ZAMA

INTRODUCTION TO FHE AND APPLICATIONS TO ML

Seminario di Logica e Informatica Teorica
Università Roma Tre
December 18th, 2020

Ilaria Chillotti

TABLE OF CONTENTS

TABLE OF CONTENTS

HOMOMORPHIC ENCRYPTION

Allows to perform computations on encrypted messages, without decrypting.

m_1

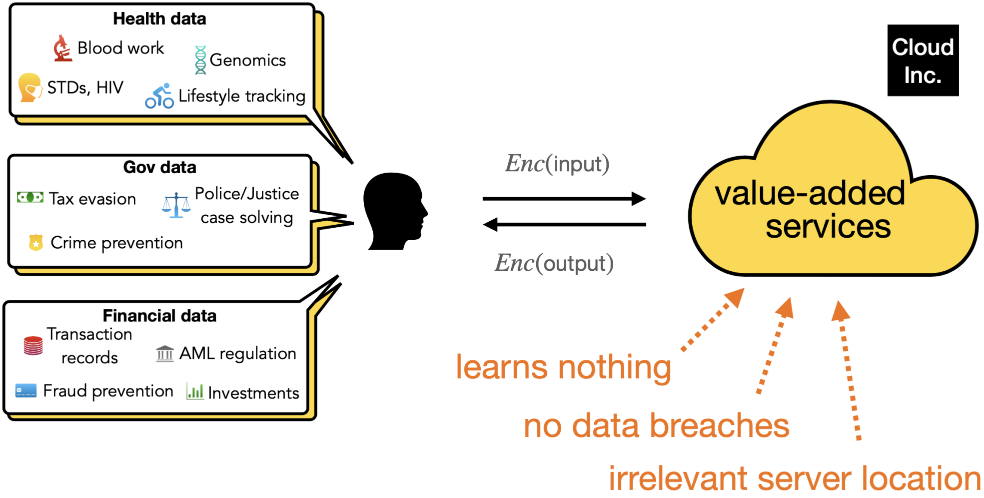
m_2

$$m_1 +_{hom} m_2 = m_1 + m2$$

$$m_1 \times_{hom} m_2 = m_1 \times m2$$

- Possibly any function
- Different message spaces
- Secret and public key solutions

WHERE FHE COULD BE USED IRL?



ONCE UPON A TIME...

- **1978** - Rivest, Adleman, Dertouzos: *privacy homomorphisms*
- . . .
- **2009** - Gentry: first **fully** homomorphic encryption construction

What happened in the meantime?

Many schemes are homomorphic...

- RSA
- ElGamal
- ...
- Paillier
- Goldwasser-Micali

...but only **partially**.

Some schemes can support both addition and multiplication, but "**with limits**":

- **somewhat**: example the scheme by Boneh, Goh and Nissim 2005
- **leveled**.........

A WORLD FULL OF NOISE...

Example: [DGHV10]

Scheme based on the Approximate GCD problem [HG01], proposed by Van Dijk, Gentry, Halevi, Vaikuntanathan in 2010.

$$\boxed{c = m + 2 \cdot r + p \cdot q}$$

- $m \in \{0, 1\}$ message
- $p \in \mathbb{Z}$ secret key
- $q \in \mathbb{Z}$ large ($p \ll q$)
- $r \in \mathbb{Z}$ small noise ($r \ll p$)

To decrypt: ciphertext modulo p and then modulo 2.

A WORLD FULL OF NOISE...

$$c_1 = m_1 + 2r_1 + pq_1$$

$$c_2 = m_2 + 2r_2 + pq_2$$

Addition (XOR)

$$c_1 + c_2 = (m_1 + m_2) + 2(r_1 + r_2) + p(q_1 + q_2)$$

Noise amount *: double*

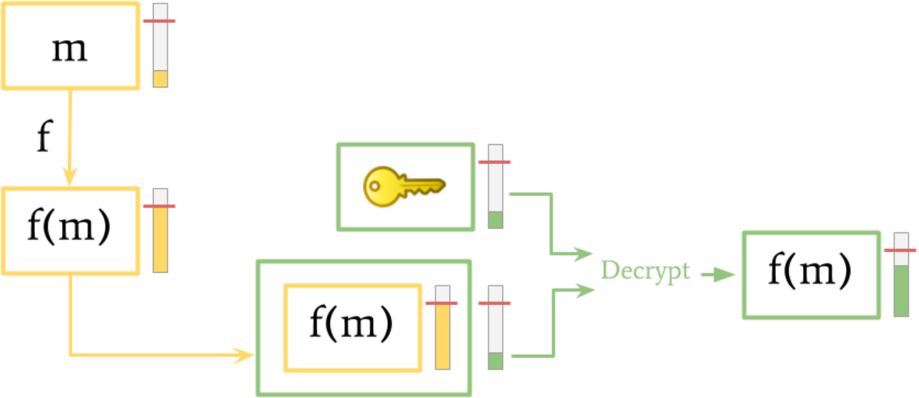
Multiplication (AND)

$$c_1 \cdot c_2 = (m_1 \cdot m_2) + 2(2r_1 \cdot r_2 + \ldots) + p(q_1 \cdot q_2 + \ldots)$$

Noise amount *: square*

If noise grows too much, a correct decryption cannot be guaranteed!

BOOTSTRAPPING [GEN09] AND FHE



BOOTSTRAPPING

Bootstrapping is very costly!

"To bootstrap, or not to bootstrap, that is the question" (semi cit.)

Leveled homomorphic

Set the function, there exist parameters to homomorphically evaluate it.

- ✔ Fast evaluations (for low depth circuits)
- ✘ The depth has to be known in advance

Fully homomorphic

Set the parameters, it is possible to homomorphically evaluate any function.

- ✘ Slower evaluations (**Bootstrapping**)
- ✔ No depth limitations

EXISTING SCHEMES

Lattice problems

Approximate-GCD [HG01], NTRU [HPS98], **(Ring-)LWE** [Reg05],[SSTX09],[LPR10]

Some (Ring-)LWE-based schemes

"BGV-like"

- B(G)V: [BV11], [BGV12]
- B/FV: [Bra12], [FV12]
- HEAAN: [CKKS17]

"GSW-like"

- GSW: [GSW13]
- FHEW: [DM15]
- TFHE: [CGGI16-17]

In practice, they are less different than expected: Chimera [BGGJ19]

Some implementations

- cuFHE
- FHEW
- HEAAN
- HElib
- Lattigo
- Microsoft SEAL
- NFLlib
- nuFHE
- Palisade
- TFHE
- ...

TABLE OF CONTENTS

FHEW

[DM15]

- GSW-based construction
- **They build a FHE brick:**
  a bootstrapped NAND gate
- Slow (but significantly improved):
  $\sim$ 0.69 seconds per bootstrapped NAND gate
- Large bootstrapping keys:
  $\sim$ 1 GByte

[DM15]: L. Ducas, D. Micciancio, *FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second*, EUROCRYPT 2015

TFHE

Bootstrapped versions [CGGI16]

- Slow (but significantly improved):
  ~~~ 0.69~~ $\sim$ 0.05 seconds per bootstrapped NAND gate
- Slow (but significantly improved) [CGGI17]:
  ~~~ 0.69~~ ~~~ 0.05~~ $\sim$ 0.013 seconds per bootstrapped NAND gate
- ~~Large~~ bootstrapping keys: ~~$\sim$ 1 GByte~~ $\sim$ 23.4 MBytes

[CGGI16]: I. Chillotti, N. Gama, M. Georgieva, M. Izabachène, *Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds*, ASIACRYPT 2016

Leveled versions [CGGI17]

- Fast(er) for small depth circuits
- New techniques to improve leveled evaluations
- New Bootstrapping for larger circuits

[CGGI17]: I. Chillotti, N. Gama, M. Georgieva, M. Izabachène, *Faster Packed Homomorphic Operations and Efficient Circuit Bootstrapping for TFHE*, ASIACRYPT 2017

THE REAL TORUS $\mathbb{T} = \mathbb{R}/\mathbb{Z} = \mathbb{R} \mod 1$

Torus

$(\mathbb{T}, +, \cdot)$ is a $\mathbb{Z}$-module

(the external product $\cdot : \mathbb{Z} \times \mathbb{T} \to \mathbb{T}$ is well defined)

- ✔ It is an abelian group: $x + y \mod 1$, $-x \mod 1$, ...
- ✔ It is a $\mathbb{Z}$-module: $0 \cdot \frac{1}{2} = 0$ is defined!
- ✘ It is **not** a Ring: $0 \times \frac{1}{2}$ is **not** defined!

Torus polynomials

$(\mathbb{T}_N[X], +, \cdot)$ is a $\mathfrak{R}$-module

- Here, $\mathfrak{R} = \mathbb{Z}[X]/(X^N + 1)$
- And $\mathbb{T}_N[X] = \mathbb{T}[X] \mod (X^N + 1)$

TFHE CIPHERTEXTS - LWE

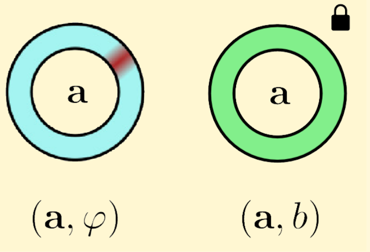
LWE

Message $\mu \in \mathbb{T}$, secret key $\mathbf{s} \in \mathbb{B}^n$

$$\boxed{\mathbf{c} = (\mathbf{a}, b) \in \mathbb{T}^{n+1}}$$

- $\mathbf{a}$ random mask, $b = \mathbf{s} \cdot \mathbf{a} + \varphi$
- $\varphi = e + \mu$, $e \in \mathbb{T}$ Gaussian

$\mathbb{T} = \mathbb{R} \mod 1, \mathbb{B} = \{0, 1\}$



$$\begin{array}{ccc}
\mathrm{LWE}_{\mathbf{s}}(\mu_1) & \boxed{\quad \mathbf{a}_1 \quad} \; \boxed{b_1} & + \\
+ & & \\
\mathrm{LWE}_{\mathbf{s}}(\mu_2) & \boxed{\quad \mathbf{a}_2 \quad} \; \boxed{b_2} & = \\
= & \overline{\hphantom{\quad \mathbf{a}_2 \quad \; b_2 \quad}} & \\
\mathrm{LWE}_{\mathbf{s}}(\mu_1 + \mu_2) & \boxed{\quad \mathbf{a} \quad} \; \boxed{b} & \text{s.t.} \quad \begin{array}{l} \mathbf{a} = \mathbf{a}_1 + \mathbf{a}_2 \\ b = b_1 + b_2 \end{array}
\end{array}$$

TFHE CIPHERTEXTS - RLWE

RLWE

Message $\mu \in \mathbb{T}_N[X]$, secret key $s \in \mathbb{B}_N[X]$

$$\boxed{\mathbf{c} = (a, b) \in \mathbb{T}_N[X]^2}$$

- a random mask, $b = s \cdot a + e + \mu$, $e \in \mathbb{T}_N[X]$ Gaussian

$\mathbb{T}_N[X] = \mathbb{R}[X]/(X^N + 1) \bmod 1$, $\mathbb{B}_N[X] = \mathbb{Z}[X]/(X^N + 1)$ with binary coefs

$\mathrm{RLWE}_{\mathbf{s}}(\mu_1)$	a_1	b_1	$+$
$+$			
$\mathrm{RLWE}_{\mathbf{s}}(\mu_2)$	a_2	b_2	$=$
$=$			
$\mathrm{RLWE}_{\mathbf{s}}(\mu_1 + \mu_2)$	a	b	

s.t. $\quad a = a_1 + a_2$

$\quad b = b_1 + b_2$

TFHE CIPHERTEXTS - RGSW

RGSW

Message $m \in \mathbb{Z}_N[X]$, secret key $\mathbf{s} \in \mathbb{B}_N[X]$ as in RLWE

$$\boxed{C = Z + m \cdot G_2 \in \mathbb{T}_N[X]^{2\ell \times 2}}$$

- with Z is a list of 2ℓ RLWE encryptions of 0
- with G_2 the **gadget** matrix

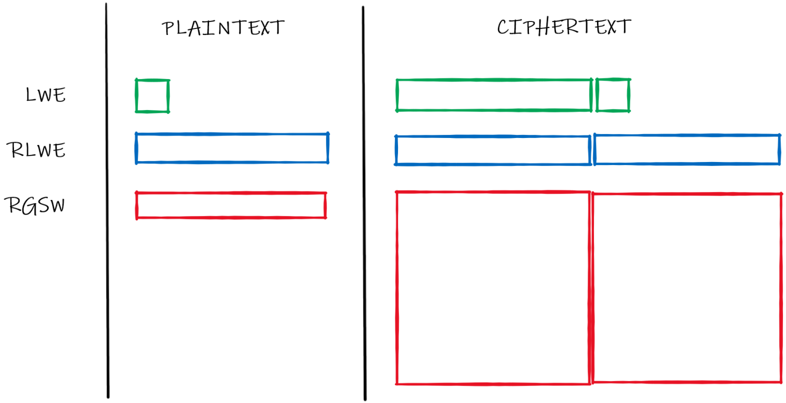
$$G_2 = \left(\begin{array}{c|c} \mathbf{g} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{g} \end{array} \right), \text{ with } \mathbf{g}^T = (2^{-1}, \dots, 2^{-\ell})$$

G_2^{-1}: easy to decompose $\mathbb{T}_N[X]$ elements w.r.t. G_2

$\mathbb{Z}_N[X] = \mathbb{Z}[X]/(X^N + 1)$

TFHE CIPHERTEXTS

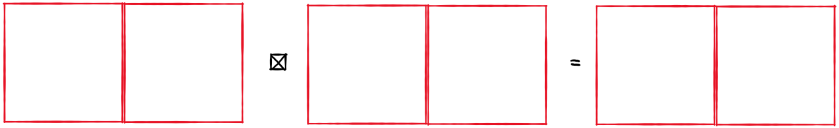
	plaintext	ciphertext	linear combinations	product
LWE	$\mathbb{T}$	$\mathbb{T}^{n+1}$	✔	✘
RLWE	$\mathbb{T}_N[X]$	$\mathbb{T}_N[X]^2$	✔	✘
RGSW	$\mathbb{Z}_N[X]$	$\mathbb{T}_N[X]^{2\ell\times 2}$	✔	✔



TFHE PRODUCTS

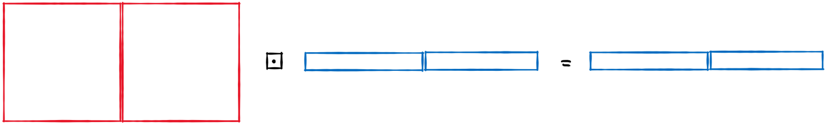
Internal RGSW product

$$C \boxtimes D = G_2^{-1}(D) \cdot C = \begin{bmatrix} G_2^{-1}(\mathbf{d_1}) \cdot C \\ \vdots \\ G_2^{-1}(\mathbf{d_{2\ell}}) \cdot C \end{bmatrix} = \begin{bmatrix} C \boxdot \mathbf{d_1} \\ \vdots \\ C \boxdot \mathbf{d_{2\ell}} \end{bmatrix}$$



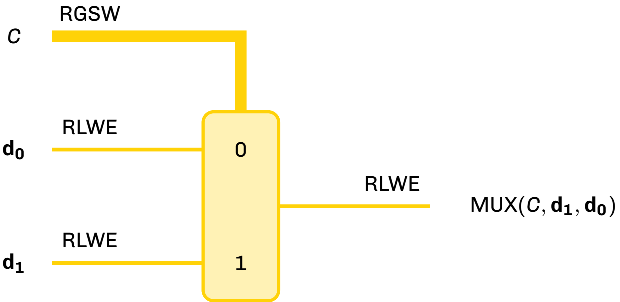
External RGSW – RLWE product [CGGI16],[BP16]

$$C \boxdot \mathbf{d} = G_2^{-1}(\mathbf{d}) \cdot C$$



TFHE MUX

$$\mathrm{MUX}(C, \mathbf{d_1}, \mathbf{d_0}) = C \boxdot (\mathbf{d_1} - \mathbf{d_0}) + \mathbf{d_0}$$



TFHE Mux

Largely used in TFHE leveled and bootstrapped constructions.

MORE TFHE

What we will see in this presentation

- Bootstrapping
- How to use it in ML evaluation

More...

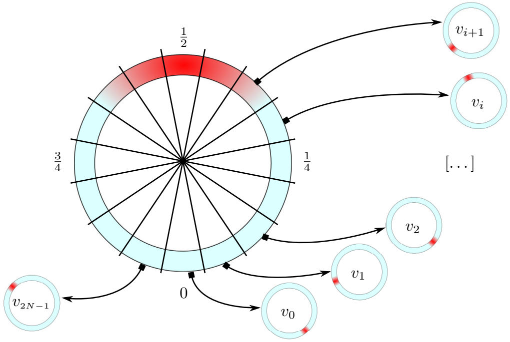
- Evaluation of leveled LUT, deterministic (weighted) finite automata, circuit bootstrapping...
- Multi-key: MK-TFHE [CCS19]
- Neural network applications: [BMMP18], TFHE-Chimera solution at iDASH 2019
- Use in MPC: Onion Ring ORAM [CCR19]

TFHE implementations

- Open source C/C++ library `https://tfhe.github.io/tfhe/` (Apache 2.0 license)
- Experimental repository `https://github.com/tfhe/experimental-tfhe`
- There exist also some GPU implementations: cuFHE, nuFHE

TABLE OF CONTENTS

GATE BOOTSTRAPPING



- Input LWE ciphertext

$$\mathbf{c} = (\mathbf{a}, b)$$

- Depending on

$$\varphi = b - \mathbf{a} \cdot \mathbf{s}$$

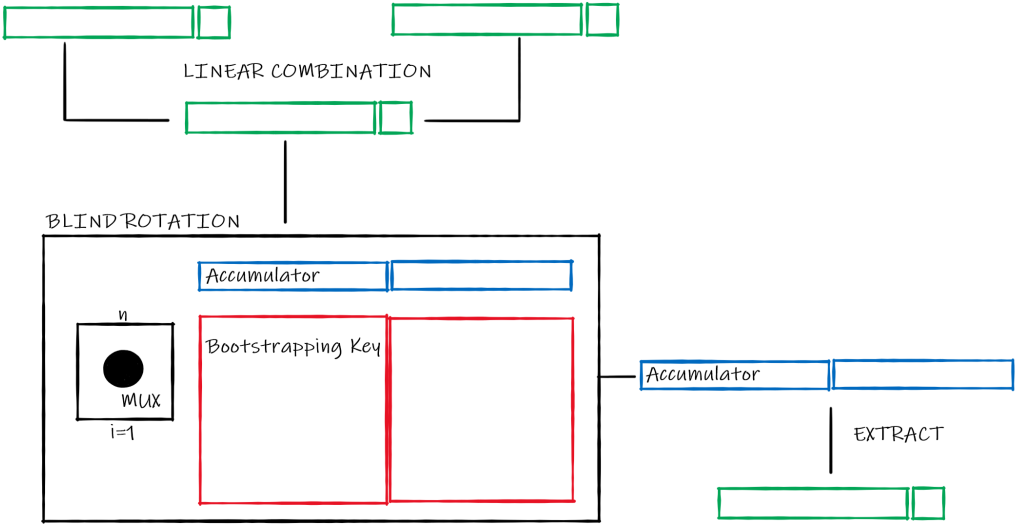
we compute an output LWE ciphertext encrypting $v_\varphi \in \mathbb{T}$

1. Start from (a trivial) RLWE ciphertext of message[1]

$$ACC = v_0 + v_1 X + \cdots + v_{N-1} X^{N-1}$$

2. Do a **blind rotation** of ACC by $-\varphi$ positions (i.e. $ACC \cdot X^{-\varphi}$)
3. **Extract** the constant term of ACC (which encrypts v_φ)

GATE BOOTSTRAPPING



PROGRAMMABLE BOOTSTRAPPING (PBS)

large variance

back to
small variance

bootstrapping

LWE(x) →  → LWE(x)

fast ✓

- any real-valued function can be programmed
- variables over reals

PROGRAMMABLE BOOTSTRAPPING (PBS)

large variance

back to
small variance

programmable
bootstrapping

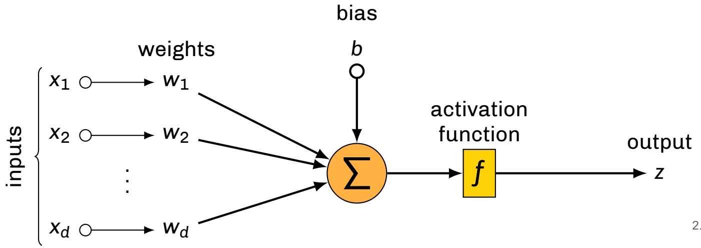
LWE(x) $\longrightarrow$  $\longrightarrow$ LWE($f(x)$)

$f(\cdot)$

- any real-valued function can be programmed
- variables over reals

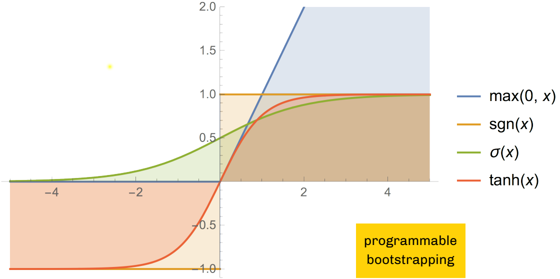
TABLE OF CONTENTS

ARTIFICIAL NEURON



$$y = \sum_{i=1}^{d} w_i \, x_i + b$$

$$z = f(y)$$



programmable bootstrapping

Let's be Concrete

`https://concrete.zama.ai/`

NUMERICAL EXPERIMENTS

- MNIST dataset
- Three neural networks:
  - NN-x where x is the number of layers with $x \in \{20, 50, 100\}$
  - networks all include dense and convolution layers with activation functions
  - every hidden layer possesses at least 92 active neurons
- Two machines:

  PC 2.6 GHz 6-Core Intel® Core™ i7 processor
  AWS 3.00 GHz Intel® Xeon® Platinum 8275CL processor with 96 vCPUs

	In the clear		Encrypted		
	PC	Accuracy	PC	AWS	Accuracy
NN-20	0.17 ms	97.5 %	115.52 s	17.96 s	97.5 %
NN-50	0.20 ms	95.4 %	233.55 s	37.69 s	95.4 %
NN-100	0.33 ms	95.2 %	481.61 s	69.32 s	90.5 %

TABLE OF CONTENTS

PERSPECTIVES & CONCLUSION

First experiments with the Concrete library demonstrate that:

- depth is no longer necessarily an issue
- deep neural networks can actually be evaluated homomorphically

Call for new challenges for fully homomorphic encryption when applied to the inference of deep neural networks

Thank you!

Questions?

https://zama.ai

@zama_crypto

ilaria.chillotti@zama.ai

@IChillotti

BIBLIOGRAPHY

[RAD78] R. L. Rivest, L. Adleman, M. L. Dertouzos. On data banks and privacy homomorphisms. Foundations of secure computation 1978.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. STOC 2009.

[RSA78] R. L. Rivest, A. Shamir, L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 1978.

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory 1985.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. EUROCRYPT 1999.

[GM82] S. Goldwasser, S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. STOC 1982.

[BGN05] D. Boneh, E-J. Goh, K. Nissim. Evaluating 2-dnf formulas on ciphertexts. TCC 2005.

[DGHV10] M. van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan. Fully homomorphic encryption over the integers. EUROCRYPT 2010.

[HG01] N. Howgrave-Graham. Approximate integer common divisors. CaLC 2001.

[HPS98] J. Hoffstein, J. Pipher, J. H. Silverman. NTRU: A ringbased public key cryptosystem. ANTS-III 1998.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. STOC 2005.

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka, K. Xagawa. Efficient public key encryption based on ideal lattices. ASIACRYPT 2009.

[LPR10] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. EUROCRYPT 2010.

[BV11] Z. Brakerski, V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. CRYPTO 2011.

[BGV12] Z. Brakerski, C. Gentry, V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. ITCS 2012.

BIBLIOGRAPHY

[Bra12] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. CRYPTO 2012.
[FV12] J. Fan, F. Vercauteren. Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive, 2012.
[CKKS17] J. H. Cheon, A. Kim, M. Kim, Y. Song. Homomorphic encryption for arithmetic of approximate numbers. ASIACRYPT 2017.
[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. CRYPTO 2013.
[DM15] L. Ducas, D. Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. EUROCRYPT 2015.
[CGGI16] I. Chillotti, N. Gama, M. Georgieva, M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. ASIACRYPT 2016.
[CGGI17] I. Chillotti, N. Gama, M. Georgieva, M. Izabachène. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. ASIACRYPT 2017.
[BGGJ19] C. Boura, N. Gama, M. Georgieva, D. Jetchev. CHIMERA: Combining Ring-LWE-based Fully Homomorphic Encryption Schemes. NutMic 2019.
[BP16] Z. Brakerski, R. Perlman. Lattice-based fully dynamic multi-key FHE with short ciphertexts. CRYPTO 2016.
[BMMP18] F. Bourse, M. Minelli, M. Minihold, P. Paillier. Fast Homomorphic Evaluation of Deep Discretized Neural Networks. CRYPTO 2018.
[CCS19] H. Chen, I. Chillotti, Y. Song. Multi-Key Homomorphic Encryption from TFHE. ASIACRYPT 2019.
[CCR19] H. Chen, I. Chillotti, L. Ren. Onion Ring ORAM: Efficient Constant Bandwidth Oblivious RAM from (Leveled) TFHE. CCS 2019.