and $\mathbb{Z}_q^*$.

Instantiate EqT

New Partition for $\mathbb{Z}_q$

Key observation: For a prime-power $q = p^r$:

$$\mathbb{Z}_{p^r} = \mathbb{Z}_{p^r}^* \cup p \cdot \mathbb{Z}_{p^{r-1}}^* \cup \cdots \cup p^{r-1} \cdot \mathbb{Z}_p^* \cup \{0\}$$

Instantiate EqT

New Partition for $\mathbb{Z}_q$

Key observation: For a prime-power $q = p^r$:

$$\mathbb{Z}_{p^r} = \mathbb{Z}_{p^r}^* \cup p \cdot \mathbb{Z}_{p^{r-1}}^* \cup \cdots \cup p^{r-1} \cdot \mathbb{Z}_p^* \cup \{0\}$$

Example: For $q = 2^3$,

$$\begin{aligned}
\mathbb{Z}_8 &= \{0,1,2,3,4,5,6,7\} = \{1,3,5,7\} \cup \{0,2,4,6\} \\
&= \{1,3,5,7\} \cup 2 \cdot \{0,1,2,3\} \\
&= \mathbb{Z}_8^* \cup 2\mathbb{Z}_4 \\
\mathbb{Z}_4 &= \{0,1,2,3\} = \{1,3\} \cup \{0,2\} = \{1,3\} \cup 2 \cdot \{0,1\} = \mathbb{Z}_4^* \cup 2\mathbb{Z}_2 \\
\mathbb{Z}_2 &= \{0,1\} = \{1\} \cup \{0\} = \mathbb{Z}_2^* \cup \{0\}.
\end{aligned}$$

Putting them all together, we have the partition of $\mathbb{Z}_8$:

$$\mathbb{Z}_8 = \mathbb{Z}_8^* \cup 2\mathbb{Z}_4^* \cup 4\mathbb{Z}_2^* \cup \{0\}.$$

Instantiate EqT

Relate EqT with Trace

Based on our new partition

$$\mathbb{Z}_8 = \mathbb{Z}_8^* \cup 2\mathbb{Z}_4^* \cup 4\mathbb{Z}_2^* \cup \{0\},$$

we can relate the equality test with the trace:

$$\begin{aligned}
\mathsf{EqT}(\zeta_8^{\alpha}, \beta) &= \sum_{i \in \mathbb{Z}_8} \zeta_8^{(\alpha-\beta)\cdot i} \\
&= \sum_{i \in \mathbb{Z}_8^*} \zeta_8^{i\cdot(\alpha-\beta)} + \sum_{i \in \mathbb{Z}_4^*} \zeta_8^{2i\cdot(\alpha-\beta)} + \sum_{i \in \mathbb{Z}_2^*} \zeta_8^{4i\cdot(\alpha-\beta)} + \zeta_8^{0\cdot(\alpha-\beta)} \\
&= \sum_{i \in \mathbb{Z}_8^*} \zeta_8^{i\cdot(\alpha-\beta)} + \sum_{i \in \mathbb{Z}_4^*} \zeta_4^{i\cdot(\alpha-\beta)} + \sum_{i \in \mathbb{Z}_2^*} \zeta_2^{i\cdot(\alpha-\beta)} + 1 \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}}(\zeta_8^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}}(\zeta_4^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}(\zeta_2^{\alpha-\beta}) + 1
\end{aligned}$$

Instantiate EqT

The Prime-Power Case

To evaluate the new trace-based equality test:

$$\mathsf{EqT}(\zeta_8^{\alpha}, \beta) = \mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}}(\zeta_8^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}}(\zeta_4^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}(\zeta_2^{\alpha-\beta}) + 1$$

Instantiate EqT

The Prime-Power Case

To evaluate the new trace-based equality test:

$$\mathsf{EqT}(\zeta_8^{\alpha}, \beta) = \mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}}(\zeta_8^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}}(\zeta_4^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}(\zeta_2^{\alpha-\beta}) + 1$$

we can use the **transitivity** of the trace: (Ignore that $\mathbb{Q}(\zeta_2) = \mathbb{Q}$)

$$\begin{array}{ccccccc} \mathbb{Q}(\zeta_8) & \xrightarrow{\mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}(\zeta_4)}} & \mathbb{Q}(\zeta_4) & \xrightarrow{\mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}(\zeta_2)}} & \mathbb{Q}(\zeta_2) & \xrightarrow{\mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}} & \mathbb{Q} \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ \zeta_8^{\alpha-\beta} & & +\zeta_4^{\alpha-\beta} & & +\zeta_2^{\alpha-\beta} & & +1 \end{array}$$

$$\mathsf{EqT}(\zeta_8^{\alpha}, \beta) = \mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}(\zeta_2)}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}(\zeta_4)}(\zeta_8^{\alpha-\beta}) + \zeta_4^{\alpha-\beta}\right) + \zeta_2^{\alpha-\beta}\right) + 1$$

Instantiate EqT

The Prime-Power Case

To evaluate the new trace-based equality test:

$$\mathsf{EqT}(\zeta_8^{\alpha}, \beta) = \mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}}(\zeta_8^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}}(\zeta_4^{\alpha-\beta}) + \mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}(\zeta_2^{\alpha-\beta}) + 1$$

we can use the **transitivity** of the trace: (Ignore that $\mathbb{Q}(\zeta_2) = \mathbb{Q}$)

$$\begin{array}{ccccccc} \mathbb{Q}(\zeta_8) & \xrightarrow{\mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}(\zeta_4)}} & \mathbb{Q}(\zeta_4) & \xrightarrow{\mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}(\zeta_2)}} & \mathbb{Q}(\zeta_2) & \xrightarrow{\mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}} & \mathbb{Q} \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ \zeta_8^{\alpha-\beta} & & +\zeta_4^{\alpha-\beta} & & +\zeta_2^{\alpha-\beta} & & +1 \end{array}$$

$$\mathsf{EqT}(\zeta_8^{\alpha}, \beta) = \mathsf{Tr}_{\mathbb{Q}(\zeta_2)/\mathbb{Q}}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_4)/\mathbb{Q}(\zeta_2)}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}(\zeta_4)}(\zeta_8^{\alpha-\beta}) + \zeta_4^{\alpha-\beta}\right) + \zeta_2^{\alpha-\beta}\right) + 1$$

- ✅ All equality tests **in parallel** (thanks to the linearity of the trace)
- ✅ Only 1 $\mathbf{EvalTr}_{\mathbb{Q}(\zeta_8)/\mathbb{Q}}$ involved (thanks to the transitivity of the trace)

Instantiate EqT

The Prime-Power Case

This equality test works for any prime power $q = p^r$:

$$\mathsf{EqT}(\zeta_q^{\alpha}, \beta) = 1 + \sum_{i=1}^{r} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}} \left(\zeta_{p^i}^{\alpha-\beta} \right)$$

Problem:

We need $\mathsf{Enc}(\zeta_{p^i}^{\alpha})$ for all i.

One regular bootstrapping only produces $\mathsf{Enc}(\zeta_{p^r}^{\alpha})$.

Instantiate EqT

The Prime-Power Case

This equality test works for any prime power $q = p^r$:

$$\mathsf{EqT}(\zeta_q^\alpha, \beta) = 1 + \sum_{i=1}^{r} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}}\left(\zeta_{p^i}^{\alpha-\beta}\right)$$

Problem:
We need $\mathsf{Enc}(\zeta_{p^i}^\alpha)$ for all i.
One regular bootstrapping only produces $\mathsf{Enc}(\zeta_{p^r}^\alpha)$.
Our Result: This can be done by

$$O(\log q) \cdot (\textbf{EvalAuto} + \textbf{HomMul})$$

Instantiate EqT

The Prime-Power Case

This equality test works for any prime power $q = p^r$:

$$\mathsf{EqT}(\zeta_q^\alpha, \beta) = 1 + \sum_{i=1}^{r} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}}\left(\zeta_{p^i}^{\alpha-\beta}\right)$$

Problem:
We need $\mathsf{Enc}(\zeta_{p^i}^\alpha)$ for all i.
One regular bootstrapping only produces $\mathsf{Enc}(\zeta_{p^r}^\alpha)$.
Our Result: This can be done by

$$O(\log q) \cdot (\textbf{EvalAuto} + \textbf{HomMul})$$

Overall complexity of **EvalFunc**:

$$O(\log q) \cdot (\textbf{EvalAuto} + \textbf{HomMul}) + 1 \cdot \textbf{EvalTr}$$

Instantiate EqT

The Composite Case

Consider a composite $q = \prod_{i=1}^{k} q_i$ where $q_i = p_i^{r_i}$.
By the Chinese Remainder Theorem (CRT),

$$\alpha = \beta \bmod q \quad \Longleftrightarrow \quad \alpha = \beta \bmod q_i \quad \text{for all } i$$

Instantiate EqT

The Composite Case

Consider a composite $q = \prod_{i=1}^{k} q_i$ where $q_i = p_i^{r_i}$.
By the Chinese Remainder Theorem (CRT),

$$\alpha = \beta \bmod q \quad \Longleftrightarrow \quad \alpha = \beta \bmod q_i \quad \text{for all } i$$

Combine with the equality test for prime-power cases,

$$\prod_{i=1}^{k} \left(1 + \sum_{j=1}^{r_i} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p_i^j})/\mathbb{Q}} \left(\zeta_{p_i^j}^{\alpha-\beta} \right) \right) = \begin{cases} q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases}.$$

We can

1. Evaluate each CRT branch
2. **HomMul** the results of all branches

Instantiate EqT

The Composite Case

Consider a composite $q = \prod_{i=1}^{k} q_i$ where $q_i = p_i^{r_i}$.
By the Chinese Remainder Theorem (CRT),

$$\alpha = \beta \bmod q \quad \Longleftrightarrow \quad \alpha = \beta \bmod q_i \quad \text{for all } i$$

Combine with the equality test for prime-power cases,

$$\prod_{i=1}^{k} \left(1 + \sum_{j=1}^{r_i} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p_i^j})/\mathbb{Q}} \left(\zeta_{p_i^j}^{\alpha-\beta} \right) \right) = \begin{cases} q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases}.$$

We can

1. Evaluate each CRT branch
2. **HomMul** the results of all branches

But ...

🙁 Consecutive **HomMul** $\Longrightarrow$ large noise growth

Instantiate EqT

The Composite Case - Exploring More Algebraic Properties of Trace

Key observation:

Suppose $q = q_1 q_2$ where q_1, q_2 are coprime, then

$$\begin{aligned}
\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^i) &= \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_{q_1}^i \cdot \zeta_{q_2}^i) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i \cdot \zeta_{q_2}^i)\right) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}\left(\zeta_{q_2}^i \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i)\right) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i) \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^i) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1})/\mathbb{Q}}(\zeta_{q_1}^i) \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^i)
\end{aligned}$$

Instantiate EqT

The Composite Case - Exploring More Algebraic Properties of Trace

Key observation:

Suppose $q = q_1 q_2$ where q_1, q_2 are coprime, then

$$\begin{aligned}
\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^i) &= \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_{q_1}^i \cdot \zeta_{q_2}^i) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i \cdot \zeta_{q_2}^i)\right) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}\left(\zeta_{q_2}^i \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i)\right) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i) \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^i) \\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1})/\mathbb{Q}}(\zeta_{q_1}^i) \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^i)
\end{aligned}$$

We can **separate/integrate** the traces!

Instantiate EqT

The Composite Case - Exploring More Algebraic Properties of Trace

Suppose $q = \prod_{i=1}^{k} q_i$ where all the q_i's are distinct prime-powers, then

$$\prod_{i=1}^{k} \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_i})/\mathbb{Q}}(\zeta_{q_i}^{j}) = \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1 \cdots q_k})/\mathbb{Q}}(\zeta_{q_1 \cdots q_k}^{j})$$

Product of traces $\longrightarrow$ Product in the cyclotomic index

Instantiate EqT

The Composite Case - Exploring More Algebraic Properties of Trace

Suppose $q = \prod_{i=1}^{k} q_i$ where all the q_i's are distinct prime-powers, then

$$\prod_{i=1}^{k} \mathrm{Tr}_{\mathbb{Q}(\zeta_{q_i})/\mathbb{Q}}(\zeta_{q_i}^{j}) = \mathrm{Tr}_{\mathbb{Q}(\zeta_{q_1 \cdots q_k})/\mathbb{Q}}(\zeta_{q_1 \cdots q_k}^{j})$$

Product of traces $\longrightarrow$ Product in the cyclotomic index

So

$$\prod_{i=1}^{k} \left(1 + \sum_{j=1}^{r_i} \mathrm{Tr}_{\mathbb{Q}\left(\zeta_{p_i^j}\right)/\mathbb{Q}} \left(\zeta_{p_i^j}^{\alpha-\beta}\right)\right) = 1 + \sum_{w|q, w\neq 1} \mathrm{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}} \left(\zeta_w^{\alpha-\beta}\right),$$

since the combinations of p_i^j are exactly the divisors of q except 1.

Product of sums $\longrightarrow$ Pure sums

Instantiate EqT

The Composite Case

Now, our equality test for composite q is

$$1 + \sum_{w|q, w \neq 1} \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta}\right) = \begin{cases} q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases}.$$

Problem:
The traces of sub-extensions are not contained in a single tower down to $\mathbb{Q}$.

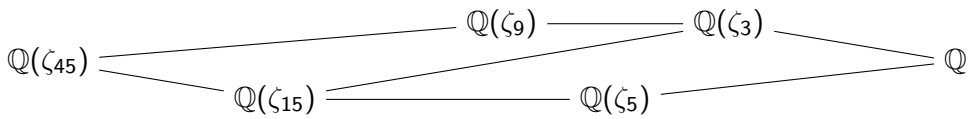
Instantiate EqT

The Composite Case

Now, our equality test for composite q is

$$1 + \sum_{w|q, w\neq 1} \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta}\right) = \begin{cases} q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases}.$$

Problem:
The traces of sub-extensions are not contained in a single tower down to $\mathbb{Q}$.
Example: For $q = 45 = 3^2 \cdot 5$ with proper divisors $w = 3, 5, 9, 15, 45$:



🙁 Evaluate the traces separately could be inefficient (in general).

Instantiate EqT

The Composite Case

Solution: Swap the sum and the trace

$$\mathrm{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(1+\sum_{w\mid q, w\neq 1}\phi(w)\cdot\zeta_w^{\alpha-\beta}\right)=\begin{cases}\phi(q)\cdot q & \text{if } \alpha=\beta \mod q\\ 0 & \text{if } \alpha\neq\beta \mod q\end{cases}.$$

✅ This equality test only requires 1 **EvalTr**.

Instantiate EqT

The Composite Case

Solution: Swap the sum and the trace

$$\mathrm{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(1+\sum_{w\mid q, w\neq 1}\phi(w)\cdot\zeta_w^{\alpha-\beta}\right)=\begin{cases}\phi(q)\cdot q & \text{if } \alpha=\beta \mod q\\ 0 & \text{if } \alpha\neq\beta \mod q\end{cases}.$$

✅ This equality test only requires 1 **EvalTr**.

Challenge:

How to compute $\mathrm{Enc}(\zeta_w^\alpha)$ for all $w \mid q$ given $\mathrm{Enc}(\zeta_q^\alpha)$?

Instantiate EqT

The Composite Case

Solution: Swap the sum and the trace

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(1+\sum_{w\mid q, w\neq 1}\phi(w)\cdot\zeta_w^{\alpha-\beta}\right)=\begin{cases}\phi(q)\cdot q & \text{if } \alpha=\beta \mod q\\ 0 & \text{if } \alpha\neq\beta \mod q\end{cases}.$$

✅ This equality test only requires 1 **EvalTr**.

Challenge:

How to compute $\mathsf{Enc}(\zeta_w^\alpha)$ for all $w \mid q$ given $\mathsf{Enc}(\zeta_q^\alpha)$?

Our Result: This can be done by

$$O(\log q)\cdot(\textbf{EvalAuto}+\textbf{HomMul})$$

if q is odd and has $O(\log q)$ divisors.

Instantiate EqT

The Composite Case

Solution: Swap the sum and the trace

$$\mathrm{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(1+\sum_{w\mid q, w\neq 1}\phi(w)\cdot\zeta_w^{\alpha-\beta}\right)=\begin{cases}\phi(q)\cdot q & \text{if } \alpha=\beta \mod q\\ 0 & \text{if } \alpha\neq\beta \mod q\end{cases}.$$

✅ This equality test only requires 1 **EvalTr**.

Challenge:

How to compute $\mathrm{Enc}(\zeta_w^\alpha)$ for all $w \mid q$ given $\mathrm{Enc}(\zeta_q^\alpha)$?

Our Result: This can be done by

$$O(\log q)\cdot(\textbf{EvalAuto}+\textbf{HomMul})$$

if q is odd and has $O(\log q)$ divisors.

Overall complexity of **EvalFunc**:

$$O(\log q)\cdot(\textbf{EvalAuto}+\textbf{HomMul})+1\cdot\textbf{EvalTr}$$

The Complexity of **EvalTr**

For a Galois extension K/F,

$$\mathrm{Tr}_{K/F}(\mu) = \sum_{\sigma \in \mathrm{Gal}(K/F)} \sigma(\mu)$$

Complexity of EvalTr$_{K/F}$: Trivially $|\mathrm{Gal}(K/F)|$ times **EvalAuto**.

The Complexity of **EvalTr**

For a Galois extension K/F,

$$\mathsf{Tr}_{K/F}(\mu) = \sum_{\sigma \in \mathsf{Gal}(K/F)} \sigma(\mu)$$

Complexity of $\mathbf{EvalTr}_{K/F}$: Trivially $|\mathsf{Gal}(K/F)|$ times **EvalAuto**.

Efficient Case 1: Tower of Field Extensions [Alperin-Sheriff-Peikert'13, Chen-Dai-Kim-Song'21]

$$K = K_r/K_{r-1}/\cdots/K_1/K_0 = F.$$

Complexity of $\mathbf{EvalTr}_{K/F}$: $\sum_{i=1}^{r} |\mathsf{Gal}(K_i/K_{i-1})|$ times **EvalAuto**.

The Complexity of **EvalTr**

For a Galois extension K/F,

$$\mathsf{Tr}_{K/F}(\mu) = \sum_{\sigma \in \mathsf{Gal}(K/F)} \sigma(\mu)$$

Complexity of $\mathbf{EvalTr}_{K/F}$: Trivially $|\mathsf{Gal}(K/F)|$ times **EvalAuto**.

Efficient Case 1: Tower of Field Extensions [Alperin-Sheriff-Peikert'13, Chen-Dai-Kim-Song'21]

$$K = K_r/K_{r-1}/\cdots/K_1/K_0 = F.$$

Complexity of $\mathbf{EvalTr}_{K/F}$: $\sum_{i=1}^{r} |\mathsf{Gal}(K_i/K_{i-1})|$ times **EvalAuto**.

Efficient Case 2: Cyclic Extensions [Halevi-Shoup'14, Zheng-Li-Wang'23]

$$\mathsf{Gal}(K/F) = \left\{ \mathsf{id} := \tau_g^0, \tau_g, \tau_g^2, \ldots, \tau_g^{|\mathsf{Gal}(K/F)|-1} \right\}.$$

Complexity of $\mathbf{EvalTr}_{K/F}$: $O(\log |\mathsf{Gal}(K/F)|)$ times **EvalAuto**.

The Complexity of **EvalTr**

For a Galois extension K/F,

$$\mathsf{Tr}_{K/F}(\mu) = \sum_{\sigma \in \mathsf{Gal}(K/F)} \sigma(\mu)$$

Complexity of EvalTr$_{K/F}$: Trivially $|\mathsf{Gal}(K/F)|$ times **EvalAuto**.

Efficient Case 1: Tower of Field Extensions [Alperin-Sheriff-Peikert'13, Chen-Dai-Kim-Song'21]

$$K = K_r/K_{r-1}/\cdots/K_1/K_0 = F.$$

Complexity of EvalTr$_{K/F}$: $\sum_{i=1}^{r} |\mathsf{Gal}(K_i/K_{i-1})|$ times **EvalAuto**.

Efficient Case 2: Cyclic Extensions [Halevi-Shoup'14, Zheng-Li-Wang'23]

$$\mathsf{Gal}(K/F) = \left\{ \mathsf{id} := \tau_g^0, \tau_g, \tau_g^2, \ldots, \tau_g^{|\mathsf{Gal}(K/F)|-1} \right\}.$$

Complexity of EvalTr$_{K/F}$: $O(\log |\mathsf{Gal}(K/F)|)$ times **EvalAuto**.

Fact: If t is an odd prime-power $\Longrightarrow$ The cyclotomic extension $F(\zeta_t)/F$ is cyclic

The Complexity of **EvalTr**

Key Observation: All cyclotomic extensions are combinations of the above two cases. Consider $\mathbb{Q}(\zeta_q)/\mathbb{Q}$ where $q = \prod_{i=1}^{k} q_i$.

The Complexity of **EvalTr**

Key Observation: All cyclotomic extensions are combinations of the above two cases. Consider $\mathbb{Q}(\zeta_q)/\mathbb{Q}$ where $q = \prod_{i=1}^{k} q_i$.

Case 1 All q_i are powers of odd primes:

$$\mathbb{Q}(\zeta_q) \overset{\text{all cyclic}}{\text{- - - - - - - - - - - -}} \mathbb{Q}(\zeta_{q_1 q_2 q_3}) \overset{\text{cyclic}}{\text{———}} \mathbb{Q}(\zeta_{q_1 q_2}) \overset{\text{cyclic}}{\text{———}} \mathbb{Q}(\zeta_{q_1}) \overset{\text{cyclic}}{\text{———}} \mathbb{Q}$$

The Complexity of **EvalTr**

Key Observation: All cyclotomic extensions are combinations of the above two cases. Consider $\mathbb{Q}(\zeta_q)/\mathbb{Q}$ where $q = \prod_{i=1}^{k} q_i$.

Case **1** All q_i are powers of odd primes:

$$\mathbb{Q}(\zeta_q) \text{ - - - -}\overset{\text{all cyclic}}{\text{- - - - - - - -}}\text{ - - - - } \mathbb{Q}(\zeta_{q_1 q_2 q_3}) \xrightarrow{\text{cyclic}} \mathbb{Q}(\zeta_{q_1 q_2}) \xrightarrow{\text{cyclic}} \mathbb{Q}(\zeta_{q_1}) \xrightarrow{\text{cyclic}} \mathbb{Q}$$

Case **2** Some $q_i = 2^r$. Suppose without loss of generality that $q_1 = 2^r$, then

$$\mathbb{Q}(\zeta_q) \text{ - - - -}\overset{\text{all cyclic}}{\text{- - - - - - - -}}\text{ - - - - } \mathbb{Q}(\zeta_{q_1 q_2}) \xrightarrow{\text{cyclic}} \mathbb{Q}(\zeta_{q_1}) \xrightarrow{\text{a base-2 logarithmic tower}} \mathbb{Q}$$

The Complexity of **EvalTr**

Key Observation: All cyclotomic extensions are combinations of the above two cases. Consider $\mathbb{Q}(\zeta_q)/\mathbb{Q}$ where $q = \prod_{i=1}^{k} q_i$.

Case 1 All q_i are powers of odd primes:

$$\mathbb{Q}(\zeta_q) \text{ - - - -}\ \overset{\text{all cyclic}}{\text{- - - - - - - -}}\ \mathbb{Q}(\zeta_{q_1 q_2 q_3}) \xrightarrow{\text{cyclic}} \mathbb{Q}(\zeta_{q_1 q_2}) \xrightarrow{\text{cyclic}} \mathbb{Q}(\zeta_{q_1}) \xrightarrow{\text{cyclic}} \mathbb{Q}$$

Case 2 Some $q_i = 2^r$. Suppose without loss of generality that $q_1 = 2^r$, then

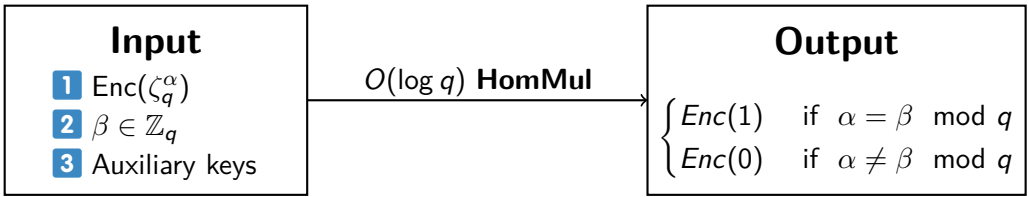
$$\mathbb{Q}(\zeta_q) \text{ - - - -}\ \overset{\text{all cyclic}}{\text{- - - - - - - -}}\ \mathbb{Q}(\zeta_{q_1 q_2}) \xrightarrow{\text{cyclic}} \mathbb{Q}(\zeta_{q_1}) \xrightarrow{\text{a base-2 logarithmic tower}} \mathbb{Q}$$

Trace in both cases requires $O(\log q)$ automorphisms.

✅ **EvalTr**$_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}$ requires $O(\log q)$ times **EvalAuto** for any q.

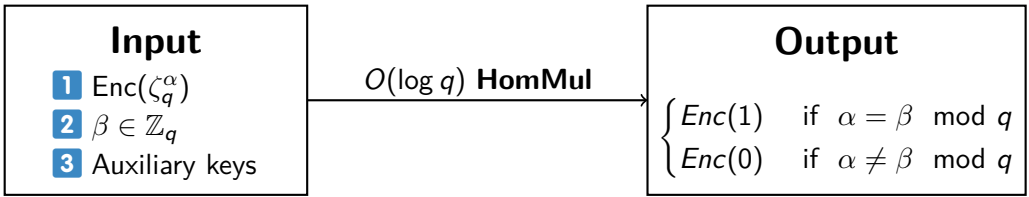
Messages to General Audiences

A New Homomorphic Equality Test

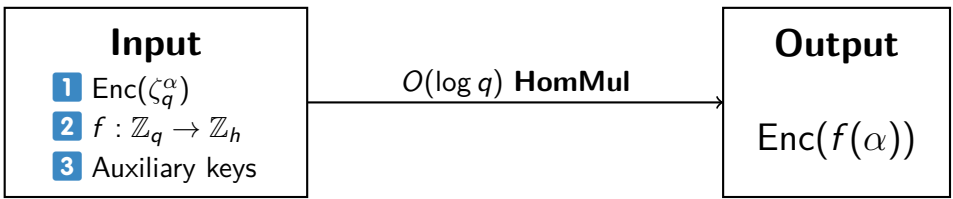


Messages to General Audiences

A New Homomorphic Equality Test



A New Arbitrary Function Evaluation



Thank You!