

Toward a generic geometric API for FHE

N. Gama and M. Georgieva

<https://tfhe.github.io>



Model of computations

- Binary, circuit computations
- Integer arithmetic computation

decimal

$$\begin{array}{r} 0011 \text{ --- carries} \\ 4567 \\ 366 \\ \hline 4933 \end{array}$$
- Approximated (fixed-point) computation

Some HE libraries and their strength

- BGV (Helib): massively parallel, finite field arithmetic
- BFV (Seal, Palisade): massively parallel, small depth polynomials
- CKKS (HEAAN): massively parallel, floating point arithmetic
- TFHE, FHEW: single eval, boolean logic, comparison, threshold

How we can represent all plaintexts over the $\mathbb{T}_N[X]$

Ciphertext $(a, b) \rightarrow \mathbb{T}_N[X] + \text{noise?}$

- ← Circuits $\mathcal{B} = (0, 1)$
- ← Integers $(\mathbb{Z}/p\mathbb{Z})^n$
- ← Fixed point \mathbb{C}

Homomorphic operations hierarchy

TRLWE

TRGSW

- small integer linear combinations $x + y, x - y$
 $a.x$ for public $a \in \mathbb{Z}_N[X]$
- External product $a.x$ for secret a
- polynomials in s (internal products)
 - Sublattice (modular ring)
 - BFV API
 - slots mod p
 - slots add
 - slots mult
 - slots rotate
 - Small Ball (real ring)
 - CKKS API
 - fixed point slots
 - slots add
 - slots mult
 - slots rotate
- $a \in \{0, 1\}$
 - cmux (selector)
 - blindrotate ($\times X^{\text{secret } \nu}$)
 - (automata)
 - TFHE Gates API
 - individual bits
 - nand, and, or, xor, ...
 - mux

Circuits over $\mathbb{T}_N[X]$ (AND)

$0 = (0,1) / (1,0)$

$\frac{3}{4} = (1,1)$

$\frac{1}{4} = (0,0)$

Integers and fixed-point over $\mathbb{T}_N[X]$

BFV: slots mod p , slots add, slots mult, slots rotate

CKKS: fixed point slots, slots add, slots mult, slots rotate

Linear combination with secret coefficients $(0, 1)$

TRLWE

$\sum s_i \cdot a_i$

TRLWE

Linear combination with secret coefficient s

Internal product requires to evaluate a polynomial in $s: (b_1 - sa_1)(b_2 - sa_2)$

HowTo: evaluate a polynomial in s ?

- dedicated relinearization/keyswitch techniques (2011, ...)
- But in fact, TRGSW provides multiplication by s !

$$C_1 \boxtimes_{P,\alpha} C_2 = (C_1, C_0) - TRGSW(s) \boxtimes_{\alpha} (C_2, 0)$$