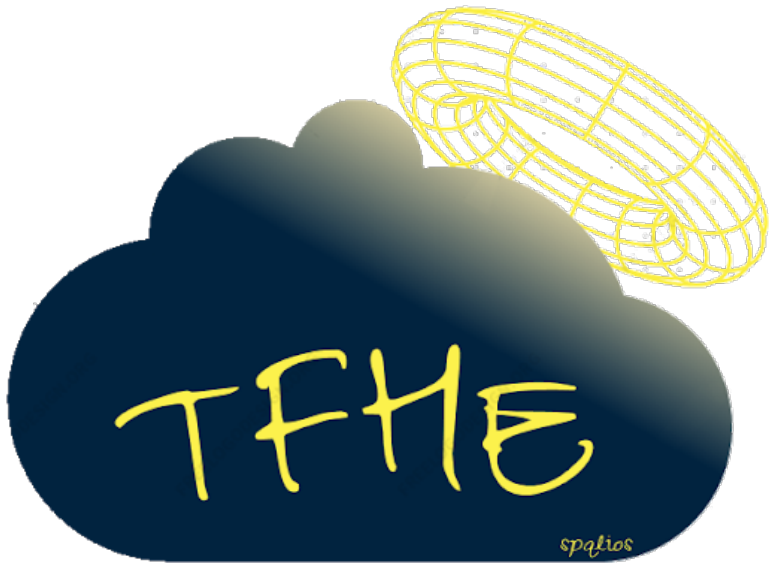
Homomorphic Encryption Standard: TFHE API

Mariya Georgieva



⊕inpher

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/1aUGdIlBijYebos8gNO2cS3HFCR90JVLwAWzBctUN3ms/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 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.

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

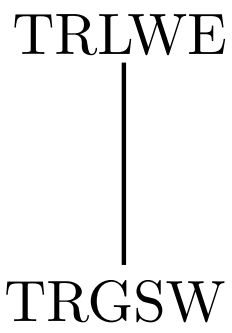
Polynomial version

- 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 polynomail

TRLWE

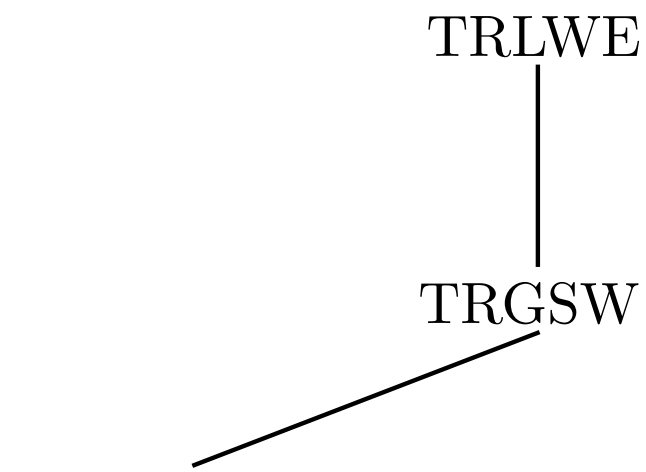
small integer linear combinations
$x + y$, $x - y$
$a.x$ for public $a \in \mathbb{Z}_N[X]$

Homomorphic operations hierarchy in TFHE



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

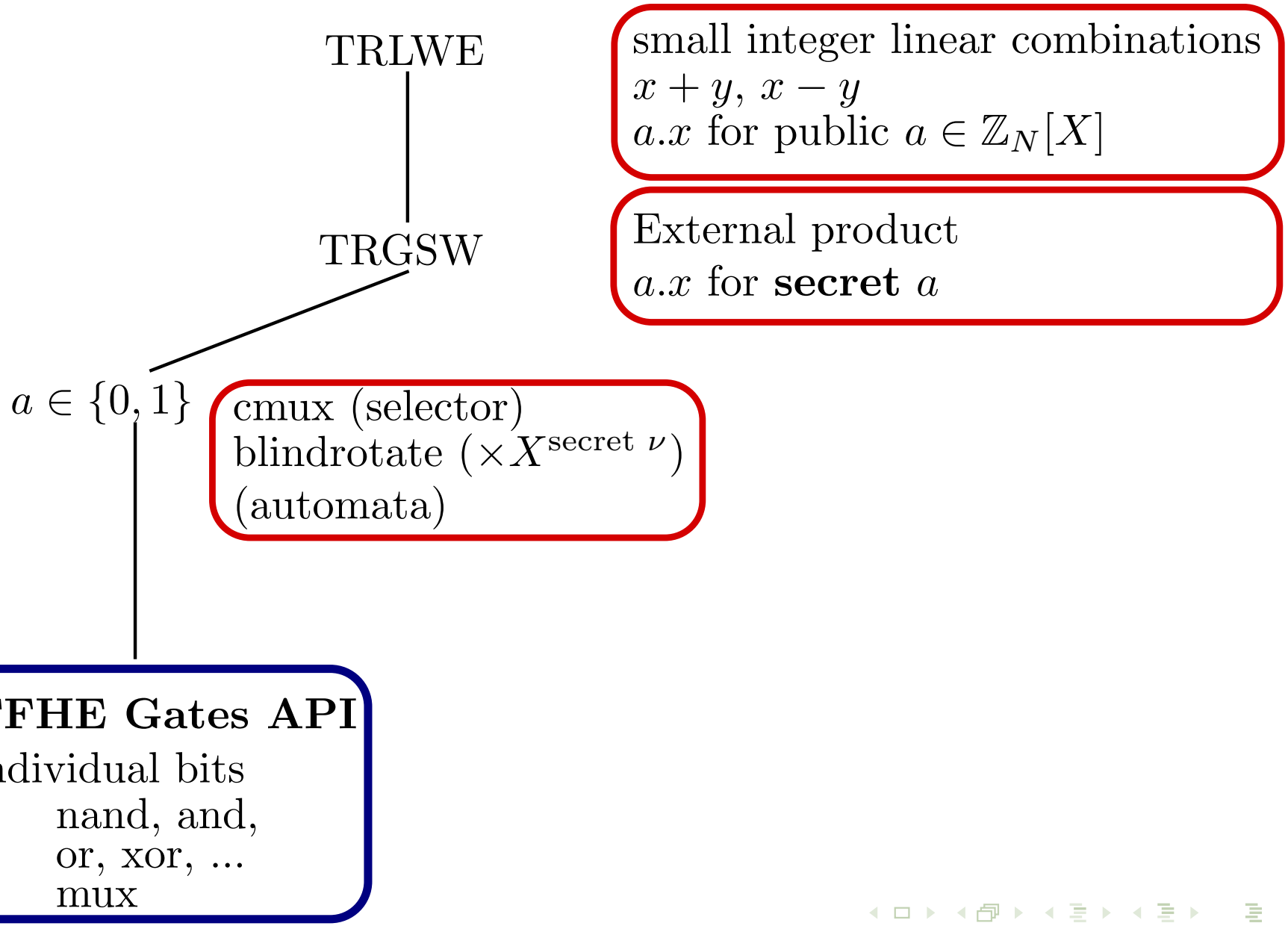
External product
$a.x$ for **secret** a



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

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


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

TFHE API Operations

Leveled Homomorphic Encryption Operations

The basic operations are:

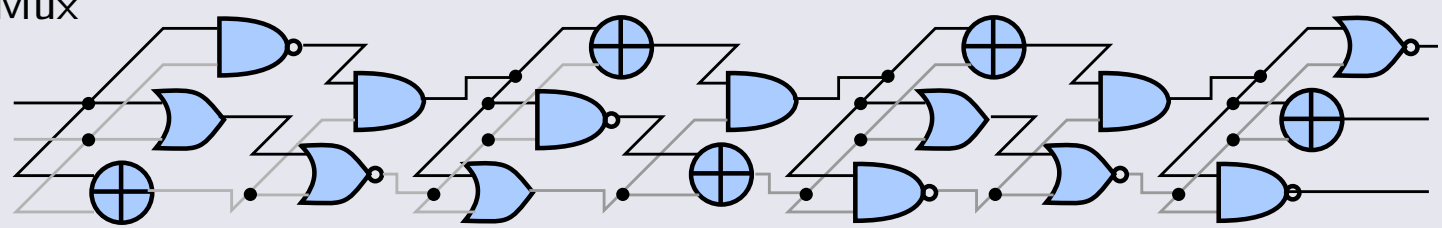
- TLWE, TRLWE, TRGSW linear combinations
- TRGSW-TRLWE external product

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 polynomail in s:

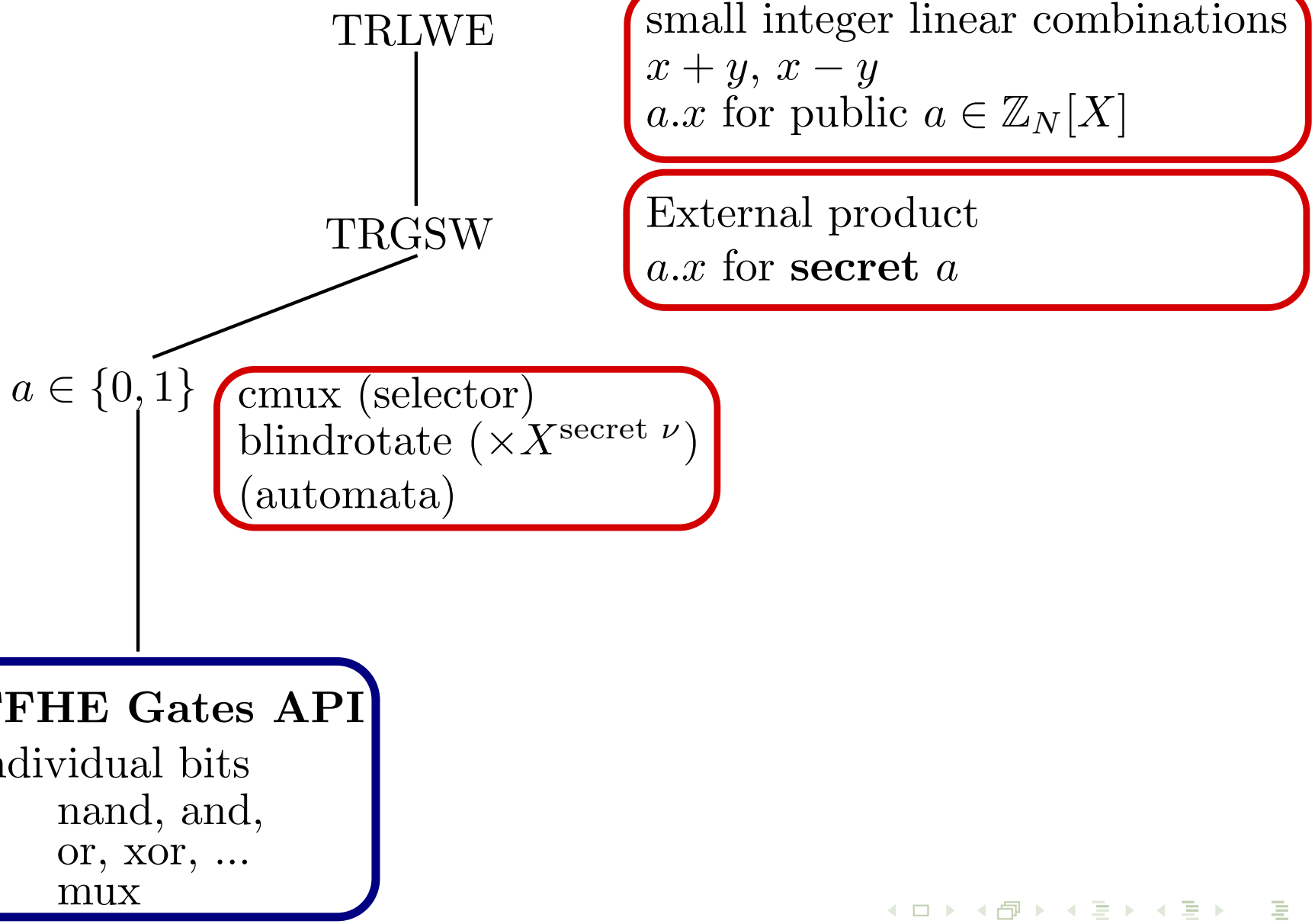
$$(b_1 - sa_1)(b_2 - sa_2) = b_1b_2 - (b_1a_2 + b_2a_1)s + a_1a_2s^2.$$

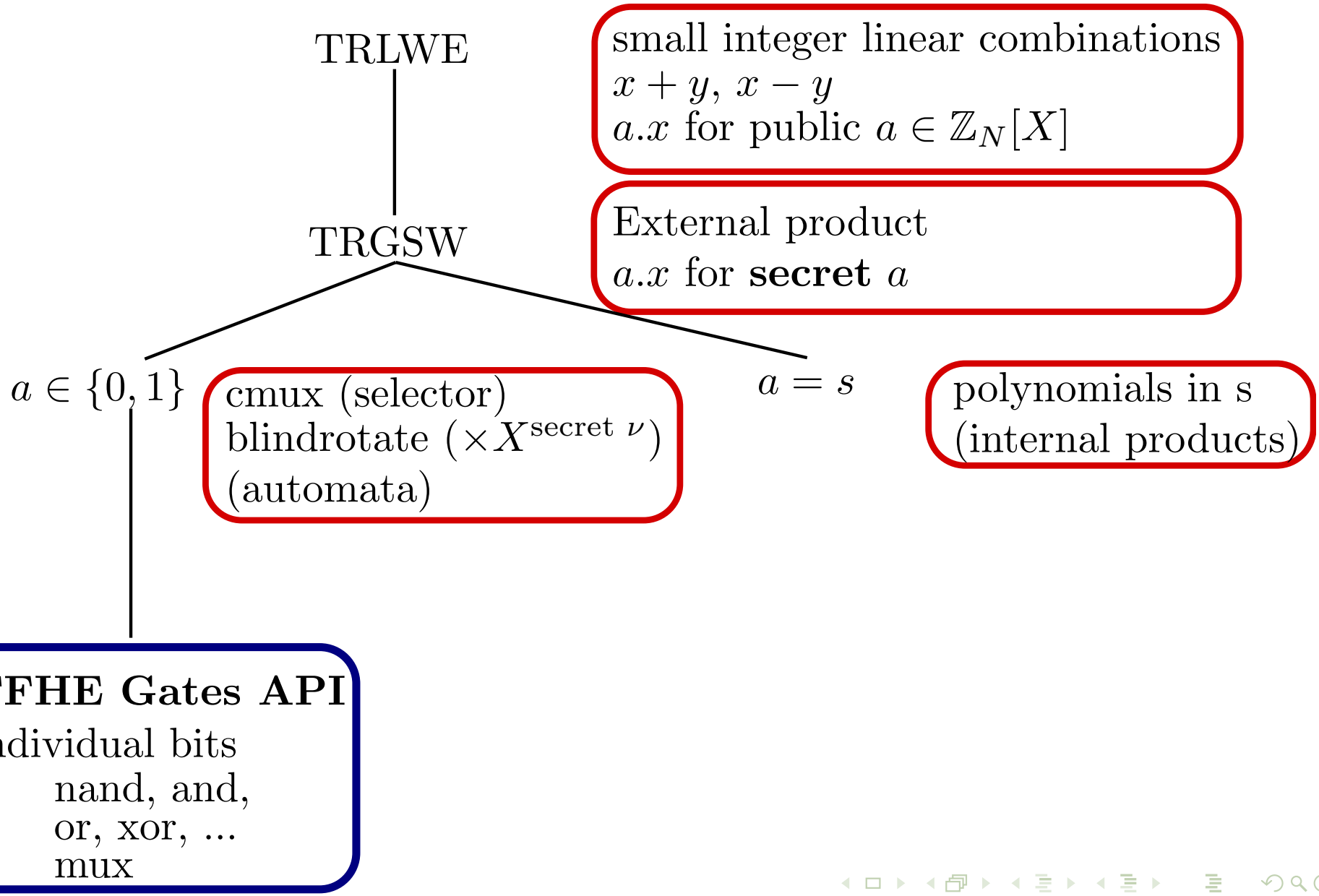
The term s^2:

- 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

$a \in \{0, 1\}$

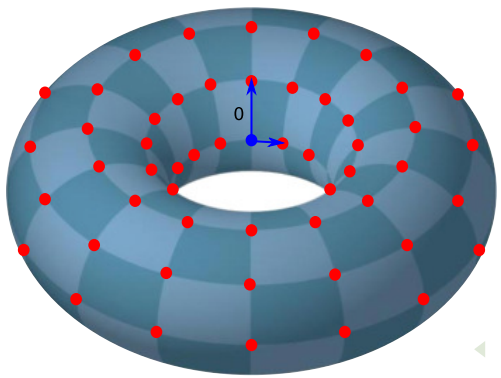
cmux (selector)
blindrotate ($\times X^{\text{secret } \nu}$)
(automata)

$a = s$

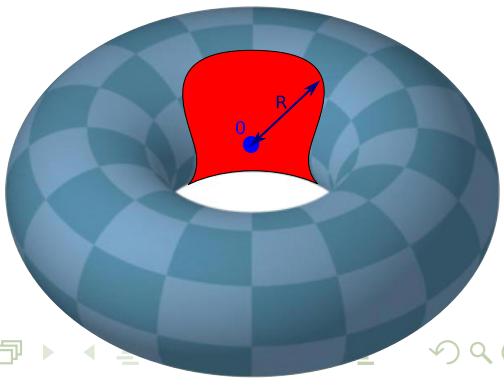
polynomials in s
(internal products)

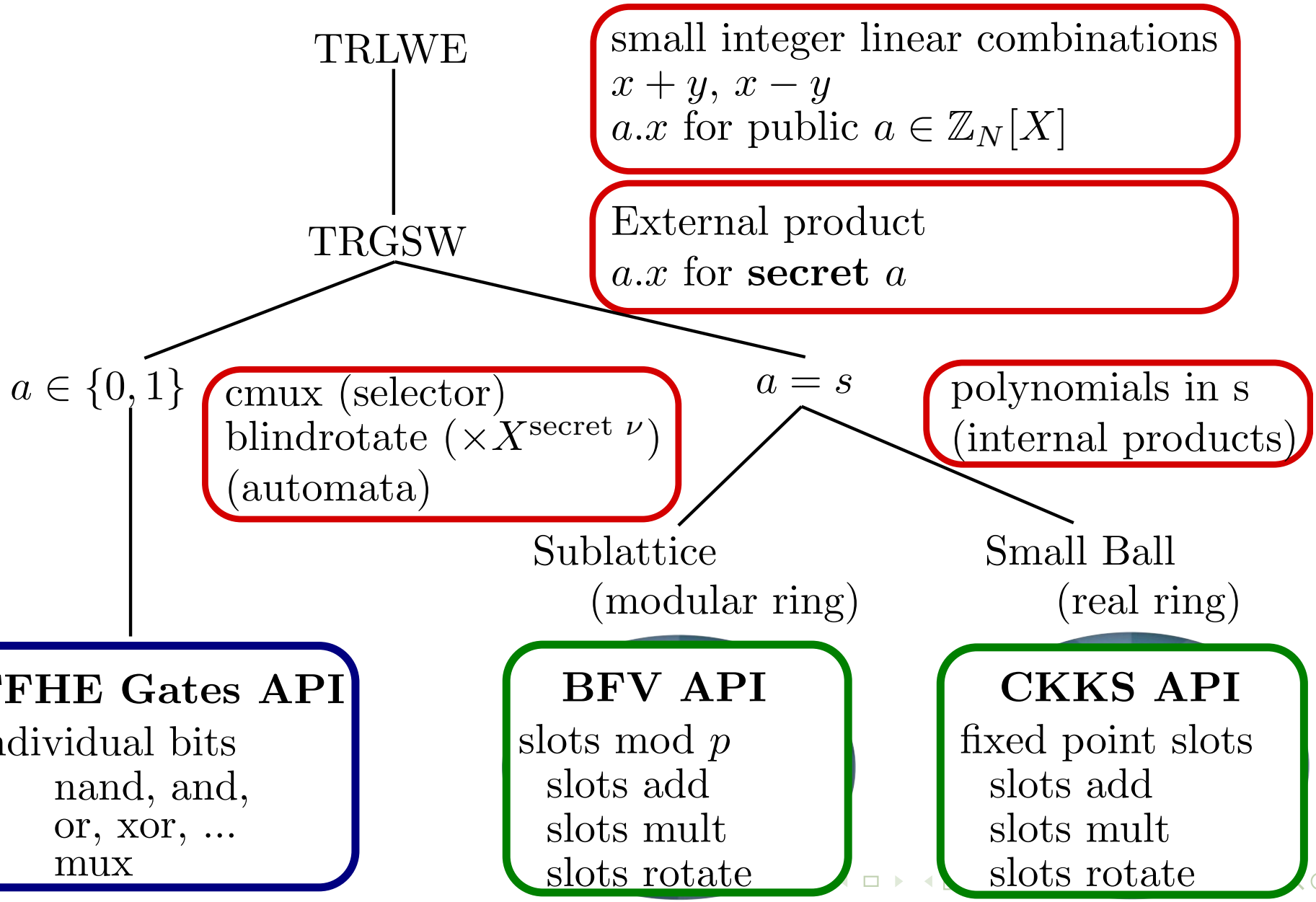
TFHE Gates API
individual bits
nand, and,
or, xor, ...
mux

Sublattice
(modular ring)



Small Ball
(real ring)




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 API
individual bits
nand, and,
or, xor, ...
mux
BFV API
slots mod p
slots add
slots mult
slots rotate
CKKS API
fixed point slots
slots add
slots mult
slots rotate

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