A general framework for building noise-free homomorphic cryptosystems

Gérald Gavin

University of Lyon, France

Email: gavin@univ-lyon1.fr

Abstract. We present a general framework for developing and analyzing homomorphic cryptosystems whose security relies on the difficulty of solving systems of nonlinear equations over $\mathbb{Z}_n$, n being an RSA modulus. In this framework, many homomorphic cryptosystems can be conceptualized. Based on symmetry considerations, we propose a general assumption that ensures the security of these schemes. To highlight this, we present an additive homomorphic private-key cryptosystem and we prove its security. Finally, we propose two motivating perspectives of this work. We first propose an FHE based on the previous scheme by defining a simple multiplicative operator. Secondly, we propose ways to remove the factoring assumption in order to get pure multivariate schemes.

Keywords. *Homomorphic cryptosystem, FHE, Multivariate encryption scheme, Factoring assumption.*

1 Introduction

In [6] and [7], new ideas and new tools were proposed to develop homomorphic cryptosystems. The authors first proposed a very simple private-key cryptosystem where a ciphertext is a vector $\boldsymbol{c}$ whose components are in $\mathbb{Z}_n$, n being an RSA modulus chosen at random. Given a secret multivariate polynomial ϕ_D, an encryption of $x \in \mathbb{Z}_n$ is a vector $\boldsymbol{c}$ chosen at random such that $\phi_D(\boldsymbol{c}) = x$. In order to resist a CPA attacker, the number of monomials of ϕ_D should not be polynomial (otherwise the cryptosystem can be broken by solving a polynomial-size linear system). In order to get polynomial-time encryptions and decryptions, ϕ_D should be written in a compact form, e.g. a factored or semi-factored form. By construction, the generic cryptosystem described above is not homomorphic in the sense that the vector sum is not a homomorphic operator. This is a *sine qua non* condition for overcoming Gentry's machinery. 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 [14], [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 [8],[3],[12],[13],[4],[9]. To overcome this, the authors propose developing *ad hoc* nonlinear homomorphic operators to get a noise-free compact FHE. However, the proof of security of their scheme is far from being completed, and only partial security results are provided.

In this paper, we adopt the same approach except that ϕ_D is a rational function instead of being a polynomial, i.e. $\phi_D(\boldsymbol{c}) = \phi_1(\boldsymbol{c})/\phi_2(\boldsymbol{c}) = x$. The polynomial $\Phi(\boldsymbol{c}) = \phi_1(\boldsymbol{c}) - x\phi_2(\boldsymbol{c})$ is equal to 0 if $\boldsymbol{c}$ encrypts x implying that its expanded representation could be recovered by solving a linear system. This kind of attacks will be called *attacks by linearization*. However, this attack fails by adjusting the parameters in order that Φ has an exponential number of monomials. By using results based on symmetry (see Section 2.2), we show the difficulty to represent ϕ_1 or ϕ_2 in a compact factored or semi-factored form assuming the hardness of factoring (see Section 5.1).

However, it is not sufficient to ensure security. Indeed, the homomorphic operators consist of applying nonlinear operators $\mathcal{Q}_1, \ldots, \mathcal{Q}_\rho$ (see Section 3). By recursively applying these operators over a challenge encryption $\boldsymbol{c}_1$ and other encryptions $\boldsymbol{c}_2, \ldots, \boldsymbol{c}_r$ in an arbitrary way, a CPA attacker can generate vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t$ in the hope to create new efficient attacks by linearization, i.e. recovering a small polynomial ϕ

such that $\phi(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t) = 0$ with a larger probability when $\boldsymbol{c}_1$ encrypts x_1 rather than 0. In Section 5, we conjecture that our scheme is IND-CPA secure if this does not happen. In Section 4.2, we develop a very simple nonlinear additive operator and we prove that our scheme is IND-CPA secure under this assumption (and another one closely related to the factoring assumption).

There are two major perspectives from this work. The principal one would be to build a compact FHE. In Section 8, we propose a very simple multiplicative operator. We are obviously convinced that the obtained FHE is IND-CPA secure but its security proof is left as an open problem for further research. A second motivating perspective would be to remove the factoring assumption to obtain a pure multivariate encryption scheme. The factoring assumption is required to get formal results (Lemma 4, Lemma 5 and Proposition 3). We propose ways to remove this assumption (see Remark 4 and Remark 5) in the hope of getting pure multivariate schemes. Basically, it consists of adding randomness to the construction in order to maintain the truth of the formal results proved under the factoring assumption.

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. Let $\kappa \in \mathbb{N} \setminus \{0\}$ and let n be a randomly chosen RSA modulus. All the computations considered in this paper will be done in $\mathbb{Z}_n$.*

- $\mathcal{K} = \{0, \ldots, \kappa - 1\}$.
- *A vector* $\boldsymbol{v} = \begin{pmatrix} v_1 \\ \cdots \\ v_{2\kappa} \end{pmatrix}$ *can be also denoted by* $(v_1, \ldots, v_{2\kappa})$.
- *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* $\mathcal{L}_i(\boldsymbol{v}) = s_i \cdot \boldsymbol{v}$.

Remark 1. The number of κ-variate monomials of degree γ is equal to $\binom{\gamma + \kappa - 1}{\gamma}$. In particular, this number is exponential provided $\kappa = \Theta(\lambda)$ and $\gamma = \Omega(\lambda)$. This will be implicitly considered in Conjecture 2.

2 Security assumptions

2.1 Roots of polynomials

Let n be an η-bit RSA modulus and let $r \in \mathbb{N} \setminus \{0\}$. Given a polynomial $\phi \in \mathbb{Z}_n[X_1, \ldots, X_r]$, z_ϕ denotes the probability that $\phi(x) = 0$ assuming x uniform over $\mathbb{Z}_n^r$, i.e. $z_\phi = |S|/n^r$ where S is the set of the roots of ϕ. In this section, we wonder whether it is possible to recover a polynomial ϕ such that z_ϕ is non-negligible. We start by showing a weaker result.

Lemma 1. *Assuming the hardness of factoring, there is no p.p.t-algorithm $\mathcal{A}$ which inputs a randomly chosen RSA modulus n and which outputs an arithmetic circuit of a polynomial $\phi \in \mathbb{Z}_n[X_1, \ldots, X_r]$ such that z_ϕ and $1 - z_\phi$ are both non-negligible.*

Proof. See Appendix B.1

□

The previous result is not sufficient because it does not exclude the possibility to recover a non-null polynomial ϕ such that $z_\phi = 1$ for instance. The following result goes in this sense.

Lemma 2. *Assuming the hardness of factoring, there is no p.p.t-algorithm $\mathcal{A}$ which inputs a randomly chosen RSA modulus n and which outputs the expanded representation of a non-null polynomial $\phi \in \mathbb{Z}_n[X]$ such that $z_\phi = 1$.*

Proof. See Appendix B.2.
□

However, this result does not strictly prove the difficulty of finding a polynomial ϕ such that $z_\phi = 1$. Indeed, it only deals with the expanded representation of such polynomials but it does not say anything about other representations, e.g. factored representations. To establish the main result of this section, we assume that this problem is also difficult.

Conjecture 1. There is no p.p.t-algorithm $\mathcal{A}$ which inputs a randomly chosen RSA modulus n and which outputs an arithmetic circuit of a non-null polynomial $\phi \in \mathbb{Z}_n[X]$ such that $z_\phi = 1$.

Since $z_{X^{\lambda(n)}-X} = 1$ ($\lambda(n)$ refers to the Euler's function), Conjecture 1 is stronger than the factoring assumption.

Lemma 3. *Assuming Conjecture 1, there is no p.p.t-algorithm $\mathcal{A}$ which inputs a randomly chosen RSA modulus n and which outputs an arithmetic circuit of a non-null polynomial $\phi \in \mathbb{Z}_n[X]$ such that z_ϕ is non-negligible.*

Proof. See Appendix B.3.
□

Lemma 4. *Assuming Conjecture 1, there is no p.p.t-algorithm $\mathcal{A}$ which inputs a randomly chosen RSA modulus n and which outputs an arithmetic circuit of 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.4.
□

2.2 κ-symmetry

Let n be an η-bit RSA modulus chosen at random and let $\kappa, t > 1$ be positive integers polynomials in η. Recall that $\mathcal{K} = \{0, \ldots, \kappa - 1\}$. Let y_1, y_2 be randomly chosen in $\mathbb{Z}_n$. It is well-known that recovering[1] y_1 given only $S = y_1 + y_2$ or $P = y_1 y_2$ is difficult assuming the hardness of factoring. In this section, we propose to extend this.

Definition 1. *A κt-variate polynomial s is κ-symmetric if for any $y_0, \ldots, y_{\kappa-1} \in \mathbb{Z}_n^t$ and for any $\sigma \in \mathcal{K}$, $s(y_0, \ldots, y_{\kappa-1}) = s(y'_0, \ldots, y'_{\kappa-1})$ where $y'_\ell = y_{\ell+\sigma \mod \kappa}$.*

Let $\mathcal{A}_S$ be an arbitrary efficient algorithm which inputs n and outputs m κ-symmetric κt-variate polynomials $s_1, \ldots, s_m$ and a non κ-symmetric κt-variate polynomial π. By construction, the polynomials $\pi, s_1 \ldots, s_m$ are built without knowing the factorization of n. We assume that $s_1, \ldots, s_m$ and π are public in the sense

[1] y_1, y_2 are the roots of the polynomial $y^2 - Sy + P$.

that they can be publicly and efficiently evaluated given any $y_0, \ldots, y_{\kappa-1}$. In other words, an efficient representation of these polynomials is published. The following problem consists in evaluating $\pi(y_0, \ldots, y_{\kappa-1})$ given only $s_1(y_0, \ldots, y_{\kappa-1}), \ldots, s_m(y_0, \ldots, y_{\kappa-1})$ where the tuples y_ℓ are chosen at random under some symmetric additive constraints.

PROBLEM 1. *Let $I_F \subseteq \{1, \ldots, t\}$, let n be a randomly chosen RSA modulus and let $(s_1, \ldots, s_m, \pi) \leftarrow \mathcal{A}_S(n)$ be public κt-variate polynomials satisfying,*

- *$s_1, \ldots, s_m$ are κ-symmetric*
- *π is a monomial defined[2] over $\{y_{\ell i} | (\ell, i) \in \mathcal{K} \times I_F\}$ such that $\deg \pi < \kappa$.*

Let $(y_0, \ldots, y_{\kappa-1})$ i.d.d. drawn according to the uniform distribution over $\mathbb{Z}_n^t$ s.t. for each $i \notin I_F$, $y_{0i} + \ldots + y_{\kappa-1,i} = x_i$ where $x_i \in \mathbb{Z}_n$ are arbitrarily chosen by the attacker.

The challenge is to recover $\pi(y_0, \ldots, y_{\kappa-1})$ given only[3] $s_1(y_0, \ldots, y_{\kappa-1}), \ldots, s_m(y_0, \ldots, y_{\kappa-1})$.

Lemma 5. *Problem 1 is difficult if factoring is hard.*

Proof. See Appendix C.1.
□

Corollary 1. *The values $y_{\ell, i \in I_F}$ cannot be recovered.*

Remark 2. By assuming Conjecture 1, Problem 1 can be simply extended by defining π as a non-κ symmetric polynomial instead of a monomial.

By construction, a CPA attacker will only know values which are κ-symmetric with respect to the tuples $y_0, \ldots, y_{\kappa-1}$ defined in the proof of Proposition 3. Thus, by Lemma 5, the CPA attackers *live in the κ-symmetric world*. In the remainder of this section, we will see that the life is difficult in this world. First, consider the bivariate polynomials[4] $s_0(X_1, X_2) = X_1^2$, $s_1(X_1, X_2) = X_2^2$ and $s_2(X_1, X_2) = X_1 X_2$. These polynomials are clearly linearly independent. Given y uniform over $\mathbb{Z}_n^2$, y cannot be recovered given only $s_1(y), s_2(y)$. Nevertheless, the equality $s_2^2 = s_0 s_1$ ensures that it is possible to find $s_0(y)$ given only $s_1(y)$ and $s_2(y)$. Lemma 6 shows that this does not happen in the *κ-symmetric world*.

Let us consider the set E_d of κ-symmetric polynomials belonging to $\mathbb{Z}_n[X_1, \ldots, X_{\kappa t}]$ defined by

$$E_d = \left\{ \sum_{\ell=0}^{\kappa-1} X_{i_1+\ell t} \cdots X_{i_d+\ell t} | \, i_1, \ldots, i_d \in \{1, \ldots, t\} \right\}$$

We denote by F_d the set of linear combinations over E_d. Let $y_0, \ldots, y_{\kappa-1}$ be κ tuples uniform over $\mathbb{Z}_n^t$. In the remainder of this section, we wonder whether it is possible to recover $s_0(y_0, \ldots, y_{\kappa-1})$ given only $s_1(y_0, \ldots, y_{\kappa-1}), \ldots, s_m(y_0, \ldots, y_{\kappa-1})$ where $s_0, s_1, \ldots, s_m \in F_d$ s.t. $s_0 \notin \mathsf{co}(s_1, \ldots, s_m)$. The following lemma shows that s_0 cannot be written as a simple rational function.

Lemma 6. *Let $s_0, s_1, \ldots, s_m$ be linearly independent polynomials belonging to F_d. There does not exist m-variate non-zero polynomials p, q satisfying $\deg p, q \leq \kappa$ and $s_0 \cdot q(s_1, \ldots, s_m) = p(s_1, \ldots, s_m)$.*

Proof. See Appendix C.2
□

[2] $\pi = \prod_{(\ell,i) \in \mathcal{K} \times \{1,\ldots,t\}} y_{\ell i}^{e_{\ell i}}$ where $e_{\ell i} = 0$ when $i \notin I_F$. Moreover $\deg \pi < \kappa \Rightarrow$ non κ-symmetric.

[3] and an efficient representation of $\pi, s_1, \ldots, s_m$.

[4] It deals with the case $\kappa = 1$.

By Lemma 5, the tuples $y_0, \ldots, y_{\kappa-1}$ cannot be recovered implying that s_0 cannot be directly evaluated knowing only the evaluations of $s_1, \ldots, s_m$. Let us assume the existence of two polynomials p, q satisfying $s_0 \cdot q(s_1, \ldots, s_m) = p(s_1, \ldots, s_m)$. According to Lemma 6, $\deg p, q \geq \kappa$ ensuring that these polynomials have an exponential number of monomials (see Remark 1) provided $\kappa = \Theta(\lambda)$ and $m = \Theta(\lambda)$. This enhances the idea that p, q and thus s_0 cannot be polynomially evaluated. This idea is encapsulated in Conjecture 2.

3 The function QGen

Let S be an invertible matrix of $\mathbb{Z}_n^{2\kappa \times 2\kappa}$ and let $\boldsymbol{v}, \boldsymbol{v}'$ be two vectors of $\mathbb{Z}_n^{2\kappa}$. 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 $\mathcal{L}_i(\boldsymbol{v}) = s_i \cdot \boldsymbol{v}$. In this section, we consider quadratic operators $\mathcal{Q}$ where $\mathcal{Q}(\boldsymbol{v}, \boldsymbol{v}')$ outputs a vector $\boldsymbol{v}''$ such that the components of $S\boldsymbol{v}''$ are (known) polynomials of the components of $S\boldsymbol{v}$ and $S\boldsymbol{v}'$.

Definition 2. *Let S be an invertible matrix and let $\sigma, \sigma' \in \mathcal{K}$.*

1. *Let $\mathcal{P} = (p_1, p_2)$ be a family of quadratic polynomials $p_i : \mathbb{Z}_n^2 \times \mathbb{Z}_n^2 \to \mathbb{Z}_n$ s.t.*
$$p_i(x, x') = \sum_{(j,k)\in\{1,2\}^2} a_{ijk} x_j x'_k$$
2. *Let $z_0, \ldots, z_{\kappa-1} : \mathbb{Z}_n^{2\kappa} \times \mathbb{Z}_n^{2\kappa} \to \mathbb{Z}_n^2 \times \mathbb{Z}_n^2$ defined by*
$$z_\ell(\boldsymbol{v}, \boldsymbol{v}') = \big(\mathcal{L}_{2\ell_\sigma+1}(\boldsymbol{v}), \mathcal{L}_{2\ell_\sigma+2}(\boldsymbol{v}), \mathcal{L}_{2\ell_{\sigma'}+1}(\boldsymbol{v}'), \mathcal{L}_{2\ell_{\sigma'}+2}(\boldsymbol{v}')\big)$$
where $\ell_\sigma = \ell + \sigma \mod \kappa$ and $\ell_{\sigma'} = \ell + \sigma' \mod \kappa$.
3. *The function QGen inputs $S, \mathcal{P}, \sigma, \sigma'$ and outputs the expanded representation of the polynomials $q_1, \ldots, q_{2\kappa}$ defined by*
$$(q_1, \ldots, q_{2\kappa}) = S^{-1}\left(p_1 \circ z_0, p_2 \circ z_0, \ldots, p_1 \circ z_{\kappa-1}, p_2 \circ z_{\kappa-1}\right)$$
4. *The operator $\mathcal{Q} \leftarrow$ QGen$(S, \mathcal{P}, \sigma, \sigma')$ consists of evaluating the polynomials q_i, i.e.*
$$\mathcal{Q}(\boldsymbol{v}, \boldsymbol{v}') = \big(q_1(\boldsymbol{v}, \boldsymbol{v}'), \ldots, q_{2\kappa}(\boldsymbol{v}, \boldsymbol{v}')\big)$$

$$\mathcal{Q}\left(S^{-1}\begin{pmatrix}a_1\\a_2\\a_3\\a_4\\a_5\\a_6\end{pmatrix}, S^{-1}\begin{pmatrix}b_1\\b_2\\b_3\\b_4\\b_5\\b_6\end{pmatrix}\right) = S^{-1}\begin{pmatrix}p_1(a_3,a_4,b_5,b_6)\\p_2(a_3,a_4,b_5,b_6)\\p_1(a_5,a_6,b_1,b_2)\\p_2(a_5,a_6,b_1,b_2)\\p_1(a_1,a_2,b_3,b_4)\\p_2(a_1,a_2,b_3,b_4)\end{pmatrix} = S^{-1}\begin{pmatrix}a_3b_5\\a_4b_6\\a_5b_1\\a_6b_2\\a_1b_3\\a_2b_4\end{pmatrix}$$

Fig. 1. $\mathcal{Q} \leftarrow$ QGen$(S, (p_1, p_2), 1, 2)$ with $p_1(x, x') = x_1x'_1$ and $p_2(x, x') = x_2x'_2$ in the case $\kappa = 3$. A toy implementation of this operator (for $\kappa = 1$) is presented in Appendix A.

Proposition 1. *The computation of $\mathcal{Q} \leftarrow$ QGen$(S, \mathcal{P}, \sigma, \sigma')$ requires $O(\kappa^4)$ modular multiplications and the computation of $\boldsymbol{v}'' \leftarrow \mathcal{Q}(\boldsymbol{v}, \boldsymbol{v}')$ requires $O(\kappa^3)$ modular multiplications.*

Proof. *(Sketch.)* A quadratic 2κ-variate polynomial has $O(\kappa^2)$ monomials.
□

4 An additively homomorphic encryption scheme

4.1 A private-key encryption

Definition 3. *Let λ be a security parameter. The functions* KeyGen, Encrypt, Decrypt *are defined as follows:*

- KeyGen(λ). *Let η, κ be positive integers indexed by λ, let n be an η-bit 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*

$$K = \{S\}$$

- Encrypt$(K, x \in \mathbb{Z}_n)$. *Choose at random $r_0, \ldots, r_{\kappa-1} \in \mathbb{Z}_n^*$ and $x_0, \ldots, x_{\kappa-1} \in \mathbb{Z}_n$ s.t. $x_0 + \ldots + x_{\kappa-1} = x$ and output*

$$\boldsymbol{c} = S^{-1} \begin{pmatrix} r_0 x_0 \\ r_0 \\ r_1 x_1 \\ r_1 \\ \cdots \\ r_{\kappa-1} x_{\kappa-1} \\ r_{\kappa-1} \end{pmatrix}$$

- Decrypt$(K, \boldsymbol{c} \in \mathbb{Z}_n^{2\kappa})$. *Output $x = \phi_D(\boldsymbol{c})$ defined by*

$$\phi_D(\boldsymbol{c}) = \sum_{\ell=0}^{\kappa-1} \mathcal{L}_{2\ell+1}(\boldsymbol{c}) / \mathcal{L}_{2\ell+2}(\boldsymbol{c})$$

Thanks to the symmetry properties of this scheme, we show in Section 5.1 that ϕ_D cannot be recovered in a compact form provided $\kappa = \Theta(\lambda)$. At this step, this encryption scheme is not homomorphic. Homomorphic operators will be developed using only operators $\mathcal{Q}$.

Remark 3. The factorization of n is not required to decrypt. One can wonder whether the factoring assumption is necessary.

4.2 An additive homomorphic operator

Let $S \leftarrow \mathsf{KeyGen}(\lambda)$ and $(p_i)_{i=1,2}$ be the family of polynomials: $\mathbb{Z}_n^2 \times \mathbb{Z}_n^2 \to \mathbb{Z}_n$ defined by

- $p_1(x, x') = x_1 x_2' + x_2 x_1'$
- $p_2(x, x') = x_2 x_2'$

$\mathsf{AddGen}(S)$ outputs the operator $\mathsf{Add} \leftarrow \mathsf{QGen}(S, (p_1, p_2), 0, 0)$.

Proposition 2. *Let* Add $\leftarrow$ AddGen(S) *is a valid additive homomorphic operator.*

Proof. Straightforward (see Figure 3).
□

By publishing this homomorphic operator, we get an additive homomorphic private-key encryption scheme. The classic way 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 Encrypt1 denote this new encryption function. Assuming the IND-CPA security of the private-key cryptosystem, it suffices that Encrypt1(pk, x) and Encrypt(K, x) are computationally indistinguishable to ensure the IND-CPA security of the public-key cryptosystem.

$$\mathsf{Add}\left(S^{-1}\begin{pmatrix} r_0x_0 \\ r_0 \\ r_1x_1 \\ r_1 \\ \cdots \\ r_{\kappa-1}x_{\kappa-1} \\ r_{\kappa-1}\end{pmatrix}, S^{-1}\begin{pmatrix} r'_0x'_0 \\ r'_0 \\ r'_1x'_1 \\ r'_1 \\ \cdots \\ r'_{\kappa-1}x'_{\kappa-1} \\ r'_{\kappa-1}\end{pmatrix}\right) = S^{-1}\begin{pmatrix} r_0r'_0(x_0+x'_0) \\ r_0r'_0 \\ r_1r'_1(x_1+x'_1) \\ r_1r'_1 \\ \cdots \\ r_{\kappa-1}r'_{\kappa-1}(x_{\kappa-1}+x'_{\kappa-1}) \\ r_{\kappa-1}r'_{\kappa-1}\end{pmatrix}$$

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

5 A general security assumption

In the previous section, we showed how to build an additively homomorphic operator using only operators $\mathcal{Q}$. In Section 8, a multiplicative homomorphic operator will be proposed. In this section, we assume that some operators $\mathcal{Q}$ are made public and we propose a general security assumption (dealing with these operators) about the IND-CPA security of the private-key encryption scheme.

Let $S \leftarrow \mathsf{KeyGen}(\lambda)$, let $\mathcal{P}_1, \ldots, \mathcal{P}_\rho$ be ρ families of quadratic polynomials indexed by n (satisfying the requirements of Definition 2), let $(\sigma_i, \sigma'_i)_{i=1,\ldots,\rho}$ be elements of $\mathcal{K}$ and let

$$\mathcal{Q}_i \leftarrow \mathsf{QGen}(S, \mathcal{P}_i, \sigma_i, \sigma'_i)$$

be ρ operators. The quantities $n, (\mathcal{P}_i, \sigma_i, \sigma'_i, \mathcal{Q}_i)_{i=1,\ldots,\rho}$ are made public while S remains secret.

5.1 An impossibility result based on κ-symmetry

Recall that $\mathcal{L}_i$ refers to the linear function defined by $\mathcal{L}_i(\boldsymbol{v}) = s_i \cdot \boldsymbol{v}$. We denote by P_S^γ the set of polynomials defined by

$$P_S^\gamma = \left\{ \prod_{t=1}^{\gamma} \mathcal{L}_{i_t} | i_t \in \{1, \ldots, 2\kappa\} \right\}$$

A CPA attacker is naturally interested in these polynomials: for instance, $\mathsf{Decrypt}$ can be written with polynomials of P_S^1. A representation R_f of an arbitrary function f is said to be effective if its storage is polynomial and its evaluation is polynomial-time. The following proposition ensures that the polynomials of $P_S^{\gamma<\kappa}$ cannot be recovered: this is derived from symmetry properties related to the parameter κ.

Proposition 3. *Let $\gamma \in \mathcal{K} \setminus \{0\}$ and let $\phi \in P_S^\gamma$. Under the factoring assumption, a CPA attacker cannot recover any effective representation R_ϕ of ϕ.*

Proof. (Sketch.) Details are given in Appendix D.

The i^{th} row of S is denoted by s_i. Let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r$ be the encryptions of $x_1, \ldots, x_r$ received by the CPA attacker from the encryption oracle, i.e. $\boldsymbol{c}_i = S^{-1}(r_{i0}x_{i0}, r_{i0}, \ldots, r_{i,\kappa-1}x_{i,\kappa-1}, r_{i,\kappa-1})$. Let us consider the κ tuples $(y_\ell)_{\ell=0,\ldots,\kappa-1}$ defined by

$$y_\ell = ((x_{i\ell}, r_{i\ell})_{i=1,\ldots,r}, s_{2\ell+1}, s_{2\ell+2})$$

These tuples are generated to probability distribution statistically indistinguishable from the probability distribution of Problem 1 (note that only the values $x_{i\ell}$ are involved in additive constraints). By construction, a CPA attacker only knows κ-symmetric polynomials defined over $(y_0, \ldots, y_{\kappa-1})$. By Lemma 5, it is not possible to polynomially recover the evaluation of any monomial π (s.t. $\deg \pi < \kappa$) defined over the coefficients s_{ij} assuming the hardness of factoring. As the knowledge of R_ϕ can be used to evaluate such a monomial π, R_ϕ cannot be recovered.

□

Corollary 2. *Assuming the hardness of factoring, S cannot be recovered by a CPA attacker.*

The decryption of a ciphertext $\boldsymbol{c}$ consists of evaluating the following function

$$\phi_D = \sum_{\ell=0}^{\kappa-1} \mathcal{L}_{2\ell+1}/\mathcal{L}_{2\ell+2}$$

Clearly, ϕ_D is a κ-symmetric polynomial in the sense that it remains unchanged if the tuples y_ℓ (as defined in the proof of Proposition 3) are permuted. Therefore, Lemma 5 cannot be directly used to prove that ϕ_D cannot be recovered. However, by Proposition 3, the linear functions $\mathcal{L}_i$, cannot be recovered implying that ϕ_D cannot be naturally represented as the sum of rational functions $\sum_{\ell=0}^{\kappa-1} \mathcal{L}_{2\ell+1}/\mathcal{L}_{2\ell+2}$. More generally, the representations involving polynomials of $P_S^{\gamma<\kappa}$ cannot be recovered. The only way to represent[5] ϕ_D without involving such polynomials consists of writing ϕ_D as a ratio of two κ-symmetric polynomials ϕ_1/ϕ_2, i.e.

$$\phi_D = \frac{\phi_1 =}{\phi_2 =} \frac{\sum_{\ell=0}^{\kappa-1} \mathcal{L}_{2\ell+1} \prod_{\ell' \neq \ell} \mathcal{L}_{2\ell'+2}}{\prod_{\ell=0}^{\kappa-1} \mathcal{L}_{2\ell+2}}$$

Note that ϕ_1 and ϕ_2 are sums of polynomials of P_S^κ and the monomial coefficients of ϕ_1 and ϕ_2 are κ-symmetric while the factored or semi-factored representations of these polynomials cannot be recovered according to Proposition 3. By construction, for any encryption $\boldsymbol{c} \leftarrow \mathsf{Encrypt}(K, x)$, the polynomial $\Phi = \phi_1 - x\phi_2$ satisfies $\Phi(\boldsymbol{c}) = 0$. It could be thus recovered by solving a linear system where the variables are the monomial coefficients of Φ. This attack is called "*attack by linearization*". However, this attack fails provided $\kappa = \Theta(\lambda)$ because the expanded representation of Φ is exponential-size in this case (see Remark 1). Nevertheless, efficient attacks by linearization involving the operators $\mathcal{Q}_i$ could be imagined: this is the object of the next section.

5.2 A conjecture about IND-CPA security

Throughout this section, $\kappa = \Theta(\lambda)$. Roughly speaking, we conjecture that our scheme is IND-CPA secure if a CPA attacker cannot mount any *attack by linearization* (informally defined in the previous section). This section aims to justify and to formalize it.

We consider an attacker $\mathcal{A}$ which has access to an encryption oracle and which can use the public operators $(\mathcal{Q}_i)_{i=1,\ldots,\rho}$ in an arbitrary way. Clearly, to break IND-CPA security, $\mathcal{A}$ wishes to recover $x_1 \in \mathbb{Z}_n$ and a polynomial $\Phi \in \mathbb{Z}_n[X_1, \ldots, X_{2\kappa}]$ such that $\Phi \circ \mathsf{Encrypt}(K, x_1) \overset{s}{\not\equiv} \Phi \circ \mathsf{Encrypt}(K, 0)$. However, to recover Φ with attacks by linearization[6], it should be ensured that $\Phi \circ \mathsf{Encrypt}(K, x_1) = 0$ (or any other constant) with non-negligible probability[7]. This leads us to restrict the set of distinguishing functions to the polynomials Φ satisfying

$$\mathsf{Adv}^{\Phi, x_1} \overset{\mathsf{def}}{=} |\mathsf{Pr}(\Phi \circ \mathsf{Encrypt}(K, x_1) = 0) - \mathsf{Pr}(\Phi \circ \mathsf{Encrypt}(K, 0) = 0)| \tag{1}$$

[5] without using the operators $\mathcal{Q}_i$.
[6] It consists of recovering the monomial coefficients of Φ (indexed by S) by solving a linear system.
[7] By Lemma 4, it follows that $\Phi \circ \mathsf{Encrypt}(K, x_1) = 0$ with probability 1.

is non-negligible.

By construction of Encrypt (see the previous section), the degree of such polynomials Φ is larger than κ implying that their expanded representation is exponential-size provided $\kappa = \Theta(\lambda)$. Moreover, from Proposition 3, $\mathcal{A}$ cannot expect to recover factored or semi-factored representations. Nevertheless, efficient attacks by linearization could appear by composing functions. For concreteness, $\mathcal{A}$ could generate new vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t$ by applying the operators $\mathcal{Q}_i$ to the challenge encryption $\boldsymbol{c}_1$ and new encryptions[8] $\boldsymbol{c}_2, \ldots, \boldsymbol{c}_r$ in the hope that there exists a *small* polynomial ϕ s.t. $\phi(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t) = \Phi(\boldsymbol{c}_1)$ (Φ satisfying (1)). We restrict the generation of these new vectors in a natural way encapsulated in the following definition.

Definition 4. *GV denotes an arbitrary efficient procedure with encryptions $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r$ as input which outputs vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_t$ built by recursively applying operators $\mathcal{Q}_i$ and/or linear combinations.*

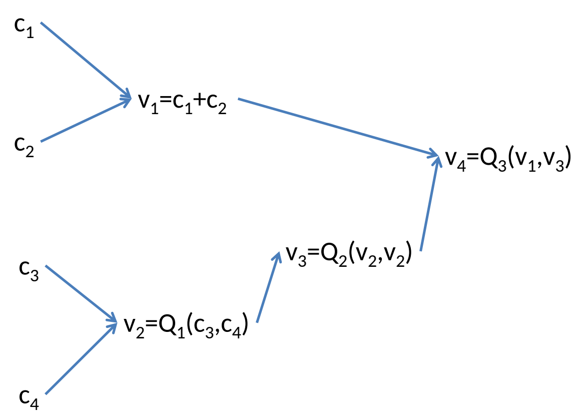

Fig. 3. Example of a procedure GV. By construction, each component of $S\boldsymbol{v}_i$ can be written as a (known) polynomial defined over the components of $S\boldsymbol{c}_1, \ldots, S\boldsymbol{c}_r$.

By fixing $\boldsymbol{c}_2, \ldots, \boldsymbol{c}_r$ and by using the arithmetic expression of the operators $\mathcal{Q}_i$ in GV, $\phi \circ \mathsf{GV}$ can be seen as a polynomial $\phi \circ \mathsf{GV}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r} \in \mathbb{Z}_n[X_1, \ldots, X_{2\kappa}]$ satisfying $\phi \circ \mathsf{GV}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}(\boldsymbol{c}_1) = \phi \circ \mathsf{GV}(\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_r)$. Let $\boldsymbol{v}$ be a vector output by GV. By construction, each component of $S\boldsymbol{v}$ can be expressed by a (known) polynomial defined over the components of $S\boldsymbol{c}_1, \ldots, S\boldsymbol{c}_r$. The key idea of our analysis is that there is no other relevant way to use the encryption oracle and the operators $\mathcal{Q}_i$. This implicitly means that an attacker cannot derive new operators[9] $\mathcal{Q}$ (for chosen families $\mathcal{P}$). Corollary 2 ensures that this cannot be directly done by recovering S. This is extensively discussed in Appendix F where it is shown that this problem is difficult in general (the discussion is based on Lemma 6 and a modified version of Lemma 5).

Informally, we restrict the set of functions satisfying (1) to the functions $\phi \circ \mathsf{GV}$ where ϕ is a small polynomial[10], i.e. $\deg \phi = o(\lambda)$.

Conjecture 2. Assume $\kappa = \Theta(\lambda)$. The CPA attacker $\mathcal{A}$ arbitrarily chooses $x \in \mathbb{Z}_n^r$ and generates encryptions $\boldsymbol{c}_2, \ldots, \boldsymbol{c}_r$ of respectively $x_2, \ldots, x_r$ by using the encryption oracle. The encryption scheme is IND-CPA secure if $\mathcal{A}$ cannot output[11] a procedure GV and an arithmetic circuit of a $o(\lambda)$-degree polynomial $\phi \in \mathbb{Z}_n[X_1, \ldots, X_{2\kappa t}]$ s.t. $\mathsf{Adv}^{\phi \circ \mathsf{GV}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}, x_1}$ is non-negligible.

[8] obtained by requesting the encryption oracle.

[9] Definition 4 is irrelevant otherwise.

[10] If $\deg \phi = \Omega(\lambda)$ then ϕ cannot be recovered with attacks by linearization because it is exponential-size (see Remark 1).

[11] with non-negligible probability

Remark 4. Ways to randomize QGen are proposed in Appendix F. This randomization makes the system of nonlinear equations derived from the operators $\mathcal{Q}_i$ widely unknown. One can reasonably wonder whether the factoring assumption can be removed by adding this randomness. In other words, does Conjecture 2 remain true if n is a small/large prime? If so, the security could entirely rely on the difficulty of solving systems of nonlinear equations.

6 Security Analysis

Proposition 4. *Assume $\kappa = \Theta(\lambda)$. The additively homomorphic encryption scheme is IND-CPA secure assuming Conjecture 1 and Conjecture 2.*

Proof. (Sketch.) Details are given in Appendix E.

To simplify the proof (and the task of the attacker), we fix $S = \mathsf{Id}$. Let ϕ be an arbitrary non-null polynomial of $\mathbb{Z}_n[X_1, \ldots, X_{2\kappa t}]$ such that $\deg \phi < \kappa$ chosen by the CPA attacker. The polynomial $\phi \circ \mathsf{GV}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}$ can be written as a polynomial $\overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r} \in \mathbb{Z}_n[X_1, \ldots, X_{2\kappa}]$ defined by

$$\overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}(r_0, \ldots, r_{\kappa-1}, x_0, \ldots, x_{\kappa-1}) = \phi \circ \mathsf{GV}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r)$$

where $\boldsymbol{c}_1 = (r_0 x_0, r_0, \ldots, r_{\kappa-1}x_{\kappa-1}, r_{\kappa-1}) \leftarrow \mathsf{Encrypt}(K, x)$.

By fixing $x_0 + \cdots + x_{\kappa-1} = x$ and by using the equality $x_{\kappa-1} = x - x_{\kappa-2} - \ldots - x_0$, $\overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}$ can be written as a polynomial $\overline{\phi \circ \mathsf{GV}}_{x,\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}$ of $\mathbb{Z}_n[X_1, \ldots, X_{2\kappa-1}]$ satisfying

$$\overline{\phi \circ \mathsf{GV}}_{x,\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}(r_0, \ldots, r_{\kappa-1}, x_0, \ldots, x_{\kappa-2}) = \overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}(r_0, \ldots, r_{\kappa-1}, x_0, \ldots, x_{\kappa-1})$$

We show that this polynomial is not null implying that $\overline{\phi \circ \mathsf{GV}}_{x,\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}(r_0, \ldots, r_{\kappa-1}, x_0, \ldots, x_{\kappa-2}) = 0$ with negligible probability assuming Conjecture 1 (see Lemma 4) proving that $\mathsf{Pr}(\phi \circ \mathsf{GV}_{\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}(\boldsymbol{c}_1) = 0)$ is negligible for each $x \in \mathbb{Z}_n$. It follows that the scheme is IND-CPA secure assuming Conjecture 2.

□

Remark 5. Randomness can be introduced in AddGen by outputting $\mathsf{Add} \leftarrow \mathsf{QGen}(S, (p_1, p_2), \sigma, \sigma')$ where σ, σ' randomly chosen in $\mathcal{K}$. Moreover, by introducing randomness as explained in Appendix F, one may reasonably think that the factoring assumption is not required anymore.

7 Efficiency

The computation of an operator $\mathcal{Q}$ requires $O(\kappa^3)$ multiplications in $\mathbb{Z}_n$. Thus, the running time of Add $O(\kappa^3 M(n))$ where $M(n)$ denotes the runtime of multiplications done in $\mathbb{Z}_n$. The running time of decryption is $O(\kappa^2 M(n))$. A ciphertext contains a 2κ-vector in $\mathbb{Z}_n$, implying that the ratio of ciphertext size to plaintext size is 2κ. In terms of storage, each operator $\mathcal{Q}$ contains $O(\kappa^3)$ elements of $\mathbb{Z}_n$, which leads to a space complexity in $O(|n|\kappa^3)$.

We identified only the attack by linearization described in Section 5.1. To ensure the irrelevancy of this attack, it suffices to choose $\kappa \geq 30$: in this case, the linear system contains more than 10^{30} variables. By choosing $\kappa = 30$, applying the operator Add requires approximatively $3 \cdot 10^4$ modular multiplications.

8 Perspectives

The first perspective of this work is to build an FHE by developing a multiplicative operator. To get a formal security proof under Conjecture 1 and Conjecture 2, it suffices (as done in the proof of Proposition 4) to show that $\overline{\phi \circ \mathsf{GV}}_{x,\boldsymbol{c}_2,\ldots,\boldsymbol{c}_r}$ is not identically zero. While we did not find a provably-secure construction, we propose the simplest construction potentially secure. The security proof is left as an open problem for further research.

Construction. Let $S \leftarrow \mathsf{KeyGen}(\lambda)$, let $\mathsf{Add} \leftarrow \mathsf{AddGen}(S)$ and let $\mathcal{P} = (p_1, p_2)$ be the family of polynomials: $\mathbb{Z}_n^2 \times \mathbb{Z}_n^2 \to \mathbb{Z}_n$ defined by

- $p_1(x, x') = x_1 x_1'$
- $p_2(x, x') = x_2 x_2'$

$\mathsf{MultGen}(S)$ outputs the operators $\mathcal{Q}_i \leftarrow \mathsf{QGen}(S, \mathcal{P}, \sigma_i, \sigma_i')$ where σ_i, σ_i' are randomly chosen in $\mathcal{K}$ ensuring that $\bigcup_{i \in \mathcal{K}} \{\sigma_i - \sigma_i' \mod \kappa\} = \mathcal{K}$ (a description of $\mathcal{Q}_i$ is given in Figure 1).

Mult$(\boldsymbol{c}, \boldsymbol{c}')$

1. $\boldsymbol{v}_0 \leftarrow \mathcal{Q}_0(\boldsymbol{c}, \boldsymbol{c}')$
2. for $i = 1$ to $\kappa - 1$
   (a) $\boldsymbol{w}_i \leftarrow \mathcal{Q}_i(\boldsymbol{c}, \boldsymbol{c}')$
   (b) $\boldsymbol{v}_i \leftarrow \mathsf{Add}(\boldsymbol{v}_{i-1}, \boldsymbol{w}_i)$
3. Output $\boldsymbol{v}_{\kappa-1}$

Proposition 5. *Let Mult $\leftarrow$ MultGen(S) is a valid multiplicative homomorphic operator.*

Proof. Let $\boldsymbol{c} = S^{-1}(r_0 x_0, r_0, \ldots, r_{\kappa-1} x_{\kappa-1}, r_{\kappa-1})$ and $\boldsymbol{c}' = S^{-1}(r_0' x_0', r_0', \ldots, r_{\kappa-1}' x_{\kappa-1}', r_{\kappa-1}')$ be two encryptions of respectively x, x' and let $\boldsymbol{c}'' \leftarrow \mathsf{Mult}(\boldsymbol{c}, \boldsymbol{c}')$. We easily check that:

$$\boldsymbol{v}_{\kappa-1} = S^{-1}\begin{pmatrix} \prod_{i=0}^{\kappa-1} r_{\sigma_i} r'_{\sigma_i'} \times \sum_{i=0}^{\kappa-1} x_{\sigma_i} x'_{\sigma_i'} \\ \prod_{i=0}^{\kappa-1} r_{\sigma_i} r'_{\sigma_i'} \\ \prod_{i=0}^{\kappa-1} r_{\sigma_i+1 \mod \kappa} r'_{\sigma_i'+1 \mod \kappa} \times \sum_{i=0}^{\kappa-1} x_{\sigma_i+1 \mod \kappa} x'_{\sigma_i'+1 \mod \kappa} \\ \prod_{i=0}^{\kappa-1} r_{\sigma_i+1 \mod \kappa} r'_{\sigma_i'+1 \mod \kappa} \\ \cdots \\ \prod_{i=0}^{\kappa-1} r_{\sigma_i-1 \mod \kappa} r'_{\sigma_i'-1 \mod \kappa} \times \sum_{i=0}^{\kappa-1} x_{\sigma_i-1 \mod \kappa} x'_{\sigma_i'-1 \mod \kappa} \\ \prod_{i=0}^{\kappa-1} r_{\sigma_i-1 \mod \kappa} r'_{\sigma_i'-1 \mod \kappa} \end{pmatrix}$$

As $\bigcup_{i \in \mathcal{K}} \{\sigma_i - \sigma_i' \mod \kappa\} = \mathcal{K}$, it is ensured that each product $x_i x_j'$ appears only once in the above sums. It follows that $\mathsf{Decrypt}(\boldsymbol{c}'') = \sum_{i,j} x_i x_j' = x x'$.
□

Can the values σ_i, σ_i' be recovered? We are strongly convinced that this problem is difficult but we do not provide any formal result in this sense. If we assume that a CPA attacker cannot recover σ_i, σ_i', recovering a distinguishing function $\phi \circ \mathsf{GV}$ seems very hard.

A second motivating perspective would consist of removing the factoring assumption required to prove formal results (Lemma 1, Lemma 5 and Proposition 3). This assumption defeats the whole "post-quantum" purpose of multivariate cryptography [11]. In our opinion, this can be overcome by introducing randomness into our scheme (see Remark 4 and Remark 5). Finally, we think that efficient multilinear maps [2], [5] or functional encryptions [1] can be developed with the material of this paper.

References

1. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. *Commun. ACM*, 55(11):56–64, 2012.
2. Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *IACR Cryptology ePrint Archive*, 2002:80, 2002.
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. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 1–17, 2013.
6. Gérald Gavin. An efficient fhe based on the hardness of solving systems of non-linear multivariate equations. Cryptology ePrint Archive, Report 2013/262, 2013. http://eprint.iacr.org/.
7. Gérald Gavin. An efficient fhe proposal based on the hardness of solving systems of nonlinear multivariate equations (ii). *IACR Cryptology ePrint Archive*, 2013:740, 2013.
8. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
9. Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, pages 465–482, 2012.
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. Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *EUROCRYPT*, pages 33–48, 1996.
12. Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. In *ASIACRYPT*, pages 377–394, 2010.
13. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010.
14. 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 Toy implementation of an operator $\mathcal{Q}$

In this section, we propose a concrete computation of $\mathcal{Q} \leftarrow \mathsf{QGen}(S, (p_1, p_2), 0, 0)$ with $p_1(x, x') = x_1 x_1'$ and $p_2(x, x') = x_2 x_2'$ for $\kappa = 1$.

Given $S := \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$

with $\Delta = s_{11}s_{22} - s_{12}s_{21} \in Z_n^*$

$$\mathcal{Q}\left(\begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{c}_1' \\ \boldsymbol{c}_2' \end{pmatrix}\right)$$
$$= \Delta^{-1} \begin{bmatrix} (s_{22}s_{11}^2 - s_{12}s_{21}^2)\boldsymbol{c}_1\boldsymbol{c}_1' + (s_{22}s_{11}s_{12} - s_{12}s_{21}s_{22})(\boldsymbol{c}_1\boldsymbol{c}_2' + \boldsymbol{c}_2\boldsymbol{c}_1') + (s_{22}s_{12}^2 - s_{12}s_{22}^2)\boldsymbol{c}_2\boldsymbol{c}_2' \\ (s_{11}s_{21}^2 - s_{21}s_{11}^2)\boldsymbol{c}_1\boldsymbol{c}_1' + (s_{11}s_{21}s_{22} - s_{21}s_{11}s_{12})(\boldsymbol{c}_1\boldsymbol{c}_2' + \boldsymbol{c}_2\boldsymbol{c}_1') + (s_{11}s_{22}^2 - s_{21}s_{12}^2)\boldsymbol{c}_2\boldsymbol{c}_2' \end{bmatrix}$$

B Proofs of the lemmas of Section 2.1

Throughout this section $n = pq$ is a randomly chosen RSA modulus such that $\eta = \lceil \log_2 p \rceil = \lceil \log_2 q \rceil$. Given a polynomial $\phi \in \mathbb{Z}_n[X_1, \ldots, X_r]$, $z_{\phi,p} = |\{x \in \mathbb{Z}_n^r | \phi(x) \equiv 0 \mod p\}|/n^r$ and $z_{\phi,q} = |\{x \in \mathbb{Z}_n^r | \phi(x) \equiv 0 \mod q\}|/n^r$.

B.1 Proof of Lemma 1

To prove this result, we assume the existence of an p.p.t algorithm $\mathcal{A}$ solving our problem and we build an algorithm $\mathcal{B}$ which factors n. Let $\phi \leftarrow \mathcal{A}(n)$. By using the Chinese remainder theorem and the specificities of $\mathcal{A}$, we have:

1. $z_{\phi,p}$ and $z_{\phi,q}$ are non-negligible (otherwise z_ϕ is negligible).
2. $1 - z_{\phi,p}$ or $1 - z_{\phi,q}$ is non-negligible (otherwise $1 - z_\phi$ is negligible).

Assume that $1 - z_{\phi,p}$ is non-negligible and pick $x \in \mathbb{Z}_n^r$ at random. The probability of the conjunction of two following independent events $\phi(x) \not\equiv 0 \mod p$ and $\phi(x) \equiv 0 \mod q$ is non-negligible, i.e. it is equal to $(1 - z_{\phi,p})z_{\phi,q}$. It follows that $q = \gcd(\phi(x), n)$ with non-negligible probability.

□

B.2 Proof of Lemma 2

Assume the existence of a p.p.t-algorithm $\mathcal{A}$ outputting the expanded representation of a non-null polynomial $\phi \in \mathbb{Z}_n[X]$ such that $z_\phi = 1$. Clearly $\phi \mod p$ (resp. $\phi \mod q$) is a multiple of $X^p - X$ (resp. $X^q - X$). It follows that ϕ has two polynomials m_1, m_2 such that $k = \deg m_1 - \deg m_2$ is a multiple of $p - 1$ and/or $q - 1$. We distinguish the two following cases:

1. k is a multiple of $p - 1$ but not of $q - 1$ (resp. k is a multiple of $q - 1$ but not of $p - 1$). In this case, $g^k - g \mod n$ is a non-zero multiple of p (resp. q) with a probability larger than $1/2$ (over the choice of g). It follows that $p = \gcd(n, g^k - g \mod n)$.
2. k is a multiple of $\mathsf{lcm}(p - 1, q - 1)$. Since k is even, $k = 2^t r$ with r odd and $t \geq 1$. A straightforward argument shows that if g is chosen at random from $\mathbb{Z}_n$ then with probability at least 1/2 (over the choice of g) one of the elements in the sequence $g^{k/2}, g^{k/4}, \ldots, g^{k/2^t} \mod n$ is a non-trivial square root of unity (not in $\{1; -1\}$) that reveals the factorization of n.

As the exponents of ϕ are polynomial-size, k is polynomial-size implying that all the previous computations are polynomial-time.

□

B.3 Proof of Lemma 3

We assume the existence of a p.p.t algorithm which outputs an arithmetic circuit of $\phi \in \mathbb{Z}_n[X]$ such that z_ϕ is non-negligible. Let $\phi_0 = X \cdot \phi$. Clearly z_{ϕ_0} is non-negligible implying that $1 - z_{\phi_0}$ is negligible (from Lemma 1). Thus, it can be assumed that $1 - z_{\phi_0} < 1/2\eta^2$. It follows that $1 - z_{\phi_0,p} < 1/2\eta^2$ and $1 - z_{\phi_0,q} < 1/2\eta^2$.

Let $\phi, \phi' \in \mathbb{Z}_n[X]$ such that $\phi'(0) = 0$ and let ϕ'' denote the polynomial defined by $\phi'' = \phi' \circ (r \cdot \phi)$. Clearly, if r is uniform over $\mathbb{Z}_n$ then the expectation of $1 - z_{\phi'',p}$ is equal to

$$E(1 - z_{\phi'',p}) = (1 - z_{\phi,p})(1 - z'_{\phi,p})$$

Since $1 - z_{\phi'',p} \geq 0$, the probability that $1 - z_{\phi'',p} \geq a \cdot E(1 - z_{\phi'',p})$ is smaller than $1/a$. It follows that

$$1 - z_{\phi'',p} \leq (1 - z_{\phi,p})/2 \tag{2}$$

with probability larger than $1 - 1/\eta^2$ provided $1 - z'_{\phi,p} \leq 1/2\eta^2$.

Let us consider the recursive sequence defined by $\phi_i(x) = \phi_0 \circ (r_i \cdot \phi_{i-1})$ where r_i uniform over $\mathbb{Z}_n$. By iterating the inequality (2), we get

$$1 - z_{\phi_i,p} \leq 2^{-i}(1 - z_{\phi_0,p})$$

with probability larger than $1 - i/\eta^2$. It follows that $1 - z_{\phi_\eta,p} < 2^{-\eta}$ implying that

$$1 - z_{\phi_\eta,p} = 0$$

with probability larger than $1 - 1/\eta > 2/3$. Similarly, we show that $1 - z_{\phi_\eta,q} = 0$ with probability larger than $1 - 1/\eta > 2/3$ implying that $1 - z_{\phi_\eta} = 0$ with probability larger than $1/3$. Moreover, ϕ_η is not null if $r_i \neq 0$ for any $i \in \{1, \ldots, \eta\}$ implying that ϕ_η is null with negligible probability. Consequently, assuming Conjecture 1, it is difficult to recover ϕ_η implying that it is difficult to recover ϕ_0 and thus ϕ. This concludes the proof.

□

B.4 Proof of Lemma 4

This result can be shown by induction over r. By Lemma 3, the result is true for $r = 1$. Let us assume the result true for any $r < r_0$ but not for $r = r_0$, i.e. there exists a p.p.t-algorithm $\mathcal{A}$ which outputs an arithmetic circuit of a non-null polynomial $\phi \in \mathbb{Z}_n[X_1, \ldots, X_{r_0}]$ such that z_ϕ is non-negligible. By fixing $X_2, \ldots, X_{r_0}$ to randomly chosen values $x_2, \ldots, x_{r_0} \in \mathbb{Z}_n$, we get an univariate polynomial $\phi_{x_2,\ldots,x_{r_0}}$ defined by $\phi_{x_2,\ldots,x_{r_0}}(x_1) = \phi(x_1, \ldots, x_{r_0})$. At least one monomial coefficient of $\phi_{x_2,\ldots,x_{r_0}}$ can be written as a non-null $(r_0 - 1)$-variate polynomial φ evaluated over $x_2, \ldots, x_{r_0}$. By using the induction hypothesis, z_φ is negligible implying that $\varphi(x_2, \ldots, x_{r_0}) \neq 0$ with overwhelming probability. It follows that $\phi_{x_2,\ldots,x_{r_0}}$ is not null with overwhelming probability. Moreover, as z_ϕ is non-negligible, $z_{\phi_{x_2,\ldots,x_{r_0}}}$ is non-negligible with non-negligible probability. This contradicts the case $r = 1$.

□

C Proofs of the lemmas of Section 2.2

C.1 Proof of Lemma 5

We start by proving a preliminary result (which can be seen as a special case of Conjecture 1).

Lemma 7. *Let p be an η-bit prime and π_1, π_2 be two monomials of $\mathbb{Z}_p[X_1, \ldots, X_r]$ such that $\pi_1 \neq \pi_2$ and $\deg \pi_1$, $\deg \pi_2$ polynomials in η. The probability that $\pi_1(x) = \pi_2(x)$ is negligible if x uniform over $\mathbb{Z}_p^r$.*

Proof. (Sketch.) Consider the case $r = 1$. As $\pi_1 \neq \pi_2$, $\pi(X) = \pi_1(X)/\pi_2(X) = X^\gamma$ where $\gamma = \deg \pi_1 - \deg \pi_2 \neq 0$ is polynomial. So the number of $x \in \mathbb{Z}_p^*$ such that $\pi(x) = 1$ is polynomial, i.e. it is smaller than γ^2.

□

Let D be the probability distribution of $(y_0, \ldots, y_{\kappa-1})$. The proof consists of building a polynomial factoring algorithm $\mathcal{A}$ by using a solver $\mathcal{B}$ of Problem 1 as subroutine[12]. Let us consider the following polynomial-time algorithm $\mathcal{A}$:

Input: $n = pq$

$(s_1, \ldots, s_m, \pi) \leftarrow \mathcal{A}_S(n)$

Repeat

1. *Let* $(y_0, \ldots, y_{\kappa-1}) \xleftarrow{\$} D$
2. *Compute* $\overline{s}_j = s_j(y_0, \ldots, y_{\kappa-1})$ *for all* $j = 1, \ldots, m$.
3. *Compute* $\mathit{\Pi} = \pi(y_0, \ldots, y_{\kappa-1})$
4. *Apply* $\mathcal{B}$ *on the inputs* $\overline{s}_1, \ldots, \overline{s}_m$*, i.e.* $\mathit{\Pi}_\mathcal{B} \leftarrow \mathcal{B}(\overline{s}_1, \ldots, \overline{s}_m)$

until $\gcd(\mathit{\Pi} - \mathit{\Pi}_\mathcal{B}, n) \neq 1$

output $\gcd(\mathit{\Pi} - \mathit{\Pi}_\mathcal{B}, n)$

By construction, this algorithm is correct. Let us show that it terminates in polynomial-time. First, each step of $\mathcal{A}$ can be computed in polynomial-time implying that $\mathcal{A}$ is polynomial if the expectation of the number of steps of $\mathcal{A}$ is polynomial (or equivalently, if the probability to get $\gcd(\mathit{\Pi} - \mathit{\Pi}_\mathcal{B}, n) \neq 1$ is not negligible). As $\deg \pi < \kappa$, π is not κ-symmetric implying that there exists $\sigma^* \in \mathcal{K}$ and $y_0^*, \ldots, y_{\kappa-1}^*$ such that $\pi(y_0^*, \ldots, y_{\kappa-1}^*) \neq \pi(y_{\sigma^*}^*, \ldots, y_{\sigma^*-1 \mod \kappa}^*)$. Let π_{σ^*} be the monomial s.t. $\deg \pi_{\sigma^*} = \deg \pi$ defined by $\pi_{\sigma^*}(y_0, \ldots, y_{\kappa-1}) = \pi(y_{\sigma^*}, \ldots, y_{\sigma^*-1 \mod \kappa})$. By construction, $\pi \neq \pi_{\sigma^*}$ implying that

$$\pi(y_0, \ldots, y_{\kappa-1}) \not\equiv \pi_{\sigma^*}(y_0, \ldots, y_{\kappa-1}) \mod q \tag{3}$$

with overwhelming probability according to Lemma 7 (because the variables $y_{\ell i}$ involved[13] in π are i.i.d. according to the uniform distribution over $\mathbb{Z}_n$).

Let us consider the function $h : \left(\mathbb{Z}_n^t\right)^\kappa \rightarrow \left(\mathbb{Z}_n^t\right)^\kappa$ such that $(y'_0, \ldots, y'_{\kappa-1}) = h(y_0, \ldots, y_{\kappa-1})$ is defined by

- $y'_{\ell i} \equiv y_{\ell i} \mod p$ for all $(\ell, i) \in \mathcal{K} \times T$
- $y'_{\ell i} \equiv y_{\ell+\sigma^* \mod \kappa, i} \mod q$ for all $(\ell, i) \in \mathcal{K} \times T$.

Because of the symmetry of the (additive) constraints, if $(y_0, \ldots, y_{\kappa-1})$ satisfies the constraints of Problem 1 then $(y'_0, \ldots, y'_{\kappa-1})$ also satisfies these constraints. It implies that $(y_0, \ldots, y_{\kappa-1})$ and $(y'_0, \ldots, y'_{\kappa-1})$ have the same probability under D, i.e.

$$\mathsf{Pr}_D(y_0, \ldots, y_{\kappa-1}) = \mathsf{Pr}_D(y'_0, \ldots, y'_{\kappa-1})$$

[12] $\mathcal{B}$ is assumed to solve Problem 1 if it outputs π with non-negligible probability

[13] According to Problem 1, $i \in I_F$.

Let $\Pi' = \pi(y'_0, \ldots, y'_{\kappa-1})$. As the functions s_j are κ-symmetric polynomials, we get$s_j(y'_0, \ldots, y'_{\kappa-1}) = s_j(y_0, \ldots, y_{\kappa-1})$ for all $j = 1, \ldots, m$. It follows that

$$\mathsf{Pr}_D(\Pi_{\mathcal{B}} = \Pi) = \mathsf{Pr}_D(\Pi_{\mathcal{B}} = \Pi')$$

As $\mathcal{B}$ is assumed to solve Problem 1, $\mathsf{Pr}_D(\Pi_{\mathcal{B}} = \Pi)$ is non-negligible implying that $\mathsf{Pr}_D(\Pi_{\mathcal{B}} = \Pi')$ is non-negligible.

By construction $\Pi \equiv \Pi' \mod p$. Since $\Pi' \equiv \pi_{\sigma^*}(y_0, \ldots, y_{\kappa-1}) \mod q$, Equation (3) implies that $\Pi \not\equiv \Pi' \mod q$ with overwhelming probability. It follows that $p = \gcd(n, \Pi - \Pi')$ with overwhelming probability. Consequently, $\mathcal{A}$ terminates (when $\Pi_{\mathcal{B}} = \Pi'$) in polynomial-time.

□

C.2 Proof of Lemma 6

A multiset is a generalization of the notion of a set in which members are allowed to appear more than once. For example, there is a unique set $\{a, b\}$ containing elements a and b and no others, but there are many multisets containing a and b (and no others) with various multiplicities. For instance, in the multiset $\{a, a, b\}$, a has multiplicity 2 and b has multiplicity 1. Given a set E, $E^{[u]}$ denotes the set of multisets $\{x_1, \ldots, x_u\}$ such that $x_i \in E$.

Let $I : \{1, \ldots, t\}^{[d]} \to E_d$ be the one-to-one function defined by $I(M) = \sum_{\ell=0}^{\kappa-1} \prod_{i \in M} X_{i+\ell t}$. Let $\alpha \in \mathbb{N} \setminus \{0\}$. As the application $J : E_d^{[\alpha]} \to \mathbb{Z}_n[X_1, \ldots, X_{\kappa t}]$ defined by $J(\phi_0, \ldots, \phi_{\alpha-1}) = \phi_0 \cdots \phi_{\alpha-1}$ is injective, any multiset $\Phi = \{\phi_0, \ldots, \phi_{\alpha-1}\} \in E_d^{[\alpha]}$ can be identified to the polynomial $J(\Phi) = \phi_0 \cdots \phi_{\alpha-1}$.

Lemma 8. *The polynomials of $E_d^{[\alpha]}$ are linearly independent for any $\alpha \leq \kappa$.*

Proof. Let $\Phi_0 = \{\phi_0, \ldots, \phi_{\alpha-1}\} \in E_d^{[\alpha]}$. By construction, $\Phi_0 = \prod_{k=0}^{\alpha-1} \sum_{\ell=0}^{\kappa-1} \prod_{i \in I^{-1}(\phi_k)} X_{i+\ell t}$. Clearly, the monomial $\prod_{k=0}^{\alpha-1} \prod_{i \in I^{-1}(\phi_k)} X_{i+kt}$ belongs to Φ_0 and does not belong to any other polynomial $\Phi \in E_d^{[\alpha]} \setminus \{\Phi_0\}$.

□

Let p, q be two arbitrary polynomials. Without loss of generality, it can be assumed that p, q are homogeneous such that $\deg p = \deg q + 1 \leq \kappa$. The polynomial $r = s_0 \cdot q(s_1, \ldots, s_m) - p(s_1, \ldots, s_m)$ can be written as a linear combination $\mathcal{L}$ over $E_d^{[\deg p]}$. Because the polynomials $s_0, s_1, \ldots, s_m$ are linearly independent, $\mathcal{L}$ is not zero. As $\deg p \leq \kappa$, Lemma 8 ensures that the polynomials of $E_d^{[\deg p]}$ are linearly independent implying that r is not identically equal to the zero polynomial.

□

D Proof of Proposition 3

The i^{th} row of S is denoted by s_i. Let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_r$ be the encryptions of $x_1, \ldots, x_r$ received by the CPA attacker from the encryption oracle, i.e. $\boldsymbol{c}_i = S^{-1}(r_{i0}x_{i0}, r_{i0}, \ldots, r_{i,\kappa-1}x_{i,\kappa-1}, r_{i,\kappa-1})$. Let us consider the κ tuples $(y_\ell)_{\ell=0,\ldots,\kappa-1}$ defined by

$$y_\ell = ((x_{i\ell}, r_{i\ell})_{i=1,\ldots,r}, s_{2\ell+1}, s_{2\ell+2})$$

By noticing that a randomly chosen matrix S is not invertible with negligible probability, these tuples are generated to probability distribution statistically indistinguishable from the probability distribution of Problem 1 (note that only the $x_{i\ell}$ are involved in additive constraints). By construction, each component of $\boldsymbol{c}_i$ can be written as κ-symmetric polynomial defined over $(y_0, \ldots, y_{\kappa-1})$.

We denote by $S^{[0]}$ the two first rows of S, $S^{[1]}$ the two next rows... and $S^{[\kappa-1]}$ the two last rows of S. Given an arbitrary $\tau \in \mathcal{K}$, S_τ denotes the matrix where the two fist rows are $S^{[\tau]}$, the two next rows are $S^{[\tau+1 \mod \kappa]}$... and the two last rows are $S^{[\tau-1 \mod \kappa]}$. By construction,

$$\mathsf{QGen}(S, \mathcal{P}, \sigma, \sigma') = \mathsf{QGen}(S_\tau, \mathcal{P}, \sigma, \sigma')$$

It follows that each monomial coefficient of $\mathcal{Q}$ can be written as a κ-symmetric multivariate polynomial defined over $(y_0, \ldots, y_{\kappa-1})$.

Consequently, a CPA attacker only knows κ-symmetric polynomials defined over $(y_0, \ldots, y_{\kappa-1})$. By Lemma 5, it is not possible to polynomially recover any non κ-symmetric monomial π defined over the coefficients s_{ij} assuming the hardness of factoring.

Let $\phi \in P_S^\gamma$, i.e. $\phi(\boldsymbol{v}) = \prod_{t=1}^{\gamma} \mathcal{L}_{i_t}(\boldsymbol{v})$ and let $\pi = \prod_{t=1}^{\gamma} s_{i_t 1}$. Because $\gamma < \kappa$, π is an evaluation of a monomial defined over $(y_0, \ldots, y_{\kappa-1})$ such that $\deg \pi < \kappa$. Clearly $\pi = \phi(1, 0, 0, \ldots)$ implying that the knowledge of R_ϕ can be used to efficiently compute π. By Lemma 5, π cannot be recovered implying that R_ϕ cannot be recovered.

□

E Proof of Proposition 4

We start by proving a useful algebraic result.

Lemma 9. *Let $\phi \in Z_n[X_0, \ldots, X_{\kappa-1}, Y_0, \ldots, Y_{\kappa-1}]$ be a non-null polynomial such that each monomial $X_0^{e_0} \cdots X_{\kappa-1}^{e_{\kappa-1}} Y_0^{e'_0} \cdots Y_{\kappa-1}^{e'_{\kappa-1}}$ satisfies*

- $\exists i \in \mathcal{K}$ *s.t.* $e_i = e'_i = 0$
- $e_i = 0 \Rightarrow e'_i = 0$

For any $\alpha \in \mathbb{Z}_n$, the polynomial $\phi_\alpha = \phi(X_0, \ldots, X_{\kappa-1}, Y_0, \ldots, Y_{\kappa-2}, \alpha - Y_0 - \ldots - Y_{\kappa-2})$ is not null.

Proof. Let $\phi = \sum_{i=1}^{\rho} a_i M_i$ where $M_i = X_0^{e_{i0}} \cdots X_{\kappa-1}^{e_{i,\kappa-1}} Y_0^{e'_{i0}} \cdots Y_{\kappa-1}^{e'_{i,\kappa-1}}$ and $a_i \in \mathbb{Z}_n^*$, let $m = \max_i e'_{i,\kappa-1}$.

If $m = 0$ then the result is trivially true. Thus, one can assume that $m > 0$. By using the equality $Y_{\kappa-1} = \alpha - Y_0 - \ldots - Y_{\kappa-2}$, we have $\phi_\alpha = \sum_{i=0}^{\rho} a_i (\alpha - Y_0 - \ldots - Y_{\kappa-2})^{e'_{i,\kappa-1}} M'_i$ where $M'_i = X_0^{e_{i,0}} \cdots X_{\kappa-1}^{e_{i,\kappa-1}} Y_0^{e'_{i,0}} \cdots Y_{\kappa-2}^{e'_{i,\kappa-2}}$.

Given a monomial $M = X_0^{e_0} \cdots X_{\kappa-1}^{e_{\kappa-1}} Y_0^{e'_0} \cdots Y_{\kappa-1}^{e'_{\kappa-1}}$, $E(M)$ denotes the set $\{j \in \mathcal{K} | e_j \neq 0\}$. Let i_0 s.t. $e'_{i_0,\kappa-1} = m$. As $\exists j \in \mathcal{K}$ s.t. $e_{ij} = e_{i'j} = 0$, one can assume that $0 \notin E(M'_{i_0})$. Let us show that the monomial $Y_0^m M'_{i_0}$ belongs to ϕ_α (implying that ϕ_α is not null). To achieve this, it suffices to show that this monomial does not belong to any polynomial $(\alpha - Y_0 - \ldots - Y_{\kappa-2})^{e'_{i,\kappa-1}} M'_i$ with $i \neq i_0$.

Suppose that there exists $i_1 \neq i_0$ s.t. $Y_0^m M'_{i_0}$ belongs to $(\alpha - Y_0 - \ldots - Y_{\kappa-2})^{e'_{i_1,\kappa-1}} M'_{i_1}$. Clearly, $0 \notin E(M'_{i_0})$ implies that $0 \notin E(M'_{i_1})$ and $e'_{i_1,\kappa-1} \geq m$ (because the constraint $e_i = 0 \Rightarrow e'_i = 0$ implies that the exponent of Y_0 in M'_{i_1} is equal to 0). By definition of m, it follows that $e'_{i_1,\kappa-1} = m$ implying that $M'_{i_0} \neq M'_{i_1}$ (because $M_{i_0} = M_{i_1}$ otherwise). Thus, $Y_0^m M'_{i_0}$ does not belong to $(\alpha - Y_0 - \ldots - Y_{\kappa-2})^m M'_{i_1}$. This concludes the proof.

□

To simplify the analysis, we enhance the power of $\mathcal{A}$ by revealing S. If $\mathcal{A}$ can recover ϕ for a specific choice of S then it can do it for any choice of S. Thus, we can fix $S = \mathsf{Id}$ without loss of generality. The CPA attacker chooses GV (see Definition 4) and $x \in \mathbb{Z}_n^r$ and then it invokes the encryption oracle to get encryptions $\boldsymbol{c}_2, \dots, \boldsymbol{c}_r$ of $x_2, \dots, x_r$. For sake of simplicity (but without loss of generality), we assume that GV consists of recursively applying operators $\mathcal{Q}_i$ but not linear combinations.

Let ϕ be an arbitrary non-null polynomial of $\mathbb{Z}_n[X_1, \dots, X_{2\kappa t}]$ such that $\deg \phi < \kappa$ chosen by the CPA attacker. The polynomial $\phi \circ \mathsf{GV}_{\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}$ can be written as a non-null polynomial $\overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\dots,\boldsymbol{c}_r} \in \mathbb{Z}_n[X_1, \dots, X_{2\kappa}]$ defined by

$$\overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}(r_0, \dots, r_{\kappa-1}, x_0, \dots, x_{\kappa-1}) = \phi \circ \mathsf{GV}_{\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}(\boldsymbol{c}_1)$$

where $\boldsymbol{c}_1 = (r_0 x_0, r_0, \dots, r_{\kappa-1} x_{\kappa-1}, r_{\kappa-1})$.

By construction of the operator Add, each vector $\boldsymbol{v}$ output by GV is independent of $\boldsymbol{c}_1$ or satisfies $\boldsymbol{v} = \big(a_0 \cdot r_0^e(e \cdot x_0 + b_0), a_0 \cdot r_0^e, \dots, a_{\kappa-1} \cdot r_{\kappa-1}^e(e \cdot x_{\kappa-1} + b_{\kappa-1}), a_{\kappa-1} \cdot r_{\kappa-1}^e\big)$ where $a_i, b_i \in \mathbb{Z}_n$ only depends on $\boldsymbol{c}_2, \dots, \boldsymbol{c}_r$. Consequently, as $\deg \phi < \kappa$, each monomial $r_0^{e_0} \cdots r_{\kappa-1}^{e_{\kappa-1}} x_0^{e'_0} \cdots x_{\kappa-1}^{e'_{\kappa-1}}$ of $\overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}$ satisfies

- $\exists i \in \mathcal{K}$ s.t. $e_i = e'_i = 0$
- $e_i = 0 \Rightarrow e'_i = 0$.

By fixing $x_0 + \dots + x_{\kappa-1} = x$ (which is the value encrypted by $\boldsymbol{c}_1$) and by using the equality $x_{\kappa-1} = x - x_{\kappa-2} - \ldots - x_0$, $\overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}$ can be written as a polynomial $\overline{\phi \circ \mathsf{GV}}_{x,\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}$ of $\mathbb{Z}_n[X_1, \dots, X_{2\kappa-1}]$ satisfying

$$\overline{\phi \circ \mathsf{GV}}_{x,\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}(r_0, \dots, r_{\kappa-1}, x_0, \dots, x_{\kappa-2}) = \overline{\phi \circ \mathsf{GV}}_{\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}(r_0, \dots, r_{\kappa-1}, x_0, \dots, x_{\kappa-1})$$

By Lemma 9, this polynomial is not null.

Consequently, assuming Conjecture 1, $\overline{\phi \circ \mathsf{GV}}_{x,\boldsymbol{c}_2,\dots,\boldsymbol{c}_r}(r_1, \dots, r_\kappa, x_1, \dots, x_{\kappa-1}) = 0$ with negligible probability according to Lemma 4. Thus, assuming Conjecture 2, our scheme is IND-CPA secure.

□

F About the impossibility of deriving new operators $\mathcal{Q}$

The definition of GV (and thus Conjecture 1) would be irrelevant if new operators $\mathcal{Q}$ (for chosen families $\mathcal{P}$) could be polynomially derived from the public operators $\mathcal{Q}_1, \dots, \mathcal{Q}_\rho$. Let $\mathcal{Q}_1 \leftarrow \mathsf{QGen}(S, \mathcal{P}_1, 0, 0)$ and $\mathcal{Q}_2 \leftarrow \mathsf{QGen}(S, \mathcal{P}_2, 0, 0)$. Clearly, the operator $\mathcal{Q}_3 = \mathcal{Q}_1 + \mathcal{Q}_2$ is the operator output by $\mathsf{QGen}(S, \mathcal{P}_1 + \mathcal{P}_2, 0, 0)$. Thus, it is possible to build new relevant operators, e.g. $\mathcal{Q}_3$. However, as linear combinations are considered in GV, this new operator is not useful in the sense that the same vectors can be derived by procedures GV using or not this operator. In this section, we wonder whether one can derive new operators $\mathcal{Q}$ dealing with families of polynomials $\mathcal{P}$ which cannot be obtained by linear combinations of $\mathcal{P}_1, \dots, \mathcal{P}_\rho$. Corollary 2 ensures that this cannot be directly done by recovering S.

F.1 A discussion based on Lemma 6

Let S be three randomly chosen invertible matrices of $\mathbb{Z}_n^{2\kappa \times 2\kappa}$, let $\mathcal{P}_0, \dots, \mathcal{P}_r$ be $\rho + 1$ families of quadratic polynomials satisfying requirements of Definition 2 and let $\mathcal{Q}_i \leftarrow \mathsf{QGen}(S, \mathcal{P}_i, 0, 0)$. Moreover, we assume that $\mathcal{P}_0 \notin \mathsf{co}(\mathcal{P}_1, \dots \mathcal{P}_\rho)$.

In order to simplify the analysis (and the task of the attacker), let us assume that S^{-1} is replaced by the identity matrix in QGen, i.e.

$$(q_1, \dots, q_{2\kappa}) = (p_1 \circ z_0, p_2 \circ z_0, \dots, p_1 \circ z_{\kappa-1}, p_2 \circ z_{\kappa-1})$$

In this case, the monomial coefficients of the public operators $\mathcal{Q}_1, \ldots, \mathcal{Q}_\rho$ can be written as polynomials of F_2 (see Section 2.2) defined over the tuples $y_0, \ldots y_{\kappa-1}$ defined by

$$y_\ell = (s_{2\ell+1}, s_{2\ell+2})$$

By using the fact that $\mathcal{P}_0 \notin \mathsf{co}(\mathcal{P}_1, \ldots, \mathcal{P}_\rho)$, we easily show that the coefficients of $\mathcal{Q}_0$ cannot be written as linear combinations defined over the coefficients of $\mathcal{Q}_1, \ldots, \mathcal{Q}_\rho$. By Lemma 6, there does not exist polynomials p, q of degree smaller than κ such that $\alpha \cdot q(\mathcal{Q}_1, \ldots, \mathcal{Q}_\rho) - p(\mathcal{Q}_1, \ldots, \mathcal{Q}_\rho)$ is identically equal to 0. In other words, assuming Conjecture 1, a CPA attacker cannot recover small degree polynomial p, q ($\deg p, q < \kappa$) s.t.

$$\alpha = p(\mathcal{Q}_1, \ldots, \mathcal{Q}_\rho)/q(\mathcal{Q}_1, \ldots, \mathcal{Q}_\rho)$$

with non-negligible probability. While this is not sufficient to prove that $\mathcal{Q}_0$ cannot be recovered, this strongly enhances this idea.

F.2 An extension of Lemma 5

Let $(\mathcal{Q}_i)_{i=1,\ldots,\rho}$ be the operators defined in the previous section and let $\mathcal{Q} \leftarrow \mathsf{QGen}(S, \mathcal{P}, \sigma, \sigma')$ be an arbitrary operator such that $\sigma \neq 0$ and/or $\sigma' \neq 0$. In this section, we show that an attacker cannot recover $\mathcal{Q}$ only given $(\mathcal{Q}_i)_{i=1,\ldots,\rho}$ (and accesses to the encryption oracle). In particular, this will prove that a CPA attacker of the additive encryption scheme cannot derive new operators Add. To achieve this, we start by strengthening the definition of κ-symmetry (see Definition 1).

Definition 5. *A polynomial $s \in \mathbb{Z}_n[X_1, \ldots, X_{\kappa t}$ is $\overline{\kappa}$-symmetric if for any $y_0, \ldots, y_{\kappa-1} \in \mathbb{Z}_n^t$ and for any permutation σ of $\mathcal{K}$, $s(y_0, \ldots, y_{\kappa-1}) = s(y_{\sigma(0)}, \ldots, y_{\sigma(\kappa-1)})$.*

Then, instead of considering non κ-symmetric monomials π, we will consider non $\overline{\kappa}$-symmetric polynomials.

PROBLEM 2. *Let $I_F \subseteq \{1, \ldots, t\}$, let n be a randomly chosen RSA modulus and let $(s_1, \ldots, s_m, \pi) \leftarrow \mathcal{A}_S(n)$ be public κt-variate polynomials satisfying,*

- *$s_1, \ldots, s_m$ are $\overline{\kappa}$-symmetric*
- *π is a non $\overline{\kappa}$-symmetric polynomial defined[14] over $\{y_{\ell i} | (\ell, i) \in \mathcal{K} \times I_F\}$.*

Let $(y_0, \ldots, y_{\kappa-1})$ i.d.d. drawn according to the uniform distribution over $\mathbb{Z}_n^t$ s.t. for each $i \notin I_F$, $y_{0i} + \ldots + y_{\kappa-1,i} = x_i$ where $x_i \in \mathbb{Z}_n$ are public.

The challenge is to recover $\pi(y_0, \ldots, y_{\kappa-1})$ given only[15] $s_1(y_0, \ldots, y_{\kappa-1}), \ldots, s_m(y_0, \ldots, y_{\kappa-1})$.

Lemma 10. *Problem 2 is difficult assuming Conjecture 1.*

Proof. (*Sketch.*) The proof exactly follows the proof of Lemma 5 given in Appendix C.1. Nevertheless, Conjecture 1 is required to ensure that π and π_{σ^*} are equal with negligible probability. □

It suffices then to notice that each value known by the CPA attacker is $\overline{\kappa}$-symmetric relatively to the tuples y_ℓ defined in the proof of Proposition 3 while each value of $\mathcal{Q}$ is not $\overline{\kappa}$-symmetric (only κ-symmetric) and thus cannot be recovered according to Lemma 10.

[14] $\pi(y_0, \ldots, y_{\kappa-1}) = \prod_{(\ell,i) \in \mathcal{K} \times I} y_{\ell i}^{e_{\ell i}}$ with $e_{\ell i} = 0$ when $i \notin I_F$.

[15] and an efficient representation of $\pi, s_1, \ldots, s_m$.

F.3 Randomizing the operators $\mathcal{Q}$

The key idea of this section is to add rows to S which are not useful for encryptions. For concreteness, S is a randomly chosen matrix of $\mathbb{Z}_n^{2\kappa+\delta}$ and an encryption $\boldsymbol{c}$ of x is

$$\boldsymbol{c} = S^{-1}\,(r_1 x_1, r_1, \ldots, r_\kappa x_\kappa, r_\kappa, 0, \ldots, 0)$$

Let E be the set[16] of the linear combinations over the vectors $s_{2\kappa+1}, \ldots, s_{2\kappa+\delta}$. 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 $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{Z}_n^{2\kappa+\delta}$ are arbitrary vectors. By construction, for any $r \in R$ and any public encryptions $\boldsymbol{c}, \boldsymbol{c}'$,

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

Let $\mathcal{Q} = (q_1, \ldots, q_{2\kappa+\delta}) \leftarrow \mathsf{QGen}(S, \mathcal{P}, \sigma, \sigma')$ and $(r_1, \ldots, r_{2\kappa+\delta})$ be randomly chosen in R. By construction, it is ensured that the operator $\mathcal{Q}^{\mathsf{rand}} = (q_1 + r_1, \ldots, q_{2\kappa+\delta} + r_{2\kappa+\delta})$ satisfies for any encryptions $\boldsymbol{c}, \boldsymbol{c}'$

$$\mathcal{Q}^{\mathsf{rand}}(\boldsymbol{c}, \boldsymbol{c}') = \mathcal{Q}(\boldsymbol{c}, \boldsymbol{c}')$$

[16] E can be recovered by the Attacker.