Arithmetic Operations in the Polynomial Modular Number System

Jean-Claude Bajard, Laurent Imbert, Thomas Plantard

Arithmetic Operations in the Polynomial Modular Number System

Jean-Claude Bajard[1], Laurent Imbert[1,2], and Thomas Plantard[1]

[1] LIRMM, CNRS UMR 5506
161 rue Ada, 34392 Montpellier cedex 5, France

[2] ATIPS, CISaC, University of Calgary
2500 University drive NW, Calgary, AB, T2N 1N4, Canada

Research Report LIRMM-04030

September 2004

Abstract

We propose a new number representation, and the corresponding arithmetic, for the elements of the ring of integers modulo p. The so-called Polynomial Modular Number System (PMNS) allows for fast polynomial arithmetic and easy parallelization. The most important contribution of this paper is certainly the fundamental theorem of a Modular Number System which gives up an upper-bound on the coefficients of the polynomials used to represent the set of non-negative integers $\{0, \ldots, p-1\}$. However, we also propose a complete set of algorithms for the arithmetic operations over a PMNS, which make this system of practical interest for people concerned about fast implementation of modular arithmetic.

Keywords: Number system, Modular arithmetic, Lattice theory, Table-based methods

1 Introduction

Efficient implementation of modular arithmetic is an important prerequisite in today's public-key cryptography [1]. The famous RSA algorithm [2], and the cryptosystems based on the discrete logarithm problem, such as Diffie-Hellman key exchange [3], need fast arithmetic modulo integers of size 1024 to roughly 15000 bits. For the same level of security, elliptic curves defined over prime fields, require operations modulo prime numbers whose size range approximately from 160 to 500 bits [4].

Classical implementations use multiprecision arithmetic, where long integers are represented in a predefined high-radix (usually a power of two depending on the word size of the targeted architecture). Arithmetic operations, namely modular reduction and multiplication, are performed using efficient algorithms, such as as Montgomery [5], or Barrett [6]. (For more details, see [1], chapter 14.) Those algorithms do not require the modulus to be of special form. When this is the case however, modular multiplication and reduction can be accelerated considerably. Mersenne numbers of the form $2^m - 1$ are the most common examples. Pseudo-Mersenne numbers [7], generalized Mersenne numbers [8], and their extension [9] are other examples of numbers allowing fast modular arithmetic.

In a recent paper [10], we defined the so-called Modular Number Systems (MNS) and Adapted Modular Number Systems (AMNS) to speed up the arithmetic operations for moduli which do not belong to any of the previous classes. In this paper, we propose a new representation, and the corresponding arithmetic, for the ring of integers modulo p (the integer p does not have to be a prime, although it is likely to be for practical cryptographic applications). We define the Polynomial Modular Number System (PMNS), into which integers are represented as polynomials. Over the classical binary representation, polynomial arithmetic offers the advantages of no carry propagation and easiest parallelization.

The main contribution of this paper is the fundamental of an MNS, which gives an upper bound on the coefficients of the polynomials used to represent the elements of $\mathbb{Z}_p$. This theorem is presented in Section 4. It uses results from lattice reduction theory [11, 12] that are briefly recalled in Section 3. The second half of the paper focuses on the arithmetic operations. In Section 5, we propose algorithms for the basic operations – addition, multiplication, conversions – which all require a last step, called coefficient reduction, that we present in details in Section 6.

Numerical examples are provided in Section 7.

2 Modular number systems

In classic positional number systems, every non-negative integer x is uniquely represented in radix r as

$$x = \sum_{i=0}^{n-1} x_i \, r^i, \text{ where } x_i \in \{0, \ldots, r-1\}. \tag{1}$$

If $x_{n-1} \neq 0$, x is said to be a n-digit radix-r number.

In most public-key cryptographic applications, computations have to be done over finite rings or fields. In prime fields $GF(p)$, we deal with representatives of equivalence classes modulo p (for simplicity we generally use the set of positive integers $\{0, 1, \ldots, p-1\}$), and the arithmetic operations – addition and multiplication – are performed modulo p.

In order to represent the set of integers modulo p, we define a *Modular number system* by extending the definition (1) of positional number systems.

Definition 1 (MNS) *A Modular Number System (MNS) $\mathcal{B}$, is a quadruple (p, n, γ, ρ), such that all positive integers $0 \leq x < p$ satisfy*

$$x = \sum_{i=0}^{n-1} x_i \, \gamma^i \bmod p, \text{ with } \gamma > 1 \text{ and } |x_i| < \rho. \tag{2}$$

The vector $(x_0, \ldots, x_{n-1})_{\mathcal{B}}$ denotes a representation of x in $\mathcal{B}$.

In the rest of the paper, we shall omit the subscript $(.)_{\mathcal{B}}$ when it is clear from the context. We shall represent the integer a either as the vector $\boldsymbol{a}$, or the polynomial A, without distinction. We shall use a_i to represent both for the ith element of $\boldsymbol{a}$, and the ith coefficient of A. (Note that we use a left-to-right notation; i.e., a_0 is the constant term.) Hence, depending on the context, we shall use $\|\boldsymbol{a}\| = \|A\|$, to refer to the norm of the vector or the corresponding polynomial. We shall also use the notation $\boldsymbol{a}_i$ to refer to the ith vector within a set of vectors or a matrix.

Example 1 *Let us consider the MNS defined with $p = 17, n = 3, \gamma = 7, \rho = 2$. Over this system, we represent the elements of $\mathbb{Z}_{17}$ as polynomials in γ of degree at most 2 with coefficients in $\{-1, 0, 1\}$ (cf. table 1).*

0	1	2	3	4	5
0	1	$-\gamma^2$	$1-\gamma^2$	$-1+\gamma+\gamma^2$	$\gamma+\gamma^2$

6	7	8	9	10	11
$-1+\gamma$	γ	$1+\gamma$	$-1-\gamma$	$-\gamma$	$1-\gamma$

12	13	14	15	16	
$-\gamma-\gamma^2$	$1-\gamma-\gamma^2$	$-1+\gamma^2$	γ^2	$1+\gamma^2$	

Table 1: The elements of $\mathbb{Z}_{17}$ in the $MNS(17,3,7,2)$

In example 1, we remark that the number of polynomials of degree 2, with coefficients in $\{-1,0,1\}$ is equal to $3^3 = 27$. Since we only have to represent 17 values, the system is clearly redundant. For example, we have $6 = 1+\gamma+\gamma^2 = -1+\gamma$, or $9 = 1-\gamma+\gamma^2 = -1-\gamma$. The level of redundancy depends on the parameters of the MNS. Note yet that, in this paper, we shall take advantage of the redundancy only by considering different representations of zero.

In a MNS, every integer $0 \leq x < p$ is thus represented as a polynomial in γ. What do we know about the coefficients of those polynomials? Are they upper-bounded? In other words, given the integers p and n, can we determine ρ and construct a MNS? In order to answer these questions, we shall use results from lattice reduction theory. We recall some basic facts about lattice theory in the next section.

3 A brief introduction to lattice reduction theory

A *lattice* $\mathcal{L}$ is a discrete sub-group of $\mathbb{R}^n$, or equivalently the set of all the integral combinations of $d \leq n$ linearly independent vectors over $\mathbb{R}$. (In this paper we shall only consider full-dimensional lattices, i.e., with $d = n$.)

$$\mathcal{L} = \mathbb{Z}\,\boldsymbol{a}_1 + \cdots + \mathbb{Z}\,\boldsymbol{a}_n = \{\lambda_1\boldsymbol{a}_1 + \cdots + \lambda_n\boldsymbol{a}_n \,:\, \lambda_i \in \mathbb{Z}\}.$$

We say that $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ is a *basis* of $\mathcal{L}$. The same lattice $\mathcal{L}$ may have many different basis.

The *fundamental domain* of $\mathcal{L}$ is given by

$$\mathcal{H} = \{x \in \mathbb{R}^n \,:\, x = \sum_{i=1}^{n} x_i\,\boldsymbol{a}_i,\ 0 \leq x_i < 1\}.$$

In the two-dimensional case, $\mathcal{H}$ is the parallelogram generated by $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, two linearly independent vectors in the plane (see Figure 1).

The *determinant* of a lattice $\mathcal{L}$ is the volume of its fundamental domain; i.e., the n-dimensional parallelepiped spanned by the vectors of the basis (see Figure 1).

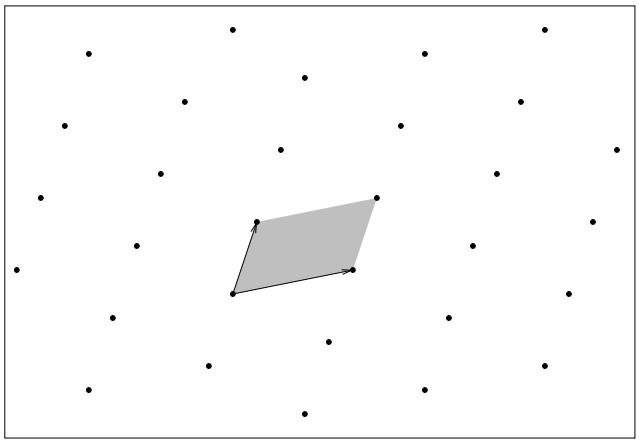


Figure 1: A lattice in dimension 2, a basis, and a geometric interpretation of the fundamental domain and the determinant

The determinant does not depend on the basis of $\mathcal{L}$, except for its sign. We have

$$\det \mathcal{L} = |\det \boldsymbol{A}|,$$

where $\boldsymbol{A}$ is any basis of $\mathcal{L}$. The following inequality, called *Hadamard's inequality*, is natural from a geometric point of view:

$$\|\boldsymbol{a}_1\| \dots \|\boldsymbol{a}_n\| \geq \det \mathcal{L}. \tag{3}$$

Hermite proved that every lattice $\mathcal{L}$ has a basis $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ such that

$$\|\boldsymbol{b}_1\| \dots \|\boldsymbol{b}_n\| \leq c_n \det \mathcal{L}, \tag{4}$$

where c_n is a constant depending only on n. The following algorithmic problem naturally follows Hermitte's result: Given a lattice $\mathcal{L} = \mathcal{L}(\boldsymbol{A})$, and $C > 0$, find a basis $\boldsymbol{B} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ of $\mathcal{L}$ such that

$$\|\boldsymbol{b}_1\| \dots \|\boldsymbol{b}_n\| \leq C \det \mathcal{L}.$$

It is known that this problem has a solution if $C \geq n^n$ (see [11]). However, it is not know how to find such a basis. A closely related problem, known as the *Shortest Vector Problem* (SVP), is the following: Given a lattice $\mathcal{L} = \mathcal{L}(\boldsymbol{A})$, and a number $\lambda > 0$, find a vector $\boldsymbol{v} \in \mathcal{L}$, $\boldsymbol{v} \neq 0$, such that $\|\boldsymbol{v}\| \leq \lambda$.

In the rest of this paper, we shall use the following result from Minkowski in the case of the ℓ_∞-norm:

Theorem 1 (Minkowski) *Every lattice $\mathcal{L}$ contains a vector $\boldsymbol{v} \neq 0$ such that*

$$\|\boldsymbol{v}\|_\infty \leq (\det \mathcal{L})^{1/n}. \tag{5}$$

Proof: See [11] in the case of full-dimensional lattices, and [13] in the general case. □

4 Polynomial Modular Number Systems

In this section we consider some special cases of MNS, where γ is a root (modulo p) of a given polynomial E. We propose the fundamental theorem of a MNS which provides an upper bound on ρ. Before we state the theorem, let us prove the following two lemmas.

Lemma 1 *Given $\mathcal{B} = MNS(p, n, \gamma, \rho)$, the lattice $\mathcal{L}$ generated by the $n \times n$ matrix*

$$\boldsymbol{A} = \begin{pmatrix} p & 0 & 0 & 0 & \dots & 0 \\ -\gamma & 1 & 0 & 0 & \dots & 0 \\ -\gamma^2 & 0 & 1 & 0 & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ -\gamma^{n-2} & 0 & 0 & \dots & 1 & 0 \\ -\gamma^{n-1} & 0 & 0 & \dots & 0 & 1 \end{pmatrix} \tag{6}$$

is the lattice of all the multiples of p in $\mathcal{B}$, i.e. $\boldsymbol{v} \in \mathcal{L} \Leftrightarrow V(\gamma) \equiv 0 \bmod p$.

Proof: By construction, all the vectors of $\boldsymbol{A}$ are linearly independent, and all equal to zero modulo p. By definition, $\mathcal{L} = \mathcal{L}(\boldsymbol{A})$ is the lattice of all the integral combinations of the vectors of $\boldsymbol{A}$. Thus $\forall \boldsymbol{v} \in \mathcal{L}$, we have $V(\gamma) \equiv 0 \pmod p$. In order to prove that $\mathcal{L}$ is actually the lattice of all the vectors representing zero in $\mathcal{B}$, we must prove that $\forall \boldsymbol{v} = (v_0, v_1, \dots, v_{n-1})$, with $V(\gamma) \equiv 0 \pmod p$, there exists $\boldsymbol{z} \in \mathbb{Z}^n$, such that $\boldsymbol{z}.\boldsymbol{A} = \boldsymbol{v}$. The vector $\boldsymbol{z} = (V(\gamma)/p, v_1, \dots, v_{n-1})$ always exists ($V(\gamma)$ is a multiple of p) and satisfies the condition. Thus, $\boldsymbol{v} \in \mathcal{L}$. □

Lemma 2 *Let $\mathcal{L}$ be the lattice defined by the matrix $\boldsymbol{A}$ in (6). Then, there exists a vector $\boldsymbol{v} \in \mathcal{L}$ such that*

$$\|\boldsymbol{v}\|_\infty \leq p^{1/n}. \tag{7}$$

Proof: The proof directly comes from Theorem 1. We simply remark that $|\det \boldsymbol{A}| = p$. $\square$

Given a modular number system $\mathcal{B} = MNS(p, n, \gamma, \rho)$, the following theorem says that one can represent every integer less than p as polynomials of degree less than n, with coefficients all less than $C \times p^{1/n}$, where C is a small constant.

Theorem 2 (Fundamental theorem of a MNS) *Let $p, n > 1$. Also define $E(X) = X^n + \alpha X + \beta$, with $\alpha, \beta \in \mathbb{Z}$, such that $E(\gamma) \equiv 0 \pmod{p}$, and E irreducible in $\mathbb{Z}[X]$. Then, we can define a modular number system $\mathcal{B} = MNS(p, n, \gamma, \rho)$ as soon as*

$$\rho \geq (|\alpha| + |\beta|)\, p^{1/n}. \tag{8}$$

Proof: Let $\mathcal{L}$ be the lattice defined by the $n \times n$ matrix $\boldsymbol{A}$ in (6), and let $\boldsymbol{u} \in \mathcal{L}$, $\boldsymbol{u} \neq 0$, be a vector which satisfies Minkowski's Theorem; i.e., such that $\|\boldsymbol{u}\|_\infty \leq p^{1/n}$.

We define a new lattice $\mathcal{L}'$ generated by the n vectors $(\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{n-1})$ with $\boldsymbol{b}_0 = \boldsymbol{u}$, and for $i \geq 1$, $\boldsymbol{b}_i$ is the vector corresponding to the polynomial $B_i = X^i\, U \bmod E$. (We recall that, according to our notations, U is the polynomial which corresponds to $\boldsymbol{u}$.) Using the property of E; $\gamma^n \equiv -\alpha\gamma - \beta \pmod{p}$, we obtain the $n \times n$ matrix

$$\boldsymbol{B} = \begin{pmatrix} u_0 & u_1 & u_2 & \ldots & u_{n-1} \\ -\beta u_{n-1} & (u_0 - \alpha u_{n-1}) & u_1 & \ldots & u_{n-2} \\ -\beta u_{n-2} & (-\beta u_{n-1} - \alpha u_{n-2}) & (u_0 - \alpha u_{n-1}) & \ldots & u_{n-3} \\ \vdots & & & \ddots & \vdots \\ -\beta u_2 & (-\beta u_3 - \alpha u_2) & (-\beta u_4 - \alpha u_3) & \ldots & u_1 \\ -\beta u_1 & (-\beta u_2 - \alpha u_1) & (-\beta u_3 - \alpha u_2) & \ldots & (u_0 - \alpha u_{n-1}) \end{pmatrix}. \tag{9}$$

Since $\boldsymbol{u} \in \mathcal{L}$, we know from Lemma 1 that $\boldsymbol{u}$ corresponds to zero in $\mathcal{B}$, i.e., $U(\gamma) \equiv 0 \pmod{p}$. By construction, $\exists Q \in \mathbb{Z}[X]$ such that $B_i(X) = X^i\, U(X) - Q(X)\, E(X)$. Since $U(\gamma) \equiv 0 \pmod{p}$ and $E(\gamma) \equiv 0 \pmod{p}$, we have $B_i(\gamma) \equiv 0 \pmod{p}$, $\forall i = 0 \ldots n-1$, or equivalently, the vectors $\boldsymbol{b}_i$ of $\boldsymbol{B}$ correspond to zero in $\mathcal{B}$. So are all the integral combinations of $\boldsymbol{b}_i$s. Thus, $\mathcal{L}' = \mathcal{L}'(\boldsymbol{B}) \subseteq \mathcal{L}$ is also a lattice of multiples of p.

In order to prove that $\boldsymbol{B}$ is a basis of $\mathcal{L}'$, we must prove that the vectors $\boldsymbol{b}_i$ are linearly independent. Let us assume that they are not linearly independent. Then, there exists a vector $\boldsymbol{z} \in \mathbb{Z}^n$, different from the null vector, such that $\boldsymbol{z}.\boldsymbol{B}$ is equal to the null vector; or equivalently,

such that $\sum_{i=0}^{n-1} z_i \, \boldsymbol{b}_i = (0, \ldots, 0)$. In the polynomial notation, since $B_i = X^i \, U \bmod E$, this is equivalent to

$$\sum_{i=0}^{n-1} z_i \, X^i \, U \equiv 0 \pmod{E}.$$

If we let Z be the polynomial with coefficients $z_0, \ldots, z_{n-1}$, we obtain the congruence

$$Z \, U \equiv 0 \pmod{E},$$

which is in contradiction with the hypothesis that Z and U are both non-null polynomials of degree less than n, and E is irreducible of degree n. Thus, we have proved that $\boldsymbol{B}$ is a basis of $\mathcal{L}'$.

Let $\mathcal{H}'$ be the fundamental domain of $\mathcal{L}'$. A well known result in lattice theory is that, every vector $\boldsymbol{v}$ ($\boldsymbol{v}$ belongs to $\mathbb{Z}^n$ in our case) is congruent to a unique vector of $\mathcal{H}'$ modulo the lattice $\mathcal{L}'$. Intuitively, this comes from the fact that we do not change the value represented by $\boldsymbol{v}$ by subtracting any vector of $\mathcal{L}'$ since they all represent zero modulo p. More formally

$$\forall \boldsymbol{v} \in \mathbb{Z}^n, \exists \boldsymbol{w} \in \mathcal{L}' \text{ such that } \boldsymbol{v} - \boldsymbol{w} \equiv \boldsymbol{v} \pmod{\mathcal{L}'}, \text{ and } \boldsymbol{v} - \boldsymbol{w} \in \mathcal{H}'.$$

As a consequence, if we let $\boldsymbol{v}'$ be the unique vector in $\mathcal{H}'$, congruent to $\boldsymbol{v}$ modulo $\mathcal{L}'$, we have

$$\|\boldsymbol{v}'\|_\infty \leq \max_{0 \leq i < n} \{\|\boldsymbol{b}_i\|_\infty, \ \boldsymbol{b}_i \in \boldsymbol{B}\}.$$

By construction, we can see in (9) that the largest vector of $\boldsymbol{B}$, in the ℓ_∞-norm, is less than $(|\alpha| + |\beta|)\|\boldsymbol{b}_0\|_\infty$, where $\boldsymbol{b}_0 = \boldsymbol{u}$ is a vector of $\mathcal{L}$ which satisfy $\|\boldsymbol{u}\|_\infty \leq p^{1/n}$. Hence

$$\|\boldsymbol{v}'\|_\infty \leq (|\alpha| + |\beta|) \, p^{1/n}. \tag{10}$$

To complete the proof, we simply remark that every integer $a \in \mathbb{N}$ can be first associated with the vector $\boldsymbol{a} = (a, 0, \ldots, 0)$, and reduced modulo $\mathcal{L}'$ to the vector $\boldsymbol{a}'$ which satisfy (10). We thus obtain a polynomial representation of $\boldsymbol{a}$ as $a_0 + a_1 X + \ldots + a_{n-1} X^{n-1}$, with $a = A(\gamma) \bmod p$, and $\|A\|_\infty \leq (|\alpha| + |\beta|) \, p^{1/n}$. This concludes the proof. □

Definition 2 (PMNS) *A modular number system $\mathcal{B} = MNS(p, n, \gamma, \rho)$ that satisfies the conditions of Theorem 2 is called a Polynomial Modular Number System (PMNS). We shall denote $\mathcal{B} = PMNS(p, n, \gamma, \rho, E)$.*

In practice, we shall define the polynomial E with α and β as small as possible.

Example 2 *We define the PMNS with* $p = 23$*,* $n = 3$*,* $\rho = 2$*,* $E(X) = X^3 - X + 1$ *(*$\alpha = -1, \beta = 1$*). We easily check that* $\gamma = 13$ *is a root of* E *in* $\mathbb{Z}_{23}$*, and* E *is irreducible in* $\mathbb{Z}[X]$*. We represent the elements of* $\mathbb{Z}_{23}$ *as polynomials of degree at most 2, with coefficients in* $\{-1, 0, 1\}$*.*

0	1	2	3	4	5	6	7
0	1	$-\gamma - \gamma^2$	$1 - \gamma - \gamma^2$	$-1 + \gamma - \gamma^2$	$\gamma - \gamma^2$	$1 + \gamma - \gamma^2$	$-1 + \gamma^2$

8	9	10	11	12	13	14	15
γ^2	$1 + \gamma^2$	$-\gamma$	$1 - \gamma$	$-1 + \gamma$	γ	$1 + \gamma$	$-\gamma^2$

16	17	18	19	20	21	22	23
$1 - \gamma^2$	$-1 - \gamma + \gamma^2$	$-\gamma + \gamma^2$	$1 - \gamma + \gamma^2$	$-1 + \gamma + \gamma^2$	$\gamma + \gamma^2$	$1 + \gamma + \gamma^2$	

Table 2: The elements of $\mathbb{Z}_{23}$ in the $PMNS(23, 3, 13, 2, X^3 - X + 1)$

5 PMNS arithmetic

In this section, we propose algorithms for the classical operations in $GF(p)$, namely addition and multiplication modulo p, when the operands are represented in a PMNS. We give solutions for the conversion from binary to PMNS, and back. For simplicity, we assume $\rho = 2^k$, and all the operands are represented in $\mathcal{B} = PMNS(p, n, \gamma, 2^k, E)$, with $E = X^n + \alpha X + \beta$.

At this point, it is important to understand that the value k we shall consider here, depends on the way we implement the coefficient reduction. From Theorem 2, we know that every integer $a < p$ can be represented in a PMNS with coefficients $|a_i| \leq (|\alpha| + |\beta|)\, p^{1/n}$, i.e., of size at most

$$\left\lceil \log_2 (|\alpha| + |\beta|) + \frac{1}{n} \log_2(p) \right\rceil \quad \text{bits.} \tag{11}$$

Algorithms designed for common problems in the lattice's world, such as CVP_∞, can be used to reach this bound (see [12] for details). However, they are unpractical from an arithmetic point of view. The algorithms we propose in Section 6 can be seen as approximation algorithms for our specific lattices. For each proposed solution, we evaluate the size of the resulting coefficients. This gives us the size that we must consider for the coefficients in the definition of a PMNS; i.e., the value of k.

5.1 Addition, subtraction

Let a, b be two integers less than p, given in their PMNS representation. We want to compute the sum $s = a + b \bmod p$. Because of the polynomial nature of the PMNS representation, additions can be carried out independently, in parallel, on each coefficients. Let $A, B \in \mathbb{Z}[X]$ be the polynomial representations of a and b respectively. We have $A(\gamma) \equiv a \pmod p$, and $B(\gamma) \equiv b \pmod p$, and we compute $C = A + B$, such that

$$C(\gamma) \equiv A(\gamma) + B(\gamma) \pmod p, \tag{12}$$

or equivalently $c \equiv a + b \pmod p$.

Yet, the result of (12) is a polynomial of degree less than n, but whose coefficients c_i can be larger than $\rho = 2^k$. It is clear however that $\|C\|_\infty < 2^{k+1}$. In order to obtain c in a valid PMNS form, we thus need to reduce its coefficients. We propose different approaches to this problem in Section 6.

5.2 Multiplication

As for the addition, we use the polynomial forms of a and b to compute the product $r = ab \bmod p$. The details are presented in Algorithm 1.

Algorithm 1 [PMNS Modular Multiplication]

Input: $A = (a_0, \ldots, a_{n-1})_{\mathcal{B}}$, $B = (b_0, \ldots, b_{n-1})_{\mathcal{B}}$, with $|a_i|, |b_i| < 2^k$

Output: $R = (r_0, ..., r_{n-1})_{\mathcal{B}}$ such that $R(\gamma) \equiv ab \bmod p$, with $|r_i| < 2^k$

1: Polynomial multiplication in $\mathbb{Z}[X]$: $C(X) \leftarrow A(X)\, B(X)$

2: Polynomial reduction: $C'(X) \leftarrow C(X) \bmod E(X)$

3: Coefficient reduction: $R \leftarrow CR(C')$

Given $A, B \in \mathbb{Z}[X]$, with $A(\gamma) \equiv a \pmod p$, and $B(\gamma) \equiv b \pmod p$, we first evaluate the product $C = A\, B$. Clearly, we have

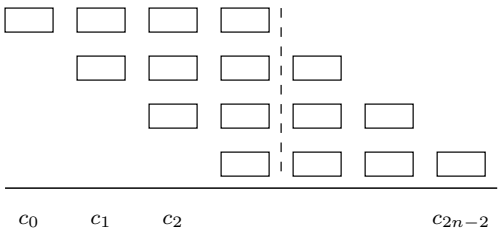
$$C(\gamma) \equiv A(\gamma)\, B(\gamma) \pmod p. \tag{13}$$

Since $\deg A, \deg B \leq n - 1$, their polynomial product C satisfies $\deg C \leq 2n - 2$. Step 2 of Algorithm 1 consists in the reduction modulo E, which reduces C to a polynomial of degree less than n. Since $E(\gamma) \equiv 0 \pmod p$, the polynomial $C' = C \bmod E$ corresponds to the same value

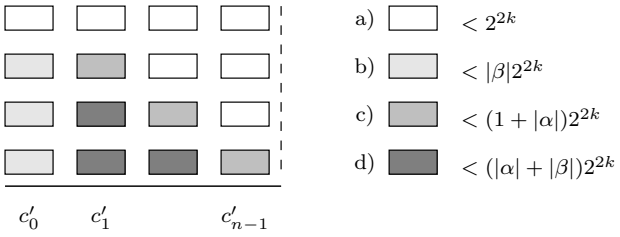
than C in $\mathcal{B}$. In other words, there exists $K \in \mathbb{Z}[X]$ such that

$$C'(\gamma) = C(\gamma) - K(\gamma)\, E(\gamma) \equiv C(\gamma) \pmod{p},$$

with $\deg C' < \deg E = n$. Note that the special form of $E(X) = X^n + \alpha X + \beta$ nicely simplifies the reduction modulo E. As shown in Fig. 2, we first compute the product $C = (c_0, \ldots, c_{2n-2})_{\mathcal{B}}$ (see Fig. 2(a)), and we reduce the terms of order $\geq n$ using the congruence $X^n \equiv -\alpha X - \beta \pmod{E}$ (see Fig. 2(b)).



(a) The polynomial $C = AB$



(b) The polynomial $C' = C \bmod E$

Figure 2: Reduction modulo $E(X) = X^n + \alpha X + \beta$

If A, B have coefficients such that $|a_i|, |b_i| < 2^k$, then, we can see in Fig. 2(b) that the coefficients of the polynomial C' reduced modulo E, satisfy

$$|c'_i| < n\,(|\alpha| + |\beta|)\, 2^{2k}, \quad \forall i = 0, \ldots, n-1, \tag{14}$$

To be a little more precise, we can remark that $|c'_i| < ((n-1)|\alpha| + (n-2)|\beta| + 2)\, 2^{2k}$; this upper bound being given by c'_1. As for the addition, the final step in the multiplication algorithm will consist in a coefficient reduction that we shall detail in Section 6.

5.3 Conversions

We briefly propose methods for the conversions from binary to PMNS, and from PMNS to binary, that can easily be implemented at low memory cost.

5.3.1 Binary to PMNS

Given the integer $0 \leq a < p$, we want to define a polynomial A with coefficients less than 2^k, which satisfy $A(\gamma) \equiv a \pmod{p}$. We first represent a in radix 2^k as in (1):

$$a = \sum_{i=0}^{n-1} d_i \left(2^k\right)^i, \quad \text{with } 0 \leq d_i < 2^k. \tag{15}$$

Our approach requires the precomputation of n values T_i, corresponding to the polynomial representations of 2^{ki} for $i = 0, \ldots, n-1$. We have

$$T_i(\gamma) \equiv 2^{ki} \pmod{p}, \quad \text{for } i = 0, \ldots, n-1.$$

We store those polynomials in a reduced form; i.e., with coefficients less than $(|\alpha| + |\beta|)2^k$. In the polynomial form, Equation (15) rewrites

$$A = \sum_{i=0}^{n-1} d_i \, T_i, \quad \text{with } 0 \leq d_i < 2^k.$$

A satisfies $\|A\|_\infty < n(|\alpha| + |\beta|)2^{2k}$, and can be reduced using one of the coefficient reduction algorithms presented in Section 6.

5.3.2 PMNS to binary

Given the polynomial A, we want to recover the corresponding integer $a = A(\gamma) \pmod{p}$. In the PMNS representation, we have

$$A(\gamma) = \sum_{i=0}^{n-1} a_i \, \gamma^i.$$

We use n precomputed integers g_i, corresponding to powers of γ:

$$g_i = \gamma^i \bmod p, \quad \text{for } i = 0, \ldots, n-1.$$

Hence, a can be computed by

$$a = \sum_{i=0}^{n-1} a_i \, g_i \bmod p.$$

6 Coefficient reduction techniques

In this section, we propose two different approaches for the coefficient reduction. The first one, presented in Section 6.1 is inspired by Barrett modular reduction over the integers. In Section 6.2, we present solutions using lookup tables.

6.1 Reduction using a Barrett-like algorithm

To simplify the notations, we shall use the symbol $*$ to denote the multiplication of two polynomials modulo E: $A * B = A\,B \bmod E$. Algorithm 2 bellow is inspired by Barrett modular reduction algorithm [6, 1].

Given $C = (c_0, \ldots, c_{n-1})_{\mathcal{B}}$ with coefficients $|c_i| < n\,(|\alpha| + |\beta|)\,2^{2k}$, we want to construct R such that $R(\gamma) \equiv C(\gamma) \pmod p$, with coefficients less than 2^k. If M is a well chosen polynomial, such that $M(\gamma) \equiv 0 \pmod p$ (or equivalently, the corresponding vector $\boldsymbol{m}$ belong to $\mathcal{L}$), then, we know that adding to C any multiple of M do not change the value of C. We propose a Barrett-like algorithm to compute a polynomial Q such that $R = C - Q * M$ satisfies $\|R\|_\infty < 2^k$. Roughly speaking, we compute Q and $C * M^{-1}$, with nearby integer coefficients, such that $R = C - Q * M$ has small coefficients. To make the analogy with Barrett's reduction algorithm over the integers, the polynomial Q can be seen as an approximation of the quotient C/M.

It is important to understand that Algorithm 2 is correct as soon as $\boldsymbol{m} \in \mathcal{L}$. As we shall see in the proof of Algorithm 2, the coefficients of the reduced polynomial R are proportional to those of M. For this reason, we shall try to define M with coefficients as small as possible. A solution to this problem is given by the LLL algorithm [14]. (See Section 7 for an example).

Algorithm 2 requires the precomputed value $\hat{M} = \lfloor 2^{2k+l} * M^{-1} \rfloor$. Since E is irreducible, M^{-1} always exists. Yet, it has rational coefficients. The improper notation $\lfloor \cdot \rfloor$ used here for $\hat{M}$, means that we take the nearest integer towards 0 of each coefficients of the polynomial $2^{2k+l} M^{-1} \bmod E$. Thus, $\hat{M}$ has integer coefficients. In general, if $A \in \mathbb{Q}[X]$ we denote

$$\lfloor A \rfloor = A + \Psi, \quad \text{with } \psi_i \in \mathbb{Q}, \quad |\psi_i| < 1.$$

Theorem 3 *The Algorithm 2 is correct; i.e., it returns R such that $R(\gamma) \equiv C(\gamma) \pmod p$ and $\|R\|_\infty < 2^k$.*

Algorithm 2 [CR: Coefficient Reduction]

Precomputed: A polynomial M s.t. $M(\gamma) \equiv 0 \pmod{p}$, and $\hat{M} = \left\lfloor 2^{2k+l} * M^{-1} \right\rfloor$,

with $k = \lceil \log_2(\|M\|_\infty) + \log_2(4n(|\alpha| + |\beta|)) \rceil$, and $l = \lceil 2\log_2(n(|\alpha| + |\beta|)) \rceil$

Input: $C = (c_0, \ldots, c_{n-1})_{\mathcal{B}}$, with $|c_i| < n\,(|\alpha| + |\beta|)\,2^{2k}$

Output: $R = (r_0, \ldots, r_{n-1})_{\mathcal{B}}$ such that $R(\gamma) \equiv C(\gamma) \bmod p$, with $|r_i| < 2^k$

1: $Q \leftarrow \left\lfloor \dfrac{\lfloor C/2^{k-1} \rfloor * \hat{M}}{2^{k+l+1}} \right\rfloor$

2: $R \leftarrow C - Q * M$

Proof: Let us first prove that r is congruent to c modulo p. Since $M(\gamma) \equiv E(\gamma) \equiv 0 \pmod{p}$, then $\exists K \in \mathbb{Z}[X]$ such that

$$R(\gamma) = C(\gamma) - (Q(\gamma)M(\gamma) - K(\gamma)E(\gamma))\,.$$

Thus, we have

$$R(\gamma) \equiv C(\gamma) \pmod{p}. \tag{16}$$

What remains to be proved is that the polynomial Q computed in step 1 actually yields R with $|r_i| < 2^k$. In step 1, we compute

$$Q = \left\lfloor \frac{\lfloor C/2^{k-1} \rfloor * \hat{M}}{2^{k+l+1}} \right\rfloor. \tag{17}$$

In order to evaluate the approximation errors on Q computed in step 1, we define three polynomials Ψ_1, Ψ_2, Ψ_3, with $\|\Psi_s\|_\infty < 1$, for $s = 1, 2, 3$. Equation (17) is equivalent to

$$Q = \frac{(C/2^{k-1} - \Psi_1) * (2^{2k+l} * M^{-1} - \Psi_2)}{2^{k+l+1}} - \Psi_3,$$

which rewrites

$$Q = C * M^{-1} - \frac{\Psi_1 * (2^{2k+l} * M^{-1}) + \Psi_2 * (C/2^{k-1}) - \Psi 1 * \Psi_2}{2^{k+l+1}} - \Psi_3.$$

Thus, step 2 gives

$$R = C - Q * M = \left(\frac{\Psi_1 * (2^{2k+l} * M^{-1}) + \Psi_2 * (C/2^{k-1}) - \Psi 1 * \Psi_2}{2^{k+l+1}} + \Psi_3 \right) * M$$

$$= 2^{k-1}\Psi_1 + \left(\frac{C * \Psi_2}{2^{2k+l}} - \frac{\Psi_1 * \Psi_2}{2^{k+l+1}} + \Psi_3 \right) * M$$

$$= 2^{k-1}\Psi_1 + \left(\frac{\Psi_2}{2^{k+l+1}} * \left(C/2^{k-1} - \Psi_1 \right) + \Psi_3 \right) * M$$

$$= 2^{k-1}\Psi_1 + \left(\frac{\Psi_2}{2^{k+l+1}} * \left\lfloor C/2^{k-1} \right\rfloor + \Psi_3 \right) * M$$

If we let $\hat{C} = \left\lfloor C/2^{k-1} \right\rfloor$, and $\tilde{C} = \left(\frac{\Psi_2}{2^{k+l+1}} * \hat{C} + \Psi_3 \right)$, then we have

$$\|\hat{C}\|_\infty \leq \left\| C/2^{k-1} \right\|_\infty < n \left(|\alpha| + |\beta| \right) 2^{k+1},$$

and thus

$$\|\tilde{C}\|_\infty < \frac{n^2 \left(|\alpha| + |\beta| \right)^2 2^{k+1}}{2^{k+l+1}} + 1.$$

Since $2^l \geq n^2(|\alpha| + |\beta|)^2$ by hypothesis, we get $\|\tilde{C}\|_\infty < 2$. From (14), it follows that

$$\|\tilde{C} * M\|_\infty < 2n \left(|\alpha| + |\beta| \right) \|M\|_\infty;$$

and since, by hypothesis, $2^k \geq 4n(|\alpha| + |\beta|)\|M\|_\infty$, we have

$$\|\tilde{C} * M\|_\infty < 2^{k-1}.$$

Finally, since $R = 2^{k-1}\Psi_1 + \tilde{C} * M$ and $\|\Psi_1\|_\infty < 1$, the coefficient of R satisfy $|r_i| < 2^k$ for $i = 0, \ldots, n-1$. This concludes the proof. □

Corollary 1 *(Modular Multiplication Stability) Let M be such that $\|M\|_\infty = p^{1/n}$. If $k \geq \left\lceil \log_2(4n \left(|\alpha| + |\beta| \right)) + \log_2(p^{1/n}) \right\rceil$, then, there exits a modular multiplication algorithm which, given two polynomials A, B with coefficients $|a_i|, |b_i| < 2^k$, returns R such that $R(\gamma) \equiv A(\gamma)\, B(\gamma) \pmod{p}$, and $|r_i| < 2^k$.*

Proof: From Theorem 1, we can always find a vector $\boldsymbol{m} \in \mathcal{L}$ such that $\|\boldsymbol{m}\|_\infty \leq p^{1/n}$. We use Algorithm 2 for the coefficient reduction part of Algorithm 1. □

6.2 Reduction using lookup tables

An alternative solution to the previous algorithm is to perform the coefficient reduction using lookup tables. Algorithm 3, presented bellow, is used to reduce the coefficients of a given polynomial by a single bit. Algorithm 4 in Section 6.2.3 performs several iterations of this algorithm to get a complete coefficient reduction.

Let $C \in \mathbb{Z}[X]$ be a polynomial of degree less than n, with coefficients $|c_i| < 2^{k+1}$. We want to define a polynomial C' which satisfies $C'(\gamma) \equiv C(\gamma) \pmod{p}$, and $|c'_i| < 2^k$. We can decompose C into two polynomials L and H, such that

$$C = L + 2^{k-1}H, \quad \text{with } |l_i| < 2^{k-1} \text{ and } |h_i| < 2^2. \tag{18}$$

Figure 3 shows the decomposition (18).

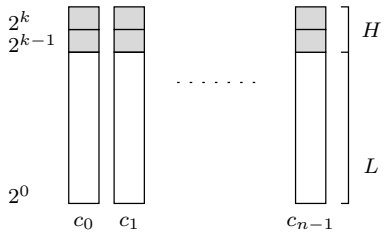


Figure 3: The decomposition of C into its lower part L, and higher part H

A polynomial H' satisfying $|h'_i| < 2^{k-1}$ is deduced from lookup tables or registers, and possibly some small computations, such that the result $C' = L + H'$ has coefficients less than 2^k.

Algorithm 3 [TCR: Table-based Coefficient Reduction]

Input: $C = (c_0, \ldots, c_{n-1})_{\mathcal{B}}$, with $|c_i| < 2^{k+1}$

Output: $C' = (c'_0, ..., c'_{n-1})_{\mathcal{B}}$ such that $c' \equiv c \pmod{p}$, and $|c'_i| < 2^k$ for $i = 0, \ldots, n-1$

1: $C = L + 2^{k-1}H$

2: $H' \leftarrow \texttt{TREAD}[H]$

3: $C' \leftarrow L + H'$

In the next two paragraphs, we propose two approaches, called horizontal and vertical, for the implementation of step 2 of Algorithm 3. As in Section 6.1, we shall define k according to the parameters p, n, α, β, such that the coefficient reduction algorithms return a polynomial with coefficients less than 2^k.

6.2.1 Horizontal approach

Let H_1, H_2 be such that $H = H_1 + 2H_2$, with $|h_{1,i}|, |h_{2,i}| \leq 1$. Step 2 of Algorithm 3 can be rewritten as

$$H' \leftarrow \mathtt{TAB}_1[H_1] + \mathtt{TAB}_2[H_2], \tag{19}$$

where $\mathtt{TAB}_1[H_1]$ and $\mathtt{TAB}_2[H_2]$ return $H'_1 = 2^{k-1}H_1$, and $H'_2 = 2^k H_2$ respectively. H'_1 and H'_2 must be stored in a reduced form, with $|h'_{1,i}|, |h'_{2,i}| < 2^{k-2}$, such that $|h'_i| < 2^{k-1}$.

From Theorem 2, there exists H'_1, H'_2 such that $\|H'_j\|_\infty \leq (|\alpha| + |\beta|)\, p^{1/n}$, for $j = 1, 2$. Since we add two such polynomials, the horizontal approach requires $(|\alpha| + |\beta|)\, p^{1/n} < 2^{k-2}$. Thus, taking

$$k \geq \lceil 2 + \tau \rceil, \tag{20}$$

where

$$\tau = \log_2 (|\alpha| + |\beta|) + \frac{1}{n} \log_2(p) \tag{21}$$

is the bound given by Theorem 2, ensures $|h'_i| < 2^{k-1}$, and thus $|c'_i| < 2^k$.

Since $h_{1,i}, h_{2,i} \in \{-1, 0, 1\}$, each table has to store $3^n - 1$ (there is no need to store the null polynomial) polynomials with n coefficients of size τ (in absolute value), plus 1 bit for their sign. The cost in memory for the horizontal approach is thus

$$|\mathtt{TAB}_1| + |\mathtt{TAB}_2| = 2(3^n - 1) \times n(\tau + 1) \text{ bits},$$

with τ defined in (21).

6.2.2 Vertical approach

From the decomposition (18), we have $h_i \in \{-3, \ldots, 3\}$, for $i = 0, \ldots, n-1$. We can use three registers $\mathtt{REG}[h_i]$, $h_i = 1, 2, 3$, to store three reduced polynomials which correspond to $|h_i|\, 2^{k-1}$, namely 2^{k-1}, $2\, 2^{k-1}$, and $3\, 2^{k-1}$. Note that there is no need to the polynomials which correspond to 0 and the negative values of h_i (we simply flip the signs of the polynomials stored for $|h_i|$ if necessary). Step 2 of Algorithm 3 rewrites

$$H' \leftarrow \sum_{i=0}^{n-1} \left(\mathtt{REG}[h_i] X^i\right) \bmod E. \tag{22}$$

In order to get C' in step 3 with coefficients less than 2^k, we need $|h'_i| < 2^{k-1}$. Since we perform n additions, plus a reduction modulo E of reduced polynomials to get H' (the

multiplications by the powers of X reduce to shifts on the coefficients), we need $\log_2(n) + \log_2(|\alpha| + |\beta|)$ extra bits. We can take

$$k \geq \lceil 1 + \log_2(n) + \log_2(|\alpha| + |\beta|) + \tau \rceil, \tag{23}$$

with τ defined in (21).

We need to store three polynomials with n coefficients of size τ (in absolute value), plus 1 bit for their sign. The memory cost of the vertical approach is thus

$$|\texttt{REG}| = 3 \times n(\tau + 1) \text{ bits.}$$

6.2.3 Complete coefficient reduction using look-up table

Generic Algorithm 3 presented at the beginning of this section admits many implementation options. In Sections 6.2.1,and 6.2.2, we proposed two different solutions to reduce the coefficients of a given polynomial C from $k+1$ bits to k bits.

For completeness, we propose a straightforward algorithm that can reduce the coefficients of any given polynomial C to a polynomial with coefficient less than 2^k. We simply reduce one-by-one the bits of the coefficients of C, by performing several iterations of a single bit reduction algorithm, until we reach the desired size. Algorithm 4 bellow summarizes the computations.

Algorithm 4 [CTCR: Complete Table-based Coefficient Reduction]

Input: $C = (c_0, \ldots, c_{n-1})_{\mathcal{B}}$, with $|c_i| < 2^{k+t}$

Output: $R = (r_0, ..., r_{n-1})_{\mathcal{B}}$ such that $r \equiv c \pmod{p}$, and $|r_i| < 2^k$

1: $R \leftarrow C$

2: **for** i **from** $t-1$ **to** 0 **do**

3: $\quad R = L + 2^i U \quad$ with $|l_i| < 2^i$ and $|u_i| < 2^{k+1}$

4: $\quad R \leftarrow L + 2^i\, TCR(U)$

5: **end for**

6.3 Comparisons and complexity

Table 3 bellow summarizes the different approaches presented in this section for the coefficient reduction. It gives the different bounds on k given by Theorem 2 and each of the proposed algorithms.

Algorithm	lower bounds on k
Theorem 2	τ
Barrett	$\log_2(4n) + \tau$
Vertical approach	$\log_2(2n\,(\lvert\alpha\rvert + \lvert\beta\rvert)) + \tau$
Horizontal approach	$2 + \tau$

Table 3: Bounds on k given by Theorem 2 and the different coefficient reduction algorithms

In Table 4, we also give the complexity (in numbers on elementary operations) and the memory requirements (in bits) of the two table-based coefficient reduction techniques. Let $PA(k)$ define the number of additions required by a polynomial addition with coefficients of size k. In Table 4, we have counted one full polynomial addition for the operation in step 4 of

Cost	Vertical	Horizontal
Memory (in bits)	$3 \times n(k - \log_2(n\,(\lvert\alpha\rvert + \lvert\beta\rvert)))$	$2(3^n - 1) \times n(k-1)$
# op. TCR	$(n+2)\,PA(k)$	$2\,PA(k)$
# op. CTCR	$t(n+3)\,PA(k)$	$3t\,PA(k)$

Table 4: Complexity, in memory and number of operations, of the horizontal and vertical coefficient reduction algorithms

Algorithm 4. Note yet, that we only perform this addition when $TCR(U)$ and L do not have the same sign. And even in this case, it reduces to the subtraction of a k-bit number by the one-bit integer 1.

In Table 5, we give some numerical examples assuming $(\lvert\alpha\rvert + \lvert\beta\rvert) \leq 2$. We give the number of bits of the largest numbers (size of p) we can represent with $n = 4, \ldots, 8$ for 32, and 64 bits word-lengths. In comparison, we also give the number of words, the same numbers require in classic binary representation. From Table 5, we clearly see that, given p, it is always possible to chose n, and k such that the number of words required by the PMNS representation is the same as the number of words required by the multiprecision representation.

7 Examples

In this section, we propose examples which illustrate some of the algorithms proposed in the previous sections. Given a prime number p, $n > 4$, and E a polynomial, we first define a PMNS, and evaluates k using the chosen coefficient reduction algorithm. We convert two integers a, b

n	Algo	$k = 32$			$k = 64$		
		size of p (in bits)	# words (in binary)	Memory (in bits)	size of p (in bits)	# words (in binary)	Memory (in bits)
$n = 4$	T	124	4	-	252	4	-
	B	108	4	-	236	4	-
	V	108	4	348	236	4	732
	H	116	4	19 840	244	4	40 320
$n = 5$	T	155	5	-	315	5	-
	B	133	5	-	293	5	-
	V	133	5	435	293	5	915
	H	145	5	75 020	305	5	152 460
$n = 6$	T	186	6	-	378	6	-
	B	158	5	-	350	6	-
	V	158	5	522	350	6	1 098
	H	174	6	270 816	366	6	550 368
$n = 7$	T	217	7	-	441	7	-
	B	183	6	-	407	7	-
	V	183	6	609	407	7	1 281
	H	203	7	948 724	427	7	1 928 052
$n = 8$	T	248	8	-	504	8	-
	B	208	7	-	464	8	-
	V	208	7	672	464	8	1 440
	H	232	8	3 253 760	488	8	6 612 480

Table 5: Number of bits of the largest numbers (size of p) we can represent with n ranging from 4 to 8, for words of 32, and 64 bits; and the number of words, the same numbers require in classic binary representation

into this PMNS. Then, we perform the multiplication $a\,b \bmod p$, using the PMNS modular multiplication presented in Algorithm 1, with both the Barrett-like (example 7.1) and the horizontal approach (example) solutions proposed in Sections 6.1, and 6.2.1 for the coefficient reduction. Finally, we convert the result back in binary.

7.1 Example using the Barrett-like algorithm

Definition of the PMNS: Let us define $p = 123456789120001$, $|p| = 47$ bits. Also define $E(X) = X^4 + 1$ ($n = 4, \alpha = 0, \beta = 1$). E is irreducible, and $\gamma = 46988594033438$ is a root of E modulo p. Let the matrix $\boldsymbol{A}$ be defined as in (6)

$$\boldsymbol{A} = \begin{pmatrix} p & 0 & 0 & 0 \\ -\gamma & 1 & 0 & 0 \\ -\gamma^2 & 0 & 1 & 0 \\ -\gamma^3 & 0 & 0 & 1 \end{pmatrix} \tag{24}$$

$\boldsymbol{A}$ is a basis of $\mathcal{L} = \mathcal{L}(\boldsymbol{A})$. In order to find a vector $\boldsymbol{m}$ with small coefficients (in absolute value), we use the LLL algorithm [14], which gives us the approximated reduced basis

$$LLL(\boldsymbol{A}) = \begin{pmatrix} -1332 & -1562 & 3497 & -9 \\ 1629 & 3191 & -306 & -297 \\ -3191 & 306 & 297 & 1629 \\ 306 & 297 & 1629 & 3191 \end{pmatrix} \tag{25}$$

We remark that the first vector of $LLL(\boldsymbol{A})$ has small enough coefficients for our purpose. We thus define $\boldsymbol{m} = (-1332, -1562, 3497, -9)$, and its polynomial form M as

$$M(X) = -1332 - 1562\,X + 3497\,X^2 - 9\,X^3.$$

Note that, although $\boldsymbol{m}$ is not the smallest vector (in the ℓ_∞-norm) in the lattice defined by $LLL(\boldsymbol{A})$, its largest coefficients has 12 bits; i.e, the minimum we can expect since $\lceil log_2(p)/n \rceil = 12$.

We shall perform the coefficient reduction using the Barrett-like algorithm presented in Section 6.1. According to Algorithm 2, we define $\rho = 2^k$ with

$$k = \lceil 2 + log_2(n) + log_2(|\alpha| + |\beta|) + log_2(\|\boldsymbol{m}\|_\infty) \rceil = 16.$$

Precomputations: Again, using Algorithm 2, we define

$$l = \lceil 2\,(log_2(n) + log_2\,(|\alpha| + |\beta|)) \rceil = 4.$$

We compute the polynomial $\hat{M} \triangleq \left\lfloor 2^{2k+l}\, M^{-1} \bmod E \right\rfloor$:

$$\hat{M} = 7576151 - 4562193\, X - 14505927\, X^2 + 5084425\, X^3.$$

Binary to PMNS: Let $a = 111111111111111, b = 22222222222222$. The binary-to-PMNS conversion algorithm presented in Section 5.3 gives the representations of a and b in $\mathcal{B}$. We have

$$A = 33086 + 1902\, X - 1128\, X^2 - 1343\, X^3$$
$$B = 29052 - 1254\, X + 1352\, X^2 - 2988\, X^3$$

It is easy to check that $a = A(\gamma) \bmod p$, and $b = B(\gamma) \bmod p$.

Modular multiplication: We use Algorithm 1 for the multiplication modulo E, and Algorithm 2 (Barrett-like algorithm) for the coefficient reduction. We have

$$\begin{aligned} C(X) = A(X)\, B(X) &= 961214472 + 13767060\, X + 9576508\, X^2 - 133891788\, X^3 \\ &\quad - 5524110\, X^4 + 1554728\, X^5 + 4012884\, X^6 \\ C'(X) = C(X) \bmod E(X) &= 966738582 + 12212332\, X + 5563624\, X^2 - 133891788\, X^3 \\ Q(X) &= -115192 - 94199\, X - 195576\, X^2 + 83334\, X^3 \\ R(X) &= 53649 - 11458\, X - 6028\, X^2 - 3437\, X^3 \end{aligned}$$

We remark that the coefficients of R satisfy $|r_i| < 2^k = 2^{16} = 65536$.

PMNS to binary: Finally, we can convert the result R in binary using the algorithm proposed in Section 5.3. We obtain $r = 76459417066083$, and we easily check that this is the correct result $r = a\, b \bmod p$.

7.2 Example using the table-based horizontal approach

Definition of the PMNS: Let us define $p = 123456789120001$, $|p| = 47$ bits. Also define $E(X) = X^4 + 1$ ($n = 4, \alpha = 0, \beta = 1$). E is irreducible in $\mathbb{Z}$, and $\gamma = 46988594033438$ is a root of E modulo p. Let the matrix $\boldsymbol{A}$ be defined as in (6)

$$\boldsymbol{A} = \begin{pmatrix} p & 0 & 0 & 0 \\ -\gamma & 1 & 0 & 0 \\ -\gamma^2 & 0 & 1 & 0 \\ -\gamma^3 & 0 & 0 & 1 \end{pmatrix} \tag{26}$$

$\boldsymbol{A}$ is a basis of $\mathcal{L} = \mathcal{L}(\boldsymbol{A})$. In order to find a vector $\boldsymbol{m}$ with small coefficients (in absolute value), we use the LLL algorithm [14], which gives us the approximated reduced basis

$$LLL(\boldsymbol{A}) = \begin{pmatrix} -1332 & -1562 & 3497 & -9 \\ 1629 & 3191 & -306 & -297 \\ -3191 & 306 & 297 & 1629 \\ 306 & 297 & 1629 & 3191 \end{pmatrix} \tag{27}$$

We remark that the first vector of $LLL(\boldsymbol{A})$ has small enough coefficients for our purpose. We thus define $\boldsymbol{m} = (-1332, -1562, 3497, -9)$, and its polynomial form M as

$$M(X) = -1332 - 1562\,X + 3497\,X^2 - 9\,X^3.$$

Note that, although $\boldsymbol{m}$ is not the smallest vector (in the ℓ_∞ norm) in the lattice defined by $LLL(\boldsymbol{A})$, its largest coefficients has 12 bits; i.e, the minimum we can expect since $\lceil log_2(p)/n \rceil = 12$.

We shall perform the coefficient reduction using the Full Table-Based Coefficient Algorithm 4, and the horizontal approach for TCR (see Algorithm 3). According to (20), we define $\rho = 2^k$ with

$$k = \left\lceil 2 + \log_2\left(|\alpha| + |\beta|\right) + \log_2(p^{1/n}) \right\rceil = 14$$

Binary to PMNS: Let $a = 111111111111111, b = 22222222222222$. The polynomials A, B, bellow, are given by Algorithm 4 with inputs $(a, 0, 0, 0)$ and $(b, 0, 0, 0)$ respectively. We have

$$A = 4466 + 6362\,X - 6906\,X^2 - 2934\,X^3$$
$$B = -2835 - 1844\,X - 2252\,X^2 - 7482\,X^3$$

It is easy to check that $a = A(\gamma) \bmod p$, and $b = B(\gamma) \bmod p$.

Modular multiplication: We use Algorithm 1 for the multiplication modulo E, and Algorithm 4 for the coefficient reduction. We have

$$C(X) = A(X)\,B(X) = -12661110 - 26271574\,X - 2210450\,X^2 - 26689282\,X^3 \\ - 26637876\,X^4 + 58278060\,X^5 + 21952188\,X^6,$$

and

$$C'(X) = C(X) \bmod E(X) = 13976766 - 84549634\,X - 24162638\,X^2 - 26689282\,X^3.$$

R_i	$r_{i,0}$	$r_{i,1}$	$r_{i,2}$	$r_{i,3}$
R_{13}	13976766	−84549634	−24162638	−26689282
R_{12}	14488766	−24305666	−20345166	−32534274
R_{11}	13759678	−9254914	−620878	−17989378
R_{10}	4661438	−2222082	1705650	−1809154
R_9	1237182	−2060802	1175730	−1774850
R_8	1237182	−2060802	1175730	−1774850
R_7	323390	−895874	−54222	−975362
R_6	247870	−310274	−70670	−395842
R_5	210110	−17474	−78894	−106082
R_4	103102	−12434	−95454	−105010
R_3	46214	−4626	−24166	−55938
R_2	7958	−6282	−26850	−22402
R_1	7130	−7624	−10082	−3274
R_0	6095	−7557	−3394	−3589

Table 6: The iterations performed by the CTCR Algorithm 4

The iterations of Algorithm 4 are given in Table 6. We remark that the coefficients of $R = R_0$ satisfy $|r_i| < 2^k = 2^{14} = 16384$.

PMNS to binary: Finally, we can convert R in binary. We obtain $r = 76459417066083$, and we easily check that this is the correct result $r = a\,b \bmod p$.

8 Conclusions

In this paper, we proposed a new representation for the ring of integers modulo p, called Polynomial Modular Number Systems. In this system, integers are represented as polynomials a predefined value γ, of degree less than n, and with coefficients bounded by $(|\alpha| + |\beta|)p^{1/n}$, where α, β are very small integers. Since $p^{1/n}$ is a minimum value, only a few extra bits are required for each coefficient. Compared to the classic multiprecision representation, the polynomial nature of PMNS allows for no-carry propagation, and improved polynomials algorithms. The solutions presented in this paper must be seen as a first step in doing the arithmetic over this new representation. Many improvements are still to come...

References

[1] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997.

[2] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[3] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[4] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2004.

[5] P. L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, April 1985.

[6] P. Barrett. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In A. M. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume 263 of *LNCS*, pages 311–326. Springer-Verlag, 1986.

[7] R. Crandall. Method and apparatus for public key exchange in a cryptographic system. U.S. Patent number 5159632, 1992.

[8] J. Solinas. Generalized mersenne numbers. Research Report CORR-99-39, Center for Applied Cryptographic Research, University of Waterloo, Waterloo, ON, Canada, 1999.

[9] J. Chung and A. Hasan. More generalized mersenne numbers. In M. Matsui and R. Zuccherato, editors, *Selected Areas in Cryptography – SAC 2003*, volume 3006 of *LNCS*, Ottawa, Canada, August 2003. Springer-Verlag. (to appear).

[10] J.-C. Bajard, L. Imbert, and T. Plantard. Modular number systems: Beyond the Mersenne family. Research report 04006, LIRMM – CNRS, 161 rue Ada, 34392 Montpellier cedex 5, France, June 2004. To appear in Selected Areas in Cryptography (SAC 2004).

[11] L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM Publications, 1986.

[12] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems, A Cryptographic Perspective*. Kluwer Academic Publishers, 2002.

[13] C. Dwork. Lattices and their applications to cryptography. Lecture notes, Stanford University, 1998.

[14] A. K. Lenstra, H. W. Jr. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.