Carousel

Blind Rotation Over the Automorphism Group

Intak Hwang, Seonhong Min, Yongsoo Song
Seoul National University

서 울 대 학 교
SEOUL NATIONAL UNIVERSITY

Summary

✓ **Blind Rotation over the automorphism group**

Functional Bootstrapping method through blind rotation over the automorphism group, instead of the multiplicative group of monomials.

✓ **Carousel**

New fully homomorphic encryption scheme from the blind rotation over the automorphism group.

✓ The bootstrapping time is 19ms and 46ms for 4-bit integer in coefficient mode and slot mode.

AP-like cryptosystems

✓ **Blind Rotation**

Homomorphic evaluation of a look-up table (LUT) of size M for the input LWE ciphertext $(b, \vec{a}) \in \mathbb{Z}_M^{(n+1)}$ using the multiplicative group of monomials $\{1, X, X^2, \dots, X^{M-1}\}$ over the ring $\mathbb{Z}[X]/\Phi_M(X)$. (i.e., compute $tv \cdot X^{b+<\vec{a},\vec{s}>}$ for the polynomial tv with pre-assigned coefficients.)

✓ **Problem**

- Not arithmetic friendly.
- Only can be instantiated over the cyclotomic rings.
- Does not support finite field arithmetic.

Our Solution

✓ **Automorphism Group**

Unlike the multiplicative group of monomials, the automorphism group forms a nice structure. We only consider the case in which the automorphism group is cyclic.

✓ **Blind rotation over automorphism group**

- For the generator $\Psi: X \mapsto X^g$ of the automorphism group $\{id, \Psi, \Psi^2, \dots, \Psi^{N-1}\}$ of the base ring, we compute $\Psi^{b+<\vec{a},\vec{s}>}(tv)$ for tv, a polynomial encoding of the LUT.
- Given that the key $\vec{s}$ is a binary vector,
$$\Psi^{a_i s_i}(tv) = tv + (\Psi^{a_i}(tv) - tv) \cdot s_i$$
for all $1 \le i \le n$.
- Using this relation, iteratively 'rotate' the input test vector $\Psi^b(tv)$ by $<\vec{a}, \vec{s}>$ to obtain $\Psi^{b+<\vec{a},\vec{s}>}(tv)$.

Programming test vector

✓ **Slot mode**

- In SIMD FHE schemes, we can encode the message vector into a polynomial by interpolating at the root of unities.
- The message vector rotates with automorphisms.
- Therefore, LUT can be directly encoded into the slots.
- At the end of the computation, extract the first slot.

- +) Arithmetic-friendly (addition, multiplication⋯)
  +) Finite field arithmetic is supported.
  -) Large noise growth from slot extraction

✓ **Coefficient mode**

- If the base ring is a subring of some ring of integer of prime cyclotomic degree, we can directly set the coefficient vector as the LUT (with the right order and basis).
- It is not so straightforward in other rings⋯

- +) Small noise growth
  -) Multiplicative operation is difficult.

Implementation

✓ **Setting**

- Cyclotomic degree $M = 65537 = 2^{16} + 1$.
- Plaintext modulus : $p^r = 2^r$.
- Base ring is a subring of $\mathbb{Z}[X]/\Phi_M(X)$ invariant to the automorphism $X \mapsto X^p$.
- Ring degree $N = \frac{M-1}{ord(p,M)} = 2048$
- By setting so, we can obtain the full packing density for integer vector.

✓ **Experiments**

- Julia : https://github.com/SNUCP/carousel
  Go : https://github.com/sp301415/carousel
- Machine : 11th Gen Intel(R) Core(TM) i9-11900 @ 2.50GHz 32GB RAM

	UInt2	UInt3	UInt4
Carousel (Coeff mode)	19ms	19ms	19ms
Carousel (Slot mode)	28ms	36ms	46ms

Latency of a single functional bootstrapping (Go)