DOI: https://doi.org/10.1109/JPROC.2022.3205665

Survey on Fully Homomorphic Encryption, Theory and Applications

Chiara Marcolla, Victor Sucasas, *Member, IEEE,* Marc Manzano, Riccardo Bassoli, *Member, IEEE,* Frank H.P. Fitzek, *Senior Member, IEEE* and Najwa Aaraj

***Abstract*—Data privacy concerns are increasing significantly in the context of Internet of Things, cloud services, edge computing, artificial intelligence applications, and other applications enabled by next generation networks. Homomorphic Encryption addresses privacy challenges by enabling multiple operations to be performed on encrypted messages without decryption. This paper comprehensively addresses homomorphic encryption from both theoretical and practical perspectives. The paper delves into the mathematical foundations required to understand fully homomorphic encryption (FHE). It consequently covers design fundamentals and security properties of FHE, and describes the main FHE schemes based on various mathematical problems. On a more practical level, the paper presents a view on privacy-preserving Machine Learning using homomorphic encryption, then surveys FHE at length from an engineering angle, covering the potential application of FHE in fog computing, and cloud computing services. It also provides a comprehensive analysis of existing state-of-the-art FHE libraries and tools, implemented in software and hardware, and the performance thereof.**

***Index Terms*—Fully Homomorphic Encryption, Homomorphic Encryption, Lattices, Neural Networks, Fog Computing, Cloud Computing, IoT.**

I. Introduction

The notion of fully homomorphic encryption, originally called privacy homomorphism, was introduced by Rivest, Adleman and Dertouzos [1] in 1978. For more than 30 years, this concept was considered to be the holy grail of cryptography, until 2009, when Gentry proposed the first fully homomorphic encryption scheme in his PhD thesis [2]. Homomorphic encryption enables operations on plaintexts without decryption. Namely, a set of operations can be performed over ciphertexts such that these operations are reflected as additions and multiplications on the corresponding plaintexts. Thus, homomorphic encryption allows data manipulation in the encrypted domain. This has a tremendous application potential since it allows privacy-preserving data processing, which can be adopted in new emerging fields such as machine learning, cloud computing, or in the different data processing layers of new generation networks.

Homomorphic encryption schemes that allow one type of operation, or a limited number of operations, have existed for a long time. Some examples are the RSA cryptosystem by Rivest, Shamir and Adleman [3] (1978), encryption scheme of Goldwasser and Micali [4] (1982), ElGamal [5] (1985), Benaloh [6] (1994), Naccache and Stern [7] (1998), Paillier [8] (1999) and Boneh, Goh and Nissim [9] (2005). In particular in [9], the authors proposed the first scheme capable of performing two operations: an arbitrary number of additions and just one multiplication, then again an arbitrary number of additions. Later, Aguilar Melchor, Gaborit and Herranz [10] (2008) proposed a theoretical approach that permits chaining several homomorphic schemes in order to have a fixed amount of multiplications, i.e. more than one, for a given public key.

However, it was not until 2009, when Gentry [2], [11] proposed the first fully homomorphic encryption (FHE) scheme which supports the evaluation of arbitrary circuits. In his thesis, Gentry not only proposed an FHE scheme, but also provided a method for constructing a general FHE scheme from a scheme with limited but sufficient homomorphic evaluation capacity. Since then, homomorphic encryption has triggered significant interest, and novel constructions on FHE have been proposed following Gentry's idea, being BGV [12], FV [13], TFHE [14], and CKKS [15] the most representative.

The majority of research efforts for FHE schemes focused on public key encryption schemes. Symmetric FHE schemes have gained less popularity among the scientific community, due to their more limited applicability to cloud computing. Also, some proposed symmetric key schemes still suffer from security vulnerabilities, as pointed in [16]. Nevertheless, there are some papers that proposed symmetric key FHE schemes, which can be divided into two categories: i) schemes with both symmetric key and public key versions [2], [17]–[19]; ii) and purely symmetric key FHE schemes [20], [21]. It is also worth commenting that, in 2011, Rothblum [22] published a method to convert a symmetric key homomorphic encryption scheme with sufficient homomorphic evaluation capacity, into an asymmetric one. This survey only covers public key FHE schemes, but more information on symmetric key schemes can be found in [16].

This survey provides a comprehensive vision on homomorphic encryption since its genesis. We extend previous surveys on the topic [16], [23]–[26], to cover the most relevant advances on FHE and its applications. Specifically, the survey is structured as follows: i) Section II provides the preliminaries,

C. Marcolla, V. Sucasas and N. Aaraj are with the Technology Innovation Institute, Masdar City Abu Dhabi, United Arab Emirates. (e-mail: {chiara.marcolla,victor.sucasas,najwa.aaraj}@tii.ae).

M. Manzano is with Sandbox Quantum, Palo Alto, CA, US. (e-mail: marc@sandboxquantum.com). M. Manzano is also with the Electronics and Computing Department, Faculty of Engineering, Mondragon Unibertsitatea, Mondragon, Spain.

R. Bassoli and F.H.P. Fitzek are with the Deutsche Telekom Chair of Communication Networks, Institute of Communication Technology, Faculty of Electrical and Computer Engineering, Technische Universität Dresden, Dresden, Germany.

F.H.P. Fitzek is also with Centre for Tactile Internet with Human-in-the-Loop (CeTI) , Cluster of Excellence, Dresden, Germany. (e-mail: {riccardo.bassoli,frank.fitzek}@tu-dresden.de).

containing the required definitions on Number Theory and Probability Theory to understand the constructions of FHE schemes; ii) Section III introduces the mathematical definition of lattices, and the main mathematical problems used in lattice-based cryptography; iii) Section IV defines the concept and the construction of fully homomorphic encryption; iv) Section V caters for an extensive description of FHE schemes in the state of the art, classified according to their generation; and v) Section VI discusses the security of FHE schemes by presenting the different mathematical problems in which they are based. Following a more practical perspective, the survey also describes: i) the application of homomorphic encryption in machine learning in Section VII; ii) the potential adoption of homomorphic encryption for data aggregation in fog computing in Section VIII; iii) previously proposed techniques for the application of homomorphic encryption in cloud computing in Section IX; and iv) homomorphic encryption in practice, describing the current frameworks and libraries, in Section X. Finally, Section XI discusses open challenges in the field and concludes this survey.

II. Preliminaries

This section presents the mathematical foundations and the notation required to understand the developments of homomorphic encryption covered in the next sections.

A. Number Theory

Groups: A group G is a set with an associative operation such that there exists an identity element of G and every element of G has an inverse. If the group operation is commutative, the group is *abelian* (also called *commutative*). For $t \in \mathbb{N}$, a *finite group* $\mathbb{Z}_t$ is the subgroup of positive integers modulo t. This group is also referred to as $\mathbb{Z}/t\mathbb{Z}$ in number theory, and it can be seen as the set of integers in $(-t/2, t/2]$. A multiplicative group $\mathbb{Z}_t^{\times}$, for some $t \in \mathbb{N}$, is the set of integers modulo t that are co-prime to t. Note that if t is prime, then $\mathbb{Z}_t$ is the (finite) field $\mathbb{F}_t$.

Fields: A field is a set with two operations, addition and multiplication. The set is an abelian group under addition with 0 as identity and its nonzero elements are an abelian group under multiplication with 1 as identity. The multiplication is distributive over addition. A *finite field* $\mathbb{F}_q$ is a field with q elements and it exists if and only if q is prime or a prime power. Finally, a *number field* is a vector space with finite dimension over rational numbers $\mathbb{Q}$ and a *cyclotomic field* is a number field obtained by adjoining a complex root of unity to $\mathbb{Q}$.

Rings: A ring generalizes a field since multiplication does not need to be commutative and some elements do not have the multiplicative inverse. A *quotient ring* $R = \mathbb{Z}[x]/\langle f(x)\rangle$, is a ring of polynomials with integer coefficients modulo the (monic) polynomial $f(x)$. Note that the multiplication between two polynomials in R is a modular multiplication. If the coefficients of the polynomial are in $\mathbb{Z}_q$ (integers modulo q), then we denote it R_q, specifically $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$. It is worth pointing out that the ring R is a field if and only if $f(x)$ is an irreducible polynomial over $\mathbb{Z}$.

Sets: Given a set E, we denote by E^n the set of vectors such that $E^n = \{\mathbf{e} = (e_1, \ldots, e_n) : e_i \in E\}$. Let E be a commutative ring, then the dot product of two vectors $\mathbf{u}, \mathbf{v}$ in E^n is defined as $\langle \mathbf{u}, \mathbf{v}\rangle = \sum_{i=1}^{n} u_i \cdot v_i$. Similarly, $\mathcal{M}_{h,w}(E)$ denotes the set of matrices of size $h \times w$ with entries in E. Specifically, the term $(\mathbb{Z}_q)^h$ denotes a vector of size h with elements in $\mathbb{Z}_q$ and $(\mathbb{Z}_q)^{h\times w}$ a matrix of size $h \times w$ also with elements in $\mathbb{Z}_q$.

Ideal: An *ideal* I is a subset of a ring R containing 0 (i.e. the inverse element of the addition) such that the addition of two elements in I is also in I, and the multiplication of an element in I by an element in R is also in I. A *principal* ideal is an ideal generated by one element. In other words, a principal ideal generated by a is the set of multiples of a.

Real Torus: The *real torus* $\mathbb{T}$ is the set $\mathbb{R}/\mathbb{Z}$ of real numbers modulo 1. Note that $\mathbb{T}$ is a group, when using the addition.

Norms: Let $\mathbf{x}$ be a vector in E, then we define $||\mathbf{x}||_\ell := \sqrt[\ell]{\sum_i |x_i|^\ell}$ the ℓ-norm and $||\mathbf{x}||_\infty := \max_i |x_i|$ the infinity-norm of $\mathbf{x}$, where x_i are the elements in x. We denote by $||\mathbf{x}||$ the Euclidean norm of the vector $\mathbf{x}$, that is equivalent to the 2-norm. The Euclidean norm can also be referred as the length of a vector. The norms $||g(x)||_\ell$ and $||g(x)||_\infty$ of a real or integer polynomial $g(x)$ are the norms of its coefficient vector. If $g(x)$ is a polynomial mod $x^n + 1$, we take the norm of its unique representative of degree less or equal than $n - 1$. Moreover, when E is the real torus $\mathbb{T}$, then the ℓ-norm of $\mathbf{x} \in \mathbb{T}^k$ is the ℓ-norm of the representative of $\mathbf{x}$ with all coefficients in $(-1/2, 1/2]$. With abuse of notation, we denote it by $||\mathbf{x}||_\ell$.

B. Probability Theory

Negligible and Overwhelming probability: Let $f(\kappa) : \mathbb{N} \to \mathbb{R}$ be a function where for any possible integer c there exists a value N such that for all $\kappa > N$ it holds that $|f(\kappa)| < \frac{1}{\kappa^c}$. This function is said to be negligible. If the the output of a negligible function is a probability, then the probability is negligible. In cryptography κ is frequently referred to as security parameter and it represents the length of the secret values. Normally, provably secure schemes are defined by presenting an attack which probability of success is negligible with respect to the security parameter, i.e. it can become arbitrarily small by increasing κ. Analogously, an overwhelming probability is the output of a function $f'(\kappa) = 1 - f(\kappa)$ such that f is a negligible function. Hence, an overwhelming probability can become arbitrarily close to 1 by increasing κ.

Gaussian distribution: The general form of a *normal* (or *Gaussian*) distribution χ for a random variable $x \in \mathbb{R}$ is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where μ is the center or mean of the distribution and σ is its standard deviation. Note that in the majority of the FHE literature χ is defined as a discrete Gaussian distribution on $\mathbb{Z}$ with centre zero and width parameter αq, denoted by $\mathcal{D}_{\mathbb{Z},\alpha q}$. The discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z},\alpha q}$ over the integers is defined by assigning a weight proportional to $\exp(-\pi x^2/(\alpha q)^2)$ to all $x \in \mathbb{Z}$. Namely, [27] for any $x \in \mathbb{Z}$,

$$\mathcal{D}_{\mathbb{Z},\alpha q}(x) = \frac{f(x)}{f(\mathbb{Z})} \text{ where } f(\mathbb{Z}) = \sum_{z \in \mathbb{Z}} \frac{1}{\alpha q} e^{-\pi\left(\frac{z}{\alpha q}\right)^2}.$$

Moreover, the standard deviation of $\mathcal{D}_{\mathbb{Z},\alpha q}$ is $\sigma \approx \alpha q/\sqrt{2\pi}$ if σ is bigger than the smoothing parameter $\eta_\epsilon(\mathbb{Z})$ of $\mathbb{Z}$ [28].

Let $R = \mathbb{Z}[x]/\langle f(x) \rangle$ be a polynomial ring. Informally, we denote a B-*bounded* distribution χ over R, if the norm of the coefficients of a polynomial sampled from χ is less than B with overwhelming probability. In general, B is set to be as small as possible while maintaining security (e.g., if χ in R_q, B $\ll q$). In the rest of this work we denote *small element* as any sample from a B-bounded distribution χ.

III. Lattices

This section introduces the mathematical definition of lattices, and it also describes the main mathematical problems, on which the security of lattice-based homomorphic encryption schemes relies. The section intends to be self-contained, but it also provides references for interested readers to find full mathematical descriptions and proofs.

A. Definitions

A k-dimensional lattice is a discrete additive subgroup of $\mathbb{R}^n$. Let $B = (\mathbf{b}_1, \ldots, \mathbf{b}_k)$ be linearly independent vectors in $\mathbb{R}^n$, then we can define the *lattice* $\mathcal{L}(B)$ generated by B as the set of all integer linear combinations of elements of B:

$$\mathcal{L} = \mathcal{L}(B) = \Big\{ \sum_{i=1}^{k} \gamma_i \mathbf{b}_i : \quad \gamma_i \in \mathbb{Z}, \mathbf{b}_i \in B \Big\}.$$

The term B is called *base* of the lattice, and the *parallelepiped* associated to the basis B, is defined as

$$\mathcal{P}(B) = \Big\{ \sum_{i=1}^{k} x_i \mathbf{b}_i : \quad x_i \in [-1/2, 1/2) \Big\}.$$

The *rank* k of a lattice $\mathcal{L} \subset \mathbb{R}^n$ is the dimension of its linear span, that is, $k = \dim(\text{span}(\mathcal{L}))$. When $k = n$, the lattice is said to be *full rank*. The *volume* of $\mathcal{L}$ (also called *determinant*) is defined as $\mathsf{Vol}(\mathcal{L}) = \sqrt{\det(B^t B)}$. In the special case that $\mathcal{L}$ is a full rank lattice, we have that $\mathsf{Vol}(\mathcal{L}) = |\det(B)|$.

The *dual* of a lattice $\mathcal{L}$ is the set

$$\mathcal{L}^* = \{\mathbf{v} \in \text{span}(\mathcal{L}) : \quad \langle \mathbf{v}, \mathbf{b} \rangle \in \mathbb{Z} \text{ for all } \mathbf{b} \in \mathcal{L}\}.$$

Note that, for any $B \in \mathbb{R}^{n \times n}$, $\mathcal{L}(B)^* = \mathcal{L}((B^{-1})^T)$. From this it follows that $\det(\mathcal{L}^*) = 1/\det(\mathcal{L})$. Moreover, given a matrix $A \in (\mathbb{Z}_q)^{m \times n}$ for some integers q, m, n, we can define two integer q-*ary* lattices [29],

$$\begin{aligned} \mathcal{L}_q(A) :=& \quad \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x} = A\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\}, \\ \mathcal{L}_q^{\perp}(A) :=& \quad \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}A \equiv 0 \bmod q\}. \end{aligned}$$

Note that $\mathcal{L}_q^{\perp}(A)$ is a *scaled* dual lattice of $\mathcal{L}_q(A)$, namely, $\mathcal{L}_q^{\perp}(A) = q \cdot \mathcal{L}_q(A)^*$.

It is also worth defining the *ideal lattice* $\mathcal{L}(I)$, which is an integer lattice $\mathcal{L}(B) \subseteq \mathbb{Z}^n$ where $B = \{g \bmod f : g \in I\}$, $I \subseteq \mathbb{Z}[x]/\langle f \rangle$ is an ideal, and f a monic polynomial of degree n.

The *Hermite factor* δ_0^n is defined as

$$\delta_0^n = ||\mathbf{b}_1|| / \mathsf{Vol}(\mathcal{L})^{1/n} \tag{1}$$

where $\mathbf{b}_1$ is a first basis vector, i.e. a shortest vector, in the base B of the full rank lattice $\mathcal{L}$. The factor δ_0 is called the *root Hermite factor*. It is worth highlighting that root Hermite factor is an important indicator about the quality of some lattices attacks, as shown in Section VI.

B. Lattice Distance

Most of the known attacks to FHE schemes are based on the notion of *distance*. The concept of distance comes in natural a way. Specifically, for any vector $\mathbf{t}$ in $\mathbb{R}^n$ and any element $\mathbf{v}$ of a lattice $\mathcal{L}$, the distance between these two vectors is defined as $\text{dist}(\mathbf{t}, \mathbf{v}) = ||\mathbf{t} - \mathbf{v}||$. Consequently, the minimum distance between $\mathbf{t}$ and any element in $\mathcal{L}$, is

$$\text{dist}(\mathbf{t}, \mathcal{L}) = \min\{||\mathbf{t} - \mathbf{v}|| : \mathbf{v} \in \mathcal{L}\}.$$

Let us define the *minimum distance* of lattice $\mathcal{L}$ as $\lambda_1(\mathcal{L})$, which is the length of a shortest non-zero vector in $\mathcal{L}$, i.e.

$$\lambda_1(\mathcal{L}) = \min\{||\mathbf{v}|| : \mathbf{v} \in \mathcal{L}, \mathbf{v} \neq 0\}.$$

We can generalize the notion of minimum distance by defining the *i-th successive minimum* $\lambda_i(\mathcal{L})$ as the smallest radius r of a zero-centred ball that contains i (or more) linearly independent lattice points.

A comprehensive explanation of lattices in cryptography can be found in [30], [31] by Peikert and in [29] by Micciancio and Regev.

C. Shortest Vector Problem

The security of many lattice-based FHE schemes relies on the intractability of the shortest vector problem and its variants. The *Shortest Vector Problem* (SVP) consists of finding the shortest non-zero vector in a given lattice. If we restrict the set of input lattices to ideal lattices, then we obtain the *Ideal*-SVP [32]. The relevant variants of SVP problem are defined below:

- γ-*approximate Shortest Vector Problem* (SVP_γ) consists in identifying a vector that is *almost* the shortest vector. Formally, given $\gamma \geq 1$, it consists in finding a non-zero vector $\mathbf{v} \in \mathcal{L}$ such that $||\mathbf{v}|| \leq \gamma \cdot \lambda_1(\mathcal{L})$.

- *Decisional Shortest Vector Problem* ($\mathsf{GapSVP}_{\gamma,r}$) consists in establishing which given bound for the shortest vector is correct. Specifically, given $\gamma \geq 1$ and $r > 0$, it consists in deciding if either $\lambda_1(\mathcal{L}) \leq r$ or

$\lambda_1(\mathcal{L}) \geq \gamma \cdot r$.

- *γ-unique Shortest Vector Problem* (uSVP_γ) consists in finding a shortest non-zero vector in a lattice $\mathcal{L}$ where $\lambda_1(\mathcal{L})$ is $\gamma\lambda_1(\mathcal{L}) < \lambda_2(\mathcal{L})$ for $\gamma \geq 1$. Namely, the shortest vector is guaranteed to be at least γ times smaller than $\lambda_2(\mathcal{L})$.

Note that if the shortest vector problem is solvable, then the decisional shortest vector problem is also solvable, namely, $\mathsf{GapSVP}_{\gamma,r} \leq \mathsf{SVP}$. Interested readers can find concrete instantiations of the SVP problem in [33].

D. Closest Vector Problem

A generalization of the shortest vector problem is the *Closest Vector Problem* (CVP). In this generalization, a target vector $\mathbf{t} \in \mathbb{R}^n$ is given instead of using the zero vector, and the problem consists of finding a vector in the lattice $\mathbf{v} \in \mathcal{L}$ that is the closest to $\mathbf{t}$, i.e. $\mathrm{dist}(\mathbf{t}, \mathbf{v}) = \mathrm{dist}(\mathbf{t}, \mathcal{L})$. This problem also has some variants:

- *γ-approximate Closest Vector Problem* (CVP_γ) consists in finding a vector in the lattice that is *almost* a closest to the target vector. Formally, given $\gamma \geq 1$ and $\mathbf{t} \in \mathbb{R}^n$, it consists in finding $\mathbf{v} \in \mathcal{L}$ such that $\mathrm{dist}(\mathbf{t}, \mathbf{v}) \leq \gamma \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L})$.

- *Decisional Closest Vector Problem* ($\mathsf{DCVP}_{\gamma,r}$), given $\gamma \geq 1$, $r > 0$ and $\mathbf{t} \in \mathbb{R}^n$, consists in deciding if either $\mathrm{dist}(\mathbf{t}, \mathcal{L}) \leq r$ or $\mathrm{dist}(\mathbf{t}, \mathcal{L}) \geq \gamma \cdot r$.

- Let $\alpha \leq 1$. *α-Bounded Distance Decoding* (BDD_α) problem consists in, given a target vector $\mathbf{t}$ such that $\mathrm{dist}(\mathbf{t}, \mathcal{L}) < \alpha\lambda_1(\mathcal{L})$, finding a lattice vector $\mathbf{v} \in \mathcal{L}$ closest to $\mathbf{t}$.

E. Relations Between Shortest and Closest Vector Problems

Kumar and Sivakumar [34] proved that the uSVP_γ problem is NP-hard when $\gamma = 1 + 2^{-n^c}$, for some constant c, whereas Liu, Lyubashevsky and Micciancio [35] showed that the BDD_α problem is NP-hard for $\alpha > 1/\sqrt{2}$. Moreover, Lyubashevsky and Micciancio [36] also proved that $\mathsf{uSVP} \leq \mathsf{BDD}$ and $\mathsf{uSVP} \leq \mathsf{GapSVP}$. More specifically, they proved that for any $\gamma \geq 1$ and $\gamma \leq poly(n)$, the problem uSVP_γ, $\mathsf{BDD}_{1/\gamma}$ and GapSVP_γ are equivalent up to polynomial approximation factors. Finally, Bai, Stehlé and Wen [37], preprocessing the lattice with Khot's sparsification technique [38], gave a *probabilistic* polynomial-time reduction from $\mathsf{BDD}_{1/(\sqrt{2}\gamma)}$ to uSVP_γ, for any $\gamma > 1$ and γ polynomial in n. The latter restriction was lifted in Wen's PhD thesis [39].

F. Short Integer Solution Problem

The *Short Integer Solution* (SIS) problem was introduced in 1996 by Ajtai [40]. The goal of the SIS problem is to find an integer vector with small norm that is a solution of a given system of integer equations. More specifically, let $q \in \mathbb{Z}$ and let $A \in (\mathbb{Z}_q)^{m \times n}$ be a matrix. Then, given $\beta < q$, the $\mathsf{SIS}_{q,m,\beta}$ problem consists of finding a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ with $||\mathbf{x}|| \leq \beta$ such that $\mathbf{x}A \equiv 0 \bmod q$. It is worth noting that solving the $\mathsf{SIS}_{q,m,\beta}$ problem is equivalent to finding a vector $\mathbf{v}$ with norm $||\mathbf{v}|| \leq \beta$ in the scaled dual $\mathcal{L}_q^{\perp}(A)$ of the lattice $\mathcal{L}_q(A)$. So this problem can be seen as a kind of SVP_γ for this particular family of lattices. Moreover, it is important to highlight that the solution to the SIS problem needs to be bounded on the length, i.e. $||\mathbf{x}|| \leq \beta$. Without this restriction, it would be easy to find $\mathbf{x}$ with the Gaussian elimination technique.

The ring version of this problem is given by Micciancio in 2002 [41] (extended version in 2007 [42]). Informally, let $q \in \mathbb{Z}$ and $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$ where $f(x) \in \mathbb{Z}[x]$ is a monic polynomial of degree d, and let $\mathbf{a} \in (R_q)^m$ be a vector of m polynomials. Then, given $\beta < q$, the Ring-$\mathsf{SIS}_{q,m,\beta}$ problem [43] consists of finding a non-zero vector of small polynomials $\mathbf{x} \in R^m$ with $||\mathbf{x}|| \leq \beta$ such that $\mathbf{a}^T\mathbf{x} \equiv 0 \bmod q$.

G. Learning With Errors Problem

The key role that lattice-based problems play in cryptography nowadays is especially due to the *Learning With Errors* (LWE) problem and its *Decisional* version, which were introduced by Regev in 2005 [44] (full version [45] in 2009) as an extension of the "learning from parity with error" problem of Blum, Furst, Kearns, and Lipton [46]. Their definitions are provided below:

- Given a vector $\mathbf{b} \in \mathbb{Z}_q^m$ and a matrix $A \in (\mathbb{Z}_q)^{m \times n}$, the LWE problem[1] consists in finding an unknown vector $\mathbf{s} \in \mathbb{Z}_q^n$ such that

$$A\,\mathbf{s} + \mathbf{e} = \mathbf{b} \bmod q$$

  where $\mathbf{e} \in \mathbb{Z}_q^m$ is sampled coordinate-wise from an error distribution χ. In other words, the goal is to find a vector $\mathbf{s} \in \mathbb{Z}_q^n$ given a list of $m = n + 1$ *noisy* equations from

$$A_{\mathbf{s},\chi} = \{(\mathbf{a}_i, b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q : \mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_q^n,\ e_i \xleftarrow{\$} \chi\};$$

- the *Decision Learning With Errors* (DLWE) problem consists of distinguishing (with non-negligible advantage) m samples chosen according to $A_{\mathbf{s},\chi}$ (for uniformly random $s \in \mathbb{Z}_q^n$), from m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

In [44], Regev reduced the worst case decisional shortest vector GapSVP in a lattice to the LWE problem via a *quantum* reduction. Namely, the paper shows that if it is possible to find an algorithm to solve the LWE problem in polynomial time, then it is also possible to solve quantumly the GapSVP problem in polynomial time. Because of that, the security of LWE-based homomorphic encryption schemes is intimately related to lattice problems like SVP and its variants. In the same paper, Regev proved that the LWE problem and

[1] This instance is also referred to as the *search* version of the LWE problem.

its decisional version are computationally equivalent for any prime q that is polynomially-bounded $poly(n)$. Subsequently, this result was extended to any modulus q in [47]–[51]. In 2009 Peikert [47] (and successively Lyubashevsky and Micciancio [36]) provided a classical reduction from GapSVP problem to LWE with exponential modulus. In [51] Brakerski, Langlois, Peikert, Regev and Stehlé proved that the LWE with *polynomial* modulus is at least as hard as worst-case lattice problems, via a classical reduction.

H. Ring Learning With Errors Problem

The *Ring Learning With Errors* (RLWE) problem, introduced by Stehlé, Steinfeld, Tanaka and Xagawa in [32], is the "ring version" of LWE. Specifically, the LWE is in $(\mathbb{Z}_q)^{n+1}$, while in RLWE is in $(R_q)^2$, where $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$ where $f(x) \in \mathbb{Z}[x]$ is a monic, irreducible polynomial of degree d and q is a prime.

- The RLWE problem is to discover $s \in R_q$ given access to arbitrarily many independent samples $(a, b = s \cdot a + e) \in R_q \times R_q$, where a is chosen uniformly at random in R_q, and $e \in R_q$ is sampled from an error distribution χ.
- The *Decision Ring Learning With Errors* (DRLWE) problem consists in distinguishing with non-negligible advantage between independent and uniformly random samples in $R_q \times R_q$ and the same number of independent RLWE instances (where $s \in R_q$ is uniformly random).

It is worth highlighting that the secret s can be chosen from the error distribution since, as Applebaum, Cash, Peikert and Sahai [52] proved, it does not affect the hardness of the LWE problem.

Initially, the RLWE problem was called *Ideal*-LWE problem in [32] and, later, its decision version was called *Polynomial Learning With Errors* (PLWE) in [18] by Brakerski and Vaikunthanathan. Also, Lyubashevsky, Peikert and Regev [53] slightly modified the initial definition of RLWE proposed in [32]. Namely, the secret s and the noisy polynomial b are in $R_q^\vee$, where $R^\vee$ is a *particular* ideal that is dual to R. One of the main contributions in this paper is a search form to the decision reduction.

I. Relations Between LWE, RLWE and Hard Lattice Problems

As explained in Sections III-G and III-H the hardness of LWE and RLWE problems is related to various well-known hard lattice problems [28], [29], [45], [51], [53]. Specifically, the hardness of the RLWE problem is described in [32], where the search variant was proved hard under the ideal-SVP and, in [53] where Lyubashevsky, Peikert and Regev cater for a quantum reduction from the γ-approximate SVP to the RLWE problem and a classical reduction from search to decisional RLWE assumption. Unlike the quantum reduction, which works for any number field and (almost) any modulus, the classical reduction works only for *particular* modulus q and defining polynomial $f(x)$. In [54] Ducas and Durmus partially improved [53] by generalizing over $f(x)$. The restriction on q was lifted by Langlois and Stehlé [55]. Finally, in 2017 Peikert, Regev and Stephens-Davidowitz [56] presented a polynomial-time quantum reduction from worst-case (ideal) lattice problems to decision RLWE which works in any number field and any modulus.

For completeness, we also mention the *General Learning With Errors* (GLWE) problem introduced by Brakerski, Gentry and Vaikunthanathan in [12]. The GLWE problem is the interpolation between the RLWE and LWE problems. Specifically, it consists in finding $s \in R_q^k$ given a list of noisy equations from:

$$\{(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e) \in R_q^k \times R_q, \text{ where } \mathbf{a} \xleftarrow{\$} R_q^k, \mathbf{e} \xleftarrow{\$} \chi\}$$

For $R = \mathbb{Z}$ we have LWE and for $k = 1$ we have RLWE. The GLWE problem was proven to be hard by Langlois and Stehlé in [55], hence generalizing the results of [53]. From a cryptographic point of view, schemes based on the RLWE problem are more computationally efficient and more compact (in terms of ciphertext size) than the ones based on LWE [31]. We refer readers to [31] by Peikert for a comprehensive and more detailed explanation about LWE, RLWE and the hardness of these problems.

IV. Fully Homomorphic Encryption

A *fully homomorphic encryption* scheme can be defined as an encryption scheme where, given some ciphertexts, any operation over the plaintexts can be performed without decryption by manipulating the ciphertexts directly. Such functionality is achievable if and only if addition and multiplication operations can be performed homomorphically, since these two operations constitute a functionally complete set over finite rings. Specifically, any boolean (arithmetic) circuit can be represented using only XOR (addition) and AND (multiplication) gates. In other words, given two ciphertexts $\mathsf{Enc}(x)$ and $\mathsf{Enc}(y)$, where x and y are plaintexts and Enc is the encryption operation, we can obtain the encryption of $x + y$ (or $x \cdot y$) without decrypting $\mathsf{Enc}(x)$ and $\mathsf{Enc}(y)$, by simply adding (or multiplying) these two ciphertexts, and this is sufficient to evaluate any function over encrypted data such that

$$\mathsf{Dec}(\mathsf{Enc}(x) \vartriangle \mathsf{Enc}(y)) = x \vartriangle y,$$

where $\vartriangle$ is the operation sum or product and Dec is the decryption operation.

As shown in Section IV-A, homomorphic encryption is based on probabilistic algorithms. Generally, the encryption procedure adds a random element r which is called *noise* or *error*[2]. The intrinsic feature of FHE is that the error increases any time a homomorphic operation is carried out. So, after a certain number of multiplications (or additions) the ciphertext cannot be decrypted correctly due to the error growth. This problematic limits encryption schemes with homomorphic properties from being fully homomorphic. Only a bounded number of operations can be performed homomorphically

[2]The noise can be added directly as input in the encryption algorithm or included indirectly, e.g. in the public key [57]

before the plaintext cannot be decrypted correctly, hence these schemes are referred to as *somewhat homomorphic*. To overcome this limitation, Gentry [2], [11] introduced a new technique, *bootstrapping*, which is detailed in Section IV-B. This procedure can be used to convert a scheme that is not fully homomorphic into one that is. The method proposed by Gentry can be divided into two steps:

1) Creating a *somewhat homomorphic* encryption scheme, that is, a scheme that supports a limited number of homomorphic operations;
2) Using *bootstrapping* to reduce the error added during encryption, and making the ciphertext compact.

It is worth commenting that Gentry proposes an additional step, before bootstrapping, to reduce the decryption complexity. This technique, called (*squashing*), and it is used to express the decryption function as a function with a lower degree. Squashing the decryption circuit is necessary in order to have a bootstrappable decryption circuit in Gentry's scheme. However, unlike bootstrapping, the squashing technique was not generally adopted in subsequent schemes.

We refer the reader to the following interesting overviews on fully homomorphic encryption [23], [24] and to the Homomorphic Encryption Security Standard white paper [58], where the authors provide tables of recommended parameters used for specific FHE schemes at various security levels, considering particular attacks.

A. Definitions and Basic Notions

A public key homomorphic encryption scheme $\mathcal{E}$ [2] is composed of a set of probabilistic polynomial-time (PPT) algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ such that:

- The public key-generation algorithm KeyGen takes as input the security parameter λ and outputs the secret key sk, the public key pk and the (public) evaluation key evk, which is needed to perform homomorphic operations over ciphertexts.

- The public encryption algorithm Enc takes as input the public key pk and a message m from the message space. Subsequently, it outputs a ciphertext c.

- The decryption algorithm Dec takes as input the secret key sk and a ciphertext c. Next, it outputs a message m. The algorithm provides as output $\perp$, if the decryption algorithm cannot successfully recover the encrypted message m.

- The evaluation algorithm Eval takes as input the evaluation key evk, a function f and t-uple of ciphertexts $(c_1, \ldots, c_t)$. It outputs a ciphertext c_f, such that it decrypts to the result of the evaluation of $(m_1, \ldots, m_t)$ over f, i.e. $c_f = \mathsf{Eval}_{\mathsf{evk}}(f, (c_1, \ldots, c_t))$ and $\mathsf{Dec}_{\mathsf{sk}}(c_f) = f(m_1, \ldots, m_t)$. Note that the ciphertexts c_f and $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(f(m_1, \ldots, m_t))$ are equivalent in the sense that they decrypt to the same plaintext, but different in their construction (e.g. they may have different noise levels).

There are two essential characteristics of a homomorphic encryption scheme $\mathcal{E}$: i) the maximum degree of a function that the scheme supports; ii) the length increase of the ciphertext after each homomorphic operation. The first property defines what functions $\mathcal{E}$ is able to evaluate correctly. Specifically, the scheme $\mathcal{E}$ is *$\mathcal{F}$-homomorphic* if it can correctly evaluate any function f in $\mathcal{F}$, that is, if there exists an evaluation algorithm Eval such that

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}_{\mathsf{evk}}(f, c_1, \ldots, c_t)) = f(m_1, \ldots, m_d) \text{ for all } f \in \mathcal{F},$$

where $c_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_i)$ for any $i \in \{1, \ldots, t\}$. Also, it defines whether an evaluated ciphertext c_f, namely an output of Eval, can be used as an input of the evaluation algorithm. Specifically, in a *multi-hop homomorphic* scheme [59], a sequence of any (polynomial) i functions can be homomorphically evaluated one by one on a ciphertext c produced by encrypting a message m. The second property refers to the ciphertext expansion, i.e. how much the ciphertext bit-length grows after each evaluation. In the sense, if the bound of the bit-length growth is independent of the complexity of f it is called *compact* [59].

Depending on the previous notions, we have different definitions of homomorphic encryption schemes:

- *Fully Homomorphic Encryption* scheme $\mathcal{E}$ is an encryption scheme where the ciphertexts are compact, and the scheme is $\mathcal{F}$-homomorphic, where $\mathcal{F}$ is the set of all the (efficiently computable) functions [60].

- *Leveled fully homomorphic* scheme is a scheme that supports the evaluation of specific depth circuits. More formally, it is $\mathcal{F}$-homomorphic, where $\mathcal{F}$ is the set of all functions of some specific degree and the bound over the length of the ciphertext is independent of such degree (i.e. it is compact) [12]. In this type of schemes the depth is treated as a setup parameter of the scheme and can adopt any value. It is worth commenting that if the degree is bounded to a maximum value L then the scheme is called *L-Leveled fully homomorphic* scheme. Note that L-leveled schemes may not be compact.

- *Somewhat Homomorphic Encryption* (SHE) scheme is a scheme that is $\mathcal{F}$-homomorphic for a limited class $\mathcal{F}$, e.g. capable of evaluating "low-degree" multivariate polynomials homomorphically [12]. Similarly to L-leveled schemes, in an SHE scheme the compactness of ciphertexts could be violated.

It is worth highlighting that, as mentioned in Section IV, an SHE scheme (and a leveled fully homomorphic scheme), with sufficient homomorphic evaluation capacity, can be transformed into an FHE scheme by using the bootstrapping technique.

B. Bootstrapping

Bootstrapping is a technique to decrease the error of the ciphertext, proposed by Gentry in [2], [11]. Essentially, it

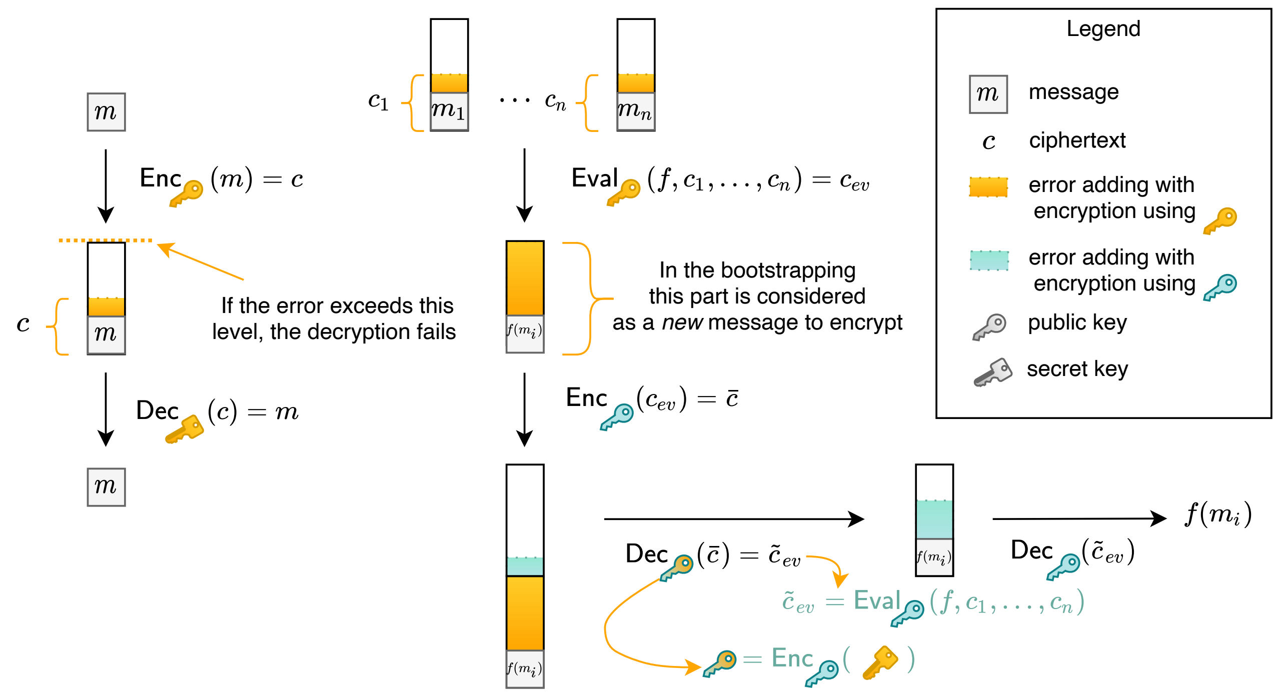

Fig. 1. Bootstrapping technique.

consists of a re-encryption procedure of a ciphertext c to *refresh* it, i.e. encrypt it again under another key obtaining a new ciphertext (for the same plaintext) but with a smaller error.

Let us consider a somewhat homomorphic encryption scheme $\mathcal{E}$, two pairs of keys $(\mathsf{sk}_1, \mathsf{pk}_1)$ and $(\mathsf{sk}_2, \mathsf{pk}_2)$, the encryption algorithm Enc and the ciphertext $c = \mathsf{Enc}_{\mathsf{pk}_1}(m)$ that encrypts m under pk_1. The procedure to refresh c is conducted in three steps as follows:

- Encrypting the secret key sk_1 under pk_2: $\mathsf{Enc}_{\mathsf{pk}_2}(\mathsf{sk}_1) \rightarrow \overline{\mathsf{sk}_1}$. The encryption of sk_1 may require its bit decomposition and thus produce many ciphertexts.

- Encrypting the ciphertext c under pk_2: $\mathsf{Enc}_{\mathsf{pk}_2}(c) = \bar{c}$. In most schemes this step is essentially vacuous, in the sense that $\mathsf{Enc}_{\mathsf{pk}_2}(c)$ is obtained by using null randomness, i.e. viewing a plaintext directly as a ciphertext (with proper padding/scaling).

- Decrypting homomorphically the *new* ciphertext using the encrypted secret key: $\mathsf{Dec}_{\overline{\mathsf{sk}_1}}(\bar{c})$. In this way, an encryption of the same message under the second public key $\mathsf{Enc}_{\mathsf{pk}_2}(m)$ is obtained. Namely:

$$\begin{aligned} &\mathsf{Eval}_{evk}(\mathsf{Dec}, \bar{c}, \overline{sk_1}) = \\ &\mathsf{Eval}_{evk}(\mathsf{Dec}, \mathsf{Enc}_{pk_2}(c), \mathsf{Enc}_{pk_2}(sk_1)) = \\ &\widehat{\mathsf{Enc}}_{pk_2}(\mathsf{Dec}_{sk_1}(c)) = \widehat{\mathsf{Enc}}_{pk_2}(m) \end{aligned} \quad (2)$$

the value $\widehat{\mathsf{Enc}}_{pk_2}(m)$ is equivalent to $\mathsf{Enc}_{pk_2}(m)$ in the sense that both decrypt to m, .i.e

$$\mathsf{Dec}_{sk_2}(\widehat{\mathsf{Enc}}_{pk_2}(m)) = \mathsf{Dec}_{sk_2}(\mathsf{Enc}_{\mathsf{pk}_2}(m)) = m$$

The objective of bootstrapping is to reduce the error of the ciphertext. Conceptually, bootstrapping applies the decryption function and simultaneously performs a second encryption. These operations produce a new ciphertext. This new ciphertext contains the error of a new encryption plus the error increase resulting from the homomorphic evaluation of the decryption circuit. Hence, the error of the obtained ciphertext is higher than a fresh ciphertext (obtained with the encryption algorithm) but lower than a ciphertext obtained after homomorphic evaluating functions with higher depth than the decryption circuit. The idea of bootstrapping is illustrated in Figure 1. Bootstrapping can be applied to any ciphertext, hence it can be used after several homomorphic evalations to reduce the error level. This enables the homomorphic evaluation of functions with arbitrarily large depth, but requires the decryption circuit to be boostrappable.

The cornerstone result obtained by Gentry is the proof that to construct an FHE scheme suffices to construct a scheme that is capable of evaluating only a particular set of small degree functions, i.e. an SHE scheme, which security holds when the encryption of the secret key is published. If that is the case, then bootstrapping can be applied (and squashing if necessary) to obtain an FHE scheme. It is worth highlighting that bootstrapping is the only known way to obtain fully homomorphic encryption schemes. Unfortunately, bootstrapping is computationally complex and requires a large memory space.

C. Security Properties

A homomorphic encryption scheme must be semantically secure, but optionally it can also be function or circuit private. The scheme is secure if and only if it is semantically secure. Semantic security is formally captured by the concept of

Fig. 2. Timeline of the main FHE schemes.

indistinguishability under chosen-plaintext attack (IND-CPA security), where an attacker can obtain encryptions of arbitrary plaintexts, but it cannot decrypt arbitrary ciphertexts (note that the encryption algorithm and the encryption key pk are public, whereas the decryption key sk is not). If we restrict the message space to $\{0, 1\}$, then:

Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be a public homomorphic scheme, and* m_b *a message with* $\{0, 1\}$ *as the message space. Let us define an adversary* $\mathcal{A}$ *that knows the evaluation key* evk *and the public key* pk *and is given an encryption* $\mathsf{Enc}_{\mathsf{pk}}(m)$ *for* $m \in \{0, 1\}$. $\mathcal{A}$ *can make queries to the encryption oracle. After a polynomial number of queries,* $\mathcal{A}$ *tries to guess whether* $m = 0$ *or* $m = 1$*. Then, the scheme is IND-CPA secure if for an efficient adversary* $\mathcal{A}$*, it holds that:*

$$\Big| \Pr[\mathcal{A}(\mathsf{pk}, \mathsf{evk}, \mathsf{Enc}_{\mathsf{pk}}(0)) = 1] - \Pr[\mathcal{A}(\mathsf{pk}, \mathsf{evk}, \mathsf{Enc}_{\mathsf{pk}}(1))) = 1] \Big| = \mathsf{negl}(\lambda)$$

where $(\mathsf{sk}, \mathsf{pk}, \mathsf{evk}) \leftarrow \mathsf{KeyGen}(\lambda)$.

The definition above expresses the fact that the adversary is not able to tell apart encryptions of 0 from encryptions of 1 with non-negligible probability. It is worth noting that IND-CPA security is only achievable if the encryption scheme randomizes the ciphertexts. If there is no randomization, the adversary can encrypt messages, and then compare them with the received ciphertext $\mathsf{Enc}_{\mathsf{pk}}(m)$.

Finally, an encryption scheme that is secure against adversaries who observe an encryption of the scheme's secret key under its public key is called *circular* secure. Current constructions of fully homomorphic encryption schemes require an encryption of the secret key to be bootstrappable, hence all known FHE constructions require circular security. This implies that IND-CPA security has to hold under circular security. Most FHE schemes are not proven IND-CPA secure under circular security, and it is in general adopted as an additional assumption on top of the scheme's underlying security assumptions. Optionally, the homomorphic encryption scheme can be *function private*. That is, a ciphertext that has been homomorphically evaluated over a function f and does not reveal any information about f, beyond the outputs for the queried inputs. Function privacy is the relaxed version of the original *circuit privacy* [2], which requires that the evaluated ciphertext is statistically indistinguishable from a fresh ciphertext [61]. Note that for a scheme to be circuit or function private, the property has to hold even against an adversary that knows the secret key and can decrypt any ciphertext.

V. Fully Homomorphic Encryption Schemes

Gentry's seminal works [2], [11] paved the way for novel FHE schemes in four main research branches: i) the schemes based on ideal lattices (Section V-A); ii) the schemes over integers based on the *Approximate - Greatest Common Divisor* (AGCD) problem (Section V-B); iii) the schemes based on the LWE problem and on its ring version (Sections V-C, V-E and V-F); and iv) the schemes based on NTRU (Section V-D). Additionally, other research works have proposed schemes based on other mathematical problems (Section V-H). Figure 2 describes the timeline of the main FHE schemes.

A. *First Generation:* FHE *based on ideal lattices*

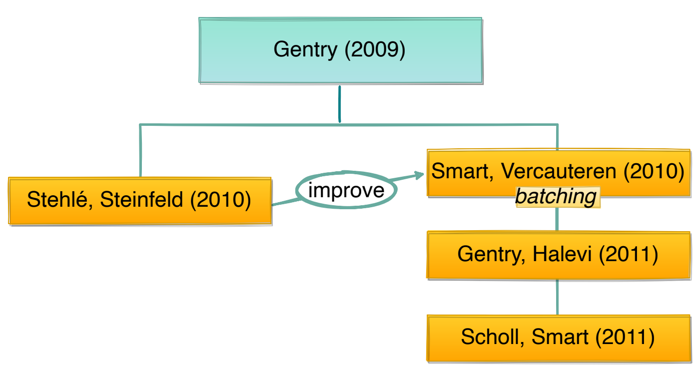

Fig. 3. Main FHE schemes based on ideal lattices.

The first fully homomorphic encryption scheme presented by Gentry [2], [11] is based on ideal lattices. Figure 3 describes the main schemes based on ideal lattices.

Following the notation of [62], we describe one of the constructions presented by Gentry. This scheme uses the integer sublattice $\mathcal{L}(I) \subseteq \mathbb{Z}^m$ defined by the ideal I. The encryption and decryption algorithms work as follows:

- *encryption*: the message $m \in \mathbb{F}_2$ (the message is a single bit, $m \in \{0,1\}$) is encoded into a point $\mathbf{a} = m + 2\mathbf{e}$ in $\mathbb{R}^d$, where $\mathbf{e}$ is a small error, namely a random vector with coefficients in $\{0, \pm 1\}$, where the values ± 1 are taken with equal probability. The ciphertext $\mathbf{c}$ is the translation of $\mathbf{a}$ into the parallelepiped $\mathcal{P}(B_{\mathsf{pk}})$ where B_{pk} is the public key. Namely, $\mathbf{c} = \mathbf{a} - (\lceil \mathbf{a} B_{\mathsf{pk}}^{-1} \rfloor B_{\mathsf{pk}})$, where $\lceil \cdot \rfloor$ denotes rounding to the nearest integer.

- *decryption*: computes $\mathbf{a}' = \mathbf{c} - (\lceil \mathbf{c} B_{\mathsf{sk}}^{-1} \rfloor B_{\mathsf{sk}})$, which is the translation of $\mathbf{c}$ into the parallelepiped $\mathcal{P}(B_{\mathsf{sk}})$ where B_{sk} is the secret key, and then outputs $\mathbf{a}' \bmod 2$ as the decrypted plaintext.

Both the public and secret keys, B_{pk} and B_{sk}, are bases of the ideal lattice $\mathcal{L}(I)$. However, the public key is a bad basis for I, in the sense that it is formed by skewed vectors, whereas the secret key is a good basis because it is formed by orthogonal vectors (see Figure 4).

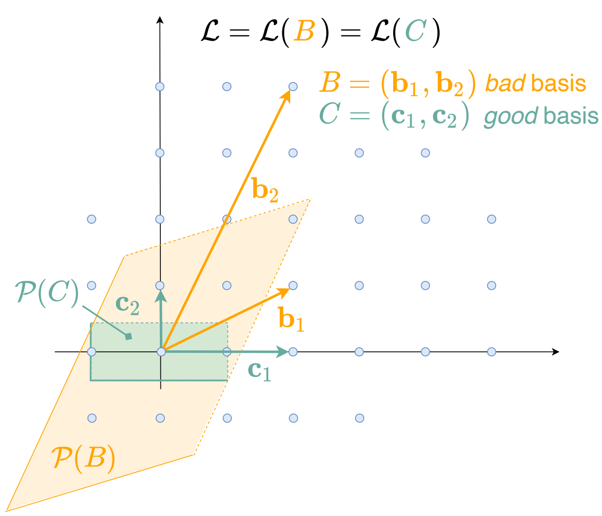

Fig. 4. Good and bad bases for an ideal lattice.

Unfortunately, the scheme that we just described is not *bootstrappable* due to the complexity of the decryption algorithm (i.e. it cannot be evaluated homomorphically). To solve this problem, Gentry proposed a method to *squash* the decryption function of an SHE scheme. This method consists of transforming the original SHE scheme $\mathcal{E}$ into another scheme $\mathcal{E}^*$ with the same homomorphic capacity but with a simpler decryption function that allows the bootstrapping. To reduce the decryption algorithm's complexity, Gentry proved [2], [11] that it is enough to add to the evaluation key some "extra information" about the secret key. Such extra information consists of a set of vectors $\mathcal{S} = \{\mathbf{s}_i \mid i = 1, \ldots, S\}$ from which a subset T is derived. The secret key sk is the sum of elements of T, and the public information included in the evaluation key is the set $\mathcal{S}$. The security of this new scheme $\mathcal{E}^*$ is based on fact that set T is sparse and secret such that the *Sparse Subset Sum Problem* (SSSP) applies. SSSP consists in verifying, given a set of n integers $S = \{a_1, \ldots, a_n\} \subseteq \mathbb{Z}$, whether a sparse subset of S exists such that $\sum_{i \in I} a_i = 0$, where $I \subseteq \{1, \ldots, n\}$. Its security is based on three mathematical problems: i) the *Sparse Subset Sum Problem*; ii) the *Bounded Distance Decoding Problem* (Section III-D); iii) the *Ideal-Shortest Vector Problem* (Section III-C). Circular security is also required but no proof is given in the paper, hence it is an additional security assumption for this scheme.

Gentry's scheme was initially implemented by Smart and Vercauteren [63] who used principal ideal lattices and introduced the *batching technique*. The batched version of a scheme enables the packing of a vector of ℓ plaintexts to be encrypted in a single ciphertext using the Chinese Remainder Theorem (CRT). This technique permits processing several messages simultaneously. The Smart-Vercauteren implementation was improved in 2011 by Gentry and Halevi [64], who also simplified the *squashing procedure*. Successively, Scholl and Smart [65] ameliorated the Gentry-Halevi technique providing a generalization over any cyclotomic field. Stehlé and Steinfeld in [66] reduced the bit complexity (i.e. quantity of operations per bit) for refreshing the ciphertext. Their technique can be applied to different FHE schemes, such as Gentry [2] and Smart and Vercauteren [63].

A drawback of ideal lattice-based FHE schemes is that they are based on mathematical constructions that are difficult to implement efficiently. Also, a vulnerability in schemes using principal ideals was found by Cramer, Ducas, Peikert and Regev [67] in 2016. Specifically, a key-recovery attack for cryptographic constructions based on principal ideal lattices is possible, given a quantum polynomial-time or classical $2^{n^{2/3-\epsilon}}$-time algorithm for finding the short generator of the principal ideal problem.

B. First Generation: FHE *based on the* AGCD *problem*

A new (and simpler than ideal lattice-based) family of FHE schemes dawned in 2010 thanks to van Dijk, Gentry, Halevi and Vaikuntanathan, who introduced a fully homomorphic encryption scheme over integers [19]. The basic construction of the DGHV scheme is the following:

- *key generation*: outputs the secret key p, i.e. an odd random integer, and the public key $(x_0, \ldots, x_n)$ where x_0 is odd and $x_0 > x_i \ \forall i$, where $x_i = pq_i + r_i$ with q_i, r_i random integers.

- *encryption*: the message $m \in \mathbb{F}_2$ is encoded into the ciphertext $c = (m + 2r + 2\sum_{i \in S} x_i) \bmod x_0$, where r is a random integer and S is a random subset of $\{1, \ldots, n\}$.

- *decryption*: computes $(c \bmod p) \bmod 2$.

The security of this scheme is based on SSSP and the Approximate - Greatest Common Divisor (AGCD) problem

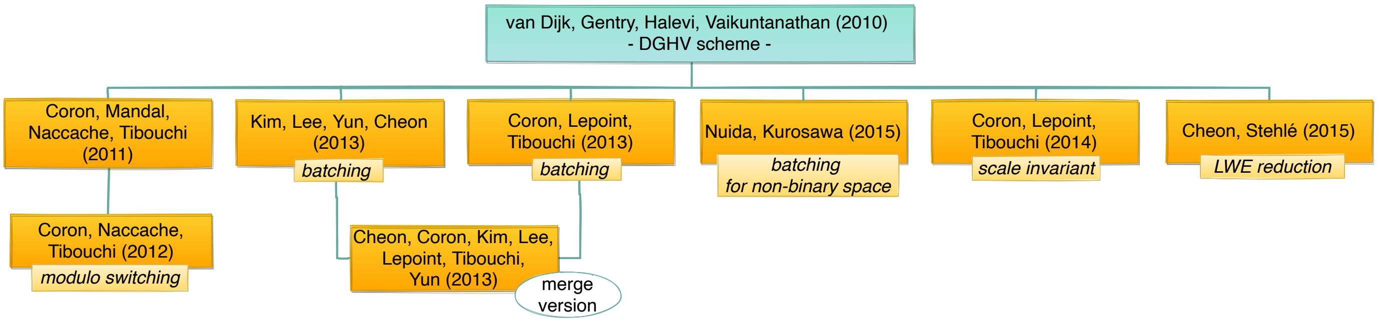

Fig. 5. Main FHE schemes based on the AGCD problem.

which consists of finding the "common near divisor" p, given a set of integers $\{x_0, \ldots, x_n\} \in \mathbb{Z}$, all randomly chosen and close to multiples of a large integer p. The scheme also assumes circular security.

The main drawbacks of the DGHV scheme are its high computational complexity and large public key size. Several optimizations and implementations have been proposed, as depicted in Figure 5. Specifically, in [68], Coron, Mandal, Naccache and Tibouchi reduced the elements of the public key that have to be stored ($2\sqrt{n}$ instead of n elements), since the rest of the key elements can be recovered. This optimization only requires a slight modification of the encryption procedure. In [69] Chan and Nguyen presented new algorithms to solve the AGCD problem, which are exponentially faster than the previous one. As a consequence, they proved that the scheme in [68] achieves lower security level than initially claimed. Later Coron, Naccache and Tibouchi [70] further reduced the size of the public key using the *modulo switching* technique. The security of both these schemes is based on the AGCD assumption with error-free problem. Informally, this problem is similar to the AGCD problem but with the stronger assumption that x_0 does not have the error r_0 considered in the conventional AGCD assumption. It is worth commenting that this is quantumly broken by Shor [71].

A *batch* version of DGHV scheme was independently proposed in 2013 by Kim, Lee, Yun and Cheon [72] and by Coron, Lepoint and Tibouchi [73] (the merged version is in [74]). Moreover, Nuida and Kurosawa [75] proposed the batching technique for non-binary message spaces. In 2014 Coron, Lepoint and Tibouchi [76] improved the DGHV scheme using the *scale-invariance* property given by Brakerski [50]. In 2015, Cheon and Stehlé [77], inspired by [50], introduced a reduction from LWE to AGCD and then presented a new AGCD-based fully homomorphic encryption scheme based on the hardness of this new variant. It is worth highlighting that this construction is the first DGHV variant that did not require the SSSP hardness assumption.

C. *Second Generation:* FHE *based on* LWE *and* RLWE

In 2011, Brakerski and Vaikuntanathan introduced, leveraging the bootstrapping technique, two FHE schemes based on the LWE [17] (extended version in 2014 [78]) and the RLWE [18] (see Sections III-G to III-I) problems, and the circular security assumption. These works, described in Figure 6, initiated the second generation of FHE schemes. The LWE-based symmetric scheme described in [78], known as BV, is described as follows:

- *encryption:* the message $m \in \mathbb{F}_2$ is encoded into a ciphertext $\mathbf{c}$ such that
$$\mathbf{c} = (\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + 2e + m) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$
where e is the error randomly chosen from an error distribution χ and $\mathbf{s} \in \mathbb{Z}_q^n$ is the secret key composed of random elements in $\mathbb{Z}_q$.
- *decryption*: outputs the plaintext $(b - \langle \mathbf{a}, \mathbf{s} \rangle \bmod q) \bmod 2$, which is equal to $(2e + m \bmod q) \bmod 2$. The decryption works properly if e is *small enough*, specifically $e < q/2$.

In this paper, the authors also introduced two novel techniques called *re-linearization* and *dimension-modulus reduction*. The re-linearization is needed to reduce the multiplication ciphertext size from almost $n^2/2$ back to regular size, i.e. $n + 1$. To obtain this reduction the authors transform the quadratic equation of $\mathbf{c}_1 \cdot \mathbf{c}_2$ into a linear equation by means of "encrypting" all the terms of the symmetric key under a new key. Later on, Brakerski, Gentry and Vaikunthanathan [12] called this technique *key switching*

The *dimension-modulus reduction* technique (called also *modulus switching* [12] in subsequent works) converts an SHE into an FHE scheme transforming a ciphertext $\mathbf{c}$ modulo q into another ciphertext $\mathbf{c}'$ modulo p, where p is sufficiently smaller than q. Specifically, each element in $\mathbb{Z}_q$ is converted into an element in $\mathbb{Z}_p$ by first multiplying it by p/q, and then taking the closest integer. An interesting side effect of this operation is that the error in the ciphertext decreases[3]. The lower noise growth due to the adoption of modulus switching allows to homomorphically evaluate the decryption circuit without the squashing method proposed by Gentry. Thus, the SSSP assumption is no longer required (see Section IV-B).

In [18], Brakerski and Vaikunthanathan proposed a new version of the BV scheme, where the scheme security is

[3]The error decreases when comparing ciphertexts before and after modulus switching, but the new ciphertext has also reduced modulus, and the error level relative to its modulus is actually higher after modulus switching since this technique introduces some error.

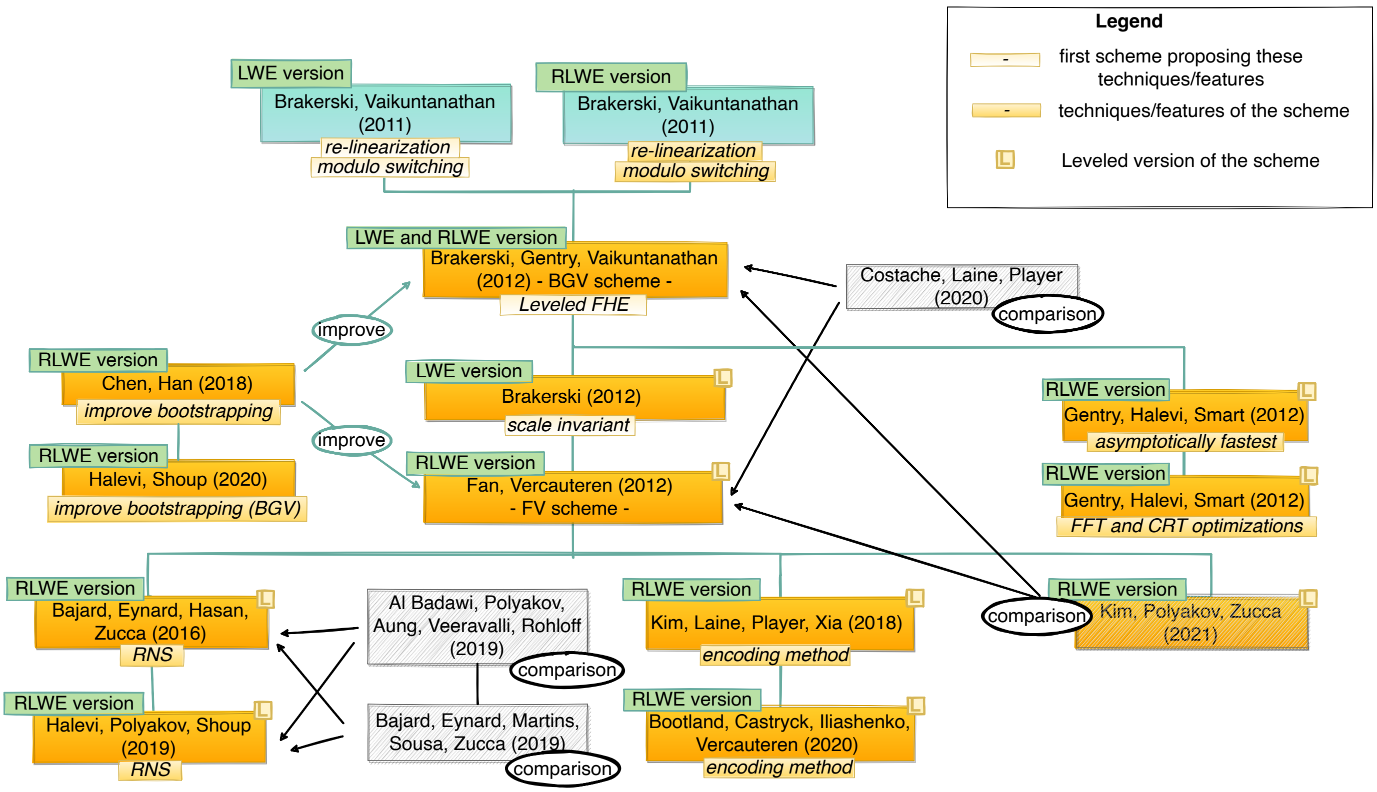

Fig. 6. Main second generation FHE schemes based on the LWE and RLWE problem.

based on the polynomial LWE (PLWE) problem. Note that the PLWE problem is equivalent to the RLWE problem [32] (see Section III-H). The main difference between the LWE version of the BV scheme and the RLWE version is that it represents the message, the ciphertext and the keys, as elements in R_q, where $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$ with $f(x) \in \mathbb{Z}[x]$ is a polynomial of degree d and q is a prime.

In [12], Brakerski, Gentry and Vaikunthanathan proposed a method for defining a leveled fully homomorphic scheme (see Section IV-A), which avoids the computationally expensive bootstrapping technique. Based on this method, they defined the BGV scheme, and provided batching and modulus switching techniques. Additionally, the authors also gave a bootstrapping technique to transform the leveled version into an FHE version.

This new method made the scheme applicable to practical scenarios, thus fostering increasing interest from the research community. The authors introduced two variants of the BGV scheme (one based on LWE and another on RLWE assumption). The RLWE-based version of BGV scheme, described in Scheme V.1, is more efficient than the LWE counterpart, and it is implemented in the widely used FHE library HElib [79] (from IBM). The library, implemented by Shai Halevi and Victor Shoup, is open source and it enables the construction of a boolean circuit of any depth (more information about libraries can be found in Section V-G). It is worth commenting that the architecture of the library is based on a variant of the BGV scheme that introduces some optimizations, proposed by Gentry, Halevi and Smart [80], for the specific implementation of the AES circuit [81]. Also, in [82] Gentry, Halevi and Smart, using the batch techniques of Smart-Vercauteren [63] and Brakerski-Gentry-Vaikuntanathan [17], provided the asymptotically fastest scheme, with significant impact towards fast bootstrapping.

Scheme V.1: BGV RLWE-based scheme [12]

Let d be a power of 2, q be an odd positive integer modulus, and χ be an error distribution over R, where $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$. Let B be a bound (with overwhelming probability) on the length of elements outputted by χ. B is set to be as small as possible while maintaining security. For any natural integer p we write $R_p = \mathbb{Z}_p[x]/\langle x^d + 1\rangle$. The scheme works as follows:

- *key generation*: takes as input the security parameter λ and randomly chooses a small secret element $s \in \chi$ and sets the secret key sk $= \mathbf{s} = (1, s) \in R_q^2$. It generates $a' \in R_q$ uniformly at random and computes $b = a's + 2e$ where e is a random error in χ. It outputs sk and pk $= \mathbf{a} = (b, -a')$.
  Note that $\langle \mathbf{a}, \mathbf{s}\rangle = 2e$.

- *encryption*: takes as input the public key pk $\in R_q^2$ and the message $m \in R_2$. It converts m into a vector $\mathbf{m} = (m, 0) \in R_q^2$ and chooses random $r, e_0, e_1 \in \chi$. It outputs the ciphertext $\mathbf{c} = \mathbf{m} + 2(e_0, e_1) + \mathbf{a}r$. Namely, $\mathbf{c} = (c_0, c_1) = (m + 2e_0 + br, 2e_1 - a'r) \in R_q^2$.

- *decryption*: takes as input the secret key $\mathbf{s} \in R_q^2$ and the ciphertext $\mathbf{c} \in R_q^2$. It computes $\langle \mathbf{c}, \mathbf{s} \rangle = c_0 + c_1 s = m + 2e_0 + 2e_1 s + 2er$ and outputs $((m + 2(e_0 + e_1 s + er)) \bmod q) \bmod 2 = m$. Note that the decryption works properly because e, e_0, e_1 and $\mathbf{s}$ are *small enough* (since are elements of χ).

- *(homomorphic) properties.*
  - The addition is a component-wise addition. The decryption works as long as the resulting error does not overlap the modulus q.
  - The multiplication is slightly more complicated. Note that,

$$\begin{aligned}\langle \mathbf{c}, \mathbf{s} \rangle \cdot \langle \mathbf{c}', \mathbf{s} \rangle &= (c_0 + c_1 s)(c_0' + c_1' s) \\ &= c_0 c_0' + (c_0 c_1' + c_1 c_0')s + c_1 c_1' s^2 \\ &= d_0 + d_1 s + d_2 s^2.\end{aligned}$$

    So the *extended* ciphertext (d_0, d_1, d_2) can be decrypted using a *extended* secret key $(1, s, s^2)$. The inconvenience is that every multiplication expands the decryption key. So to reduce the decryption key, the authors use the key switching technique. Roughly speaking, the basic idea of this method is to convert the ciphertext term $d_2 s^2$ to $\bar{c}_0 + \bar{c}_1 s$ using the encryption of s^2 under s (this is possible under a circular security assumption). Indeed, $\mathsf{Enc}_{\mathbf{s}}(s^2) = (\beta, -\alpha)$, where

$$\begin{aligned}(\beta, -\alpha) &= (s^2 + 2e + a'rs, 2e_1 - a'r) \\ &\approx (s^2 + \alpha s, -\alpha).\end{aligned}$$

    Thus, $s^2 \approx \beta - \alpha s$ and the extended ciphertext $d_0 + d_1 s + d_2 s^2$ becomes a *normal* ciphertext $\tilde{c}_0 + \tilde{c}_1 s$ encrypting the same plaintext.

The leveled version of the schemes can be found in [12, sec. 4.4].

Brakerski [50] provided another variant of the BGV scheme, based on a technique called *scale invariant*. This technique reduces the error increase produced by homomorphic multiplications from exponential to linear. Intuitively, the idea behind the scale invariant technique is to scale down the ciphertext and the error by a factor of q, where q is the ciphertext modulus. This method replaces the modulus switching technique.

Another relevant contribution of [50] is a classical reduction to prove that the security of the scheme is based on the hardness of the GapSVP problem (see Section III-C). Previous schemes were, initially, proven secure only with quantum reductions (although now they also count with classical reductions).

The RLWE version of the Brakerski scheme was implemented and optimized by Fan and Vercauteren [13], and named FV scheme (see Scheme V.2). In the following sections of this work we refer to these two schemes as the B/FV scheme, namely for the LWE/RLWE variants. The FV scheme is one of the three schemes implemented in the Microsoft's Simple Encrypted Arithmetic Library (SEAL) [83], and it allows modular arithmetic to be performed on encrypted integers. Other libraries implementing the B/FV scheme can be found in Section V-G.

Following this research line, Bajard, Eynard, Hasan and Zucca [84] proposed an optimization, called RNS FV (BEHZ variant), when the ciphertext has large coefficients. It is based on the Residue Number System (RNS) since it uses CRT representation. This variant is improved in a subsequent work by Halevi, Polyakov and Shoup [85] (HPS variant). Both approaches were evaluated in [86] by Al Badawi, Polyakov, Aung, Veeravalli and Rohloff, where the authors show that the HPS variant [85] has better decryption and homomorphic multiplication runtimes with respect to [84]. However, a subsequent note on this work by Bajard, Eynard, Martins, Sousa and Zucca [87] shows that the noise growth for the BEHZ and HPS variants is actually very close, and that HPS provides only slightly better runtime with respect to BEHZ.

Chen and Han [88] improved the bootstrapping technique of both BGV and FV schemes for a large plaintext modulus (i.e. a large prime power). Based on this work, Halevi and Shoup [89] proposed an improved variant for BGV, which was implemented by the same authors in HElib [90]. Another modification of the FV scheme is proposed in [91] by Kim, Laine, Player and Xia. The plaintext space is switched from R_t to $\mathbb{Z}_{b^n+1}$, by means of using the Hoffstein and Silverman trick [92] where the plaintext modulus t is substituted by a polynomial $x - b$, specifically:

$$R_t = \mathbb{Z}[x]/\langle x - b, x^n + 1 \rangle = \mathbb{Z}[x]/\langle x - b, b^n + 1 \rangle \cong \mathbb{Z}_{b^n+1}.$$

A recent generalization of this work is given in [93] where Bootland, Castryck, Iliashenko and Vercauteren proposed a plaintext modulus as $x^m + b$ instead of $x - b$. Both works, [91] and [93], follow a strand of articles ([94]–[96]) studying an encoding method for transforming a real input data (namely, integer, rational or a complex number) into a polynomial, which is an element of the message space (i.e. plaintext) of a RLWE scheme. It is worth highlighting that both [91] and [93] achieve a reduction of the error growth when compared to the original version of the FV scheme and, as a consequence, both of them enable the evaluation of circuits with higher multiplicative depth. The main difference between these two approaches is that whereas [91] encodes fractional numbers, [93] encodes complex numbers.

In BGV and B/FV schemes (and similarly in other FHE schemes) each ciphertext includes an error that grows with each homomorphic operation. To avoid decryption failure, the error must be below a certain threshold. This implies a trade-off between security level and error margin that influences the parameter selection, and that is specific to each use case. Such parameter choice requires a complex study of the error growth, which has motivated some research works. Namely, Costache, Laine and Player [97], extending a previous performance evaluation of Costache and Smart [98], compared the error growth for BGV and FV schemes. Specifically, in [97] the authors proved that, for a particular small plaintext modulus and circuit depth, BGV requires a larger parameter set than FV. On the other hand, BGV outperforms FV when the plaintext modulus is medium or large. However, the analysis performed in [97] does not consider some of the available optimizations for BGV and FV. Mono, Marcolla, Land, Güneysu and Aaraj [99], provided a dynamic (i.e. level dependent) noise estimation following previous works [80], [97], [98] but also considering the new fetures on BGV. Moreover, they provided an easy-to-use interactive parameter generator tool[4].

Kim, Polyakov and Zucca [100] proposed several optimizations to the FV and the BGV schemes and suggested a different approach to compute the ciphertext modulus of the BGV scheme, which does not require dynamic noise estimation, but at the cost of increasing the ciphertext modulus. With these optimizations, their FV variant has better noise growth than BGV for all plaintext moduli. However, their FV variant is faster than BGV only for small plaintexts, while BGV is still faster for intermediate and large plaintexts. The differences regarding the error growth between [97] and [100] can be explained by two factors: i) in [100] the authors performed a static noise estimation whereas in [97] is dynamic; and ii) the analysis of [97] has some inaccuracies when computing the noise growth for FV, as pointed in [100].

Scheme V.2: FV scheme [13]

Let $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$ where d is a power of 2. Let q and p be positive integers, let $\Delta = \lfloor q/p \rfloor$ and $r_t(q) = q \bmod p$. For any natural number t we write $R_t = \mathbb{Z}_t[x]/\langle x^d + 1\rangle$. Let χ be a B-bounded probability distribution over R_q. Then, the FV scheme is constructed as follows:

- *key generation*: takes as input the security parameter λ and outputs the small secret key $\mathsf{sk} = s \in \chi$. It generates $a \in R_q$ uniformly at random and computes $-(a \cdot s + e) \bmod q$ where $e \in \chi$ is a small random error. It outputs sk and $\mathsf{pk} = (p_0, p_1) = (-(a \cdot s + e) \bmod q, a)$.

- *encryption*: takes as input the message $m \in R_p$ and the public key $\mathsf{pk} \in R_q^2$. It chooses at random the values $u, e_1, e_2 \in \chi$ and outputs the ciphertext $\mathbf{c} = (c_0, c_1)$, where $c_0 = (p_0 \cdot u + e_1 + \Delta m) \bmod q$ and $c_1 = (p_1 \cdot u + e_2) \bmod q$.

- *decryption*: takes as input the secret key $s \in R_q$ and the ciphertext $\mathbf{c} \in R_q^2$. It computes $\lfloor \frac{p \cdot (c_0 + c_1 \cdot s) \bmod q}{q} \rceil \bmod p = m$.

- *(homomorphic) properties.* An extensive description can be found in [13, sec. 4]), here we cater for a brief overview. The ciphertext $\mathbf{c}$ can be seen as a polynomial evaluated in s, i.e. $c(s)$, instead of a vector with two components.
  - The addition is trivial: $c_1(s) + c_2(s) \bmod q$
  - The multiplication of two ciphertexts gives as a result a quadratic polynomial

    $$c_1(s) \cdot c_2(s) = \alpha_0 + \alpha_1 \cdot s + \alpha_2 \cdot s^2$$

    which can be transformed into a decryptable ciphertext by means of a *re-linearization* process, which reduces by one the degree of the ciphertext.

[4]https://github.com/Crypto-TII/fhegen

D. *Second Generation:* FHE *based on NTRU*

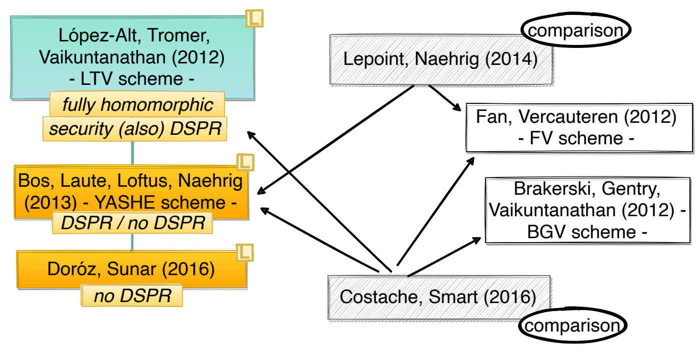

Fig. 7. Main FHE schemes based on NTRU.

NTRU [101] is a lattice-based encryption scheme introduced by Hoffstein, Pipher and Silverman in 1996, with a provisional patent filed, and granted in 2000 [102]. Since its inception, the security of this scheme was under discussion by the research community, until 2011, when Stehlé and Steinfeld [103] slightly modified the scheme to obtain a variant which security is based on the RLWE assumption. One year later, López-Alt, Tromer and Vaikuntanathan [104] introduced the first FHE scheme inspired from the Stehlé-Steinfeld NTRU variant. It is considered part of the second generation of FHE schemes. Specifically, the authors propose a new notion of homomorphic encryption scheme called *Multi-Key* FHE, which supports computation on ciphertexts encrypted under different keys (for more details on Multi-Key FHE see [25]). This scheme, called LTV, uses the bootstrapping and modulus switching techniques and it is constructed as follows:

- *key generation*: chooses two small random polynomials $f', g \in \chi$, where χ is a B-bounded distribution over $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$ and d is a power of 2. It outputs the secret key $f = 2f' + 1 \in R$, where $f \equiv 1 \bmod 2$ and f is invertible in R_q, and the public key $h = 2gf^{-1} \bmod q \in R_q$.

- *encryption*: computes the ciphertext as $c = hs + 2e + m \in R_q$, where $m \in \mathbb{F}_2$ is the message and s and e are random small elements in χ. The ciphertext is an element of $R_q = \mathbb{Z}_q[x]/\langle x^d + 1\rangle$.

- *decryption*: computes $m = (fc \bmod q) \bmod 2$.

The security of this scheme is based on circular security, the RLWE problem and the *Decisional Small Polynomial Ratio* (DSPR) problem. The DSPR problem states that it is hard to distinguish between h (as defined in the scheme construction) and an uniformly random polynomials in R_q. One year later Bos, Laute, Loftus and Naehrig [105] modified the LTV scheme [104] proposing two schemes: i) YASHE (i.e. Yet Another Somewhat Homomorphic Encryption scheme), for which they removed the DSPR assumption using the scale invariant at the cost of having a large evaluation key and a complex key switching method; ii) a YASHE version including again the DSPR assumption to achieve a more practical construction. In 2016, Albrecht, Bai, and Ducas [106] and, in an independent work, Cheon, Jeong and Lee [107], provided a *sub-field lattice* attack which renders any NTRU-like scheme based on the DSPR problem insecure, for some particular parameters choice. Finally, Doröz and Sunar [108], adapted the GSW scheme [57] (see Section V-E) to the NTRU setting by removing the DSPR assumption.

Although NTRU-like schemes are, in general, faster than RLWE schemes, the work by Lepoint and Naehrig [109] showed that the FV scheme [19] has lower error than the YASHE scheme [105]. Also, Kim and Lauter [110] proved that YASHE is better than BGV for a low-degree computation but BGV is more efficient than YASHE for high circuit depths. Moreover, Costache and Smart [98] demonstrated that YASHE is more efficient than BGV for small plaintexts modulus, but BGV is more efficient for large plaintexts. Figure 7 shows the diagram of second generation schemes based on NTRU, including the most relevant papers providing performance comparisons. To conclude this section, it is worth commenting that the parameters choice for NTRU-based schemes is affected by the attacks proposed in [106], [107]. Namely, to obtain a secure NTRU-based scheme the parameters should be significantly increased with respect to the sizes proposed before these attacks were published. This rendered NTRU-based schemes significantly less efficient than its counter-parts, thus they are no longer used nor supported by any library.

E. Third Generation: FHE *based on* LWE *and* RLWE

A second family of (R)LWE schemes (also called third generation FHE, as Peikert notes in [31]) started with the Gentry, Sahai and Waters [57] (GSW) scheme. The GSW scheme proposes a different approach to perform homomorphic operations, introducing the *approximate eigenvector method*, which removes the requirement for key and modulus switching techniques. This new technique reduces the error growth introduced by homomorphic multiplications to a small polynomial-factor. As shown in [111], when multiplying ℓ ciphertexts, all starting with the same error level, the final error grows by a ℓ poly(n) factor, where n is the dimension of the scheme (i.e. the dimension of the lattice). This is a distinctive aspect with respect to previous schemes, such as BGV or FV, for which the final error grows by a quasi-polynomial-factor. The RLWE version of this scheme was given by Khedr, Gulak and Vaikuntanathan in [112]. Figure 8 describes this second family of (R)LWE schemes.

Adopting the notations used in [113] and [60], the (simplified) GSW scheme construction[5] is as follows:

- *key generation*: outputs the secret key $\mathbf{s} = (1, s_2, \ldots, s_n) \in \mathbb{Z}_q^n$, where the s_i's are chosen at random and a public key $A \in \mathbb{Z}_q^{n\times n}$ that is a matrix $n \times n$ such that $A \cdot \mathbf{s} = \mathbf{e} \approx 0$;

- *encryption*: computes the ciphertext $C = m\,\mathrm{I}_n + RA$, where $m \in \mathbb{Z}_q$ is the message, I_n is the identity matrix, R is a random matrix with size $n \times n$ and with coefficients in $\mathbb{F}_2$. This means that the entries of R are *small*;

- *decryption*: First, computes $C\mathbf{s} = m\,\mathrm{I}_n\mathbf{s} + RA\mathbf{s} = m\,\mathrm{I}_n + R\mathbf{e} \approx m\,\mathrm{I}_n s$. Note that because R is small, if $A\mathbf{s} \approx 0$ then $RA\mathbf{s} \approx 0$. Finally, outputs the first element of the vector $x \approx m\,\mathrm{I}_n\mathbf{s} \approx (ms_1, \ldots, ms_n)$, which is m, since $s_1 = 1$.

The main drawbacks of the GSW scheme are the high communication costs (the ciphertext is large with respect to the corresponding plaintext) and the computation complexity. To reduce the computational overhead, various optimizations have been proposed to improve the bootstrapping procedure (see Figure 8). Specifically, Alperin-Sheriff and Peikert [113], [114] suggested a new bootstrapping algorithm considering decryption as an *arithmetic* function instead of a boolean circuit. Hiromasa, Abe and Okamoto [115] optimized the Alperin-Sheriff and Peikert procedure [113] constructing an FHE scheme (based on GSW scheme) that supports homomorphic matrix operations. Moreover, Brakerski and Vaikunthanathan [111], starting from GSW, proposed the first work that managed to get FHE scheme based on GapSVP with polynomial approximation factors (and circular security). Ducas and Micciancio [116] proposed a ring variant, the FHEW scheme, of the Alperin-Sheriff and Peikert (AP) bootstrapping technique [113]. They introduced a new method to homomorphically compute the NAND of two standard LWE ciphertexts (with standard we refer to the ones of Regev's scheme [44]) by evaluating a look-up table during bootstrapping. This technique was later called programmable bootstrapping (PBS) [117].

In this work, they also adopt the complex FFT which enables the implementation of the scheme with the *Fastest Fourier Transform in the West* [118] library (the "W" of the scheme's name FHEW comes from this). These set of optimizations render the GSW's scheme bootstrapping procedure faster than the BGV's scheme one.

In a subsequent paper, Chillotti, Gama, Georgieva and Izabachène [14] improved the Ducas-Micciancio's result and used a different bootstrapping technique, namely, the one proposed by Gama, Izabachène, Nguyen and Xie (GINX) in [119]. The authors proposed three different FHE schemes

[5]Note that, for the sake of clarity, this version is a simplified description of the scheme and it is not homomorphic for multiplication. To get homomorphic multiplications we would also need the bit decomposition.

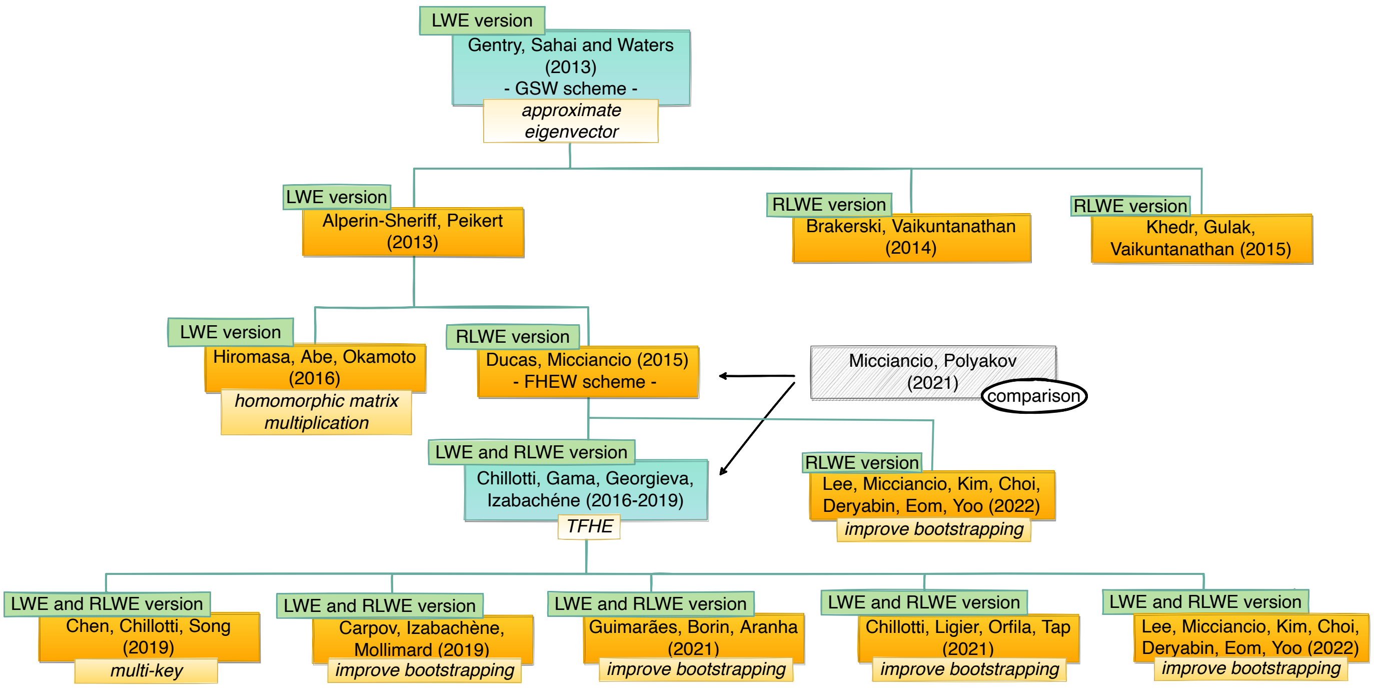

Fig. 8. Main third generation of FHE schemes based on the LWE and RLWE problems.

over the Torus, generally called TFHE: i) TLWE, which is a generalized version of the LWE problem for the Torus; TRLWE, which is its ring variant; and iii) TRGSW, which improves the ring version of GSW scheme. The messages of the TLWE scheme are in the torus $\mathbb{T}$ and the ciphertexts in $\mathbb{T}^{n+1}$, whereas the ring version TRLWE of this scheme works with plaintexts in the R-module, where R is the ring of integer polynomials. Namely, the message space is $T = \mathbb{R}[x]/\langle x^d + 1\rangle \bmod 1$. The TRGSW scheme encrypts elements of the ring of integer polynomials into a vector of TRLWE ciphertexts, namely, $C \in T^{(k+1)\ell}$. It is worth commenting that [14] is an extended and improved version of previous articles presented in Asiacrypt 2016 [120], where they speed up the bootstrapping procedure and reduce the bootstrapping key size compared to FHEW; and Asiacrypt 2017 [121], where they improve the leveled version of TFHE.

A detailed description of the TRLWE scheme is provided in Scheme V.3.

Scheme V.3: TRLWE scheme [14]

Let us consider the following notations: $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$, where d is a power of 2, $T = \mathbb{R}[x]/\langle x^d + 1\rangle \bmod 1$ and $R_2 = \mathbb{F}_2[x]/\langle x^d + 1\rangle$, that is, any element in R_2 is a polynomial in R with binary coefficients (in a recent work [117], the authors used $\mathbb{Z}_q$ instead of a torus). Then, the TRLWE scheme is constructed as follows:

- *key generation*: takes as input the security parameter λ and outputs the small secret key $\mathbf{s} \in R_2^n$.
- *encryption*: takes as input the secret key $\mathbf{s} \in R_2^n$, the error parameter α, and the message $m \in T$. Then, it chooses a uniformly random mask $\mathbf{a} \in T^n$, and a small error $e \in \chi$, where χ is a B-bounded distribution. Then it outputs the ciphertext.
$$\mathbf{c} := (\mathbf{a}, \mathbf{s}\cdot\mathbf{a} + m + e) \in T^n \times T.$$
- *decryption*: takes as input the secret key $\mathbf{s} \in R_2^n$ and the ciphertext $\mathbf{c} \in T^{n+1}$. Then it computes the secret linear κ-Lipschitz function $\varphi_{\mathbf{s}}$ (called *phase*) of the ciphertext $\mathbf{c}$. The phase $\varphi_{\mathbf{s}} : T^n \times T \to T$ is such that $\varphi_{\mathbf{s}}(\mathbf{a}, b) = b - \mathbf{s}\cdot\mathbf{a}$. Note that this function is parametrized by the secret key $\mathbf{s} \in R_2^n$. The phase $\varphi_{\mathbf{s}}(\mathbf{c})$ is close to the actual message:
$$\varphi_{\mathbf{s}}(\mathbf{c}) = \mathbf{s}\cdot\mathbf{a} + m + e - \mathbf{s}\cdot\mathbf{a} = m + e.$$
To conclude, it rounds $\varphi_{\mathbf{s}}(\mathbf{c})$ to the nearest point in the message space $M \subset T$.
- *(homomorphic) linear combinations of ciphertexts*. Let $\mathbf{c}_1, \ldots, \mathbf{c}_p$ be p independent ciphertexts under the same key $\mathbf{s}$, and let $f_1, \ldots, f_p$ be integer polynomials in R.
We consider $\mathbf{c} = \sum_{i=1}^{p} f_i \cdot \mathbf{c}_i$ such that the error amplitude remains smaller than $1/4$, that

is, $||e||_\infty \leq 1/4$. Then by [14, Fact 3.5], $\mathbf{c}$ is a ciphertext and

$$\mathsf{Dec}_{\mathbf{s}}(\mathbf{c}) = \sum_{i=1}^{p} f_i \cdot \mathsf{Dec}_{\mathbf{s}}(\mathbf{c}_i).$$

As the authors of [14] pointed out, the ciphertexts can be linearly combined to obtain a new ciphertext which is the linear combination of the messages. But when we have to manipulate the ciphertext non-linearly, TLWE seems to miss some properties. In order to avoid this problem, the authors of [14] proposed the generalized scale invariant version of GSW, called TRGSW. Note that to "switch" from TRLWE to TLWE (and vice-versa) we only have to consider the real torus instead of T, $\{0, 1\}$ instead of B and $\mathbb{Z}$ instead of R.

In [122], Micciancio and Polyakov compared FHEW and TFHE schemes, and proved that the performance difference is determined by the different bootstrapping algorithms adopted in both schemes, AP [113] for FHEW and GINX [119] for TFHE. Specifically, TFHE is faster than FHEW for a binary secret, whereas for higher secret sizes (above ternary) FHEW outperforms TFHE in running time. In terms of memory, TFHE has a bootstrapping key smaller than FHEW. However, it is worth commenting that in a very recent paper [123], Lee, Micciancio, Kim, Choi, Deryabin, Eom, and Yoo introduced a new bootstrapping procedure that achieves the advantages of both algorithms, AP and GINX. The new method supports arbitrary secret key distributions without increasing the running time (like AP/FHEW) and it also achieves a considerable small bootstrapping key size (like GINX/TFHE).

A relevant feature of the TFHE scheme is that the bootstrapping technique enables an univariate function to be evaluated simultaneously to the noise reduction operation [117] (i.e. programmable bootstrapping). Several optimizations have been proposed to improve the TFHE scheme and, in particular, its programmable bootstrapping procedure. Namely, in [124], Carpov, Izabachène and Mollimard proposed a multi-output version of the PBS, that is, a new technique to perform several homomorphic operations over different variables with a single bootstrapping execution. This construction can also be used to evaluate homomorphically several functions over the same encrypted message simultaneously. In a recent work, Guimarães, Borin and Aranha [125] optimized the bootstrapping procedure to evaluate multiple functions on large ciphertexts and Chillotti, Ligier, Orfila and Tap [117] proposed several enhancements, including a generalized method to evaluate several functions at once without adding additional error. Finally, in [126], Chen, Chillotti and Song proposed a multi-key homomophic encryption scheme from TFHE. They provide two methods to multiply a ciphertext encrypted with a single key by a ciphertext encrypted with multiple keys. To conclude this section, we would like to refer interested readers and TFHE practitioners to the recently published guide of Joye [127], which presents implementation details, theoretical examples and a clear description of the programmable bootstrapping technique.

F. *Fourth Generation:* FHE *based on* LWE *and* RLWE

In 2017 a new generation of FHE schemes (see Figure 9) was introduced by Cheon, Kim, Kim and Song [15]. The authors proposed a method to construct a leveled Homomorphic Encryption scheme for Approximate Arithmetic Numbers, and included an open-source library implementing the scheme. This scheme was initially called HEAAN, but nowadays research community refers to the scheme as CKKS (from the authors' names) while HEAAN is used to denote the library. The scheme construction is described in Scheme V.4. The scheme was extended one year later to a fully homomorphic encryption scheme by Cheon, Han, Kim, Kim and Song [128], and, subsequently [129] the same authors presented a variant of CKKS scheme by means of including a ciphertext packing technique based on the CRT. In [130] Boemer, Costache, Cammarota and Wierzynski introduced several optimizations, based on complex packing, to improve the runtime of scalar encoding and of ciphertext-plaintext addition and multiplication operations. Moreover, a different kind of complex packing was introduced by Kim and Song in [131]. In [132], Kim, Papadimitriou and Polyakov improved the usability of CKKS and its RNS variant proposing a new technique that minimizes the error during computation. Specifically, the idea is to rescale the ciphertext before multiplication and not after, thus obtaining a smaller error before performing the multiplication. Also, the bootstrapping version [128] of CKKS was enhanced in [133] by Chen, Chillotti and Song, whereas, in [134], Han and Ki discussed and improved the bootstrapping version of [129]. The bootstrapping version of [128] includes an homomorphic modular reduction, which is approximated by a trigonometric function to improve efficiency. Parallel works improved this approximation, such as Lee, Lee, Lee, Kim and No [135], who improved the bootstrapping of the RNS-CKKS leveraging the technique proposed in [132]. Also, Jutla and Manohar [136] proposed a sine series to approximate the modular reduction, and achieved a significantly higher precision than the previous works. It is also worth mentioning other works that approximate the modular reduction without relying on trigonometric functions, such as Jutla and Manohar in [137] and Lee, Lee, Kim, and No in [138]. Bossuat, Mouchet, Troncoso-Pastoriza and Hubaux [139] proposed the most efficient RNS-CKKS bootstrapping implementation, and the first practical instance of a bootstrapping algorithm with a dense secret. The problematic of performing bootstrapping with a dense secret is that the bootstrapping circuit depth increases, as well as the bootstrapping failure probability. On the other hand, adopting a sparse secret makes bootstrapping more efficient but renders the scheme less secure due to some recent attacks (see Li-Micciancio attack in the next paragraph and Section VI). Motivated by this trade-off Bossuat, Troncoso-Pastoriza and Hubaux [140] proposed a sparse-secret encapsulation technique, which overcomes the security vulnerability problem of sparse secrets while preserving a negligible bootstrapping failure probability. However, the bootstrapping technique in [139] is still faster than [140], at the cost of higher failure probability and a slightly less precision.

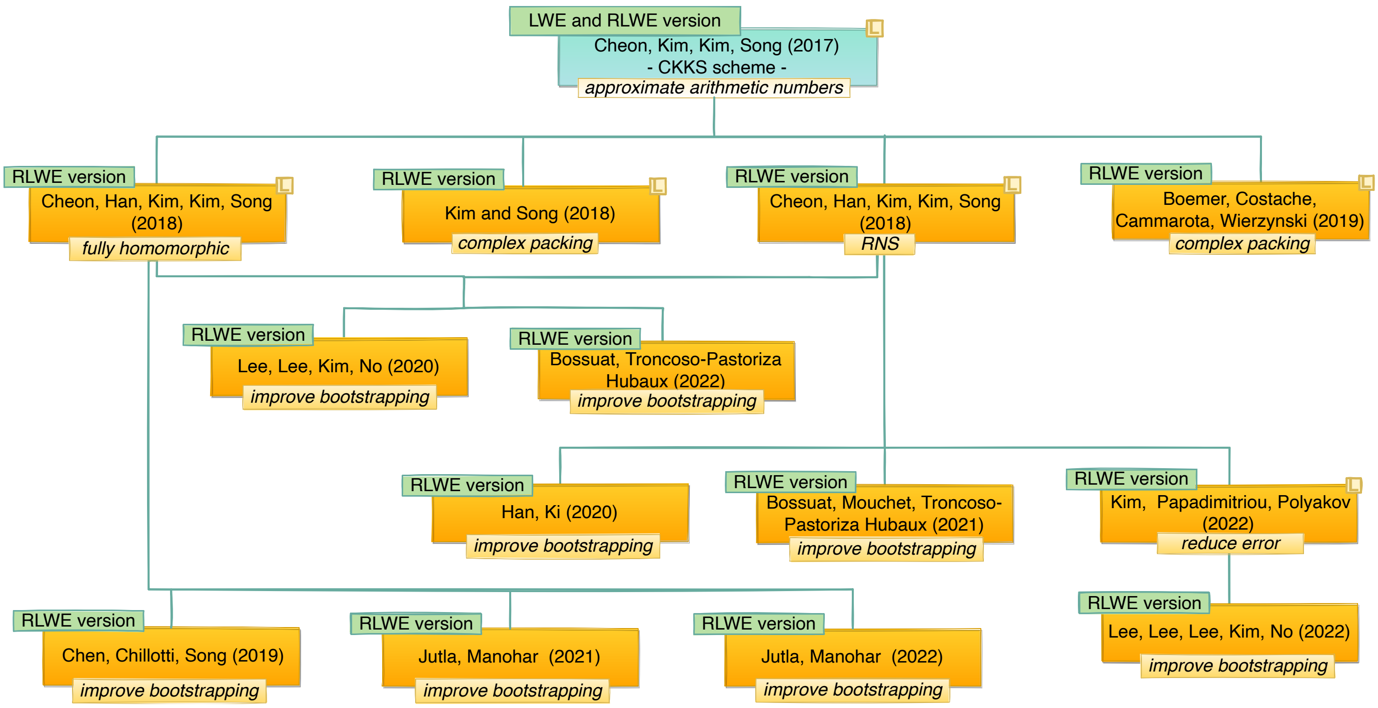

Fig. 9. Main fourth generation FHE schemes based on the LWE and RLWE problems.

In a recent work, Li and Micciancio [141] presented an attack against CKKS that works for a specific application scenario. The authors proved that the CKKS scheme (as well as all improvements and implementations of approximate encryption schemes) is vulnerable to an attack by an adversary that has access to the encryption functionality. Namely, the adversary can extract a secret key using only linear algebra or lattice reduction techniques. Specifically, the secret key can be obtained if the decrypted element and corresponding ciphertext are both known, because the error is a linear combination of the secret key's elements (see Scheme V.4). However, as Cheon, Hong and Kim pointed out in [142], this attack can be prevented if the owner of the secret key does not share the result of the decrypted messages. Unfortunately, some applications such as secure multi-party computation or differential privacy techniques require sharing some plaintexts. In such cases, Li and Micciancio suggested that their attack could be probably avoided by modifying the decryption function by means of adding an error at the end of the decryption process, as suggested in [141].

Scheme V.4: CKKS scheme [15]

Let $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$ and $d = 2^M$. For a base p, a modulus q_0 and a natural integer L (chosen level), let $q_\ell = p^\ell \cdot q_0$ for $\ell = 1, \ldots, L$. Note that a ciphertext of level ℓ is a vector in R_{q_ℓ}. Let us consider the following relevant distributions. For a real number σ, $DG(\sigma^2)$ is a vectorial discrete Gaussian distribution over $\mathbb{Z}^d$, which samples each of its components from independent discrete Gaussian distributions of variance σ^2. For a real number $0 < \rho < 1$, the distribution $ZO(\rho)$ is the distribution over $\{-1, 0, 1\}^d$ which draws 0 with probability $1-\rho$ and -1 or 1 with probability $\rho/2$. Finally, let us consider χ a B-*bounded* distribution.
The leveled CKKS scheme is constructed as follows:

- *key generation*: takes as input the security parameter λ and chooses M, integers h and t, and a real number σ so that the best attack against the associated RLWE instance achieves a complexity 2^λ. It computes a secret key $\mathsf{sk} = (1, s)$, where $s \in \chi$. It generates $a \in R_{q_L}$ uniformly at random and computes $-as + e \bmod q_L$, where e is $DG(\sigma^2)$. Finally, it samples $a' \in R_{t \cdot q_L}$ and $e' \in DG(\sigma^2)$ and computes $b' = -as + e' + ts' \bmod t \cdot q_L$. It outputs the secret key $\mathsf{sk} = (1, s)$, the public key $\mathsf{pk} = (b, a)$ and the evaluation key $\mathsf{evk} = (b', a')$.
- *encryption*: takes as input the message $m \in R$ and the public key $\mathsf{pk} \in R^2_{q_L}$. It chooses at random the values $v \in ZO(1/2)$ and $e_0, e_1 \in DG(\sigma^2)$ and outputs

$$c = (\beta, \alpha) = v\mathsf{pk} + (m + e_0, e_1) \bmod q_\ell.$$

- *decryption*: takes as input the secret key $\mathsf{sk} = (1, s)$ and the ciphertext $\mathbf{c} \in R^2_{q_\ell}$. It computes $m = \langle \mathbf{c}, \mathsf{sk}\rangle \bmod q_\ell = \beta + \alpha s \bmod q_\ell$.
- *(homomorphic) properties.*
  - The addition is trivial: $\mathbf{c}_1 + \mathbf{c}_2$

– The multiplication of two ciphertexts $\mathbf{c}_i = (\beta_i, \alpha_i)$ with $i = 1, 2$ is

$$\mathbf{c}_1 \cdot \mathbf{c}_2 = (d_0, d_1) + \lfloor t^{-1} d_2 \mathsf{evk} \rceil \bmod q_\ell,$$

where $(d_0, d_1, d_2) = (\beta_1\beta_2,\ \alpha_1\beta_2 + \alpha_2\beta_1,\ \alpha_1\alpha_2) \bmod q_\ell$.

It is worth clarifying that when we have two ciphertexts $\mathbf{c}, \mathbf{c}'$ of two different levels $\ell' < \ell$, we should reduce the level of the ciphertext with a higher level to match both levels, i.e. $\ell' = \ell$. This can be achieved with a *rescaling* procedure that takes a ciphertext $\mathbf{c} \in R^2_{q_\ell}$ at level ℓ and outputs $\mathbf{c}' = \lfloor q_{\ell'}/q_\ell \rceil \bmod q_{\ell'}$.

An interesting feature of CKKS is that the message space can be represented as elements in the extension field $\mathbb{C}$. Informally, the message m can be embedded in $S = \mathbb{R}[x]/\langle x^d + 1 \rangle$. Since, the roots of $x^d + 1$ are the complex primitive roots of unity in $\mathbb{C}$, to convert the $m \in S$ into a vector of complex numbers it is sufficient to evaluate it at these complex roots. For more details see [15, sec. 3].

The fourth generation schemes are similar to the second generation. The main difference is that the fourth generation schemes are approximate schemes, i.e. they use approximate computation, which is considerably faster. Specifically, the fourth generation embeds the message space into a complex hyperplane, and the error during encryption is inserted as part of the approximation error that is inherently introduced during a computation over real-valued numbers. An interesting feature of CKKS is the capability to homomorphically operate over approximations of real numbers, which makes it a suitable scheme to work with floating-point arithmetic. Another similarity is that the schemes from both generations have efficient packing techniques, and they can only compute fast sum and product (any non-linear operation becomes computationally expensive).

It is important to highlight the work proposed by Boura, Gama, Georgieva and Jetchev [143], the CHIMERA scheme, which is a hybrid solution combining three RLWE-based FHE schemes: TFHE [14], B/FV [13], [50] and CKKS [128]. CHIMERA has the special property that it enables the switching between the three schemes. The authors start by defining a common plaintext space between the three schemes by constructing an embedding[6] of the different message spaces. By leveraging the bootstrapping technique, CHIMERA enables switching ciphertexts from TFHE to FV (and vice-versa) and from CKKS to FV (and vice-versa). FV must be used as intermediate step for transformations between TFHE and CKKS. CHIMERA was first presented as a solution to the Idash'18 Track 2 competition [144], [124], and it was later improved in PEGASUS [145] by Lu, Huang, Hong, Ma, and Qu.

[6]An instance of a mathematical structure that is contained within another instance

G. Final Considerations

To the best of the authors knowledge, BGV [12], B/FV [13], TFHE [14] and CKKS [128] are the most practical and widely adopted schemes. The second generation schemes, BGV and B/FV, are suitable to work with finite fields in the modular exact arithmetic. They are equipped with efficient packing which enables the use of SIMD (namely, single instruction multiple data) instructions to perform computations over vectors of integers (i.e. batching). Thus, these schemes are excellent candidates when large arrays of numbers are to be processed simultaneously.

Second generation schemes are not good candidates for circuits where bootstrapping is required (i.e. circuits with large multiplicative depth), or where non-linear functions are to be implemented. Third generation schemes should be adopted instead, namely TFHE, which can outperform previous schemes for bit-wise operations, i.e. when computations are expressed as boolean circuits [58]. The main limitation of TFHE is the lack of support for CRT packing (i.e. batching), hence the scheme can be outperformed by previous approaches when processing large amounts of data simultaneously. The fourth generation, i.e. CKKS, is the best option for real numbers arithmetic. Table I provides a comparison among the schemes' families, and Figure 10 depicts the main applications for each generation of schemes. It is worth clarifying that, although TFHE provides the fastest bootstrapping procedure, the batching feature of 2nd and 4th generation schemes allows for the parallel bootstrapping of several plaintexts. For the specific case of CKKS, it is possible to obtain a more efficient amortized bootstrapping than for TFHE (this special case has been reflected in Table I with the ● symbol). This is however not true for 2nd generation schemes because the number of slots is significantly lower than for CKKS. For example, in BGV the number of slots is only about 1000 as compared to 2^{15} or so for CKKS. This renders CKKS bootstrapping more than one order of magnitude (often two) faster than BGV bootstrapping.

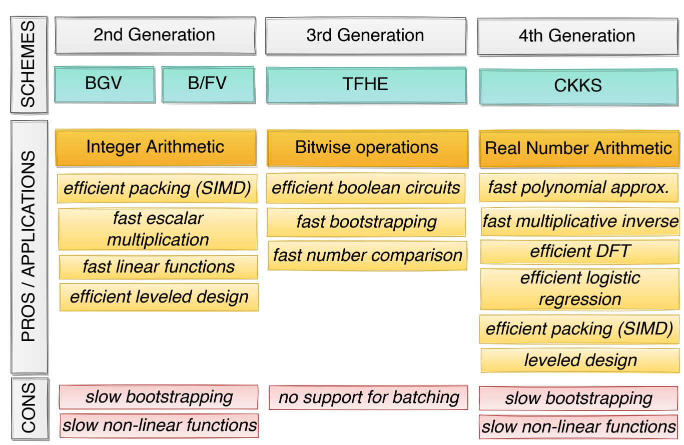

Fig. 10. Pros/cons of FHE schemes by generation.

H. Other Works

In a recent work by Doröz, Hoffstein, Pipher, Silverman, Sunar, White and Zhang [146], an FHE scheme based on a

TABLE I
PROPERTIES OF THE MOST WIDELY ADOPTED FHE SCHEMES.

Scheme	Fast operations			Fast packing/batching	Leveled design	Fast bootstrapping
	scalar mult	arithmetic	non arithmetic			
Second Generation (e.g. BGV, B/FV)	●	●	○	●	●	○
Third Generation (e.g. FHEW, TFHE)	●	●	●	○	●	●
Fourth Generation (e.g. CKKS)	●	●	○	●	●	● (grey)

new hard problem was introduced: *Finite Field Isomorphism Problems*, that is based on the difficulty of recovering a secret isomorphism between two finite fields. Moreover, in 2019 Joux [147] proposed a scheme whose techniques are similar to those of (R)LWE, but with arithmetic modulo large *Fermat numbers*, namely numbers given by the expression $F = 2^{2^f} + 1$, where $f \in \mathbb{N}$.

VI. SECURITY OF SCHEMES BASED ON (R)LWE PROBLEMS

As described in Section III-G, the LWE problem consists of finding the secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, given $\mathbf{b} \in \mathbb{Z}_q^m$ and $A \in (\mathbb{Z}_q)^{m\times n}$ such that $A\mathbf{s} + \mathbf{e} = \mathbf{b} \bmod q$, where $\mathbf{e} \in \mathbb{Z}_q^m$ is sampled from the error distribution χ. The security of LWE-based schemes depends on the intractability of this problem, and attacks on these schemes are based on finding efficient algorithms to solve them. In this framework, Albrecht, Player and Scott [148], presented three different methodologies to solve the LWE-problem (Figure 11): i) based on the BDD problem (Section VI-B); ii) based on the SIS problem (Section VI-C); and iii) a direct search of the secret $\mathbf{s}$ (Section VI-D). Conceptually, the central part of the first and the second methodology is based on a lattice reduction. Namely, starting from a bad (i.e. long) lattice basis, find a better (i.e. reduced and more orthogonal) basis. Note that, in [148], the authors showed that there is no single-best attack against all possible parameters.

Regarding the schemes based on the RLWE problem, the same considerations apply. According to the Homomorphic Encryption Security Standard [58], if we choose correctly the error distribution then there are no better attacks on RLWE than on LWE. This is because the best known attacks do not leverage any property of the ring structure. In this claim, the correct choice of the error distribution only refers to a sufficiently *well spread* distribution [149]–[151] (Section VI-E).

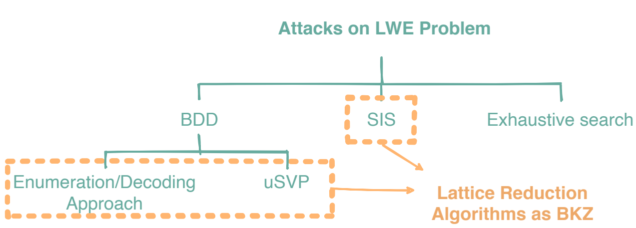

Fig. 11. Different solving approach to LWE problem

A comprehensive explanation of lattice attacks can be found in [148], [152]–[154]. Moreover, the work of Bindel, Buchmann, Göpfert and Schmidt [155] extends previous studies with an analysis of how the number of samples affects the hardness of LWE. It is worth highlighting the work of Albrecht, Player and Scott [148], which not only describes in detail LWE attacks, but also provides a software tool to determine the security level of LWE instances. The first version of this tool is referred to as the LWE Estimator[7], and has been adopted in the Homomorphic Encryption Security Standard [58] to provide specific parameters for FHE schemes. The new version, denoted by Lattice Estimator[8], *"was born out of frustration with the limitations of the old codebase"* as Albrecht mentioned in his blog [156]. Figure 12 shows the timeline and fundamentals of the main lattice attacks proposed until the present time.

A. Lattice Reduction Algorithms

The most well-known lattice reduction algorithm used in practice is BKZ (block Korkin-Zolotarev reduction) due to Schnorr and Euchner [157]. This is a block-wise iterative algorithm for basis reduction that can be seen as a generalization of the LLL algorithm introduced in 1982 by Lenstra, Lenstra and Lovász [158]. Recently, several variants of the BKZ algorithm have been proposed [159]–[163]. In these algorithms, the time complexity and the outcome quality (i.e. the orthogonality of the reduced basis) is characterised by the Hermite factor [164] and is given as a trade-off. Specifically, the run time of the BKZ algorithm is higher when the root-Hermite factor δ_0 is smaller [157]. This is also shown in Lindner and Peikert's estimation [165] (Figure 12). This is result is also supported by a more realistic estimation[9] provided in [148]. Starting from the data provided by Liu and Nguyen [166], a similar relation between δ_0 and the run time of the BKZ 2.0 algorithm [159] was found by Albrecht, Cid, Faugére, Fitzpatrick and Perret [167] (Figure 12). It is also worth commenting that the quality of the output decreases with higher values of δ_0. The formula linking the root-Hermite factor attainable by BKZ and the block-size b of this algorithm was heuristic established (and well-verified experimentally) by Chen in his PhD thesis [168]:

$$\delta_0 = \left((\pi b)^{\frac{1}{b}} \cdot \frac{b}{2\pi e} \right)^{\frac{1}{2(b-1)}}.$$

For more details about lattice reduction algorithms we refer the readers to [148] by Albrecht, Player and Scott, and the survey on algorithms for the SVP and CVP by Hanrot, Pujol and Stehlé [169].

[7] https://bitbucket.org/malb/lwe-estimator
[8] https://github.com/malb/lattice-estimator
[9] As Albrecht pointed out [152] the LP model for estimating the cost of lattice-reduction is not correct for several reasons and the formula proposed by Lindner and Peikert turns out to be too optimistic.

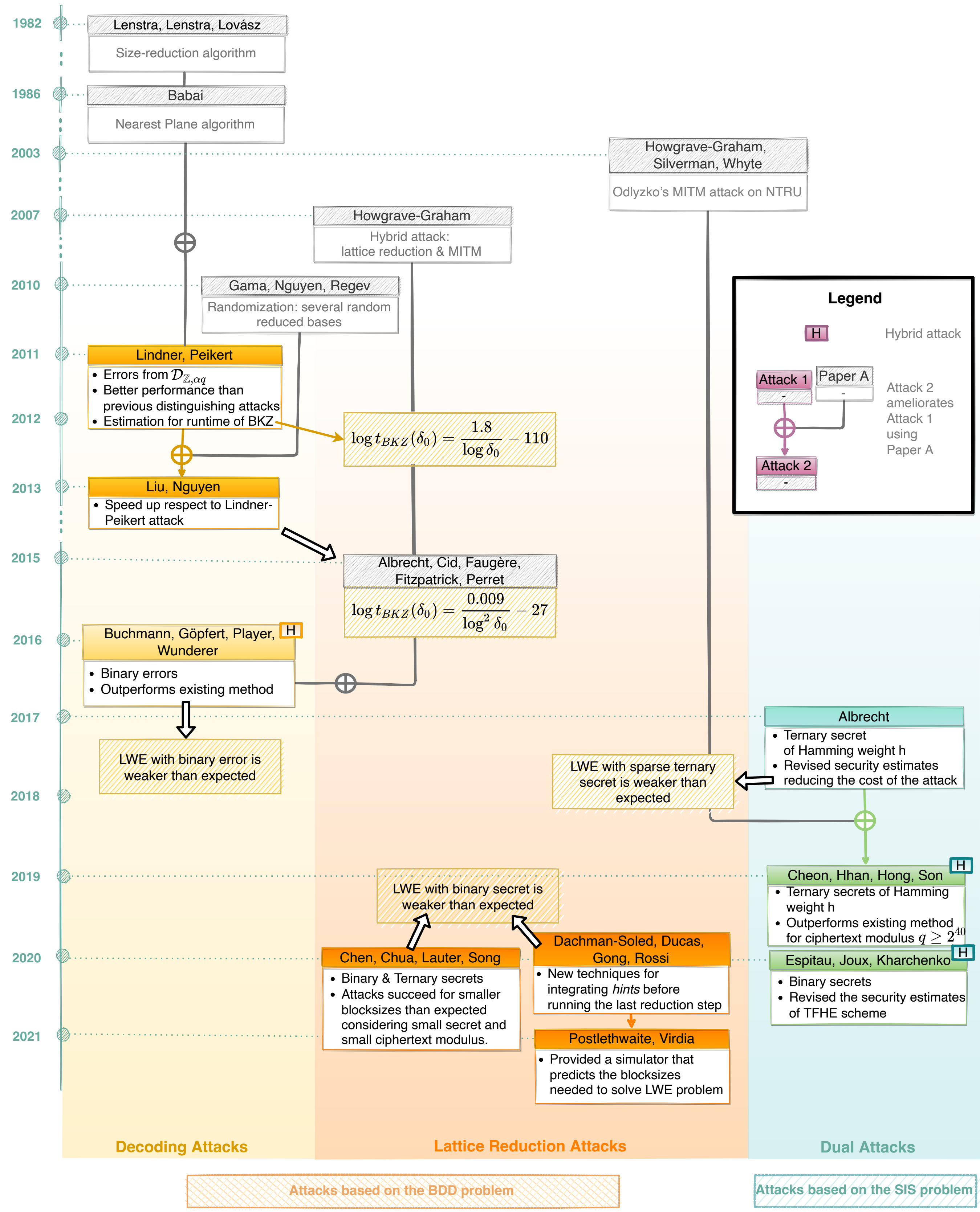

Fig. 12. Timeline of the main attacks on the LWE problem.

B. Attacks Based on Bounded Distance Decoding Problem

These attacks are based on solving LWE by solving the BBD problem (Section III-D). Specifically, the main strategy of this kind of attacks consists in finding $\mathbf{v}$, the closest vector to $A\mathbf{s}+\mathbf{e}$, for a lattice $\mathcal{L}=\mathcal{L}(A)$. Note that knowing $\mathbf{v}$, which equals $A\mathbf{s}$, we can obtain $\mathbf{s}$ and thus solve the LWE problem. The following strategies have been proposed:

1) BDD enumeration (decoding): This attack was proposed by Lindner and Peikert [165], who modified the *Nearest Plane* algorithm analyzed by Babai [170], using multiple planes to decrease the failure probability of finding the vector $A^t\mathbf{s}$. The Lindner-Peikert algorithm has two main steps. First, it applies lattice reduction (using BKZ) to obtain a new basis $\{\beta_1,\ldots,\beta_n\}$. Second, it performs a recursive search for an integer combination of the basis vectors β_i that is close to the target vector $\mathbf{v}$. In 2013, Liu and Nguyen [166], starting from the paper by Gama, Nguyen and Regev [171], ameliorated the Lindner-Peikert algorithm as well as Babai's.

Subsequently, Buchmann, Göpfert, Player and Wunderer [172] provided a hybrid attack for the LWE setting, following the approach of Howgrave-Graham [173], who combined the lattice reduction and the meet-in-the-middle (MITM) attacks. The authors showed that, for specific parameters and in the binary error setting (i.e. the errors are random vectors in $\{0,1\}^m$) their attack surpasses previous attacks on LWE. We refer to Wunderer's PhD thesis for the analysis of the hybrid decoding attack [174].

2) Reduction of BDD problem to uSVP: The attacks based on uSVP try to solve the LWE problem by constructing an integer embedding lattice using either the Bai-Galbraith [175] or Kannan [176] technique. The main idea is to embed the lattice $\mathcal{L}(A)=\{A\mathbf{y}\,:\,\mathbf{y}\in\mathbb{Z}_q^n\}$ into a higher-dimensional lattice $\mathcal{L}(A')$, where $A'=(A|I_m|-\mathbf{b})$ and

$$\mathcal{L}(A')=\{\mathbf{x}\in\mathbb{Z}_q^{n+m+1}\,:\,A'\mathbf{x}=\mathbf{0}\bmod q\}.$$

This new lattice $\mathcal{L}(A')$ has dimension $d=n+m+1$ (note that the dimension of $\mathcal{L}(A)$ is n) and a unique shortest vector $\mathbf{v}=(\mathbf{s},\mathbf{e},1)$, where $\mathbf{e}$ is the error and $\mathbf{s}$ is the secret of the LWE instance $(A,A\mathbf{s}+\mathbf{e})$. Thus, finding the shortest vector $\mathbf{v}$ in $\mathcal{L}(A')$ is equivalent to solving the LWE problem. The advantage of adopting the lattice $\mathcal{L}(A')$ is that finding the shortest vector in $\mathcal{L}(A')$ is tractable, whereas in $\mathcal{L}(A)$ it is not since we do not know whether $A\mathbf{s}$ is a shortest vector in $\mathcal{L}(A)$.

There are two methods to estimate the cost for solving LWE using the uSVP strategy. The first one is proposed by Gama and Nguyen [164] (called *2008 estimate*) and later updated by Albrecht, Fitzpatrick and Göpfert [177]. The second method (called *2016 estimate*) is given in [178] by Alkim, Ducas, Pöppelmann and Schwabe, where the authors predicted that $\mathbf{e}$ can be found if

$$\sigma\sqrt{b}\leq\delta_0^{2b-d-1}\cdot q^{m/d},$$

where σ is the standard deviation of the error distribution, b is the block size of the underlying lattice reduction algorithm and δ_0 is the root-Hermite factor (Section III-A). After that, Albrecht, Göpfert, Virdia and Wunderer [179] compared these two estimates and verified experimentally the prediction of [178] when the error vector was sampled coefficient-wise from a discrete Gaussian distribution. In 2019, Bai, Miller and Wen [180] revisited the previous analysis of [178], [179], and provided experiments on estimating the cost of solving LWE via the uSVP suggesting that the 2016 estimate has higher accuracy than the 2008 estimate.

In a recent work, Dachman-Soled, Ducas, Gong and Rossi [181] generalized the uSVP attack and proved that the predictions of [178], [179] are not accurate for small block sizes (i.e. $b\leq 30$). In a parallel work, Chen, Chua, Lauter and Song [182] showed similar results. Namely, they found that for settings with a small error and secret values, i.e. sampled from a binary and ternary distributions, the 2008 and 2016 estimations [178] are optimistic for a small block size (i.e. $b\leq 45$). These studies show that the security levels for small block sizes are smaller than initially described by the online LWE Estimator [148]. However, it is important to mention that as Dachman-Soled, Ducas, Gong and Rossi [181] confirm, small block sizes are not relevant. In 2021, Postlethwaite and Virdia [183] improved the result of [181] and provided a simulator that predicts the block sizes needed to solve uSVP instances via lattice reduction.

C. Attacks Based on the Short Integer Solution Problem

The attacks based on the dual strategy (also called *dual attack*) consist in solving the LWE problem via the SIS strategy (Section III-F), namely, of finding a short vector in the scaled dual lattice $\mathcal{L}_q^{\perp}=\{\mathbf{x}\in\mathbb{Z}_q^m\,:\,\mathbf{x}A\equiv 0\bmod q\}$. Note that this problem is equivalent to solve the Decision-LWE problem. Indeed, given LWE samples $(A,\mathbf{b})$, we can decide whether $\mathbf{b}=A\mathbf{s}+e$ or $\mathbf{b}$ is uniformly random by computing $\langle\mathbf{v},\mathbf{b}\rangle$ where $\mathbf{v}$ is the short vector in the lattice $\mathcal{L}_q^{\perp}$. In fact, if $\mathbf{b}$ is *not* random, i.e. $\mathbf{b}=A\mathbf{s}+\mathbf{e}$, we have

$$\langle\mathbf{v},\mathbf{b}\rangle=\langle\mathbf{v}A,\mathbf{s}\rangle+\langle\mathbf{v},\mathbf{e}\rangle\equiv\langle\mathbf{v},\mathbf{e}\rangle\bmod q.$$

Since $\langle\mathbf{v},\mathbf{e}\rangle$ is short (i.e. $\mathbf{v}$ and $\mathbf{e}$ are sufficiently short), the adversary has to check if $\langle\mathbf{v},\mathbf{b}\rangle$ is close to zero modulo q.

The advantage of distinguishing $\langle\mathbf{v},\mathbf{e}\rangle$ from random, computed by Lindner and Peikert in [165], is close to

$$e^{-\pi(||\mathbf{v}||\alpha)^2},$$

where α is given by the Gaussian distribution χ, namely, αq is the width parameter of χ (Section II-B). To produce such a short $\mathbf{v}$ we require a lattice reduction algorithm. Note that, the outcome of the lattice reduction is a vector $||\mathbf{v}||\approx\delta_0^m q^{n/m}$, but δ_0 depends on the algorithm used, and Micciancio and Regev [29] showed that the minimum for $f(m)=\delta_0^m q^{n/m}$ is obtained when $m=\sqrt{n\log q/\log\delta_0}$.

Albrecht in [152] presented a variant for the dual attack taking into consideration small and sparse secrets. Also, Cheon, Hhan, Hong and Son [184], proposed a new hybrid attack combining the dual attack of Albrecht [152] and the MITM attack on NTRU by Howgrave-Graham, Silverman and Whyte [185]. This hybrid attack outperforms the dual attack for some specific parameter sets of the homomorphic encryption scheme, namely, for sparse ternary secrets, but it was extended by Espitau, Joux and Kharchenko [186] to binary secrets as well.

D. Exhaustive Search on Secret Key $\mathbf{s}$

This strategy consists of directly finding $\mathbf{s}$ such that $||A\mathbf{s} - \mathbf{b}||$ is small. This can be achieved by performing the Arora-Ge algorithm [187]. This algorithm uses a linearization technique that mainly consists of adding new variables in the system to transform non-linear into linear equations. It also adopts the assumption that the error lies in a fix range. The Arora-Ge algorithm solves the LWE in time $2^{\widetilde{O}(n^{2\varepsilon})}$, where ε is such that $\alpha q = n^{\varepsilon}$ and αq is the width parameter of the Gaussian distribution χ (Section II-B).

E. Attacks on the RLWE Problem

As previously mentioned, RLWE-based schemes are, for known attacks, equally secure as the LWE version when the error distribution is correctly chosen. However, there are known examples of error distributions that are insecure for certain rings. In 2015, Elias, Lauter, Ozman and Stange [188] provided an attack on the decision version of the RLWE problem for two specific families of polynomial functions (namely, the definition of the polynomial considers the ciphertext modulus of the scheme). In [189], Chen, Lauter and Stange generalized this attack to certain Galois number fields and defined a new solution for the RLWE problem. These papers were later improved by Chen, Lauter and Stange in [190] and by Castryck, Iliashenko and Vercauteren in [149], [150].

Note that in the Homomorphic Encryption Security Standard paper [58], the authors provide secure parameters for RLWE schemes over power-of-two cyclotomic rings. On the other hand, for generic cyclotomic rings, Ducas and Durmus [54], Lyubashevsky, Peikert and Regev [53], [191] and Crockett and Peikert [192] investigated the types of the error distribution and proposed different ways of choosing a safe error polynomial.

F. Concrete Parameters

Finding an optimal set of parameters for a specific FHE scheme is challenging since it is function dependent. For example, for the second generation schemes, the complexity (i.e. depth) of the function to be homomorphically evaluated impacts the error growth. Higher depths, require higher ciphertext modulus q, and the adoption of a higher modulus decreases the security level. The security level can be increased by adopting a higher polynomial degree, but this impacts efficiency. Some works [97]–[99] have proposed theoretical bounds for error growth estimation, which can be used to obtain the parameters heuristically. However, these works are too conservative with respect to the parameters used in practice. The reason for this is that these theoretical bounds seek for very low failure probability (e.g., less than 2^{-55}), whereas in practical scenarios smaller values are still probabilistically acceptable. The main open problem in the field of parameter selection is that there is a significant gap between the parameters obtained theoretically using the previously proposed heuristics and the parameters used in practice and obtained in a trial an error fashion [97].

Homomorphic Encryption Security Standard [58] presents some recommended (and conservative) parameters for FHE schemes, following the LWE Estimator [148]. Specifically, starting from the dimension $n = 2^k$ (with $k = \{10, \ldots, 15\}$), the authors of [58] cater for recommended values of the q, for a given security level $\lambda \in \{128, 192, 256\}$. In this standard, the error follows a discrete Gaussian distribution with standard deviation $\sigma \approx 3.2$, whereas the distribution for the secret key can be:

- uniform ternary, i.e. the secret $\mathbf{s}$ is chosen uniformly at random from $\{-1, 0, 1\}^n$;
- uniform, i.e. the secret $\mathbf{s}$ is chosen uniformly at random from $\mathbb{Z}_q^n$:
- Gaussian with $\sigma \approx 3.2$, the same as the error distribution.

It is worth commenting that, despite the valuable contribution and the impact of the Homomorphic Encryption Security Standard [58], there are some limitations, which have been pointed out by Curtis and Player [193], namely:

1) The standard does not consider a sparse ternary secret of Hamming weight h (i.e. a distribution where the elements are sampled uniformly at random from $\{-1, 0, 1\}^n$ with exactly h components different from zero). Note that many implementations use exactly this secret distribution (e.g. CKKS uses sparse ternary secret with $h = 64$).

2) The consideration of sparse secrets is not included in the standard since there exists a wider range of attacks that can be applied [193].

3) The standard, as well as the LWE Estimator, does not consider hybrid attacks. In particular, when the secret vector follows a sparse distribution with Hamming weight $h = 128$, hybrid attacks are very powerful and, as a result, we have a noticeable security loss [193]. In fact, the new version of the Estimator, i.e. the Lattice Estimator, considers this kind of attacks. It is also worth mentioning that the Lattice Estimator was updated to state-of-the-art attacks, hence covering not only hybrid but also exhaustive search and MITM attacks (see the blog [194] of Curtis and Walter).

To conclude this section we highlight the weakness of adopting secrets from a sparse distribution and the deleterious effects of hybrid attacks with one example reported by [193, Table 2] in Table II. Specifically, in Table II, $\lambda_{\mathsf{target}}$ is the currently standardised LWE security level for specific $n = 2^{10}$ and q for a uniform ternary secret [88]. The last four columns represent the security of each parameter set against uSVP attacks (Section VI-B2), dual attacks (Section VI-C), hybrid decoding attacks [172] (Section VI-B1), and hybrid dual attacks [184] (Section VI-C) for a *sparse* secret with Hamming

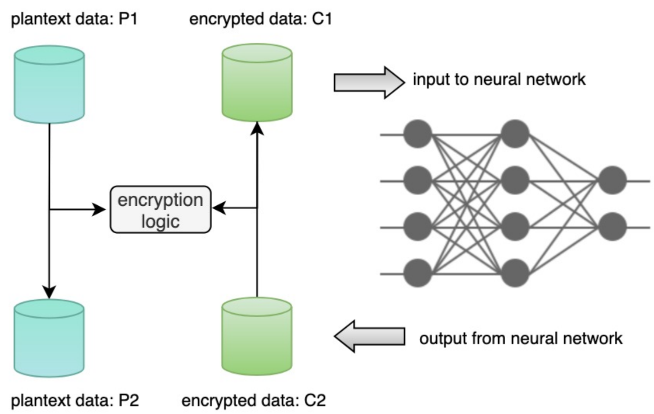

Fig. 13. Example of a potential privacy-preserving ML model.

weight $h = 128$, using the BKZ algorithm. Note that [193] used a conservative analysis for both hybrid attacks.

TABLE II
SECURITY LEVEL IN BITS FOR A SPARSE TERNARY SECRET OF HAMMING WEIGHT $h = 128$ AND HYBRID ATTACKS AS IN [193, TABLE 2].

n	λ_{target}	$\log q$	uSVP	dual	hybrid-dec	hybrid-dual
1024	128	27	124.9	127.8	111.5	**106.2**
	192	19	178.2	178.8	146.2	**141.8**
	256	14	235.5	238.5	181.5	**176.6**

VII. FHE FOR MACHINE LEARNING

This section provides a comprehensive view on the combined topic of privacy-preserving and machine learning, which, though fairly new, has been the subject of multiple research efforts. Machine Learning (ML) refers to a set of algorithms and computing systems used to build models that incorporate or learn structural knowledge of input datasets. A limitation to a wide adoption is the fact that machine learning mandates access to a large amount of data to achieve high accuracy rates, thus introducing data privacy and security concerns. FHE facilitates arithmetic evaluations of encrypted data of real numbers, which in turn enables the development of privacy-preserving machine learning training algorithms, and potentially provides a way to overcome aforementioned privacy and security concerns. FHE plays a critical role in distributed Machine Learning as it has the ability to support confidential secure computing scenarios. An example of a potential privacy-preserving machine learning model is shown in Figure 13.

A. Support Vector Machines

Support Vector Machines (SVMs) are widely used for their performance in classifications tasks, and multiple privacy-preserving SVM computing schemes have been proposed. Laur, Lipmaa and Mielikäinen propose in [195] a privacy-preserving scheme for both SVM training and classification using additive homomorphic encryption and secret sharing or secure multiparty computation protocols. Park, Byun, Lee, Cheon and Lee [196] present an algorithm based on Homomorphic Encryption for the SVM training phase which avoids inefficient operations within an encrypted domain. Rahulamathavan, Phan, Veluru, Cumanan and Rajarajan propose in [197] a two-class and multi-class classification protocol which uses SVMs, which exploits Paillier's cryptosystem [8] and secure two-party computation (client and server parties hold a share of the secret). In a more practical implementation, Makri, Rotaru, Smart and Vercauteren present EPIC [198], an image classification system trained with an SVM computing scheme, while input features are extracted based on the techniques of transfer learning. EPIC used Multi-Party Computation (MPC) tools to achieve privacy-preserving classification tasks and can be applied to homomorphic encryption domain.

B. Neural Networks and Other Machine Learning Models

Neural networks can be thought of as a generalization of regression to present elaborate relationships between high dimensional input data and output data. Privacy-preserving machine learning with neural networks has been addressed by multiple research efforts, even though computational complexity remains a challenge especially when neural networks are used for training over encrypted data.

Graepel, Lauter and Naehrig, in [199] use training algorithms, which can be expressed as low degree polynomials, in order to train over encrypted data leveraging SHE. While this works well on very limited applications, the accuracy of the proposed system is relatively low and cannot compete with neural networks. It also cannot be scaled to more complex operations such as division or exponentiation. Nikolaenko, Weinsberg, Ioannidis, Joye, Boneh and Taft [200] created a high performance ridge regression system using homomorphic encoding (Additively Homomorphic Encryption) and Garbled Circuits and evaluated it on very large scale datasets. Bost, Popa, Tu and Goldwasser [201] propose a scheme that uses three homomorphic systems (i.e. Paillier cryptosystem, Quadratic Residuosity, and BGV scheme) and garbled circuits to provide privacy-preserving classification for three different machine learning algorithms: Hyperplane Decision, Naive Bayes, and Decision trees, where features' description is assumed public. Mohassel and Zhang [202] present protocols for privacy-preserving machine learning for linear regression, logistic regression and neural network training using the stochastic gradient descent method. Aslett, Esperança and Holmes [203] present methodologies to train machine learning models such as random forests - using a stochastic fraction estimator - and naïve Bayes - using a semi-parametric model for class decision boundary - and demonstrate their accuracy when applied to data encrypted with homomorphic encryption. Khedr, Gulak and Vaikuntanathan [204] present a hardware architecture that implements Bayesian filters and Decision Trees (DT) for homomorphically encrypted data. Li *et al.* [205] investigate multi-key FHE using collaborative learning over input datasets encrypted with different encryption schemes and keys. The approach, however, suffers from scalability issues and high computational complexity. Dowlin *et al.* [206] present CryptoNets which applies a neural network - an artificial feed-forward neural network, known to a specific party and trained on plaintext data - to make predictions with a high accuracy on homomorphically encrypted data. The performance of CryptoNets is rather limited due to the replacement of the sigmoidal activation function and the computational overhead. Zhang, Yang, Chen, Li and Deen [207] propose a privacy-preserving deep learning model - a double-projection deep computation model whereas learning is outsourced to a cloud layer to improve the learning efficiency - trained with a back-propagation algorithm and uses BGV scheme. Improving on CryptoNets, Brutzkus, Elisha and Gilad-Bachrach [208] present a version of this latter which improves latency and memory usage. Lee, Kang, Lee, Choi, Eom and Deryabin [209] show the possibility of applying FHE (with bootstrapping) to a deep neural network model by implementing ResNet-20 over the residue number system CKKS scheme.

The viability of FHE's usage on large scale data and sharing frameworks has been demonstrated in multiple works. Hesamifard, Takabi, Ghasemi and Wright [210] present a methodology to train a convolutional neural network (CNN) model using homomorphically encrypted data, yielding high performance overheads. Al Badawi *et al.* [211] presents a CNN used for image classification with FHE properties on Graphics Processing Units (GPU), to accelerate classification, while maintaining a high accuracy rate. Blat, Gusev, Polyakov and Goldwasser [212] propose a toolbox of optimized statistical techniques that leverages FHE in order to perform studies on reformulated genomic data, and prove the viability of using homomorphic encryption on large scale data. Zhang and Zhu [213] propose the usage of homomorphic encryption to preserve privacy in sharing frameworks. The authors present a novel privacy-preserving architecture, which collaboratively trains a deep neural network while preserving the privacy of the data of sharing parties via homomorphic encryption.

C. The Industry Role

In addition to research efforts, multiple commercial products are being proposed to solve real-world problems across industry verticals. Zama's open source technology [214] enables trained machine learning models, regardless of the underlying architecture or training method, to run inference on encrypted user data using homomorphic encryption. Application of this technology could be extended to the medical field, image classification, autonomous environments and smart cities data processing. Intel [215] and Ant Group [216] have announced a joint effort [217] to build Privacy-Preserving Machine Learning (PPML) on top of Intel's Software Guard Extensions (SGX) and Occlum, Ant Group's memory-safe, multi-process library operating system for Intel SGX, using cryptographic technologies such as homomorphic encryption and differential privacy. Duality Technologies [218] is a company providing privacy-preserving data collaboration platforms using homomorphic encryption. It has been chosen by DARPA along with other top research institutes to accelerate the use of FHE as part of DARPA's Data Protection in Virtual Environments (DPRIVE) program, which seeks to develop a hardware accelerator for FHE computations [219].

D. Research Directions

While considerable advances have been achieved, privacy-preserving neural networks using homomorphic encryption still suffer from high computational complexity, low efficiencies, and inadequacy of deployment in real world scenarios. Further research is required to develop efficient frameworks enabling training and evaluation of complex neural networks over encrypted data or encrypted neural networks trained over plaintext data. Research directions could include:

- Algorithmic improvements: this includes (i) usage of pre-trained models to reduce computational complexity during the training phase, (ii) approximation of activation functions using polynomials, etc.
- Hardware acceleration: this includes parallelization and partitioning of the implementation of privacy-preserving models using homomorphic encryption and inherent operations on GPU cores (including hybrid CPU-GPU architectures), FPGAs, ASICs, and reconfigurable processors.

VIII. HE IN FOG COMPUTING FOR IOT

Fog computing was initially proposed by CISCO to support scalable massive IoT deployments [220] [221], but a similar concept has been adopted in 5G/6G cellular networks and referred to as edge computing [222]. It defines a layer between the IoT device and the cloud service, as close as possible to the device, where data is pre-processed. Pushing pre-processing operations close to the device is paramount to reduce both bandwidth consumption and the latency of IoT applications. In this scenario, HE can provide the missing privacy feature since pre-processing tasks can be done over encrypted data. However, fog computing also presents intrinsic characteristics that must be taken into account at the time of applying HE. Unlike cloud computing, fog computing considers data in motion, i.e. moving through the network at the generation rate of each specific device. Data processing is event-driven (triggered by the device) and performed packet-by-packet. Thus, data processing is delay intolerant and the scope of the processing tasks is limited to the information contained in a single data packet: relevance/category evaluation; formatting; encoding; expanding/compressing; filtering; or assessing thresholds and real-time alerts.

The Smart City scenario provides an example of the applicability of fog computing and how HE can become relevant. Smart cities cover a wide set of applications [223] such as intelligent transportation, efficient resource distribution (lighting, water and waste management), safety and security, or environmental monitoring. These applications have a common requirement, they are supported by massive IoT deployments composed of small sensors constantly fuelling data to smart city data collectors. The fog layer is in charge of pre-processing the data in intermediate gateways, which is fundamental to maintain scalability [224]. HE can be adopted to provide privacy preservation for citizens. Specifically, sensors can encrypt the data with the data collector's public key, and the fog layer processing operations can be performed over encrypted data. Only the data collector can decrypt the data with its secret key. This approach not only prevents data disclosure in fog nodes, but also provides privacy for IoT nodes with respect to the data collector since data samples are aggregated [225], [226].

Despite its potential, HE presents a difficult fit in IoT due to its computational complexity and the ciphertext expansion. The latter refers to the fact that a ciphertext is far larger than the corresponding plaintext, hence it adds a considerable communication overhead. The former incurs a computational delay at the time of acquiring and transmitting data samples. This is aggravated by the limited hardware capabilities and computational constraints of IoT devices, and the bandwidth constraints of current IoT communication standards. Aforementioned constraints have fostered research on hybrid protocols combining HE and symmetric key encryption (SKE). In hybrid homomorphic encryption (HHE), the IoT device encrypts data using a SKE scheme, with a randomly generated key, and then encrypts this key with a HE scheme using the data collector's public key. A SKE scheme is less complex and is not affected by ciphertext expansion. The intermediate fog nodes can homomorphically evaluate the decryption circuit of the SKE scheme and convert SK-encrypted data into HE-encrypted data. Then, the data can be processed homomorphically and sent to the data collector, see Fig. 14.

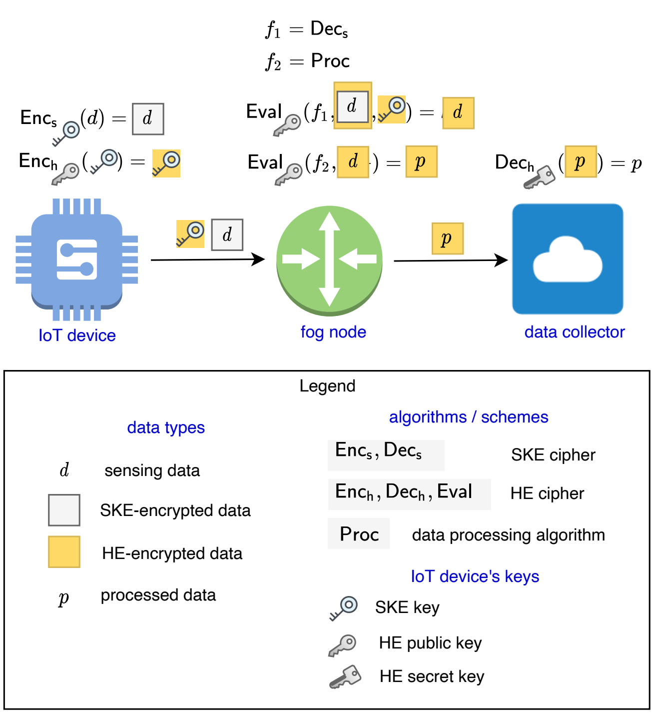

Fig. 14. Hybrid FHE scheme applied to fog computing for IoT.

The advantage is that the ciphertext expansion and the complexity is moved from the IoT devices to the fog nodes. The IoT device is only required to encrypt homomorphically a short key. Several works have proposed AES as a SKE scheme, [80] [76], since hardware acceleration for AES is a common feature in modern chips integrated in IoT devices. However, AES decryption circuit requires a multiplicative depth of at least 40, which increases the complexity in fog nodes. Some other approaches suggest public key encryption (PKE) instead of SKE [227], since the multiplicative depth is lower. However PKE has higher ciphertext expansion and complexity than SKE. In this framework, low-depth symmetric key ciphers such as [228], [229], [230] can provide noticeable gains.

Fog computing is not limited to the Smart City scenario. New emerging concepts such as Industry 4.0 or eHealth, which have coined the terms of Industrial IoT (IIoT) and Internet of Medical Things (IoMT), will also depend on the feasibility of deploying scalable micro-sensor systems. These scenarios will impose even more strict privacy requirements that could be potentially solved with HE. In general, hybrid HE should be envisioned as a solution for privacy-preserving data aggregation. However, it is worth commenting that hybrid protocols are effective when data is encrypted with the receiver's public key (the data collector in the smart city scenario). In a scenario where the IoT device encrypts with its own public key and the communication is bidirectional (the IoT device receives the encrypted processed data), the ciphertext expansion problem is unavoidable.

IX. HE in Cloud Computing

Homomorphic encryption can become a cornerstone component for technologies within the 5G/6G realm, namely for fog computing, but also for overarching technologies such as cloud computing. While fog computing is a distributed and decentralized infrastructure, cloud computing is a centralized system where data processing is query-based, and can be performed over large data sets from multiple application sessions. At first glance, it seems that HE provides a perfect solution to achieve privacy for cloud services. However, some cloud service scenarios impose requirements that make plain HE schemes unsuitable.

A. Homomorphic Proxy Re-Encryption

One of such scenarios occurs when a cloud service processes data from multiple users. The majority of HE schemes only support homomorphic operations over ciphertexts encrypted with the same public key. Hence, ciphertexts from different users must be converted into ciphertexts encrypted with the same key. This is called homomorphic proxy re-encryption (HPRE).

Proxy re-encryption (PRE) is a widely adopted technique in cloud computing for conventional (non homomorphic) encryption. PRE is used to transform a ciphertext from one user (the delegator) into a ciphertext of a different user (the delegatee) through a proxy. As a result, the delegatee can decrypt the delegator's ciphetext without learning the delegator's secret key. The proxy can convert ciphertexts without learning the plaintext or the users' keys. In HPRE, this process has an additional value, since it allows the cloud service to perform homomorphic operations over converted ciphertexts. Fortunately, in his thesis, Gentry proposes a simple construction to achieve HPRE. The delegator generates two ciphertexts: i) encrypts homomorphically the secret key with the delegatee's public key; and ii) encrypts the data with its own public key. Then, the proxy can evaluate the decryption circuit of the homomorphic scheme to re-encrypt the ciphertext with the delegatee's public key (this technique is identical to bootstrapping). Figure 15 shows how this process can be used to evaluate data from different users.

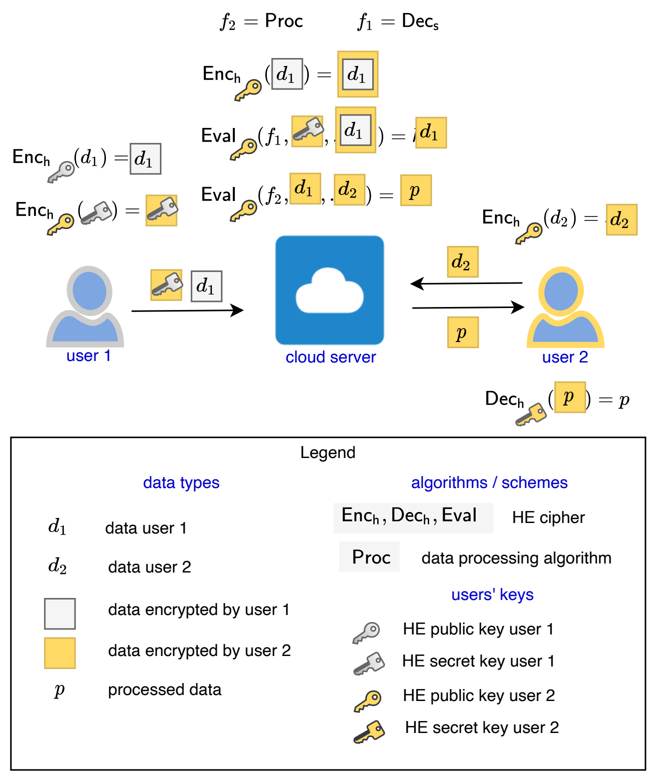

Fig. 15. HPRE for homomorphic evaluation of multi-user data.

Unfortunately, Gentry's approach is not resilient to weak collusion attacks. Specifically, the delegatee and the proxy can collude to obtain the delegator's secret key. New works based on key-switching techniques propose HPRE schemes that are resilient to collusion. Specifically, Derler *et al.* [231] and Kawai *et al.* [232] propose a single-hop (only one re-encryption is allowed) HPRE scheme that is partially homomorphic (only some homomorphic operations are possible after ciphertext conversion). Other works cater for fully homomorphic single-hop proxy re-encryption schemes, such as Ma *et al.* [233] and Yasuda *et al.* [234]. Polyakov *et al.* provided two IND-CPA secure constructions for multi-hop HPRE for BV and NTRU schemes [235], which outperform previous lattice-based proxy re-encryption schemes [236] [237] based on NTRU and BV respectively. Also, Li, Ma and Wang [238] [239] provided multi-hop HPRE schemes that are fully homomorphic via branching programs.

All these works solve the collusion attack that Gentry's approach presents, however they are not resilient to strong collusion attacks. Namely, the proxy and the delegatee cannot obtain the delegator's secret key but they can still obtain some information about the delegator's secret key [240]. Moreover, as stated in [241], known HPRE schemes are only CPA secure, which is not adequate in some scenarios [242]. Although it is well known that HE schemes cannot achieve CCA2 security (according to its standard definition), some can be CCA1-secure [243]. This is not true for HPRE, all known CCA1-secure HPRE schemes are only partially homomorphic.

B. Homomorphic Authenticated Encryption

In some scenarios, privacy is not sufficient. The user may pay for a specific service [244], or use remote data processing for safety-critical applications [245]. Thus, a guarantee that the data has been processed correctly by the cloud service may be required. Note that, even if the cloud service is not malicious, it maybe be willing to submit wrong data to avoid the heavy computational load of processing homomorphically encrypted data. In this scenario, the user should be able to verify that the decrypted data is the result of a specific arithmetic circuit over the transmitted encrypted data. Fortunately, this feature can be achieved with homomorphic authenticated encryption (HAE). HAE can be obtained by composing HE and homomorphic authentication (HA) [246], [247]. Specifically, the user sends the ciphertexts and attaches homomorphic authenticators in the form of homomorphic signatures (HS). These signatures can be evaluated homomorphically, similarly to ciphertexts, to produce a valid signature for the processed data (see Fig. 16). In fact, composing HE and HA caters for the interesting property that if both the HE and HA schemes are CPA secure then the resulting HAE scheme is CCA1 secure [246].

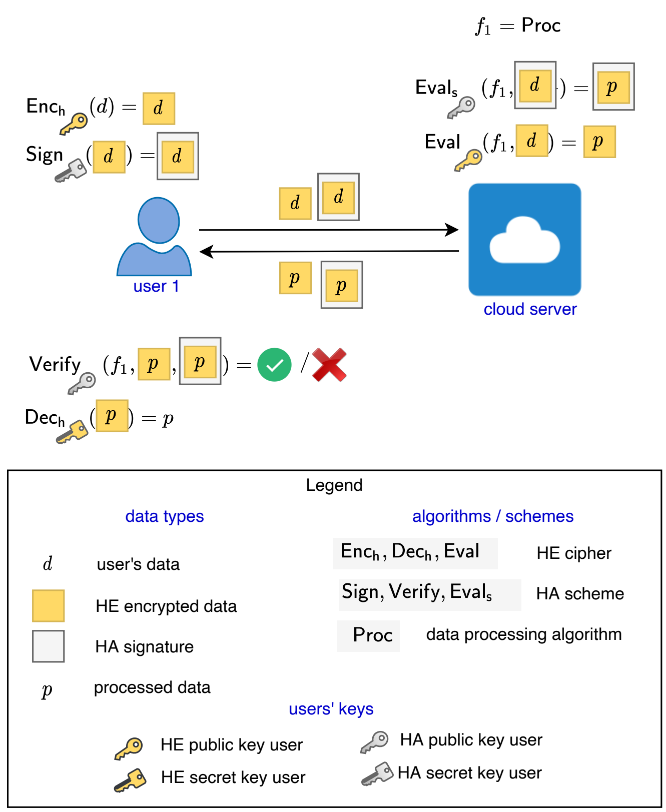

Fig. 16. FHAE by composition encrypt-then-sign.

HS were initially proposed for linear arithmetic circuits [248], [249], [250], [251]. Scenarios such as secure random linear network coding (RLNC) adopted HS [252] to counteract tag pollution attacks, although a symmetric-key based solution such as homomorphic MACs [253] can also fit in RLNC. Subsequent works provided HS schemes that accept polynomial homomorphic operations [254], [255], [256]. Gorbunov, Vaikuntanathan and Wichs [257] and Gennaro and Wichs [258] provided constructions with the desirable feature of being fully homomorphic (although the former is only leveled fully homomorphic), hence enabling fully homomorphic authenticated encryption (FHAE). However, previous HS schemes (except for [257]) are selectively secure (i.e. the attacker is only provided with signatures of chosen messages before the challenge is available). Adaptive security was first achieved by Boyen, Fan and Shi [259] and by subsequent works such as [257], [260]. Unfortunately, previous HS constructions have a limitation in terms of efficiency for circuits of polynomial depth [259]. Another solution, potentially more efficient, is the adoption of verifiable computation (VC) schemes that work over encrypted data [261], [262], [263]. A VC scheme provides a proof that each arithmetic gate of the arithmetic circuit has had its inputs processed. Moreover, it is even possible to provide such a proof of computation over a partially private circuit (known only to the cloud service), since the cloud service could prove that part in zero-knowledge.

C. *Homomorphic Encryption in Multi-Party Computation*

HE provides a solution for the centralization of private computations. However, in a scenario where several parties aim to interact, the direct application of HE is not so intuitive. Specifically, several cloud services may want to evaluate a function combining their private datasets without leaking any information about the inputs (except for what can be inferred from the output). Such scenario can be addressed with secure multi-party computation (MPC) [264] [265] [266]. There are different kinds of MPC protocols optimized for arithmetic and boolean circuits, based on secret sharing techniques [267], [268] and garbled circuits [269], [270] respectively. Interestingly, some of these protocols follow a pre-processing model where the computation is divided into two phases. The first phase happens before the parties' inputs are defined, and consists of the generation of cryptographic material (secret-shared elements or gabled circuits) that is later consumed to speed up the second phase. In the second phase, parties define their inputs and evaluate the circuit privately. The concept of consuming elements refers to the fact that this material cannot be used twice, hence it must be generated for each execution. It is precisely in the generation of the pre-processing material where HE still plays a fundamental role in MPC. Protocols like SHE-BMR [270], Overdrive [268], and the matrix multiplication protocol in [271] use leveled HE. It is worth commenting that, in this context, HE could be replaced by oblivious transfer (OT). In some protocols OT is more efficient than HE, but this must be evaluated per individual cases. Specifically, Overdrive [268] adopts a HE-based approach that improves the OT-based version of the same protocol, i.e. MASCOT [272]. However, HSS17 [273] adopts OT and is more efficient than SHE-BMR [270], which uses SHE. The key piece that makes HE faster than OT in Overdrive is the existence of an efficient zero-knowledge proof to prove knowledge of a plaintext in a HE-encrypted ciphertext, which is required to provide active security. On the other hand, HSS17 adopts OT because the OT protocol is compatible with an optimization generally adopted in garbled circuits (the FreeXOR technique).

Although less efficient than previous approaches, MPC can also be constructed directly with Multi-Key FHE (MKFHE). The work in [274] proposes a construction that requires only two communication rounds, see Fig. 17. In the first round, each party encrypts its inputs under multiple keys and broadcasts the ciphertexts to all parties. Then, each party evaluates the circuit homomorphically. Finally, in the second round, each party partially decrypts the result and broadcasts its share of the output. All shares can be combined locally by each party to obtain the output.

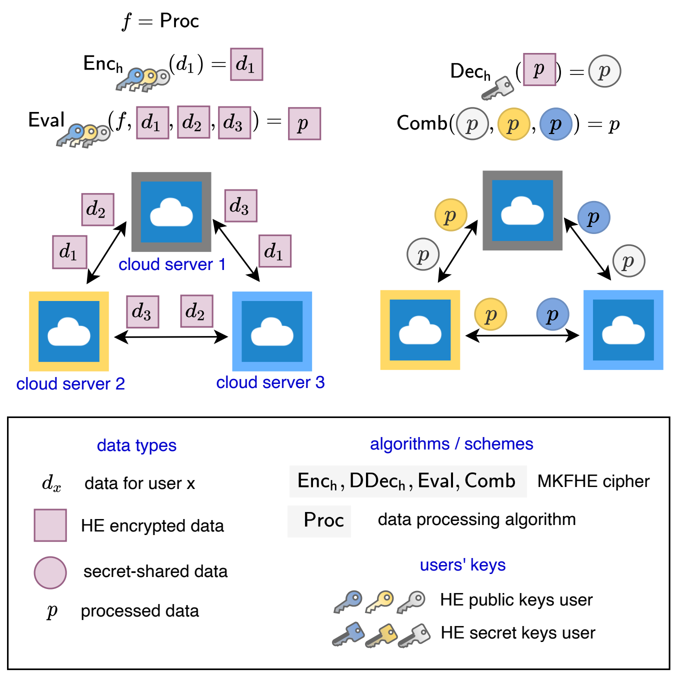

Fig. 17. MPC constructed with MKFHE. Only the operations performed by cloud server 1 have been detailed in the figure, but the rest of the parties behave analogously. For simplicity, the algorithm *Enc* in the figure generates a ciphertext for multiple keys, but in [274] this is achieved with an *Expand* algorithm.

In MKFHE, a ciphertext cannot be decrypted without all partial decryptions from the secret key holders. Hence, the input privacy is guaranteed. However, this also means that if only one party fails in delivering its share of the output, the MPC protocol would fail. This was addressed in [275] with a three-round MPC protocol that adopts Threshold MKFHE (TMFHE). TMFHE enables a smaller subset of parties to decrypt a ciphertext. Hence, after an initial round of input sharing, any subset of sufficient size can reconstruct the output. This makes the protocol resilient to failures, but it also requires trust since smaller subsets can decrypt the parties' private inputs. It is worth commenting that MKFHE is sufficient to provide MPC with passive security, i.e. malicious parties follow the protocol specification and try to extract information from the transmitted messages. But in a setting where parties can deviate from the protocol specification (e.g. transmit wrong shares of the output) active security is needed, and can be achieved by integrating zero-knowledge proofs.

X. Towards practical FHE-based applications

The first FHE schemes were about 10^9 times slower than plaintext computations [2], and were hence considered far from being practical. Optimizations achieved over the past decade have tremendously improved the performance of FHE schemes [31].

From a software perspective, FHE libraries have been pivotal in helping researchers and practitioners write the first FHE-based applications, and their evolution and optimization has significantly increased the efficiency of such applications over the past years. However, utilizing such APIs requires deep knowledge of FHE schemes. Recently, higher-level tools have evolved, attempting to bridge the gap between engineers developing privacy-preserving applications and the technical FHE libraries at hand. Along the same lines, a number of FHE compilers have become available with the objective of converting high-level programs to FHE-based implementations. These compilers are a key step towards making FHE available to non-experts that require such foundational blocks to design privacy-preserving applications, hence contributing to broad FHE adoption.

From a hardware point of view, considerable efforts leveraging specific hardware architectures (e.g., FPGA and ASIC-based) have been made. Such designs are referred to as FHE accelerators, and provide substantial improvements of FHE schemes performance in software.

The rest of this section presents an overview of the most relevant software and hardware works proposed in literature, discusses trade-offs observed in terms of performance and other metrics, and describes few real-world applications currently leveraging FHE in production systems.

A. FHE *Libraries*

The principal objective of FHE libraries is to make FHE scheme operations available via an API. Besides the core functionality provided by KeyGen, Enc, Dec and Eval, most of the widely used libraries incorporate additional features that allow ciphertext maintenance (i.e. noise growth management during computations) and manipulation, as well as homomorphic addition and multiplication methods. However, the correct utilization is left for the developers who must have an in-depth knowledge of what each API call entails in a given privacy-preserving solution.

Table III provides available open source FHE libraries, the language in which they are written, supported FHE schemes, and the date of the last update release. The first library ever published is HElib (Homomorphic Encryption Library), by Halevi and Shoup [79], [287], which is implemented in C++ and built on top of the NTL library [288]. SEAL (Simple Encrypted Arithmetic Library), which is developed by Microsoft [83], is implemented in C++ and C# (to support `.NET`). It utilizes Intel's HEXL [289], a library providing efficient implementations of homomorphic encryption operations, specifically targeting AVX512-enabled processors. PALISADE is developed by a DARPA consortium including Duality Technologies, the New Jersey Institute of Technology, Raytheon BBN Technologies, the MIT, University of California San Diego, and others [276]. It is written in C++ and can be configured to use the NTL library.

Lattigo [277] was proposed by Mouchet, Bossuat, Troncoso-Pastoriza and Hubaux [290] and is the first library written in Go. The FHEW library [278] by Ducas and Micciancio, is written in C++, however, has not been updated since 2017. The TFHE library [279] was provided by the authors of the TFHE paper [14], is written in C++ and C, and requires at least one Fast Fourier Transform (FFT) processor to run. TFHE is considered to be the successor of the FHEW library. Concrete [280], [291] is Zama's variant of TFHE implemented in Rust. The HEAAN library [281] is implemented in C++ and is built on top of the NTL library. The RNS-HEAAN library [282] by Kyoohyung and Miran is implemented in C++; it has not been updated since 2018. FV-NFLlib [283] is written in C++ and it is based on the NFLlib C++ library [292]. NFLlib is a library dedicated to ideal lattice-based cryptography, and it is based on the Number Theoretic Transform (NTT). It is important to note that FV-NFLlib has not been updated in the last 5 years. The CuFHE [284] and NuFHE [285] are two GPU-based libraries that implement TFHE in CUDA. Specifically, the CuFHE library adopts an implementation of NTT, GPU-accelerated, that is based on [293] by Dai and Sunar. The NuFHE library provides support for either FFT or purely integer NTT. Finally, OpenFHE [286] is a new library (published in July 2022) designed by the authors of PALISADE, HElib, HEAAN, and FHEW libraries. It is written in C++ and it includes all relevant FHE schemes: BGV, B/FV, FHEW, TFHE and CKKS. It also implements some recent improvements that are not covered by PALISADE.

B. FHE *Compilers*

FHE compilers are high-level tools that aim at abstracting the technical APIs exposed by FHE libraries, so that a wider range of developers are able to implement privacy-preserving mechanisms securely. As noted by Viand, Jattke and Hithnawi [294], FHE compilers tackle some of the most common engineering challenges that exist nowadays when designing FHE-based applications:

- Parameters choice: defining appropriate parameter values for FHE schemes resulting in secure and efficient instances is not a simple task. Some FHE compilers allow for some sort of automatic parameter generation according to some predefined requirements.
- Plaintext encoding: in FHE, the semantics of the plaintext message are strictly related to the type of homomorphic computations that can be conducted. Some context-specific FHE compilers can already be used to aid in this particular item (e.g. nGraph-HE).
- Data-independent execution: given that FHE operations are data-independent by nature, it is not trivial to conduct data-dependent branching steps using FHE because they can break privacy properties. In this case, it is possible to

TABLE III
OPEN SOURCE LIBRARIES FOR FHE SCHEMES

Library	Language	Scheme BGV	B/FV	FHEW	TFHE	CKKS	Date of last commit
HElib [79]	C++	●	○	○	○	●	1/10/2021
SEAL [83]	C++/C#	●	●	○	○	●	24/3/2022
PALISADE [276]	C++	●	●	●	●	●	30/4/2022
Lattigo [277]	Go	○	●	○	○	●	13/6/2022
FHEW [278]	C++	○	○	●	○	○	30/5/2017
TFHE [279]	C++/C	○	○	○	●	○	16/9/2021
concrete [280]	Rust	○	○	○	●	○	10/5/2022
HEAAN [281]	C++	○	○	○	○	●	27/1/2022
RNS-HEAAN [282]	C++	○	○	○	○	●	26/10/2018
FV-NFLlib [283]	C++	○	●	○	○	○	26/7/2016
CuFHE [284]	Cuda/C++	○	○	○	●	○	9/2/2019
NuFHE [285]	Python	○	○	○	●	○	18/3/2020
OpenFHE [286]	C++	●	●	●	●	●	18/8/2022

have branching operations by means of evaluating both branches and selecting the result at the end.

- Packing or batching: FHE schemes allowing for message packing or batching into a single ciphertext can directly leverage SIMD instruction sets. Some FHE compilers already actively optimize for vectorized operations.
- Ciphertext maintenance: optimally managing how noise grows during FHE operations is not straightforward, and FHE compilers are starting to use advanced strategies to assist in this traditionally complicated part.

Table IV presents a list of FHE compilers showing in which programming language they are written, what FHE libraries (from the ones previously highlighted in this work) they utilize as well as the date of their latest released update. ALCHEMY [295], [306], [307] is a compiler written in Haskell by Crockett, Peikert and Sharp. It implements BGV utilizing $\Lambda \circ \lambda$ [308] (pronounced "L O L"), a library for ring-based lattice cryptography that supports also FHE. Cingulata (previously called Armadillo) [296], [309] is a compiler written in C++ by Carpov, Dubrulle and Sirdey. It is built on top of the FLINT [310] and Sage [311] libraries. The Encrypt-Everything-Everywhere (E^3) [297], [312] is a framework presented by Chielle, Mazonka, Tsoutsos and Maniatakos. It is mainly written in C++ and supports a variety of FHE libraries. SHEEP [298], a recursive acronym for SHEEP is a Homomorphic Encryption Evaluation Platform, is a framework developed by the Turing Institute, written in C++ and that comes with several off the shelf Jupyter notebooks containing examples on how to use SHEEP. The Encrypted Vector Arithmetic Language and Compiler (EVA) [299], [313] was presented by Dathathri et al., is written in C++, and incorporates CHET [303] to support tensor circuits. Marble [300], [314] is a C++ compiler written by Viand and Shafagh, and RAMPARTS [301] is a compiler written in Julia by Archer *et al.* The Transpiler [302], [315] is a C++ tool developed by Gorantala *et al.* that is currently leveraging one FHE library. The nGraph-HE [304], [316] compiler by Boemer, Lao, Cammarota and Wierzynski is based on Intel's nGraph ML compiler [317]. Support for non-polynomial activation functions was added subsequently [318]. Finally, SEALion [305] was presented by van Elsloo, Patrini and Ivey-Law. It is important to highlight that CHET, nGraph-HE and SEALion are domain-specific FHE compilers designed particularly for ML applications. To the best of the authors knowledge, we note that the libraries without a date provided for their last committed update have not been found publicly. For more details about the FHE libraries and compilers, we would like to refer the readers to [294] by Viand, Jattke and Hithnawi.

C. FHE *Accelerators*

Previous sections presented state-of-the-art software tools that are proving crucial in the wider adoption of FHE for developing privacy-preserving solutions. Although such tools have enabled a significant acceleration of FHE schemes, the corresponding performance still falls short with respect to plaintext computations. Therefore, FHE hardware accelerators have emerged as a practical alternative to highly optimized software implementations, thus, enabling a wider range of use cases where FHE can be utilized.

Doröz, Öztürk and Sunar [319] designed an accelerator for the Gentry and Halevi scheme [64] and were able to significantly improve run times of FHE operations. Later on, Cousins, Rohloff and Sumorok [320] designed an FPGA-based accelerator focusing on the second-generation, NTRU-based scheme, LTV. They targeted a Xilinx Virtex-7 FPGA and benchmarked the performance of the CRT (and its inverse). Their results showed an improvement of 2 orders of magnitude compared to the available reference software and CPU-based version. In [321], Roy *et al.* presented a RLWE-based co-processor targeting NTT optimizations and achieved considerable speeds on a Virtex-6 FPGA. Moreover, Roy *et al.* [322] proposed an architecture where they are able to offload operations of the FV FHE scheme to an FPGA-based accelerator. They validated their design on a Xilinx Zynq UltraScale+ FPGA and obtained an improvement of over 13 times with respect to a reference FV scheme optimized software implementation. Riazi, Laine, Pelton and Dai [323] provided a new hardware design that heavily improved the NTT operation, implemented the CKKSN scheme, and can be used with a wide range of parameter sets. They compared their proposal on two FPGA devices from Intel, namely Arria 10 and Stratix 10, with an optimized version of the SEAL library

TABLE IV
PUBLICLY AVAILABLE FHE COMPILERS

Compiler	Language	Library						Date of last commit
		HElib	SEAL	PALISADE	FHEW	TFHE	HEAAN	
ALCHEMY [295]	Haskell	○	○	○	○	○	○	15/3/2020
Cingulata [296]	C++	○	○	○	○	●	○	7/12/2020
E^3 [297]	C++	●	●	●	●	●	○	31/5/2022
SHEEP [298]	C++	●	●	●	○	●	○	11/11/2019
EVA [299]	C++	○	●	○	○	○	○	1/5/2021
Marble [300]	C++	●	●	○	○	○	○	23/12/2020
RAMPARTS [301]	Julia	○	○	●	○	○	○	-
Transpiler [302]	C++	○	○	●	○	●	○	21/6/2022
CHET [303]	C++	○	●	○	○	○	●	-
nGraph-HE [304]	C++	○	●	○	○	○	○	8/7/2021
SEALion [305]	C++	○	●	○	○	○	○	-

and demonstrated an improvement of more than 2 orders of magnitude. Finally, Turan, Roy and Verbauwhede [324] proposed the first accelerator to leverage FPGAs available in the Amazon AWS cloud, and achieved a 20 times improvement with respect to the software implementation of the smart meter application that they consider in their case study.

Even though FPGA-based designs bear considerable improvements and can be run on accessible FPGA platforms, they miss an important element of FHE computations, the data movement, which, in extremely optimized designs, becomes a non-negligible bottleneck. ASIC-based designs enable the possibility to tackle potential issues caused by data movement congestion. Following this approach, Juvekar, Vaikuntanathan and Chandrakasan [325] designed Gazelle, which combines two conventional techniques, namely, homomorphic encryption and garbled circuits. A low-latency ASIC targeting a secure neural network inference running on Gazelle achieved 2-3 orders of magnitude speedups. Subsequently, Gazelle was improved (i.e. 79 times faster) by Reagen *et al.* with Cheetah [326]. Recently, Feldmann, Samardzic *et al.* have presented F1 [327], a programmable FHE accelerator that employs a wide-vector processor, which is based on a static scheduling strategy and minimizes data movement. F1 is capable of producing speedups of up to 3-4 orders of magnitude with respect to state-of-the-art software implementations, and supports BGV, HEEAN and GSW. Moreover, F1 demonstrates that ASIC-based accelerators can also be programmable, given that the same resulting hardware can accelerate a variety of programs, including multiple FHE schemes.

D. Standardization and Broad Adoption

Section X-A, Section X-B and Section X-C provide a comprehensive list of the most relevant works in software and hardware that have contributed towards making FHE more practical, hence, broadening its adoption.

However, other types of initiatives equally play a key role towards having large-scale FHE deployments. For instance, the Homomorphic Encryption open industry / government / academic consortium [58] is working on a standard for homomorphic encryption. The consortium was created in 2017 by Microsoft, IBM and Duality Technologies, and, at the time of writing this work, has more than 40 participants amongst industry, government and academia. Moreover, industrial players such as Duality Technologies [218], Zama AI [214], or Cryptolab [328] aiming at deploying FHE-based solutions, are greatly contributing to the overall ecosystem. Lastly, large tech-based companies like Intel and Google, are starting to leverage homomorphic encryption in privacy-preserving solutions such as building PPML on top of Intel's SGX [217] and Google's Password Checkup [329], which employs Private Set Intersection.

XI. CONCLUSIONS

Homomorphic encryption has been a prolific research field over the past decade. Since Gentry's first proposed scheme in 2009, several generations of schemes have emerged, fostered by the evolution of privacy-preserving technologies. Moreover, synergies with other research fields (such as Machine Learning) and with other cryptographic protocols (such as secure multi-party computation) have increased its relevance.

Nevertheless, despite the tremendous potential of the field, current FHE schemes still present limitations that hinder their applicability within real environments. The computational complexity and ciphertexts expansion render FHE unsuitable to delay-intolerant or bandwidth-limited applications. These latter are the main impediments for FHE's widespread adoption in new generation networks.

Additionally, there are no known common schemes that encompass features offered by second, third and fourth generation schemes simultaneously, which would otherwise be convenient in some scenarios, such as privacy-preserving Machine Learning. More specifically, second and fourth generation schemes are equipped with packing techniques, which make them efficient for matrix multiplication, while third generation schemes are the only ones to enable efficient evaluation of non-linear functions. Moreover, second and fourth generation schemes are not equipped with fast bootstrapping techniques; this limits their application to their leveled version, not their fully homomorphic version. Another limitation is the absence of thorough efforts related to noise analysis, mainly for second generation schemes. There is still a gap between theoretical bounds and the real noise growth, which increases the complexity of parameter selection. These limitations have triggered numerous research proposals, not only on new schemes and analytical studies on parameters, but also on hardware accelerators. New technologies such as memory-based computation have also been proposed for memory hungry applications, such as private deep neural networks. It is precisely the advancement on hardware acceleration that can make FHE a reality, by reducing its time complexity by several orders of magnitude: cloud computing will require more efficient implementations that cannot be achieved with software-based optimizations. The potential integration of FHE in massive IoT deployments will also depend on the ongoing research efforts on hardware. These new scenarios will define strict specifications in terms of latency, bandwidth and energy efficiency, and the FHE layers would have to meet these requirements.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Damien Stehlé for the technical discussions and his feedback on this manuscript. Also, the authors would like to thank the anonymous reviewers for their thorough reading of the manuscript, and the very detailed comments provided.

This work has been partly funded by the European Commission through the H2020 projects Hexa-X (Grant Agreement no. 101015956). This work has also been partially funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden. Marc Manzano is a member of the Intelligent Systems for Industrial Systems research group of Mondragon Unibertsitatea (IT1676-22), supported by the Department of Education, Universities and Research of the Basque Country.

REFERENCES

[1] R. L. Rivest, L. Adleman, M. L. Dertouzos *et al.*, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[2] C. Gentry, *A fully homomorphic encryption scheme*. Stanford university Stanford, 2009, vol. 20, no. 9.

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[4] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, ser. STOC '82. New York, NY, USA: Association for Computing Machinery, 1982, p. 365–377.

[5] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[6] J. Benaloh, "Dense probabilistic encryption," in *Proceedings of the workshop on selected areas of cryptography*, 1994, pp. 120–128.

[7] D. Naccache and J. Stern, "A new public key cryptosystem based on higher residues," in *Proceedings of the 5th ACM conference on Computer and communications security*, 1998, pp. 59–66.

[8] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[9] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography Conference*. Springer, 2005, pp. 325–341.

[10] C. A. Melchor, P. Gaborit, and J. Herranz, "Additively homomorphic encryption with d-operand multiplications," in *Annual Cryptology Conference*. Springer, 2010, pp. 138–154.

[11] C. Gentry, "Computing arbitrary functions of encrypted data," *Communications of the ACM*, vol. 53, no. 3, pp. 97–105, 2010.

[12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.

[13] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptology ePrint Archive*, 2012.

[14] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: fast fully homomorphic encryption over the torus," *Journal of Cryptology*, pp. 1–58, 2019.

[15] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017*, T. Takagi and T. Peyrin, Eds. Cham: Springer International Publishing, 2017, pp. 409–437.

[16] T. S. Fun and A. Samsudin, "A survey of homomorphic encryption for outsourced big data computation," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 10, no. 8, pp. 3826–3851, 2016.

[17] Z. Brakerski and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE," in *IEEE 52nd Annual Symposium on Foundations of Computer Science – FOCS*, 2011, pp. 97–106, full version in https://eprint.iacr.org/2011/344.

[18] ——, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in *Advances in Cryptology – CRYPTO 2011*, P. Rogaway, Ed. Berlin, Heidelberg: Springer, 2011, pp. 505–524.

[19] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in Cryptology – EUROCRYPT 2010*, H. Gilbert, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 24–43.

[20] N. Aggarwal, C. Gupta, and I. Sharma, "Fully homomorphic symmetric scheme without bootstrapping," in *Proceedings of 2014 International Conference on Cloud Computing and Internet of Things*. IEEE, 2014, pp. 14–17.

[21] C. Gupta and I. Sharma, "A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds," in *2013 Fourth International Conference on the Network of the Future (NoF)*. IEEE, 2013, pp. 1–4.

[22] R. Rothblum, "Homomorphic encryption: From private-key to public-key," in *Theory of cryptography conference*. Springer, 2011, pp. 219–234.

[23] P. Martins, L. Sousa, and A. Mariano, "A survey on fully homomorphic encryption: An engineering perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–33, 2017.

[24] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–35, 2018.

[25] A. Aloufi, P. Hu, Y. Song, and K. Lauter, "Computing blindfolded on data homomorphically encrypted under multiple keys: An extended survey," *arXiv preprint arXiv:2007.09270*, 2020.

[26] J. H. Cheon, A. Costache, R. C. Moreno, W. Dai, N. Gama, M. Georgieva, S. Halevi, M. Kim, S. Kim, K. Laine *et al.*, "Introduction to homomorphic encryption and schemes," in *Protecting Privacy through Homomorphic Encryption*. Springer, 2021, pp. 3–28.

[27] A. Hülsing, T. Lange, and K. Smeets, "Rounded gaussians," in *IACR International Workshop on Public Key Cryptography*. Springer, 2018, pp. 728–757.

[28] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

[29] ——, *Lattice-based Cryptography*, D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds. Berlin, Heidelberg: Springer, 2009.

[30] C. Peikert, "Lattices in Cryptography," 2013, https://web.eecs.umich.edu/~cpeikert/lic13/.

[31] ——, "A decade of lattice cryptography," *Foundations and Trends ® in Theoretical Computer Science*, vol. 10, no. 4, pp. 283–424, 2016.

[32] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa, "Efficient public key encryption based on ideal lattices," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2009, pp. 617–635.

[33] SVP challenge, https://www.latticechallenge.org/svp-challenge/.

[34] S. R. Kumar and D. Sivakumar, "On the unique shortest lattice vector problem," *Theoretical computer science*, vol. 255, no. 1-2, pp. 641–648, 2001.

[35] Y.-K. Liu, V. Lyubashevsky, and D. Micciancio, "On bounded distance decoding for general lattices," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2006, pp. 450–461.

[36] V. Lyubashevsky and D. Micciancio, "On bounded distance decoding, unique shortest vectors, and the minimum distance problem," in *Advances in Cryptology - CRYPTO 2009*, S. Halevi, Ed. Berlin, Heidelberg: Springer, 2009, pp. 577–594.

[37] S. Bai, D. Stehlé, and W. Wen, "Improved reduction from the bounded distance decoding problem to the unique shortest vector problem in lattices," in *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[38] S. Khot, "Hardness of approximating the shortest vector problem in high ℓ_p norms," *Journal of Computer and System Sciences*, vol. 72, no. 2, pp. 206–219, 2006.

[39] W. Wen, "Contributions to the hardness foundations of lattice-based cryptography. (contributions aux fondements de complexité de la cryptographie sur réseaux)," Ph.D. dissertation, University of Lyon, France, 2018. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01949339

[40] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 99–108.

[41] D. Micciancio, "Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions," in *IEEE 43rd Annual Symposium on Foundations of Computer Science – FOCS*, 2002, pp. 356–365.

[42] ——, "Generalized compact knapsacks, cyclic lattices, and efficient one-way functions," *Comput. Complex.*, vol. 16, no. 4, pp. 365–411, 2007. [Online]. Available: https://doi.org/10.1007/s00037-007-0234-9

[43] P. Bert, P.-A. Fouque, A. Roux-Langlois, and M. Sabt, "Practical implementation of ring-sis/lwe based signature and ibe," in *International Conference on Post-Quantum Cryptography*. Springer, 2018, pp. 271–291.

[44] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, 2005, pp. 84–93.

[45] ——, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.

[46] A. Blum, M. Furst, M. Kearns, and R. J. Lipton, "Cryptographic primitives based on hard learning problems," in *Advances in Cryptology — CRYPTO' 93*, D. R. Stinson, Ed. Berlin, Heidelberg: Springer, 1994, pp. 278–291.

[47] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proceedings of the 41st annual ACM symposium on Theory of computing*, 2009, pp. 333–342.

[48] D. Micciancio and P. Mol, "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions," in *Advances in Cryptology – CRYPTO 2011*, P. Rogaway, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 465–484.

[49] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Advances in Cryptology – EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds. Berlin, Heidelberg: Springer, 2012, pp. 700–718.

[50] Z. Brakerski, "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP," in *Advances in Cryptology – CRYPTO 2012*, R. Safavi-Naini and R. Canetti, Eds. Berlin, Heidelberg: Springer, 2012, pp. 868–886.

[51] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical hardness of learning with errors," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 2013, pp. 575–584.

[52] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *Advances in Cryptology - CRYPTO 2009*, S. Halevi, Ed. Springer, 2009, pp. 595–618. [Online]. Available: https://doi.org/10.1007/978-3-642-03356-8_35

[53] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology – EUROCRYPT 2010*, H. Gilbert, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–23.

[54] L. Ducas and A. Durmus, "Ring-LWE in Polynomial Rings," in *Public Key Cryptography – PKC 2012*, M. Fischlin, J. Buchmann, and M. Manulis, Eds. Berlin, Heidelberg: Springer, 2012, pp. 34–51.

[55] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Designs, Codes and Cryptography*, vol. 75, no. 3, pp. 565–599, 2015.

[56] C. Peikert, O. Regev, and N. Stephens-Davidowitz, "Pseudorandomness of ring-lwe for any ring and modulus," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 2017, pp. 461–473.

[57] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology – CRYPTO 2013*, R. Canetti and J. A. Garay, Eds. Berlin, Heidelberg: Springer, 2013, pp. 75–92.

[58] M. R. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan, "Homomorphic encryption security standard," HomomorphicEncryption.org, Toronto, Canada, Tech. Rep., November 2018.

[59] C. Gentry, S. Halevi, and V. Vaikuntanathan, "i-Hop Homomorphic Encryption and Rerandomizable Yao Circuits," in *Advances in Cryptology – CRYPTO 2010*, T. Rabin, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 155–172.

[60] Z. Brakerski, "Fundamentals of Fully Homomorphic Encryption-A Survey," in *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 25, 2018, p. 125.

[61] F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jäschke, C. A. Reuter, and M. Strand, "A guide to fully homomorphic encryption." *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 1192, 2015.

[62] A. Silverberg, "Fully homomorphic encryption for mathematicians," *Women in Numbers 2: Research Directions in Number Theory*, vol. 606, p. 111, 2013.

[63] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *International Workshop on Public Key Cryptography*. Springer, 2010, pp. 420–443.

[64] C. Gentry and S. Halevi, "Implementing Gentry's Fully-Homomorphic Encryption Scheme," in *Advances in Cryptology – EUROCRYPT 2011*, K. G. Paterson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 129–148.

[65] P. Scholl and N. P. Smart, "Improved key generation for gentry's fully homomorphic encryption scheme," in *IMA International Conference on Cryptography and Coding*. Springer, 2011, pp. 10–22.

[66] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2010, pp. 377–394.

[67] R. Cramer, L. Ducas, C. Peikert, and O. Regev, "Recovering short generators of principal ideals in cyclotomic rings," in *Advances in Cryptology – EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds. Berlin, Heidelberg: Springer, 2016, pp. 559–585.

[68] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully homomorphic encryption over the integers with shorter public keys," in *Advances in Cryptology – CRYPTO 2011*, P. Rogaway, Ed. Berlin, Heidelberg: Springer, 2011, pp. 487–504.

[69] Y. Chen and P. Q. Nguyen, "Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers," in *Advances in Cryptology - EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds. Springer, 2012, pp. 502–519.

[70] J.-S. Coron, D. Naccache, and M. Tibouchi, "Public key compression and modulus switching for fully homomorphic encryption over the integers," in *Advances in Cryptology – EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 446–464.

[71] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. IEEE, 1994, pp. 124–134.

[72] J. Kim, M. S. Lee, A. Yun, and J. H. Cheon, "CRT-based Fully Homomorphic Encryption over the Integers," Cryptology ePrint Archive, Report 2013/057, 2013, https://eprint.iacr.org/2013/057.

[73] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Batch Fully Homomorphic Encryption over the Integers," Cryptology ePrint Archive, Report 2013/036, 2013, https://eprint.iacr.org/2013/036.

[74] J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun, "Batch fully homomorphic encryption over the integers," in *Advances in Cryptology – EUROCRYPT 2013*, T. Johansson and P. Q. Nguyen, Eds. Berlin, Heidelberg: Springer, 2013, pp. 315–335.

[75] K. Nuida and K. Kurosawa, "(Batch) Fully Homomorphic Encryption over Integers for Non-Binary Message Spaces," in *Advances in Cryptology – EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer, 2015, pp. 537–555.

[76] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Scale-invariant fully homomorphic encryption over the integers," in *International Workshop on Public Key Cryptography*. Springer, 2014, pp. 311–328.

[77] J. H. Cheon and D. Stehlé, "Fully homomophic encryption over the integers revisited," in *Advances in Cryptology – EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer, 2015, pp. 513–536.

[78] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[79] Shai Halevi and Victor Shoup, "HElib," https://github.com/homenc/HElib.

[80] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic Evaluation of the AES Circuit," in *Advances in Cryptology – CRYPTO 2012*, R. Safavi-Naini and R. Canetti, Eds. Berlin, Heidelberg: Springer, 2012, pp. 850–867.

[81] F. Maino, U. Blumenthal, and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model," RFC 3826, 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3826.txt

[82] C. Gentry, S. Halevi, and N. P. Smart, "Fully homomorphic encryption with polylog overhead," in *Advances in Cryptology – EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds. Berlin, Heidelberg: Springer, 2012, pp. 465–482.

[83] "Microsoft SEAL (release 3.4)," https://github.com/Microsoft/SEAL, Oct. 2019, microsoft Research, Redmond, WA.

[84] J.-C. Bajard, J. Eynard, M. A. Hasan, and V. Zucca, "A full RNS variant of FV like somewhat homomorphic encryption schemes," in *International Conference on Selected Areas in Cryptography*. Springer, 2016, pp. 423–442.

[85] S. Halevi, Y. Polyakov, and V. Shoup, "An improved RNS variant of the BFV homomorphic encryption scheme," in *Cryptographers' Track at the RSA Conference*. Springer, 2019, pp. 83–105.

[86] A. Q. A. Al Badawi, Y. Polyakov, K. M. M. Aung, B. Veeravalli, and K. Rohloff, "Implementation and performance evaluation of RNS variants of the BFV homomorphic encryption scheme," *IEEE Transactions on Emerging Topics in Computing*, 2019.

[87] J. C. Bajard, J. Eynard, P. Martins, L. Sousa, and V. Zucca, "Note on the noise growth of the rns variants of the bfv scheme," *Cryptology ePrint Archive*, 2019.

[88] H. Chen and K. Han, "Homomorphic lower digits removal and improved FHE bootstrapping," in *Advances in Cryptology – EUROCRYPT 2018*, J. B. Nielsen and V. Rijmen, Eds. Cham: Springer International Publishing, 2018, pp. 315–337.

[89] S. Halevi and V. Shoup, "Faster homomorphic linear transformations in HElib," in *Annual International Cryptology Conference*. Springer, 2018, pp. 93–120.

[90] ——, "Bootstrapping for HElib," *Journal of Cryptology*, vol. 34, no. 1, pp. 1–44, 2021.

[91] H. Chen, K. Laine, R. Player, and Y. Xia, "High-precision arithmetic in homomorphic encryption," in *Cryptographers' Track at the RSA Conference*. Springer, 2018, pp. 116–136.

[92] J. Hoffstein and J. Silverman, "Optimizations for NTRU," *Public-Key Cryptography and Computational Number Theory, De Gruyter Proceedings in Mathematics*, pp. 77–88, 2000.

[93] C. Bootland, W. Castryck, I. Iliashenko, and F. Vercauteren, "Efficiently processing complex-valued data in homomorphic encryption," *Journal of Mathematical Cryptology*, vol. 14, no. 1, pp. 55–65, 2020.

[94] S. Arita and S. Nakasato, "Fully homomorphic encryption for point numbers," in *International Conference on Information Security and Cryptology*. Springer, 2016, pp. 253–270.

[95] C. Bonte, C. Bootland, J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren, "Faster homomorphic function evaluation using non-integral base encoding," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 579–600.

[96] A. Jäschke and F. Armknecht, "Accelerating homomorphic computations on rational numbers," in *International Conference on Applied Cryptography and Network Security*. Springer, 2016, pp. 405–423.

[97] A. Costache, K. Laine, and R. Player, "Evaluating the effectiveness of heuristic worst-case noise analysis in FHE," in *European Symposium on Research in Computer Security*. Springer, 2020, pp. 546–565.

[98] A. Costache and N. P. Smart, "Which ring based somewhat homomorphic encryption scheme is best?" in *Cryptographers' Track at the RSA Conference*. Springer, 2016, pp. 325–340.

[99] J. Mono, C. Marcolla, G. Land, T. Güneysu, and N. Aaraj, "Finding and Evaluating Parameters for BGV," *Cryptology ePrint Archive*, 2022.

[100] A. Kim, Y. Polyakov, and V. Zucca, "Revisiting homomorphic encryption schemes for finite fields," 2021.

[101] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *International Algorithmic Number Theory Symposium*. Springer, 1998, pp. 267–288.

[102] ——, "Public key cryptosystem method and apparatus," U.S. Patent 6,081,597, (2000).

[103] D. Stehlé and R. Steinfeld, "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices," in *Advances in Cryptology – EUROCRYPT 2011*, K. G. Paterson, Ed. Berlin, Heidelberg: Springer, 2011, pp. 27–47.

[104] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 2012, pp. 1219–1234.

[105] J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," in *IMA International Conference on Cryptography and Coding*. Springer, 2013, pp. 45–64.

[106] M. Albrecht, S. Bai, and L. Ducas, "A Subfield Lattice Attack on Overstretched NTRU Assumptions," in *Advances in Cryptology – CRYPTO 2016*, M. Robshaw and J. Katz, Eds. Berlin, Heidelberg: Springer, 2016.

[107] J. H. Cheon, J. Jeong, and C. Lee, "An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero," *LMS Journal of Computation and Mathematics*, vol. 19, no. A, pp. 255–266, 2016.

[108] Y. Doröz and B. Sunar, "Flattening NTRU for Evaluation Key Free Homomorphic Encryption," *IACR Cryptology ePrint Archive*, vol. 2016, p. 315, 2016.

[109] T. Lepoint and M. Naehrig, "A comparison of the homomorphic encryption schemes FV and YASHE," in *International Conference on Cryptology in Africa*. Springer, 2014, pp. 318–335.

[110] M. Kim and K. Lauter, "Private genome analysis through homomorphic encryption," in *BMC medical informatics and decision making*, vol. 15, no. 5. BioMed Central, 2015, pp. 1–12.

[111] Z. Brakerski and V. Vaikuntanathan, "Lattice-based FHE as secure as PKE," in *Proceedings of the 5th conference on Innovations in theoretical computer science*, 2014, pp. 1–12.

[112] A. Khedr, G. Gulak, and V. Vaikuntanathan, "SHIELD: scalable homomorphic implementation of encrypted data-classifiers," *IEEE Transactions on Computers*, vol. 65, no. 9, pp. 2848–2858, 2015.

[113] J. Alperin-Sheriff and C. Peikert, "Faster bootstrapping with polynomial error," in *Advances in Cryptology – CRYPTO 2014*, J. A. Garay and R. Gennaro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 297–314.

[114] ——, "Practical bootstrapping in quasilinear time," in *Advances in Cryptology – CRYPTO 2013*, R. Canetti and J. A. Garay, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–20.

[115] R. Hiromasa, M. Abe, and T. Okamoto, "Packing messages and optimizing bootstrapping in GSW-FHE," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 99, no. 1, pp. 73–82, 2016.

[116] L. Ducas and D. Micciancio, "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second," in *Advances in Cryptology – EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 617–640.

[117] I. Chillotti, D. Ligier, J.-B. Orfila, and S. Tap, "Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for tfhe," *IACR Cryptology ePrint Archive*, 2021.

[118] M. Frigo and S. G. Johnson, "The design and implementation of fftw3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[119] N. Gama, M. Izabachène, P. Q. Nguyen, and X. Xie, "Structural lattice reduction: generalized worst-case to average-case reductions and homomorphic cryptosystems," in *Advances in Cryptology – EUROCRYPT 2016*. Springer, 2016, pp. 528–558.

[120] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds," in *international conference on the theory and application of cryptology and information security*. Springer, 2016, pp. 3–33.

[121] ——, "Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 377–408.

[122] D. Micciancio and Y. Polyakov, "Bootstrapping in fhew-like cryptosystems," in *Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2021, pp. 17–28.

[123] Y. Lee, D. Micciancio, A. Kim, R. Choi, M. Deryabin, J. Eom, and D. Yoo, "Efficient FHEW Bootstrapping with Small Evaluation Keys,

and Applications to Threshold Homomorphic Encryption," *Cryptology ePrint Archive*, 2022.

[124] S. Carpov, N. Gama, M. Georgieva, and J. R. Troncoso-Pastoriza, "Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption," Cryptology ePrint Archive, Report 2019/101, 2019, https://eprint.iacr.org/2019/101.

[125] A. Guimarães, E. Borin, and D. F. Aranha, "Revisiting the functional bootstrap in TFHE," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 229–253, 2021.

[126] H. Chen, I. Chillotti, and Y. Song, "Multi-key homomorphic encryption from tfhe," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2019, pp. 446–472.

[127] M. Joye, "Guide to Fully Homomorphic Encryption over the [Discretized] Torus," *Cryptology ePrint Archive*, 2021.

[128] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "Bootstrapping for approximate homomorphic encryption," in *Advances in Cryptology – EUROCRYPT 2018*, J. B. Nielsen and V. Rijmen, Eds. Cham: Springer International Publishing, 2018, pp. 360–384.

[129] ——, "A full RNS variant of approximate homomorphic encryption," in *International Conference on Selected Areas in Cryptography – SAC 2018*. Springer, 2018, pp. 347–368.

[130] F. Boemer, A. Costache, R. Cammarota, and C. Wierzynski, "NGraph-HE2: A High-Throughput Framework for Neural Network Inference on Encrypted Data," in *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, ser. WAHC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 45–56.

[131] D. Kim and Y. Song, "Approximate homomorphic encryption over the conjugate-invariant ring," in *International Conference on Information Security and Cryptology*. Springer, 2018, pp. 85–102.

[132] A. Kim, A. Papadimitriou, and Y. Polyakov, "Approximate homomorphic encryption with reduced approximation error," in *Cryptographers' Track at the RSA Conference*. Springer, 2022, pp. 120–144.

[133] H. Chen, I. Chillotti, and Y. Song, "Improved bootstrapping for approximate homomorphic encryption," in *Advances in Cryptology – EUROCRYPT 2019*, Y. Ishai and V. Rijmen, Eds. Cham: Springer International Publishing, 2019, pp. 34–54.

[134] K. Han and D. Ki, "Better bootstrapping for approximate homomorphic encryption," in *Topics in Cryptology – CT-RSA 2020*, S. Jarecki, Ed. Cham: Springer International Publishing, 2020, pp. 364–390.

[135] J.-W. Lee, E. Lee, Y. Lee, Y.-S. Kim, and J.-S. No, "High-precision bootstrapping of RNS-CKKS homomorphic encryption using optimal minimax polynomial approximation and inverse sine function," in *Advances in Cryptology – EUROCRYPT 2021*. Springer, 2021, pp. 618–647.

[136] C. S. Jutla and N. Manohar, "Sine series approximation of the mod function for bootstrapping of approximate HE," in *Advances in Cryptology – EUROCRYPT 2022*. Springer, 2022, pp. 491–520.

[137] ——, "Modular lagrange interpolation of the mod function for bootstrapping of approximate HE," *Cryptology ePrint Archive*, 2020.

[138] Y. Lee, J.-W. Lee, Y.-S. Kim, and J.-S. No, "Near-optimal polynomial for modulus reduction using L2-norm for approximate homomorphic encryption," *IEEE Access*, vol. 8, pp. 144 321–144 330, 2020.

[139] J.-P. Bossuat, C. Mouchet, J. Troncoso-Pastoriza, and J.-P. Hubaux, "Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys," in *Advances in Cryptology – EUROCRYPT 2021*. Springer, 2021, pp. 587–617.

[140] J.-P. Bossuat, J. Troncoso-Pastoriza, and J.-P. Hubaux, "Bootstrapping for approximate homomorphic encryption with negligible failure-probability by using sparse-secret encapsulation," in *International Conference on Applied Cryptography and Network Security*. Springer, 2022, pp. 521–541.

[141] B. Li and D. Micciancio, "On the security of homomorphic encryption on approximate numbers," in *Advances in Cryptology – EUROCRYPT 2021*, A. Canteaut and F.-X. Standaert, Eds. Cham: Springer International Publishing, 2021, pp. 648–677.

[142] J. H. Cheon, S. Hong, and D. Kim, "Remark on the security of CKKS scheme in practice," *IACR Cryptol. ePrint Arch*, vol. 2020, p. 1581, 2020.

[143] C. Boura, N. Gama, M. Georgieva, and D. Jetchev, "CHIMERA: Combining ring-LWE-based fully homomorphic encryption schemes," Cryptology ePrint Archive, Report 2018/758. https://eprint.iacr.org/2018/758, Tech. Rep., 2018.

[144] "Idash privacy & security workshop 2018 - secure genome analysis competition," http://www.humangenomeprivacy.org/2018/index.html, Dec. 2018.

[145] W.-j. Lu, Z. Huang, C. Hong, Y. Ma, and H. Qu, "Pegasus: bridging polynomial and non-polynomial evaluations in homomorphic encryption," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 1057–1073.

[146] Y. Doröz, J. Hoffstein, J. Pipher, J. H. Silverman, B. Sunar, W. Whyte, and Z. Zhang, "Fully homomorphic encryption from the finite field isomorphism problem," in *IACR International Workshop on Public Key Cryptography*. Springer, 2018, pp. 125–155.

[147] A. Joux, "Fully homomorphic encryption modulo Fermat numbers," Cryptology ePrint Archive, Report 2019/187, 2019, https://eprint.iacr.org/2019/187.

[148] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.

[149] W. Castryck, I. Iliashenko, and F. Vercauteren, "On error distributions in ring-based LWE," *LMS Journal of Computation and Mathematics*, vol. 19, no. A, pp. 130–145, 2016.

[150] ——, "Provably weak instances of ring-lwe revisited," in *Advances in Cryptology – EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 147–167.

[151] C. Peikert, "How (not) to instantiate Ring-LWE," in *International Conference on Security and Cryptography for Networks*. Springer, 2016, pp. 411–430.

[152] M. R. Albrecht, "On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL," in *Advances in Cryptology – EUROCRYPT 2017*, J.-S. Coron and J. B. Nielsen, Eds. Cham: Springer International Publishing, 2017, pp. 103–129.

[153] E. Alkim, A. Roberto, B. Joppe, L. Ducas, d. l. P. Antonio, P. Schwabe, S. Douglas *et al.*, "NewHope: algorithm specifications and supporting documentation (2019)," 2019, https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[154] M. R. Albrecht and L. Ducas, "Lattice attacks on NTRU and LWE: A history of refinements," 2021.

[155] N. Bindel, J. Buchmann, F. Göpfert, and M. Schmidt, "Estimation of the hardness of the learning with errors problem with a restricted number of samples," *Journal of Mathematical Cryptology*, vol. 13, no. 1, pp. 47–67, 2019.

[156] M. R. Albrecht, "Lattice Estimator, Rebooted," 2021, https://martinralbrecht.wordpress.com/2021/12/21/lattice-estimator-rebooted/.

[157] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical programming*, vol. 66, no. 1-3, pp. 181–199, 1994.

[158] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coeficients," 1982.

[159] Y. Chen and P. Q. Nguyen, "BKZ 2.0: Better lattice security estimates," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2011, pp. 1–20.

[160] S. Bai, D. Stehlé, and W. Wen, "Measuring, simulating and exploiting the head concavity phenomenon in BKZ," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2018, pp. 369–404.

[161] M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, and M. Stevens, "The general sieve kernel and new records in lattice reduction," in *Advances in Cryptology – EUROCRYPT 2019*, Y. Ishai and V. Rijmen, Eds. Cham: Springer International Publishing, 2019, pp. 717–746.

[162] M. R. Albrecht, S. Bai, P.-A. Fouque, P. Kirchner, D. Stehlé, and W. Wen, "Faster enumeration-based lattice reduction: root hermite factor k^{1/(2k)} time k^{k/8+ o (k)}," in *Advances in Cryptology – CRYPTO 2020*. Springer, 2020, pp. 186–212.

[163] M. R. Albrecht, S. Bai, J. Li, and J. Rowell, "Lattice reduction with approximate enumeration oracles," in *Advances in Cryptology – CRYPTO 2021*. Springer, 2021, pp. 732–759.

[164] N. Gama and P. Q. Nguyen, "Predicting lattice reduction," in *Advances in Cryptology – EUROCRYPT 2008*, N. Smart, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 31–51.

[165] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Cryptographers' Track at the RSA Conference*. Springer, 2011, pp. 319–339.

[166] M. Liu and P. Q. Nguyen, "Solving BDD by enumeration: An update," in *Cryptographers' Track at the RSA Conference*. Springer, 2013, pp. 293–309.

[167] M. R. Albrecht, C. Cid, J.-C. Faugere, R. Fitzpatrick, and L. Perret, "On the complexity of the bkw algorithm on lwe," *Designs, Codes and Cryptography*, vol. 74, no. 2, pp. 325–354, 2015.

[168] Y. Chen, "Réduction deréseau et sécurié concrete du chiffrement completement homomorphe," Ph.D. dissertation, Paris 7, 2013.

[169] G. Hanrot, X. Pujol, and D. Stehlé, "Algorithms for the shortest and closest lattice vector problems," in *International Conference on Coding and Cryptology*. Springer, 2011, pp. 159–190.

[170] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[171] N. Gama, P. Q. Nguyen, and O. Regev, "Lattice enumeration using extreme pruning," in *Advances in Cryptology – EUROCRYPT 2010*, H. Gilbert, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 257–278.

[172] J. Buchmann, F. Göpfert, R. Player, and T. Wunderer, "On the hardness of lwe with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack," in *International Conference on Cryptology in Africa*. Springer, 2016, pp. 24–43.

[173] N. Howgrave-Graham, "A Hybrid Lattice-Reduction and Meet-in-the-Middle Attack Against NTRU," in *Advances in Cryptology - CRYPTO 2007*, A. Menezes, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 150–169.

[174] T. Wunderer, "On the security of lattice-based cryptography against lattice reduction and hybrid attacks," 2018.

[175] S. Bai and S. D. Galbraith, "Lattice decoding attacks on binary LWE," in *Australasian Conference on Information Security and Privacy*. Springer, Cham, 2014, pp. 322–337.

[176] R. Kannan, "Minkowski's convex body theorem and integer programming," *Mathematics of operations research*, vol. 12, no. 3, pp. 415–440, 1987.

[177] M. R. Albrecht, R. Fitzpatrick, and F. Göpfert, "On the efficacy of solving LWE by reduction to unique-SVP," in *International Conference on Information Security and Cryptology*. Springer, 2013, pp. 293–310.

[178] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-Quantum Key Exchange: A New Hope," in *Proceedings of the 25th USENIX Conference on Security Symposium*, ser. SEC'16. USA: USENIX Association, 2016, p. 327–343.

[179] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, "Revisiting the expected cost of solving uSVP and applications to LWE," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 297–322.

[180] S. Bai, S. Miller, and W. Wen, "A Refined Analysis of the Cost for Solving LWE via uSVP," in *International Conference on Cryptology in Africa*. Springer, 2019, pp. 181–205.

[181] D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi, "Lwe with side information: Attacks and concrete security estimation," in *Advances in Cryptology – CRYPTO 2020*, D. Micciancio and T. Ristenpart, Eds. Cham: Springer International Publishing, 2020, pp. 329–358.

[182] H. Chen, L. Chua, K. E. Lauter, and Y. Song, "On the concrete security of lwe with small secret." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 539, 2020.

[183] E. W. Postlethwaite and F. Virdia, "On the success probability of solving unique svp via bkz." in *Public Key Cryptography (1)*, 2021, pp. 68–98.

[184] J. H. Cheon, M. Hhan, S. Hong, and Y. Son, "A Hybrid of Dual and Meet-in-the-Middle Attack on Sparse and Ternary Secret LWE," *IEEE Access*, vol. 7, pp. 89 497–89 506, 2019.

[185] N. Howgrave-Graham, J. H. Silverman, and W. Whyte, "A Meet-in-the-Middle Attack on an NTRU Private key," Technical report, NTRU Cryptosystems, June 2003. Report, Tech. Rep., 2003.

[186] T. Espitau, A. Joux, and N. Kharchenko, "On a dual/hybrid approach to small secret LWE," in *International Conference on Cryptology in India*. Springer, 2020, pp. 440–462.

[187] S. Arora and R. Ge, "New algorithms for learning in presence of errors," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2011, pp. 403–415.

[188] Y. Elias, K. E. Lauter, E. Ozman, and K. E. Stange, "Provably weak instances of ring-lwe," in *Advances in Cryptology – CRYPTO 2015*, R. Gennaro and M. Robshaw, Eds. Berlin, Heidelberg: Springer, 2015, pp. 63–92.

[189] H. Chen, K. Lauter, and K. E. Stange, "Attacks on the search RLWE problem with small errors," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 665–682, 2017.

[190] ——, "Security considerations for galois non-dual rlwe families," in *International Conference on Selected Areas in Cryptography*. Springer, 2016, pp. 443–462.

[191] V. Lyubashevsky, C. Peikert, and O. Regev, "A Toolkit for Ring-LWE Cryptography," in *Advances in Cryptology – EUROCRYPT 2013*, T. Johansson and P. Q. Nguyen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 35–54.

[192] E. Crockett and C. Peikert, "$\Lambda o \lambda$: Functional Lattice Cryptography," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 993–1005.

[193] B. R. Curtis and R. Player, "On the feasibility and impact of standardising sparse-secret lwe parameter sets for homomorphic encryption," in *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2019, pp. 1–10.

[194] B. R. Curtis and M. Walter, "Estimating the Security of Homomorphic Encryption Schemes," 2021, https://medium.com/zama-ai/estimating-the-security-of-homomorphic-encryption-schemes-cb798f9378f.

[195] S. Laur, H. Lipmaa, and T. Mielikäinen, "Cryptographically private support vector machines," 08 2006, pp. 618–624.

[196] S. Park, J. Byun, J. Lee, J. H. Cheon, and J. Lee, "He-friendly algorithm for privacy-preserving svm training," *IEEE Access*, vol. 8, pp. 57 414–57 425, 2020.

[197] Y. Rahulamathavan, R. C.-W. Phan, S. Veluru, K. Cumanan, and M. Rajarajan, "Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 5, pp. 467–479, 2013.

[198] E. Makri, D. Rotaru, N. Smart, and F. Vercauteren, *EPIC: Efficient Private Image Classification (or: Learning from the Masters): The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4–8, 2019, Proceedings*, 01 2019, pp. 473–492.

[199] T. Graepel, K. Lauter, and M. Naehrig, "Ml confidential: Machine learning on encrypted data," vol. 7839, 11 2012, pp. 1–21.

[200] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 334–348.

[201] R. Bost, R. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," 01 2015.

[202] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," 05 2017, pp. 19–38.

[203] L. Aslett, P. Esperança, and C. Holmes, "Encrypted statistical machine learning: new privacy preserving methods," 08 2015.

[204] A. Khedr, G. Gulak, and V. Vaikuntanathan, "Shield: Scalable homomorphic implementation of encrypted data-classifiers," *IEEE Transactions on Computers*, vol. 65, pp. 1–1, 11 2015.

[205] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, 03 2017.

[206] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 201–210.

[207] Q. Zhang, L. Yang, Z. Chen, P. Li, and M. Deen, "Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 07 2017.

[208] A. Brutzkus, O. Elisha, and R. Gilad-Bachrach, "Low latency privacy preserving inference," 12 2018.

[209] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30 039–30 054, 2022.

[210] E. Hesamifard, D. Takabi, M. Ghasemi, and R. Wright, "Privacy-preserving machine learning as a service," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, pp. 123–142, 06 2018.

[211] A. Al Badawi, J. Chao, J. Lin, C. Mun, S. Jie, B. Tan, X. Nan, K. Aung, and V. Chandrasekhar, "The alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus," 11 2018.

[212] M. Blatt, A. Gusev, Y. Polyakov, and S. Goldwasser, "Secure large-scale genome-wide association studies using homomorphic encryption," *Proceedings of the National Academy of Sciences of the United States of America*, 2020.

[213] Y. Zhang and H. Zhu, "Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning," 07 2020.

[214] "Zama," https://zama.ai.

[215] "Intel," https://www.intel.com.

[216] "Ant group," https://www.antgroup.com/en.

[217] "Better together: Privacy-preserving machine learning powered by intel sgx and intel dl boost," https://www.intel.com/content/www/us/en/artificial-intelligence/posts/alibaba-privacy-preserving-machine-learning.html.

[218] "Duality technologies," https://dualitytech.com.

[219] "Darpa selects researchers to accelerate use of fully homomorphic encryption," https://www.darpa.mil/news-events/2021-03-08.

[220] CISCO, "The internet of things reference model," *white paper*, 2014.

[221] ——, "Fog computing and the internet of things: Extend the cloud to where the things are," *white paper*, 2015, https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf.

[222] ETSI, "Multi access edge computing (mec); terminology," *ETSI Industry Specification Group*, 2019, https://www.etsi.org/deliver/etsi_gs/MEC/001_099/001/02.01.01_60/gs_MEC001v020101p.pdf.

[223] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The internet of things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, 2016.

[224] M. Yannuzzi, F. van Lingen, A. Jain, O. L. Parellada, M. M. Flores, D. Carrera, J. L. Pérez, D. Montero, P. Chacin, A. Corsaro, and A. Olive, "A new era for cities with fog computing," *IEEE Internet Computing*, vol. 21, no. 2, pp. 54–67, 2017.

[225] Z. Erkin, J. R. Troncoso-pastoriza, R. Lagendijk, and F. Perez-Gonzalez, "Privacy-preserving data aggregation in smart metering systems: an overview," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 75–86, 2013.

[226] L. Zhu, M. Li, Z. Zhang, C. Xu, R. Zhang, X. Du, and N. Guizani, "Privacy-preserving authentication and data aggregation for fog-based smart grid," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 80–85, 2019.

[227] J. H. Cheon and J. Kim, "A hybrid scheme of public-key encryption and somewhat homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1052–1063, 2015.

[228] P. Hebborn and G. Leander, *IACR Transactions on Symmetric Cryptology*, vol. 2020, no. 3, pp. 46–86, Sep. 2020.

[229] P. Méaux, C. Carlet, A. Journault, and F.-X. Standaert, "Improved filter permutators for efficient fhe: Better instances and implementations," in *Progress in Cryptology – INDOCRYPT 2019*, F. Hao, S. Ruj, and S. Sen Gupta, Eds. Cham: Springer International Publishing, 2019, pp. 68–91.

[230] C. Dobraunig, L. Grassi, L. Helminger, C. Rechberger, M. Schofnegger, and R. Walch, "Pasta: A case for hybrid homomorphic encryption," Cryptology ePrint Archive, Report 2021/731, 2021, https://ia.cr/2021/731.

[231] D. Derler, S. Ramacher, and D. Slamanig, "Homomorphic proxy re-authenticators and applications to verifiable multi-user data aggregation," in *Financial Cryptography and Data Security*, A. Kiayias, Ed. Cham: Springer International Publishing, 2017, pp. 124–142.

[232] Y. Kawai, T. Matsuda, T. Hirano, Y. Koseki, and G. Hanaoka, "Proxy Re-Encryption That Supports Homomorphic Operations for Re-Encrypted Ciphertexts," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. 102, no. 1, pp. 81–98, Jan. 2019.

[233] C. Ma, J. Li, and W. Ouyang, "A homomorphic proxy re-encryption from lattices," 11 2016, pp. 353–372.

[234] S. Yasuda, Y. Koseki, R. Hiromasa, and Y. Kawai, "Multi-key homomorphic proxy re-encryption," in *Information Security*, L. Chen, M. Manulis, and S. Schneider, Eds. Cham: Springer International Publishing, 2018, pp. 328–346.

[235] Y. Polyakov, K. Rohloff, G. Sahu, and V. Vaikuntanathan, "Fast proxy re-encryption for publish/subscribe systems," *ACM Trans. Priv. Secur.*, vol. 20, no. 4, sep 2017. [Online]. Available: https://doi.org/10.1145/3128607

[236] D. Nuñez, I. Agudo, and J. López, "Ntrureencrypt: An efficient proxy re-encryption scheme based on ntru," *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, 2015.

[237] L. T. Phong, L. Wang, Y. Aono, M. H. Nguyen, and X. Boyen, "Proxy re-encryption schemes with key privacy from lwe," *IACR Cryptol. ePrint Arch.*, vol. 2016, p. 327, 2016.

[238] Z. Li, C. Ma, and D. Wang, "Towards multi-hop homomorphic identity-based proxy re-encryption via branching program," *IEEE Access*, vol. 5, pp. 16 214–16 228, 2017.

[239] L. Zengpeng, M. Chunguang, and W. Ding, "Achieving multi-hop PRE via branching program," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 45–58, 2020.

[240] J. Li, Z. Qiao, K. Zhang, and C. Cui, "A lattice-based homomorphic proxy re-encryption scheme with strong anti-collusion for cloud computing," *Sensors*, vol. 21, no. 1, 2021.

[241] F. Luo, S. Al-Kuwari, W. Susilo, and D. H. Duong, "Chosen-ciphertext secure homomorphic proxy re-encryption," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020.

[242] A. Cohen, "What about Bob? the inadequacy of CPA security for proxy reencryption," in *Public-Key Cryptography – PKC 2019*, D. Lin and K. Sako, Eds. Cham: Springer International Publishing, 2019, pp. 287–316.

[243] R. Canetti, S. Raghuraman, S. Richelson, and V. Vaikuntanathan, "Chosen-ciphertext secure fully homomorphic encryption," in *Public-Key Cryptography – PKC 2017*, S. Fehr, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 213–240.

[244] P. K. Shanu and K. Chandrasekaran, "Generating privacy-preserved recommendation using homomorphic authenticated encryption," in *2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2016, pp. 46–53.

[245] J. H. Cheon, K. Han, S.-M. Hong, H. J. Kim, J. Kim, S. Kim, H. Seo, H. Shim, and Y. Song, "Toward a secure drone system: Flying with real-time homomorphic authenticated encryption," *IEEE Access*, vol. 6, pp. 24 325–24 339, 2018.

[246] C. Joo and A. Yun, "Homomorphic authenticated encryption secure against chosen-ciphertext attack," in *Advances in Cryptology – ASIACRYPT 2014*, P. Sarkar and T. Iwata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 173–192.

[247] J. Kim and A. Yun, "Secure fully homomorphic authenticated encryption," *IEEE Access*, vol. 9, pp. 107 279–107 297, 2021.

[248] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: Signature schemes for network coding," in *Public Key Cryptography – PKC 2009*, S. Jarecki and G. Tsudik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 68–87.

[249] W. Chen, H. Lei, and K. Qi, "Lattice-based linearly homomorphic signatures in the standard model," *Theor. Comput. Sci.*, pp. 47–54, 2016.

[250] D. M. Freeman, "Improved security for linearly homomorphic signatures: A generic framework," in *Public Key Cryptography – PKC 2012*, M. Fischlin, J. Buchmann, and M. Manulis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 697–714.

[251] B. Libert, T. Peters, M. Joye, and M. Yung, "Linearly homomorphic structure-preserving signatures and their applications," in *Advances in Cryptology – CRYPTO 2013*, R. Canetti and J. A. Garay, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 289–307.

[252] N. Attrapadung and B. Libert, "Homomorphic network coding signatures in the standard model," in *Public Key Cryptography – PKC 2011*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 17–34.

[253] A. Esfahani, G. Mantas, and J. Rodriguez, "An efficient null space-based homomorphic mac scheme against tag pollution attacks in rlnc," *IEEE Communications Letters*, vol. 20, no. 5, pp. 918–921, 2016.

[254] D. Catalano, D. Fiore, and B. Warinschi, "Homomorphic signatures with efficient verification for polynomial functions?"

[255] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *Advances in Cryptology – EUROCRYPT 2011*, K. G. Paterson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 149–168.

[256] R. Hiromasa, Y. Manabe, and T. Okamoto, "Homomorphic signatures for polynomial functions with shorter signatures," 2013.

[257] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, "Leveled fully homomorphic signatures from standard lattices," in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 469–477.

[258] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators," in *Advances in Cryptology - ASIACRYPT 2013*, K. Sako and P. Sarkar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 301–320.

[259] X. Boyen, X. Fan, and E. Shi, "Adaptively secure fully homomorphic signatures based on lattices," Cryptology ePrint Archive, Report 2014/916, 2014.

[260] C. Wang, B. Wu, and H. Yao, "Leveled adaptively strong-unforgeable identity-based fully homomorphic signatures," *IEEE Access*, vol. 8, pp. 119 431–119 447, 2020.

[261] C. G. A. Nitulescu and E. Soria-Vazquez, "Rinocchio: Snarks for ring arithmetic," Cryptology ePrint Archive, Report 2021/322, 2021.

[262] D. Fiore, A. Nitulescu, and D. Pointcheval, "Boosting verifiable computation on encrypted data," Cryptology ePrint Archive, Report 2020/132, 2020.

[263] A. Bois, I. Cascudo, D. Fiore, and D. Kim, "Flexible and efficient verifiable computation on encrypted data," Cryptology ePrint Archive, Report 2020/1526, 2020.

[264] E. Soria-Vázquez, "Towards secure multi-party computation on the internet: Few rounds and many parties," Ph.D. dissertation, University of Bristol, 2019.

[265] A. Aly and M. Van Vyve, "Practically efficient secure single-commodity multi-market auctions," in *Financial Cryptography and Data Security*, J. Grossklags and B. Preneel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 110–129.

[266] M. A. Mustafa, S. Cleemput, A. Aly, and A. Abidin, "A secure and privacy-preserving protocol for smart metering operational data collection," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6481–6490, 2019.

[267] M. Keller, "MP-SPDZ: A versatile framework for multi-party computation," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1575–1590. [Online]. Available: https://doi.org/10.1145/3372297.3417872

[268] M. Keller, V. Pastro, and D. Rotaru, "Overdrive: Making spdz great again," in *Advances in Cryptology – EUROCRYPT 2018*, J. B. Nielsen and V. Rijmen, Eds. Cham: Springer International Publishing, 2018, pp. 158–189.

[269] A. Aly, E. Orsini, D. Rotaru, N. P. Smart, and T. Wood, "Zaphod: Efficiently Combining LSSS and Garbled Circuits in SCALE," in *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, ser. WAHC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 33–44.

[270] Y. Lindell, N. P. Smart, and E. Soria-Vazquez, "More efficient constant-round multi-party computation from bmr and she," in *Theory of Cryptography*, M. Hirt and A. Smith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 554–581.

[271] H. Chen, M. Kim, I. Razenshteyn, D. Rotaru, Y. Song, and S. Wagh, "Maliciously secure matrix multiplication with applications to private deep learning," in *Advances in Cryptology – ASIACRYPT 2020*, S. Moriai and H. Wang, Eds. Cham: Springer International Publishing, 2020, pp. 31–59.

[272] M. Keller, E. Orsini, and P. Scholl, "Mascot: Faster malicious arithmetic secure computation with oblivious transfer," Cryptology ePrint Archive, Report 2016/505, 2016.

[273] C. Hazay, P. Scholl, and E. Soria-Vazquez, "Low cost constant round mpc combining bmr and oblivious transfer," in *Advances in Cryptology – ASIACRYPT 2017*, T. Takagi and T. Peyrin, Eds. Cham: Springer International Publishing, 2017, pp. 598–628.

[274] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key fhe," in *Advances in Cryptology – EUROCRYPT 2016*, M. Fischlin and J.-S. Coron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 735–763.

[275] E. Kim, H.-S. Lee, and J. Park, "Towards round-optimal secure multiparty computations: Multikey fhe without a crs," *International Journal of Foundations of Computer Science*, vol. 31, no. 02, pp. 157–174, 2020.

[276] PALISADE, https://palisade-crypto.org.

[277] Lattigo, http://github.com/ldsec/lattigo.

[278] FHEW, https://github.com/lducas/FHEW.

[279] TFHE, https://github.com/tfhe/tfhe.

[280] Concrete, https://github.com/zama-ai/concrete.

[281] HEAAN, https://github.com/snucrypto/HEAAN.

[282] RNS-HEAAN, https://github.com/KyoohyungHan/FullRNS-HEAAN.

[283] FV-NFLlib, https://github.com/CryptoExperts/FV-NFLlib.

[284] cuFHE, https://github.com/vernamlab/cuFHE.

[285] nuFHE, https://github.com/nucypher/nufhe.

[286] A. A. Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, I. Quah, Y. Polyakov, S. R.V., K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca, "OpenFHE: Open-Source Fully Homomorphic Encryption Library," Cryptology ePrint Archive, Paper 2022/915, 2022, https://eprint.iacr.org/2022/915. [Online]. Available: https://eprint.iacr.org/2022/915

[287] S. Halevi and V. Shoup, "Algorithms in helib," in *Advances in Cryptology – CRYPTO 2014*, J. A. Garay and R. Gennaro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 554–571.

[288] V. Shoup, "NTL: A Library for doing Number Theory," 2016, [Online] https://libntl.org/.

[289] F. Boemer, S. Kim, G. Seifu, F. D. de Souza, V. Gopal *et al.*, "Intel HEXL (release 1.2)," https://github.com/intel/hexl, 2021.

[290] C. Mouchet, J.-P. Bossuat, J. Troncoso-Pastoriza, and J.-P. Hubaux, "Lattigo: A multiparty homomorphic encryption library in go," in *Proceedings of the 8th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, ser. WAHC '20, 2020.

[291] I. Chillotti, M. Joye, D. Ligier, J.-B. Orfila, and S. Tap, "CONCRETE: Concrete operates on ciphertexts rapidly by extending TfhE," in *Proceedings of the 8th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, ser. WAHC '20, vol. 15, 2020.

[292] NFLlib, https://github.com/quarkslab/NFLlib.

[293] W. Dai and B. Sunar, "cuhe: A homomorphic encryption accelerator library," in *International Conference on Cryptography and Information Security in the Balkans*. Springer, 2015, pp. 169–186.

[294] A. Viand, P. Jattke, and A. Hithnawi, "SoK: Fully Homomorphic Encryption Compilers," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2021, pp. 1092–1108.

[295] ALCHEMY, https://github.com/cpeikert/ALCHEMY.

[296] Cingulata, https://github.com/CEA-LIST/Cingulata.

[297] Encrypt-Everything-Everywhere, https://github.com/momalab/e3.

[298] SHEEP, https://github.com/alan-turing-institute/SHEEP.

[299] EVA, https://github.com/microsoft/EVA.

[300] Marble, https://github.com/MarbleHE/Marble.

[301] D. W. Archer, J. M. Calderón Trilla, J. Dagit, A. Malozemoff, Y. Polyakov, K. Rohloff, and G. Ryan, "RAMPARTS: A programmer-friendly system for building homomorphic encryption applications," in *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2019, pp. 57–68.

[302] Transpiler, https://github.com/google/fully-homomorphic-encryption.

[303] R. Dathathri, O. Saarikivi, H. Chen, K. Laine, K. Lauter, S. Maleki, M. Musuvathi, and T. Mytkowicz, "CHET: an optimizing compiler for fully-homomorphic neural-network inferencing," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2019, pp. 142–156.

[304] nGraph-HE2, https://github.com/IntelAI/he-transformer.

[305] T. van Elsloo, G. Patrini, and H. Ivey-Law, "SEALion: A framework for neural network inference on encrypted data," *arXiv preprint arXiv:1904.12840*, 2019.

[306] E. Crockett, "Simply safe lattice cryptography," Ph.D. dissertation, Georgia Institute of Technology, 2017.

[307] E. Crockett, C. Peikert, and C. Sharp, "Alchemy: A language and compiler for homomorphic encryption made easy," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1020–1037.

[308] "$\lambda \circ \lambda$," https://github.com/cpeikert/Lol.

[309] S. Carpov, P. Dubrulle, and R. Sirdey, "Armadillo: a compilation chain for privacy preserving applications," in *Proceedings of the 3rd International Workshop on Security in Cloud Computing*, 2015, pp. 13–19.

[310] D. Harvey and W. Hart, "FLINT: Fast Library for Number Theory," 2007, [Online] https://www.flintlib.org/index.html.

[311] The Sage Developers, *SageMath, the Sage Mathematics Software System*, `https://www.sagemath.org`.

[312] E. Chielle, O. Mazonka, N. G. Tsoutsos, and M. Maniatakos, "E3: A Framework for Compiling C++ Programs with Encrypted Operands." *IACR Cryptology ePrint Archive*, vol. 2018, p. 1013, 2018, [Online] https://eprint.iacr.org/2018/1013.pdf.

[313] R. Dathathri, B. Kostova, O. Saarikivi, W. Dai, K. Laine, and M. Musuvathi, "EVA: An encrypted vector arithmetic language and compiler for efficient homomorphic computation," in *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2020, pp. 546–561.

[314] A. Viand and H. Shafagh, "Marble: Making fully homomorphic encryption accessible to all," in *Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2018, pp. 49–60.

[315] S. Gorantala, R. Springer, S. Purser-Haskell, W. Lam, R. J. Wilson, A. Ali, E. P. Astor, I. Zukerman, S. Ruth, C. Dibak, P. Schoppmann, S. Kulankhina, A. Forget, D. Marn, C. Tew, R. Misoczki, B. Guillen, X. Ye, D. Kraft, D. Desfontaines, A. Krishnamurthy, M. Guevara, I. M. Perera, Y. Sushko, and B. Gipson, "A General Purpose Transpiler for Fully Homomorphic Encryption," *arXiv preprint arXiv:2106.07893*, 2021. [Online]. Available: https://arxiv.org/abs/2106.07893

[316] F. Boemer, Y. Lao, R. Cammarota, and C. Wierzynski, "nGraph-HE: a graph compiler for deep learning on homomorphically encrypted data," in *Proceedings of the 16th ACM International Conference on Computing Frontiers*, 2019, pp. 3–13.

[317] S. Cyphers, A. K. Bansal, A. Bhiwandiwalla, J. Bobba, M. Brookhart, A. Chakraborty, W. Constable, C. Convey, L. Cook, O. Kanawi *et al.*, "Intel nGraph: An intermediate representation, compiler, and executor for deep learning," *arXiv preprint arXiv:1801.08058*, 2018.

[318] F. Boemer, A. Costache, R. Cammarota, and C. Wierzynski, "nGraph-HE2: A high-throughput framework for neural network inference on encrypted data," in *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2019, pp. 45–56.

[319] Y. Doröz, E. Öztürk, and B. Sunar, "Accelerating fully homomorphic encryption in hardware," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1509–1521, 2014.

[320] D. B. Cousins, K. Rohloff, and D. Sumorok, "Designing an FPGA-accelerated homomorphic encryption co-processor," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 2, pp. 193–206, 2016.

[321] S. S. Roy, F. Vercauteren, N. Mentens, D. D. Chen, and I. Verbauwhede, "Compact ring-LWE cryptoprocessor," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2014, pp. 371–391.

[322] S. S. Roy, F. Turan, K. Jarvinen, F. Vercauteren, and I. Verbauwhede, "FPGA-based high-performance parallel architecture for homomorphic computing on encrypted data," in *2019 IEEE International symposium on high performance computer architecture (HPCA)*. IEEE, 2019, pp. 387–398.

[323] M. S. Riazi, K. Laine, B. Pelton, and W. Dai, "HEAX: An architecture for computing on encrypted data," in *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020, pp. 1295–1309.

[324] F. Turan, S. S. Roy, and I. Verbauwhede, "HEAWS: An Accelerator for Homomorphic Encryption on the Amazon AWS FPGA," *IEEE Transactions on Computers*, vol. 69, no. 8, pp. 1185–1196, 2020.

[325] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1651–1669.

[326] B. Reagen, W. Choi, Y. Ko, V. Lee, G.-Y. Wei, H.-H. S. Lee, and D. Brooks, "Cheetah: Optimizing and accelerating homomorphic encryption for private inference," in *Proceedings of the 27th IEEE international symposium on High Performance Computer Architecture (HPCA-27)*, 2021.

[327] A. Feldmann, N. Samardzic, A. Krastev, S. Devadas, R. Dreslinski, K. Eldefrawy, N. Genise, C. Peikert, and D. Sanchez, "F1: A Fast and Programmable Accelerator for Fully Homomorphic Encryption (Extended Version)," *arXiv preprint arXiv:2109.05371*, 2021.

[328] "Cryptolab," https://www.cryptolab.co.kr/.

[329] "Protect your accounts from data breaches with Password Checkup," 2019, [Online] https://security.googleblog.com/2019/02/protect-your-accounts-from-data.html.

BIOGRAPHIES

Chiara Marcolla is a senior researcher at the Cryptography Research Centre at the Technology Innovation Institute (TII). She specialized in the theory of affine-variety codes, using geometrical and algebraic methods such as the proprieties of Norm-Trace curves and Gröbner bases theory of zero-dimensional ideals. Her current research interests include rank-based post-quantum cryptography, searchable and fully homomorphic encryption. She received her Ph.D. degrees in Mathematics from the University of Trento (Italy) in 2013. Between 2013 and 2014, she was a postdoctoral researcher and she led the Cloud Security Team at the Department of Mathematics of the University of Trento. She was a postdoctoral researcher and had a research fellowship from the Department of Mathematics of the University of Turin (2015-2016).

Victor Sucasas received his Ph.D. on Electrical and Electronic Engineering from the 5G Innovation Centre at the University of Surrey (UK) in 2016. He was also a Marie Curie ESR at Instituto de Telecomunicações (Portugal) and a visiting researcher at University of Trento (Italy). In 2016 he became a postdoctoral researcher in network security and privacy preserving systems at Instituto de Telecomunicações where he participated in European projects ECSEL-SWARMs, CATRENE-H2O and ECSEL-SECREDAS. His research interests cover privacy-preserving authentication mechanisms, secure protocols, secure multi-party computation and privacy-preserving machine learning. He has also worked on wireless ad hoc networks, underwater networks, secure cognitive radio systems and physical layer security for wireless communications. Currently, he leads the Confidential Computing team at the Cryptography Research Centre at the Technology Innovation Institute (TII), Abu Dhabi, EAU. He is an IEEE and ComSoc member and an EAI Fellow.

Marc Manzano leads the Quantum Security group at Sandbox Quantum. Prior to that, he was a Senior Staff Software Engineer at Sandbox@Alphabet, a group within Google X devoted to quantum technologies and AI, where he focuses on research and development of quantum-secure communication solutions. His current research interests include post-quantum cryptography, lightweight cryptography, fully-homomorphic encryption, the intersection between machine learning and cryptanalysis, performance optimisations of cryptographic implementations on a wide range of architectures, and quantum algorithms. Over the past ten years, Dr Manzano has led the development of many secure cryptographic libraries and protocols. Dr Manzano was formerly the Vice President of the Cryptography Research Centre at the Technology Innovation Institute, a UAE-based scientific research centre. Prior to that, he held several positions where he was responsible for implementing pivotal cryptographic components of a variety of secure communication products, including an electronic voting platform. Dr Manzano holds a PhD in Computers Network Security, which he earned under the supervision of the University of Girona (Spain) and Kansas State University (United States). He earned an MSc in Computer Science from the University of Girona (Spain), while he did research stays at UC3M (Spain) and at DTU (Denmark). He initiated his research career while finalizing his BSc in Computer Engineering at Strathclyde University (UK).

Riccardo Bassoli is a senior researcher at the Deutsche Telekom Chair of Communication Networks, Faculty of Electrical and Computer Engineering, Technische Universität Dresden (Germany). He received his B.Sc. and M.Sc. degrees in Telecommunications Engineering from the University of Modena and Reggio Emilia (Italy) in 2008 and 2010 respectively. Next, he received his Ph.D. degree from the 5G Innovation Centre at the University of Surrey (UK), in 2016. He was also a Marie Curie ESR at the Instituto de Telecomunicações (Portugal) and visiting researcher at Airbus Defence and Space (France). Between 2016 and 2019, he was a postdoctoral researcher at the University of Trento (Italy). He is an IEEE and ComSoc member. He is also a member of Glue Technologies for Space Systems Technical Panel of IEEE AESS.

Frank H.P. Fitzek is a Professor and head of the Deutsche Telekom Chair of Communication Networks at the Technische Universität Dresden, coordinating the 5G Lab Germany. He is the spokesman of the DFG Cluster of Excellence CeTI. He received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in electrical engineering from the Technical University Berlin, Germany in 2002 and became Adjunct Professor at the University of Ferrara, Italy in the same year. In 2003 he joined Aalborg University as Associate Professor and later became Professor. In 2005 he won the YRP award for the work on MIMO MDC and received the Young Elite Researcher Award of Denmark. He was selected to receive the NOKIA Champion Award several times in a row from 2007 to 2011. In 2008 he was awarded the Nokia Achievement Award for his work on cooperative networks. In 2011 he received the SAPERE AUDE research grant from the Danish government and in 2012 he received the Vodafone Innovation prize. In 2015 he was awarded the honorary degree Doctor Honoris Causa from Budapest University of Technology and Economics (BUTE).

Najwa Aaraj is the Chief Researcher at the Cryptography Research Centre at the Technology Innovation Institute (TII). Her interests include post-quantum cryptography (PQC), lightweight cryptography, hardware cryptographic cores, privacy-preserving protocols, and applied machine learning for cryptographic technologies. She is also Acting Chief Researcher at TII's Autonomous Robotics Research Centre, which is dedicated to breakthrough developments in robotics and autonomy. Dr. Aaraj earned a PhD with Highest Distinction in Applied Cryptography and Embedded Systems Security from Princeton University (USA). She has extensive expertise in applied cryptography, trusted platforms, security architecture for embedded systems, software exploit detection and prevention systems, and biometrics. She has over 15 years of experience with global firms, working in multiple geographies from Australia to the United States. Prior to TII, Dr Aaraj assumed positions at DarkMatter, Booz & Company, IBM T.J. Watson Security Research, NY, Intel Research, Oregon, and NEC Laboratories, Princeton. Dr Aaraj has written multiple conference and journal papers, and received patents on applied cryptography, embedded system security, and machine learning-based protection of Internet of Things (IoT) systems. Dr Aaraj is on the advisory board of New York-based NeuTigers, a leading-edge startup revolutionising the next generation of energy/latency-efficient artificial intelligence (AI). She is also Adviser within the Strategic Advisory Group at Paladin Capital Group (Cyber Venture Capital) and Adjunct Professor at the Mohamed Bin Zayed Artificial Intelligence University (Machine Learning Research Group). In addition, she is an Adviser to multiple security and Machine Learning startups including Okinawa Institute of Science and Technology Graduate University.