

Homomorphic Encryption Standard: TFHE API

Mariya Georgieva



August 17, 2019

TFHE: Fast Fully Homomorphic Encryption over the Torus

- TFHE API draft shared to the mailing list: (11 July 2019)
(S. Carpov, I. Chillotti, N. Gama, M. Georgieva)
<https://docs.google.com/document/d/1aUGdI1BijYebos8gN02cS3HFCR90JVLwAWzBctUN3ms/edit#>
 - Parameters
 - Data Encoding and Ciphertexts
 - Secret-Key Encryption
 - Public-Key Encryption
 - Leveled Homomorphic Encryption Operations
 - Bootstrapped Homomorphic Encryption Operations
- TFHE open source library:
<https://tfhe.github.io>

- Security parameter λ
- Noise rate α – (*auto-deduced in bootstrapped mode*)
- Ring dimension n – (*auto-deduced in bootstrapped and leveled mode*)

In bootstrapped mode

The values of α and n are derived from the security parameter λ by the library in order to enable full bootstrapping cycle.

In leveled mode

α serves as a measure for the number of homomorphic operations that can be run on a ciphertext before saturating the noise. The ring dimension n is then determined by the security level λ .

- TLWE ciphertexts encrypt plaintext in \mathbb{T}
 - $\mathbb{T} = \mathbb{R} \bmod 1$:
(Torus arithmetic, as a \mathbb{Z} - module)
 - $3 \cdot 0.6 = 0.8 \bmod 1$
 - external product by integers

Polynomial version

- TRLWE ciphertexts encrypt plaintext in $\mathbb{T}_N[X]$
- TRGSW ciphertexts encrypt plaintext in $\mathbb{Z}_N[X]$
 - $\mathbb{T}_N[X] = \mathbb{R}[X] \bmod X^N + 1 \bmod 1$:
(Torus polynomial arithmetic, as a $\mathbb{Z}_N[X]$ - module)
 - $(2X + 3) \cdot (0.4X + 0.5) = (0.2X + 0.7) \bmod X^2 + 1 \bmod 1$
 - external product by integers polynomial

TRLWE

small integer linear combinations

 $x + y, x - y$ $a.x$ for public $a \in \mathbb{Z}_N[X]$

TRLWE



TRGSW

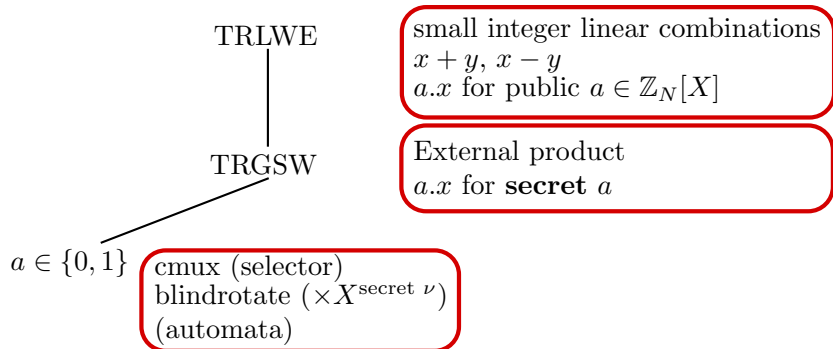
small integer linear combinations

$x + y, x - y$

$a.x$ for public $a \in \mathbb{Z}_N[X]$

External product

$a.x$ for **secret** a



TRLWE

small integer linear combinations

 $x + y, x - y$ $a.x$ for public $a \in \mathbb{Z}_N[X]$

TRGSW

External product

 $a.x$ for **secret** a $a \in \{0, 1\}$

cmux (selector)

blindrotate ($\times X^{\text{secret } \nu}$)

(automata)

TFHE Gates API

individual bits

nand, and,

or, xor, ...

mux

Leveled Homomorphic Encryption Operations

The basic operations are:

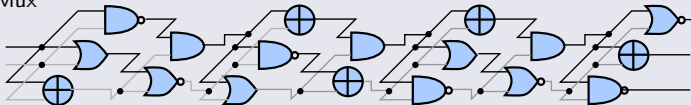
- TLWE, TRLWE, TRGSW linear combinations
- TRGSW-TRLWE external product

Some useful derived operation:

- TRGSW-TRLWE-TRLWE private/oblivious selector (CMux)
- BlindRotate

Bootstrapped Homomorphic Encryption Operations

- Constant gates: Zero, One
- Unary gate: Not
- Binary gates: And, Or, Xor, Xnor, AndNot, OrNot, Nor, Nand.
- Ternary gate: Mux



Internal product requires to evaluate a polynomial in s :

$$(b_1 - sa_1)(b_2 - sa_2) = b_1b_2 - (b_1a_2 + b_2a_1)s + a_1a_2s^2.$$

The term s^2 :

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

The meaning of a_1a_2 :

- sublattices: a_i are exact multiples of $\frac{1}{p}$ for a fixed small p
- small ball: a_i is bounded

TRLWE

small integer linear combinations

 $x + y, x - y$ $a.x$ for public $a \in \mathbb{Z}_N[X]$

TRGSW

External product

 $a.x$ for **secret** a $a \in \{0, 1\}$

cmux (selector)

blindrotate ($\times X^{\text{secret } \nu}$)

(automata)

TFHE Gates API

individual bits

nand, and,

or, xor, ...

mux

TRLWE

small integer linear combinations

 $x + y, x - y$ $a.x$ for public $a \in \mathbb{Z}_N[X]$

TRGSW

External product

 $a.x$ for **secret** a $a \in \{0, 1\}$

cmux (selector)

blindrotate ($\times X^{\text{secret } \nu}$)

(automata)

 $a = s$ polynomials in s

(internal products)

TFHE Gates API

individual bits

nand, and,

or, xor, ...

mux

TRLWE

small integer linear combinations

 $x + y, x - y$ $a.x$ for public $a \in \mathbb{Z}_N[X]$

TRGSW

External product

 $a.x$ for **secret** a $a \in \{0, 1\}$

cmux (selector)

blindrotate ($\times X^{\text{secret } \nu}$)

(automata)

 $a = s$ polynomials in s

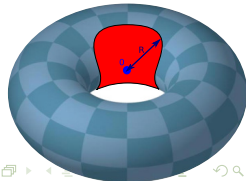
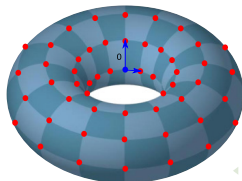
(internal products)

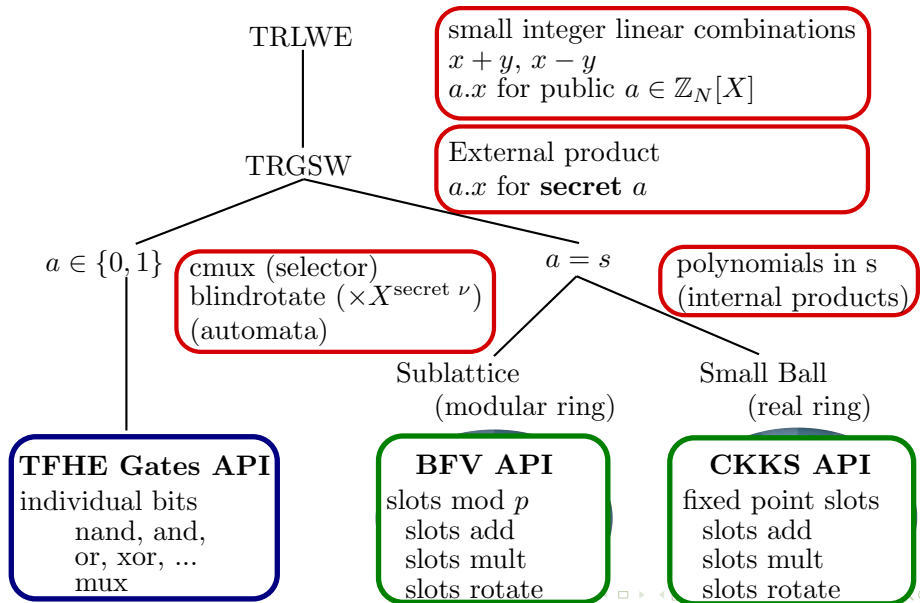
Sublattice

(modular ring)

Small Ball

(real ring)

TFHE Gates APIindividual bits
nand, and,
or, xor, ...
mux



Join the poster session to discuss about a generic API!!!

Toward a generic geometric API for FHE
 N. Gama and M. Georgieva <https://tthe.github.io>

<p>Model of computations</p> <ol style="list-style-type: none"> Binary circuit computations Integer arithmetic computation $\begin{matrix} 1111 & + & 1111 \\ 1111 & + & 1111 \\ 1111 & + & 1111 \\ \hline 1111 & + & 1111 \end{matrix}$ Approximated (fixed-point) computation 	<p>Some HE libraries and their strength</p> <ul style="list-style-type: none"> BCV (HElib): massively parallel, finite field arithmetic S/FV (SEAL, PALISADE): massively parallel, small depth polynomials CKKS (HEAAN): massively parallel, floating point arithmetic TFHE, FHEW: single eval, boolean logic, comparison, threshold <p>How we can represent all plaintexts over the $\mathbb{T}_N[X]$</p> <div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center;"> <p>Ciphertext (a, b)</p> </div> <div style="margin: 0 10px;"> \rightarrow </div> <div style="border: 1px solid black; border-radius: 50%; padding: 10px; text-align: center;"> $\mathbb{T}_N[X]$ + noise? </div> <div style="margin-left: 10px;"> <p>←</p> </div> <div style="text-align: center;"> <p>Circuits $\mathbb{F} = \{0, 1\}$</p> <p>Integers $\mathbb{Z}_q/2^k$</p> <p>Fixed point \mathbb{C}</p> </div> </div>
<p>Homomorphic operations hierarchy</p> <div style="text-align: center;"> <p>TFHEW</p> <p>↓</p> <p>TKGSW</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>$a \in \{0, 1\}$</p> <p>linear (boolean) homomorphism $(a, b) \mapsto a+b$ (boolean ring)</p> <p>TFHE Geometric API</p> <p>data: add, mul, dot, inner prod, ...</p> </div> <div style="text-align: center;"> <p>$a \in \mathbb{F}$</p> <p>addition (modular ring)</p> <p>BFV API</p> <p>data: add, mul, inner prod, ...</p> </div> <div style="text-align: center;"> <p>$a \in \mathbb{C}$</p> <p>addition in \mathbb{C} (real ring)</p> <p>CKKS API</p> <p>data: add, mul, inner prod, ...</p> </div> </div> <p><small>small integer linear combinations $a = \sum_{i=1}^n \alpha_i x_i$ for public $a \in \mathbb{Z}_q[X]$</small></p> <p><small>External product or for secret a</small></p> </div>	
<p>Circuits over the $\mathbb{T}_2[X]$ (AND)</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>0</p> </div> <div style="text-align: center;"> <p>1</p> </div> </div>	<p>Integers and fixed-point over the $\mathbb{T}_2[X]$</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>BFV</p> </div> <div style="text-align: center;"> <p>CKKS</p> </div> </div>
<p>Linear combination with secret coefficients (0, 1)</p>	<p>Linear combination with secret coefficient a</p> <p>External product requires $(a - aa) \times (a - aa)$ to evaluate a poly. in a.</p> <p>How to evaluate a polynomial in a?</p> <ul style="list-style-type: none"> dedicated externalization/keyswitch techniques (2011, ...) But in fact, TKGSW provides it! <p>$(C_1, \dots, C_n) = (C_1, C_2) + \text{TKGSW}(a) \otimes (C_3, C_4)$</p>