TFHE API Introduction

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/1aUGdIlBijYebos8gN02cS3HFCR90JVLwAwzBctUN3ms/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
TFHE parameters

- Security parameter $\lambda$
- Noise rate $\alpha$ – *(auto-deduced in bootstrapped mode)*
- Ring dimension $n$ – *(auto-deduced in bootstrapped and leveled mode)*

**In bootstrapped mode**

The values of $\alpha$ and $n$ are derived from the security parameter $\lambda$ by the library in order to enable full bootstrapping cycle.

**In leveled mode**

$\alpha$ 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 $\lambda$. 
Data Encoding and Ciphertexts

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

**Polynomial version**

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

TRLWE

small integer linear combinations
\(x + y, x - y\)
\(a.x\) for public \(a \in \mathbb{Z}_N[X]\)
Homomorphic operations hierarchy in TFHE

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 \( a \)
Homomorphic operations hierarchy in TFHE

- **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)
Homomorphic operations hierarchy in TFHE

- **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 $a$

- **TFHE Gates API**
  - Individual bits
  - nand, and, or, xor, ...
  - mux

- **cmux (selector)**
- **blindrotate** ($\times X^{\text{secret } \nu}$)
- **(automata)**
TFHE API Operations

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
General internal product in BFV and CKKS

Internal product requires to evaluate a polynomial in $s$:

$$(b_1 - sa_1)(b_2 - sa_2) = b_1 b_2 - (b_1 a_2 + b_2 a_1)s + a_1 a_2 s^2.$$ 

The term $s^2$:
- dedicated relinearization/keyswitch techniques (2011, ...)
- but in fact, TRGSW provides the multiplication by $s$!

The meaning of $a_1 a_2$:
- sublattices: $a_i$ are exact multiples of $\frac{1}{p}$ for a fixed small $p$
- small ball: $a_i$ is bounded
Homomorphic operations hierarchy

- **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 $a$

**TFHE Gates API**
- individual bits
  - nand, and, or, xor, ...
- mux

- $a \in \{0, 1\}$
- cmux (selector)
- blindrotate ($\times X$secret $\nu$)
- (automata)
Homomorphic operations hierarchy

- **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

- $a = s$
  - polynomials in $s$
  - (internal products)
Homomorphic operations hierarchy

TRLWE
- small integer linear combinations $x + y, x - y$
- $a \cdot x$ for public $a \in \mathbb{Z}_N[X]

TRGSW
- External product
  - $a \cdot x$ 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

Sublattice (modular ring)

Small Ball (real ring)

polynomials in $s$
(internal products)
Homomorphic operations hierarchy

- **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 $a$

- **TFHE Gates API**
  - individual bits
  - nand, and, or, xor, ...
  - mux

- **Sublattice (modular ring)**
  - cmux (selector)
  - blindrotate (× $X^{\text{secret } \nu}$) (automata)

- **Small Ball (real ring)**
  - $a = s$
  - polynomials in $s$ (internal products)

- **BFV API**
  - slots mod $p$
  - slots add
  - slots mult
  - slots rotate

- **CKKS API**
  - fixed point slots
  - slots add
  - slots mult
  - slots rotate
Questions?

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