An efficient FHE based on the hardness of solving systems of non-linear multivariate equations

Gérald Gavin

Laboratory ERIC

University of Lyon, France

Email: gavin@univ-lyon1.fr

Abstract. We propose a general framework to develop fully homomorphic encryption schemes (FHE) without using the Gentry's technique. Initially, a private-key cryptosystem is built over $\mathbb{Z}_n$ (n being an RSA modulus). An encryption of $x \in \mathbb{Z}_n$ is a randomly chosen vector e such that $\Phi(e) = x$ where Φ is a secret multivariate polynomial. This private-key cryptosystem is not homomorphic in the sense that the vector sum is not a homomorphic operator. Non-linear homomorphic operators are then developed. The security relies on the difficulty of solving systems of non-linear equations (which is a $\mathcal{NP}$-complete problem). While the security of our scheme has not been reduced to a provably hard instance of this problem, security is globally investigated.

1 Introduction

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 [3]. More recently, further fully homomorphic schemes were presented [8],[9],[1],[4] 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 [6], [5].

In this paper, we propose a general framework to develop FHE without using the Gentry's technique. We first propose a very simple private-key cryptosystem where a ciphertext is a vector e whose the components are in $\mathbb{Z}_n$, n being an RSA modulus chosen at random. Given a secret multivariate polynomial Φ, an encryption of $x \in \mathbb{Z}_n$ is a vector e chosen at random such that[1] $\Phi(e) = x$. In order to resist to a CPA attacker, the number of monomials of Φ 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 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. It is a *sine qua none* condition to overcome Gentry's machinery. Indeed, as a ciphertext e is a vector, it is always possible to write it as a linear combination of other known ciphertexts. Thus, if the vector sum is a homomorphic operator, the cryptosystem is not secure at all. So, in order to use the vector sum as a homomorphic operator, noise should be injected in encryptions as it is done in all existing FHE. To overcome this, we propose to develop *ad-hoc* non-linear homomorphic operators. The public key contains these operators and public encryptions while the secret key contains the multivariate polynomial Φ.

Our contribution. A very simple additively homomorphic cryptosystem is developed in Section 3. Its performance is low compared to existing additively homomorphic cryptosystems (El Gamal [2], Paillier [7], etc...). Even if improvements leading to an efficient scheme are proposed, the main objective of this

[1] or $\Phi(e) = f(x)$ where $f : E \subset \mathbb{Z}_n \to f(E)$ is a one-to-one function such that f^{-1} is efficient.

section is to highlight the underlying ideas involved in the construction and in the security analysis of our FHE.

In this paper, the security of cryptosystems is related to the difficulty of solving nonlinear equations in $\mathbb{Z}_n$. Unfortunately, we did not reduce the whole security of these cryptosystems to a provably hard instance of this problem. However, several partial security results (see Proposition 3 and Proposition 9) are proven and extensively discussed in order to globally investigate the security of our schemes. These results provide a formal framework for the cryptanalysis by restricting the set of possible attacks. In our opinion, the main interest of this paper is to provide new directions and new material for the development of efficient FHE.

2 Security assumptions

Let $n = pq$ be a η-bit RSA-modulus and κ, t be positive integers. Throughout this paper, all the computations are done in $\mathbb{Z}_n$. Let y_1, y_2 be randomly chosen in $\mathbb{Z}_n$. It is well-known that recovering[2] y_1 only given $S = y_1 + y_2$ or $P = y_1 y_2$ is difficult assuming the hardness of factorization. In this section, we propose to extend this.

Definition 1. *A multivariate polynomial* $s : \left(\mathbb{Z}_n^t\right)^\kappa \to \mathbb{Z}_n$ *is said to be:*

- *efficiently valuable if it can be computed in polynomial-time (with respect to* η*) without knowing the factorization of* n*.*
- κ*-symmetric if for any* $y_1, ..., y_\kappa \in \mathbb{Z}_n^t$ *and for any permutation* σ *of* $\{1, ..., \kappa\}$*,*

$$s(y_1, ..., y_\kappa) = s(y_{\sigma(1)}, ..., y_{\sigma(\kappa)})$$

Let π be a non κ-symmetric product of values y_{li}. The two following problems consist of recovering π only given κ-symmetric functions $s_j(y_1, ..., y_\kappa)$ where the tuples y_l are chosen at random under symmetric additive and multiplicative constraints. These two problems only differ with respect to their constraints.

Problem 1. Let $I_F \subset I$ *be a non-empty set,* $(a_i)_{i \in I \setminus I_F}$ *be an arbitrary family of public values belonging to* $\mathbb{Z}_n^*$ *and* $(y_l)_{l=1,...,\kappa} = (y_{l1}, ..., y_{lt})$ *be* κ *tuples of* $\mathbb{Z}_n^t$ *chosen at random such that for all* $i \in I \setminus I_F$*,*

$$\prod_{l=1}^{\kappa} y_{li} = a_i$$

Let $\pi(y_1, ..., y_\kappa)$ *be an arbitrary efficiently valuable non* κ*-symmetric product* π *of values belonging to* $\{y_{li} \mid l = 1, ..., \kappa \ ; i \in I_F\}$*.*

Problem: recovering $\pi(y_1, ..., y_\kappa)$ only given $s_1(y_1, ..., y_\kappa), ..., s_m(y_1, ..., y_\kappa)$ where $s_1, ..., s_m$ are public efficiently valuable κ-symmetric polynomials (m polynomial in η).

Problem 2. Let $I_1^\times, ..., I_r^\times$ *and* $I_1^+, ..., I_{r'}^+$ *be* $r + r'$ *public disjoint subsets of* $I = \{1, ..., t\}$ *such that*

$$I_F = I \setminus \left(\bigcup_{j=1}^{r} I_j^\times \cup \bigcup_{j=1}^{r'} I_j^+ \right) \neq \emptyset$$

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

Let κ tuples $y_l = (y_{l1}, ..., y_{lt}) \in \mathbb{Z}_n^t$ chosen at random such that

$$\forall j = 1, ..., r \;\; \forall l = 1, ..., \kappa \quad \prod_{i \in I_j^\times} y_{li} = a_j$$
$$\forall j = 1, ..., r' \;\; \forall l = 1, ..., \kappa \quad \sum_{i \in I_j^+} y_{li} = a'_j$$

where $(a_j)_{j=1,...,r}$ and $(a'_j)_{j=1,...,r'}$ are arbitrary public values of respectively $\mathbb{Z}_n^*$ and $\mathbb{Z}_n$

Let $\pi(y_1, ..., y_\kappa)$ be an arbitrary efficiently valuable non κ-symmetric product π of values belonging to $\{y_{li} \mid l = 1, ..., \kappa \; ; \; i \in I_F\}$.

Problem: recovering $\pi(y_1, ..., y_\kappa)$ only given $s_1(y_1, ..., y_\kappa), ..., s_m(y_1, ..., y_\kappa)$ where $s_1, ..., s_m$ are public efficiently valuable κ-symmetric polynomials (m polynomial in η).

Proposition 1. *Problem 1 and Problem 2 are difficult assuming the hardness of factorization.*

Proof. The proof looks like the famous Rabin's proof showing that extracting square roots is equivalent to factoring. See Appendix A for the detail of the proof.
□

Remark 1. Proposition 1 can be generalized by considering efficiently valuable non κ-symmetric polynomials π (instead of products) in Problem 1 and Problem 2 ensuring that π is not trivial, i.e. there exists a permutation σ of $\{1, ..., \kappa\}$ such that the probability to get $\pi(y_1, ..., y_\kappa) = \pi(y_{\sigma(1)}, ..., y_{\sigma(\kappa)})$ is negligible.

3 An additive homomorphic cryptosystem

Let $\delta \in \mathbb{N}^*$ and n be an RSA modulus. All the computations of this section will be done in $\mathbb{Z}_n$.

- *The set of all square m-by-m matrices over $\mathbb{Z}_n$ is denoted by $\mathbb{Z}_n^{m \times m}$.*
- *Throughout this paper, a vector $\overrightarrow{w} = \begin{pmatrix} w_1 \\ ... \\ w_t \end{pmatrix}$ can be also denoted by w or $(w_1, ..., w_t)$.*
- *Given two vectors w and w', the inner product of these vectors is denoted by ww'.*
- *The number of monomials of degree d defined over v variables is equal to $\begin{pmatrix} d+v-1 \\ v \end{pmatrix}$*

3.1 A basic private-key cryptosystem

We first define a very simple private-key cryptosystem where the plaintext space is $E = \{0, ..., M\}$. Let $S \in \mathbb{Z}_n^{\delta \times \delta}$ be a secret invertible matrix chosen at random and g be an arbitrarily element of $\mathbb{Z}_n^*$ of order larger than M. Basically, to encrypt x, it suffices to randomly choose a vector $r = (r_1, ..., r_\delta)$ such that $r_1 ... r_\delta = g^x$ and to hide it with S^{-1}, i.e. $e = S^{-1}r$. To decrypt e, it suffices to compute $d = Se$ and then to compute the discrete logarithm of the product of the components of d, i.e. $x = \mathsf{DL}_g(d_1 ... d_\delta)$. Note that the plaintext space E should be "small" because there does not exist efficient algorithm DL. At this step, the cryptosystem is not homomorphic in the sense that the (vector) sum is not a homomorphic operator.

Definition 2. *Let λ be a security parameter and $E = \{0, , ..., M\}$ be a polynomial-size set of integers E (E will be the plaintext set). The functions KeyGen0, Encrypt0, Decrypt0 are defined as follows:*

1. *KeyGen0*(λ). *Let η, δ be positive integers indexed by λ. Let n be a public η-bit RSA modulus chosen at random and $g \leftarrow \mathbb{Z}_n^*$ such that its order is larger than M. Let S be an invertible matrix of $\mathbb{Z}_n^{\delta \times \delta}$ chosen at random. The i^{th} row of S is denoted by s_i and $\Phi_S : \mathbb{Z}_n^\delta \to \mathbb{Z}_n$ denotes the δ-degree multivariate polynomial defined by $\Phi_S(w) = \prod_{i \in \{1,...,\delta\}} s_i w$.*
$$K = \{g, S\}$$
2. *Encrypt0*$(K, x \in E)$. *Choose at random a vector $r = (r_1, ..., r_\delta)$ such that $\prod_{i=1}^{\delta} r_i = g^x$ and output*
$$e = S^{-1} r$$
3. *Decrypt0*$(K, e \in \mathbb{Z}_n^\delta)$. *Output*
$$x = \mathsf{DL}_g\left(\Phi_S(e)\right)$$

3.2 Operator $\mathcal{Q}_S$

Let S be the invertible matrix of $\mathbb{Z}_n^{\delta \times \delta}$ output by $\mathsf{KeyGen0}(\lambda)$. The function $\mathcal{Q}_S : \mathbb{Z}_n^\delta \times \mathbb{Z}_n^\delta \to \mathbb{Z}_n^\delta$ is defined by
$$\mathcal{Q}_S(w', w'') \stackrel{\mathsf{def}}{=} \begin{pmatrix} q_1(w', w'') \\ \dots \\ q_\delta(w', w'') \end{pmatrix} = S^{-1} \begin{pmatrix} s_1 w' \times s_1 w'' \\ \dots \\ s_\delta w' \times s_\delta w'' \end{pmatrix}$$
The function QGen inputs S and outputs the expanded representation of the polynomials $q_1, ..., q_\delta$, i.e. all the monomial coefficients of the polynomials q_i.

An implementation of this operator for $\delta = 2$ is presented in Appendix J. Concretely, by denoting $a = Sw'$, $b = Sw''$ and $c = S\mathcal{Q}_S(w', w'')$, we have $c_i = a_i b_i$ for all $i = 1, ..., \delta$ (see Figure 1).

$$\mathcal{Q}_S\left(S^{-1} \begin{pmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \end{pmatrix}, S^{-1} \begin{pmatrix} w''_1 \\ w''_2 \\ w''_3 \\ w''_4 \\ w''_5 \\ w''_6 \end{pmatrix} \right) = S^{-1} \begin{pmatrix} w'_1 w''_1 \\ w'_2 w''_2 \\ w'_3 w''_3 \\ w'_4 w''_4 \\ w'_5 w''_5 \\ w'_6 w''_6 \end{pmatrix}$$

Fig. 1. Illustration of the operator $\mathcal{Q}_S$ for $\delta = 6$. Clearly, $\mathcal{Q}_S$ is an additively homomorphic operator of the private-key cryptosystem.

Proposition 2. *The computation of $\mathcal{Q}_S = (q_1, ..., q_\delta) \leftarrow$ QGen(S) requires $O(\delta^4)$ modular multiplications and the computation of $w \leftarrow \mathcal{Q}_S(w', w'')$ requires $O(\delta^3)$ modular multiplications. Each monomial coefficient of $\mathcal{Q}_S$ is an efficiently valuable δ-symmetric polynomial defined over the δ tuples $y_i = (s_{ij})_{j=1,...,\delta}$.*

Proof. (Sketch) To establish complexity results, it suffices to notice that the number of monomials of the polynomials p_i and q_i is $O(\delta^2)$. Each monomial coefficient can be written as a ratio of two polynomials defined over the tuples y_i. It is well-known that any function defined over a field (here $\mathbb{Z}_p$ and $\mathbb{Z}_q$) can be written as a polynomial. It follows that each monomial coefficient can be written as a polynomial defined over the tuples y_i. By noticing that the computation of the polynomials q_i does not require the knowledge of the factorization of n, each monomial coefficient is efficiently valuable.

Let σ be an arbitrary permutation of $\{1, ..., \delta\}$. Let T be the matrix such that its i^{th} row is equal to the $\sigma(i)^{th}$ row of S, i.e. $t_i = s_{\sigma(i)}$. It implies that the columns of T^{-1} are a σ-permutation of the columns of S^{-1}, i.e. the j^{th} column of T^{-1} is equal to the $\sigma(j)^{th}$ column of S^{-1}. It follows that for all $w \in \mathbb{Z}_n^\delta$, $S^{-1}w = T^{-1}\sigma(w)$ ensuring that $\mathsf{QGen}(S) = \mathsf{QGen}(T)$. This proves that each monomial coefficient is a κ-symmetric function defined over the tuples y_i.

□

Corollary 1. *According to Proposition 1, it is not possible to recover any non δ-symmetric product of values s_{ij} only given $\mathcal{Q}_S$ assuming the hardness of factorization.*

3.3 The additive homomorphic scheme

To get an additively homomorphic public-key cryptosystem, it suffices to publish $m = \Theta(\lambda)$ encryptions e_v of public values $x_v \in E$ and the operator $\mathcal{Q}_S \leftarrow \mathsf{QGen}(S)$. For instance $x_v = 2^v$ for all $v = 1, ..., \lfloor \log_2 M \rfloor$ and $x_v = 0$ for all $v = \lfloor \log_2 M \rfloor + 1, ..., m$

Definition 3. *Let λ be security parameter.*

- *KeyGen*(λ). *Let $K = \{S, g\} \leftarrow$ KeyGen0(λ), $(x_v)_{v=1,...,m}$ be m values*[3] *of E, $e_v \leftarrow$ Encrypt0(K, x_v) and $\mathcal{Q}_S \leftarrow$ QGen(S)*

$$sk = \{\Phi_S\} \ ; \ pk = \{\mathcal{Q}_S, (x_v, e_v)_{v=1,...,m}\}$$

- *Operator* $\oplus$. *Given two encryptions e and e'*

$$e \oplus e' = \mathcal{Q}_S(e, e')$$

- *Encrypt*$(pk, x \in E)$. *Choose a subset of $m/2$ public encryptions $(e_{v_i})_{i=1,...,m/2}$ at random such that $x = x_{v_1} + ... + x_{v_{m/2}}$ and output*

$$e = \bigoplus_{i=1}^{m/2} e_{v_i}$$

- *Decrypt*(sk, e). *Exactly follows* Decrypt0.

It is straightforward to check correctness of this scheme. A toy implementation of this scheme is presented in Appendix J.

3.4 Security analysis.

Given $\gamma \in \mathbb{N}^*$, SP^γ refers to the set of multi-variate polynomials $\phi : \left(\mathbb{Z}_n^\delta\right)^r \rightarrow \mathbb{Z}_n$ defined by

$$\phi(w_1, ..., w_r) = \prod_{t=1}^{\gamma} s_{i_t} w_{v_t}$$

where $r \in \mathbb{N}^*$, $i_t \in \{1, ..., \delta\}$ and $v_t \in \{1, ..., r\}$. A representation R_ϕ of ϕ is said to be effective if its storage is polynomial and if R_ϕ allows to evaluate ϕ in polynomial time. For instance, provided $\delta = \Theta(\lambda)$, the factored representation of Φ_S is effective while its expanded representation is not (the number of monomials is exponential).

[3] $x_v = 2^v$ for all $v = 1, ..., \lfloor \log_2 M \rfloor$ and $x_v = 0$ otherwise.

Proposition 3. *Let λ be a security parameter, $(pk, sk) \leftarrow \mathsf{KeyGen}(\lambda)$ and $\gamma \in \mathbb{N}^*$ such that γ is a not multiple of δ ($|\gamma|$ polynomial in λ). Let $\phi \in \mathsf{SP}^\gamma$ and R_ϕ be an effective representation of ϕ. By assuming the hardness of factorization, recovering R_ϕ only given pk is difficult.*

Proof. Let us denote by $(r_{vi})_{(v,i)\in\{1,...,m\}\times\{1,...,\delta\}}$ the random values chosen in the public encryptions $(e_v)_{v=1,...,m}$, i.e. $(r_{v1}, ..., r_{v\delta}) = Se_v$. Let $y_1^{sk}, ..., y_\delta^{sk}$ be the δ (secret) tuples defined by

$$y_i^{sk} = (s_i, r_{vi})_{v=1,...,m}$$

These tuples y_i^{sk} are generated according to a probability distribution statistically indistinguishable from the probability distribution considered in Problem 1 (by choosing the coefficient of S at random, the probability that S is not invertible is negligible) where the values s_{ij} are not involved in multiplicative constraints and $\prod_{i=1}^{\delta} r_{vi} = g^{x_v}$. Moreover, each public value of pk is an efficiently valuable δ-symmetric polynomial defined over the tuples $(y_1^{sk}, ..., y_\delta^{sk})$ (see Proposition 2 for the monomial coefficients of $\mathcal{Q}_S$ and it is straightforward to check it for each component of e_v by arguing similarly to the proof of proposition 2).

Consequently, according to Proposition 1, it is not possible to polynomially recover any non δ-symmetric product π of values s_{ij} assuming the hardness of the factorization.

Let ϕ be an element of SP^γ, i.e. $\phi(w_1, ..., w_r) = \prod_{t=1}^{\gamma} s_{i_t} w_{v_t}$. Let $w_1^* = ... = w_r^* = (1, 0, 0, ...)$ and $\pi = \phi(w_1^*, ..., w_r^*)$. Because γ is not a multiple of δ, π is a non δ-symmetric (efficiently valuable) product of values of $\{s_{i1} | i = 1, ..., \delta\}$. R_ϕ allows to efficiently compute π. Thus, according to Proposition 1, π cannot be recovered implying that R_ϕ cannot be recovered.

□

As $\Phi_S \in \mathsf{SP}^\delta$, this result does not prove that Φ_S cannot be recovered. Worse, it is easy to see that Φ_S can be easily recovered by solving a linear system[4] provided $\delta = O(1)$. However, provided $\delta = \Theta(\lambda)$, this attack does not work anymore because the number of monomials of Φ_S becomes exponential, i.e. $\Omega(4^\delta)$. Besides, Proposition 3 implies that it is not possible to recover any factored form of Φ_S. It implies that it is difficult to recover the expanded representation or any effective factored representation of Φ_S.

But it may be possible to polynomially recover other effective representations R_{Φ_S} of Φ_S, e.g. semi-factored forms of Φ_S. Proposition 3 can be generalized by showing that it is difficult to recover any non δ-symmetric values defined over the tuples y_i^{sk} (by extending Proposition 1 as explained by Remark 1). Consequently, to be polynomially recovered and evaluated, Φ_S should be written with a δ-symmetric effective representation[5] R_{Φ_S}, i.e. R_{Φ_S} should be expressed by a polynomial number of δ-symmetric values defined over the tuples y_i^{sk}. We conjecture that such effective δ-symmetric representations do not exist (see Appendix G for a toy example highlighting this). By extending this analysis to any $\phi \in \mathsf{SP}^{t\delta}$, we propose the following conjecture.

Conjecture 1. Assume that $\delta = \Theta(\lambda)$ and let $\phi \in \bigcup_{\gamma>0} \mathsf{SP}^\gamma$. By assuming the hardness of factorization, recovering any effective representation of ϕ is difficult only given pk.

Unfortunately, Conjecture 1 is not sufficient to prove semantic security while we do see how semantic security could be broken without the knowledge of polynomials of SP. Roughly speaking, this situation looks like to RSA security analysis where it is shown that recovering the decryption polynomial is difficult assuming the hardness of factorization while the security (*one-wayness*) is not formally reduced to this assumption.

---

[4] $\Phi_S(e_v) = x_v$ (for a number of encryptions e_v larger than the number of monomials of Φ_S) where the variables are the monomial coefficients of Φ_S

[5] The expanded representation of Φ_S is δ-symmetric but ineffective and conversely, the factored representation of Φ_S is effective but not δ-symmetric.

A weak version of Proposition 3 is proposed in Appendix F. In this proof, $\mathcal{Q}_S$ is built only given a randomly chosen δ-degree polynomial p having δ^2 distinct roots over $\mathbb{Z}_n$. It is shown that the rows of S are the eigenvectors of a matrix M which can be directly derived from $\mathcal{Q}_S$. Thus, it is no more difficult to recover S given $\mathcal{Q}_S$ when knowing the factorization of n. In Appendix K, we propose ways to randomize the operator $\mathcal{Q}_S$. An interesting question arising in this setting consists of wondering whether n could be chosen as a large/small prime. This would lead to a scheme (very) competitive with respect to other existing additively homomorphic schemes.

4 A basic private-key cryptosystem

Let $\delta \in \mathbb{N}^*$ and n be an RSA modulus. In the following of the paper, all the computations will be done in $\mathbb{Z}_n$.

- *A vector $\textsc{b}$ is said to be basic if $\textsc{b}$ is a δ-vector, i.e. $(b_1, ..., b_\delta) \in \mathbb{Z}_n^\delta$ and if*

$$\prod_{i=1}^{\delta} b_i = 1$$

  Throughout this paper, basic vectors will be denoted with (small) capital letters.
- *Given a basic vector $\textsc{b}$ and $a \in \mathbb{Z}_n$, $\textsc{b}a$ denotes the δ-vector $(b_1a, b_2, ..., b_\delta)$.*
- *Let $w_1, ..., w_t$ be t vectors of size m, $(w_1, ..., w_t)$ denotes the concatenation of these vectors, i.e. $(w_1, ..., w_t) = (w_{11}, ..., w_{1m}, ..., w_{t1}, ..., w_{tm})$.*
- *Given a vector w and a matrix S, $|w|_S = Sw$. Note that $|w|_S$ could be denoted by $|w|$ when S is implicitly known.*

First, we define a private-key cryptosystem where the plaintext space is $\mathbb{Z}_n$ and where the secret key contains ϑ randomly chosen invertible matrices S_z of $\mathbb{Z}_n^{\kappa\tau\delta \times \kappa\tau\delta}$. For $\kappa = \tau = 1$, a valid encryption e of x is composed of ϑ vectors $c_1, ..., c_\vartheta$ defined by

$$c_z = S_z^{-1} \ (\textsc{b}_z x_z)$$

where $\textsc{b}_z$ are random basic vectors and x_z random values satisfying $x_1 + ... + x_\vartheta = x$. The decryption consists of evaluating a δ-degree multivariate polynomial Φ, i.e. $\Phi(e) = x$. This polynomial can be written as a sum of ϑ polynomials, each one being factorizable as a product of δ linear functions. The role of the parameter ϑ will be explained in Section 8. We let the reader see why the scheme cannot be semantically secure with $\vartheta = 1$ (an attacker could easily decide if an encryption encrypts 0 or not). The parameter τ is not indexed by the security parameter λ. It is introduced in order to provide randomness useful for the construction of homomorphic operators. In Section 6, we propose a construction for $\tau = 3$. Contrarily to the previous cryptosystem, the FHE developed in next sections is not *naturally symmetric*. To overcome this, the parameter κ is artificially introduced in order to exploit Proposition 1 in the security analysis.

Definition 4. *Let λ be a security parameter and $\tau \in \mathbb{N}^*$. The functions KeyGen1, Encrypt1, Decrypt1 are defined as follows:*

1. *KeyGen1(λ, τ). Let $\eta, \kappa, \delta, \vartheta$ be positive integers indexed by λ. Let n be a η-bit RSA modulus chosen at random and $(S_z)_{z=1,...,\vartheta}$ be ϑ invertible matrices of $\mathbb{Z}_n^{\kappa\tau\delta \times \kappa\tau\delta}$ chosen at random. The i^{th} row of S_z is denoted by s_{zi}. For any $l \in \{1, ..., \kappa\}$, $\Phi_l : \left(\mathbb{Z}_n^{\kappa\tau\delta}\right)^\theta \to \mathbb{Z}_n$ denotes the δ-degree multivariate polynomial defined by:*

$$\Phi_l(w_1, ..., w_\vartheta) = \sum_{z=1}^{\vartheta} \prod_{i \in I_l} s_{zi} w_z$$

with $I_l = \{(l-1)\tau\delta+1, ..., (l-1)\tau\delta+\delta\}$

$$K = \{(S_z)_{z=1,...,\vartheta}\}$$

2. *Encrypt1*$(K, x \in \mathbb{Z}_n)$*. Choose at random* $\vartheta\kappa\tau$ *basic vectors* $(\mathrm{B}_{zlt})_{(z,l,t)\in\{1,...,\vartheta\}\times\{1,...,\kappa\}\times\{1,...,\tau\}}$ *and* $\vartheta\kappa$ *values* $(x_{zl})_{(z,l)\in\{1,...,\vartheta\}\times\{1,...,\kappa\}}$ *belonging to* $\mathbb{Z}_n$ *such that for all* $l = 1, ..., \kappa$

$$\sum_{z=1}^{\vartheta} x_{zl} = x$$

Let $(c_z)_{z=1,...,\vartheta}$ *be the* ϑ *vectors defined by:*

$$S_z c_z \ (\stackrel{def}{=} |c_z|_{S_z}) = \left(\boxed{\mathrm{B}_{z,1,1}x_{z,1}, \mathrm{B}_{z,1,2}, ..., \mathrm{B}_{z,1,\tau}}, \boxed{\mathrm{B}_{z,2,1}x_{z,2}, \mathrm{B}_{z,2,2}..., \mathrm{B}_{z,2,\tau}}, ..., \boxed{\mathrm{B}_{z,\kappa,1}x_{z,\kappa}, \mathrm{B}_{z,\kappa,2}, ..., \mathrm{B}_{z,\kappa,\tau}}\right)$$

Output

$$e = (c_1, ..., c_\vartheta)$$

3. *Decrypt1*$(K, e \in \left(\mathbb{Z}_n^{\kappa\tau\delta}\right)^{\vartheta}$*. Choose* $l \in \{1, ..., \kappa\}$ *arbitrarily and output*

$$x = \Phi_l(e)$$

Proposition 4. *Let* $e \leftarrow$ *Encrypt1*(K, x) *and* $(\mathrm{B}_{zlt})_{(z,l,t)\in\{1,...,\vartheta\}\times\{1,...\kappa\}\times\{1,...,\tau\}}$ *be the random basic vectors and* $(x_{zl})_{(z,l)\in\{1,...,\vartheta\}\times\{1,...\kappa\}}$ *be the random values used by Encrypt1 to generate* e*. Let* $(y_l)_{l=1,...,\kappa}$ *be* κ *tuples defined by*

$$y_l = (s_{zi}, \mathrm{B}_{zlt}, x_{zl})_{(z,t,i)\in\{1,...,\vartheta\}\times\{1,...,\tau\}\times\{(l-1)\tau\delta+1,...l\tau\delta\}}$$

Each component of e *is an efficiently valuable* κ*-symmetric polynomial defined over the tuples* $(y_1, ..., y_\kappa)$*.*

Proof. See Appendix B.

A short informal security analysis. Let $(e_v)_{v=1,...,m}$ be m encryptions of $(x_v)_{v=1,...,m}$ known by the CPA attacker. Let us consider the linear system $\Phi_1(e_v) = x_v$ for all $v = 1, ..., m$ where the variables are the monomial coefficients of Φ_1. As $\Phi_{l=2,...,\kappa}$ are also solutions of this system, its resolution provides a linear combination

$$\Phi = \alpha_1\Phi_1 + ... + \alpha_\kappa\Phi_\kappa$$

with $\alpha_1 + ... + \alpha_\kappa = 1$ which breaks semantic security. However, provided $\delta = \Theta(\lambda)$, Φ has an exponential number of monomials making this brute force attack fail.

Nevertheless, one could hope to recover a *compact representation* of Φ, e.g. a factored or semi-factored representation. However, to achieve this, one should solve a nonlinear multivariate equation system which is a difficult problem in general. This first analysis suggests that Φ cannot be recovered by a CPA attacker. Proposition 1 will be used in the security analysis of our FHE to formalize this analysis.

5 κ-symmetric operators $\mathcal{Q}$

This section can be seen as a generalization of Section 3.2. Let $m \in \mathbb{N}^*$ and S be an arbitrary invertible matrix of $\mathbb{Z}_n^{m\times m}$ where the i^{th} row is denoted by s_i. Let $p_i : \mathbb{Z}_n^m \times \mathbb{Z}_n^m \rightarrow \mathbb{Z}_n$ be m arbitrary polynomials. The function $\mathcal{Q}_{S,p_1,...,p_m} : \mathbb{Z}_n^m \times \mathbb{Z}_n^m \rightarrow \mathbb{Z}_n^m$ is defined by

$$\mathcal{Q}_{S,p_1,...,p_m}(w', w'') \stackrel{\text{def}}{=} \begin{pmatrix} q_1(w', w'') \\ \dots \\ q_m(w', w'') \end{pmatrix} = S^{-1} \begin{pmatrix} p_1(w', w'') \\ \dots \\ p_m(w', w'') \end{pmatrix}$$

The multivariate polynomials $q_1, ..., q_m$ are linear combinations of the polynomials $p_1, ..., p_m$. The function QGen inputs S and the polynomials $p_1, ..., p_m$ and outputs the expanded representation of the polynomials $q_1, ..., q_m$, i.e. all the monomial coefficients of the polynomials q_i.

Proposition 5. *Let $p_i : \mathbb{Z}_n^m \times \mathbb{Z}_n^m \rightarrow \mathbb{Z}_n$ be m arbitrary 2-degree polynomials. The computation of $\mathcal{Q}_{S,p_1,...,p_m} \leftarrow$ QGen$(S, p_1, ..., p_m)$ requires $O(m^4)$ modular multiplications and the computation of $w \leftarrow \mathcal{Q}_{S,p_1,...,p_m}(w', w'')$ requires $O(m^3)$ modular multiplications.*

Proof. (Sketch.) The number of monomials of each 2-degree polynomial p_i is $O(m^2)$.
□

Definition 5. *(κ-**symmetric operators** $\mathcal{Q}$). Let $\kappa \in \mathbb{N}^*$. Let S, S', S'' be three invertible matrices of $\mathbb{Z}_n^{\kappa m\times\kappa m}$. The i^{th} row of S, S', S'' is respectively denoted by s_i, s'_i, s''_i. Let $(p_i)_{i=1,...,\kappa m} : \mathbb{Z}_n^{\kappa m} \times \mathbb{Z}_n^{\kappa m} \rightarrow \mathbb{Z}_n^{\kappa m}$ be κm 2-degree multivariate polynomials defined by*

$$p_i(w', w'') = \sum_{j=1}^{\alpha_i} s'_{u'_{ij}} w' \times s''_{u''_{ij}} w''$$

where $\alpha_i \in \mathbb{N}^*$, $u'_{ij}, u''_{ij} \in \{1, ..., \kappa m\}$. The operator $\mathcal{Q}_{S,p_1,...,p_{\kappa m}}$ (also denoted by $\mathcal{Q}_{S\leftarrow(S',S''),p_1,...,p_{\kappa m}}$) is said to be κ-symmetric with respect to S, S', S'' if for all $i \in \{1, ..., m\}$, $j \in \{1, ..., \alpha_i\}$ and $l \in \{1, ...\kappa\}$

$$\begin{cases} \alpha_{i+lm} = \alpha_i \\ u'_{i+lm,j} = u'_{ij} + lm \\ u''_{i+lm,j} = u''_{ij} + lm \end{cases}$$

Let $\mathcal{Q}_{S\leftarrow(S',S''),p_1,...,p_{\kappa m}}$ be a κ-symmetric operator and $w \leftarrow \mathcal{Q}_{S\leftarrow(S',S''),p_1,...,p_{\kappa m}}(w', w'')$. Each component of $|w|_S$ is a 2-degree polynomial function of the components of $|w'|_{S'}$ and $|w''|_{S''}$. Higher degree polynomials p_i could be considered but it would lead to very costly operators $\mathcal{Q}$: the running time of such operators $\mathcal{Q}$ is exponential with the degree of the polynomials p_i. Roughly speaking, information hidden in $|w'|_{S'}$ and $|w''|_{S''}$ can be manipulated by operators $\mathcal{Q}$. *κ-symmetry* provides privacy properties (see Proposition 9) for the matrices S, S', S'': they result from Proposition 1 combined with the following proposition.

Proposition 6. *Let S, S', S'' be three invertible matrices of $\mathbb{Z}_n^{\kappa m\times\kappa m}$ and $\mathcal{Q}_{S\leftarrow(S',S''),p_1,...,p_{\kappa m}}$ be an arbitrary κ-symmetric operator with respect to S, S', S''. Let $(y_l)_{l=1,...,\kappa}$ be κ tuples defined by*

$$y_l = (s_i, s'_i, s''_i)_{i=(l-1)m+1,...,lm}$$

where s_i, s'_i, s''_i are the i^{th} row of respectively S, S', S''. Each monomial coefficient of $\mathcal{Q}_{S\leftarrow(S',S''),p_1,...,p_{\kappa m}}$ is an efficiently valuable κ-symmetric polynomial defined over the tuples $y_1, ..., y_\kappa$.

Proof. See Appendix D.

Remark 2. An operator $\mathcal{Q}$ is said to be κ-symmetric if it can be generated by efficiently valuable κ-symmetric polynomials of the tuples $y_1, ..., y_\kappa$ but it does not mean that Q is itself a κ-symmetric function.

6 Homomorphic operators

Throughout this section, $\tau = 3$.

6.1 Overview

Let $e = (c_z)_{z=1,...,\vartheta}$ and $e' = (c'_z)_{z=1,...,\vartheta}$ be two encryptions of x and x'. We wish to elaborate a public algorithm which computes a valid encryption $e'' = (c''_z)_{z=1,...,\vartheta}$ of $x + x'$ or xx' only using κ-symmetric operators $\mathcal{Q}$. Intuitively, operators $\mathcal{Q}$ allow to manipulate components of $|c_z|_{S_z}$ and $|c'_z|_{S_z}$. For concreteness, 2-degree polynomials can be computed by these operators. By combining these operators, (almost) arbitrary polynomials can be computed. Thanks to the constraints introduced in Encrypt1, it is possible to define the components of $|c''_z|_{S_z}$ as polynomials of the components of $|c_1|_{S_1}, ..., |c_\vartheta|_{S_\vartheta}$ and $|c'_1|_{S_1}, ..., |c'_\vartheta|_{S_\vartheta}$: it follows that it is possible to implement homomorphic operators by only applying (κ-symmetric) operators $\mathcal{Q}$. In the next section, we propose a construction using $O(\vartheta^2\delta)$ κ-symmetric operators $\mathcal{Q}$.

6.2 Product of basic vectors

In this section, we propose to define simple operators over basic vectors.

Definition 6. *(Products of basic vectors). Let $\delta, t \in \mathbb{N}^*$. Let $\mathrm{B} = (b_1, ..., b_\delta)$ and $\mathrm{B}' = (b'_1, ..., b'_\delta)$ be two basic vectors.*

- $\mathrm{B} \times \mathrm{B}'$ *denotes the basic vector*
$$\mathrm{B} \times \mathrm{B}' = (b_1b'_1, ..., b_\delta b'_\delta)$$
- *Let σ, σ' be two permutations of $\{1, ..., \delta\}$, $\mathrm{B} \star_{\sigma,\sigma'} \mathrm{B}'$ denotes the following basic vector*
$$\mathrm{B} \star_{\sigma,\sigma'} \mathrm{B}' = \left(b_{\sigma(1)}b'_{\sigma'(1)}, ..., b_{\sigma(\delta)}b'_{\sigma'(\delta)}\right)$$
- B^t *denotes the basic vector*
$$\mathrm{B}^t = (b_1^t, ..., b_\delta^t)$$
- *Let $I = \{1, ..., t\} \times \{1, ..., \delta\}$ and $(u_{ij})_{(i,j)\in I}$ be a family of indices of $\{1, ..., \delta\}$ such that*
$$\forall k \in \{1, ..., \delta\}, \quad \#\{(i,j) \in I | u_{ij} = k\} = t$$
  $\mathrm{B}^{\star(u_{11},...,u_{t\delta})}$ *denotes the basic vector*
$$\mathrm{B}^{\star(u_{11},...,u_{t\delta})} = (b_{u_{11}}...b_{u_{t1}}, ..., b_{u_{1\delta}}...b_{u_{t\delta}})$$

The permutations σ, σ' and the indices u_{ij} will be chosen at random for each operator and they will be omitted in notation, i.e. $\mathrm{B} \star_{\sigma,\sigma'} \mathrm{B}'$ and $\mathrm{B}^{\star(u_{11},...,u_{t\delta})}$ will be denoted by $\mathrm{B} \star \mathrm{B}'$ and $\mathrm{B}^{\star t}$.

Example. $\mathrm{B} = (b_1, b_2, b_3)$ *and* $\mathrm{B}^{\star 3} = (b_1b_1b_3, b_2b_3b_2, b_1b_2b_3)$.

6.3 Implementation

The operator Add is the key tool in the construction of homomorphic operators.

Definition 7. *(Operator* Add*). Let $\kappa \in \mathbb{N}^*$. Let T, T', T'' be three invertible matrices of $\mathbb{Z}_n^{3\kappa\delta \times 3\kappa\delta}$ and $w, w' \in \mathbb{Z}_n^{3\kappa\delta}$ be two vectors such that $|w|_T = (\mathrm{A}_{11}x_1, \mathrm{A}_{12}, \mathrm{A}_{13}, ..., \mathrm{A}_{\kappa 1}x_\kappa, \mathrm{A}_{\kappa 2}, \mathrm{A}_{\kappa 3})$ and $|w'|_{T'} = (\mathrm{A}'_{11}x'_1, \mathrm{A}'_{12}, \mathrm{A}'_{13}, ..., \mathrm{A}'_{\kappa 1}x'_\kappa, \mathrm{A}'_{\kappa 2}, \mathrm{A}'_{\kappa 3})$ where $(x_l, x'_l)_{l=1,...,\kappa}$ belong to $\mathbb{Z}_n$ and $(\mathrm{A}_{li}, \mathrm{A}'_{li})_{(l,i)\in\{1,...,\kappa\}\times\{1,...,3\}}$ are basic vectors. The operator $\mathsf{Add}_{T''\leftarrow(T,T')} : \mathbb{Z}_n^{3\kappa\delta} \times \mathbb{Z}_n^{3\kappa\delta} \to \mathbb{Z}_n^{3\kappa\delta}$ is defined by*

$$|\mathsf{Add}_{T''\leftarrow(T,T')}(w, w')|_{T''} = (\mathrm{H}_1(x_1 + x'_1), \mathrm{I}_1, \mathrm{J}_1, ..., \mathrm{H}_\kappa(x_\kappa + x'_\kappa), \mathrm{I}_\kappa, \mathrm{J}_\kappa)$$

where $\mathrm{E}_l \leftarrow \mathrm{A}_{l3} \star \mathrm{A}'_{l3}$, $\mathrm{F}_l \leftarrow \mathrm{E}_l^{\star\delta}$, $\mathrm{H}_l \leftarrow \mathrm{F}_l \star \mathrm{E}_l$, $\mathrm{G}_l \leftarrow \mathrm{E}_l^{\star 2\delta}$ and $\mathrm{I}_l, \mathrm{J}_l \leftarrow \mathrm{G}_l \star \mathrm{E}_l$ for any $l \in \{1, ..., \kappa\}$.

We propose to implement Add by using only κ-symmetric operators $\mathcal{Q}$. Many constructions can be imagined. The security of our scheme is strongly related to this construction. For security considerations detailed in Section 8, we propose a non-deterministic construction. Appendix E provides an implementation (where the operators $\mathcal{Q}$ are determined) of this construction and an implementation dealing with toy parameters is presented in Figure 2.

Proposition 7. *Operators Add can be implemented with $2\delta + 1$ κ-symmetric operators $\mathcal{Q}$.*

Proof. Our construction only deals with κ-symmetric operators $\mathcal{Q}$. It follows that the case $\kappa > 1$ can be straightforwardly deduced from the case $\kappa = 1$. For these reasons, we fix $\kappa = 1$. For any basic vector Z, the i^{th} component of Z is denoted by z_i. Let $T_1, ..., T_{2\delta}$ be invertible matrices of $\mathbb{Z}_n^{3\delta\times 3\delta}$ chosen at random.

1. Let $\mathrm{B} \leftarrow \mathrm{A}_{11} \star \mathrm{A}'_{12}$, $\mathrm{C} \leftarrow \mathrm{A}_{12} \star \mathrm{A}'_{11}$ and $\mathrm{E} \leftarrow \mathrm{A}_{13} \star \mathrm{A}'_{13}$. There exists a κ-symmetric operator $\mathcal{Q}_{T_1\leftarrow(T,T'),...}$ allowing to compute the vector $w_1 = \mathcal{Q}_{T_1\leftarrow(T,T'),...}(w, w')$ defined by

$$|w_1|_{T_1} = (\mathrm{B}x_1, \mathrm{C}x'_1, \mathrm{E})$$

2. For sake of simplicity, the i^{th} component of $|w_1|_{T_1}$ is denoted by α_i. By using the fact that B, C are basic vectors, there is an exponential number of ways to choose (see Appendix E to see such a choice) the values $(k_{iu})_{(i,u)\in\{1,...,3\delta\}\times\{1,...,2\delta\}}$ belonging to $\{1, ..., 3\delta\}$ such that

$$\left(\prod_{u=1}^{2\delta} \alpha_{k_{iu}}\right)_{i=1,...,3\delta} = (\mathrm{F}x_1, f_1 x'_1, \rho_2, ..., \rho_\delta, \mathrm{G})$$

   where $\mathrm{F} \leftarrow \mathrm{E}^{\star\delta}$, $\mathrm{G} \leftarrow \mathrm{E}^{\star 2\delta}$, $\rho_2, ..., \rho_\delta$ are arbitrary values and f_1 is the first component of F. Let us choose such values k_{iu} at random. By using κ-symmetric operators, we compute the recursive sequence $w_2, ..., w_{2\delta}$ defined by

$$w_u = Q_{T_u\leftarrow(T_{u-1},T_1)}(w_{u-1}, w_1) \text{ for } u = 2, ..., 2\delta$$

   such that $|w_u|_{T_u} = (\prod_{l=1}^{u} \alpha_{k_{il}})_{i=1,...,3\delta}$ ensuring that

$$|w_{2\delta}|_{T_{2\delta}} = (\mathrm{F}x_1, f_1 x'_1, \rho_2, ..., \rho_\delta, \mathrm{G})$$

3. Let $\mathrm{H} \leftarrow \mathrm{F} \star \mathrm{E}$, $\mathrm{I} \leftarrow \mathrm{G} \star \mathrm{E}$, $\mathrm{J} \leftarrow \mathrm{G} \star \mathrm{E}$. There exists a κ-symmetric operator $\mathcal{Q}_{T''\leftarrow(T_{2\delta},T_1),...}$ allowing to compute the vector $w_{2\delta+1} = \mathcal{Q}_{T''\leftarrow(T_{2\delta},T_1),...}(w_{2\delta}, w_1)$ defined by

$$|w_{2\delta+1}|_{T''} = (\mathrm{H}(x_1 + x'_1), \mathrm{I}, \mathrm{J})$$

   Output $w_{2\delta+1}$.

INPUTS:

$$w = T^{-1}\begin{pmatrix} r_1x_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} ; w' = T'^{-1}\begin{pmatrix} r'_1x'_1 \\ r'_2 \\ r'_3 \\ r'_4 \\ r'_5 \\ r'_6 \end{pmatrix}$$

INTERMEDIATE VECTORS:

$$w_1 = T_1^{-1}\begin{pmatrix} \alpha_1 = r_1r'_3x_1 \\ \alpha_2 = r_2r'_4 \\ \alpha_3 = r_3r'_1x'_1 \\ \alpha_4 = r_4r'_2 \\ \alpha_5 = r_5r'_5 \\ \alpha_6 = r_6r'_6 \end{pmatrix}$$

By choosing $[k_{iu}] = \begin{bmatrix} 1 & 5 & 2 & 5 \\ 6 & 5 & 6 & 6 \\ 3 & 4 & 5 & 5 \\ 4 & 2 & 6 & 1 \\ 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 \end{bmatrix}$, the vectors w_2, w_3, w_4 are defined by

$$w_2 = T_2^{-1}\begin{pmatrix} \alpha_1\alpha_5 \\ \alpha_6\alpha_5 \\ \alpha_3\alpha_4 \\ \alpha_4\alpha_2 \\ \alpha_5\alpha_5 \\ \alpha_6\alpha_6 \end{pmatrix} ; w_3 = T_3^{-1}\begin{pmatrix} \alpha_1\alpha_5\alpha_2 \\ \alpha_6\alpha_5\alpha_6 \\ \alpha_3\alpha_4\alpha_5 \\ \alpha_4\alpha_2\alpha_6 \\ \alpha_5\alpha_5\alpha_5 \\ \alpha_6\alpha_6\alpha_6 \end{pmatrix} ; w_4 = T_4^{-1}\begin{pmatrix} \alpha_1\alpha_5\alpha_2\alpha_5 = f_1x_1 \\ \alpha_6\alpha_5\alpha_6\alpha_6 = f_2 \\ \alpha_3\alpha_4\alpha_5\alpha_5 = f_1x'_1 \\ \alpha_4\alpha_2\alpha_6\alpha_1 = \rho_2 \\ \alpha_5\alpha_5\alpha_5\alpha_5 = g_1 \\ \alpha_6\alpha_6\alpha_6\alpha_6 = g_2 \end{pmatrix}$$

OUTPUT:

$$w_5 = T''^{-1}\begin{pmatrix} (\alpha_1\alpha_5\alpha_2\alpha_5 + \alpha_3\alpha_4\alpha_5\alpha_5)\alpha_6 \\ \alpha_6\alpha_5\alpha_6\alpha_6\alpha_5 \\ \alpha_5\alpha_5\alpha_5\alpha_5\alpha_6 \\ \alpha_6\alpha_6\alpha_6\alpha_6\alpha_5 \\ \alpha_5\alpha_5\alpha_5\alpha_5\alpha_5 \\ \alpha_6\alpha_6\alpha_6\alpha_6\alpha_6 \end{pmatrix} = T''^{-1}\begin{pmatrix} r_5r'_5(x_1 + x'_1) \\ r_6r'_6 \\ r_5^3r_5'^3 \\ r_6^3r_6'^3 \\ r_5^5r_5'^5 \\ r_6^5r_6'^5 \end{pmatrix}$$

Fig. 2. An implementation of Add for the toy parameters $\delta = 2$, $\tau = 3$ and $\kappa = 1$. Recall that constraints on vectors input in Add ensure that $r_1r_2 = r'_1r'_2 = r_3r_4 = r'_3r'_4 = r_5r_6 = r'_5r'_6 = 1$.

□

Remark 3. The construction of Add is probabilistic: randomness coming from the randomness of operators $\star$ and from the choice of the values k_{iu} and the matrices T_u. This randomness is introduced to limit *malicious uses* of the operators $\mathcal{Q}$ (see Section 8.3 for deeper explanations).

Proposition 8. *The operator $\oplus$ can be implemented with ϑ operators Add and the operator $\odot$ can be implemented with ϑ^2 κ-symmetric operators $\mathcal{Q}$ and $\vartheta(\vartheta-1)$ operators Add.*

Proof. By arguing similarly to the proof of Proposition 7, our construction only requires to be defined for $\kappa = 1$. Let $e = (c_z)_{z=1,\ldots,\vartheta}$ and $e' = (c'_z)_{z=1,\ldots,\vartheta}$ be two encryptions of x and x' such that:

- $|c_z|_{S_z} = (\text{A}_{z1}x_z, \text{A}_{z2}, \text{A}_{z3})$
- $|c'_z|_{S_z} = (\text{A}'_{z1}x'_z, \text{A}'_{z2}, \text{A}'_{z3})$

where $(\text{A}_{zi}, \text{A}'_{zi})_{i=1,\ldots,3}$ are basic vectors. Let us start by the operator $\oplus$. By using ϑ operators Add_z, one can compute

$$w_z = \mathsf{Add}_{z,S_z\leftarrow(S_z,S_z)}(c_z, c'_z)$$

Clearly, $(w_z)_{z=1,\ldots,\vartheta}$ is a valid encryption of $x + x'$. We state

$$e \oplus e' = (w_z)_{z=1,\ldots,\vartheta}$$

For sake of simplicity, the operator $\odot$ is detailed for $\vartheta = 2$ (the extension to the general case is straightforward). Let $(T_{zz'})_{(z,z')\in\{1,2\}^2}$ be 4 invertible matrices of $\mathbb{Z}_n^{3\delta\times3\delta}$ chosen at random. First, let us build 4 vectors $(w_{zz'})_{(z,z')\in\{1,2\}^2}$ defined by

$$|w_{zz'}|_{T_{zz'}} = (\text{A}_{z1} \star \text{A}'_{z'1}x_z x'_{z'}, \text{A}_{z2} \star \text{A}'_{z'2}, \text{A}_{z3} \star \text{A}'_{z'3})$$

Each vector $w_{zz'}$ can be obtained by applying a κ-symmetric operator $\mathcal{Q}$, i.e.

$$w_{zz'} = \mathcal{Q}_{T_{zz'}\leftarrow(S_z,S_{z'}),\ldots}(c_z, c'_{z'})$$

By using 2 operators Add_1 and Add_2, one can compute $w_1 = \mathsf{Add}_{1,S_1\leftarrow(T_{11},T_{22})}(w_{11}, w_{22})$ and $w_2 = \mathsf{Add}_{2,S_2\leftarrow(T_{12},T_{21})}(w_{12}, w_{21})$. Clearly, (w_1, w_2) is a valid encryption of xx'. We state,

$$e \odot e' = (w_1, w_2)$$

□

Given a key $K \leftarrow \mathsf{KeyGen1}(\lambda, 3)$, $\mathsf{OpGen}(K)$ outputs $\oplus, \odot$ by invoking QGen in order to output the $\vartheta^2(2\delta+2)$ κ-symmetric operators $\mathcal{Q}$ involved in the homomorphic operators. The function OpGen requires to compute $O(\vartheta^2\kappa^4\delta^5)$ multiplications in $\mathbb{Z}_n$ and its storage is $O(|n|\vartheta^2\kappa^3\delta^4)$.

7 The FHE

The private-key encryption scheme of Section 4 can be transformed in an FHE by publishing the homomorphic operators $\oplus, \odot$ and m encryptions $(e_v)_{i=1,\ldots,m}$ of public values $x_v \in \mathbb{Z}_n$: for instance $x_v = 2^v \bmod n$.

Definition 8. *Let λ be a security parameter.*

– *KeyGen*(λ). *Let* $K = \{(S_z)_{z=1,...,\vartheta}\} \leftarrow$ *KeyGen1*$(\lambda, 3)$, $\{\oplus, \odot\} \leftarrow$ *OpGen*(K) *and for all* $v = 1, ..., m$, $e_v \leftarrow$ *Encrypt1*(K, x_v).

$$sk = \{(S_z)_{z=1,...,\vartheta}\} \ ; \ pk = \{\oplus, \odot, (e_v)_{v=1,...,m}\}$$

– *Evaluate*$(C, e_1, ..., e_m)$. *To evaluate* $C(e_1, ..., e_m)$, *it suffices to compute each gate with the public homomorphic operators* $\oplus$ *and* $\odot$.

– *Encrypt*$(pk, x \in \mathbb{Z}_n)$. *It consists of evaluating a secret circuit* C *over the encryptions* $(e_v)_{v=1,...,m}$ *such that* $x = C(x_1, ..., x_m)$, *i.e. output* *Evaluate*$(C, e_1, ..., e_m)$

– *Decrypt*(sk, e). *Exactly follows* *Decrypt1*.

OpGen(K) outputs $\vartheta^2(2\delta+2)$ κ-symmetric operators $\mathcal{Q}$. These operators deal with the ϑ matrices $S_1, ..., S_\vartheta$ of sk and $\vartheta^2(2\delta+2) - 2\vartheta$ other intermediate invertible matrices denoted by $S_{\vartheta+1}, ..., S_{\vartheta^2(2\delta+2)-\vartheta}$. In the following of the paper, we will consider the (secret) tuples y_l^{sk} defined by

$$y_l^{sk} = (s_{ui}, \mathrm{B}_{vzlt}, x_{vzl})_{(u,v,z,i,t)\in\{1,...,\vartheta+\vartheta^2(2\delta-1)\}\times\{1,...,m\}\times\{1,...,\vartheta\}\times\{3(l-1)\delta+1,...,3l\delta\}\times\{1,...,\tau\}}$$

where s_{ui} denotes the i^{th} row of S_u and where $\mathrm{B}_{vzlt}, x_{vzl}$ are respectively the random basic vectors and the random values used by Encrypt1 to generate the public encryptions $(e_v)_{v=1,...,m}$. According to Proposition 4 and Proposition 6, all the public values of *pk* are κ-symmetric polynomials of these tuples. By extending definitions of Section 4.1, we define the following sets of polynomials:

– SP refers to the set of multi-variate polynomials $\phi : \left(\mathbb{Z}_n^{3\kappa\delta}\right)^r \to \mathbb{Z}_n$ defined by

$$\phi(w_1, ..., w_r) = \prod_{t=1}^{\gamma} s_{u_t i_t} w_{k_t}$$

where $\gamma, r \in \mathbb{N}^*$, $i_t \in \{1, ..., 3\kappa\delta\}$, $u_t \in \{1, ..., \vartheta^2(2\delta+2) - \vartheta\}$ and $k_t \in \{1, ..., r\}$.

– LSP: the set of polynomial-size linear combinations of polynomials of SP.

– SP_l: the subset of SP such that $i_t \in \{3(l-1)\delta + 1, ..., 3l\delta\}$

– SP^γ: the set of polynomials of SP of degree equal to γ, i.e.

$$\mathsf{SP}^\gamma = \{\phi \in \mathsf{SP} |\deg(\phi) = \gamma\}$$

Remark 4. The number of monomials of any $\phi \in \mathsf{SP}^\gamma$ is $\Omega\left(\frac{\delta^{\gamma-1}}{\gamma!}\right)$. Note that this number is exponential provided $\gamma = \Theta(\lambda^{\epsilon>0})$ and $\delta = \Theta(\lambda)$.

Security naturally deals with these polynomials because they allow to compute polynomials over the components of $|w_k|_{S_u}$. A representation R_ϕ of $\phi \in \mathsf{LSP}$ is said to be effective if its storage is polynomial and if it allows to evaluate ϕ in polynomial-time. The following result is a direct application of Proposition 1, 4 and 6.

Proposition 9. *Let* $l \in \{1, ..., \kappa\}$ *and* $\gamma \in \mathbb{N}^*$ *such that* γ *is not a multiple of* κ *(*$|\gamma|$ *polynomial in* λ*). Let* $\phi \in SP_l \cup SP^\gamma$ *be a polynomial and* R_ϕ *be an effective representation of* ϕ. *By assuming the hardness of factorization, recovering* R_ϕ *only given pk is difficult.*

Proof. (Sketch. See Appendix C for a complete detailed proof.) According to Proposition 4 and Proposition 6, pk contains only κ-symmetric efficiently valuable polynomials evaluated over the tuples $y_1^{sk}, ..., y_\kappa^{sk}$. These tuples are chosen at random according to the probability distribution considered in Problem 2. Consequently, according to Proposition 1, a polynomial attacker cannot recover any non κ-symmetric product $\pi(y_1^{sk}, ..., y_\kappa^{sk})$. To conclude, it suffices to notice that an effective representation R_ϕ of ϕ allows to polynomially compute a non κ-symmetric product $\pi(y_1^{sk}, ..., y_\kappa^{sk})$.
□

Corollary 2. *By assuming the hardness of factorization, the secret matrices $(S_u)_{u=1,...,\vartheta^2(2\delta+2)-\vartheta}$ and the polynomials $(\Phi_l)_{l=1,...,\kappa}$ cannot be polynomially recovered only given pk.*

A natural arising question consists of wondering whether polynomials $\phi \in \mathsf{SP}^{t\kappa}$ with $t > 0$ can be recovered. Let us assume $\kappa = \Theta(\lambda^{\epsilon>0})$ and $\delta = \Theta(\lambda)$. In this case, ϕ has an exponential number of monomials (see Remark 4). Thus, its expanded form cannot be polynomially output. Proposition 9 ensures that a polynomial attacker cannot factor ϕ with small polynomials. Consequently, a polynomial attacker cannot recover the expanded representation or any effective factored representation of any $\phi \in \mathsf{SP}$ only given pk assuming the hardness of factorization.

A representation R_ϕ of $\phi \in$ LSP is said to be κ-symmetric if it can be generated by (an arbitrary number of) efficiently valuable κ-symmetric polynomials of $y_1^{sk}, ..., y_\kappa^{sk}$. For instance, the expanded representation of the polynomial $\Phi_1 + ... + \Phi_\kappa$ is κ-symmetric[6]. Proposition 9 can be extended to show that it is difficult to recover non κ-symmetric effective representations (by extending Proposition 1 as explained by Remark 1). However, it is not sufficient for ensuring security. Indeed, the expanded representation of the polynomial $\Phi = \Phi_1 + ... + \Phi_\kappa$ or $\Phi = \Phi_1 ... \Phi_\kappa$ is κ-symmetric and its knowledge would break semantic security. Nevertheless, assuming $\delta = \Theta(\lambda)$, the number of monomials of Φ is exponential implying that its expanded representation is not effective. Besides, according to Proposition 9, it is difficult to find any of its natural effective representations, i.e. sum of products of *small* polynomials of SP, provided $\delta = \Theta(\lambda)$ and $\kappa = \Theta(\lambda^{\epsilon>0})$. To be polynomially recovered and evaluated, Φ should be represented by a κ-symmetric effective representation R_Φ, i.e. R_Φ should be expressed by a polynomial number of κ-symmetric values defined over the tuples y_l^{sk}. We conjecture that such a representation does not exist (see Appendix H for a toy example highlighting this).

A polynomial attacker can only hope to recover polynomials ϕ having κ-symmetric effective representations. The construction of the FHE should ensure that such polynomials could not be used to break semantic security. In particular, polynomials having a κ-symmetric effective expanded representation could be recovered with attacks by linearization (consisting of solving linear systems where the variables are the monomial coefficients). Intuitively, randomness introduced in Add should prevent our scheme against such attacks. This is extensively studied in the next section.

8 Attacks by linearization.

The public key pk can be naturally regarded as a system (Sys) of nonlinear equations (partially unknown because of the randomness over the choice of each operator $\mathcal{Q}$ belonging to $\oplus$ or $\odot$) where each tuple y_l^{sk} is a solution. Thanks to κ-symmetry, the previous section tends to show that the resolution of (Sys) is quite intractable. The attack by linearization proposed for the additively homomorphic cryptosystem can be straightforwardly transposed for the FHE. It consists of solving the linear system

$$\phi(e_v) = x_v$$

[6] Each monomial coefficient is a κ-symmetric polynomial of $y_1^{sk}, ..., y_\kappa^{sk}$.

where $(e_v)_{v=1,...m}$ are encryptions of $(x_v)_{v=1,...,m}$ and ϕ is a multivariate polynomial[7] of degree δ such that its monomial coefficients are the variables of the linear system. Its resolution provides a linear combination[8] ϕ^* of the decryption polynomials $(\Phi_l)_{l=1,...,\kappa}$. However, provided $\delta = \Theta(\lambda)$, this attack fails because the number of monomials of ϕ is exponential. Because of the introduction of homomorphic operators, efficient attacks by linearization could appear. In this section, we give a general framework to analyze the security of the FHE with respect to these attacks.

8.1 General framework

By considering t encryptions $e_1, ..., e_t$ of known values $x_1, ..., x_t$, an attacker can build r vectors $w_1, ..., w_r$, for instance, by using public κ-symmetric operators $\mathcal{Q}$ in an arbitrary way. Let us imagine that there are m (m being polynomial) multivariate polynomials $\phi_1, ..., \phi_m$ satisfying

$$z_1\phi_1(w_1, ..., w_r) + ... + z_m\phi_m(w_1, ..., w_r) = 0 \qquad (1)$$

where $z_1, ..., z_m$ are functions of the encrypted values $x_1, ..., x_t$. For each choice of known encryptions $e_1, ..., e_t$, we get a linear equation where the variables are the monomial coefficients of ϕ_i. By iterating this process on new encryptions, we get a linear system which can be solved in polynomial time if the number of monomials of each ϕ_i is polynomial. The knowledge of a solution $(\phi_1^*, ..., \phi_m^*)$ satisfying this system[9] can be used to break semantic security. Indeed, given a new encryption e_1° of an unknown value x_1°, the attacker builds the vectors $(w_1^\circ, ..., w_r^\circ)$ by considering the encryptions $e_1^\circ, e_2..., e_t$. The knowledge of the polynomials $\phi_1^*, ..., \phi_m^*$ provides relationships between $x_1^\circ, x_2..., x_t$. Fortunately, this attack does not work if it exists i_0 s.t. the expanded representation of $\phi_{i_0}^*$ is exponential-size[10]. In next sections, we will consider the two following oracles:

- the oracle $\mathcal{O}_1$ inputs two (valid) encryptions e_1, e_2 belonging to pk or previously output by itself and outputs $e_1 \oplus e_2$, $e_1 \odot e_2$ and all the intermediate vectors computed during the computation of homomorphic operators (vectors output by operators $\mathcal{Q}$).
- let $\mathcal{Q}_1$,...,$\mathcal{Q}_{\vartheta^2(2\delta+2)}$ be the $\vartheta^2(2\delta+2)$ κ-symmetric operators of pk. The oracle $\mathcal{O}_2$ inputs $i \in \{1, ..., \vartheta^2(2\delta+2)\}$ and two vectors w, w' belonging to $\mathbb{Z}_n^{3\kappa\delta}$ and outputs $\mathcal{Q}_i(w, w')$.

Before considering the real-life setting, linearization attacks will be analyzed in the two following relaxed settings:

- **Setting 1.** *The public operators $\mathcal{Q}$ are replaced by accesses to $\mathcal{O}_1$*
- **Setting 2.** *The public operators $\mathcal{Q}$ are replaced by accesses to $\mathcal{O}_2$*

8.2 Linearization attacks in setting 1

In this section, the operators $\mathcal{Q}$ of pk are replaced by accesses to $\mathcal{O}_1$. An attacker can recursively invoke $\mathcal{O}_1$ over encryptions $e_1, ..., e_m$ of pk and/or encryptions previously output by $\mathcal{O}_1$. The main tool of our construction is the operator Add. At first, let us study it separately[11] from the whole construction by adopting the notation of Definition 7 and Proposition 7. Two vectors w and w' are input and $2\delta + 1$

[7] having the same monomials than the decryption polynomials Φ_l.
[8] The monomial coefficients depend on the secret values r_{vi}, r'_{vi} used in the encryptions e_v.
[9] Note that each monomial coefficient of ϕ_i^* is a κ-symmetric value defined over the tuples $y_1^{sk}, ..., y_\kappa^{sk}$.
[10] meaning that the number of monomials is exponential.
[11] vectors input in each operator Add can be randomized without introducing interesting polynomials relations. It enforces the idea that the public operators Add of pk can be studied separately and that vectors output by Add are pseudo-random (under constraints linked to their definition). An example of such randomization is provided in Appendix H.

intermediate vectors $w_1, ..., w_{2\delta+1}$ are computed during the execution of $\mathsf{Add}(w, w')$, each one being output by a κ-symmetric operator $\mathcal{Q}_{T_1 \leftarrow ...}, ..., \mathcal{Q}_{T_{2\delta} \leftarrow ...}, \mathcal{Q}_{T'' \leftarrow ...}$, i.e.

$$\begin{cases} w_1 &= Q_{T_1 \leftarrow (T,T')}(w, w') \\ w_u &= Q_{T_u \leftarrow (T_{u-1}, T_1)}(w_{u-1}, w_1) \text{ for } u = 2, ..., 2\delta \\ w_{2\delta+1} &= \mathcal{Q}_{T'' \leftarrow (T_{2\delta}, T_1)}(w_{2\delta}, w_1) \end{cases}$$

According to the definition of setting 1, it is assumed that Add output all these vectors (and not only $w_{2\delta+1}$), i.e.

$$(w_1, ..., w_{2\delta+1}) \leftarrow \mathsf{Add}(w, w')$$

In order to homogenize notation, we rename T, T', T'' to respectively $T_{-1}, T_0, T_{2\delta+1}$. Let us consider the subset $\mathsf{SP}_{\mathsf{Add}} \subset \mathsf{SP}$ of polynomials $\phi : \left(\mathbb{Z}_n^{3\kappa\delta}\right)^{2\delta+3} \to \mathbb{Z}_n$ defined by

$$\phi(w_{-1}, w_0, ..., w_{2\delta+1}) = \prod_{r=1}^{\gamma} t_{u_r i_r} w_{u_r}$$

with $\gamma > 0$, $u_r \in \{-1, ..., 2\delta + 1\}$, $i_r \in \{1, ..., 3\kappa\delta\}$, and t_{ui} is the i^{th} row of T_u. By definition, $\phi(w, w', \mathsf{Add}(w, w'))$ is a product of components of $|w|_T$, $|w'|_{T'}$, $|w_1|_{T_1} ... |w_{2\delta+1}|_{T_{2\delta+1}}$ (denoted by $|w|$, $|w'|$, $|w_1| ... |w_{2\delta+1}|$ in the following of this section). Because of multiplicative constraints introduced in our scheme, the knowledge of polynomials of $\mathsf{SP}_{\mathsf{Add}}$ could intuitively be relevant to break security. Primarily, we wonder whether it is possible to recover *a linear combination* $z \in \mathsf{co}(\{x_i, x'_i \mid i = 1, ..., \kappa\})$ (e.g. $z = x_1$, $z = x'_1$, $z = x_1 + x'_1$) with small polynomials of $\mathsf{SP}_{\mathsf{Add}}$. At first, we are looking for two (small) polynomials ϕ_1 and ϕ_2 such that for all $(w, w') \in \mathbb{Z}_n^{3\kappa\delta} \times \mathbb{Z}_n^{3\kappa\delta}$ satisfying constraints of Definition 7,

$$\phi_1(w, w', \mathsf{Add}(w, w')) = z\phi_2(w, w', \mathsf{Add}(w, w')) \tag{2}$$

In other words, we are looking for two small products π_1, π_2 of components of $|w|, |w'|, |w_1|, ..., |w_{2\delta+1}|$ such that $\pi_1 = z\pi_2$. For instance, in the toy example presented in Figure 2, it can be easily verified that there exists two linear functions ϕ_1 and ϕ_2 satisfying (2), e.g.

$$t_{31} w_3 = x_1 t_{15} w_1$$

where t_{31} and t_{15} are respectively the 1^{st} row of T_3 and the 5^{th} row of T_1.

Let us examine why such relationships are damageable for security by considering the homomorphic operator $\oplus$ (the same analysis can be done for $\odot$) in the case $\vartheta = 2$ (for sake of simplicity). This operator consists of computing $\mathsf{Add}_{1,S_1 \leftarrow (S_1,S_1)}(c_1, c'_1)$ and $\mathsf{Add}_{2,S_2 \leftarrow (S_2,S_2)}(c_2, c'_2)$ (see notation of the proof of Proposition 8). Assume that there are small polynomials satisfying (2) for both operators Add_1 and Add_2, i.e.

$$\phi_{11}(c_1, c'_1, \mathsf{Add}_1(c_1, c'_1)) = x_{11}\phi_{12}(c_1, c'_1, \mathsf{Add}_1(c_1, c'_1))$$
$$\phi_{21}(c_2, c'_2, \mathsf{Add}_2(c_2, c'_2)) = x_{21}\phi_{22}(c_2, c'_2, \mathsf{Add}_2(c_2, c'_2))$$

As $x = x_{11} + x_{21}$, it provides the following polynomial relationship (leading to a linearization attack), i.e.

$$\phi_{11}(c_1, c'_1, ...)\phi_{22}(c_2, c'_2, ...) + \phi_{12}(c_1, c'_1, ...)\phi_{21}(c_2, c'_2, ...) = x\phi_{12}(c_1, c'_1, ...)\phi_{22}(c_2, c'_2, ...)$$

The construction of operators Add was oriented in order to avoid such relationships, i.e. small polynomials satisfying (2). Intuitively, the probability (where the coin toss is the choice of the k_{iu} in the operator Add) that such relationships exist is expected to exponentially decrease with δ. Before to experiment this intuition, the following lemma implies that one can restrict our analysis to the case $\kappa = 1$.

Lemma 1. *If there exists ϕ_1, ϕ_2 belonging to $\mathsf{SP}_{\mathsf{Add}}$ satisfying (2) then there exists polynomials ϕ'_1, ϕ'_2 belonging to $\mathsf{SP}_{\mathsf{Add}} \cap \mathsf{SP}_1$ satisfying (2) such that* $\deg(\phi'_1) = \deg(\phi_1)$ *and* $\deg(\phi'_2) = \deg(\phi_2)$.

Proof. See Appendix I.

Experiments consisting of exhaustively searching ϕ_1, ϕ_2 with $\deg \phi_1 + \deg \phi_2 = d$ and $z = x_1$, $z = x'_1$ or $z = x_1 + x'_1$ have been done for small values of $d = 1, 2, 3$. Concretely, the values k_{iu} and the vector w were randomly generated (under the constraints of Definition 7 and proof of Proposition 7). To increase the probability of collisions, we stated $|w'| = |w|$. Finally, the vectors $|w_1|,\ldots,|w_{2\delta+1}|$ were generated[12] as specified in the proof of Proposition 7. Then, we were looking for two products π_1 and π_2 respectively of d_1 and d_2 components of these vectors such that $d_1 + d_2 = d$ and $\pi_1 = z\pi_2$. For fixed values of d, the probability (the toss coin being the choice of the indexes k_{iu}) of the existence of such polynomials ϕ_1, ϕ_2 seems to exponentially decrease with δ. The results of these experiments are presented in Figure 3. Besides, in Appendix E, we propose an instantiation of Add (where the indexes k_{iu} are fixed) and we prove

$d \setminus \delta$	2	3	4	5	6	7	8	9	10	11	12	13
1	0.785	0.383	0.156	0.060	0.023	0.015	0.002	0.000	-	-	-	-
2	1.000	0.812	0.593	0.299	0.103	0.043	0.012	0.006	0.000	-	-	-
3	1.00	1.00	1.00	0.99	0.95	0.88	0.70	0.50	0.38	0.23	0.17	0.01

Fig. 3. Estimate of the probability that there exists polynomials ϕ_1, ϕ_2 with $\deg \phi_1 + \deg \phi_2 = d$ satisfying (2) in function of δ (with $\tau = 3$ and $\kappa = 1$). Each value of the table is the mean of 1000 experiments for $d = 1, 2$ and 100 for $d = 3$.

(see Proposition 10) that there are not polynomials ϕ_1, ϕ_2 satisfying (2) such that $\deg \phi_1 + \deg \phi_2 < \delta/2$. By assuming that the mean case is *not too far* from the worst case, we propose the following conjecture.

Conjecture 2. It exists $\epsilon_0 > 0$ such that the probability (the coin toss being the choice of coefficients k_{iu} in the construction of Add) that there exists polynomials $\phi_1, \phi_2 \in \mathsf{SP}_{\mathsf{Add}}$ satisfying (2) with $\deg \phi_1 + \deg \phi_2 = O(\delta^{\epsilon_0})$ *exponentially decreases with δ.*

In other words, provided $\delta = \Theta(\lambda)$, there does not exist polynomials ϕ_1, ϕ_2 satisfying (2) having a number of monomials in $O\left(2^{\lambda^{\epsilon_0}}\right)$ (see Remark 4). It implies that the attack (described above) is exponential.

Remark 5. In Appendix E, we propose an operator Add ensuring the non-existence of small polynomials satisfying (2). It can be wondered why this operator is not adopted. The main reason is that randomness is needed in the construction of Add in order to resist against linearization attacks in setting 2. Nevertheless, we believe that it is possible to add randomness in the construction proposed in Appendix E and to keep true proposition 10 at the same time.

Remark 6. Conjecture 2 assumes that the probability of the existence of small polynomials satisfying (2) exponentially decreases. In fact, it would suffice that this probability is smaller than 1/2 and to have an efficient procedure to test it (the existence of such polynomials).

Remark 7. We have investigated the problem of the existence of efficient linear attacks. However, the non-existence of such attacks is not a necessary condition for the security of our scheme. Indeed, it would suffice to show that the attacker is not able to efficiently find such attacks.

[12] These experiments do not deal with the matrices $(T_u)_{u=-1,\ldots,2\delta+1}$: they can be arbitrarily fixed to the identity matrix.

• *Justification of the parameter* ϑ. An obvious relationship (intrinsic to the operator Add) deals with the vector $w_{2\delta}$ (see proof of Proposition 7). Indeed, by construction

$$|w_{2\delta}| = (f_1 x_1, f_2, ..., f_\delta, f_1 x_1', ...)$$

Roughly speaking, the same coefficient f_1 hides both x_1 and x_1'. It follows that there are two linear functions ϕ_1 and ϕ_2 satisfying

$$x_1' \phi_1(w_{2\delta}) = x_1 \phi_2(w_{2\delta})$$

with $\phi_1(w_{2\delta}) = t_{2\delta,1} w_{2\delta}$ and $\phi_2(w_{2\delta}) = t_{2\delta,\delta+1} w_{2\delta}$. This could be *a priori* a source of failures for our scheme. Let us see what happens when considering the ϑ operators Add_z involved in $\oplus$ (the same analysis can be done for $\odot$). Let $e = (c_z)_{z=1,...,\vartheta}$ and $e' = (c_z')_{z=1,...,\vartheta}$ be two encryptions (see notation of Proposition 8) of x and x' and

$$(w_{z1}, ..., w_{z,2\delta+1}) \leftarrow \mathsf{Add}_{z,S_z \leftarrow (S_z,S_z)}(c_z, c_z')$$

According to the above analysis, there are 2ϑ linear functions $(\phi_{z1}, \phi_{z2})_{z=1,...,\vartheta}$ such that

$$\begin{aligned}
&x_{11} \phi_{11}(w_{1,2\delta}) = x_{11}' \phi_{12}(w_{1,2\delta}) \\
&... \\
&x_{\vartheta 1} \phi_{\vartheta 1}(w_{\vartheta,2\delta}) = x_{\vartheta 1}' \phi_{\vartheta 2}(w_{\vartheta,2\delta})
\end{aligned}$$

We let the reader see how deriving this relationship to get an efficient linear attack for the case $\vartheta = O(1)$. Let us see that linear attacks linked to these relationships become exponential provided $\vartheta = \Theta(\lambda^{\epsilon_0>0})$ (providing a justification for this parameter). To achieve this, we first enforce the adversarial power by revealing the values x_{z1}' to the attacker, e.g. $x_{z1}' = 1$ for sake of simplicity. In this case, we get

$$\begin{aligned}
&x_{11} = \phi_{12}(w_{1,2\delta}) / \phi_{11}(w_{1,2\delta}) \\
&... \\
&x_{\vartheta 1} = \phi_{\vartheta 2}(w_{\vartheta,2\delta}) / \phi_{\vartheta 1}(w_{\vartheta,2\delta})
\end{aligned}$$

implying the following natural (and simplest) relationship (exploiting $x = x_{11} + ... + x_{\vartheta 1}$),

$$x \prod_{z=1}^{\vartheta} \phi_{z1}(w_{z,2\delta}) = \sum_{z=1}^{\theta} \phi_{z2}(w_{1,2\delta}) \prod_{t \in \{1,...,\vartheta\} \setminus \{z\}} \phi_{t1}(w_{t,2\delta})$$

This leads to a linear attack where the degree of the involved polynomials is ϑ. By using the main argument of this paper, these polynomials are exponential-size provided $\delta = \Theta(\lambda)$ and $\vartheta = \Theta(\lambda^{\epsilon_0>0})$ making this attack fail.

The analysis of this section can be investigated in an informal but more intuitive way. Indeed, given an encryption $c = (c_z)_{z=1,...,\vartheta}$, a subset of strictly less than ϑ vectors c_z is statistically indistinguishable from random ones. Thus, intuitively, attacks exploiting the intrinsic relationship presented above should involve at least ϑ vectors leading to attacks dealing with polynomials of degree larger than ϑ (and thus exponential provided $\delta = \Theta(\lambda)$ and $\vartheta = \Theta(\lambda^{\epsilon_0>0})$).

Conjecture 3. Assuming $\delta = \Theta(\lambda)$ and $\vartheta = \Theta(\lambda^{\epsilon_0>0})$, there are efficient linearization attacks in setting 1 with negligible probability.

8.3 Linearization attacks in setting 2

In this section, the public operators $\mathcal{Q}$ are replaced by $\mathcal{O}_2$ which simulates the computation of any public operator $\mathcal{Q}$. Arbitrary vectors can be input in $\mathcal{O}_2$. To compute $e_1 \oplus e_2$ or $e_1 \odot e_2$, $m = \vartheta^2(2\delta + 2)$ vectors $(w_u)_{u=1,...,m}$ are output by operators $\mathcal{Q}$, i.e. $w_u = \mathcal{Q}_{S_u \leftarrow (S_{u'}, S_{u''})}(w_{u'}, w_{u''})$ where S_u is an invertible matrix chosen at random. Roughly speaking, the secret information contained in w_u are the components of $S_u w_u$. As the matrices $(S_u)_{i=1,...,\vartheta^2(2\delta+1)-\vartheta}$ are randomly and independently chosen, $S_{u' \neq u} w_u$ and $S_u w_u$ are independent: it ensures that an attacker does not get any advantage by substituting w_u by $w_{u' \neq u}$ in the computation of homomorphic operators. In particular, this prevents our scheme against the existence of relevant operators Add inputting pairs of vectors $(c_z, c_{z'})$ belonging to the same encryption e, i.e. $e = (c_1, ..., c_\vartheta)$ (see the previous section to understand why this would be damageable for security).

Nevertheless, the adversary can substitute w_u by an *old* vector w'_u previously computed. This is not relevant assuming pseudo-randomness of encryptions produced by homomorphic operators. Other guarantees against such substitutions come from randomness in the choice of operators $\mathcal{Q}$. Let us present an attack (exponential in δ) highlighting this.

An attack. Let $a \in \mathbb{Z}_n^{3\kappa\delta}$ be an arbitrary vector and let us assume that an attacker has guessed the set

$$L_{\mathsf{Add}} = \{u \in \{1, ..., 2\delta\} : k_{1u} \notin \{1, ..., \delta\}\}$$

where the values $(k_{1u})_{u=1,...,2\delta}$ are the ones used to build Add. For instance, $L_{\mathsf{Add}} = \{2, 4\}$ in the example of Figure 2. We let the reader check that L_{Add} contains exactly δ elements. In step 2 of the construction of Add, by substituting w_1 with a each time $u \in L_{\mathsf{Add}}$, it is ensured that the first component of $|w_{2\delta}|$ is equal to Ax_1 where A is a constant depending only of a. It leads to an obvious efficient linearization attack if the sets L_{Add_z} have been guessed for all the ϑ operators Add_z involved in the operator $\oplus$. To prevent the scheme against this attack, an attacker should not guess the sets $(L_{\mathsf{Add}_z})_{z=1,...,\vartheta}$ with a non negligible probability. This probability is equal to

$$\binom{2\delta}{\delta}^{-\vartheta}$$

The attack fails[13] provided $\vartheta\delta = \Theta(\lambda)$.

Conjecture 4. Assuming $\delta = \Theta(\lambda)$ and $\vartheta = \Theta(\lambda^{\epsilon_0 > 0})$, the non-existence of efficient linearization attacks in setting 1 $\Rightarrow$ the non-existence of efficient linearization attacks in setting 2.

8.4 Linearization attacks in real-life setting

The only difference with the previous setting is that κ-symmetric operators $\mathcal{Q}$ are not anymore simulated by $\mathcal{O}_2$. New linearization attacks could appear. For instance, the values k_{iu} used in the construction of Add could be polynomially recovered making efficient the linearization attack described in the previous section. Moreover, one could imagine that new κ-symmetric operators $\mathcal{Q}$ can be polynomially derived from public operators $\mathcal{Q}$. Let us argue against this.

At this step of the paper, the authors assume that the reader should be convinced of the security of the additively homomorphic encryption scheme. This scheme deals with the operator $\mathcal{Q}_S$. The security of this scheme suggests that this operator does not introduce intrinsic failures. In the FHE, each operator $\mathcal{Q}_{S_u \leftarrow (S_{u'}, S_{u''}),...}$ can be associated to a system (Sys) of nonlinear equations (2-degree equations) where the variables are the coefficients of the invertible matrices $S_u, S_{u'}, S_{u''}$. In our construction, it does not exist

[13] Note that this attack is relevant for the construction proposed in Appendix E. Indeed, in this construction, L_{Add} is deterministic and thus implicitly known by the attacker, i.e. $L_{\mathsf{Add}} = \{2, 4, ..., 2\delta\}$.

two different operators $\mathcal{Q}$ dealing with the same triplet of matrices $S_u, S_{u'}, S_{u''}$. Proposition 9 says that the coefficients of $S_u, S_{u'}, S_{u''}$ cannot be found, meaning that the system of equations derived from each operator $\mathcal{Q}$ is quite intractable. Furthermore, because of the randomness introduced in Add, the operators $\mathcal{Q}$ are randomly chosen. Thus, (Sys) is widely unknown. Moreover, many ways to add randomness in each operator $\mathcal{Q}$ can be imagined. The simplest way consists of adding free (not involved in constraints) components $i = 3\kappa\delta + 1, ...$ and of choosing p_i (see Section 5) at random: an arbitrary number (each p_i provides $\Theta(\delta^2)$ new variables) of new variables[14] are introduced in the equations induced by each operator $\mathcal{Q}$. Another one is presented in detail in Appendix K (presented for the operator $\mathcal{Q}_S$ but the extension to any operator Q is straightforward).

Conjecture 5. Assuming $\delta = \Theta(\lambda)$ and $\vartheta = \Theta(\lambda^{\epsilon_0 > 0})$, the non-existence of efficient linearization attacks in setting 2 $\Rightarrow$ the non-existence of efficient linearization attacks in the real-life setting.

8.5 Efficiency

The computation of an operator $\mathcal{Q}$ requires $O(\kappa^3\delta^3)$ multiplications in $\mathbb{Z}_n$. Moreover, $\oplus$ requires the application of $O(\vartheta\delta)$ operators $\mathcal{Q}$ and $O(\vartheta^2\delta)$ for $\odot$. Thus, by denoting by $M(n)$ the runtime of multiplications in $\mathbb{Z}_n$, the running time per addition gate is $O(\vartheta\kappa^3\delta^4 M(n))$ and the running time per multiplication gate is $O(\vartheta^2\kappa^3\delta^4 M(n))$. The running time of decryption is $O(\vartheta\kappa\tau\delta^2 M(n))$. A ciphertext contains ϑ $3\kappa\delta$-vectors in $\mathbb{Z}_n$ implying that the ratio cipher size/plaintext size is equal to $3\kappa\vartheta\delta$. In term of storage, the biggest part of the public key is the operator $\mathcal{Q}$ containing $O(\kappa^3\delta^3)$ elements of $\mathbb{Z}_n$ leading to a space complexity in

$$O(|n|\vartheta^2\kappa^3\delta^4)$$

Attacks (in particular attacks by linearization) should be better quantified in order to propose instantiations of parameters.

9 Discussion and open questions

In this paper, a very simple FHE based on very simple tools was developed. Its security is linked to the difficulty of solving nonlinear systems of equations. By using arguments of symmetry, it was shown that the resolution of the system of equations (derived from pk) is intractable. However, it is not sufficient to ensure security against attacks by linearization. The main obstacle to prove security consists of showing that all linear attacks are exponential. We argue in this sense but further investigations should be done. Moreover, improvements of our scheme deal with important open questions:

- κ-symmetry provides formal security guarantees but this parameter is not useful to protect the scheme against attacks by linearization. Can this parameter be fixed to 1?
- the resolution of systems of nonlinear equations is $\mathcal{NP}$-complete in Z_n even if the factorization of n is known. Thus, it can be wondered whether n can be chosen as a large prime? a small prime?

A positive answer to these questions would lead to an efficient FHE competitive with other classical (even not homomorphic) cryptosystems.

References

1. 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.

[14] independent of other variables of pk

2. T. Elgamal. A public key cryptosystem and a signature sheme based on discrete logarithms. In *IEEE transactions on Information Theory*, pages 31:469–472, 1985.
3. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
4. Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, pages 465–482, 2012.
5. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the aes circuit. In *CRYPTO*, pages 850–867, 2012.
6. Kristin Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? *IACR Cryptology ePrint Archive*, 2011:405, 2011.
7. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.
8. Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. In *ASIACRYPT*, pages 377–394, 2010.
9. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010.

A Proof of Proposition 1

The proof consists of building a polynomial algorithm of factorization A by using a solver B of Problem 1 (resp. problem 2) as subroutine. Let us denote by D the probability distribution of $(y_1, ..., y_\kappa)$ induced by Problem 1 (resp. problem 2). D is effective in the sense that D can be simulated in polynomial-time, i.e. $(y_1, ..., y_\kappa)$ can be generated at random according to D in polynomial-time given a (polynomial-time) random generator of elements of $\mathbb{Z}_n$. Let us consider the following polynomial-time algorithm A:

Repeat

1. Let $(y_1, ..., y_\kappa) \stackrel{D}{\leftarrow} \mathbb{Z}_n^{t\kappa}$
2. Compute $s_j = s_j(y_1, ..., y_\kappa)$ for all $j = 1, ..., m$.
3. Compute $\Pi = \pi(y_1, ..., y_\kappa)$
4. Apply B on the inputs $s_1, ..., s_m$, i.e. $\Pi_B \leftarrow B(s_1, ..., s_m)$

until $\gcd(\Pi - \Pi_B, n) \neq 1$

output $\gcd(\Pi - \Pi_B, n)$

By construction, this algorithm is correct. Let us show that it terminates in polynomial time. First, each step of A can be computed in polynomial-time implying that A is polynomial if the number of steps of A is polynomial (or equivalently, if the probability to terminate at each iteration is non negligible). As the product π is assumed to be non κ-symmetric, it can be assumed (without loss of generality) that $\pi(y_1, y_2, ..., y_\kappa) \neq \pi(y_2, y_1, ..., y_\kappa)$. Let us consider the function $h : \mathbb{Z}_n^{t\times\kappa} \to \mathbb{Z}_n^{t\times\kappa}$ such that $(y'_1, ..., y'_\kappa) = h(y_1, ..., y_\kappa)$ is defined by

- $y'_l = y_l$ for $l > 2$
- $y'_{1i} \equiv y_{1i} \mod p$ and $y'_{2i} \equiv y_{2i} \mod p$ for all $i = 1, ..., t$
- $y'_{1i} \equiv y_{2i} \mod q$ and $y'_{2i} \equiv y_{1i} \mod q$ for all $i = 1, ..., t$.

Because of the symmetry of constraints, one easily checks that if $(y_1, ..., y_\kappa)$ satisfies constraints of Problem 1 (resp. Problem 2) then $(y'_1, ..., y'_\kappa)$ also satisfies these constraints[15]. It implies that $(y_1, ..., y_\kappa)$ and $(y'_1, ..., y'_\kappa) = h(y_1, ..., y_\kappa)$ have the same probability under D, i.e. $P_D(y_1, ..., y_\kappa) = P_D(y'_1, ..., y'_\kappa)$. Let $\Pi' = \pi(y'_1, ..., y'_\kappa)$. As the functions s_j are κ-symmetric polynomials, we get[16] $s_j(y'_1, ..., y'_\kappa) = s_j(y_1, ..., y_\kappa)$ for all $j = 1, ..., m$. As the variables y_{li} involved[17] in π are i.i.d. according to the uniform distribution over $\mathbb{Z}_n$, the probability that $\Pi \equiv \Pi' \mod q$ is negligible (because it was assumed that $\pi(y_1, y_2, ..., y_\kappa) \neq \pi(y_2, y_1, ..., y_\kappa)$) and the probability that $\Pi_B = \Pi$ is equal to the probability that $\Pi_B = \Pi'$. As B is assumed to solve Problem 1 (resp. Problem 2), $\Pi_B = \Pi$ with non negligible probability. It implies that $\Pi_B = \Pi'$ with non negligible probability. As $\Pi \equiv \Pi' \mod p$ and $\Pi \not\equiv \Pi' \mod q$, we have $p = \gcd(n, \Pi - \Pi')$. It implies that A terminates (when $\Pi_B = \Pi'$) in polynomial-time.
□

B Proof of Proposition 4

First of all, by arguing similarly to proposition 2, one shows that each component is an efficiently valuable polynomial defined over the tuples $y_1, ..., y_\kappa$. Now, let us focus on κ-symmetry. There is an implicit canonical function between the κ tuples y_l and the invertible matrices $(S_z)_{z=1,...,\vartheta}$ and e. The subscript $y_1, ..., y_\kappa$ is added to precise the tuples which are considered: for instance, $e_{y_1,...,y_\kappa} = (c_{z,y_1,...,y_\kappa})_{z=1,...,\vartheta}$

[15] Because of these constraints are κ-symmetric.
[16] It is not true in general, i.e. for arbitrary κ-symmetric functions s_j.
[17] According to Problem 1 (resp. Problem 2), $i \in I_F$.

is the encryption related to the tuples $y_1, ..., y_\kappa$. Let us show that $c_{z,y_1,...,y_\kappa}$ is a κ-symmetric function defined over the tuples y_l. By definition

$$c_{z,y_1,...,y_\kappa} = S^{-1}_{z,y_1,...,y_\kappa} v_{z,y_1,...,y_\kappa}$$

with $v_{z,y_1,...,y_\kappa} = (\mathrm{B}_{z11} x_{z1}, ...)$. Let σ be an arbitrary permutation of $\{1, ..., \kappa\}$. Then, we define the permutation β over $\{1, ..., \kappa\tau\delta\}$ as follows

$$\beta(i) = \left(\sigma\left(\left\lceil \frac{i}{\tau\delta} \right\rceil\right) - 1\right)\tau\delta + (i - 1 \mod \tau\delta) + 1$$

Let $T_z = [t_{zij}]$ be the (invertible) matrix defined by $t_{zi} = s_{z\beta(i)}$[18] and $w_{z,y_1,...,y_\kappa} = \beta(v_{z,y_1,...,y_\kappa})$[19]. By arguing similarly to the proof of Proposition 2,

$$c_{z,y_1,...,y_\kappa} = S^{-1}_{z,y_1,...,y_\kappa} v_{z,y_1,...,y_\kappa} = T^{-1}_{z,y_1,...,y_\kappa} w_{z,y_1,...,y_\kappa}$$

Clearly $T_{z,y_1,...,y_\kappa} = S_{z,y_{\sigma(1)},...,y_{\sigma(\kappa)}}$ and $w_{z,y_1,...,y_\kappa} = v_{z,y_{\sigma(1)},...,y_{\sigma(\kappa)}}$ implying that

$$c_{z,y_1,...,y_\kappa} = S^{-1}_{z,y_1,...,y_\kappa} v_{z,y_1,...,y_\kappa} = T^{-1}_{z,y_1,...,y_\kappa} w_{z,y_1,...,y_\kappa} = S^{-1}_{z,y_{\sigma(1)},...,y_{\sigma(\kappa)}} v_{z,y_{\sigma(1)},...,y_{\sigma(\kappa)}} = c_{z,y_{\sigma(1)},...,y_{\sigma(\kappa)}}$$

□

C Proof of Proposition 9

The tuples y_l^{sk} are generated according to a probability distribution statistically indistinguishable with the probability distribution considered in Problem 2 (by choosing the coefficients of S_u at random, S_u is not invertible with negligible probability): the sets $I_j^\times$ (see notation of Problem 2) contain the δ components of the basic vectors B_{vzlt} randomly generated in Encrypt1 to encrypt x_v and the sets I_j^+ are the sets $\{x_{v1l}, ..., x_{v\vartheta l}\}_{(v,l)\in\{1,...,m\}\times\{l=1,...,\kappa\}}$ satisfying $x_{v1l} + ... + x_{v\vartheta l} = x_v$.

Proposition 4 and Proposition 6 ensure that all public values can be polynomially computed only knowing κ-symmetric efficiently valuable polynomials defined over the tuple y_l^{sk}. Thus, assuming hardness of factorization, Proposition 1 ensures that it is not possible to recover any non κ-symmetric product defined over the coefficients s_{uij}.

Let $\gamma \in \mathbb{N}^*$ such that γ is not a multiple of κ and ϕ be an element of $\mathsf{SP}^\gamma \cup \mathsf{SP}_l$. Let $w_1^* = ... = w_r^* = (1, 0, 0, ...)$. R_ϕ allows to efficiently compute $\pi = \phi(w_1^*, ..., w_r^*)$ which is a product of values $s_{.,.,1}$. As γ is not a multiple of κ, π is a non κ-symmetric product (efficiently valuable) of values belonging to $\{s_{ui1} | u = 1, ..., \vartheta^2(2\delta + 2) - \vartheta; i = 1, ..., 3\kappa\delta\}$. Thus, according to Proposition 1, π cannot be recovered implying that R_ϕ cannot be recovered.

□

D Proof of Proposition 6

First of all, by arguing similarly to proposition 2, one shows that each monomial coefficient is an efficiently valuable polynomial defined over the tuples $y_1, ..., y_\kappa$. Now, let us focus on κ-symmetry. We consider notation and conventions adopted in the proof of Proposition 4. Let $x_u x'_v$ be a monomial. We denote by α_i (resp. a_i) the coefficient of this monomial in p_i (resp. q_i). Let $a = (a_1, ..., a_\kappa)$ and $\alpha = (\alpha_1, ..., \alpha_\kappa)$. By definition of the operator $\mathcal{Q}$,

$$a_{y_1,...,y_\kappa} = S^{-1}_{y_1,...,y_\kappa} \alpha_{y_1,...,y_\kappa}$$

[18] t_{zi} and s_{zi} refer to the i^{th} row of respectively T and S

[19] The components of v are permuted according to β.

Given a permutation σ of $\{1,...,\kappa\}$, β is the permutation of $\{1,...,\kappa m\}$ derived from σ as done in the proof of Proposition 4 (where $\tau\delta$ is replaced by m), and T is the matrix where the rows of S are permuted with β. It follows that

$$a_{y_1,...,y_\kappa} = S^{-1}_{y_1,...,y_\kappa}\alpha_{y_1,...,y_\kappa} = T^{-1}_{y_1,...,y_\kappa}\beta(\alpha_{y_1,...,y_\kappa})$$

Clearly $T = S_{y_{\sigma(1)},...,y_{\sigma(\kappa)}}$ and because of the constraints ($u'_{i+lm,j} = u'_{ij}+lm$, $u''_{i+lm,j} = u''_{ij}+lm$, $\alpha_{i+lm} = \alpha_i$) introduced in the definition of κ-symmetric operators $\mathcal{Q}$,

$$\beta(\alpha_{y_1,...,y_\kappa}) = \alpha_{y_{\sigma(1)},...,y_{\sigma(\kappa)}}$$

it follows that

$$a_{y_1,...,y_\kappa} = S^{-1}_{y_1,...,y_\kappa}\alpha_{y_1,...,y_\kappa} = S^{-1}_{y_{\sigma(1)},...,y_{\sigma(\kappa)}}\alpha_{y_{\sigma(1)},...,y_{\sigma(\kappa)}} = a_{y_{\sigma(1)},...,y_{\sigma(\kappa)}}$$

□

E An instantiation of Add

Here, we propose a deterministic construction of the operator Add. We let the reader check that it is an instantiation of the construction proposed in the proof of Proposition 7.

1. Just replace by $\mathrm{B} = \mathrm{A}_{11} \star \mathrm{A}'_{12}$, $\mathrm{C} = \mathrm{A}_{12} \star \mathrm{A}'_{11}$ and $\mathrm{E} = \mathrm{A}_{13} \star \mathrm{A}'_{13}$ by respectively $\mathrm{B} = \mathrm{A}_{11} \times \mathrm{A}'_{12}$, $\mathrm{C} = \mathrm{A}_{12} \times \mathrm{A}'_{11}$ and $\mathrm{E} = \mathrm{A}_{13} \times \mathrm{A}'_{13}$.
2. Let $u \in \{2,...,\delta\}$. we define $z_u = (z_{ui})_{i=1,...,3\delta}$ by

$$z_{ui} = \begin{cases} x_1 e_1^{\lfloor u/2 \rfloor} b_1 ... b_{\lfloor u/2 \rfloor + (u \mod 2)} & \text{if } i = 1 \\ e_1 ... e_u & \text{if } i = 2,...,\delta \\ x'_1 e_1^{\lfloor u/2 \rfloor} c_1 ... c_{\lfloor u/2 \rfloor + (u \mod 2)} & \text{if } i = \delta+1 \\ e_1 .... e_u & \text{if } i = \delta+1,...,3\delta \end{cases}$$

Let $u \in \{\delta+1,...,2\delta\}$. we define $z_u = (z_{ui})_{i=1,...,3\delta}$ by

$$z_{ui} = \begin{cases} x_1 e_1^{\lfloor u/2 \rfloor} b_1 ... b_{\lfloor u/2 \rfloor + (u \mod 2)} & \text{if } i = 1 \\ e_1 ... e_\delta e_i^{u-\delta} & \text{if } i = 2,...,\delta \\ x'_1 e_1^{\lfloor u/2 \rfloor} c_1 ... c_{\lfloor u/2 \rfloor + (u \mod 2)} & \text{if } i = \delta+1 \\ e_1 .... e_\delta e_1 ... e_{u-\delta} & \text{if } i = \delta+1,...,3\delta \end{cases}$$

Each component of z_u is a product of u components of $|w_1|_{T_1}$ and as $b_1...b_\delta = c_1...c_\delta = e_1...e_\delta = 1$, we have

$$z_{2\delta} = (e_1^\delta x_1, e_2^\delta, ..., e_\delta^\delta, e_1^\delta x'_1, 1, ..., 1)$$

By using κ-symmetric operators, we build the sequence $w_2, ..., w_{2\delta}$

$$w_u = Q_{T_u \leftarrow (T_{u-1}, T_1)}(w_{u-1}, w_1)$$

such that for all $u = 2, ..., 2\delta$, the 3δ first components of $|w_u|$ are equal to z_u: the other ones being deduced by using κ-symmetry.

3. Just replace $\mathrm{H} = \mathrm{F} \star \mathrm{E}$, $\mathrm{I} = \mathrm{G} \star \mathrm{E}$, $\mathrm{J} = \mathrm{G} \star \mathrm{E}$ by respectively $\mathrm{H} = \mathrm{F} \times \mathrm{E}$, $\mathrm{I} = \mathrm{G} \times \mathrm{E}$, $\mathrm{J} = \mathrm{G} \times \mathrm{E}$.

Proposition 10. *Let us adopt notation of Section 8.2. If there exists two ϕ_1 and ϕ_2 belonging to $\mathsf{SP}_{\mathsf{Add}}$ such that for all w, w' satisfying constraints of Definition 4 we have $z\phi_1(w, w', \mathit{Add}(w, w')) = \phi_2(w, w', \mathit{Add}(w, w'))$ then*

$$\deg(\phi_1) + \deg(\phi_2) \geq \delta/2$$

Proof. Because of Lemma 1, we fix $\kappa = 1$. Given a set $E \subseteq \mathbb{Z}_n$ and $I \subseteq \mathbb{N}$, E_I denotes the set defined by

$$E_I = \left\{ \prod_{x \in E} x^{i_x} \mid i_x \in \mathbb{Z} \; s.t. \sum_{x \in E} |i_x| \in I \right\}$$

According to notation of Definition 7, w is defined by $|w| = (\mathrm{A}_{11}x_1, \mathrm{A}_{12}, \mathrm{A}_{13})$. Let us denote the vector $\mathrm{A} = (\mathrm{A}_{11}, \mathrm{A}_{12}, \mathrm{A}_{13})$ by $\mathrm{A} = (a_1, ..., a_{3\delta})$. Moreover, we state $|w'| = |w|$ implying that $\mathrm{B} = \mathrm{C}$.

By definition, for any $\phi \in \mathsf{SP}_{\mathsf{Add}}$, $\phi(w, w', \mathsf{Add}(w, w'))$ is equal to the product of $\deg(\phi)$ components of the $|w_u|$ for $u = 0, ..., 2\delta + 1$ (with $|w_0| = |w| = |w'|$). Let us list and categorize these components (Because of κ-symmetry, we only consider the 3δ first components of each $|w_u|$):

- Some components belong to the set

$$X = \{a_1 x_1, 2e_1^{\delta+1} x_1, b_1 x_1, b_1 e_1 x_1, b_1 e_1 b_2 x_1, ..., b_1 ... b_\delta e_1^\delta x_1 = e_1^\delta x_1\}$$

- The other components belong to the set

$$Y = \{a_2, ..., a_{3\delta}, b_2, ..., b_\delta, e_1, ..., e_\delta, \prod_{i=1}^{t} e_i, e_j^u \mid t = 1, ..., \delta; j = 2, ..., \delta; u = 2, ..., \delta + 1\}$$

According to basic vectors constraints, $a_1 = (a_2 ... a_\delta)^{-1}$ and $b_1 = (b_2 .... b_\delta)^{-1}$. Consequently,

$$X = \left\{ \frac{x_1}{a_2 ... a_\delta}, 2e_1^\delta x_1, \frac{e_1^{\lfloor u/2 \rfloor} x_1}{b_{\lfloor u/2 \rfloor + (u \mod 2) + 1} ... b_\delta} \mid u = 1, ..., 2\delta \right\}$$

Let $\Omega = \{a_2, ..., a_\delta, b_2, ..., b_\delta, e_1\}$. Clearly, each component of X belongs to $\Omega_{\{\delta-1,\delta\}} x_1$. Let us consider two products (i.e. polynomials) $\phi_1 = \pi_1 \pi'_1$ and $\phi_2 = \pi_2 \pi'_2$ of elements of $X \cup Z$ such that

$$\phi_1 / \phi_2 = x_1 (= z)$$

where π_1, π_2 are products of respectively m_1, m_2 elements of X and π'_1, π'_2 are products of respectively n_1, n_2 elements of Y with m_1, m_2, n_1, n_2 positive integers s.t. $k = m_1 + m_2 \leq \delta$. Note that

$$\deg \phi_1 + \deg \phi_2 = m_1 + m_2 + n_1 + n_2$$

The constraint $\pi_1 \pi'_1 / \pi_2 \pi'_2 = x_1$ implies that $m_1 = m_2 + 1$. It follows that $\pi_1 \in x_1^{m_1} \Omega_{k_1 \geq m_1(\delta-1)}$ and $\pi_2 \in x_1^{m_1 - 1} \Omega_{k_2 \leq (m_1 - 1)\delta}$ implying that

$$\pi'_2 / \pi'_1 \in \Omega_{k_0 \geq k_1 - k_2 \geq \delta - k} \tag{3}$$

Recall π'_1 and π'_2 are products of respectively n_1 and n_2 of elements of Y. Thus, without loss of generality, it can be assumed that $\pi'_1 / \pi'_2 \in Y_{n_1 + n_2}$. Given $t \in \mathbb{N}^*$, we can easily show that $\pi \in \Omega_t \Rightarrow \pi \notin Y_{k < t/2}$ (because the only possible simplifications are $b_i / a_i = a_{i+\delta}$ for any $i = 1, ..., \delta$). It implies that $n_1 + n_2 \geq k_0/2 \geq \frac{\delta - k}{2}$ implying that

$$n_1 + n_2 + m_1 + m_2 \geq k + \frac{\delta - k}{2} \geq \delta/2$$

implying that $\deg \phi_1 + \deg \phi_2 \geq \delta/2$.

□

F A weak version of Proposition 3

Proposition 11. *Let λ be a security parameter, $(pk, sk) \leftarrow \mathsf{KeyGen}(\lambda)$ and $\gamma \in \mathbb{N}^*$ such that γ is not a multiple of δ ($|\gamma|$ polynomial in λ). Let $\phi \in \mathsf{SP}^\gamma$ and R_ϕ be an effective representation of ϕ. By assuming the hardness of factorization, recovering R_ϕ only given $\mathcal{Q}_S$ is difficult.*

Proof. Let $x_1, ..., x_\delta$ be randomly chosen in Z_n and $\alpha_{\delta-1}, ..., \alpha_0$ be the monomial coefficients of the polynomial $p(x) = (x - x_1)...(x - x_\delta)$, i.e. $p(x) = x^\delta + \alpha_{\delta-1}x^{\delta-1} + ... + \alpha_0$. The aim of this proof consists of building $\mathcal{Q}_S$ according to a distribution statistically indistinguishable from $\mathsf{QGen}(K \leftarrow \mathsf{KeyGen}(\lambda))$ such that the knowledge of $R_\phi \Rightarrow$ the knowledge of a non-symmetric product of $x_1, ..., x_\delta$ which is difficult assuming Proposition 1. This following construction is polynomial and can be decomposed in 2 steps.

Step 1. This step consists of generating a matrix M at random an in polynomial time such that $x_1, ..., x_\delta$ are eigenvalues of M. Let us start by considering the case $\delta = 2$, i.e. $p(x) = x^2+\alpha_1 x+\alpha_0$. The characteristic polynomial of M is $r(x) = (a_{11} - x)(a_{22} - x) - a_{12}a_{21}$. The values a_{ij} can be chosen in polynomial time such that $r = p$, i.e. $a_{12}a_{21} = \alpha_0$ and $a_{11} + a_{22} = -\alpha_1$. Indeed, it suffices to choose at random a_{12} and a_{11} in $\mathbb{Z}_n$ and then to compute $a_{21} = \alpha_0 a_{12}^{-1}$ and $a_{22} = -(\alpha_1 + a_{11})$. For $\delta > 2$, it suffices to randomly choose a_{ij} for $j > 1$ and to adjust the coefficients a_{i1} to ensure $r = p$ by solving a linear system.

Let $s_1, ..., s_\delta$ be the eigenvectors of M associated to the eigenvalues $x_1, ..., x_\delta$ such that $s_{11} = x_1, ..., s_{\delta 1} = x_\delta$. Let S be the matrix such that its i^{th} row is equal to s_i. Clearly S is distributed as specified in KeyGen0, i.e. at random according to the uniform distribution among invertible matrices (the probability that S is not invertible is negligible). In the following step, we build $\mathcal{Q}_S$ only given M (and without knowing S).

Step 2. For sake of simplicity, let us detail the construction for $\delta = 2$. The extension to the case $\delta > 2$ is straightforward and will be explained later. The challenge consists of building $\mathcal{Q}_S = (q_1, q_2)$ only knowing M (in particular, without knowing S). By writing the polynomials q_1 and q_2 as:

- $q_1(w, w') = a_1 x_1 x'_1 + a_2(x_1 x'_2 + x'_1 x_2) + a_3 x_2 x'_2$
- $q_2(w, w') = b_1 x_1 x'_1 + b_2(x_1 x'_2 + x'_1 x_2) + b_3 x_2 x'_2$

and by definition of these polynomials, for all $w, w' \in \mathbb{Z}_n^\delta$ and $i \in \{1, 2\}$, we have

$$\begin{aligned} & s_i(q_1(w, w'), q_2(w, w')) = (s_i w).(s_i w') \\ \Leftrightarrow & (s_{i1}a_1 + s_{i2}b_1)x_1 x'_1 + (s_{i1}a_2 + s_{i2}b_2)(x_1 x'_2 + x'_1 x_2) + (s_{i1}a_3 + s_{i2}b_3)x_2 x'_2 \\ & \qquad = s_{i1}^2 x_1 x'_1 + s_{i1}s_{i2}(x_1 x'_2 + x'_1 x_2) + s_{i2}^2 x_2 x'_2 \end{aligned}$$

giving the following equalities

$$\begin{cases} a_1 s_{i1} + b_1 s_{i2} = s_{i1}^2 \\ a_2 s_{i1} + b_2 s_{i2} = s_{i1}s_{i2} \\ a_3 s_{i1} + b_3 s_{i2} = s_{i2}^2 \end{cases}$$

where $i \in \{1, 2\}$. First, we can remark that the vectors s_1 and s_2 are eigenvectors of the matrix

$$\begin{bmatrix} a_1, b_1 \\ a_2, b_2 \end{bmatrix}$$

with associated eigenvalues $\lambda_1 = s_{11}$ and $\lambda_2 = s_{21}$. Thus, this matrix is equal to M, i.e. $a_1 = m_{11}, a_2 = m_{21}, b_1 = m_{21}, b_2 = m_{22}$. Let us see how to recover a_3 and b_3 in order to finish the construction of q_1 and q_2. It is achieved by noting that the vectors s_1 and s_2 are also eigenvectors of the matrix

$$A = \begin{bmatrix} a_2, b_2 \\ a_3, b_3 \end{bmatrix}$$

For any $x, y \in \mathbb{Z}_n$ s_1 and s_2 are eigenvectors of $T_{xy} = xI + yM$. To get the values (a_3, b_3), it suffices to adjust $x, y \in \mathbb{Z}_n$ in order that the first row of $T_{xy} = xI + yM$ is equal to (a_2, b_2). Let $T = [t_{ij}]$ be this matrix. Thus, T and A have the same eigenvectors with the same associated eigenvalues. It follows that

$$A = T$$

implying that $a_3 = t_{21}$ and $b_3 = t_{22}$ finishing the construction of the polynomials q_1, q_2 only given M. More generally, for $\delta > 2$, we proceed in the same way by noticing that the matrices $I, M, M^2, ..., M^{\delta-1}$ are linearly independent because of Cayley-Hamilton theorem (the characteristic polynomial and the minimal polynomial have the same roots implying that the degree of the minimal polynomial is at least δ with non negligible probability).

To conclude. Assuming p is chosen at random, M is a matrix chosen at random such that its eigenvalues are equal to the roots of p. S is defined as (but not built) the matrix whose the rows are the eigenvectors of M with $s_{i1} = x_i$. We have shown that $\mathcal{Q}_S$ can be built in polynomial-time only given M. Let $w_1^* = ... = w_r^* = (1, 0, 0, ...)$. R_ϕ allows to efficiently compute $\pi = \phi(w_1^*, ..., w_r^*)$ which is a non-symmetric product of roots of p. Consequently, according to Proposition 1, the existence of such an attacker is not possible assuming the hardness of factorization.

□

G κ-symmetric representations of polynomials

Let $t = \Theta(\lambda)$, $a = (a_1, ..., a_t)$ and $b = (b_1, ..., b_t)$ be two tuples of $\mathbb{Z}_n^t$ and $\phi : \mathbb{Z}_n^t \to \mathbb{Z}_n$ the polynomial defined by

$$\phi(x_1, ..., x_t) = (a_1x_1 + ... + a_tx_t)(b_1x_1 + ... + b_tx_t)$$

The $m = t(t+1)/2$ monomial coefficients m_{ij} of ϕ are 2-symmetric values defined over (a, b), i.e. $m_{ii} = a_ib_i$ and $m_{ij,i\leq j} = a_ib_j + a_jb_i$. Thus, the expanded representation of ϕ, i.e.

$$R_\phi(x) = \sum_{(i,j)\in\{1,...,t\}^2, i\leq j} m_{ij}x_ix_j$$

is a 2-symmetric representation of ϕ.

Here, we wonder whether there is a more efficient (in term of storage for instance) representation R_ϕ of ϕ only using 2-symmetric values. Clearly, R_ϕ allows to polynomially compute all the monomial coefficients m_{ij} (for instance $m_{11} = \phi(1, 0, 0, ...)$). Thus, the existence of such a representation R_ϕ implies the existence of a set E containing $m' < m$ 2-symmetric values (defined over (a, b)) allowing to polynomially compute (without knowing the factorization of n) all the monomial coefficients m_{ij}. We did not manage to solve this challenge consisting of finding such a set E (which is easier than finding a more efficient representation R_ϕ only using 2-symmetric values). For instance, in the case $t = 4$, the challenge consists of finding a set E' of strictly less than 10 2-symmetric values allowing to polynomially recover the 10 values $m_{ij,i\leq j}$, i.e. a_1b_1, a_2b_2, $a_1b_2 + a_2b_1$, a_3b_3, $a_1b_3 + a_3b_1$,...
Empirical searches do not allow us to succeed this challenge. Authors are convinced that such sets E' do not exist while they are unable to formally prove it.

H Pre-processing (randomization) vectors input in Add

Let $\rho \in \mathbb{N}$ be a parameter indexed by λ. Let T and T' be two given invertible matrices and let w be a vector such that $|w|_T = (\mathrm{A}_1x_1, \mathrm{A}_2, \mathrm{A}_3)$ (see notation of Definition 7. In order to simplify notation, we fix $\kappa = 1$). The operator Rand computes the vector $\mathsf{Rand}_{T'\leftarrow T}(w)$ defined by

$$|\mathsf{Rand}_{T'\leftarrow T}(w)|_{T'} = (\mathrm{A}_{\rho,1}x_1, \mathrm{A}_{\rho,2}, \mathrm{A}_{\rho,3})$$

where the basic vectors A_{ji} are defined by the following recursive sequence $\mathrm{A}_{11} = \mathrm{A}_1, \mathrm{A}_{12} = \mathrm{A}_2, \mathrm{A}_{13} = \mathrm{A}_3$ and for $j = 1...\rho$

$$\begin{cases} \mathrm{A}_{j1} = \mathrm{A}_{j-1,1} \star \mathrm{A}_{j-1,3} \\ \mathrm{A}_{j2} = \mathrm{A}_{j-1,2} \star \mathrm{A}_{j-1,3} \\ \mathrm{A}_{j3} = \mathrm{A}_{j-1,3} \star \mathrm{A}_{j-1,3} \end{cases}$$

Let $T_2, ..., T_{\rho-1}$ be $\rho - 2$ invertible matrices chosen at random and $T_\rho = T'$. Similarly to the operator Add, the vector $\mathsf{Rand}_{T' \leftarrow T}(w) = w_\rho$ can be computed by a recursive sequence where $w_1 = w$ and

$$w_j = \mathcal{Q}_{T_j \leftarrow (T_{j-1}, T_{j-1})}(w_{j-1}, w_{j-1})$$

where $|w_j|_{T_j} = (\mathrm{A}_{j1}x_1, \mathrm{A}_{j2}, \mathrm{A}_{j3})$.

Analysis. The number of possible operator Rand is exponential in ρ, i.e. $\Omega(2^\rho)$. Thus, one can assume that the vectors $\mathrm{A}_{\rho i}$ and the vectors A_{1i} are pseudo-independent provided $\rho = \Theta(\lambda)$. Clearly, Rand does not provide new linearization attacks provided the vectors $\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3$ are randomly and independently generated. Each vector input in Add can be *randomized* with Rand. This can be done in order to remove possible interactions between operators Add of *pk*.

I Proof of Lemma 1

Let $\phi \in \mathsf{SP}_{\mathsf{Add}}$, i.e. $\phi(w_{-1}, ..., w_{2\delta+1}) = \prod_{r=1}^{\delta} t_{u_r i_r} w_{u_r}$ and $\Lambda : \mathsf{SP}_{\mathsf{Add}} \to \mathsf{SP}_{\mathsf{Add}} \cap \mathsf{SP}_1$ such that $\phi' = \Lambda(\phi)$ is defined by $\phi'(w_{-1}, w_0, ..., w_{2\delta+1}) = \prod_{r=1}^{\delta} t_{u_r, i_r \bmod 3\delta} w_{u_r}$. Clearly ϕ and ϕ' have the same degree.

Let us assume that there exists two polynomials ϕ_1, ϕ_2 satisfying (2) and let us consider two vectors w and w' such that $|w|_T = (\mathrm{A}_1x_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{A}_1x_1, \mathrm{A}_2, \mathrm{A}_3...)$ and $|w|_{T'} = (\mathrm{A}'_1y_1, \mathrm{A}'_2, \mathrm{A}'_3, \mathrm{A}'_1y_1, \mathrm{A}'_2, \mathrm{A}'_3...)$. As all the operator $\mathcal{Q}$ involved in Add are κ-symmetric, $\phi_1(w, w', \mathsf{Add}(w, w')) = \phi'_1(w, w', \mathsf{Add}(w, w'))$ and $\phi_2(w, w', \mathsf{Add}(w, w')) = \phi'_2(w, w', \mathsf{Add}(w, w'))$ if $\phi'_1 = \Lambda(\phi_1)$ and $\phi'_2 = \Lambda(\phi_2)$. It implies that

$$\phi'_1(w, w', \mathsf{Add}(w, w')) = z'\phi'_2(w, w', \mathsf{Add}(w, w'))$$

with z' being a linear combination of x_1, y_1. As ϕ'_1 and ϕ'_2 only deals with the 3δ first components of $|w|_T$ and $|w'|_{T'}$, the previous relation remains true for all vectors w and w' satisfying constraints of Definition 7 implying that ϕ'_1 and ϕ'_2 satisfy (2).

□

J Toy implementation of the additive homomorphic scheme

In this section, we provide an example of the implementation of the homomorphic scheme for $\delta = 2$. Given $S := \begin{bmatrix} s_{1,1} & s_{1,2} \\ s_{2,1} & s_{2,2} \end{bmatrix}$

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

$$\mathcal{Q}_S(x, y) = \Delta^{-1} \begin{bmatrix} (s_{22}s_{11}^2 - s_{12}s_{21}^2)x_1y_1 + (s_{22}s_{11}s_{12} - s_{12}s_{21}s_{22})(x_1y_2 + x_2y_1) + (s_{22}s_{12}^2 - s_{12}s_{22}^2)x_2y_2 \\ (s_{11}s_{21}^2 - s_{21}s_{11}^2)x_1y_1 + (s_{11}s_{21}s_{22} - s_{21}s_{11}s_{12})(x_1y_2 + x_2y_1) + (s_{11}s_{22}^2 - s_{21}s_{12}^2)x_2y_2 \end{bmatrix}$$

Numerical application.

- $n = 7 * 5 = 35$
- $g = 2$
- $S = \begin{bmatrix} 3 & 8 \\ 2 & 4 \end{bmatrix}$, $S^{-1} = \begin{bmatrix} 34 & 2 \\ 18 & 8 \end{bmatrix}$
- $\Phi_S \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 6x_1^2 + 28x_1x_2 + 32x_2^2$
- $e_1 = \begin{pmatrix} 13 \\ 9 \end{pmatrix} \leftarrow \mathsf{Encrypt}(x_1 = -3)$
- $e_2 = \begin{pmatrix} 11 \\ 1 \end{pmatrix} \leftarrow \mathsf{Encrypt}(x_2 = 4)$
- $e_3 = \begin{pmatrix} 17 \\ 22 \end{pmatrix} \leftarrow \mathsf{Encrypt}(x_3 = -2)$
- $\mathcal{Q}_S \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right) = S^{-1} \begin{pmatrix} 9x_1y_1 + 24(x_1y_2 + x_2y_1) + 29x_2y_2 \\ 4x_1y_1 + 8(x_1y_2 + x_2y_1) + 16x_2y_2 \end{pmatrix}$
  $= \begin{pmatrix} 34x_1y_1 + 27(x_1y_2 + x_2y_1) + 3x_2y_2 \\ 19x_1y_1 + 6(x_1y_2 + x_2y_1) + 20x_2y_2 \end{pmatrix}$

Verification of the homomorphic operator:

- $e_1 \oplus e_2 = Q_S(e_1, e_2) = \begin{pmatrix} 3 \\ 34 \end{pmatrix}$
- $e_2 \oplus e_3 = Q_S(e_2, e_3) = \begin{pmatrix} 12 \\ 17 \end{pmatrix}$
- $e_1 \oplus e_3 = Q_S(e_1, e_3) = \begin{pmatrix} 32 \\ 4 \end{pmatrix}$
- $\mathsf{Decrypt}(e_1 \oplus e_2) = \mathsf{DL}_{g=2} \left(\Phi_S \begin{pmatrix} 3 \\ 34 \end{pmatrix} = 2 \right) = 1 = x_1 + x_2$
- $\mathsf{Decrypt}(e_2 \oplus e_3) = \mathsf{DL}_{g=2} \left(\Phi_S \begin{pmatrix} 12 \\ 17 \end{pmatrix} = 4 \right) = 2 = x_2 + x_3$
- $\mathsf{Decrypt}(e_1 \oplus e_3) = \mathsf{DL}_{g=2} \left(\Phi_S \begin{pmatrix} 32 \\ 4 \end{pmatrix} = 23 \right) = -5 = x_1 + x_3$

K Randomization of operators $\mathcal{Q}$

In this section, we present ways to randomize operators $\mathcal{Q}$. For sake of simplicity, we focus on the additive homomorphic encryption scheme (the extension to the FHE is straightforward). Let $\delta' > 0$ and S be an invertible matrix of $\mathbb{Z}_n^{(\delta+\delta')\times(\delta+\delta')}$.

K.1 First method

To generate public encryptions $e_v \in \mathbb{Z}_n^{\delta+\delta'}$ of x_v, δ values $r_i \in \mathbb{Z}_n^*$ such that $r_1, ..., r_\delta = g^{x_v}$ are randomly chosen and $e_v = S^{-1}\,(r_1, ..., r_\delta, 0, ..., 0)$ $(\Phi_S(w) = \prod_{i=1}^{\delta} s_i w)$.

Let E be the set of all linear combination of the vectors $s_{\delta+1}, ..., s_{\delta+\delta'}$. By construction, for any $u \in E$, $ue_v = 0$. Let F be the set of (2-degree) polynomials z defined by $z(w, w') = uw \times r'w' + rw \times u'w'$ where $u, u' \in E$ and $r, r' \in \mathbb{Z}_n^{\delta+\delta'}$ are arbitrary vectors. By construction, for any $z \in F$ and any public encryptions $e_v, e_{v'}$,

$$z(e_v, e_{v'}) = 0$$

Let $\mathcal{Q}_S = (q_1, ..., q_{\delta+\delta'}) \leftarrow \mathsf{Qgen}(S)$ and $z_1, ..., z_{\delta+\delta'}$ be randomly chosen in F. By construction, it is ensured that the operator $\mathcal{Q}_S^{\mathsf{rand}} = (q_1 + z_1, ..., q_{\delta+\delta'} + z_{\delta+\delta'})$ satisfies for any encryptions e, e'

$$\mathcal{Q}_S^{\mathsf{rand}}(e, e') = \mathcal{Q}_S(e, e')$$

K.2 Second Method

To generate public encryptions $e_v \in \mathbb{Z}_n^{\delta+\delta'}$ of x_v, one picks up at random $\delta + \delta'$ values $r_i \in \mathbb{Z}_n^*$ such that $r_1, ..., r_\delta = g^{x_v}$, $e_v = S^{-1}(r_1, ..., r_\delta, r_{\delta+1}, ..., r_{\delta+\delta'})$ $(\Phi_S(w) = \prod_{i=1}^{\delta} s_i w)$.

Let $p_i : \mathbb{Z}_n^{\delta+\delta'} \times \mathbb{Z}_n^{\delta+\delta'} \to \mathbb{Z}_n$ be δ' 2-degree polynomials chosen at random. The operator $\mathcal{Q}_S : \mathbb{Z}_n^{\delta+\delta'} \times \mathbb{Z}_n^{\delta+\delta'} \to \mathbb{Z}_n^{\delta+\delta'}$ is defined by

$$\mathcal{Q}_S(w', w'') = \begin{pmatrix} q_1(w', w'') \\ \ldots \\ q_{\delta+\delta'}(w', w'') \end{pmatrix} = S^{-1} \begin{pmatrix} s_1 w' \times s_1 w'' \\ \ldots \\ s_\delta w' \times s_\delta w'' \\ p_1(w', w'') \\ \ldots \\ p_{\delta'}(w', w'') \end{pmatrix}$$