Prime Field Multiplication in Adapated Modular Number System using Lagrange Representation

Christophe Negre, Thomas Plantard

Prime Field Multiplication in Adapted Modular Number System using Lagrange Representation

Christophe Negre[(1)], Thomas Plantard[(2)]

(1) *Équipe DALI, LP2A, Unversité de Perpignan*
avenue P. Alduy, F66860 Perpignan, France
christophe.negre@univ-perp.fr
(2) *Centre for Computer and Information Security Research*
School of Computer Science & Software Engineering
University of Wollongong, Australia
thomaspl@uow.edu.au

Abstract. In SAC'04 Bajard *et al.* introduced a new system of representation for integer arithmetic modulo a prime integer P, the Adapted Modular Number System. The multiplication in the AMNS consists of a multiplication of two polynomials and a AMNS-Carrying process. In this paper, we propose to use a *Lagrange Representation* to perform the polynomial multiplication in the AMNS system. This method provides a multiplier which decreases by 2 the complexity of previous best known methods.

1 Introduction

For efficient implementation of cryptographic applications like Diffie-Hellman key-exchange protocol [6] or ECC [10, 9] its is necessary to have efficient modular integer arithmetic. Specifically for Diffie-Hellman key exchange, the main operation is an exponentiation modulo a prime integer P. This operation is generally done using a chain of square and multiply modulo P. For ECC the main operation is a scalar multiplication which requires between 1000 and 2000 additions and multiplications modulo a prime integer.
A multiplication modulo P consists to multiply two integers A and B and after that to compute the remainder modulo P. Methods to perform this operation differ if the integer P has a special form or not. If P is arbitrary, the most used methods are the method of Montgomery [11] and the method of Barrett [3]. The cost of these two methods is roughly equal to the cost of three integer multiplications. If we choose the integer P with sparse binary representation, like generalized Mersenne number [14], or with a pseudo Mersenne form $P = 2^n - \lambda$ with λ small [5], the reduction modulo P can be done really efficiently. Those cases are, for now, the most efficients. To give a brief idea, the cost of the modular multiplication is mainly equal to one integer multiplication, followed by few additions/subtractions. Consequently Standards [12] recommend this type of prime integer.
Recently Bajard, Imbert and Plantard [2] proposed a new method to perform modular arithmetic by using a new representation of integers modulo P, the Adapted Modular Number System. In this system the representation of an integer A modulo P looks like classical γ-adic representation

$$A \equiv \sum_{i=0}^{n-1} a_i \gamma^i \mod P, \text{ with } |a_i| \lesssim P^{1/n}$$

The main difference is that there is a congruence (instead of an equality in classical γ-adic representation). Moreover P and γ are chosen such that $\gamma^n \equiv \lambda \mod P$ with small λ. From experiment [2] such prime P are more abundant than pseudo Mersenne prime.
In this representation the multiplication of A and B is done in two steps: the first step consists to multiply the polynomials $A(\gamma)$ and $B(\gamma)$ modulo $\gamma^n - \lambda$, the second step consists to carry out the product using a *small* AMNS-representation of 2^k.
Our contribution in this paper deals essentially with the polynomial multiplication modulo $E = \gamma^n - \lambda$. Bajard *et al.* only deal with the problem of reduction, they recommend to use classical method (Karatsuba, FFT) to multiply the two polynomials A and B. Thus they have to compute first $C = A \times B$

and after that reduce modulo E. Our approach here consists to keep advantage of the binomial form of E to perform $A \times B \mod E$ once at all. By using Lagrange representation approach we obtain an Algorithm which reduces the amount of complexity of modular multiplication by a factor 2 compared to previous modular multiplication using FFT (AMNS or Mersenne combined with Schonhage-Strassen). This article is organized as follows: in Section 2 and 3 we will recall the AMNS representation and the original AMNS multiplier presented in [2]. Then, in Section 4 we will recall the Lagrange representation [1] and set the Lagrange form of AMNS multiplication. Finally we evaluate the complexity of our algorithm, and compare it with others methods, and finish by a brief conclusion.

2 The Adapted Modular Number System

Modular multiplication consists to multiply two integers $0 \leq A, B < P$ modulo an third integer P. This is done by first computing the product $C = AB$ of the two integer A and B and after that to compute the remainder R of the euclidean division $C = QP + R$.

2.1 Multiplication modulo Mersenne Prime

The efficiency of integer arithmetic is generally closely related to the system used to represent integers. The most used method to represent integer consists to express them as a polynomial in a basis $\gamma \in \mathbb{N}$

$$A = a_0 + a_1\gamma + a_2\gamma^2 + \cdots + a_{n-1}\gamma^{n-1} \tag{1}$$

where a_i lies in $\{0, \ldots, \gamma - 1\}$ and $n = \lceil \log_\gamma(P) \rceil$. In practice, γ is generally equal to 2 or a power of 2 due to binary representation in computer systems. In the sequel we will prefer to rewrite (1) in the indeterminate X

$$A(X) = a_0 + a_1X + a_2X^2 + \cdots + a_{n-1}X^{n-1} \tag{2}$$

We get back to the integer A by evaluating $A(X)$ in γ. We will also use the notation $\|A(X)\|_\infty = \sup\{|a_i|, i = 0, \ldots, n-1\}$ for a polynomial $A(X)$.

In this representation, multiplication is just a polynomial multiplication followed by carrying process. Specifically, the product of $A(X)$ and $B(X)$ gives a polynomial $C(X)$ of degree $2n-2$ with coefficients satisfying

$$\|C\|_\infty < n\gamma^2.$$

The carrying process consists to split the c_i in $c_i = \underline{c}_i + \gamma\overline{c}_i$ with $0 \leq \underline{c}_i < \gamma$ and then to add the upper part $\overline{c}_i$ to c_{i+1}. Figure 2.1 illustrates this process.

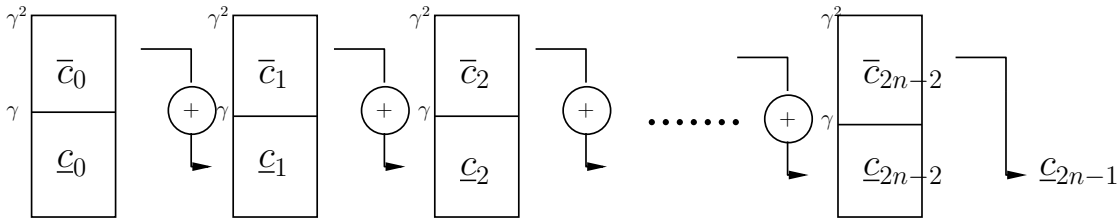

Fig. 1. Carrying

If P has a sparse expression as γ-polynomial, the reduction modulo P can be done efficiently. This is the case of pseudo Mersenne prime [5] where P is such that $P = \gamma^n - \lambda$ with λ is small relatively to P. Then if C is the carried out product of A and B it has degree at most $2n-1$. The reduction consists to reduce the degree of C in such that C has degree less than n. To reduce C it suffices to decompose C as

$$C(X) = \underline{C}(X) + X^n\overline{C}(X) \text{ with } \deg(\underline{C}), \deg(\overline{C}) < n$$

and use the identity $\gamma^n = \lambda \mod P$

$$C \equiv \underline{C} + \lambda \overline{C} \mod P$$

The multiplication by λ is not costly since λ is small. By repeating this process a number of time sufficient we get the reduced form of C.

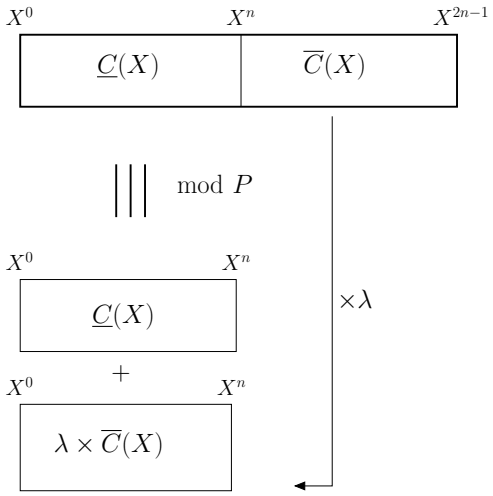

Fig. 2. Mersenne reduction process

For cryptographic application, integer P is generally quite big (200-500 bits for elliptic curve cryptography, 2000-4000 bits for Diffie-Hellman in the multiplicative group). Mersenne prime are relatively rare. Because of this small number of Mersenne P, it is an important challenge to find other type of prime which provides reduction as simple as Mersenne prime.

2.2 Adapted Modular Number System

In 2003 Bajard *et al.* proposed a new type of representation, the Adapted Modular Number System (AMNS), which gives alternative primes for efficient modular arithmetic. In AMNS representation, an integer A is expressed again as a polynomial in an integer γ

$$A \equiv a_0 + a_1\gamma + a_2\gamma^2 + \cdots + a_{n-1}\gamma^{n-1} \mod P$$

The main difference is that they want A to be *equivalent modulo* P to the polynomial, not any more to be equal to the polynomial. Moreover the size of γ and the a_i are not correlated. Specifically, in AMNS the coefficients a_i lies (roughly) in $[0, P^{1/n}]$ and γ could be bigger than $P^{1/n}$ (γ generally of size $\gamma \cong P$). To get a Mersenne like multiplication modulo P, Bajard *et al.* chose γ satisfying

$$\gamma^n \equiv \lambda \mod P \text{ with } \lambda \text{ small}$$

Consequently to have suitable prime P to build AMNS, we must have $P|\gamma^n - \lambda$, but not any more an equality $P = \gamma^n - \lambda$, as it was the case for pseudo Mersenne prime. The constraint on P and γ is thus less strict, and thus such prime are more abundant.

Definition 1 (AMNS [2]). *An Adapted Modular Number System $\mathcal{B}$, is an array $\mathcal{B} = (P, n, \rho, \gamma, \lambda, M)$ such that*

1. *n represents polynomial length of the AMNS.*
2. *λ is an integer (possibly negative) small compared to P.*

3. *γ is an integer which satisfies $0 \leq \gamma < P$ and $\gamma^n - \lambda \equiv 0 \mod P$.*
4. *$\rho = 2^k$ is the bound on the coefficient, it has generally a size close to $P^{1/n}$.*
5. *$M(X)$ is a degree $(n-1)$ sparse polynomial with coefficients in $\{-1, 0, 1\}$ and satisfies*

$$2^k \equiv M(\gamma) \mod P. \tag{3}$$

In the sequel, we will call $M(X)$ the carrying polynomial of the AMNS.
6. *For all positive integers $0 \leq A < P$ there exists a polynomial $A_{\mathcal{B}}(X)$ such that*

$$A_{\mathcal{B}}(\gamma) \equiv A \bmod P, \quad \deg(A_{\mathcal{B}}) < n, \quad \|A_{\mathcal{B}}\|_\infty < \rho \tag{4}$$

Polynomial $A_{\mathcal{B}}$ is a representation of A in $\mathcal{B}$.

In the sequel we will generally use the same notation for the integer A and its AMNS representation $A_{\mathcal{B}}$, i.e., we will often omit the subscript $\mathcal{B}$. We will also denote by E the polynomial $E = X^n - \lambda$.

Example 1. Let us consider the following system

$$\mathcal{B} = (p = 41, n = 4, \rho = 2, \gamma = 27, \lambda = -1, M = -1 + X^3).$$

We can check that this $\mathcal{B}$ satisfies the condition of Definition 1. We have

$$\gamma^n - \lambda \equiv 27^4 + 1 \equiv 0 \pmod{41}$$

and $M(\gamma) \equiv -1 + 27^3 \equiv \rho \pmod{41}$.
Table 1 gives AMNS representation of integer in the range $[0, 20]$. Other integers A can be obtained as the opposite of the AMNS representation of $P - A$.

0	1	2	3	4	5
0	1	$-1+X^3$	X^3	$1+X^3$	$-1-X^2-X^3$

6	7	8	9	10	11
$-X^2-X^3$	$1-X^2-X^3$	$-1-X^2$	$-X^2$	$1-X^2$	$-1-X^2+X^3$

12	13	14	15	16	17
$-X^2+X^3$	$1+X^2+X^3$	$-1+X+X^2-X^3$	$X+X^2-X^3$	$1+X+X^2-X^3$	$-1+X+X^2$

18	19	20
$X+X^2$	$1+X+X^2$	$-1+X+X^2+X^3$

Table 1. Elements of $\mathbb{Z}_{41}$ in $\mathcal{B} = AMNS(41, 4, 2, 27, -1, -1 + X^3)$

To check that an AMNS $A_{\mathcal{B}}$ is correct, we have just to evaluate $A_{\mathcal{B}}$ in γ modulo P and check that the result is equal to A. For example, if we evaluate $(X - X^2 - X^3)$ in γ, we have $27 - 27^2 - 27^3 = -20385 \equiv= 33 \bmod 41$. We have also $\deg(X - X^2 - X^3) = 3 < 4 = n$ and $\|X - X^2 - X^3\|_\infty = 1 < 2$ as required. $\Diamond$

In Table 2 we propose a list of prime integer with corresponding AMNS parameter. These primes have suitable cryptographic length (between 48 bits and 15872 bits). We only give the most interesting prime numbers. The complete AMNS, including prime number and root γ, are given in Appendix A. Moreover, experimental result shows that AMNS prime are more dense than Mersenne primes.

Table 2. Proposition of efficient prime field.

$\|p\|$	n	λ	E	ρ	M	m^1
48	8	-1	X^8+1	2^6	$X-X^2$	$2^{16}+1$
112	8	-1	X^8+1	2^{14}	$X+X^4$	$2^{32}+1$
208	16	-1	$X^{16}+1$	2^{13}	$X-X^8$	$2^{32}+1$
465	16	-1	$X^{16}+1$	2^{29}	$-1+X+X^8$	$2^{64}+1$
927	32	-1	$X^{32}+1$	2^{29}	$X+X^{20}$	$2^{64}+1$
1952	32	-1	$X^{32}+1$	2^{61}	$X-X^{18}$	$2^{128}+1$
3840	64	-1	$X^{64}+1$	2^{60}	$1+X-X^{34}$	$2^{128}+1$
7937	64	-1	$X^{64}+1$	2^{124}	$-1+X+X^{56}$	$2^{256}+1$
15872	128	-1	$X^{128}+1$	2^{124}	$1+X-X^{24}$	$2^{256}+1$

3 AMNS Multiplication

In an AMNS $\mathcal{B}=(P,n,\rho,\gamma,\lambda,M)$, the multiplication is done in a similar way as in classical multiplication modulo Mersenne prime. The main difference is in the carrying process. We perform first the multiplication of two polynomials $A(X)$ and $B(X)$

$$C(X)=A(X)\times B(X)$$

we get a polynomial $C(X)$ with $\|C\|_\infty \leq n\|A\|_\infty\|B\|_\infty \leq n\rho^2$. Then, we perform the polynomial reduction modulo $E=X^n-\lambda$. This operation is done in the same way as in Generalized Mersenne prime case (Figure 2.1) . We get

$$R=C \mod E$$

and at this step

$$\|R\|_\infty \leq 2n|\lambda|\rho^2. \tag{5}$$

The AMNS-carrying process is done only after this stage. It is not possible to perform a classical carrying here since γ is generally big, i.e., it is not possible to write

$$r_i=\underline{r}_i+\gamma\overline{r}_i \text{ with } \underline{r}_i\leq\rho.$$

In AMNS, instead of splitting r_i relatively to γ, we split it relatively to $\rho=2^k$

$$r_i=\underline{r}_i+2^k\overline{r}_i \text{ with } \underline{r}_i\leq\rho=2^k. \tag{6}$$

From equation (3) of the AMNS definition, we have $2^k\equiv M(\gamma) \mod P$. If we replace 2^k by $M(\gamma)$ in equation (6) we get

$$r_i\equiv\underline{r}_i+\left(\sum_{j=0}^{n-1}m_j\gamma^j\right)\overline{r}_i \mod P$$

Consequently, the carry is not any more added uniquely to the next term r_{i+1} but is added to the r_{i+k} such that $m_k=1$. We can rewrite this in a polynomial form

$$R=\underline{R}+2^k\overline{R}\equiv\underline{R}+\underbrace{\left(\sum_{j=0}^{n-1}m_jX^j\right)\overline{R}}_{(*)}$$

The $(*)$ part are the carries. In Figure 3 we give an arithmetic circuit for the carrying in the special case $M(X)=X+X^2$.
The carrying process must be performed several times until we get $\|R\|_\infty\leq\rho=2^k$. In their paper, Bajard *et al.* gives an explicit bound on the number of required carrying process to get $\|R\|_\infty\leq\rho$. When M is really sparse this number is equal to 2 or 3. Finally, the AMNS-multiplication is sum up in Algorithm 1.

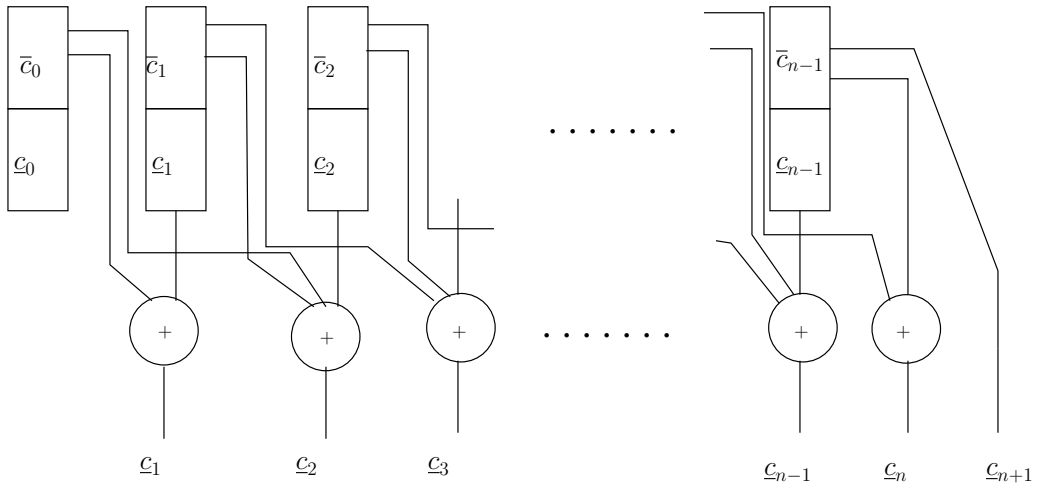

Fig. 3. AMNS Coefficients Reductions Process

Algorithm 1: AMNS-Multiplication

Input : An AMNS $\mathcal{B} = (P, n, \rho, \gamma, \lambda, M)$ A, B two polynomials such that $A, B \in \mathcal{B}$
Output: R a polynomial such that $R \in \mathcal{B}$ and $R(\gamma) \equiv A(\gamma)B(\gamma) \pmod{P}$
begin
  $R \leftarrow A \times B \bmod E$;
  while $\|R\|_\infty \geq \rho$ **do**
    $R = \underline{R} + 2^k \overline{R}$ with $\|\underline{R}\|_\infty < 2^k$;
    $R \leftarrow \underline{R} + M \times \overline{R} \bmod E$;
  end
end

Example 2. We give here an example of a AMNS multiplication using Algorithm 1. We propose to multiply $A = 17$ and $B = 23$ modulo $P = 41$ using the AMNS given in Example 1. The AMNS representation of $A = 17$ is $A_{\mathcal{B}} = -1 + X + X^2$ and the representation of $B = 23$ is $B_{\mathcal{B}} = -1 + X - X^3$. The first part of the algorithm computes $R \leftarrow A_{\mathcal{B}} \times B_{\mathcal{B}} \bmod E$ where $E = X^4 + 1$

$$R \leftarrow (-1 + X + X^2) \times (-1 + X - X^3) \bmod X^4 + 1 = 2 - X + 2X^3$$

We have $\|R\|_\infty = 2 \geq \rho = 2$, so we have to perform a carrying process on R. We split R as $R = \underline{R} + 2 \times \overline{R}$ where

$$\overline{R} = 1 + X^3, \quad \underline{R} = -X.$$

We then get

$$\begin{aligned} R \leftarrow \underline{R} + M \times \overline{R} \bmod E &\equiv -X + (-1 + X^3) \times (1 + X^3) \bmod (X^4 + 1) \\ &\equiv -X + (-1 + X^6) \bmod (X^4 + 1) \\ &\equiv -1 - X + X^6 \bmod (X^4 + 1) \\ &\equiv -1 - X - X^2 \bmod (X^4 + 1) \end{aligned}$$

At this step we have $\|R\|_\infty = 1 < \rho = 2$, thus the algorithm quit the loop `While` and return $R = -1 - X - X^2$. The polynomial R is the correct AMNS representation of $22 = A \times B \mod P$ since

$$R(\gamma) \bmod P = -1 - 27 - 27^2 \equiv -757 \equiv 22 \bmod 41.$$

$\Diamond$

Bajard *et al.* only focused on the carrying process. For the polynomial multiplication step, they said that it can be performed with the best know method: mainly school-book for very low degree n, Karatsuba [8] for intermediate degree, and FFT approach for larger degree.
Our contribution concerns the polynomial multiplication part in AMNS-multiplication . We will see that we can get benefit of the binomial form of E to improve the FFT approach of polynomial multiplication.

4 Lagrange AMNS Multiplication

The AMNS multiplication (Algorithm 1) requires one polynomial multiplication modulo $E = X^n - \lambda$. We study in this section a modified version of Algorithm 1 which uses a Lagrange representation of the polynomials. This representation provides a polynomial multiplication which includes the reduction modulo E. We will see that this approach decrease the complexity of multiplication using FFT approach by a factor 2. We begin by a brief review on Lagrange Representation [1].

4.1 Lagrange Representation

The Lagrange representation consists to represent a polynomial by its values at n points, the roots of $E = \prod_{i=1}^{n}(X - \alpha_i)$ modulo an integer m. In an arithmetic point of view, this is related to the Chinese remainder Theorem which asserts that the following application is an isomorphism

$$\begin{array}{ccc} \mathbb{Z}/m\mathbb{Z}[X]/(E) & \longrightarrow & \mathbb{Z}/m\mathbb{Z}[X]/(X-\alpha_1) \times \cdots \times \mathbb{Z}/m\mathbb{Z}[X]/(X-\alpha_n) \\ A(X) & \longmapsto & (A \bmod (X-\alpha_1), \ldots, A \bmod (X-\alpha_n)) \,. \end{array} \tag{7}$$

We remark that the computation of $A \mod (X - \alpha_i)$ is simply the computation of $A(\alpha_i)$. In other words the image of $A(X)$ by the isomorphism (7) is nothing else that the multi-points evaluation of A at the roots of E.

Definition 2 (Lagrange representation). *Let $A \in \mathbb{Z}[X]$ with $\deg A < n$, and $\alpha_1, \ldots, \alpha_n$ the n distinct roots modulo m of $E(X)$*

$$E(X) = \prod_{i=1}^{n}(X - \alpha_i) \mod m.$$

If $a_i = A(\alpha_i) \mod m$ for $1 \leq i \leq k$, the Lagrange representation A_{LR} of $A(X)$ modulo m is defined by

$$A_{LR} = (a_1, \cdots, a_n). \tag{8}$$

The advantage of the Lagrange representation to perform operations modulo E is a consequence of the Chinese remainder Theorem. Specifically arithmetic modulo E in classical polynomial representation can be costly if E has a high degree. In Lagrange representation this arithmetic is decomposed into n independent arithmetic units. Each consists of arithmetic modulo a very simple polynomial $(X-\alpha_i)$. But arithmetic modulo $(X-\alpha_i)$ is the arithmetic modulo m since the product of two degree zero polynomials is just the product modulo m of the two constant coefficients. In other words, the multiplication of two polynomials A and B is done as follows

$$A_{LR} \times B_{LR} = (a_1 \times b_1, a_2 \times b_2, \ldots, a_n \times b_n). \tag{9}$$

Example 3. We consider here $m = 17$ and the polynomial $E = X^2 + 1 = (X + 4)(X + 13) \mod m$. Let $A = 15 + 11X$ and $B = 7 + 5X$ their corresponding Lagrange representation are

$$A_{LR} = (8, 5), \quad B_{LR} = (10, 4).$$

If we compute

$$C_{LR} = A_{LR} \times B_{LR} = (12, 3)$$

we can check that C_{LR} is the correct Lagrange representation of $C = A \times B \mod E = 16X + 16$.

4.2 Improved AMNS multiplication using Lagrange representation

Let us go back to Algorithm 1 and let us see how to use Lagrange representation to perform polynomial arithmetic in AMNS multiplication. In view to use Lagrange representation, we select an integer m such that the polynomial $E = (X^n - \lambda)$ splits in $\mathbb{Z}/m\mathbb{Z}[X]$

$$E = \prod_{i=1}^{n}(X - \alpha_i) \mod m.$$

We can then represent the polynomials A and B in Lagrange representation. The polynomial multiplication $R = A(X) \times B(X) \mod E$ in Algorithm 1 can be then done using equation (9). From equation (5) we see that if m is such that $m > 2n\rho^2 \geq \|R\|_\infty$, there is no loss in the coefficients of R. When we interpolate the Lagrange representation of R we get the correct polynomial expression of R.

Algorithm 2: Lagrange-AMNS Multiplication

Input : $A_{LR}, B_{LR}, \mathcal{B} = (P, n, \rho, \gamma, \lambda, M)$
Output: R_{LR} such that $R \in \mathcal{B}$ and $R(\gamma) = A(\gamma)B(\gamma) \bmod P$
begin
  $R_{LR} \leftarrow A_{LR} \times B_{LR}$;
  $R \leftarrow Convert_{LR \to Pol}(R_{LR})$;
  while $\|R\|_\infty \geq \rho$ **do**
    $R = \underline{R} + 2^k \overline{R}$ with $\|\underline{R}\|_\infty < 2^k$;
    $R \leftarrow \underline{R} + M \times \overline{R} \bmod E$;
  end
  $R_{LR} \leftarrow Convert_{Pol \to LR}(R)$;
end

After the multiplication $A_{LR} \times B_{LR}$, we must perform the carrying process on R. It is not possible to do this through the Lagrange representation of R. We must re-construct the polynomial form of R, carry out on R and go back to Lagrange representation.

4.3 Conversion

To have a complete and efficient AMNS multiplication in Lagrange representation we need to make explicit how we perform the conversions between Lagrange and Polynomial representations.

We first deal with the computation of the Lagrange representation A_{LR} from the polynomial representation of A. We will use for this a property of the roots of the binomial polynomial $E = X^n - \lambda$. Let $\alpha_1, \ldots, \alpha_n$ be the roots of E, the n distinct elements α_i/α_0 are the n-th roots of unity in $\mathbb{Z}/(m\mathbb{Z})$

$$(\alpha_i/\alpha_0)^n \equiv \lambda/\lambda \equiv 1 \mod m$$

Thus if we set $\mu = \alpha_0$ and if ω a primitive n-th root of unity in $\mathbb{Z}/(m\mathbb{Z})$, we can rewrite the α_i (even after a reordering) as

$$\alpha_i = \mu\omega^i \mod m.$$

The computation of the Lagrange representation of a polynomial $A(X) = \sum_{i=0}^{n-1} a_i X^i$ can be done as follows

1. $\widetilde{A}(X) = A(\mu X) = \sum_{i=0}^{n-1} \mu^i a_i X^i$
2. $DFT_n(\widetilde{A}) = (\widetilde{A}(1), \widetilde{A}(\omega), \widetilde{A}(\omega^2), \ldots, \widetilde{A}(\omega^{n-1}))$

Here DFT_n refer to the so-called Discrete Fourier Transform at the n-th root of unity.
We can check that this process is correct. We have $\widetilde{A}(\omega^i) \equiv A(\mu\omega^i) \equiv A(\alpha_i) \mod m$, i.e., the n-uplet $DFT(\widetilde{A})$ is the correct LR representation of A.

The reverse conversion, from Lagrange to polynomial, can be simply done by reversing the previous process. From A_{LR} we compute

$$\widetilde{A} = DFT_n^{-1}(A_{LR}) = \frac{1}{n} DFT_n(A_{LR})$$

and then we get

$$A(X) = \widetilde{A}(\mu^{-1}X) \mod m.$$

Finally the cost of each conversion is equal to $n-1$ multiplication modulo m and one DFT computation. This is really interesting when the integer n is a power of 2 since in this case we can use the Fast Fourier Transform for the DFT computation.

4.4 Fast Fourier Transform.

We briefly recall how the FFT works. For a detailed presentation on FFT we refer to the book of Gathen *et al.* [7]. The Fast Fourier Transform is a recursive algorithm which computes the DFT of a polynomial A at the $n = 2^\ell$ roots of unity. Let us denote ω a primitive n-th root of unity, and $\widehat{a}_i = A(\omega^i)$ the i-th coefficient of the DFT of A. The FFT is based on the following property

$$\begin{array}{c} \hat{a}_i = A(\omega^i) = A_1((\omega^2)^i) + \omega^i A_2((\omega^2)^i), \\ \hat{a}_{i+(n/2)} = A(\omega^{i+n/2}) = A_1((\omega^2)^i) - \omega^i A_2((\omega^2)^i), \end{array} \tag{10}$$

where the polynomial $A_1(X)$ is the even part of A and $A_2(X)$ is the odd part of A

$$A_1(X) = \sum_{i=0}^{n/2-1} a_{2i} X^i, \quad A_2(X) = \sum_{i=0}^{n/2-1} a_{2i+1} X^i.$$

In equation (10), we remark that the computation of $A_1((\omega^2)^i)$ and $A_2((\omega^2)^i)$ are common to $\hat{a}_i$ and $\hat{a}_{i+(n/2)}$. Consequently, to compute $FFT_n(A)$, the two arrays $FFT_{n/2}(A_1)$ and $FFT_{n/2}(A_2)$ are recursively computed and after that we deduce $FFT_n(A)$ using relations (10).

Algorithm 3: FFT [4]

Input : ω a primitive $n = 2^\ell$-th root of unity modulo m, and $A = [a_0, a_1, \ldots, a_{n-1}]$ the coefficients of a degree $n-1$ polynomial A in $\mathbb{Z}/m\mathbb{Z}[X]$.
Output: $\widehat{A} = (\widehat{a}_1, \ldots, \widehat{a}_n)$ the DFT_n of A.

```
if n = 0 then
    FFT_n(A) ← A
else
    A_1 ← [a_0, a_2, ..., a_{2(n-1)}];
    R_1 ← FFT_{n/2}(A_1);
    A_2 ← [a_1, a_3, ..., a_{2n-1}];
    R_2 ← FFT_{n/2}(A_2);
    for i = 0, ..., n/2 - 1 do
        λ ← ω^i R_2[i] mod m;
        Â[i] ← (R_1[i] + λ mod m);
        Â[i + n/2] ← (R_1[i] - λ mod m);
    end
end
```

The complexity of a FFT_n is equal to

$$\frac{n}{2}\log_2(n)M + n\log_2(n)A \tag{11}$$

where M denote a multiplication and A an addition or a subtraction modulo m.

On the use of Fermat number for m.
FFT involves many multiplications with root of unity. One strategy to get efficient FFT is to use m such that product with roots of unity can computed efficiently. Following Schonhage-Strassen idea [13], we can reach this goal by taking m as Fermat number $2^{2^\ell}+1$. Modulo such m the $2^{\ell+1}$-roots of unity are power of 2. Consequently the multiplication modulo m of an integer a by a $2^{\ell+1}$-root of unity consists of simple shift of binary representation, followed by truncation and subtraction to reduce it modulo m. Specifically, if we set $\omega = 2$, then $c = a\omega^i \mod m$ is computed as follows

$$\begin{aligned} c &= a\omega^i \mod m \\ &= ((a \uparrow i) \mod 2^{n/2}) - ((a \uparrow i)/2^{n/2}), \end{aligned}$$

where $(a \uparrow i)$ is the integer a shifted by i places. Consequently any multiplication by ω^i has a cost of one addition.
Using this strategy, the overall complexity of the FFT is equal to $\frac{3}{2}\log(n)$ additions/subtractions of integer of length $\log_2(m)$.
In conversions between Lagrange to Polynomial, several multiplications by μ (an arbitrary root of E) are required. If $E = X^n + 1$ with Fermat moduli $m = 2^n + 1$, such μ can be taken as $\mu = 2$. As above, multiplication modulo μ has a cost of one subtraction modulo m. The cost of one conversion is thus equal to

$$\left(\frac{3n}{2}\log_2(n) + n - 1\right) A \tag{12}$$

5 Example of Lagrange-AMNS multiplication

In this section we present an example of Lagrange-AMNS multiplication. The AMNS used is the system given in Example 1

$$\mathcal{B} = (p = 41, n = 4, \rho = 2, \gamma = 27, \lambda = -1, M = -1 + X^3),$$

with Lagrange representation in the roots of $E = X^4 + 1$ modulo $m = 2^4 + 1$

$$E \equiv (X-2)(X-8)(X-9)(X-15) \mod 17.$$

The two elements μ and ω used in the conversion processes are here $\mu = 2$ and $\omega = 4$.
In Table 3, we give an example of an execution of the Lagrange-AMNS multiplication in this AMNS for the two elements

$$A = -1 + X + X^2,$$
$$B = -1 + X - X^3.$$

Table 3. Example of AMNS Multiplication

Entries	
A_{LR}	$(5, 3, 1, 4)$
B_{LR}	$(10, 5, 5, 10)$

Multiplication	
$R_{LR} = A_{LR}B_{LR}$	$(16, 15, 5, 6)$
$R = Convert_{LR\to Pol}(R_{LR})$	$2 - X + 2X^3$
$R = CarriedOut(R)$	$-1 - X - X^2$
$R_{LR} = Convert_{Pol\to LR}(R)$	$(10, 12, 14, 11)$

We can check that R is the correct result since

$$A(\gamma)B(\gamma) \mod P = R(\gamma) \mod P.$$

$\Diamond$

6 Complexity

In this section we evaluate the complexity of the Lagrange-AMNS multiplication (Algorithm 2).
We compare it to best methods based on FFT, known from the authors, to perform multiplication modulo prime integer. These are the following

1. Multiplication with Schonhage-Strassen algorithm [13] and reduction modulo a pseudo Mersenne prime.
2. Multiplication in AMNS using FFT for the product of Polynomial.

We first recall how these two methods proceed.

Schonhage-Strassen method and Mersenne reduction
This method follows the approach depicted in Section 2. An integer A modulo P are expressed as polynomial of degree $n-1$ in $\gamma = 2^k$ such that $n, k \cong \sqrt{\log_2(P)}$

$$A = \sum_{i=0}^{n-1} a_i \gamma^i \quad \text{with } 0 \leq a_i \leq \gamma.$$

We use here the Schonhage-Strassen algorithm compute the product of polynomial $C(X) = A(X)B(X)$. The carrying process and a reduction modulo a Mersenne is performed on the resulting polynomial C. To perform the product $A(X) \times B(X)$, Schonhage-Strassen's method use a Lagrange approach. Lagrange representation is used with the roots of $E = X^{2n} - 1$ modulo the Fermat $m = 2^{2n} + 1$. And it uses a similar approach to Lagrange-AMNS multiplication. Specifically

1. They compute $C_{LR} = A_{LR} \times B_{LR}$ where each $a_i b_i$ is computed recursively with recursive Schonhage-Strassen method.
2. Then they get back to the polynomial form of C by computing

$$C(X) = \frac{1}{2n} FFT_{2n}^{-1}(C_{LR})$$

3. They get $R = AB \mod P$ by performing carries on C and reduction modulo P
4. They compute the Lagrange representation of R by computing

$$R_{LR} = FFT_{2n}(R)$$

Remark 1. Original Schonhage-Strassen Algorithm was for integer multiplication. Consequently A and B was given in binary form, not in Lagrange and C was also kept in binary form. FFT was applied on $A(X)$ and $B(X)$ to obtain their Lagrange representation. Here to reduce the number of FFT computation, we keep A and B in Lagrange representation. This is interesting if sufficient number of multiplications is done modulo P, this is the case for example in cryptographic applications.

Remark 2. We must mention that, for the sake of simplicity, we give a simplified version of Schonhage-Strassen Algorithm. Specifically, due to the growth of coefficient c_i could be equal $n\gamma^2$. In Schonhage-Strassen algorithm, additional computation of $A(X) \times B(X)$ modulo n is done to get the full coefficients of C.

AMNS with FFT
The straightforward strategy to perform AMNS multiplication with FFT consists to proceed as in Schonhage-Strassen case.
We perform the product $C(X) = A(X)B(X)$ modulo $E = (X^{2n} + 1)$ in $\mathbb{Z}/m\mathbb{Z}[X]$. The integer m and n are chosen as in Schonhage-Strassen algorithm. After we reduce C modulo $E = X^n - \lambda$ given in the AMNS and carry out R. In this situation we can again stay in Lagrange representation, in view to minimize the number of FFT computation. We obtain an AMNS-Multiplication which has same form as Schonhage-Strassen modulo Mersenne prime.

Complexity and Comparison
We assume that we use AMNS such that n is the as in Schonhage-Strassen multiplication, for prime moduli of the same size. We then can use the same modulo m as in Schonhage-Strassen algorithm. Under this assumption a quick comparison show that Lagrange-AMNS multiplication differ from the two other

Operation	Schonhage-Strassen & Mersenne	AMNS-FFT	Lagrange-AMNS	
			$\lambda \neq -1$	$\lambda = -1$
Lagrange Multiplication	$2nM$	$2nM$	nM	nM
$Convert_{Lag \to Pol}$	$3n\log_2(n)A$	$3n\log_2(n)A$	$\frac{3n}{2}\log_2(n)A + (n-1)M$	$(\frac{3n}{2} + (n-1))A$
Carrying	$n\kappa A$	$n\kappa' A$	$n\kappa' A$	$n\kappa' A$
$Convert_{Lag \to Pol}$	$3n\log_2(n)A$	$3n\log_2(n)A$	$\frac{3n}{2}\log_2(n)A + (n-1)M$	$(\frac{3n}{2}\log_2(n) + (n-1))A$
Total	$2nM$ $+n(\kappa + 6\log_2(n))A$	$2nM$ $+n(\kappa' + 3\log_2(n))A$	$(3n-2)M$ $+n(\kappa' + 3\log_2(n))A$	nM $+n(\kappa' + 3\log_2(n) + 1)A$

method, by the degree of E. In other word, in Lagrange-AMNS we use a Lagrange representation of length n, instead of $2n$ in others methods.
We give the cost of each step of (simplified) Schonhage-Strassen, AMNS-FFT and Lagrange-AMNS in Table 6. The complexity is expressed in term of the number of additions (A) and multiplications (M) of integer of bit-length $\log_2(m)$. For Lagrange-AMNS we evaluate the cost when $\lambda = -1$ and $\lambda \neq -1$, since the conversion is cheaper in the former case.
The constants κ, κ' are related to the size of λ and the hamming weight of $M(X)$. Generally these constants lies are less than 5.
Table 6 shows that Lagrange-AMNS multiplication is better, when $\lambda = -1$, by a factor 2 than Schonhage-Strassen modulo Mersenne or AMNS-FFT.

7 Conclusion

In this paper we have presented a modified AMNS Multiplication for integer multiplication modulo a prime. In the approach of Bajard *et al.* [2], we modified the polynomial multiplication part of AMNS multiplication. We used a Lagrange Representation combined with FFT for conversion. We obtain an algorithm which has a complexity two times less than Schonhage-Strassen approach for multiplication modulo Mersenne prime or previous AMNS-FFT approach.

References

1. J.-C. Bajard, L. Imbert, C. Negre, and T. Plantard. Efficient multiplication in GF (p^k) for elliptic curve cryptography. In *ARITH'16: IEEE Symposium on Computer Arithmetic*, pages 181–187, June 2003.
2. J.-C. Bajard, L. Imbert, and T. Plantard. Modular number systems: Beyond the Mersenne family. In *SAC'04: 11th International Workshop on Selected Areas in Cryptography*, pages 159–169, August 2004.
3. 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.
4. G Brassard, S Monet, and D Zuffellato. Algorithms for very large integer arithmetic. *Tech. Sci. Inf.*, 5(2):89–102, 1986.
5. R. Crandall. Method and apparatus for public key exchange in a cryptographic system. U.S. Patent number 5159632, 1992.
6. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.
7. J. V. Z. Gathen. *Modern Computer Algebra*. Cambridge University Press, 1999.
8. A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Doklady Akademii Nauk SSSR*, 145(2):293–294, 1962.
9. N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.
10. V. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology, proceeding's of CRYPTO'85*, volume 218 of *LNCS*, pages 417–426. Springer-Verlag, 1986.

11. P. L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, Apr 1985.
12. National Institute of Standards and Technology. *FIPS PUB 197: Advanced Encryption Standard (AES)*. FIPS PUB. National Institute for Standards and Technology, November 2001.
13. A. Schonhage and V. Strassen. Schnelle multiplikation grosser zahlen. *Computing*, 7:281–292, 1971.
14. J. Solinas. Generalized Mersenne numbers. Research Report CORR-99-39, Center for Applied Cryptographic Research, University of Waterloo, Waterloo, ON, Canada, 1999.

A Appendix: AMNS

In this section we give a list of practical AMNS basis associated to prime integer.

A.1 AMNS for modulo 48bits

1. $p = 140737503076097$
2. $n = 8$
3. $\rho = 2^6$
4. $\gamma = 85613165376383$
5. $\lambda = -1$
6. $M[X] = X - X^2$
7. $m = 2^{16} + 1$

A.2 AMNS for modulo 112bits

1. $p = 2596148467953040258123756591841281$
2. $n = 8$
3. $\rho = 2^{14}$
4. $\gamma = 84309851953528350105183947308107 1$
5. $\lambda = -1$
6. $M[X] = X + X^4$
7. $m = 2^{32} + 1$

A.3 AMNS for modulo 208bits

1. $p = 20568809418508068793756365771907968539747717254211001866472 6529$
2. $n = 16$
3. $\rho = 2^{13}$
4. $\gamma = 18344936802441593432126303015454000671894920928467078165490 6812$
5. $\lambda = -1$
6. $M[X] = X - X^8$
7. $m = 2^{32} + 1$

A.4 AMNS for modulo 465bits

1. $p = 476341040550437977226889013263855019900816493318155715779109819340797244981303787403466964296146462127684681000607912216427575264424818117 13$
2. $n = 16$
3. $\rho = 2^{29}$
4. $\gamma = 147541146931416837760636585975654116248513189291020743754228702987199238047697960747848363942900511063871381708355709170610546789725355265 55$
5. $\lambda = -1$
6. $M[X] = -1 + X + X^8$
7. $m = 2^{64} + 1$

A.5 AMNS for modulo 927bits

1. $p = 1134503866941667986143541334648056457619674836471095626110651666151475317998414976158886313332194139280000745437282626350934920222689991296360993051395248134864246433590796851410700593214806316800878230397904801714192854102042879703637145515529947341809434176688910346492219228 17$
2. $n = 32$
3. $\rho = 2^{29}$
4. $\gamma = 3827886604388370780755694051731008334471652743466079294972382820576212256317777717575967548728443693383325007221387499917590520592854879150776965884385059674658861403923609131746896534525943060008784373842812776630881213062246301365612529780207388158871855567306030083938626701687$
5. $\lambda = -1$
6. $M[X] = X + X^{20}$
7. $m = 2^{64} + 1$

A.6 AMNS for modulo 1952bits

1. $p = 2039489813075786849042062700437018179439827928847064698341360754386937268720406126932437527295338721555221831918536179936735519932368019857850660791894365049954451711762131064978595720625229858144448159446877808292625233152531998466014812755051894615557985963293380314509582246623088344635766553131390435480904187275288490093524874232458644410365414758493986844341335604367736795192927346121444105428396600892490352243707163622886965218086640958467292014056634513669602784194228374468070266275480515126443616787299279854084897024467326277100633806268368954744488882142438791829796453089291 6$
2. $n = 32$
3. $\rho = 2^{61}$
4. $\gamma = 1955809704526156206751076423416199531925611556278560904133086337820207738710845546062375284506130566580385855582668296208328440498140932421004374816104292142700442132701501819648421553546301432017021906892087538458774509279066450932394754354473345792873313361266823994918564963913190682807777482998067170985980779375693083002252649260680392538274953706934291129011455552264162968961425535292344535741573251925417422592921943493606572971029820253498600558067481450610955593750830286860413309584746837342037393600206885465761944409003535143499635747099525214253074381752515893732320651984$
5. $\lambda = -1$
6. $M[X] = X - X^{18}$
7. $m = 2^{128} + 1$

A.7 AMNS for modulo 3840bits

1. $p = 9019518416950528057689627210055266123393610844964668360284683176962644432039216216192510279822647400987309397692077381271507065837794290762477229873784676597380409623197140800269512856156220572795289915009075897579550499180911573343940075144419318535768436382007055308727852745873725045682248114516251605782926506218294493811294563330113486982036416246475131066201672706309421001691995080893476389049108604848034950527921927240195762261968654783705521595670014908604579007885360154828126036413280110978189460780994891938053631664963667716863284057951438373085772809257846150334100220750460875820010933681241740447383491879963048594196658939361732310253854635863490646192669301323857895894799520658466357573127804950025771938994041945627997209521632484808858657669125785178232558272961145255223570186866192093547673966852929937177009734567208306498690877295894972808779193432166783287366181825536723164054273492720855276519598989741071405977458430754317761852800482581570231515443652779587762874834196310056072815817865261953601868765293458540366428500019296151137384146427759013268227953443015222230891179575120625918454784220248009260865024085190697275777$
2. $n = 64$
3. $\rho = 2^{60}$

4. $\gamma = 32742517420076235904398012880166621755630816981325833651250972539538516177014035904$
$919275874687364864616033102953330415210685807456559401476422077071293369534280361561736$
$555919839328750088682699023116900154214053925196489254710065907895782014272453065546518$
$824847897434933431664569271368351755123927296923374361441152014971573070121008710732707$
$498662298495359560056612256169399927357185263575542443291005680426135307337789366872807$
$838036815854250319855092337554316919281827658130123266681062040822967080062791559925351$
$926590284548982452507224281081471650988315489789192867431122844473536521343789348319054$
$196522017900468695588206921294272700560138175589656926746300983827318928266347378682438$
$544978775971778352893460053726336475424697823882040587018731559749154190473703788174426$
$649211024673735629613269429449925523168734965496667236572756993122946645337914243653461$
$059738237801441481355913907102007281405826070249397160592824173447708382029166878771128$
$252459938330727188131458116942273686151517316032607328702450637778279257505862105450113$
$059682511354869402730423773334661093162216075799297167782740578919464767879497204521200$
$730052076068008285644824817 91$
5. $\lambda = -1$
6. $M[X] = 1 + X - X^{34}$
7. $m = 2^{128} + 1$

A.8 AMNS for modulo 7937bits

1. $p = 941988475036429059655376002505010213724115630311559821373016218897400350584144108 37$
$490071747620753220520624766053127863436294050770012261329516433377885760458682588687669$
$727686308163642716704865821413643540859184620596450060892563176792316864145793460762307$
$198784480421054265177518125514144747810599916409793680683812894528521522708176290387750$
$131039568935069249342271798883675550755503254767535438603173606400794243867809638159426$
$675390152382747587503866630033584039649553444949706409893142366526468527124491956544584$
$217214409517883671777150016111985142912359790874811727102977677709986716716747636340727$
$329012517891671806024874581005771898964706392821758468320115609272974472619079654944766$
$089312997867204673233642775440827022732677076578093472615007540562264175847027343217615$
$974085027655758037095549920780825862535695330338305080668953817163823124128585964978640$
$235702324593476464080239633506772008298130878076923239496646173961350739781634135312495$
$592597586251499882535392849991458004329118497454994881802351124023309543464707420861762$
$254922700852623805606872419513659376475435424520403919266462870462440131218447513319799$
$390268689209542643517460970075433886389847486742676623927124629291712911715235742571160$
$884339926817417019626950585339617776557010959023016201643486317392899293617320141760209$
$489210204172085522355019255112442611479990634755627221641657319837075828775193020334491$
$098858413739781666802506815568666479348188425392067047228308999574897659017545251815385$
$489523398526704429890631989556277742841467180516881477771596246485832440310398467677273$
$802436428113436482566220967432222882299225065397568648631305346803249079455369369463246$
$508393928134131946761461832201612076101163459268954760390464784352160208127160008649501$
$455135389004879996914669211113993202741186905836688856406176532662647392396568054224619$
$789113707449091362390933525591756372988329987090875463545816892232913870070886329666 56$
$640636271287213779522559571621491437937604151809422852772631738279766357292525742583913$
$735417103113387553081786116758321535812885891844481825119563587209487045150328585856938$
$612397738415641838372660174591269838007961960664829290391189889523007948500105356865438$
$759067256056047592297944323385844042484639176003833935846790089957563197573283062223099$
$800104842057384331301698322570223842412411276716443236283017417450512655952582872528597$
$488039198368647407795270977121616576434579 21$
2. $n = 64$
3. $\rho = 2^{124}$
4. $\gamma = 489120909897793218914004175086921568902559423172013928694668206053413434405214687 07$
$237063414019340923102916660736260164631029334244159604965067468192262107544503478806776$
$063048785976152017380610870751295747411924905186101804015496966198183648146587965967902$
$770632032374500846581814685637465023128227285118234489162637071371843244612525987785 51$
$696991681578370602094115197405317545656451447233255774046268430053613714514680082346890$
$443030417464285843418311760694865379944443391824484316257994267345994097649847305341371$

860669056895636116661191512282340917548433573963541818366107514640149472571189506115098
102974617912744711340172041613933383502113529183092802100252123850511511339102168078867
864577442625088641298661394001538736279272879220261021317581256933618309532168730713001
820639627527590169903775384590622714113858490088696045279518950296396077872166521863224
775742410321277435174022457296569393758373924506298529964450607700202872384946582892208
532137886836325249085429408758500858828904238290326403885468591580434613146363654851172
715106160771316515188454964297388391872559132825530065588317706060337302537461305559412
789093933187385333959090606797988564995651405606647726095244778077825376681015369012660
555711313374012844253915096167189951535802588645621927477286750954485002419964720345315
769503274703717139215780692389404924874362212378327513943840574849840251932013462810991
103938421501970332541608794525756392100943697129388029661052161954307011237284815121813
818801681323503160354868251641670380405896683793264185767721599825251089691611179457152
302419761664312677835048148170132228756931655390846400257521069123190253577582996708631
640009617430136305882936259744625802120785757508752976972174886021375011405620345068475
798744193364933593787972403515235220515194740908910964539031479856572537900341844254723
505980824037149001884362567907446109792458631821704736664916831524800705530140203990097
463790901360548591030564882596545070304901354732139816805755947977413076548438515327914
328224950067654744428393830117099977046006961457487677399848117305736269442510005691399
915868799720359251928036549194246372346538755483526312371445758630693750288563511980599
132644680540780715359152587696567363645025443008039332003312673520633788664022755557955
646237824589825909639267042988980292943202249138825116191601120385851582205406380644031
038664544717489094158869166929111717620363 10
$\lambda = -1$
$M[X] = -1 + X + X^{56}$
$m = 2^{256} + 1$

A.9 AMNS for modulo 15872bits

$p = 887342287101457133702230408689956435044209280732913966186699779657746558584344982049067393536730606966998776768897896306305190283710435506611191484879817797245746350844408836521071434991762100852366647489738303840724562076438859546714678021336147145914084166382138259751517611438985032545868382587052016808078146222599559757371665558745309393631184679509639089211323245650884568644040208910871023741816277911800064442166113992883601078823237050617640542613471057536427106449895226611812509467663131900579863592514135040497207218119048092477796484031407897581406795255283758198557689120842113000441889996973364473275277601719728511364035442841466949635078740377949092547488301283684331510082207408682180377283723704635200776345031636260888031670705902488778628144051347207634567821496401426644878471045816258095052057795482528014638358649901452055942114220037257916655928657774367575507878140845376837096180033467493652063667578297781183733965421441436551203748096476793734077823970152522572626662777639282294039219134775236409083994824046912680809542966625910378559377769227380796281262049651716517053929343519689760550094037334983380283922947313692337709363968724600931985101718546226088936119923337390128044902507585857424307628863785130918074305502643845798246654172221823579951652494206537276028029893531585638717886811640535635541609708516224978436310908565102754171164386126981657599675178649711960373119333324813874122359244200877145200899006332045158614932997190077693659427043417243163258992970306190913410820587949540743851487807527485397494305495350712203368189283192278651156285462951973725083508980417224093153619693441214971773529903886135924524351335120613960772689417880442492529765451195737966167014258301903594127913921895559240923287062291025592754964239943075245727767119612404042546752181064820806628687652489727050353769519086142852682603780741274309845847519303818124356358683351070368527609593884304972509796542152895764671311380448850914668840656761613722993150923281179173646893600528138333285172816186957886061909316887959015729944446411234666145292991644565003129415424673238619145639491501205343074288466597496122744343320039288609441778911065289816104778180643418168153953039898075928738536645361491252950166609329189241811906993641653962353065107229002230602920651341927246286061825466657674845173483997698388212180854718957359876040544495695685188972771513316144614026885819914213258559000941380670

391451463427452874816429068238834324131985664346336966254236350805801801466402632487252
539717686403788645564028240597030650709750050214668295934186135968219913622865173607810
631297292169996361606848883430318395051248091978842345638114728657733726345540646702142
482427912881551634154599524683466673766623773611091522124828390123586047854015052807949
594799142214947059720299060486750620718073111706866574374518512547208342600703703482775
476249417487209318861395453807106603191696946768380808679036719465405372040687688138679
939489808052305922781032133841806593208366608924917605480375944355893342129762060356604
130597024972872569481661570996319240399974167267622876585440023940076782981200727634712
043465378061496985853410847253518148710514202181145858735136038304922553793591727421383
682207507089588511413025692802832783330414130090729155018218817606636389262102497928717
964958225089117798609071167525200132512412431501222517914278277220920321816221548821805
832898412435748705704494788594554333735777086116650498629068057640700219240848807147612
653728437336501570124130783976602579389351857935667338688894232479270749659294749582134
220689446774679550226759580687932861114002291791979038008104048720431927320322971743459
376504601406619364754182463309446901763876618598522677023201974818942811928746470109337
834618336577358568565788818234582856270555425019240634568155466008351325790942736886565
617796315266022292009041658533431309913890178692831567098437766905076096302098037161998
802023284567575607437243786769218090976287004327363919486632181790473557334786183937724
903166694303155718081839579360579165037566436068376499240935791070587854304593225551957
898611995414308766982609831992059567129318054975760060617357981561396139094588520602660
350418612176708031704025471733606891982736029973826956099369993509791575518213876063101
439322646160358443580330033773478894959611496513692310759187475246416989954130256482793
739335626272476259754887822781529959686685176623967866353099807035253821227902725752462
693008231976724674905102969761247061663390865658152438990180187429677918399174721891886
444657921964039263514158075726493516396955081765105691997700642011364855676699227887573
974343209583583817057574849000167745982237100816906971155436480426266363118972961017729
953791205138126459631581275632785981383978238779072482358243983440941206512740682497

2. $n = 128$
3. $\rho = 2^{124}$
4. $\gamma = 115462121921483609872348688324780275333780193505582199876353334593953951142778733520245198840563873455738438998978124683417183859878656725730994243304898095303681732234419872185526204688768073555504367980677992749375585189887118158450178794073255532177279500854751436947327967071341506525945463564616770944462687620132606519644017328141842121118559646825703067632942386768647776268742917131908146433643005071263254586412574981580204479492126192745008504773440551336413779695580318605173881509319847848262417840612668101099599434905631706541560720840249352491224337231061454967496873359553027461978435374143470335248320287735987445268188742377338149973213117943655400516160104256270694688047791627460742205261192252179122001722658416256468184903014649045087045683484680403767544038931219836258999335641597854603705430003864614984125156903985323451263436246176040194545869447085679083746440667143804266192621360047066476323152191661736001340055702260166697602220467562292616065743947163629344451054131537372283943453036788904381413499569828419430692642949077174713528058718936906219993184130908511811932175786003955355754540489187663140511746355179228890558911282982625982565168671847202268060724503699663741192633093095352380042047600536663113803882944489262275440865040956737533053892326351879902827218903151822051294082743224820473508296390411860509813806578086303349094336162284440463168410772214550682610164448118577091822563984688089181806709538023170133057130873633636043485591988278232621709459191079981387614329204918415664633064300008072362693718612839818981907285205918339316100260047647921671164174316013147502662317795750144573281915836157940338856801999558009037003958737743367035387092238930968275748123005271590020644918677349518941973456013570781781248852960315565764587881291492194010232054228908168790014591988780307294741253688441540883200167396594871498112015012549221790995344714669056653804613749337961246318047450759862444278625482301751217299841192224870712347703022937774023759852926071862318672319061082501675172867224362843410688541707506670485796349235328128137708952077930140266864189103970693685449739885082426547430318542226886017239933810175883990277914948929754293200961199363918558785167376614061737992353310768960735210715259769390552655726203217908580782004591925240150951791354767218946464435784936340100631763

5141760956138468218975640279851451479588076179875334314604824602756987711060350159224497422231109462236145375748645051031935038594209748662033559724127126834927398736610468171241530686651947502388770771029344116409116451307508915888884752125486297892088592272359791797725816528575390160023202157676803594291324026300730949680439125022328789565332217484243945984770218606538264986939158056540182103374581213788930782023112044436304138695848109427173801338445884592857058354694095603118934932937773376216503982113615607571307063306654132068822833488046142363523444119953119329228757680273621798093386514375711343455634446610300158745616778896745106372132127102279346928561482384422698018124330219059428328393608225882567776801039707646147130891550722337475855093058340352866379107954284941479212570381498756402241211637491086543652795165537146712702401548362205915299385455200564415268256271106613397986182437143424266107162553143584157128564497949639897835157103570645420637938834599866072675285256339312568346276949556104344788402653483663306071186552415233913778338197596939711626890219570193421887904375367148304189889759827985469205072748812650718722719095239681293620909356671731603614298270200811081762183974551977366483154967123752248031636796370353894905062006382561563620799859031303203514416297136696580697607467544337568661892078671032821652541801094387947939453820252953589223787691494010586741640063666744601118695916355802426658646300438396415753362027103602435542330830963484613525662732680456021688855774642723454002771316414656346024747976943005888165045234637502307456166964743284048326054910770034298649825819713398101868992910942049088486656535877914053910313421206426258683626387211032718826616183506028732432807766481008295504764590999058987577525900313077648817098459116779128615213897368573742621361472510217924107949732480171027175168428480393247723284426765542154941461154492340875913434228271604791930657284549715042188299809994560434845494458188592258187077566644426608834935779740393986764375966295469032119699100164100878839601392012152829222873360061037164245317541290281866399899767582013491029001223208862090797732774819486308972964477751987405844525539218829631582022537363588604011332713331791138865925814340462004598359617292299035162813088252968092879758679010978601711316291540935999862339965025918643809429195088751367877724438530367996372813169910392945632980626693970935127597807512269555740
5. $\lambda = -1$
6. $M[X] = 1 + X - X^{24}$
7. $m = 2^{256} + 1$