Research Article

Massimo Giulietti, Paolo Martinelli, and Marco Timpanella*

Modern techniques in somewhat homomorphic encryption

https://doi.org/10.1515/jmc-2024-0041
received November 17, 2024; accepted January 13, 2025

Abstract: The term "homomorphism" was introduced in cryptography by Rivest, Adleman, and Dertouzos in 1978 to address performing calculations on encrypted data without decryption. Since then, researchers have increasingly aimed to design schemes supporting numerous operations. This article aims to synthesize the current state of the art in the so-called somewhat homomorphic encryption.

Keywords: homomorphic encryption, somewhat homomorphic encryption, learning with errors

MSC 2020: 94A60

1 Introduction

From a historical perspective, the term "homomorphism" was first introduced in cryptography by Rivest, Adleman, and Dertouzos in 1978 as a solution to the problem of performing calculations on encrypted data without necessarily decrypting it. Since then, an increasing number of researchers have sought to design such schemes to support as many operations as possible. These attempts can be essentially divided into three categories based on the type and the number of operations allowed on the encrypted data.

- *Partially homomorphic encryption (PHE)* schemes, which allow an unlimited number of operations of a single type (e.g., an unlimited number of additions or multiplications).
- *Somewhat homomorphic encryption (SWHE)* schemes, which allow various types of operations but only in a limited number.
- *Fully homomorphic encryption (FHE)* schemes, which allow an unlimited number of operations of any type.

The objective of this article is to provide a synthesis of the state of the art in SWHE schemes. For details on the development of FHE and PHE see the study by Acar et al. [1].

In this context, and more generally in the field of homomorphic cryptography, Gentry's work in 2009 marked a significant turning point. Gentry demonstrated how it is possible to construct an FHE scheme from an SWHE scheme using the bootstrapping technique, thereby giving a significant boost to the construction of new SWHE schemes. Prior to 2009, SWHE schemes in the literature were standalone algorithms, conceived as intermediate results leading toward the definition of an FHE scheme. However, since 2009, SWHE schemes have become an integral part of FHE schemes. Consequently, advancements in somewhat homomorphic cryptography after 2009 are often indistinguishable from those in fully homomorphic cryptography.

In this work, we first distinguish between SWHE schemes introduced before 2009, which we refer to as first-generation schemes, and schemes developed after Gentry's work. Regarding first-generation schemes, we

* **Corresponding author: Marco Timpanella,** Dipartimento di Matematica e Informatica, University of Perugia, Via Vanvitelli, 1 - 06123 Perugia, Italy, e-mail: marco.timpanella@unipg.it
Massimo Giulietti, Paolo Martinelli: Dipartimento di Matematica e Informatica, University of Perugia, Via Vanvitelli, 1 - 06123 Perugia, Italy

focus particularly on the BGN (Boneh, Goh, and Nissim) algorithm from 2005, considered a milestone in the development of somewhat homomorphic cryptography. Among the algorithms developed after 2009, considerable attention is given to schemes based on the well-known "learning with errors (LWEs)" problem (e.g., FV, Brakerski-Gentry-Vaikuntanathan (BGV), NTRU-like), which are currently considered among the most secure and promising. This research line also includes the Cheon-Kim-Kim-Song (CKKS) scheme, notable for its ability to operate on real and complex numbers (as opposed to integers, like previous examples), making it particularly suitable for applications such as machine learning.

Finally, a different approach to the SWHE problem is described, with a scheme based on integers and the Approximate-Greatest Common Divisor (AGCD) problem.

2 Classic SWHE schemes

In the literature on homomorphic encryption schemes, one of the earliest attempts at SWHE is represented by the Polly Cracker scheme [2]. This scheme allows both multiplication and addition operations on encrypted texts. However, the size of the encrypted text grows exponentially when performing such operations, making the scheme impractical from a practical standpoint. In particular, the multiplication operation turns out to be particularly costly. Subsequently, more efficient variants of this scheme have been proposed [3,4], but almost all of them have later been found to be vulnerable to attacks. Therefore, these early schemes are either insecure or impractical. Another attempt was made by Sander et al. [5] in 1999. This scheme also allows for evaluating both types of operation, but the size of the encrypted text increases multiplicatively with each multiplication. This limits the depth of circuits that can be homomorphically evaluated. Further improvements following the path taken by Sander, Young, and Yung were made by Ishai and Paskin in 2007 [6] and by Melchor et al. in 2008 [7], but without definitively overcoming the problem of the growth of the size of the encrypted text. The BGN algorithm (Boneh, Goh, Nissim) of 2005, on the other hand, overcomes these limitations, allowing homomorphic evaluation of an arbitrary number of additions plus one multiplication, while keeping the size of the encrypted text constant. Given the importance that the BGN algorithm has had in the development of homomorphic cryptography, in this article we will provide its explicit mathematical description.

3 BGN scheme

Introduced in 2005 by Boneh et al. [8], the BGN algorithm marks a milestone in the history of homomorphic cryptography. The security of the BGN scheme relies on the difficulty of the so-called Subgroup Decision Problem, which involves determining whether a certain element of a group $\mathbb{G}$ of order $n = p \cdot q$, with p and q distinct prime numbers, belongs to the subgroup of $\mathbb{G}$ with order p.

Below are the main phases of the BGN operation.

Given a security parameter $\kappa \in \mathbb{N}$, we choose two κ-bit primes p and q and compute $n = pq$. We consider:

- $\mathbb{G}$ and $\mathbb{G}'$ as cyclic groups of order n;
- g as a generator of $\mathbb{G}$;
- $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}'$ as a bilinear map such that $e(g, g)$ is a generator of $\mathbb{G}'$.

One possible construction of such groups for an $n > 3$ without square factors and not divisible by 3 is as follows. Let l be the smallest integer such that $p = ln - 1$ is prime and $p \equiv 2 \pmod 3$. Consider a supersingular elliptic curve $y^2 = x^3 + 1$. Since $p \equiv 2 \pmod 3$, the number of $\mathbb{F}_p$-rational points on this curve is $p + 1 = ln$, forming a group. Take $\mathbb{G}$ as the subgroup of order n of this group. Finally, if $\mathbb{G}'$ is the subgroup of $\mathbb{F}^*_{p^2}$ of order n, then the Weil pairing on the curve ensures the existence of the desired bilinear form.

Given $(p, q, \mathbb{G}, \mathbb{G}', e)$ and g as a generator of $\mathbb{G}$, we choose another random generator u of $\mathbb{G}$ and compute $h = u^q$, which will be a generator of the subgroup of $\mathbb{G}$ of order p. The public key is then $pk = (n, g, h, \mathbb{G}, \mathbb{G}', e)$, while the private key is $sk = p$.

Suppose we want to encrypt an element $m \in \mathbb{Z}_2$. We simply choose a random element r from $\mathbb{Z}_n$ and use the public key $pk = (n, g, h, \mathbb{G}, \mathbb{G}', e)$ to generate the encryption c of m, defined as follows:

$$c = g^m h^r \in \mathbb{G}.$$

To decrypt c, we use $sk = p$ to compute:

$$c^p = (g^m h^r)^p = (g^p)^m (h^p)^r = (g^p)^m, \quad \text{since } \operatorname{ord}(h) = p.$$

At this point, we simply calculate the discrete logarithm $m = \log_{g^p}(c^p)$, for example, using Pollard's lambda algorithm, to retrieve the message. Note that for this step to be feasible, the message space must be chosen small enough.

As mentioned earlier, BGN supports any number of homomorphic additions. Given $m, \tilde{m} \in \mathbb{Z}_2$ and their respective encryptions $c = E(m) = g^m h^r$, $\tilde{c} = E(\tilde{m}) = g^{\tilde{m}} h^{\tilde{r}}$, anyone can construct a valid encryption c_{add} for $m + \tilde{m}$ by choosing s randomly from $\mathbb{Z}_n$ and computing:

$$c_{\text{add}} = c\tilde{c}h^s = (g^m h^r)(g^{\tilde{m}} h^{\tilde{r}})h^s = (g^{m+\tilde{m}})h^{r_{\text{add}}},$$

where $r_{\text{add}} = r + \tilde{r} + s$. It is clear that the obtained result is indeed an encryption of $m + \tilde{m}$:

$$(c_{\text{add}})^p = (g^p)^{m+\tilde{m}} (h^p)^{r_{\text{add}}} = (g^p)^{m+\tilde{m}},$$

and we can calculate its discrete logarithm to obtain $m + \tilde{m}$.

Regarding homomorphic multiplication, the structure of BGN allows only one such operation through the bilinear map e. Let $g' = e(g, g) \in \mathbb{G}'$ and $h' = e(g, h) \in \mathbb{G}'$. By assumption, $ord(g') = n$, while, due to the bilinearity of e, $ord(h') = p$. Also, write $h = g^{\alpha q}$ for some $\alpha \in \mathbb{Z}$. Given two encrypted texts $c, \tilde{c} \in \mathbb{G}$ as mentioned earlier, we can compute the encryption c_{mult} of $m \cdot \tilde{m}$ by choosing $s \in \mathbb{Z}_n$ randomly and setting:

$$c_{\text{mult}} = e(c, \tilde{c})(h')^s = e(g^m h^r, g^{\tilde{m}} h^{\tilde{r}})(h')^s = (g')^{m\tilde{m}} (h')^{m\tilde{r} + \tilde{m}r + \alpha q r \tilde{r} + s} = (g')^{m\tilde{m}} (h')^{r_{\text{mult}}} \in \mathbb{G}'.$$

Clearly, as r is chosen, r_{mult} is also uniformly distributed in $\mathbb{Z}_n$. Thus, c_{mult} is an encryption for $m \cdot \tilde{m}$. However, now c_{mult} is in $\mathbb{G}'$ rather than $\mathbb{G}$, so although decryption and homomorphic addition are still possible, homomorphic multiplication is not.

4 SWHE schemes based on LWE

LWE, initially introduced by Oded Regev as an extension of the "learning from parity with error" problem, is considered one of the most challenging problems to solve even for quantum computers. Regev himself reduced the complexity of some of the most well-known lattice problems, such as shortest vector problem (SVP), to LWE. This implies that if an algorithm were found to efficiently solve LWE, the same algorithm would also solve SVP efficiently. Since then, LWE has become one of the most studied and promising problems in the field of post-quantum cryptography. In this section, we will delve into the workings of the main SWHE schemes based on LWE.

In the context of homomorphic cryptography, the fame of this problem is largely due to its use in constructing SWHE and FHE schemes. A fundamental step in this regard was taken in 2011 by Brakerski and Vaikuntanathan [9]. The description of the algorithm they constructed (called BV) is reported in Section 5.

Subsequently, in 2013, Lyubashevsky et al. [10] proposed a significant improvement of LWE, introducing a variant called ring-LWE (RLWE), which shifts the problem into the context of polynomial rings. Since then, several researchers have worked to transfer the BV algorithm into this new environment, with the aim of achieving performance improvements. In this direction, we particularly mention the work of Fan and Vercauteren (FV scheme) [11] and BGV scheme [12].

A completely different approach to building an SWHE scheme using well-known lattice problems was presented by López-Alt et al. in 2012 in [13], with an algorithm inspired by the famous NTRU. NTRU-Encrypt is

an example of a lattice-based cryptosystem that has already been extensively tested and strongly standardized, but whose homomorphic properties have only recently been recognized. In Section 8, we will describe this approach in more detail along with some encountered challenges.

4.1 Notations and LWE

Let S be a finite set and χ be a probability distribution over S. In the following, we will use the symbol $x \xleftarrow{\$} \chi$ to indicate that x is chosen from S according to the distribution χ. Instead, we write $x \xleftarrow{\$} S$ to indicate that x is chosen from S uniformly. Scalars and vectors will be denoted respectively in italic and bold (e.g., x is a scalar, $\mathbf{v}$ is a vector), while matrices will be indicated in uppercase bold. The dot product between two vectors, denoted by $\langle \mathbf{u}, \mathbf{v} \rangle$, is the usual dot product $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^T \cdot \mathbf{u}$. If $\mathbf{v}$ is a vector of dimension n, the ith component of $\mathbf{v}$ is indicated by $\mathbf{v}[i]$. By convention, $\mathbf{v}[0] \coloneqq 1$.

Now let's describe the LWE problem. Let n and q be two natural numbers with $q \geq 2$. Also, let $\mathbf{s} \in \mathbb{Z}_q^n$ be a vector and χ be a probability distribution over $\mathbb{Z}_q$. Consider the distribution $A_{\mathbf{s},\chi}$, which given a random vector $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and "noise" $e \xleftarrow{\$} \chi$, returns the pair $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. The $\mathbf{LWE}_{n,m,q,\chi}$ problem is thus defined as follows: given m independent samples from the distribution $A_{\mathbf{s},\chi}$ (for some $\mathbf{s} \in \mathbb{Z}_q^n$), output $\mathbf{s}$ with high probability.

The decisional version of this problem, denoted by $\mathbf{DLWE}_{n,m,q,\chi}$, instead consists of distinguishing m samples according to the distribution $A_{\mathbf{s},\chi}$ (where $\mathbf{s}$ is chosen uniformly from $\mathbb{Z}_q^n$), from m samples uniformly drawn from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

For cryptographic applications, the DLWE problem is particularly interesting. Indeed, reductions (both classical and quantum) to lattice problems considered difficult to solve are known. It is not the objective of this report to delve into the nature and complexity of such lattice problems, it is sufficient to recall that the most efficient algorithms for solving these problems have an almost exponential complexity with respect to the lattice dimension, and for this reason, the cryptosystems based on them are considered secure.

5 BV scheme

The underlying idea behind Gentry's construction is to exploit the difficulty of certain problems related to the so-called ideal lattices (a particular family of lattices). Ideals are mathematical objects that naturally appear in the context of homomorphic cryptography, as they are closed by definition under both addition and multiplication (while a generic lattice is closed only under addition). Despite the widespread use of lattices in cryptography, ideal lattices are a specific category about which we know relatively little. After Gentry, several constructions of FHE have been based on his approach, but although promising, these early "second-generation" schemes have never truly become practical.

In the study by Brakerski and Vaikuntanathan [9], however, the authors introduced a scheme, which we will refer to as BV from now on, whose security is based on generic lattice problems (not necessarily ideal lattices). In particular, the security of this system is solely based on the complexity of LWE, a problem that is widely studied and considered "reliable." Furthermore, the definition of this scheme does not directly depend on lattices, making it easier to understand and implement.

5.1 Parameters

BV is a public-key SWHE scheme with plaintext space $\mathbb{Z}_2$. The scheme has parameters consisting of two natural numbers n and m, an odd modulus q, and an error distribution χ over $\mathbb{Z}_q$. Let $\kappa \in \mathbb{N}$ be a security parameter

and $L \in \mathbb{N}$ be a parameter indicating the maximum number of homomorphic multiplications the scheme can perform. A possible choice of parameters is as follows: n polynomial in κ, $m \geq n \log q + 2\kappa$ polynomial in n, $q \in [2^{n^\varepsilon}, 2 \cdot 2^{n^\varepsilon})$, where $\varepsilon \in (0,1)$ is a constant (thus q is subexponential in n), and $L \approx \varepsilon \log n$.

5.2 Key generation

First, we extract $L+1$ vectors $\mathbf{s}_0, \ldots, \mathbf{s}_L \xleftarrow{\$} \mathbb{Z}_q^n$. For each $\ell \in \{1, \ldots, L\}$, for each i, j with $0 \leq i \leq j \leq n$, and for each $\tau \in \{0, \ldots, \lfloor \log q \rfloor\}$, we choose

$$\mathbf{a}_{\ell,i,j,\tau} \xleftarrow{\$} \mathbb{Z}_q^n, \qquad e_{\ell,i,j,\tau} \xleftarrow{\$} \chi$$

and consider the pairs $\psi_{\ell,i,j,\tau} \coloneqq (\mathbf{a}_{\ell,i,j,\tau}, b_{\ell,i,j,\tau}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where

$$b_{\ell,i,j,\tau} \coloneqq \langle \mathbf{a}_{\ell,i,j,\tau}, \mathbf{s}_\ell \rangle + 2 \cdot e_{\ell,i,j,\tau} + 2^\tau \cdot \mathbf{s}_{\ell-1}[i] \cdot \mathbf{s}_{\ell-1}[j],$$

where by convention $\mathbf{s}_{\ell-1}[0] \triangleq 1$. We then define $\Psi \coloneqq \{\psi_{\ell,i,j,\tau}\}$ as the set of these pairs. As the last step, the algorithm involves choosing a uniformly random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, a vector of errors $\mathbf{e} \xleftarrow{\$} \chi^m$, and computing $\mathbf{b} \coloneqq \mathbf{A}\mathbf{s}_0 + 2\mathbf{e} \in \mathbb{Z}_q^m$.

At this point, the private key is $sk = \mathbf{s}_L$, the public key is $pk = (\mathbf{A}, \mathbf{b})$, while $evk = \Psi$ (public) is required for homomorphically evaluating functions on ciphertexts. The key generation function, $BV.\ KeyGen(1^\kappa)$, thus takes as input the unary representation of the security parameter k and outputs the public key for encryption, the public key for homomorphic evaluation, and the private key for decryption.

5.3 Encryption

To encrypt a message $\mu \in \mathbb{Z}_2$, we choose a random vector $\mathbf{r} \xleftarrow{\$} \{0,1\}^m$ and, using the public key $pk = (\mathbf{A}, \mathbf{b})$, compute

$$\mathbf{v} \coloneqq \mathbf{A}^T\mathbf{r} \in \mathbb{Z}_q^n, \qquad w \coloneqq \mathbf{b}^t\mathbf{r} + \mu \in \mathbb{Z}_q.$$

The encryption of μ contains, in addition to the pair $(\mathbf{v}, w)$, a label that serves to keep track of the number of multiplications already performed. Consequently, for the first encryption, the label will be equal to 0.

Formally, the encryption algorithm $BV.\ Enc_{pk}(\mu)$ takes as input a message $\mu \in \mathbb{Z}_2$ and returns the ciphertext $c \coloneqq ((\mathbf{v}, w), 0)$ for a message encrypted for the first time.

5.4 Homomorphic evaluation of functions

Let $f : \mathbb{Z}_2^t \to \mathbb{Z}$. Suppose f is represented by a binary arithmetic circuit with "+" gates and "×" gates. The "+" gates can have an arbitrary number of inputs, while the "×" gates can only have two inputs. Furthermore, we require that the circuit be "layered," in the sense that it must be composed of homogeneous levels consisting of either only addition gates or only multiplication gates (every arithmetic circuit can be written in this form). Finally, we require that the depth of the circuit with respect to multiplication, i.e., the total number of "×" layers, be exactly L. The function is evaluated homomorphically, so it is sufficient to analyze two cases: the addition of any number of ciphertexts and the multiplication of two ciphertexts.

During the evaluation of the function, the ciphertexts will be of the form $c = ((\mathbf{v}, w), \ell)$, where ℓ is a label indicating the number of multiplications already performed. The requirement that the circuit be layered ensures that at each level, the various inputs of each gate have the same label. Finally, we will verify each time that the output of any gate $c = ((\mathbf{v}, w), \ell)$ satisfies the condition

$$w - \langle \mathbf{v}, \mathbf{s}_\ell \rangle = \mu + 2 \cdot e \pmod{q}, \tag{1}$$

where μ is the plaintext corresponding to the output of the gate, and e is an error (also called noise) that depends on the inputs of the gate. This condition will be necessary for the decryption phase.

The evaluation of an addition homomorphically takes as input a certain number of ciphertexts $c_1, \dots, c_t$, where $c_i = ((\mathbf{v}_i, w_i), \ell)$, and returns

$$c_{\text{add}} = ((\mathbf{v}_{\text{add}}, w_{\text{add}}), \ell) \coloneqq \left(\left(\sum_i \mathbf{v}_i, \sum_i w_i\right), \ell\right).$$

We can observe that:

$$w_{\text{add}} - \langle \mathbf{v}_{\text{add}}, \mathbf{s}_\ell \rangle = \sum_i (w_i - \langle \mathbf{v}_i, \mathbf{s}_\ell \rangle) = \sum_i (\mu_i + 2e_i) = \sum_i \mu_i + 2\sum_i e_i,$$

where μ_i is the plaintext corresponding to c_i, i.e., the sum of the ciphertexts is the encryption of the sum of the plaintexts, taking the sum of the errors.

We will now show how multiplication between two ciphertexts $c = ((\mathbf{v}, w), \ell)$, $c' = ((\mathbf{v}', w'), \ell')$ results in a ciphertext $c_{\text{mult}} = ((\mathbf{v}_{\text{mult}}, w_{\text{mult}}), \ell + 1)$.

Consider the following polynomial:

$$\phi(\mathbf{x}) = \phi_{c,c'}(\mathbf{x}) = (w - \langle \mathbf{v}, \mathbf{x} \rangle) \cdot (w' - \langle \mathbf{v}', \mathbf{x} \rangle)$$

in the variables $\mathbf{x} = (\mathbf{x}[1], \dots, \mathbf{x}[n])$. Being a quadratic polynomial, it can be rewritten as follows:

$$\phi(\mathbf{x}) = \sum_{0 \le i \le j \le n} h_{i,j} \cdot \mathbf{x}[i] \cdot \mathbf{x}[j].$$

Now, consider the binary representation of the coefficients $h_{i,j}$, denoting by $h_{i,j,\tau}$ the τth bit of its representation, i.e.,

$$h_{i,j} = \sum_{\tau=0}^{\lfloor \log q \rfloor} h_{i,j,\tau} \cdot 2^\tau.$$

Then,

$$\phi(\mathbf{x}) = \sum_{\substack{0 \le i \le j \le n \\ \tau \in 0, \dots, \lfloor \log q \rfloor}} h_{i,j,\tau} \cdot (2^\tau \cdot \mathbf{x}[i] \cdot \mathbf{x}[j]).$$

Now, recalling that the evaluation key $evk = \Psi$ is composed by pairs $\psi_{\ell,i,j,\tau} = (\mathbf{a}_{\ell,i,j,\tau}, b_{\ell,i,j,\tau})$ such that

$$2^\tau \mathbf{s}_\ell[i] \mathbf{s}_\ell[j] \approx b_{\ell+1,i,j,\tau} - \langle \mathbf{a}_{\ell+1,i,j,\tau}, \mathbf{s}_{\ell+1} \rangle$$

up to an error $2e_{\ell+1,i,j,\tau}$, the algorithm for the homomorphic multiplication of c and c' consists in computing

$$\mathbf{v}_{\text{molt}} \coloneqq \sum_{\substack{0 \le i \le j \le n \\ \tau \in 0, \dots, \lfloor \log q \rfloor}} h_{i,j,\tau} \cdot \mathbf{a}_{\ell+1,i,j,\tau}, \quad \text{and} \quad w_{\text{molt}} \coloneqq \sum_{\substack{0 \le i \le j \le n \\ \tau \in 0, \dots, \lfloor \log q \rfloor}} h_{i,j,\tau} \cdot b_{\ell+1,i,j,\tau},$$

and let $c_{\text{molt}} \coloneqq ((\mathbf{v}_{\text{molt}}, w_{\text{molt}}), \ell + 1)$. It can be verified that property (1) still holds for the ciphertext c_{mult}. The error of the new message depends both on the errors of the ciphertexts c, c' and the evaluation key.

5.4.1 Decryption

Suppose the multiplicative depth of the circuit is L. It is necessary to decrypt only ciphertexts of the form $c = (\mathbf{v}, w, L)$. To perform this step, it is sufficient to compute

$$((w - \langle \mathbf{v}, \mathbf{s}_L \rangle) \bmod q) (\bmod 2),$$

since, by (1),

$$(w - \langle \mathbf{v}, \mathbf{s}_L \rangle) (\bmod q) = \mu + 2e \ (\bmod q) \overset{e \ll q}{=} \mu + 2e.$$

Consequently, the decryption phase requires an error e negligible compared to q. In the study by Brakerski and Vaikuntanathan [9], the potential growth of errors during the various stages of the algorithm is considered, and it is demonstrated that, under reasonable assumptions, it is possible to achieve a multiplicative depth L on the order of $\varepsilon \log n$ with $\varepsilon < 1$.

6 Schemes based on RLWE

One of the most important algorithms that managed to adapt the Brakerski and Vaikuntanathan scheme and bring it into the context of the RLWE problem was described by Fan and Vercauteren in [11]. In this way, it was possible to obtain an SWHE scheme with a smaller key space compared to BV and faster computations. Since this scheme is essentially an adaptation of the BV algorithm, we will not provide the explicit details of the individual phases, but we will only describe the new setting in which it operates.

The mathematical environment in which Fan and Vercauteren define their scheme is the ring of polynomials $R = \mathbb{Z}[x]/(f(x))$, where $f(x) \in \mathbb{Z}[x]$ is a monic irreducible polynomial of degree d. Usually, $f(x)$ is the minimal polynomial of a primitive mth root of unity, namely, a cyclotomic polynomial $\phi_m(x)$. The most common choice is $f(x) = x^d + 1$ with $d = 2^n$.

Let $q > 1$ be an integer, and let Z_q be the set of integers in the interval $(-q/2, q/2]$ (not to be confused with the ring $\mathbb{Z}_q$). Consider the set R_q of polynomials in R with coefficients in Z_q. All arithmetic of the FV scheme will take place in the ring R and the set R_q. Given $a \in \mathbb{Z}$, denote by $[a]_q$ the unique integer in Z_q such that $[a]_q \equiv a \pmod q$. Similarly, if $\mathbf{a} \in R$, let $[\mathbf{a}]_q$ be the element of R obtained by applying $[\cdot]_q$ to all its coefficients.

As in the scheme of Brakerski and Vaikuntanathan, to define the RLWE problem, it is necessary to consider a distribution over R. The basic idea is to use the Gaussian distribution over integers to define a distribution over R. The most natural approach to do this is simply to select polynomials in R by choosing coefficients according to the Gaussian distribution. This approach is not applicable in the general case, but it works if $f(x) = x^d + 1$ with $d = 2^n$. The RLWE problem, in its decisional version, can then be stated as follows.

Definition 1. Let $f(X)$ be a cyclotomic polynomial $\phi_m(x)$ of degree $\varphi(m)$. Let also $R = \mathbb{Z}[x]/(f(x))$ and $q \geq 2$. Given $\mathbf{s} \in R_q$ random and a distribution χ over R, let $A^{(q)}_{\mathbf{s},\chi}$ be the distribution obtained by uniformly choosing $\mathbf{a} \in R_q$ and an error term $e \leftarrow \chi$, and outputting the pair $(\mathbf{a}, [\mathbf{a} \cdot \mathbf{s} + e]_q)$. The decisional RLWE problem is therefore to distinguish between the distribution $A^{(q)}_{\mathbf{s},\chi}$ and the uniform distribution.

The RLWE problem can be reduced (via a quantum algorithm) to the SVP problem on an ideal lattice. Moreover, it is possible to choose $\mathbf{s} \in R_q$ according to the distribution χ and not necessarily uniformly, without compromising security. Finally, it has been shown that the complexity of the problem does not depend on the exact form of q, which can therefore be chosen simply as a power of 2.

Another significant step toward a practical SWHE scheme is the BGV scheme (Brakersi-Gentry-Vaikuntanathan), introduced in [12]. This scheme, like FV, bases its security on the RLWE problem, but unlike its predecessors, it significantly improves performance thanks to a noise management technique called “modulus switching.” The setting from which Brakersi et al. start is the same as the FV scheme. As seen, in order to decrypt a ciphertext encrypted with the BV scheme (or equivalently with the FV scheme), it is necessary for the accumulated noise during homomorphic operations to be less than $q/2$. Furthermore, the sum of two ciphertexts with noise less than or equal to B results in a ciphertext whose noise is at most $2B$. Multiplication of the same ciphertexts would result in a ciphertext with an error in the order of B^2. By iteratively using the modulus switching technique, the BGV scheme manages to keep the noise level substantially constant, provided that the modulus size (q according to previous notation) is sacrificed and therefore gradually sacrificing the remaining homomorphic capabilities of the system. The essence of modulus switching can be summarized in the following lemma. The notations are the same used before.

Lemma 1. *Let p and q be two odd natural numbers and c be a vector of integers. Let $\mathbf{c}'$ be the vector of integers closest (with respect to a fixed norm) to $(p/q)\cdot \mathbf{c}$ and such that $\mathbf{c}' = \mathbf{c} \pmod 2$. Then, for any $\mathbf{s}$ such that $|[\langle \mathbf{c}, \mathbf{s}\rangle]_q| < q/2 - (q/p)\ell_1(\mathbf{s})$, it holds that*

$$[\langle \mathbf{c}', \mathbf{s}\rangle]_p = [\langle \mathbf{c}, \mathbf{s}\rangle]_q \pmod 2 \quad \textit{and} \quad |[\langle \mathbf{c}', \mathbf{s}\rangle]_p| < (p/q)\cdot |[\langle \mathbf{c}, \mathbf{s}\rangle]_q| + \ell_1(\mathbf{s}),$$

where $\ell_1(\mathbf{s})$ is the ℓ_1 norm of $\mathbf{s}$.

In short, this lemma ensures that a user who does not know the secret key $\mathbf{s}$, but knows a bound on its norm, can transform a ciphertext c modulo q into a different ciphertext c' modulo p while preserving the correctness of the scheme, i.e., $[\langle \mathbf{c}', \mathbf{s}\rangle]_p = [\langle \mathbf{c}, \mathbf{s}\rangle]_q \pmod 2$. This transformation simply involves scaling by a factor p/q and appropriately approximating. Furthermore, if the norm of s is sufficiently small, and p is sufficiently small compared to q, then the noise in the ciphertext is reduced. With this technique, a user can control the noise during the execution of the scheme, without knowing the secret key and without the need to introduce a public homomorphic evaluation key as in the BV scheme.

7 CKKS scheme

All the SWHE schemes seen so far are built to naturally support arithmetic of integers or binaries. However, since for various applications, such as machine learning, it is necessary to operate on real/complex numbers, there is a need for a scheme that can homomorphically handle approximated data. It is worth noting that even the FV and BGV schemes described previously are capable of evaluating homomorphic operations on real numbers, but this requires the introduction of sophisticated and often inefficient encoding procedures. The CKKS algorithm, introduced in [14], is instead an SWHE algorithm specifically designed to deal with this type of data and is considered the most efficient SWHE algorithm for applications involving such data.

From a formal point of view, the operation of CKKS has many similarities with other RLWE-based schemes, especially with BGV. However, in terms of security, there is an important difference between CKKS and BGV. In fact, Li and Micciancio [15] exhibited an attack against CKKS that exploits its decryption function, whose approximate results weaken the complexity of the RLWE problem. In the best-case scenario, through a simple algebraic manipulation, this attack allows the secret key to be recovered in a single attempt. Both FV and BGV are secure against this attack. In response to this vulnerability, Li and Micciancio [15] therefore proposed modifying the encryption function of CKKS, adding additional noise to the ciphertexts, so as not to weaken the underlying RLWE problem.

8 NTRU-based schemes

NTRU-Encrypt is a cryptosystem proposed by Hoffstein et al. [16] in 1998, and is one of the first attempts at lattice-based cryptography. Compared to RSA and GGH (Goldreich–Goldwasser–Halevi) cryptosystems, NTRU shows greater efficiency both in terms of hardware and software implementation. However, for 15 years, until the work of Stehlé and Steinfeld [17], there were doubts about its actual security. In this work, the authors succeeded, by introducing modifications to the original algorithm, in reducing the security of NTRU to the RLWE problem. This increased researchers' interest in this scheme significantly. Furthermore, in 2012, both López-Alt et al. and Gentry independently observed how the NTRU scheme naturally possesses homomorphic properties. In particular, in [13], López-Alt et al. propose modifications to the classic NTRU algorithm to obtain an SWHE scheme. The mathematical environment in which this scheme operates is the same as that of the FV and BGV schemes. Let $R = \mathbb{Z}[x]/(x^d + 1)$, where d is a power of 2, q is an odd prime number, and χ is a "B-bounded" distribution over R, i.e., the coefficients of the polynomials selected through the distribution χ must be less than B, where B is a relatively small number compared to q. Let Z_q be the set of integers in the

interval $(-q/2, q/2]$ and R_q be the set of polynomials in R with coefficients in Z_q. As mentioned earlier, given a polynomial $\mathbf{a} \in R$, $[a]_q$ denotes the element of R obtained by applying the reduction $[\cdot]_q$ to all its coefficients.

For key generation, the algorithm involves selecting two polynomials f' and g through the distribution χ, and setting $f \coloneqq 2f' + 1$, so that $f \equiv 1 \pmod 2$. Let f^{-1} be the inverse of f in R_q (if f were not invertible in R_q, the algorithm involves reselecting f'). The public key of the scheme is the polynomial $h \coloneqq [2gf^{-1}]_q \in R_q$, while the private key is f. To encrypt a message $m \in \mathbb{Z}_2$, it is necessary to select two polynomials s and e through χ and compute

$$c \coloneqq E(m) = [hs + 2e + m]_q \in R_q.$$

To decrypt the ciphertext c using the secret key f, it is sufficient to compute $[fc]_q \pmod 2$. In fact,

$$[fc]_q = [fhs + 2fe + fm]_q = [2gs + 2fe + fm]_q.$$

At this point if we assume that $[2gs + 2fe + fm]_q = 2gs + 2fe + fm$, then

$$[fc]_q \pmod 2 = 2gs + 2fe + fm \pmod 2 = fm \pmod 2 = m.$$

By choosing the parameters appropriately, it is possible to ensure that decryption actually works correctly (i.e., there is no reduction modulo q in the calculation of fc). As is easy to observe, this scheme possesses homomorphic properties. However, with each operation on the ciphertexts, the error increases (especially in the case of multiplications), thus imposing a limit on the depth of the circuit to be evaluated.

The scheme proposed by López-Alt et al. also represents a new type of SWHE, called "Multikey SWHE," as it has the ability to evaluate ciphertexts with different and independent keys. In other words, each user can encrypt data with their own public key, and a third party can perform operations homomorphically on all these data. The only interaction required among the various users is the sharing of a common secret key, which depends on all the individual secret keys and which will then be used to decrypt the encrypted data. Suppose, for example, there are two users encrypting two messages m_1, m_2 using their own public keys $h_i \coloneqq [2g_i f_i^{-1}]_q$, obtaining the two ciphertexts

$$c_i = E_i(m_i) = [h_i s_i + 2e_i + m_i],$$

$i = 1, 2$. Let $f_{1,2} \coloneqq f_1 \cdot f_2$ be the common secret key of the two users. It is then possible to decrypt the two ciphertexts $c_1 + c_2$ and $c_1 \cdot c_2$ using the key $f_{1,2}$ by calculating

$$[f_{1,2}(c_1 + c_2)]_q = [2f_1 f_2 e_1 + 2f_1 f_2 e_2 + 2f_2 g_1 s_1 + 2f_1 g_2 s_2 + f_1 f_2 (m_1 + m_2)]_q$$

and

$$[f_{1,2} c_1 \cdot c_2]_q = [4g_1 g_2 s_1 s_2 + 2g_1 s_1 f_2 (2e_2 + m_2) + 2g_2 s_2 f_1 (2e_1 + m_1) + 2f_1 f_2 (e_1 m_2 + e_2 m_1 + 2e_1 e_2) + f_1 f_2 (m_1 m_2)]_q.$$

Therefore, since $f_1 = f_2 = 1 \pmod 2$, assuming that the error associated with these two ciphertexts has not exceeded the maximum threshold of $q/2$, we have that

$$[f_{1,2}(c_1 + c_2)]_q \pmod 2 = m_1 + m_2$$

and

$$[f_{1,2}(c_1 \cdot c_2)]_q \pmod 2 = m_1 \cdot m_2.$$

Obviously, in this case too, the error in the ciphertexts increases with each operation, and therefore, the correctness of decryption will only hold for a limited number of operations. What López-Alt et al. demonstrate in [13] is that the scheme can correctly evaluate circuits of approximately depth $\varepsilon \log(n)$, with $q = 2^{n^\varepsilon}$ and B polynomial in n.

Despite the efficiency properties derived from NTRU, and the property of being a multikey scheme, some limitations have prevented this scheme from asserting itself compared to others based on LWE. One of the problems to consider regarding the scheme by López-Alt et al. are its starting assumptions, namely, the complexity of the RLWE and DSPR (decisional small polynomial ratio) problems. While RLWE has been widely

studied and is now considered a standard, the assumption about the complexity of DSPR is less reassuring. For this reason, several researchers have tried to modify the scheme to remove the dependence on DSPR, but these attempts have resulted in the reintroduction of public evaluation keys and other complicated procedures, thus making the scheme inefficient. By reintroducing the DSPR problem, it is possible to obtain seemingly usable versions of these schemes. However, in 2016, Albrecht et al. in [18] showed an attack (based on lattices) on every SWHE scheme based on NTRU that uses the DSRP assumption. As a consequence, when the modulus q of these schemes is large compared to the dimension n, the scheme is not secure. At the same time, to increase the depth of the evaluable circuits, it is necessary to consider larger moduli, and consequently larger dimensions (to make the system secure). This problem has made SWHE schemes based on NTRU noncompetitive compared to other schemes based on RLWE. Research in this area is therefore trying to overcome this problem, for example, by further reducing the growth of errors during circuit evaluation, in order to keep the ratio between modulus and scheme dimension relatively low [19].

9 Somewhat homomorphic schemes based on integers

In 2010, 1 year after the famous Gentry scheme, van Dijk et al. [20] presented a new SWHE scheme whose security is based on the AGCD problem, namely, the problem of recovering an odd integer p, knowing a set of values $x_i = q_i p + r_i$. Consequently, the scheme is defined over integers, and one of its strengths is certainly its conceptual simplicity. The various phases of this scheme can be summarized as follows. Let p be a sufficiently large prime number representing the secret key of the scheme. Given a message $m \in \mathbb{Z}_2$, choose a sufficiently large prime number q and a number r sufficiently small compared to p. Then m can be encrypted as $c = E(m) = m + 2r + pq$. If $m + 2r < p/2$, the ciphertext can be decrypted simply by computing $m = (c \pmod p) \pmod 2$. Therefore, as in the previous schemes, the presence of an "error" term means that the number of performable homomorphic operations is not unlimited. The homomorphic properties of the scheme can be observed as follows. With respect to addition, it holds that

$$E(m_1) + E(m_2) = m_1 + 2r_1 + pq_1 + m_2 + 2r_2 + pq_2 = (m_1 + m_2) + 2(r_1 + r_2) + (q_1 + q_2)p.$$

Clearly, $E(m_1) + E(m_2)$ still belongs to the space of ciphertexts, and it can be decrypted if $|(m_1 + m_2) + 2(r_1 + r_2)| < p/2$. Since both r_1 and r_2 are relatively small compared to p, several additions on ciphertexts can be performed before the error exceeds the threshold of $p/2$.

With respect to multiplication, we can write

$$E(m_1)E(m_2) = (m_1 + 2r_1 + pq_1)(m_2 + 2r_2 + pq_2) = m_1m_2 + 2(m_1r_2 + m_2r_1 + 2r_1r_2) + kp,$$

which will be decryptable if $|m_1m_2 + 2(m_1r_2 + m_2r_1 + 2r_1r_2)| < p/2$. Performing a multiplication on ciphertexts, we observe that the error grows exponentially, greatly limiting the number of multiplications that can be performed.

As mentioned, the scheme introduced by van Dijk et al. is extremely simple, but as observed, it is also inefficient. Over the years, several attempts have been made to improve the efficiency of this type of schemes; however, a level has not yet been reached that makes these algorithms applicable in practice.

Acknowledgements: This work was partially funded by the SERICS project (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union-NextGenerationEU. The authors also thank the Italian National Group for Algebraic and Geometric Structures and their Applications (GNSAGA-INdAM), which supported the research.

Funding information: Authors state no funding involved.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and consented to its submission to the journal, reviewed all the results and approved the final version of the manuscript.

Conflict of interest: The authors state no conflict of interest.

References

[1] Acar A, Aksu H, Uluagac AS, Conti M. A survey on homomorphic encryption schemes: Theory and implementation. ACM Comput Surv. 2018 Jul;51(4):Art. 79. doi: 10.1145/3214303.

[2] Fellows M, Koblitz N. Combinatorial cryptosystems galore! In: Finite fields: theory, applications, and algorithms (Las Vegas, NV, 1993). vol. 168 of Contemporary Mathematics. American Mathematical Society, Providence, RI, 1994. p. 51–61. doi: 10.1090/conm/168/01688.

[3] Levy-dit Vehel F, Perret L. A Polly Cracker system based on satisfiability. In: Coding, cryptography and combinatorics. vol. 23 of Progr. Comput. Sci. Appl. Logic. Birkhäuser, Basel; 2004. p. 177–92.

[4] Ly LV. Polly two: a new algebraic polynomial-based public-key scheme. Appl Algebra Engrg Comm Comput. 2006;17(3–4):267–83. doi: 10.1007/s00200-006-0010-0.

[5] Sander T, Young A, Yung M. Non-interactive cryptocomputing for NC^1. In: 40th Annual Symposium on Foundations of Computer Science (New York, 1999). IEEE Computer Soc., Los Alamitos, CA; 1999. p. 554–66. doi: 10.1109/SFFCS.1999.814630.

[6] Ishai Y, Paskin A. Evaluating branching programs on encrypted data. In: Theory of cryptography. vol. 4392 of Lecture Notes in Comput. Sci., Berlin: Springer; 2007. p. 575–94. doi: 10.1007/978-3-540-70936-7_31.

[7] Aguilar Melchor C, Gaborit P, Herranz J. Additively homomorphic encryption with d-operand multiplications. In: Advances in cryptology-CRYPTO 2010. vol. 6223 of Lecture Notes in Comput. Sci. Berlin: Springer; 2010. p. 138–54. doi: 10.1007/978-3-642-14623-7_8.

[8] Boneh D, Goh EJ, Nissim K. Evaluating 2-DNF formulas on ciphertexts. In: Theory of cryptography. vol. 3378 of Lecture Notes in Comput. Sci. Berlin: Springer; 2005. p. 325–41. doi: 10.1007/978-3-540-30576-7_18.

[9] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science; 2011. p. 97–106.

[10] Lyubashevsky V, Peikert C, Regev O. On ideal lattices and learning with errors over rings. J ACM. 2013;60(6):Art. 43, 35. doi: 10.1145/2535925.

[11] Fan J, Vercauteren F. Somewhat practical fully homomorphic encryption. 2012. https://eprint.iacr.org/2012/144.

[12] Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans Comput Theory. 2014;6(3):Art. 13, 36. doi: 10.1145/2633600.

[13] López-Alt A, Tromer E, Vaikuntanathan V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: STOC'12-Proceedings of the 2012 ACM Symposium on Theory of Computing. New York: ACM; 2012. p. 1219–34. doi: 10.1145/2213977.2214086.

[14] Cheon JH, Kim A, Kim M, Song Y. Homomorphic encryption for arithmetic of approximate numbers; 2016. https://eprint.iacr.org/2016/421. Cryptology ePrint Archive, Paper 2016/421. https://eprint.iacr.org/2016/421.

[15] Li B, Micciancio D. On the security of homomorphic encryption on approximate numbers; 2020. https://eprint.iacr.org/2020/1533. Cryptology ePrint Archive, Paper 2020/1533.

[16] Hoffstein J, Pipher J, Silverman JH. NTRU: A ring-based public key cryptosystem. In: Buhler JP, editor. Algorithmic number theory. Berlin, Heidelberg: Springer Berlin Heidelberg; 1998. p. 267–88.

[17] Stehl D, Steinfeld R. Making NTRUEncrypt and NTRUSign as secure as standard worst-case problems over ideal lattices; 2013. https://eprint.iacr.org/2013/004. Cryptology ePrint Archive, Paper 2013/004.

[18] Albrecht M, Bai S, Ducas L. A subfield lattice attack on overstretched NTRU assumptions: Cryptanalysis of some FHE and graded encoding schemes; 2016. https://eprint.iacr.org/2016/127. Cryptology ePrint Archive, Paper 2016/127.

[19] Kluczniak K. NTRU-ν-um: Secure fully homomorphic encryption from NTRU with small modulus; 2022. https://eprint.iacr.org/2022/089. Cryptology ePrint Archive, Paper 2022/089.

[20] van Dijk M, Gentry C, Halevi S, Vaikuntanathan V. Fully homomorphic encryption over the integers; 2009. https://eprint.iacr.org/2009/616. Cryptology ePrint Archive, Paper 2009/616.