An efficient somewhat homomorphic encryption scheme based on factorization

Gérald Gavin

ERIC - Université de Lyon,

5 av. Mendés France, Bron 69676, France

Email: gavin@univ-lyon1.fr

Abstract. Surprisingly, most of existing provably secure FHE or SWHE schemes are lattice-based constructions. It is legitimate to question whether there is a mysterious link between homomorphic encryptions and lattices. This paper can be seen as a first (partial) negative answer to this question. We propose a very simple private-key (partially) homomorphic encryption scheme whose security relies on factorization. This encryption scheme deals with a secret multivariate rational function ϕ_D defined over $\mathbb{Z}_n$, n being an RSA-modulus. An encryption of x is simply a vector c such that $\phi_D(c) = x+\mathsf{noise}$. To get homomorphic properties, nonlinear operators are specifically developed. We first prove IND-CPA security in the generic ring model assuming the hardness of factoring. We then extend this model in order to integrate lattice-based cryptanalysis and we reduce the security of our scheme (in this extended model) to an algebraic condition. This condition is extensively discussed for several choices of parameters. Some of these choices lead to competitive performance with respect to other existing homomorphic encryptions. While quantum computers are not only dreams anymore, designing factorization-based cryptographic schemes might appear as irrelevant. But, it is important to notice that, in our scheme, the factorization of n is not required to decrypt. The factoring assumption simply ensures that solving nonlinear equations or finding non-null polynomials with many roots is difficult. Consequently, the ideas behind our construction could be re-used in rings satisfying these properties.

1 Introduction

The prospect of outsourcing an increasing amount of data storage and management to cloud services raises many new privacy concerns for individuals and businesses alike. The privacy concerns can be satisfactorily addressed if users encrypt the data they send to the cloud. If the encryption scheme is homomorphic, the cloud can still perform meaningful computations on the data, even though it is encrypted.

The theoretical problem of constructing a fully homomorphic encryption scheme (FHE) supporting arbitrary functions f, was only recently solved by the breakthrough work of Gentry [5]. More recently, further fully homomorphic schemes were presented [15],[16],[4],[6] following Gentry's framework. The underlying tool behind all these schemes is the use of Euclidean lattices, which have previously proved powerful for devising many cryptographic primitives. A central aspect of Gentry's fully homomorphic scheme (and the subsequent schemes) is the ciphertext refreshing Recrypt operation. Even if many improvements have been made, this operation remains very costly [11], [7].

In many real-world applications, in the medical, financial, and the advertising domains, which require only that the encryption scheme is *somewhat* homomorphic. Somewhat homomorphic encryption schemes (SWHE), which support a limited number of homomorphic operations, can be much faster, and more compact than fully homomorphic encryption schemes. Even if several quite efficient lattice-based SWHE exist in the literature, significant efficiency improvements should be done for most real-world applications. This paper aims at elaborating an efficient SWHE whose security relies on factorization.

Many cryptographic constructions are based on the famous problem LWE [13]. In particular, this cryptographic problem is currently the most relevant to build FHE [8], [2]. Typically, the secret key is a vector $\boldsymbol{s} \in \mathbb{Z}_n^\kappa$ and an encryption $\boldsymbol{c}$ of a value $x \ll n$ is a randomly chosen vector satisfying[1] $\boldsymbol{s} \cdot \boldsymbol{c} = x + \mathsf{noise}$. This scheme is born (partially) additively homomorphic making it vulnerable against lattice-based attacks. We propose a slight modification to remove this homomorphic property. In our scheme, the secret key becomes a pair of vectors $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ and $\boldsymbol{c}$ is a randomly chosen vector satisfying $\boldsymbol{s}_1 \cdot \boldsymbol{c} / \boldsymbol{s}_2 \cdot \boldsymbol{c} = x + \mathsf{noise} \pmod n$. Clearly, the vector sum is not a homomorphic operator anymore. This is a *sine qua non* condition for overcoming lattice-based attacks. Indeed, as a ciphertext $\boldsymbol{c}$ is a vector, it is always possible to write it as a linear combination of other known ciphertexts. Thus, if the vector sum were a homomorphic operator, the cryptosystem would not be secure at all. This simple remark suffices to prove the weakness of the homomorphic cryptosystems presented in [17], [10]. In order to use the vector sum as a homomorphic operator, noise should be injected into the encryptions as done in all existing FHE [5],[3],[15],[16],[4],[6] and lattice-based attacks can be mount to recover linear combinations with small coefficients. To resist against such attacks, the dimension of $\boldsymbol{c}$ should be chosen sufficiently large which dramatically degrades performance.

To obtain homomorphic properties, nonlinear homomorphic operators Add and Mult should be developed and published. Quadratic homomorphic operators can be naturally defined. However, it should be ensured that these operators do not leak information about the secret key. We get results in this sense under the factoring assumption where n is a product of large secret primes. In particular, we prove the IND-CPA security of our scheme in the generic ring model [9], [1] for any $\kappa \geq 2$. In this model, the CPA attacker is assumed to only perform arithmetic operations $+, -, \times, /$. A security proof in the generic model indicates that the idea of basing the security on factorization is not totally flawed. This leads us more or less to the situation of RSA where it was recently shown that breaking the security of RSA in the generic ring model is as difficult as factoring [1]. A classical objection against security analysis in the generic ring model deals with the Jacobi symbol J_n. For concreteness, it was shown in [9] that computing J_n is difficult in the generic ring model while it is not in general. However, this result is neither surprising nor relevant because J_n is not a rational function[2]. Indeed, we can even show that $\phi(x) = J_n(x)$ with probability smaller than $1/2$ provided ϕ is a rational function and x uniform over $\mathbb{Z}_n^*$. In our scheme, the function ϕ defined by $\phi(\boldsymbol{c}) = x + \mathsf{noise}$ is rational suggesting that a security analysis in the generic ring model is meaningful.

However, the security analysis in the generic ring model is not sufficient because lattice-based cryptanalysis exploiting the fact that $x + \mathsf{noise}$ is small is not considered in this model. In Section 5, we propose a general characterization of lattice-based attacks which naturally extends the generic ring model. We reduce the non-existence of such attacks to an algebraic condition. This condition is discussed in Section 5.3 for several choices of κ. We prove that this condition is satisfied for $\kappa = \Theta(\lambda)$ proving the non-existence of lattice-based attacks. Moreover, the simplest and most natural lattice-based attack is shown inefficient provided $\kappa = \Omega(\log \lambda)$. By assuming that this attack is also the most efficient, choosing $\kappa = \Theta(\log \lambda)$ could hopefully ensure the non-existence of efficient lattice-based attacks.

Notation. *We use standard Landau notations. Throughout this paper, we let λ denote the security parameter: all known attacks against the cryptographic scheme under scope should require $2^{\Omega(\lambda)}$ bit operations to mount.*

- *$\delta \geq 2$ is a positive integer independent of λ.*

[1] $\boldsymbol{s} \cdot \boldsymbol{c}$ denoting the scalar product between $\boldsymbol{s}$ and $\boldsymbol{c}$.

[2] It comes from the fact that $J_n(x) \mod p$ (resp. $J_n(x) \mod q$) is not a function of $x \mod p$ (resp. $x \mod q$)

- *The inner product of two vectors $\boldsymbol{v}$ and $\boldsymbol{v}'$ is denoted by $\boldsymbol{v} \cdot \boldsymbol{v}'$*
- *The set of all square $2\kappa - by - 2\kappa$ matrices over $\mathbb{Z}_n$ is denoted by $\mathbb{Z}_n^{2\kappa\times 2\kappa}$. The i^{th} row of $S \in \mathbb{Z}_n^{2\kappa\times 2\kappa}$ is denoted by s_i and $\mathcal{L}_i$ denotes the linear function defined by $\mathcal{L}_i(\boldsymbol{v}) = s_i \cdot \boldsymbol{v}$.*
- *A δ-RSA modulus n is a product of δ η-bit primes $p_1 \cdots p_\delta$ where η is chosen sufficiently large to ensure that the factorization of n requires $\Omega(2^\lambda)$ bit operations provided $p_1, \ldots, p_\delta$ are randomly chosen.*
- *The set of the positive integer strictly smaller than ξ is denotes by $I_\xi = \{0, \ldots, \xi - 1\}$.*

Remark 1. The number $M(m, d)$ of m-variate monomials of degree d is equal to $\begin{pmatrix} d+m-1 \\ d \end{pmatrix}$. In particular, $M(2\kappa, \kappa) > 3^\kappa$ for any $\kappa \geq 2$.

2 Preliminary definitions and results

Let $\delta \geq 2$ be a positive integer (independent of the security parameter) and let $n = p_1 \cdots p_\delta$ be a randomly chosen δ-RSA modulus. Given a r-variate function ϕ and a subset $I \subseteq \mathbb{Z}_n^r$, $z_{\phi,I}$ denotes the probability over I that $\phi(x) = 0$,

$$z_{\phi,I} \overset{\mathsf{def}}{=} \frac{|\{x \in I | \phi(x) = 0\}|}{|I|}$$

$z_{\phi,\mathbb{Z}_n^r}$ will be simply denoted by z_ϕ.

2.1 Roots of polynomials

The following result proved in [1] establishes that it is difficult to output a polynomial ϕ such that z_ϕ is non-negligible without knowing the factorization of n. The security of RSA in the generic ring model can be quite straightforwardly derived from this result (see [1]).

Theorem 1. ***(Lemma 4 of [1]).*** *Assuming factoring is hard, there is no p.p.t-algorithm $\mathcal{A}$ which inputs n and which outputs*[3] *a $\{+, -, \times\}$-circuit computing a non-null polynomial $\phi \in \mathbb{Z}_n[X]$ such that z_ϕ is non-negligible.*

Thanks to this lemma, showing that two polynomials[4] are equal with non-negligible probability becomes an algebraic problem: it suffices to prove that they are identically equal. This lemma is a very powerful tool which is the heart of the security proofs proposed in this paper. We extend this result to the multivariate case.

Proposition 1. *Assuming factoring is hard, there is no p.p.t algorithm $\mathcal{A}$ which inputs n and which outputs*[3] *a $\{+, -, \times\}$-circuit computing a non-null polynomial $\phi \in \mathbb{Z}_n[X_1, \ldots, X_r]$ such that z_ϕ is non-negligible.*

Proof. See Appendix B.
□

The following proposition yields links between $z_{\phi,I}$ and z_ϕ for particular subsets $I \subseteq \mathbb{Z}_n^r$.

[3] with non-negligible probability (the coin toss being the choice of n and the internal randomness of $\mathcal{A}$)

[4] built without knowing the factorization of n

Proposition 2. *Let $\phi \in \mathbb{Z}_n[X_1, \ldots, X_r]$ and let $I = I_{\xi_1} \times \cdots \times I_{\xi_r}$ with $\xi_j \geq \max(p_1, \ldots, p_\delta)$ for any $j = 1, \ldots, r$. If z_ϕ is negligible then $z_{\phi,I}$ is negligible.*

Proof. See Appendix A
□

By considering the notation of the two previous propositions, if $\phi \leftarrow \mathcal{A}(n)$ and $z_{\phi,I}$ is non-negligible then ϕ is null[5] assuming factoring is hard. This is the heart of our security proofs.

2.2 κ-symmetry

The following definition naturally extends the classical definition of symmetric polynomials.

Definition 1. *A polynomial $\phi \in \mathbb{Z}_n[X_{11}, \ldots, X_{1t}, \ldots, X_{\kappa 1} \ldots, X_{\kappa t}]$ is said to be κ-symmetric if for any permutation σ of $\{1, \ldots, \kappa\}$,*

$$\phi(X_{11}, \ldots, X_{1t}, \ldots, X_{\kappa 1} \ldots, X_{\kappa t}) = \phi(X_{\sigma(1)1}, \ldots, X_{\sigma(1)t}, \ldots, X_{\sigma(\kappa)1}, \ldots, X_{\sigma(\kappa)t})$$

This property is at the heart of our construction. Informally, all the values known by the CPA attacker are evaluations of κ-symmetry polynomials while the decryption function[6] does not satisfy this property. Our security proofs are all based on this fact.

2.3 Rational functions

Throughout this paper, we will consider the class of rational functions useful in our security proof in the generic ring model.

Definition 2. *A function $\phi : \mathbb{Z}_n^r \to \mathbb{Z}_n$ is said to be rational if there exists a $\{+, -, \times, /\}$-circuit computing this function.*

Throughout this paper, recovering a rational function means recovering a $\{+, -, \times, /\}$-circuit computing this function. The following result states that a rational function can be represented by a $\{+, -, \times, /\}$-circuit or equivalently by two $\{+, -, \times\}$-circuits.

Lemma 1. *Given $\mathcal{C}$ be a polynomial-size $\{+, -, \times, /\}$-circuit, we denote by $\phi_\mathcal{C}$ the (rational) function computing by $\mathcal{C}$. There exists a p.p.t. algorithm $\mathcal{A}$ such that $\mathcal{A}(\mathcal{C})$ outputs two polynomial-size $\{+, -, \times\}$-circuits $\mathcal{C}', \mathcal{C}''$ satisfying $\phi_\mathcal{C} = \phi_{\mathcal{C}'}/\phi_{\mathcal{C}''}$.*

Proof. By induction on the gates of $\mathcal{C}$ (see [1]).
□

3 A somewhat homomorphic encryption (SWHE)

3.1 A private-key encryption

Let $\delta > 2$ be a constant and let λ be a security parameter and let η denote the bit size of the prime factors of δ-RSA moduli.

Definition 3. *The functions KeyGen, Encrypt, Decrypt are defined as follows:*

[5] with overwhelming probability

[6] which is not a polynomial but a rational function.

- *KeyGen*(λ, δ). *Let κ be a parameter indexed by λ. Let ξ be an arbitrary $(\eta + 1)$-bit integer, let n be a δ-RSA modulus chosen at random and let S be an invertible matrix of $\mathbb{Z}_n^{2\kappa \times 2\kappa}$ chosen at random. The i^{th} row of S is denoted by s_i and $\mathcal{L}_i$ denotes the linear function defined by $\mathcal{L}_i(\boldsymbol{v}) = s_i \cdot \boldsymbol{v}$. Output*
$$K = \{S\} \ ; \ pp = \{n, \xi\}$$
- *Encrypt*$(K, pp, x \in I_\xi)$. *Choose at random $r_1, r_2, r_2', \ldots, r_\kappa, r_\kappa' \in \mathbb{Z}_n^*$, $k \in I_\xi$ and output*
$$\boldsymbol{c} = S^{-1} \begin{pmatrix} r_1\overline{x} \\ r_1 \\ r_2 \\ r_2' \\ \cdots \\ r_\kappa \\ r_\kappa' \end{pmatrix}$$
where $\overline{x} = x + k\xi$.
- *Decrypt*$(K, pp, \boldsymbol{c} \in \mathbb{Z}_n^{2\kappa})$. *Output $x = \mathcal{L}_1(\boldsymbol{c})/\mathcal{L}_2(\boldsymbol{c}) \mod n \mod \xi$*

In the rest of the paper, it will be assumed that $pp = \{n, \xi\}$ is public. Correctness can be straightforwardly shown by noticing that $\mathcal{L}_1(\boldsymbol{c}) = r_1\overline{x}$ and $\mathcal{L}_2(\boldsymbol{c}) = r_1$. As claimed in the introduction, $\boldsymbol{c}$ is a randomly chosen vector satisfying $\mathcal{L}_1(\boldsymbol{c})/\mathcal{L}_2(\boldsymbol{c}) = \overline{x}$. However, we have adopted a slightly more complex definition in order to introduce material useful when defining the homomorphic operators.

3.2 The multiplicative operator

Let $S \leftarrow$ KeyGen(λ, δ).

Proposition 3. *There exists a (unique) tuple of quadratic 4κ-variate polynomials $(q_1, \ldots, q_{2\kappa})$ such that for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}_n^{2\kappa}$ the vector $\boldsymbol{w} = (q_1(\boldsymbol{u}, \boldsymbol{v}), \ldots, q_{2\kappa}(\boldsymbol{u}, \boldsymbol{v}))$ satisfies*
$$S\boldsymbol{w} = (a_1b_1, \ldots, a_{2\kappa}b_{2\kappa})$$
where $a = S\boldsymbol{u}$, $b = S\boldsymbol{v}$.

Proof. (Sketch.) It suffices to define the polynomials q_i by
$$\begin{pmatrix} q_1(\boldsymbol{u}, \boldsymbol{v}) \\ \vdots \\ q_{2\kappa}(\boldsymbol{u}, \boldsymbol{v}) \end{pmatrix} = S^{-1} \begin{pmatrix} \mathcal{L}_1(\boldsymbol{u})\mathcal{L}_1(\boldsymbol{v}) \\ \vdots \\ \mathcal{L}_{2\kappa}(\boldsymbol{u})\mathcal{L}_{2\kappa}(\boldsymbol{v}) \end{pmatrix}$$
□

We consider the function MultGen which inputs S and outputs the **expanded representation** of the polynomials $q_1, \ldots, q_{2\kappa}$ defined in Proposition 3. By using the fact that each quadratic polynomial q_i has $O(\kappa^2)$ monomial, it is not hard to show that the running time of MultGen is $O(\kappa^4)$. The operator Mult $\leftarrow$ MultGen(S) consists of evaluating these polynomials, i.e. Mult$(\boldsymbol{u}, \boldsymbol{v}) = (q_1(\boldsymbol{u}, \boldsymbol{v}), \ldots, q_{2\kappa}(\boldsymbol{u}, \boldsymbol{v}))$, leading to a running time in $O(\kappa^3)$.

Proposition 4. *Mult $\leftarrow$ MultGen(S) is a valid multiplicative homomorphic operator.*

Proof. Straightforward (see Fig. 1).
□

$$\boxed{\mathsf{Mult}\left(S^{-1}\begin{pmatrix} r_1\overline{x} \\ r_1 \\ \vdots \\ r_\kappa \\ r'_\kappa \end{pmatrix}, S^{-1}\begin{pmatrix} t_1\overline{y} \\ t_1 \\ \vdots \\ t_\kappa \\ t'_\kappa \end{pmatrix}\right) = S^{-1}\begin{pmatrix} r_1t_1\overline{xy} \\ r_1t_1 \\ \vdots \\ r_\kappa t_\kappa \\ r'_\kappa t'_\kappa \end{pmatrix}}$$

Fig. 1. Description of the operator $\mathsf{Mult} \leftarrow \mathsf{MultGen}(S)$.

3.3 The additive operator

Let $S \leftarrow \mathsf{KeyGen}(\lambda, \delta)$.

Proposition 5. *There exists a (unique) tuple of quadratic 4κ-variate polynomials $(q_1, \ldots, q_{2\kappa})$ such that for any $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}_n^{2\kappa}$ the vector $\boldsymbol{w} = (q_1(\boldsymbol{u}, \boldsymbol{v}), \ldots, q_{2\kappa}(\boldsymbol{u}, \boldsymbol{v}))$ satisfies*

$$S\boldsymbol{w} = (a_1b_2 + a_2b_1, a_2b_2, \ldots, a_{2\kappa-1}b_{2\kappa} + a_{2\kappa}b_{2\kappa-1}, a_{2\kappa}b_{2\kappa})$$

where $a = S\boldsymbol{u}$, $b = S\boldsymbol{v}$.

Proof. (Sketch.) It suffices to define the polynomials q_i by

$$\begin{pmatrix} q_1(\boldsymbol{u}, \boldsymbol{v}) \\ \vdots \\ q_{2\kappa}(\boldsymbol{u}, \boldsymbol{v}) \end{pmatrix} = S^{-1}\begin{pmatrix} \mathcal{L}_1(\boldsymbol{u})\mathcal{L}_2(\boldsymbol{v}) + \mathcal{L}_2(\boldsymbol{u})\mathcal{L}_1(\boldsymbol{v}) \\ \mathcal{L}_2(\boldsymbol{u})\mathcal{L}_2(\boldsymbol{v}) \\ \vdots \\ \mathcal{L}_{2\kappa-1}(\boldsymbol{u})\mathcal{L}_{2\kappa}(\boldsymbol{v}) + \mathcal{L}_{2\kappa}(\boldsymbol{u})\mathcal{L}_{2\kappa-1}(\boldsymbol{v}) \\ \mathcal{L}_{2\kappa}(\boldsymbol{u})\mathcal{L}_{2\kappa}(\boldsymbol{v}) \end{pmatrix}$$

□

We consider the function AddGen which inputs S and outputs the **expanded representation** of the polynomials $q_1, \ldots, q_{2\kappa}$ defined in Proposition 5. By using the fact that each quadratic polynomial q_i has $O(\kappa^2)$ monomial, it is not hard to show that the running time of AddGen is $O(\kappa^4)$. The operator $\mathsf{Add} \leftarrow \mathsf{AddGen}(S)$ consists of evaluating these polynomials, i.e. $\mathsf{Add}(\boldsymbol{u}, \boldsymbol{v}) = (q_1(\boldsymbol{u}, \boldsymbol{v}), \ldots, q_{2\kappa}(\boldsymbol{u}, \boldsymbol{v}))$, leading to a running time in $O(\kappa^3)$.

Proposition 6. *$\mathsf{Add} \leftarrow \mathsf{AddGen}(S)$ is a valid additive homomorphic operator.*

Proof. Straightforward (see Fig. 2).

□

$$\boxed{\mathsf{Add}\left(S^{-1}\begin{pmatrix} r_1\overline{x} \\ r_1 \\ \vdots \\ r_\kappa \\ r'_\kappa \end{pmatrix}, S^{-1}\begin{pmatrix} t_1\overline{y} \\ t_1 \\ \vdots \\ t_\kappa \\ t'_\kappa \end{pmatrix}\right) = S^{-1}\begin{pmatrix} r_1t_1(\overline{x}+\overline{y}) \\ r_1t_1 \\ \vdots \\ r_\kappa t'_\kappa + r'_\kappa t_\kappa \\ r'_\kappa t'_\kappa \end{pmatrix}}$$

Fig. 2. Description of the operator $\mathsf{Add} \leftarrow \mathsf{AddGen}(S)$.

3.4 Discussion

Clearly the operators Add and Mult are valid homomorphic operators provided $\delta \geq 4$. Note that these operators are commutative. By publishing these homomorphic operators, we get a somewhat homomorphic private-key encryption scheme. Arithmetic circuits of depth $\delta/2 \approx \log n/2 \log \xi$ can be evaluated. For instance, if n is a 10-RSA Modulus, circuits of depth 5 can be evaluated.

The classic way (see [14]) to transform a private-key cryptosystem into a public-key cryptosystem consists of publicizing encryptions $\boldsymbol{c}_i$ of known values x_i and using the homomorphic operators to encrypt x. Let $\mathsf{Encrypt1}$ denote this new encryption function. Assuming the IND-CPA security of the private-key cryptosystem, it suffices that $\mathsf{Encrypt1}(pk, x)$ and $\mathsf{Encrypt}(K, pp, x)$ are computationally indistinguishable to ensure the IND-CPA security of the public-key cryptosystem.

4 Security Analysis

All the security results of this section are true for any $\kappa \geq 2$. Thus, to simplify notation, we set $\kappa = 2$ throughout this section. Let $K = \{S\} \leftarrow \mathsf{KeyGen}(\lambda, \delta)$. To break semantic security, an attacker is required to find a function φ satisfying

$$\mathsf{Adv}^{\varphi} \overset{\mathsf{def}}{=} |\mathsf{Pr}(\varphi \circ \mathsf{Encrypt}(K, pp, 1) = 1) - \mathsf{Pr}(\varphi \circ \mathsf{Encrypt}(K, pp, 0) = 1)| \text{ is non-negligible.}$$

Externalizing the generation of $\boldsymbol{n}$. To clearly understand the role of the factoring assumption in our security proof, it is important to notice that the factorization of n is not used by KeyGen. Consequently, the generation of n could be externalized[7] (for instance generated by an oracle). In other words, n could be a public input of KeyGen. This means that all the polynomials considered in this section are built without using the factorization of n implying that they are equal to 0 with negligible probability provided they are not null (according to Proposition 1).

Randomness $\boldsymbol{\theta_n}$. After n is publicized, the CPA attacker receives the public operators Add and Mult and it has access to an encryption oracle. It chooses $(x_i)_{i=1,\ldots,t} \in I_\xi$ and receives encryptions $(\boldsymbol{c}_i = S^{-1}(r_i\overline{x}_i, r_i, r'_i, r''_i))_{i=1,\ldots,t}$ of $(x_i)_{i=1,\ldots,t}$ from the encryption oracle. The randomness of its knowledge comes from the internal randomness of KeyGen and the one of the encryption oracle. This randomness is encapsulated in the tuple θ_n of elements belonging to $\mathbb{Z}_n$ defined by

$$\theta_n = (s_1, s_2, r_1\overline{x}_1, r_1 \ldots, r_t\overline{x}_t, r_t, s_3, s_4, r'_1, r''_1 \ldots, r'_t, r''_t)$$

Knowledge of the CPA attacker. We first assume that $\varDelta = \det S$ is revealed to the CPA attacker. By doing this, it can be assumed that $(\varDelta \cdot \boldsymbol{c}_i)_{i=1,\ldots,t}$, $\varDelta \cdot \mathsf{Add}$ and $\varDelta \cdot \mathsf{Mult}$ are revealed to the CPA attacker instead of $(\boldsymbol{c}_i)_{i=1,\ldots,t}$, Add and Mult. It follows that the CPA attacker receives a tuple $\alpha_n \in \mathbb{Z}_n^m$ where each component is the evaluation over θ_n of a polynomial[8] α_i , i.e. $\alpha_n = (\alpha_1(\theta_n), \ldots, \alpha_m(\theta_n))$. All our security analysis is based on the fact that the polynomials α_i are κ-symmetric[9] (see Definition 1). Throughout this section, we consider the function $\widehat{\alpha}$ defined by $\widehat{\alpha}(\theta_n, \boldsymbol{z}) = (\alpha_1(\theta_n), \ldots, \alpha_m(\theta_n), \boldsymbol{z})$ for any $\boldsymbol{z} \in \mathbb{Z}_n^4$. By construction, we have

$$(\alpha_n, \boldsymbol{c}) = \widehat{\alpha}(\theta_n, \boldsymbol{c})$$

[7] ensuring that its factorization was forgotten just after its generation

[8] α_i can be seen as a $\{+, -, \times\}$-circuit C (independent of n) with $|\theta_n|$ inputs.

[9] it means that $\alpha_i(s_1, s_2, r_1\overline{x}_1, r_1 \ldots, r_t\overline{x}_t, r_t, s_3, s_4, r'_1, r''_1 \ldots, r'_t, r''_t) = \alpha_i(s_3, s_4, r'_1, r''_1 \ldots, r'_t, r''_t, s_1, s_2, r_1\overline{x}_1, r_1 \ldots, r_t\overline{x}_t, r_t)$. It should be noticed that $\det S$ is a κ-symmetric polynomial defined over θ_n.

4.1 Generic Ring Model

A Generic Ring Algorithm (GRA) (see [1]) defined over a ring $\mathcal{R}$ (here $\mathcal{R} = \mathbb{Z}_n$) is an algorithm where only arithmetic operations $+, -, \times, /$ and equality tests are allowed. In the special case $\mathcal{R} = \mathbb{Z}_n$, equality tests are not needed. This is implicitly shown in [1] as a straightforward consequence of Proposition 1. Indeed, this proposition ensures that two rational functions[10] are either equal with negligible probability or equal with overwhelming probability. It follows that a GRA is simply a $\{+, -, \times, /\}$-circuit computing a rational function φ. We say that our scheme is IND-CPA secure in the classical generic model if there does not exist any p.p.t algorithm $\mathcal{A}$ such that $\mathcal{A}(n)$ outputs a $\{+, -, \times, /\}$-circuit of a rational function φ satisfying

$$|\mathsf{Pr}(\varphi \circ \widehat{\alpha}(\theta_n, \mathsf{Encrypt}(K, pp, 1)) = 1) - \mathsf{Pr}(\varphi \circ \widehat{\alpha}(\theta_n, \mathsf{Encrypt}(K, pp, 0)) = 1)| \tag{1}$$

is non-negligible.

Lemma 2. *SWHE is IND-CPA secure in the classical generic ring model assuming the hardness of factoring.*

Proof. (Sketch.) Assume $\varphi \circ \widehat{\alpha}(\theta, \mathsf{Encrypt}(K, pp, 1)) = 1$ with non-negligible probability. According to Proposition 2, $\varphi \circ \widehat{\alpha}(\boldsymbol{z}, \boldsymbol{z}_2) - 1 = 0$ with non-negligible probability provided $\boldsymbol{z}, \boldsymbol{z}_2$ uniform over $\mathbb{Z}_n^{|\theta|} \times \mathbb{Z}_n^4$. Thus, according to Proposition 1, $\varphi \circ \widehat{\alpha} - 1$ is null implying that φ does not satisfy (1). □

However, this result is not surprising because the decryption function is not rational[11]. We propose to extend this model by enhancing the power of the attacker: informally, we let it use the function $\bmod \xi$. By doing this, the CPA attacker only needs to recover the evaluation $p(\overline{x})$ of a polynomial p in order to recover x or at least to break IND-CPA security in this model. Indeed, if the degree of p and its coefficients are small enough[12] then $p(\overline{x}) \bmod n \bmod \xi = p(x) \bmod \xi$. This extension is encapsulated in the next definition.

Definition 4. (Generic IND-CPA security). *Our scheme is generically IND-CPA secure if there does not exist any p.p.t algorithm $\mathcal{A}$ such that $\mathcal{A}(n)$ outputs[13] a $\{+, -, \times, /\}$-circuit computing a rational function φ, $x \in I_\xi$ and a non-constant polynomial p satisfying*

$$\varphi \circ \widehat{\alpha}(\theta_n, \boldsymbol{c})[= \varphi(\alpha_n, \boldsymbol{c})] = p(\overline{x}) \tag{2}$$

with non-negligible probability over $\theta_n, \boldsymbol{c} \leftarrow \mathsf{Encrypt}(K, pp, x)$.

4.2 Hardness of factoring ⇒ generic IND-CPA security

In this section, we prove the generic IND-CPA security of our scheme. The proof exploits intrinsic symmetry properties of our construction. Informally, only functions (indexed by S) which are stable by permuting the two first rows of S with the two last ones can be generically recovered. In particular, the decryption function $\mathcal{L}_1/\mathcal{L}_2$ cannot be generically recovered.

Theorem 2. *SWHE is generically IND-CPA secure assuming the hardness of factoring.*

[10] built in polynomial-time under the factoring assumption.

[11] as explained for J_n in the introduction, there does not exist a rational function equal to the decryption function with non-negligible probability.

[12] Ideally $p(\overline{x}) = \overline{x}$.

[13] with non-negligible probability, the coin toss being the internal randomness of $\mathcal{A}$ and the choice of n

Proof. For the sake of simplicity, we assume that the CPA attacker does not use the encryption oracle implying that $\theta_n = (s_1, s_2, s_3, s_4) \overset{s}{\equiv} \mathbb{Z}_n^{16}$. The extension to the general case is quite straightforward considering Proposition 2. Moreover, without loss of generality, we only consider the case $\kappa = 2$ and $p(\overline{x}) = \overline{x}$.

Throughout this proof, we consider the polynomials I_j defined by $I_j(X_1, \ldots, X_{20}) = X_j$. Let $W = [w_{ij}] = S^{-1}$. The degree-3 polynomial computing $\det S \cdot w_{ij}$ is (abusively) denoted by w_{ij}, i.e. $w_{ij}(S) = \det S \cdot w_{ij}$.

Let $\varepsilon_1, \ldots, \varepsilon_4$, $\mu_1, \ldots, \mu_4$ and $\nu_1, \ldots, \nu_4$ be defined by

$$\begin{aligned}
&\varepsilon_\ell(Y_1 \ldots, Y_{16}, X_1, \ldots, X_4) = \sum_{j=1}^{4} w_{\ell j}(Y_1 \ldots, Y_{16}) X_j \\
&\nu_1(Y_1 \ldots, Y_{16}, R_1, R_2, R_3, R_4) = R_1 R_2 \\
&\nu_{2 \leq \ell \leq 4}(Y_1 \ldots, Y_{16}, R_1, R_2, R_3, R_4) = R_\ell \\
&\mu_1(Y_1 \ldots, Y_{16}, K, R_1, R_2, R_3) = x + K\xi \\
&\mu_{2 \leq \ell \leq 4}(Y_1 \ldots, Y_{16}, K, R_1, R_2, R_3) = R_{\ell - 1}
\end{aligned}$$

We consider the polynomial tuples $\varepsilon = (I_1, \ldots, I_{16}, \varepsilon_1, \ldots, \varepsilon_4)$, $\mu = (I_1, \ldots, I_{16}, \mu_1, \ldots, \mu_4)$ and $\nu = (I_1, \ldots, I_{16}, \nu_1, \ldots, \nu_4)$. By construction, $\varepsilon \circ \nu \circ \mu(\theta_n, k, r, r', r'') = (\theta_n, \boldsymbol{c} = \det S \times S^{-1}(r(x + k\xi), r, r', r''))$. It is important to note that $\boldsymbol{c}$ is an encryption of x drawn according to the underlying probability distribution of $\mathsf{Encrypt}(K, pp, x)$.
Let us assume that $\varphi = \phi'/\phi, p, x \leftarrow \mathcal{A}(n)$ satisfies[14] (2). It follows that

$$\phi^{'} \circ \widehat{\alpha} \circ \varepsilon \circ \nu \circ \mu(\theta_n, k, r, r', r'') = (x + k\xi) \cdot \phi \circ \widehat{\alpha} \circ \varepsilon \circ \nu \circ \mu(\theta_n, k, r, r', r'')$$

with non-negligible probability over the choice of $\theta_n, (k, r, r', r'') \xleftarrow{\$} I_\xi \times \mathbb{Z}_n^{*3}$. Thus, according to Proposition 2,

$$\phi^{'} \circ \widehat{\alpha} \circ \varepsilon \circ \nu \circ \mu(\theta_n, \boldsymbol{z}) = (x + z_1\xi) \cdot \phi \circ \widehat{\alpha} \circ \varepsilon \circ \nu \circ \mu(\theta_n, \boldsymbol{z})$$

with non-negligible probability over the choice of $\theta_n, \boldsymbol{z} = (z_1, \ldots, z_4) \xleftarrow{\$} \mathbb{Z}_n^4$. It follows that

$$\phi^{'} \circ \widehat{\alpha} \circ \varepsilon \circ \nu(\theta_n, \boldsymbol{z}) = z_1 \cdot \phi \circ \widehat{\alpha} \circ \varepsilon \circ \nu(\theta_n, \boldsymbol{z})$$

with non-negligible over the choice of $\theta_n, \boldsymbol{z} \xleftarrow{\$} \mathbb{Z}_n^4$. Consequently, $(I_{17}/I_{18}) \cdot \phi^{'} \circ \widehat{\alpha} \circ \varepsilon(\theta_n, \boldsymbol{z}) = \phi \circ \widehat{\alpha} \circ \varepsilon(\theta_n, \boldsymbol{z})$ implying that

$$I_{17} \cdot \phi^{'} \circ \widehat{\alpha} \circ \varepsilon(\theta_n, \boldsymbol{z}) = I_{18} \cdot \phi \circ \widehat{\alpha} \circ \varepsilon(\theta_n, \boldsymbol{z})$$

with non-negligible probability over the choice of $\theta_n, \boldsymbol{z} \xleftarrow{\$} \mathbb{Z}_n^4$. Thus according to Proposition 1,

$$I_{17} \cdot \phi^{'} \circ \widehat{\alpha} \circ \varepsilon = I_{18} \cdot \phi \circ \widehat{\alpha} \circ \varepsilon$$

contradicting the fact that $\phi^{'} \circ \widehat{\alpha} \circ \varepsilon$ and $\phi \circ \widehat{\alpha} \circ \varepsilon$ are both 2-symmetric, i.e. $\widehat{\alpha} \circ \varepsilon(z_1, \ldots, z_{20}) = \widehat{\alpha} \circ \varepsilon(z_9, \ldots, z_{16}, z_1, \ldots, z_8, z_{19}, z_{20}, z_{17}, z_{18})$.

□

[14] with non-negligible probability over $\theta_n, \boldsymbol{c} \leftarrow \mathsf{Encrypt}(K, pp, x)$

5 Lattice-based cryptanalysis

Throughout this section, we adopt the notation of the previous section. In particular, α_n denotes the knowledge of the CPA attacker and θ_n denotes the internal randomness coming from KeyGen and the encryption oracle used to produce α_n. In the previous section, we prove the generic IND-CPA security of our encryption scheme under the factoring assumption for any $\kappa \geq 2$. This indicates that the idea of basing the security on factorization is not totally flawed. However, this is not sufficient because lattice-based cryptanalysis is excluded from this analysis: indeed lattice-based algorithms *work outside* $\mathbb{Z}_n$ and they compute functions which may be not rational.

Throughout this section, we will consider the polynomial $\Phi_R = \mathcal{L}_2 \cdots \mathcal{L}_{2\kappa}$. This polynomial is indexed by S (and thus θ_n) and it can be seen as a degree-κ homogeneous polynomial ϕ_R defined over $\theta_n, \boldsymbol{c}$, i.e.

$$\phi_R(\theta_n, \boldsymbol{c}) = \Phi_R(\boldsymbol{c}) = \prod_{\ell=1,\ldots,\kappa} s_{2\ell} \cdot \boldsymbol{c} = \prod_{\ell=1,\ldots,\kappa} \left(\sum_{i=1}^{2\kappa} s_{2\ell,i} \cdot c_i \right)$$

5.1 A basic example

Let $x \in I_\xi$, let $\boldsymbol{c} \leftarrow \mathsf{Encrypt}(K, pp, x)$ and let Φ_X be the polynomial defined by $\Phi_X = \Phi_R \cdot \mathcal{L}_1 / \mathcal{L}_2$. By construction,

$$\Phi_X(\boldsymbol{c}) / \Phi_R(\boldsymbol{c}) = \overline{x}$$

Φ_X (also Φ_R) is a homogeneous degree-κ polynomial, i.e.

$$\Phi_X(\boldsymbol{c}) = \sum_{e_1 + \cdots + e_{2\kappa} = \kappa} a_{e_1,\ldots,e_{2\kappa}} c_1^{e_1} \cdots c_{2\kappa}^{e_{2\kappa}}$$

According to Theorem 2, the CPA attacker cannot generically recover both Φ_R and Φ_X. Nevertheless, let us assume that it can generically derive Φ_R from its knowledge α_n. It follows that

$$\sum_{e_1 + \cdots + e_{2\kappa} = \kappa} a_{e_1,\ldots,e_{2\kappa}} \cdot \frac{c_1^{e_1} \cdots c_{2\kappa}^{e_{2\kappa}}}{\Phi_R(\boldsymbol{c})} = \overline{x} \ll n \tag{3}$$

By exploiting the fact that $\overline{x}$ is small relatively to n and by considering sufficiently many encryptions, the monomial coefficients $a_{e_1,\ldots,e_{2\kappa}}$ of Φ_X could be classically recovered by using a lattice basis reduction algorithm, e.g. LLL or BKZ. However, this attack requires first to recover Φ_R. In the next section, we propose a characterization of lattice-based attacks and we show that recovering Φ_R or a multiple of Φ_R is a necessary condition to mount a lattice-based attack. This condition will be discussed in section 5.3.

5.2 Characterization of lattice-based attacks

In this section, we propose a general characterization of lattice-based attacks which naturally extends the generic ring model.

Given $x \in I_\xi$, let us imagine that the CPA attacker is able to recover functions $\varphi_1, \ldots, \varphi_t$ such that there are coefficients $a_1, \ldots, a_t \in \mathbb{Z}_n$ and a function ε satisfying

$$a_1 \cdot \varphi_1(\boldsymbol{c}) + \cdots + a_t \cdot \varphi_t(\boldsymbol{c}) = \varepsilon(\boldsymbol{c})$$

where $\boldsymbol{c} \leftarrow \mathsf{Encrypt}(K, pp, x)$ and $\varepsilon(\boldsymbol{c}) \ll n$. By sampling sufficiently many encryptions $\boldsymbol{c}$, the coefficients $a_1, \ldots, a_t$ and thus ε can be recovered by solving an approximate-SVP. This is a relevant attack if the knowledge of ε can be used to break IND-CPA security. This attack can be identified to the tuple $(\varphi_1, \ldots, \varphi_t)$. This is formally encapsulated in the following definition where the quantities $\varphi_1(\boldsymbol{c}), \ldots, \varphi_t(\boldsymbol{c})$ are generically derived and where $\varepsilon(\boldsymbol{c}) = p(\overline{x})$, p being a polynomial.

Definition 5. (Lattice-based attacks). *A lattice-based attack is an efficient algorithm $\mathcal{A}$ such that $\mathcal{A}(n)$ outputs*[15] *a tuple of rational functions $(\varphi_1, \ldots, \varphi_t)$, $x \in I_\xi$ and a non-constant polynomial p such that there exist*[16] *functions $a_1, \ldots, a_t$ satisfying*

$$a_1(\theta_n) \cdot \varphi_1 \circ \widehat{\alpha}(\theta_n, \boldsymbol{c}) + \ldots + a_t(\theta_n) \cdot \varphi_t \circ \widehat{\alpha}(\theta_n, \boldsymbol{c}) = p(\overline{x}) \tag{4}$$

with non-negligible probability the choice of $\theta_n, \boldsymbol{c} \leftarrow \mathsf{Encrypt}(K, pp, x)$.

If there exists a lattice-based attack $\mathcal{A}$ then the CPA attacker can obviously use it to recover rational functions $\varphi_1, \ldots, \varphi_t$ satisfying (4) then it can hope to recover $a_1(\theta_n), \ldots, a_t(\theta_n)$ and thus to break IND-CPA security by using lattice basis reduction algorithms exploiting the fact that $\overline{x} \ll n$.

Theorem 3. *Let $\mathcal{A}$ be a lattice-based attack and let assume that $(\phi'_1/\phi_1, \ldots, \phi'_t/\phi_t), x, p \leftarrow \mathcal{A}(n)$ satisfies (4). Assuming the hardness of factoring, there exists*[17] *$i \in \{1, \ldots, t\}$ such that $\gcd(\phi_i \circ \widehat{\alpha}, \phi_R) = \phi_R$.*

Proof. Let $I_j \in \mathbb{Z}_n[X_1, \ldots, X_4]$ be defined by $I_j(X_1, \ldots, X_4) = X_j$. Without loss of generality, we only consider the case $\kappa = 2$. For the sake of simplicity, we assume that the CPA attacker does not use the encryption oracle implying that $\theta_n = (s_1, s_2, s_3, s_4) \overset{s}{\equiv} \mathbb{Z}_n^{16}$. The extension to the general case is quite straightforward considering Proposition 2.

Let $\phi = \phi_1 \cdots \phi_t$. We first prove that $\gcd(\phi \circ \widehat{\alpha}, \phi_R) = \phi_R$. Let $\boldsymbol{c} = S^{-1}(r\overline{x}, r, r', r'') \leftarrow \mathsf{Encrypt}(K, pp, x)$ and let $y = (\overline{x}, r, r', r'')$. Let θ_n^* be such that (4) is satisfied with non-negligible probability over the choice of $\boldsymbol{c}$. Let ϕ_i^*, $\phi_i^{'*}$, ε^* and ν^* be the polynomial functions defined by $\phi_i^*(\boldsymbol{c}) = \phi_i \circ \widehat{\alpha}(\theta_n^*, \boldsymbol{c})$, $\phi_i^{'*}(\boldsymbol{c}) = \phi'_i \circ \widehat{\alpha}(\theta_n^*, \boldsymbol{c})$, $\nu^*(y) = (r\overline{x}, r, r', r'')$ and $\varepsilon^* \circ \nu^*(y) = \boldsymbol{c}$ and let ψ^* be the polynomial defined by

$$\psi^* = a_1(\theta_n^*) \cdot \phi_i^{'*} \prod_{i=1,\ldots,t; i \neq 1} \phi_i^* + \ldots + a_t(\theta_n^*) \cdot \phi_t^{'*} \prod_{i=1,\ldots,t; i \neq t} \phi_i^*$$

Equation (4) implies that

$$p(\overline{x}) \cdot \phi^*(\boldsymbol{c}) - \psi^*(\boldsymbol{c}) = 0$$

with non-negligible probability over the choice of $\boldsymbol{c} \leftarrow \mathsf{Encrypt}(K, pp, x)$. It follows that $p(\overline{x}) \cdot \phi^* \circ \varepsilon^* \circ \varepsilon^{*-1}(\boldsymbol{c}) - \psi^* \circ \varepsilon^* \circ \varepsilon^{*-1}(\boldsymbol{c}) = 0$ with non-negligible probability. Consequently, according to Proposition 2,

$$p(z_1) \cdot \phi^* \circ \varepsilon^* \circ \nu^*(\boldsymbol{z}) - \psi^* \circ \varepsilon^* \circ \nu^*(\boldsymbol{z}) = 0$$

with non-negligible probability provided $\boldsymbol{z} \overset{\$}{\leftarrow} \mathbb{Z}_n^4$. Thus, $p(I_1/I_2) \cdot \phi^* \circ \varepsilon^*(\boldsymbol{z}) = \psi^* \circ \varepsilon^*(\boldsymbol{z})$ with non-negligible probability. It implies that there exists a polynomial p' defined by $p'(x, y) = y^{\deg p} p(x/y)$. It follows that the equality $p'(I_1, I_2) \cdot \phi^* \circ \varepsilon^*(\boldsymbol{z}) = I_2^{\deg p} \cdot \psi^* \circ \varepsilon^*(\boldsymbol{z})$ holds with

[15] with non-negligible, the toss coin being the internal randomness of $\mathcal{A}$ and the choice of n

[16] Theorem 2 ensures that $a_1(\theta_n), \ldots, a_t(\theta_n)$ cannot be generically derived from α_n.

[17] with overwhelming probability

non-negligible probability. According to Proposition 1, $p'(I_1, I_2) \cdot \phi^* \circ \varepsilon^* = I_2^{\deg p} \cdot \psi^* \circ \varepsilon^*$ implying that

$$\phi^* \circ \varepsilon^*(z_1, 0, z_3, z_4) = 0 \tag{5}$$

Throughout this proof, we consider the polynomials I_j defined by $I_j(X_1, \ldots, X_{20}) = X_j$. We consider the degree-4 polynomial D computing the determinant of S, i.e. satisfying $D(S) = \det S$, and the polynomial $\varDelta$ defined by

$$\varDelta(Y_1, \ldots, Y_{16}, C_1, \ldots, C_4) = D(Y_1, \ldots, Y_{16})$$

By construction $\varDelta(\theta_n, \boldsymbol{c}) = \det S$. Let $W = [w_{ij}] = S^{-1}$. The degree-3 polynomial computing $\det S \cdot w_{ij}$ is (abusively) denoted by w_{ij}, i.e. $w_{ij}(S) = \det S \cdot w_{ij}$.

Let $\delta_1, \ldots, \delta_4, \varepsilon_1, \ldots, \varepsilon_4$ be defined by

$$\delta_\ell(Y_1, \ldots, Y_{16}, C_1, \ldots, C_4) = \sum_{j=1}^{4} Y_{4(\ell-1)+j} C_j$$

$$\varepsilon_\ell(Y_1, \ldots, Y_{16}, C_1, \ldots, C_4) = \sum_{j=1}^{4} w_{\ell j}(Y_1, \ldots, Y_{16}) C_j$$

We consider the polynomial tuples $\delta = (I_1, \ldots, I_{16}, \delta_1, \ldots, \delta_4)$ and $\varepsilon = (I_1, \ldots, I_{16}, \varepsilon_1, \ldots, \varepsilon_4)$. By construction,

$$\varepsilon \circ \delta \overset{\mathsf{def}}{=} (\varepsilon_1(\delta), \ldots, \varepsilon_{20}(\delta)) = (I_1, \ldots, I_{16}, \varDelta \cdot I_{17}, \ldots \varDelta \cdot I_{20})$$

According to (5),

$$\phi \circ \widehat{\alpha} \circ \varepsilon(\theta_n, z_1, 0, z_3, z_4) = 0$$

with non-negligible probability over the choice of θ_n, z_1, z_3, z_4. Thus, according to Proposition 1, $\phi \circ \widehat{\alpha} \circ \varepsilon$ can be factored by I_{18}. As $\widehat{\alpha} \circ \varepsilon$ is 2-symmetric, $\phi \circ \widehat{\alpha} \circ \varepsilon$ is 2-symmetric. It follows that $\phi \circ \widehat{\alpha} \circ \varepsilon$ can be factored by $I_{18} I_{20}$ implying that $\phi \circ \widehat{\alpha} \circ \varepsilon \circ \delta$ and thus $\psi = \phi \circ \widehat{\alpha}$ can be factored by ϕ_R (noticing that $(I_{18} I_{20}) \circ \delta = \phi_R$).

To conclude, it suffices to notice that for any $i \in \{1, \ldots, t\}$, we have either $\gcd(\phi_i \circ \widehat{\alpha}, \phi_R) = 1$ or $\gcd(\phi_i \circ \widehat{\alpha}, \phi_R) = \phi_R$ (it comes from the fact that $\phi_i \circ \widehat{\alpha} \circ \varepsilon$ is 2-symmetric).

□

Corollary 1. *There does not exist*[18] *any polynomial-size $\{+, -, \times\}$-circuit computing a polynomial ϕ satisfying* $\gcd(\phi \circ \widehat{\alpha}, \phi_R) = \phi_R$ $\Rightarrow$ *There does not exist any lattice-based attack assuming the hardness of factoring,*

This corollary provides a sufficient algebraic condition ensuring the existence of lattice-based attacks. This condition is discussed in the next section.

5.3 Analysis

We discuss Theorem 3 and Corollary 1 for several choices of κ.

- $\kappa = \varTheta(\lambda)$. As mentioned at the beginning of this section, each value known by the CPA attacker is an evaluation over θ_n of a κ-symmetric polynomial. We enhance the power of the

[18] with overwhelming probability over the choice of n.

CPA attacker by allowing it to recover evaluations $\alpha_i(\theta_n)$ of arbitrarily chosen κ-symmetric polynomials $(\alpha_i)_{i=1,\ldots,t}$. Each monomial coefficient of $\varPhi_R$ is a κ-symmetric polynomial defined over S. However, its expanded representation is exponential-size provided $\kappa = \varTheta(\lambda)$. The question arising here consists of wondering whether $\varPhi_R$ can be efficiently and generically written using only $(\alpha_i(\theta_n))_{i=1,\ldots,t}$. We provide a negative answer to this question.

Proposition 7. *Assuming the hardness of factoring, there does not exist any lattice-based attack provided $\kappa = \varTheta(\lambda)$.*

Proof. See Appendix D for details.

According to Corollary 1, it suffices to prove that there does not exist any efficient polynomial-size $\{+, -, \times\}$-circuit computing a polynomial ϕ satisfying $\gcd(\phi \circ \widehat{\alpha}, \phi_R) = \phi_R$. The proof essentially comes from the fundamental theorem of symmetric polynomials.

□

This result is fundamental in the sense that it formally proves the non-existence of lattice-based attacks for some choices of κ.

- $\kappa \geq t \log_3 \lambda$. In this case, we do not have any formal result excluding the possibility to generically recover $\varPhi_R$. However, the attack described in Section 5.1 is not efficient. Indeed, $\varPhi_X$ has a number of monomials larger than λ^t (see Remark 1) implying that the dimension of the lattice considered in this attack is also larger than λ^t, e.g. $\varPhi_X$ has more than 2×10^7 monomials for $\kappa \geq 10$. As the approximation obtained in polynomial-time with the best known lattice basis reduction algorithm is exponential, it suffices to adjust t in order that this approximation is not good enough. This choice of κ would be relevant by assuming that this attack is the most efficient. We are convinced that this assumption is true legitimating this choice of κ.

6 Efficiency

Our scheme can evaluate arithmetic circuit of depth smaller than $\delta/2$. A ciphertext is a 2κ-vector in $\mathbb{Z}_n$, implying that the ratio of ciphertext size to plaintext size is approximatively equal to $4\kappa\delta$. By assuming that the size of a δ-RSA modulus is $O(\delta)$, the running time of Encrypt/ Decrypt/ Add/ Mult is $O(\delta^2\kappa)$, $O(\delta^2\kappa)$, $O(\delta^2\kappa^3)$, $O(\delta^2\kappa^3)$. The security analysis proposed in the previous section is not sufficient to determine κ. The performance of our scheme is very competitive with respect to classic schemes with $\kappa = \varTheta(\log \lambda)$ but poor with $\kappa = \varTheta(\lambda)$. For instance, if we choose $\kappa = 10$ (a choice potentially relevant according to the previous section), applying the homomorphic operators requires around 2000 modular multiplications. Our security analysis should be refined to optimize the choice of κ.

7 Future Work

Our security proof is not complete and the main challenge is to completely reduce the security of our scheme to the factorization.

Another interesting question consists of wondering whether this SWHE can be boostrapped in order to obtain an FHE scheme. We did not think about this and we do not have any idea about the way to achieve it.

The randomization of the homomorphic operator presented in Appendix C gives hope for another motivating perspective. The factoring assumption defeats the whole "post-quantum" purpose of multivariate cryptography [12]. In our opinion, this assumption could be removed by introducing randomness into homomorphic operators in order to maintain the truth of the formal results proved under the factoring assumption.

It is important to notice that the factorization of n is not used by the decryption function of our scheme. The factoring assumption simply ensures that solving nonlinear equations or finding non-null polynomials with many roots is difficult. Consequently, the ideas behind our construction can be straightforwardly re-used in rings satisfying these properties.

References

1. Divesh Aggarwal and Ueli M. Maurer. Breaking RSA generically is equivalent to factoring. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 36–53, 2009.
2. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 97–106, Washington, DC, USA, 2011. IEEE Computer Society.
3. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. Cryptology ePrint Archive, Report 2011/344, 2011. http://eprint.iacr.org/.
4. Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 446–464, 2012.
5. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
6. Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, pages 465–482, 2012.
7. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the aes circuit. In *CRYPTO*, pages 850–867, 2012.
8. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92, 2013.
9. Tibor Jager and Jörg Schwenk. On the analysis of cryptographic assumptions in the generic ring model. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 399–416, 2009.
10. Aviad Kipnis and Eliphaz Hibshoosh. Efficient methods for practical fully homomorphic symmetric-key encrypton, randomization and verification. Cryptology ePrint Archive, Report 2012/637, 2012. http://eprint.iacr.org/.
11. Kristin Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? *IACR Cryptology ePrint Archive*, 2011:405, 2011.
12. Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *EUROCRYPT*, pages 33–48, 1996.
13. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93, 2005.
14. Ron Rothblum. *Homomorphic Encryption: From Private-Key to Public-Key*, pages 219–234. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
15. Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. In *ASIACRYPT*, pages 377–394, 2010.
16. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010.
17. Liangliang Xiao, Osbert Bastani, and I-Ling Yen. An efficient homomorphic encryption protocol for multi-user systems. *IACR Cryptology ePrint Archive*, 2012:193, 2012.

A Proof of Proposition 2

Lemma 3. *Let p be an arbitrary positive integer, let $F \subset I_p^r$, let $\xi_1, \dots, \xi_r$ be r arbitrary integers larger than p, let $I = I_{\xi_1} \times \dots \times I_{\xi_r}$ and let $G = \{x \in I | x \mod p \in F\}$. If $|F|/p^r$ is negligible then $|G|/|I|$ is negligible.*

Proof. (Sketch.) Given $0 < \rho < 1$, let us consider the set Δ_ρ of subsets $F \subset I_p^r$ such that $|F|/p^r \leq \rho$. Given $a = (a_1, \dots, a_r)$ and $b = (b_1, \dots, b_r)$ such that $0 \leq a_i \leq b_i < p$, $|\{x \in I | x \mod p = a\}| \geq |\{x \in I | x \mod p = b\}|$. It follows that it suffices to consider the subsets $F \in \Delta_\rho$ defined by $F = I_{a_1} \times \dots \times I_{a_r}$ for some integers $a_1, \dots, a_r$ belonging to I_p. For such subsets, $|F|/p^r = b_1 \cdots b_r \leq \rho$ where $b_i = a_i/p$.
Moreover, one can easily show that

$$|G|/|I| \leq \prod_{i=1}^{r} |\{x \in I_{\xi_i} | x \mod p_i \leq a_i\}|/|I_{\xi_i}| \leq \prod_{i=1}^{r} \frac{2b_i}{1+b_i}$$

Under the constraint $b_1 \cdots b_r \leq \rho$, one can show by induction that $\prod_{i=1}^{r} \frac{2b_i}{1+b_i} \leq \left(\frac{2\rho^{1/r}}{1+\rho^{1/r}}\right)^r$. A simple function analysis proves that $\left(\frac{2\rho^{1/r}}{1+\rho^{1/r}}\right)^r$ is an increasing function asymptotically equal to $\rho^{1/2}$ proving that $|G|/|I| < \rho^{1/2}$. It follows that $|G|/|I|$ is negligible provided $|F|/p^r$ is negligible.
□

As ϕ is a polynomial, there exists δ polynomials $(\phi_i)_{i=1,\dots,\delta} : \mathbb{Z}_{p_i}^r \to \mathbb{Z}_{p_i}$ such that $\phi(x) \mod p_i = \phi_i(x \mod p_i)$. Let $G = \{x \in I | \phi(x) = 0\}$, $F_i = \{x \in \mathbb{Z}_{p_i}^r | \phi_i(x) = 0\}$ and $G_i = \{x \in I | x \mod p_i \in F_i\}$. By definition $z_{\phi,I}$ non-negligible means that $|G|/|I|$ is non-negligible. As $G = G_1 \cap \dots \cap G_\delta$, $|G_i|/|I|$ is non-negligible. This directly implies that $\alpha_i = |F_i|/p_i^r$ is non-negligible according to Lemma 3. By definition, $\phi(x) = 0$ if and only if $(x \mod p_i \in F_i)_{i=1,\dots,\delta}$. As the events $\{x \in \mathbb{Z}_n^r | x \mod p_i \in F_i\}_{i=1,\dots,\delta}$ are independent, we have $z_\phi = \alpha_1 \cdots \alpha_\delta$. Thus, this quantity is not negligible (δ is a constant which is independent of λ).
□

B Proof of Proposition 1

This result can be shown by induction over r. By Lemma 1, the result is true for $r = 1$. Let us assume the result true for any $r < t$ and let us show it for $r = t$. We can identify $\mathbb{Z}_n[X_1, \dots, X_t]$ to $R[X_t]$ with $R = \mathbb{Z}_n[X_1, \dots, X_{t-1}]$. Let ϕ be a non-null polynomial $\phi \in \mathbb{Z}_n[X_1, \dots, X_t]$ output by a p.p.t. algorithm $\mathcal{A}$, i.e. $\phi \leftarrow \mathcal{A}(n)$. ϕ can be identified by a non-null polynomial $\phi' \in R[X_1]$. Thus, by fixing $X_2, \dots, X_t$ to randomly chosen values $x_2, \dots, x_t \in \mathbb{Z}_n$, the polynomial $\phi_{x_2,\dots,x_t}$ defined by $\phi_{x_2,\dots,x_t}(x_1) = \phi(x_1, \dots, x_t)$ is not (identically) null with overwhelming probability over the choice of $n, x_2, \dots, x_t$ according to the induction hypothesis. Moreover, provided $\phi_{x_2,\dots,x_t}$ is not null, $\phi_{x_2,\dots,x_t}(x_1) = 0$ with negligible probability other choice of n, x_1 according to the induction hypothesis. This proves $\phi(x_1, \dots, x_t) = 0$ with negligible over the choice of $n, x_1, \dots, x_t$.
□

C Randomizing the homomorphic operators

Let τ be an even integer. The key idea of this section is to add τ rows to S which are not useful for encryptions. For concreteness, S is a randomly chosen matrix of $\mathbb{Z}_n^{(4+\tau)\times(4+\tau)}$ and an encryption $\boldsymbol{c}$ of x is

$$\boldsymbol{c} = S^{-1}\left(r\overline{x}, r, r', r'', 0, \dots, 0\right)$$

Moreover, the construction of the homomorphic operators can be naturally extended. Let E be the set[19] of the linear combinations over the vectors $s_5, \ldots, s_{4+\tau}$. By construction, for any $\boldsymbol{u} \in E$, $\boldsymbol{u}\cdot\boldsymbol{c} = 0$. Let R be the set of quadratic polynomials r defined by $r(\boldsymbol{c}, \boldsymbol{c}') = \boldsymbol{u}\cdot\boldsymbol{c}\times\boldsymbol{v}'\cdot\boldsymbol{c}' + \boldsymbol{v}\cdot\boldsymbol{c}\times\boldsymbol{u}'\cdot\boldsymbol{c}'$ where $\boldsymbol{u}, \boldsymbol{u}' \in E$ and $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{Z}_n^{4+\tau}$ are arbitrary vectors. By construction, for any $r \in R$ and any public encryptions $\boldsymbol{c}, \boldsymbol{c}'$,

$$r(\boldsymbol{c}, \boldsymbol{c}') = 0$$

Each homomorphic operator is a tuple $(q_1, \ldots, q_{4+\tau})$ of $4 + \tau$ polynomials. Let $(r_1, \ldots, r_{4+\tau})$ be randomly chosen in R. By construction, for any encryptions $\boldsymbol{c}, \boldsymbol{c}'$ it is ensured that

$$(q_i + r_i)(\boldsymbol{c}, \boldsymbol{c}') = q_i(\boldsymbol{c}, \boldsymbol{c}')$$

Thus, the operator $(q_1 + r_1, \ldots, q_{4+\tau} + r_{4+\tau})$ can be seen as a randomized operator $(q_1, \ldots, q_{4+\tau})$.

D Proof of Proposition 7

To simplify notation, we only show that $\phi \circ \widehat{\alpha} \neq \phi_R$. Without loss of generality, we assume that $\theta_n = (s_1, \ldots, s_{2\kappa})$ and that the polynomials α_i are homogeneous. Moreover, as $\deg \phi_R = \kappa$, one can assume that $\deg \alpha_i \leq \kappa$. Consider the two sets I_1, I_2 defined by

- $I_1 = \{i \in \{1, \ldots, t\} | \deg \alpha_i < \kappa\}$,
- $I_2 = \{i \in \{1, \ldots, t\} | \deg \alpha_i = \kappa\}$

Lemma 4. *There do not exist any polynomial $q \in \mathbb{Z}_n[X_1, \ldots, X_t]$ and (κ-)symmetric polynomials $\alpha_1, \ldots, \alpha_t \in \mathbb{Z}_n[X_1, \ldots, X_\kappa]$ satisfying $\deg \alpha_i < \kappa$ and $q(\alpha_1, \ldots, \alpha_t) = X_1 \cdots X_\kappa$.*

Proof. Let $\alpha_1, \ldots, \alpha_t \in \mathbb{Z}_n[X_1, \ldots, X_\kappa]$ be arbitrary symmetric polynomials s.t. $\deg \alpha_i < \kappa$. Let us consider the κ symmetric polynomials $\sigma_k = \sum_{1 \leq i_1 < \ldots < i_k < \kappa} X_{i_1} \cdots X_{i_k}$ and an arbitrary symmetric polynomial $\phi \in \mathbb{Z}_n[X_1, \ldots, X_\kappa]$. The fundamental theorem of symmetric polynomials says that there exists a unique polynomial φ satisfying $\phi = \varphi(\sigma_1, \ldots, \sigma_\kappa)$. Thus, as $\deg \alpha_i < \kappa$, $\alpha_1, \ldots, \alpha_t$ can be written as polynomials φ_i defined over $\sigma_1, \ldots, \sigma_{\kappa-1}$ but σ_κ cannot. Thus, there is no polynomial $q \in \mathbb{Z}_n[X_1, \ldots, X_t]$ s.t. $q(\alpha_1, \ldots, \alpha_t) = \sigma_\kappa = X_1 \cdots X_\kappa$.
□

Let $V_\kappa \overset{\mathsf{def}}{=} \{0,1\}^\kappa \times \{0\}^\kappa$. For a given $\boldsymbol{v} \in \mathbb{Z}_n^{2\kappa}$, the polynomial $\phi_{R|\boldsymbol{v}}$ is defined by,

$$\phi_{R|\boldsymbol{v}=(v_1,\ldots,v_{2\kappa})}(S) = \prod_{\ell=1,\ldots,\kappa} \left(\sum_{i=1}^{2\kappa} v_i s_{2\ell,i} \right)$$

Lemma 5. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r \in V_\kappa$ and $a_1, \ldots, a_r \in \mathbb{Z}_n \setminus \{0\}$. The polynomial $a_1\phi_{R|\boldsymbol{v}_1} + \ldots + a_r\phi_{R|\boldsymbol{v}_r}$ cannot be written as a polynomial $p((\alpha_i)_{i \in I_1})$.*

Proof. (Sketch.) By Lemma 4, one can straightforwardly show that $\phi_{R|(1,0,\ldots,0)}$ ($\phi_{R|(1,0,\ldots,0)}(S) = s_{2,1}s_{4,1}\cdots s_{2\kappa,1}$) cannot be written as a polynomial $p((\alpha_i)_{i \in I_1})$. We denote by $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_t$ the polynomials $\alpha_1, \ldots, \alpha_t$ where the variables $s_{2\ell,i}$ are substituted by $\tau_i s_{2\ell,1}$ ($\tau_i \in \mathbb{Z}_n$) for any $1 \leq i \leq \kappa$ and φ_i denotes the polynomial $\phi_{R|\boldsymbol{v}_i}$ by doing the same substitution. It is important to notice that $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_t$ are κ-symmetric defined over $s_{2,1}, s_{4,1}, \cdots, s_{2\kappa,1}$. We show that $\sum_{i=1}^{r} a_i \varphi_i(s_{2,1}, s_{4,1}, \cdots, s_{2\kappa,1}) = q(\tau_1, \ldots, \tau_\kappa) s_{2,1}s_{4,1}\cdots s_{2\kappa,1}$ where q is a degree-κ non-null polynomial. Thus, according to the famous lemma of Schwartz and Lippel, $q(\tau_1, \ldots, \tau_\kappa) = 0$ with negligible probability over the choice of $\tau_1, \ldots, \tau_\kappa$. Thus, if $p((\alpha_i)_{i \in I_1}) = a_1\phi_{R|\boldsymbol{v}_1} + \ldots + a_r\phi_{R|\boldsymbol{v}_r}$ then $p((\widehat{\alpha}_i)_{i \in I_1}) \sim s_{2,1} \cdots s_{2\kappa,1}$ which was previously shown impossible.
□

[19] E can be recovered by the attacker.

Corollary 2. *The family of polynomials* $\left(\phi_{R|\boldsymbol{v}}\right)_{\boldsymbol{v}\in V_\kappa}$ *is linearly independent.*

The result is a direct consequence of this lemma. Let ϕ_v be the polynomial defined by $\phi_{\boldsymbol{v}}(\alpha_1,\ldots,\alpha_t) = \phi(\alpha_1,\ldots,\alpha_t,\boldsymbol{v})$. We can write $\phi_{\boldsymbol{v}}(\alpha_1,\ldots,\alpha_t) = \phi'_{\boldsymbol{v}}(\alpha_{i\in I_1}) + \phi''_{\boldsymbol{v}}(\alpha_{i\in I_2})$ where $\deg\phi''_{\boldsymbol{v}} = 1$.

Let us assume that $\phi\circ\widehat{\alpha} = \phi_R$ implying that for each $\boldsymbol{v}\in V_\kappa$, $\phi_{R|v} = \phi_{\boldsymbol{v}}(\alpha_1,\ldots,\alpha_t)$. As t is polynomial but not $|V_\kappa|$, there exist $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_r\in V_\kappa$ and $a_1,\ldots,a_r\in\mathbb{Z}_n\setminus\{0\}$ such that $a_1\phi''_{\boldsymbol{v}_1}(\alpha_{i\in I_2}) + \ldots + a_r\phi''_{\boldsymbol{v}_r}(\alpha_{i\in I_2}) = 0$. It follows that $a_1\phi'_{\boldsymbol{v}_1}(\alpha_{i\in I_1}) + \cdots + a_r\phi'_{\boldsymbol{v}_r}(\alpha_{i\in I_1}) = a_1\phi_{R|\boldsymbol{v}_1} + \ldots + a_r\phi_{R|\boldsymbol{v}_r}$ contradicting Lemma 5.

□