Efficient Big Integer Multiplication in Cryptography

Murat Burhan Ilter, Murat Cenk

Institute of Applied Mathematics
Middle East Technical University
Ankara, Turkey.
e-mail: milter@metu.edu.tr
mcenk@metu.edu.tr

Abstract—Most of the public key cryptography algorithms require efficient big integer multiplications. In this paper, we show how to develop efficient integer multiplication algorithms for cryptographic applications by combining different methods. We determine the complexities by taking into account the cost of single word multiplication, single word addition and double word addition on different platforms. This paper is an extended version of [11]. We add the complexity of the last term method that is used for computing complexity of multiplication of degree $n-1$ polynomials from the product of degree $n-2$ polynomials. The unbalanced refined Karatsuba 2-way multiplication algorithm is also included. These new contributions improved the complexity results introduced in [11]. Moreover, we present the best multiplication algorithm complexities for NIST primes on different implementation platforms.

Keywords—Integer multiplication, Karatsuba algorithm, Karatsuba-like algorithms, Public Key Cryptography

1. Introduction

The standard multiplication algorithm or the schoolbook method has complexity $O(n^2)$ for multiplying polynomials. In 1962, Karatsuba [2] designed a new method to multiply polynomials in which the complexity is $O(n^{1.58})$. Toom-Cook [5], [6] algorithm was suggested with the complexity $O(n^{1.46})$. After these improvements, many analyses have been done in the literature and to the best of our knowledge Harvey's method [7] gives the best complexity result in the big $-O$ notation. It should be noted that these new algorithms are generally not suitable for integer multiplication algorithms used in current cryptographic applications because of the hidden coefficients in the big$-O$ notation. For this reason, the main task is to design the efficient integer multiplication algorithms to speed up public key cryptography. To this end, the schoolbook, the Karatsuba, and the Karatsuba-like algorithms are analyzed in this work. In the literature, similar techniques for binary polynomial multiplications over binary fields have been used in [1], [3] and the best complexities were obtained for this type of polynomials.

In this paper, by using the mentioned techniques, we obtained promising results for integer multiplications which are used in cryptographic applications by considering single word multiplication, single word addition, and double word addition. First, the methods in the literature are investigated with their optimization. Following this, we find the best

multiplication algorithms with efficient complexities. Moreover, the hybrid method which combines other methods is also investigated. We also compute the complexity of the unbalanced refined Karatsuba 2-way algorithm that enhances the complexities further. It is shown that the complexity computation of multiplication of degree $n-1$ polynomials is efficiently obtained using the product of degree $n-2$ polynomials that is called the last term method.

The work is structured as follows. Preliminaries on complexity calculation are given in Section 2. We indicate the complexity analysis of 2-way and 3-way polynomial multiplication algorithms in Section 3 and Section 4. The combination of these methods is introduced in Section 5. We conclude the paper with Section 6.

2. Preliminaries

Throughout the paper, all polynomials are assumed to be defined over $\mathbb{Z}$. We can represent integers as polynomials. Let A be an integer and β be a base. Then, A can be represented by

$$A = a_{n-1}\beta^{n-1} + \cdots a_1\beta + a_0$$

where n is the length of A and we have $0 \leq a_i \leq \beta - 1$. The computation of complexities of multiplication algorithms in the recursive approach needs the following type recursion:

$$r_1 = e,\ r_n = ar_{n/b} + cn + d,$$

where a, b, and i are positive integers, and $n = b^i$, $a \neq b$, and $a \neq 1$. The solution of this recursive equation is

$$r_n = \left(e + \tfrac{bc}{a-b} + \tfrac{d}{a-1}\right)n^{log_b(a)} - \left(\tfrac{bc}{a-b}\right)n - \tfrac{d}{a-1}. \quad (1)$$

In the complexity computations A_D and A_S stand for double precision/word addition and single precision/word addition, respectively. In addition, the following notations will be used.
$M(n)$: The total number of operations required for multiplying degree $n-1$ polynomials.
$M_{\otimes}(n)$: The total number of single word multiplications required for multiplying degree $n-1$ polynomials.
$M_{\oplus D}(n)$: The total number of double word additions required for multiplying degree $n-1$ polynomials.
$M_{\oplus S}(n)$: The total number of single word additions required for multiplying degree $n-1$ polynomials.

Moreover, single word multiplication is denoted by $\widetilde{M}$. The result of the multiplication of two single words is stored in a double word.

3. 2-way Multiplication Algorithms

2-way multiplication method is analyzed in this section. In these methods, polynomials are divided into two equal parts. Let $A(x)$ and $B(x)$ be two degree $n-1$ polynomials such that

$$A(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1},$$
$$B(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{n-1}x^{n-1},$$

where n is an even positive integer.

We recursively deploy the following steps: $A(x)$ and $B(x)$ are divided into two equal parts. $A(x) = A_0 + A_1y$ where

$$\begin{aligned} A_0 &= a_0 + a_1x + \cdots + a_{\frac{n}{2}-1}x^{\frac{n}{2}-1}, \\ A_1 &= a_{\frac{n}{2}} + a_{\frac{n}{2}+1}x + \cdots + a_{n-1}x^{\frac{n}{2}-1}. \end{aligned} \quad (2)$$

and $y = x^{\frac{n}{2}}$. Similarly, $B(x) = B_0 + B_1y$ where

$$\begin{aligned} B_0 &= b_0 + b_1x + \cdots + b_{\frac{n}{2}-1}x^{\frac{n}{2}-1}, \\ B_1 &= b_{\frac{n}{2}} + b_{\frac{n}{2}+1}x + \cdots + b_{n-1}x^{\frac{n}{2}-1}. \end{aligned} \quad (3)$$

Then, $(A_0 + A_1y)(B_0 + B_1y)$ is computed with different approaches that are presented in the following sections.

3.1. The schoolbook algorithm

The schoolbook algorithm is also known as the standart multiplication algorithm in the literature. By using (2) and (3) the multiplication of the polynomials $A(x)$ and $B(x)$ with the schoolbook method can be performed as:

$$A(x)B(x) = A_0B_0 + y[A_1B_0 + A_0B_1] + y^2A_1B_1.$$

Note that there are four multiplications of polynomials of size $\frac{n}{2}, A_0B_0, A_1B_0, A_0B_1,$ and A_1B_1. Then, A_1B_0 is added to A_0B_1 which requires $n-1$ double word additions. In the computation of construction, there exist two overlaps between A_0B_0 and $(A_1B_0+A_0B_1)$ and between $(A_1B_0+A_0B_1)$ and A_1B_1. First, we need to compute the addition cost of overlap between A_0B_0 and $(A_1B_0+A_0B_1)$ which requires $\frac{n}{2}-1$ double word additions. In the manner with first case, the addition cost of $(A_1B_0 + A_0B_1)$ and A_1B_1 is also $\frac{n}{2} - 1$. In consequence, we have totally $2(\frac{n}{2} - 1) + n - 1 = 2n - 3$ double word additions. We can separate the total complexity into multiplication, and double word addition.

$$\begin{aligned} M(n) &= 4M(\tfrac{n}{2}) + (2n-3)A_D, M(1) = 1 \\ M_{\otimes}(n) &= 4M(\tfrac{n}{2}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 4M(\tfrac{n}{2}) + (2n-3), M_{\oplus D}(1) = 0. \end{aligned}$$

By using (1), we can compute complexities explicitly as:

$$\begin{aligned} M(n) &= 2n^2 - 2n + 1, \\ M_{\otimes}(n) &= n^2, \\ M_{\oplus D}(n) &= n^2 - 2n + 1. \end{aligned}$$

3.2. The Karatsuba 2-way algorithm

The Karatsuba method [2], is applied as:

$$\begin{aligned} A(x)B(x) = \; & A_0B_0 + y[(A_0 + A_1)(B_0 + B_1) \\ & -A_0B_0 - A_1B_1] + y^2A_1B_1. \end{aligned} \tag{4}$$

We compute the complexities as in the case of the schoolbook algorithm by considering single word and double word addition and single word multiplication, and the following complexities are obtained.

$$\begin{aligned} M(n) &= 3M(\tfrac{n}{2}) + (3n-4)A_D + (n)A_S, \\ & \quad M(1) = 1, \\ M_{\otimes}(n) &= 3M(\tfrac{n}{2}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 3M(\tfrac{n}{2}) + (3n-4), M_{\oplus D}(1) = 0, \\ M_{\oplus S}(n) &= 3M(\tfrac{n}{2}) + n, M_{\oplus S}(1) = 0. \end{aligned}$$

Using (1), we compute complexities explicitly as:

$$\begin{aligned} M(n) &= 7n^{1.58} - 8n + 2, \\ M_{\otimes}(n) &= n^{1.58}, \\ M_{\oplus D}(n) &= 4n^{1.58} - 6n + 2, \\ M_{\oplus S}(n) &= 2n^{1.58} - 2n. \end{aligned}$$

3.3. The refined Karatsuba 2-way algorithm

In this method [3], we first define P_1, P_2, and P_3 as in (5) in order to obtain compact formula.

$$\begin{aligned} P_1 &= A_0B_0, \\ P_2 &= (A_0 + A_1)(B_0 + B_1), \\ P_3 &= A_1B_1. \end{aligned} \tag{5}$$

To use this method, we divide polynomials into two parts as in the case of (2) and (3). Moreover, we can express the multiplication of these polynomials as:

$$A(x)B(x) = (y-1)(yP_3 - P_1) + yP_2 \tag{6}$$

It can be shown that the refined Karatsuba 2-way algorithm has the following complexities.

$$\begin{aligned} M(n) &= 3M(\tfrac{n}{2}) + (\tfrac{5n}{2} - 3)A_D + (n)A_S, \\ & \quad M(1) = 1, \\ M_{\otimes}(n) &= 3M(\tfrac{n}{2}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 3M(\tfrac{n}{2}) + (\tfrac{5n}{2} - 3), M_{\oplus D}(1) = 0, \\ M_{\oplus S}(n) &= 3M(\tfrac{n}{2}) + n, M_{\oplus S}(1) = 0. \end{aligned}$$

By the means of (1), we compute the complexities as follows:

$$\begin{aligned} M(n) &= 6.5n^{1.58} - 7n + \tfrac{3}{2}, \\ M_{\otimes}(n) &= n^{1.58}, \\ M_{\oplus D}(n) &= 3.5n^{1.58} - 5n + \tfrac{3}{2}, \\ M_{\oplus S}(n) &= 2n^{1.58} - 2n. \end{aligned}$$

3.4. The optimized Karatsuba 2-way algorithm

In this method [4], we divide the products in (5) into two parts. P_{iL} is the lower part of P_i with $\frac{n}{2}$ terms where $i = 1, 2, 3$. P_{iH} is the higher part of P_i, for $i = 1, 2, 3$ with $\frac{n}{2} - 1$ terms. Substituting these into (4), we can apply the method as follows:

$$\begin{aligned} A(x)B(x) &= P_{1L} + x^{\frac{n}{2}}[P_{1H} + P_{2L} - P_{1L} \\ &\quad -P_{3L}] + x^n[P_{2H} - P_{1H} - P_{3H} \\ &\quad +P_{3L}] + x^{\frac{3n}{2}}P_{3H} \\ &= P_{1L} + x^{\frac{n}{2}}[(P_{1H} - P_{3L}) + P_{2L} \\ &\quad -P_{1L}] + x^n[-(P_{1H} - P_{3L}) + P_{2H} \\ &\quad -P_{3H}] + x^{\frac{3n}{2}}P_{3H} \end{aligned}$$

For the optimized Karatsuba 2-way algorithm, the complexities are:

$$\begin{aligned} M(n) &= 3M(\tfrac{n}{2}) + (\tfrac{5n}{2} - 3)A_D + (n)A_S, \\ &\quad M(1) = 1, \\ M_{\otimes}(n) &= 3M(\tfrac{n}{2}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 3M(\tfrac{n}{2}) + (\tfrac{5n}{2} - 3), M_{\oplus D}(1) = 0, \\ M_{\oplus S}(n) &= 3M(\tfrac{n}{2}) + n, M_{\oplus S}(1) = 0. \end{aligned}$$

With the use of (1), we can compute complexities as follows:

$$\begin{aligned} M(n) &= 6.5n^{1.58} - 7n + \tfrac{3}{2}, \\ M_{\otimes}(n) &= n^{1.58}, \\ M_{\oplus D}(n) &= 3.5n^{1.58} - 5n + \tfrac{3}{2}, \\ M_{\oplus S}(n) &= 2n^{1.58} - 2n. \end{aligned}$$

3.5. The unbalanced refined Karatsuba 2-way algorithm for odd n

In this method [3], unlike the other 2-way multiplication algorithm methods we divide our polynomials into two unequal parts. Let $A(x)$ and $B(x)$ be two polynomials of degree $n-1$ such as:

$$\begin{aligned} A(x) &= a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1}, \\ B(x) &= b_0 + b_1x + b_2x^2 + \cdots + b_{n-1}x^{n-1}, \end{aligned}$$

where n is an odd positive integer. We recursively follow these steps: $A(x)$ and $B(x)$ are divided into two parts. $A(x) = A_0 + A_1y$ where

$$\begin{aligned} A_0 &= a_0 + a_1x + \cdots + a_{\frac{n-1}{2}}x^{\frac{n-1}{2}}, \\ A_1 &= a_{\frac{n+1}{2}} + a_{\frac{n+3}{2}}x + \cdots + a_{n-1}x^{\frac{n-3}{2}}. \end{aligned} \quad (7)$$

and $y = x^{\frac{n+1}{2}}$. Similarly, $B(x) = B_0 + B_1y$ where

$$\begin{aligned} B_0 &= b_0 + b_1x + \cdots + b_{\frac{n-1}{2}}x^{\frac{n-1}{2}}, \\ B_1 &= b_{\frac{n+1}{2}} + b_{\frac{n+3}{2}}x + \cdots + b_{n-1}x^{\frac{n-3}{2}}. \end{aligned} \quad (8)$$

In this approach, the polynomial A_0 contains one more element than the polynomial A_1 has. We use the same compact formula which is defined in (6), and the following complexities are obtained.

$$\begin{aligned} M(n) &= 2M(\tfrac{n+1}{2}) + M(\tfrac{n-1}{2}) + (\tfrac{5n-5}{2})A_D \\ &\quad +(n-1)A_S, M(1) = 1, \\ M_{\otimes}(n) &= 2M(\tfrac{n+1}{2}) + M(\tfrac{n-1}{2}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 2M(\tfrac{n+1}{2}) + M(\tfrac{n-1}{2}) + (\tfrac{5n-5}{2}), \\ &\quad M_{\oplus D}(1) = 0, \\ M_{\oplus S}(n) &= 2M(\tfrac{n+1}{2}) + M(\tfrac{n-1}{2}) + (n-1), \\ &\quad M_{\oplus S}(1) = 0. \end{aligned}$$

Remark 1: The last term of both P_1 and P_2 is $a_{\frac{n-1}{2}}b_{\frac{n-1}{2}}$. Therefore, we save one single word multiplication.

4. 3-way Multiplication Algorithms

In this section, 3-way multiplication algorithms are discussed. We can calculate $A(x)B(x)$ by using a 3-way algorithm. First, the polynomials $A(x)$ and $B(x)$ are divided into three parts as $A(x) = A_0 + A_1y + A_2y^2$ where

$$\begin{aligned} A_0 &= a_0 + a_1x + \cdots + a_{\frac{n}{3}-1}x^{\frac{n}{3}-1}, \\ A_1 &= a_{\frac{n}{3}} + a_{\frac{n}{3}+1}x + \cdots + a_{\frac{2n}{3}-1}x^{\frac{n}{3}-1}, \\ A_2 &= a_{\frac{2n}{3}} + a_{\frac{2n}{3}+1}x + \cdots + a_{n-1}x^{\frac{n}{3}-1}, \end{aligned} \quad (9)$$

and $y = x^{\frac{n}{3}}$. Similarly, $B(x) = B_0 + B_1y + B_2y^2$ where

$$\begin{aligned} B_0 &= b_0 + b_1x + \cdots + b_{\frac{n}{3}-1}x^{\frac{n}{3}-1}, \\ B_1 &= b_{\frac{n}{3}} + b_{\frac{n}{3}+1}x + \cdots + b_{\frac{2n}{3}-1}x^{\frac{n}{3}-1}, \\ B_2 &= b_{\frac{2n}{3}} + b_{\frac{2n}{3}+1}x + \cdots + b_{n-1}x^{\frac{n}{3}-1}. \end{aligned} \quad (10)$$

Then $(A_0 + A_1y + A_2y^2)(B_0 + B_1y + B_2y^2)$ is calculated.

4.1. The schoolbook algorithm

In the schoolbook algorithm, we use (9) and (10), and calculate the multiplication of polynomials $A(x)$ and $B(x)$ as follows:

$$\begin{aligned} A(x)B(x) &= A_0B_0 + y[A_1B_0 + A_0B_1] \\ &\quad +y^2(A_0B_2 + A_1B_1 + A_2B_0) \\ &\quad +y^3(A_1B_2 + A_2B_1) + y^4A_2B_2. \end{aligned}$$

The complexity of schoolbook algorithm is:

$$\begin{aligned} M(n) &= 9M(\tfrac{n}{3}) + (4n-8)A_D, M(1) = 1, \\ M_{\otimes}(n) &= 9M(\tfrac{n}{3}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 9M(\tfrac{n}{3}) + 4n - 8, M_{\oplus D}(1) = 0. \end{aligned}$$

By using (1), we can compute complexities explicitly:

$$\begin{aligned} M(n) &= 2n^2 - 2n + 1 \\ M_{\otimes}(n) &= n^2, \\ M_{\oplus D}(n) &= n^2 - 2n + 1. \end{aligned}$$

4.2. The Karatsuba-like 3-way algorithm

In the Karatsuba-like 3-way algorithm [8], we can use (9) and (10) and compute the multiplication of polynomials $A(x)$ and $B(x)$ as follows:

$$\begin{aligned} A(x)B(x) = {} & A_0B_0 + y[(A_0 + A_1)(B_0 + B_1) \\ & -A_0B_0 - A_1B_1] + y^2[(A_0 + A_2) \\ & (B_0 + B_2) - A_0B_0 - A_2B_2 \\ & +A_1B_1] + y^3[(A_1 + A_2)(B_1 + B_2) \\ & -A_1B_1 - A_2B_2] + y^4A_2B_2 \end{aligned}$$

The complexity of the Karatsuba-like 3-way algorithm is:

$$\begin{aligned} M(n) &= 6M(\tfrac{n}{3}) + (6n-11)A_D + (2n)A_S, \\ &\quad M(1) = 1, \\ M_{\otimes}(n) &= 6M(\tfrac{n}{3}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 6M(\tfrac{n}{3}) + 6n - 11, M_{\oplus D}(1) = 0, \\ M_{\oplus S}(n) &= 6M(\tfrac{n}{3}) + 2n, M_{\oplus S}(1) = 0. \end{aligned}$$

By applying (1), we can compute complexities explicitly as:

$$\begin{aligned} M(n) &= 6.8n^{1.63} - 8n + 2.2 \\ M_{\otimes}(n) &= n^{1.63}, \\ M_{\oplus D}(n) &= 3.8n^{1.63} - 6n + 2.2, \\ M_{\oplus S}(n) &= 2n^{1.63} - 2n. \end{aligned}$$

4.3. The optimized Karatsuba-like 3-way algorithm

In the optimized Karatsuba-like 3-way method, we use (9) and (10), and calculate the multiplication of polynomials. The case over binary fields is in [9]. In order to get the compact formula we define:

$$\begin{aligned} P_1 &= A_0B_0,\ P_2 = (A_0 + A_1)(B_0 + B_1), \\ P_3 &= A_1B_1,\ P_4 = (A_0 + A_2)(B_0 + B_2), \\ P_5 &= A_2B_2,\ P_6 = (A_1 + A_2)(B_1 + B_2). \end{aligned}$$

P_{iL} is the lower part of P_i with $(\frac{n}{3})$ terms where $i = 1, \cdots, 6$. P_{iH} is the higher part of P_i, for $i = 1, \cdots, 6$ with $(\frac{n}{3}-1)$ terms. Then, the multiplication of $A(x)B(x)$ is expressed as follows:

$$\begin{aligned} A(x)B(x) &= P_{1L} + y[P_{1H} + P_{2L} - P_{1L} - P_{3L}] \\ &\quad +y^2[P_{2H} - P_{1H} - P_{3H} + P_{4L} \\ &\quad -P_{1L} - P_{5L} + P_{3L}] + y^3[P_{4H} \\ &\quad -P_{1H} - P_{5H} + P_{3H} + P_{6L} - P_{3L} \\ &\quad -P_{5L}] + y^4[P_{6H} - P_{3H} - P_{5H} \\ &\quad +P_{5L}] + y^5P_{5H} \\ &= P_{1L} + y(P_{1H} - P_{3L}) + P_{2L} - P_{1L}] \\ &\quad +y^2[-(P_{1H} - P_{3L}) - P_{3H} + P_{2H} \\ &\quad +P_{4L} - P_{1L} - P_{5L}] + y^3[(P_{3H} \\ &\quad -P_{5L}) + P_{4H} - P_{1H} - P_{5H} + P_{6L} \\ &\quad -P_{3L}] + y^4[-(P_{3H} - P_{5L}) + P_{6H} \\ &\quad -P_{5H}] + y^5P_{5H} \end{aligned}$$

The complexity of the optimized Karatsuba-like 3-way algorithm is:

$$\begin{aligned} M(n) &= 6M(\tfrac{n}{3}) + (\tfrac{16}{3}n - 9)A_D + (2n)A_S, \\ &\quad M(1) = 1, \\ M_{\otimes}(n) &= 6M(\tfrac{n}{3}), M_{\otimes}(1) = 1, \\ M_{\oplus D}(n) &= 6M(\tfrac{n}{3}) + (\tfrac{16}{3}n - 9), M_{\oplus D}(1) = 0, \\ M_{\oplus S}(n) &= 6M(\tfrac{n}{3}) + 2n, M_{\oplus S}(1) = 0. \end{aligned}$$

By using (1), we can compute complexities as follows:

$$\begin{aligned} M(n) &= \tfrac{98}{15}n^{1.63} - \tfrac{22}{3}n + \tfrac{9}{5}, \\ M_{\otimes}(n) &= n^{1.63}, \\ M_{\oplus D}(n) &= \tfrac{53}{15}n^{1.63} - \tfrac{16}{3}n + \tfrac{9}{5}, \\ M_{\oplus S}(n) &= 2n^{1.63} - 2n. \end{aligned}$$

4.4. Complexity comparison of 2-way and 3-way multiplication algorithms

The complexity results that were calculated in Section 3 and Section 4 are presented in Table 1. In 2-way multiplication algorithms, the refined Karatsuba 2-way and the optimized Karatsuba 2-way algorithms lead to the best complexity. In 3-way multiplication algorithms, the optimized Karatsuba-like 3-way algorithm is the best algorithm. Note that these asymptotic complexities are useful when the input size is very huge. For cryptographic sizes, combination of different algorithms is more efficient than the use of a single algorithm. In the next section, we discuss this approach.

5. Combined Methods

It can be observed that the complexity of Karatsuba-like algorithms has better complexities than the schoolbook method. However; in some cases, using combined methods, applying the refined Karatsuba or the optimized Karatsuba algorithm before the schoolbook method yields more desirable results. Moreover, sometimes using the product of degree $n-2$ polynomials for computing complexity of product of degree $n-1$ polynomials leads to better results. We call this method the last term method. In this method, to multiply degree $n-1$ polynomials $A(x)$ and $B(x)$ where n is an odd number, we use the best multiplication method for degree $n-2$ polynomials, and compute the result in the following way:

$$\begin{aligned} A(x)B(x) &= A_1(x)B_1(x) + b_n x^n A_1(x) \\ &\quad + a_n x^n B_1(x) + a_n b_n x^{2n-2}, \end{aligned}$$

where $A_1(x)$ and $B_1(x)$ are obtained by deleting the last terms of $A(x)$ and $B(x)$ respectively. In other words, first we apply the best multiplication method for degree $n-2$, after we multiply last terms with schoolbook multiplication. The complexity of this algorithm is given as follows:

$$M(n) = M(n-1) + (2n-1)\widetilde{M} + (2n-3)A_D$$

where $\widetilde{M}$ denotes single word multiplication.

In order to find the best multiplication algorithm, first, the best complexity for $n=2$ is searched for a given platform. Then, we search for $n=3$ by using the previous results. We continue similarly and when we come to $n=l$, we check all possible cases by using the previous complexities. Since we have limited number of algorithms, this search ends quickly. In this section, we will analyze this approach.

Example 1: Assume that we have a platform where a single and a double word addition and a single word multiplication have the same complexity. In this platform, we want to multiply two polynomials of degree 5. If we use the schoolbook method, we need 36 word multiplications and 25 double word additions. Therefore, this computation results in 61 operations. However, using the refined Karatsuba 2-way algorithm first, then the schoolbook algorithm, we have

$$\begin{aligned} M(6) &= 3M(3) + (\tfrac{5\cdot 6}{2} - 3)A_D + (6)A_S \\ &= 3(9M + 4A_D) + 12A_D + 6A_S \\ &= 27M + 24A_D + 6A_S, \end{aligned}$$

and since we assume M, A_D, and A_S have equal costs, the number of operations reduces to 57.

Example 2: Assume that we have the same platform as in the Example 1, and we want to multiply

TABLE 1
Complexity Comparison of 2-way and 3-way multiplication algorithms

Algorithm	$M_{\oplus_S}(n)$	$M_{\oplus_D}(n)$	$M_{\otimes}(n)$	$M(n)$
SB		n^2-2n+1	n^2	$2n^2-2n+1$
KA2	$2n^{1.58}-2n$	$4n^{1.58}-6n+2$	$n^{1.58}$	$7n^{1.58}-8n+2$
RK2	$2n^{1.58}-2n$	$3.5n^{1.58}-5n+1.5$	$n^{1.58}$	$6.5n^{1.58}-7n+1.5$
OPK2	$2n^{1.58}-2n$	$3.5n^{1.58}-5n+1.5$	$n^{1.58}$	$6.5n^{1.58}-7n+1.5$
SB		n^2-2n+1	n^2	$2n^2-2n+1$
KA3	$2n^{1.63}-2n$	$3.8n^{1.63}-6n+2.2$	$n^{1.63}$	$6.8n^{1.63}-8n+2.2$
OPK3	$2n^{1.63}-2n$	$\frac{53}{15}n^{1.63}-\frac{16}{3}n+\frac{9}{5}$	$n^{1.63}$	$\frac{98}{15}n^{1.63}-\frac{22}{3}n+\frac{9}{5}$

two polynomials of degree 10. If we use the schoolbook method, we need 121 word multiplications and 100 double word additions, so this calculation comes out 221 operations. Also, if we use the last term method we obtain,

$$\begin{aligned} M(11) &= M(10)+21M+19A_D \\ &= 75M+10A_S+70A_D+21M+19A_D \\ &= 96M+89A_D+10A_S, \end{aligned}$$

and 195 operations. On the other hand, if we use the unbalanced refined Karatsuba 2-way algorithm first by dividing these polynomials into degree 5 and degree 4 polynomials, and then use the best multiplication algorithms, it can be obtained that

$$\begin{aligned} M(11) &= 2M(6)+M(5)+(25)A_D+10A_S \\ &= 54M+48A_D+12A_S+25M+16A_S \\ &\quad +25A_D+10A_S \\ &= 79M+73A_D+38A_S, \end{aligned}$$

which results in 190 operations. Also, by using (3.5) the number of calculations reduce to 189 operations.

As it can be seen from Example 1 and 2, computation of the minimum number of operations can be done with a hybrid method where one single word addition, one double word addition, and one single word multiplication have the same cost. The list of this operations for multiplying two polynomials up to size 20 is given in Table 2 where SB is used for the schoolbook algorithm, RK2 is used for the refined Karatsuba 2-way algorithm, and URK2 is used for the unbalanced refined Karatsuba 2-way algorithm. The last column in Table 2 shows the name of the algorithm used to obtain the indicated complexities. If there are two algorithm names, it means that first apply the leftmost algorithm, and then apply the rightmost one.

Suppose that in a given platform, we have one single word addition and one single word multiplication having the same cost; however, one double word addition is twice as costly as one single word addition and one single word multiplication. The Table 3 is constructed for the best multiplication algorithms for size up to 19 polynomials in the given platform.

5.1. NIST Primes

In this section, we include the NIST Primes as an application for efficient polynomial multiplication with different prime fields. Recommendations for the elliptic curves over the following prime fields are given in the NIST standard 186-2 [10].

$$\begin{aligned} p_{192} &= 2^{192}-2^{64}-1,\ p_{224}=2^{224}-2^{96}+1, \\ p_{256} &= 2^{256}-2^{224}+2^{192}+2^{96}-1, \\ p_{384} &= 2^{384}-2^{128}-2^{96}+2^{32}-1, \\ p_{521} &= 2^{521}-1. \end{aligned}$$

TABLE 2
Minimum number of operations for 1 Single Word addition = 1 Double Word Addition = 1 Single Word Multiplication

n	Mult.	A_S	A_D	# Operation	Algorithm
1	1			1	SB
2	4		1	5	SB
3	9		4	13	SB
4	16		9	25	SB
5	25		16	41	SB
6	27	6	24	57	RK2, M(3)
7	40	6	35	81	M(6), last term
8	48	8	44	100	RK2, M(4)
9	65	8	59	132	M(8), last term
10	75	10	70	155	RK2, M(5)
11	78	22	73	189	URK2, M(6),M(5)
12	81	30	99	210	RK2,M(6)
13	106	30	122	258	M(12), last term
14	120	32	137	289	RK2, M(7)
15	135	36	158	329	URK2, M(8),M(7)
16	144	40	169	353	RK2, M(8)
17	177	40	200	417	M(16), last term
18	195	42	219	456	RK2, M(9)
19	214	46	244	504	URK2, M(10), M(11)
20	225	50	257	532	RK2, M(10)

TABLE 3
Minimum number of operations for 2 Single Word addition = 1 Double Word Addition = 2 Single Word Multiplication

n	Mult.	A_S	A_D	Tot. Operation	Algorithm
1	1			1	SB
2	4		1	6	SB
3	9		4	17	SB
4	16		9	34	SB
5	25		16	57	SB
6	27	6	24	81	RK2, M(3)
7	40	6	35	116	M(6), last term
8	48	8	44	144	RK2, M(4)
9	65	8	59	191	M(8), last term
10	75	10	70	225	RK2, M(5)
11	78	22	89	278	URK2, M(6),M(5)
12	81	30	99	309	RK2, M(6)
13	106	30	122	380	M(12), last term
14	120	32	137	426	RK2, M(7)
15	135	36	158	487	URK2, M(8),M(7)
16	144	40	169	522	RK2, M(8)
17	177	40	200	617	M(16), last term
18	195	42	219	655	RK2, M(9)
19	214	46	244	748	URK2, M(10), M(11)
20	225	50	257	789	RK2, M(10)

For instance, if we want to use the field p_{192} in 32-bit implementation platform, then we use the best multiplication algorithm for degree 6 polynomials. In the Table 4, the sizes of the polynomial are given for 8-bit, 16-bit, 32-bit, and 64-bit implementation platforms. The best results for polynomials up to degree 19 are given in the Table 2 and 3 on two different implementation platforms. For larger degrees, similar calculations can be done to obtain the best multiplication algorithms.

TABLE 4
NIST primes and polynomial sizes

	p_{192}	p_{224}	p_{256}	p_{384}	p_{521}
8-bit	25	28	32	48	66
16-bit	12	14	16	24	33
32-bit	6	7	8	12	17
64-bit	3	4	4	6	9

5.2. Results for different cost metric

The choice of the best algorithm for multiplication might change on different implementation platforms since the cost of a single word multiplication, single word addition, and double word addition are different. In order to obtain the best results, we have to measure these costs and choose the corresponding multiplication algorithm by considering these results. For instance, if the cost of a double word addition is higher than a single word addition and a single word multiplication, we need to choose a multiplication algorithm which contains less double word additions with respect to others. On the other hand, as it can be seen from Table 2 and 3, even though the operation costs are different, the used algorithms for best results can be the same.

6. Conclusion

In this paper, 2-way and 3-way polynomial multiplication algorithms that are used for current cryptographic applications are analyzed over the ring of integers by considering arithmetic complexities. Although the refined Karatsuba 2-way multiplication algorithm has the best complexity, using the hybrid approach with the schoolbook method gives better complexity results. The last term method and the unbalanced split of polynomials also reduces the complexities further.

Acknowledgments

This work is supported in part by TÜBİTAK under grant number 115R289.

References

[1] Cenk, M., & Hasan, M. A. (2015). Some new results on binary polynomial multiplication. Journal of Cryptographic Engineering, 5(4), 289-303. 1345-1361.

[2] Karatsuba, A. (1963). Multiplication of multidigit numbers on automata. In Sov. Phys. Dokl. (Vol. 7, No. 7, pp. 595-596).

[3] Bernstein, D. (2009). Batch binary Edwards. In: Advances in Cryptology CRYPTO 2009, LNCS, (vol. 5677, pp. 317336).

[4] Zhou, G., & Michalik, H. (2010). Comments on" A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Field". IEEE Transactions on Computers, 59(7), 1007-1008.

[5] Toom, A. L. (1963). The complexity of a scheme of functional elements realizing the multiplication of integers. In Soviet Mathematics Doklady (Vol. 3, No. 4, pp. 714-716).

[6] Cook, S. A., & Aanderaa, S. O. (1969). On the minimum computation time of functions. Transactions of the American Mathematical Society, 142, 291-314.

[7] Harvey, D., Van Der Hoeven, J., & Lecerf, G. (2016). Even faster integer multiplication. Journal of Complexity, 36, 1-30.

[8] Weimerskirch, A., & Paar, C. (2006). Generalizations of the Karatsuba Algorithm for Efficient Implementations. IACR Cryptology ePrint Archive, 2006, 224.

[9] Cenk, M., Hasan, M. A., & Negre, C. (2014). Efficient subquadratic space complexity binary polynomial multipliers based on block recombination. IEEE Transactions on Computers, 63(9), 2273-2287.

[10] Fips, P. U. B. (2000). 186-2. digital signature standard (dss). National Institute of Standards and Technology (NIST), 20, 13.

[11] İlter, M. B., & Cenk, M. (2017). Efficient big integer multiplication in cryptography. ISCTurkey 2017 Conference Proceedings, 8-13.