Published in final edited form as:

IEEE Trans Comput. 2019 ; 68: . doi:10.1109/TC.2018.2874662.

Better Circuits for Binary Polynomial Multiplication

Magnus Gaudal Find,

National Institute of Standards and Technology, Gaithersburg, MD 20899.

René Peralta

Computer Security Division, National Institute of Standards and Technology, Gaithersburg, MD 20899.

Abstract

We develop a new and simple way to describe Karatsuba-like algorithms for multiplication of polynomials over $\mathbb{F}_2$. We restrict the search of small circuits to a class of circuits we call *symmetric bilinear*. These are circuits in which AND gates only compute functions of the form $\sum_{i \in S} a_i \cdot \sum_{i \in S} b_i$ ($S \subseteq \{0, \dots, n-1\}$). These techniques yield improved recurrences for $M(kn)$, the number of gates used in a circuit that multiplies two kn -term polynomials, for $k = 4, 5, 6$, and 7 . We built and verified the circuits for n -term binary polynomial multiplication for values of n of practical interest. Circuits for n up to 100 are posted at <http://cs-www.cs.yale.edu/homes/peralta/CircuitStuff/BinPolMult.tar.gz>.

Keywords

Binary polynomial multiplication; circuits; symmetric bilinear circuits

1 Introduction

Let A, B be polynomials of degree $n-1$ over $\mathbb{F}_2$. This paper is about finding small circuits that compute the polynomial $A \cdot B$. We consider circuits over the basis $(\wedge, \oplus, 1)$ (that is, arithmetic over $\mathbb{F}_2$). We aim for circuits with as few gates as possible.

Applications.

Binary polynomial multiplication is the main operation in the arithmetic of finite fields of characteristic two. It is of particular importance in elliptic curve cryptography (see [1], [2] and the references therein). Additionally finite field multiplication is used in the Galois Counter mode of operation [3]. Other important applications include binary Goppa codes and derived cryptosystems, both symmetric and asymmetric.

Our Technique.

We consider generalizations of the algorithm due to Karatsuba. Karatsuba's algorithm splits a polynomial into two parts and then does recursive multiplication. Researchers have already considered generalizations of this algorithm which split the input into $k - 3$ parts. We provide a unifying description of these generalized "Karatsuba-like" algorithms, allowing for a systematic search for such recurrences.

Contributions.

In this paper, the *size* of a circuit is the number of gates it contains. For $k = 4, 5, 6, 7$ we improve on the recurrences for Karatsuba-like multiplication. We obtain smaller circuits than the previously known best bounds (see Cenk and Hasan [4] and Bernstein [1]) for almost all cryptographically relevant values of n . Table 5 in Section 6 shows circuit sizes and depths for a range of n .

2 Definitions

Polynomials.

The input to our problem will be polynomials of degree $n - 1$

$$A = \sum_{i=0}^{n-1} a_i x^i \quad \text{and} \quad B = \sum_{i=0}^{n-1} b_i x^i \quad (a_i, b_i \in \{0, 1\}).$$

We refer to a polynomial of degree $n - 1$ as an n -term polynomial (even though some of the terms may be zero). For integers $i < j$, we identify a polynomial $A = a_i x^i + a_{i+1} x^{i+1} + \dots + a_j x^j$ with the tuple $(a_i, a_{i+1}, \dots, a_j)$, and denote $A[i] = a_i$ and $A[i..j] = (a_i, \dots, a_j)$.

Symmetric Circuits.

For the purpose of this paper, we restrict ourselves to circuits with the following structure, which we call *symmetric bilinear circuits*. A symmetric bilinear circuit contains only binary XOR (addition) and binary AND (multiplication) gates. It consists of

- a top layer consisting only of XOR gates;¹
- a multiplication layer that computes only functions of the form

$$\sum_{i \in S} a_i \cdot \sum_{i \in S} b_i \quad (S \subseteq \{0, \dots, n-1\}),$$

(note that the sets of indeces on the right and left factors must be equal);

- a bottom layer that uses only XOR gates and outputs c_0 through c_{2n-1} , where

¹We visualize circuits as having the inputs at the top and the outputs at the bottom.

$$c_t = \sum_{i+j=t} a_i b_j.$$

For an integer $n > 1$, let $M(n)$ be the size of the smallest circuit over $(\wedge, \oplus, 1)$ computing the polynomial product of two n -term polynomials. We emphasize that there does not necessarily exist a symmetric bilinear circuit of size $M(n)$ for all n , although all the best sizes we know of can be achieved using circuits of this form.

We will use three metrics for the class of symmetric bilinear circuits. Let $\mathcal{C}$ be a circuit in the class. The *multiplicative cost of $\mathcal{C}$* , denoted $M_{\wedge}(\mathcal{C})$, is the number of AND gates. The *upper additive cost of $\mathcal{C}$* , denoted $M^{\oplus}(\mathcal{C})$, is the number of XOR gates in the top layer of $\mathcal{C}$. The *bottom additive cost of $\mathcal{C}$* , denoted $M_{\oplus}(\mathcal{C})$, is the number of XOR gates in the bottom layer of $\mathcal{C}$.

Example.

Consider $A = a_0 + a_1x$ and $B = b_0 + b_1x$. The product of A and B is $C = c_0 + c_1x + c_2x^2$, where

$$c_0 = a_0b_0; c_1 = a_0b_1 + a_1b_0; c_2 = a_1b_1.$$

With respect to multiplicative complexity, there is only one optimal symmetric bilinear circuit for $n = 2$. The top layer calculates $s_1 = a_0 + a_1$, $s_2 = b_0 + b_1$. The multiplication layer calculates

- $c_0 = a_0b_0$;
- $c_2 = a_1b_1$;
- $t = s_1s_2 = (a_0 + a_1)(b_0 + b_1)$.

The bottom layer calculates $c_1 = c_0 + c_2 + t$. The multiplicative cost is three (which is optimal, among all boolean circuits). Both additive costs $M^{\oplus}$ and $M_{\oplus}$ are 2.

For degree-2 polynomials, a computer search yields exactly six circuits of multiplicative complexity six in the prescribed class.² Of these, two have bottom additive cost 6 (the other four circuits have bottom additive costs 7, 7, 8, 8).

Linear Operators and Representation of Circuits.

Let W be an $n \times m$ matrix over $\mathbb{F}_2$. The function $\mathbf{x} \mapsto W \cdot \mathbf{x}$ can be computed using only XOR gates. We denote by $s(W)$ be number of XOR gates necessary to compute this mapping. For a symmetric bilinear circuit $\mathcal{C}$ that on input $\mathbf{a} = (a_0, \dots, a_{n-1})$ and $\mathbf{b} = (b_0, \dots, b_{n-1})$ computes the output $\mathbf{c} = (c_0, \dots, c_{2n-1})$, there exists a unique matrix T such that the i th

². Any symmetric bilinear circuit computing the product of two quadratic polynomials has at least six F2 multiplications. See Section 5.

AND gate computes the i th coordinate of the Hadamard product $(T \cdot \mathbf{a}) \circ (T \cdot \mathbf{b})$. We call this matrix T the *top matrix* of $\mathcal{C}$. Similarly, the bottom part of the circuit can be described as a matrix, which we call the *main matrix* of $\mathcal{C}$, denoted R . More precisely, given the top matrix T , the main matrix R satisfies

$$\mathbf{c} = R \cdot [(T \cdot \mathbf{a}) \circ (T \cdot \mathbf{b})].$$

A symmetric bilinear circuit $\mathcal{C}$ is completely described by the two matrices (T, R) , along with XOR circuits computing them. The two matrices corresponding to the circuit in Fig. 1 are

$$T = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, R = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Note that the matrix T is used twice in the circuit $\mathcal{C}$.

3 Previous Work

Asymptotic Complexity.

Much work has been put into giving asymptotically good algorithms for binary polynomial multiplication, see [5], [6], [7]. Currently, the asymptotically best algorithm is due to Harvey, van der Hoeven, and Lecerf who showed that $M(n) = O(n \log n 8^{\log^* n})$, where $\log^*$ ($\cdot$) denotes the iterated logarithm, an unbounded but extremely slow growing function.

Concrete Complexity.

For values of n that are interesting for cryptographic purposes (say, $n = 1000$), the asymptotic bounds do not say much about the *concrete circuit complexity*. For this we need to employ a combination of different recursive relations.

This problem has received much attention in recent years, see [1], [2], [4], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. Bernstein, in [1], used various (new and old) recursive constructions to build small circuits for n -term polynomial multiplication for $2 \leq n \leq 1000$. Notably he obtains results “better than anything that can be found in the hardware literature” [1, page 7], including results reported in [9], [17], [20], [24], [25]. In recent work [4], Cenk and Hasan show that smaller circuits exist for many values of n . Their work applies new recurrence relations in a manner similar to that of [1]. Recurrence relations sufficient to obtain the values stated in [4] are shown on Table 1.

3.1 Known Recursive Constructions

Many different recursive constructions for polynomial multiplications have been suggested. Most of these constructions are based on either interpolation or a generalization of Karatsuba.

Interpolation methods are usually based on the approach of Toom in [26]. Concrete constructions for $k = 2, 3$ have been proposed by Bernstein [1] and by Cenk and Hasan [4].

That Karatsuba multiplication can be generalized has been observed many times. Notable references include Montgomery [11], Weimerskirch and Paar [14], Bernstein [1], and more recently Cenk et al. [4], [27]. Some recurrences have been patented, as in Montgomery's patent [28] and in Koç and Erdem's patent [29].

Table 1 contains known recurrences. The "alg" column in each of the tables assigns numbers to some of the associated algorithms. These number will be used later in this paper, when reporting how our circuits are obtained. This paper improves on the recurrences for $M(4n)$, $M(5n)$, $M(6n)$, and $M(7n)$ (Table 2).

4 Finding New Karatsuba-Like Recurrences

There is a standard way to convert any symmetric bilinear circuit $\mathcal{C}$, for polynomial multiplication of k bits, into a recurrence. The recurrence yields an upper bound on $M(k \cdot n)$ in the following way: to multiply two $k \cdot n$ -term polynomials A, B , divide A, B into k blocks. That is, write A as

$$A = A_0 + A_1x^n + \dots + A_{k-2}x^{n \cdot (k-2)} + A_{k-1}x^{n \cdot (k-1)},$$

with each A_i an n -term polynomial, and similarly write B as

$$B = B_0 + B_1x^n + \dots + B_{k-2}x^{n \cdot (k-2)} + B_{k-1}x^{n \cdot (k-1)}.$$

The product $A \cdot B$ can be written as

$$A \cdot B = \sum_{t=0}^{2k-2} U_t x^{t \cdot n},$$

where $U_t = \sum_{i+j=t} A_i \cdot B_j$. To compute the polynomials $U_0, \dots, U_{2k-2}$ we use the circuit $\mathcal{C}$, where each XOR gate is replaced with polynomial addition (bitwise XOR) and each AND gate is replaced with a circuit for n -term multiplication. Each of the top XOR gates in $\mathcal{C}$ results in n gates, and each of the bottom XOR gates result in $2n-1$ gates. Each AND gate is replaced by a circuit for n -term multiplication, using $M(n)$ gates. We immediately have that the cost of computing $U_0, \dots, U_{2k-2}$ is $n \cdot M^\oplus(\mathcal{C}) + (2n-1)M_\oplus(\mathcal{C}) + M_\wedge(\mathcal{C}) \cdot M(n)$.

Finally, to obtain the actual bits of the result, we have to take care of the overlap between $U_i x^{i \cdot n}$ and $U_{i+1} x^{(i+1) \cdot n}$. One way to do this is by doing bitwise XOR with the high-order $n-1$ bits from U_i and the low-order $n-1$ bits from U_{i+1} for $i = 0, \dots, 2k-3$. This uses $(2k-2) \cdot (n-1)$ gates. This gives the generic recurrence

$$M(kn) \leq M_{\wedge}(\mathcal{C})M(n) + M^{\oplus}(\mathcal{C})n + M_{\oplus}(\mathcal{C})(2n-1) + (n-1)(2k-2).$$

(1)

An upper bound for the cost $M^{\oplus}(\mathcal{C})$ is twice the cost of multiplication by the top matrix T . An upper bound for the cost $M_{\oplus}(\mathcal{C})$ is the cost of multiplication by the main matrix R .

Let $\mathbf{A} = (A_0, \dots, A_{k-1})$, $\mathbf{B} = (B_0, \dots, B_{k-1})$ and $\mathbf{U} = (U_0, \dots, U_{2k-2})$. The Karatsuba recurrence is

$$\mathbf{U} = R \cdot [(T \cdot \mathbf{A}) \circ (T \cdot \mathbf{B})].$$

As we noted earlier, the polynomials U_i overlap.

Let $\tau = M_{\wedge}(\mathcal{C})$ (i.e., the number of AND gates in $\mathcal{C}$). Then $\mathbf{P} = (P_0, \dots, P_{\tau-1})$ satisfies

$$\mathbf{P} = (T \cdot \mathbf{A}) \circ (T \cdot \mathbf{B}).$$

We will show that the bits of $\mathbf{c}$ can be obtained by multiplying $\mathbf{P}$ with an extended version of the matrix R .

4.1 Example: Karatsuba ($k = 2$)

To illustrate how to optimize the bottom layer, we describe a way to obtain the Karatsuba-recurrence $M(2n) = 3M(n) + 7n - 3$ in Table 1. This recurrence is not novel, but this particular way of deriving it naturally generalizes to arbitrary values of k . Let the input polynomials be

$$A = a_0 + a_1x + \dots + a_{2n-1}x^{2n-1} = A_0 + A_1x^n,$$

and

$$B = b_0 + b_1x + \dots + b_{2n-1}x^{2n-1} = B_0 + B_1x^n,$$

for n -term polynomials A_0, A_1, B_0, B_1 . Let the result be

$$C = A \cdot B = c_0 + c_1x + \dots + c_{4n-2}x^{4n-2},$$

and let $U_t = \sum_{i+j=t} A_i \cdot B_j$, for $t = 0, 1, 2$.

Multiplication by the matrices T, R corresponding to Fig. 1 can be done with one and two additions, respectively. Thus Eq. (1) yields

$$\begin{aligned} M(2n) &\leq 3M(n) + 2n + 2 \cdot (2n - 1) + (n - 1)(2 \cdot 2 - 2) \\ &= 3M(n) + 8n - 4. \end{aligned}$$

To improve on this, we consider the overlap more carefully.

For a $(2n - 1)$ -term polynomial D let $L(D)$, $M(D)$, and $H(D)$ be the unique polynomials with $n - 1$, 1, and $n - 1$ terms, respectively, that satisfy

$$D = L(D) + M(D)x^{n-1} + H(D)x^n.$$

Recall

$$C = A \cdot B = U_0 + U_1x^n + U_2x^{2n},$$

and

$$(P_0, P_1, P_2) = (T \cdot A) \circ (T \cdot B).$$

That is, $P_0 = A_0 \cdot B_0$, $P_1 = (A_0 + A_1) \cdot (B_0 + B_1)$, and $P_2 = A_1 \cdot B_1$.

Our aim is to write the polynomial C in terms of P_0, P_1, P_2 . Using the fact that $U_0 = P_0$, $U_1 = P_0 + P_1 + P_2$, and $U_2 = P_2$, we have

- $C[0..n-2] = L(U_0) = L(P_0)$,
- $C[n-1] = M(U_0) = M(P_0)$,
- $C[n..2n-2] = L(U_1) + H(U_0) = (L(P_0) + L(P_1) + L(P_2)) + H(P_0)$,
- $C[2n-1] = M(U_1) = M(P_0) + M(P_1) + M(P_2)$,
- $C[2n..3n-2] = L(U_2) + H(U_1) = L(P_2) + (H(P_0) + H(P_1) + H(P_2))$,
- $C[3n-1] = M(U_2) = M(P_2)$,
- $C[3n..4n-2] = H(U_2) = H(P_2)$.

Now we can write the components of the polynomial C as linear functions of the low, middle, and high parts of the polynomials computed by the multiplication gates

$$\begin{pmatrix} C[n-1] \\ C[2n-1] \\ C[3n-1] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} M(P_0) \\ M(P_1) \\ M(P_2) \end{pmatrix},$$

(2)

and

$$\begin{pmatrix} C[0 \dots n-2] \\ C[n \dots 2n-2] \\ C[2n \dots 3n-2] \\ C[3n \dots 4n-2] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} L(P_0) \\ L(P_1) \\ L(P_2) \\ H(P_0) \\ H(P_1) \\ H(P_2) \end{pmatrix}.$$

(3)

All matrices in Eqs. (2) and (3) are matrices of polynomials.

Now it remains to find good circuits to compute the linear operators in Eqs. (2) and (3). Since these matrices are small, it is easy to see that the first can be computed using two additions, and the second can be computed using five additions. Each addition in the computation of the first linear mapping costs 1 XOR gate, and in the second linear mapping each operation costs $n-1$ XOR gates. The top part still uses $2n$ XOR gates. This yields the previously stated recurrence

$$\begin{aligned} M(2n) &\leq 3M(n) + 2n + 5 \cdot (n-1) + 2 \\ &= 3M(n) + 7n - 3. \end{aligned}$$

(4)

4.2 Generalizing to $k = 3$

The construction of the previous section generalizes to any value of k . Given the two matrices (T, R) , which define a symmetric bilinear circuit $\mathcal{C}$ for multiplication of k -term polynomials, the output bits $C[]$ of $\mathcal{C}$ satisfy the following equations:

$$\begin{pmatrix} C[n-1] \\ C[2n-1] \\ \vdots \\ C[(2k-1)n-1] \end{pmatrix} = R \cdot \begin{pmatrix} M(P_0) \\ M(P_1) \\ \vdots \\ M(P_{\tau-1}) \end{pmatrix}.$$

(5)

Let the *extended matrix* E be defined as

$$E = \begin{pmatrix} R_1 & 0 \\ R_2 & R_1 \\ \vdots & \\ R_{2k-1} & R_{2k-2} \\ 0 & R_{2k-1} \end{pmatrix},$$

(6)

where R_i is the i th row of R .

Let $(P_0, \dots, P_\tau) = (T \cdot \mathbf{a}) \circ (T \cdot \mathbf{b})$. Then the remaining components of $C[]$ can be written as

$$\begin{pmatrix} C[0 \dots n-2] \\ C[n \dots 2n-2] \\ \vdots \\ C[(2k-1)n \dots 2kn-2] \end{pmatrix} = E \cdot \begin{pmatrix} L(P_0) \\ \vdots \\ L(P_{\tau-1}) \\ H(P_0) \\ \vdots \\ H(P_{\tau-1}) \end{pmatrix}.$$

(7)

This yields the recurrence

$$M(kn) \leq M_{\wedge}(\mathcal{E}) \cdot M(n) + 2n \cdot s(T) + (n-1) \cdot s(E) + s(R).$$

(8)

Note that this allows for a succinct description of a multiplication circuit: each such circuit is described by XOR circuits for T , R and E .

Critical Paths.—Let δ_T , δ_R , δ_E be the length of the critical paths of circuits computing the linear transformations T , R , E , respectively. The total delay Δ_C of our construction satisfies the following recurrence

$$\Delta_C(kn) \leq (\delta_T + \max\{\delta_R, \delta_E\}) + \Delta_C(n).$$

(9)

Table 5 reports actual combinational delays (i.e., depth) of our circuits. Since the circuit for polynomial multiplication is bilinear, any path from inputs to outputs goes through exactly one AND gate. Thus, Δ_C can be calculated from

$$\Delta_C = (\text{depth} - 1)\delta_{\oplus} + \delta_{\wedge},$$

where $\delta_{\oplus}$, $\delta_{\wedge}$ are the delays of an XOR gate, AND gate respectively.

Matrix Computation versus Using Common Subexpressions.—For specific values of k , similar approaches have been used. To quote Montgomery [11, p. 365]: “by taking advantage of common subexpressions”, and Zhou and Michalik [21]: “By reviewing the work of [8], we show that the gate complexity of [the Karatsuba Algorithm] can be reduced by exploring the common subexpressions”.

We remark that obtaining a circuit for linear operators purely using common subexpressions results in so-called “cancellation-free” circuits (also called SUM-circuits). For some linear operators these circuits are highly suboptimal [30], see also [31, Section 5.3]. Indeed, for the extended matrix E in the 6-way split below, the only minimal sized circuit we have found has cancellation, so it could not have been obtained using only common subexpressions.

Finding Recursion Circuits.—The recurrence in Eq. (8) suggests the following strategy to finding recurrences upper bounding $M(k \cdot n)$ for a fixed k : First find circuits for k -term multiplication with the smallest possible number of AND gates. Among these, find one where Eq. (8) is as good as possible.

We remark that both of these tasks are computationally very challenging; computing the tensor rank is **NP**-hard [32]. The problem of finding the smallest XOR circuit for a given matrix is **NP**-hard and max-**SNP**-hard [33], meaning that if **NP** = **P** even finding a circuit which is at most a particular constant larger than the optimum is intractable.

Note that the matrix T fully determines the matrices R and E . The requirement of T is that the elements of the Hadamard product $(T \cdot \mathbf{a}) \circ (T \cdot \mathbf{b})$ span the target bilinear forms in $\mathbf{c}$. We use the following randomized heuristic:

1. randomly generate candidate matrices for T ;
2. solve for matrices R, E ;
3. find (linear) circuits for multiplication by T, R , and E .

For the first step we used the heuristics of [33].³ We parametrized this search so as to favor matrices with rows of low hamming weight (heuristically, this is expected to yield circuits with fewer XOR gates). For the second step we used the heuristics of [34] for small low-depth XOR circuits, although we prioritized size. This explains the large depth of our circuits for R and E with $n = 6$ (see Table 3).

4.3 New Recurrence Relations

Using the approach described in the previous sections we obtain several new recurrence relations. These are shown in Table 2. We describe the recurrence by describing the two matrices T, R associated with the recurrence (the matrix E is derived from R). The straight-line programs computing each of the matrices are available as supplemental material to this article.

3-Way Split.—A search gives the matrices

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

It is easily seen that $s(T) = 3$, $s(R) = 6$, and $s(E) = 12$. This gives the recurrence

$$\begin{aligned} M(3n) &\leq 6M(n) + 2 \cdot 3n + (n-1)12 + 6 \\ &= 6M(n) + 18n - 6, \end{aligned}$$

which is the same recurrence as reported in [27].

4-Way Split.—A computer search gives the matrices

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

³The techniques of [2] can be used to this search.

Again, it is not hard to verify that $s(T) = 5$, $s(R) = 12$, $s(E) = 24$, and that this uses 9 multiplications. We get the recurrence

$$\begin{aligned} M(4n) &\leq 9M(n) + 2 \cdot 5n + (n-1)24 + 12 \\ &= 9M(n) + 34n - 12. \end{aligned}$$

(10)

We note that this is a little better than what one would get by applying Eq. (4) twice.

5-Way Split.—For $n = 5$, a computer search gave matrices T, M, E with $s(T) = 8$, $s(R) = 19$, and $s(E) = 38$, using 13 multiplications. The matrices are given in Appendix A.1. This gives the recurrence

$$\begin{aligned} M(5n) &\leq 13M(n) + 2 \cdot 8n + 38(n-1) + 19 \\ &= 13M(n) + 54n - 19. \end{aligned}$$

(11)

6-Way Split.—For $n = 6$, a computer search found matrices T, M, E with $s(T) = 13$ gates, $s(R) = 30$, and $s(E) = 59$, using 17 multiplications. The matrices are given in Appendix A.2. This gives the recurrence

$$\begin{aligned} T(6n) &\leq 17M(n) + 2 \cdot 13n + (n-1)59 + 30 \\ &= 17M(n) + 85n - 29. \end{aligned}$$

(12)

7-Way Split.—Similarly for $n = 7$, a computer search found matrices T, M, E with $s(T) = 16$, $s(R) = 41$, and $s(E) = 75$. This structure required 22 multiplications. This leads to the recurrence

$$\begin{aligned} M(7n) &\leq 22M(n) + 2 \cdot 16n + 75(n-1) + 41 \\ &= 22M(n) + 107n - 34. \end{aligned}$$

(13)

The matrices are given in Appendix A.3.

5 Multiplicative Complexity of Polynomial Multiplication

A natural question about the recurrences in the previous section is whether they can be improved. Do matrices giving better recurrence relations exist? In particular, is it possible to find matrices giving a smaller number of multiplications in the recursion? It turns out that using this technique, for $k = 2, 3, 4, 5$ there is not, but for $k = 6$ there could be. In this section we will briefly sketch why.

There is a known relationship between error correcting codes and quadratic boolean circuits computing finite field or polynomial multiplication (see [35], [36]). Roughly speaking, any quadratic boolean circuit computing n -term polynomial multiplication induces an error correcting code with certain parameters. Therefore the nonexistence of certain codes can be used to prove the nonexistence of certain circuits. More specifically, Kaminski [37] shows that if there exists a method for multiplying two n -term polynomials in $\mathbb{F}_2[X]$ which uses l multiplications over $\mathbb{F}_2$, then there exists a linear code of length l , weight $2n - d$, and dimension d . This holds for all $n - d < 2n$. Table 4 gives multiplicative complexity lower bounds derived using known bounds for the length of linear codes (see e.g., [38]). The column (l, d, w) indicates parameters for a code with length l , dimension d and weight w that does not exist and therefore establishes the lower bound. We leave it as open problems to close the gaps for $k = 6, 7$, and to find recurrences with better low-order terms than what we provide in the previous section.

6 Results

Although Eqs. (10), (11), (12), and (13) improve on known Karatsuba recurrences, they only yield upper bounds on the circuit complexity of multiplication. Determining the actual size of the circuits that can now be obtained require an experimental approach. Our approach was as follows: first find small circuits for $n = 2, 3 \dots, k$. Then, to find a circuit for multiplication of $(k + 1)$ -term polynomials, apply each of the applicable recursive constructions, using the previously found circuits as base cases. Then look at obvious inefficiencies (unused gates, two distinct gates computing the same function, etc.). Finally, select the smallest circuit and continue.

Table 5 shows the sizes and depths obtained, and compare them to the previous state of the art:

- s_{CH}, d_{CH} are the sizes and depths reported in [4];
- s_K, d_K are the sizes and depths obtained in this work under the constraint that only Karatsuba methods are used;
- s_I, d_I are the sizes and depths obtained in this work when interpolations methods are allowed;

In Table 5

- Alg denotes what recurrence is used, the number of the algorithm, (see Tables 1 and 2);

- The * for $n = 11$ indicates that the circuit published in [39] is used;
- *Bold text* indicates an improvement.

7 Conclusions

We proposed a new method to describe, find, and analyze Karatsuba-like recurrences. The method leads to better recurrences than previously known for splitting into 4, 5, 6, and 7 blocks. We constructed circuits for binary polynomial multiplication which are smaller, and often of lower depth, than previously known. The circuits were verified by computing the algebraic normal form of the outputs. Circuits up to degree 99 are posted at <http://cs-www.cs.yale.edu/homes/peralta/CircuitStuff/BinPolMult.tar.gz>.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors are with the US Gov., this research was part of their regular duties.

Appendix A

Matrices and Their Additive Complexity

This section contains the matrices T, R found for 5 through 7-way split. We also report upper bounds on the additive complexities of multiplying by these matrices (and by the derived matrix E). We note that we could derive circuits with lower depths than those reported in Table 5 simply by optimizing the circuits for T, R, E with respect to depth rather than size. This could be of value in applications.

A.1 5-Way Split

The matrices used in the 5-way split are as follows:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The extended matrix E is derived from R . The additive complexities of our circuits for T , R , E are 8, 19, and 38, respectively.

A.2 6-Way Split

The matrices used in the 6-way split are as follows:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The additive complexities of our circuits for T , R , E are 13, 30, and 59, respectively.

A.3 7-Way Split

The matrices used in the 7-way split are as follows:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The additive complexities of our circuits for T , R , E are 16, 41, and 75, respectively.

References

- [1]. Bernstein DJ, “Batch binary edwards,” in Proc. 29th Annu. Int. Cryptology Conf, 2009, pp. 317–336.
- [2]. Brent RP, Gaudry P, Thomé E, and Zimmermann P, “Faster multiplication in $\text{GF}(2)[x]$,” in Proc. 8th Int. Symp. Algorithmic Number Theory, 2008, pp. 153–166.

- [3]. Dworkin M, "Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC," NIST, Gaithersburg, MD, USA, Tech. Rep. SP800-38D, 11 2007.
- [4]. Cenk M and Hasan MA, "Some new results on binary polynomial multiplication," IACR Cryptology ePrint Archive, vol. 2015, 2015, Art. no. 94.
- [5]. Karatsuba AA and Ofman Y, "Multiplication of multidigit numbers on automata," Soviet Physics Doklady, vol. 7, pp. 595–596, 1963 [Online]. Available: <http://cr.yp.to/bib/entries.html#1963/karatsuba>
- [6]. Schönhage A, "Schnelle multiplikation von polynomen über Körpern der Charakteristik 2," Acta Informatica, vol. 7, pp. 395–398, 1977.
- [7]. Harvey D, van der Hoeven J, and Lecerf G, "Faster polynomial multiplication over finite fields," CoRR, vol. abs/1407.3361, 2014, <http://arxiv.org/abs/1407.3361>.
- [8]. Paar C, "A new architecture for a parallel finite field multiplier with low complexity based on composite fields," IEEE Trans. Comput, vol. 45, no. 7, pp. 856–861, 7 1996.
- [9]. Rodriguez-Henriquez F and Koç ÇK, "On fully parallel Karatsuba multipliers for $GF(2^m)$," in Proc. Int. Conf. Comput. Sci. Technol., 2003, pp. 405–410.
- [10]. Sunar B, "A generalized method for constructing subquadratic complexity $GF(2^k)$ multipliers," IEEE Trans. Comput, vol. 53, no. 9, pp. 1097–1105, 9 2004.
- [11]. Montgomery PL, "Five, six, and seven-term Karatsuba-like formulae," IEEE Trans. Comput, vol. 54, no. 3, pp. 362–369, 3 2005.
- [12]. Erdem SS, Yanik T, and Koç ÇK, "Polynomial basis multiplication over $GF(2^m)$," Acta Applicandae Mathematicae, vol. 93, no. 1, pp. 33–55, 2006.
- [13]. von zur Gathen J and Shokrollahi J, "Efficient FPGA-based Karatsuba multipliers for polynomials over $\mathbb{F}$," in Proc. Int. Conf. Sel Areas Cryptography, 2006, pp. 359–369.
- [14]. Weimerskirch A and Paar C, "Generalizations of the Karatsuba algorithm for efficient implementations," IACR Cryptology ePrint Archive, vol. 2006, 2006, Art. no. 224.
- [15]. Zimmermann P, "How fast can we multiply polynomials over $GF(2)$? [Extended abstract]," in Proc. Conf. Algorithmic Number Theory, 2007, Art. no. 165.
- [16]. Fan H and Hasan MA, "Comments on "five, six, and seven-term Karatsuba-like formulae"," IEEE Trans. Comput, vol. 56, no. 5, pp. 716–717, 5 2007.
- [17]. Peter S and Langendörfer P, "An efficient polynomial multiplier in $GF(2^m)$ and its application to ECC designs," in Proc. Conf. Des. Autom. Test Eur., 2007, pp. 1253–1258.
- [18]. Bodrato M, "Towards optimal Toom-Cook multiplication for univariate and multivariate polynomials in characteristic 2 and 0," in Proc. Int. Workshop Arithmetic Finite Fields, 6 2007, pp. 116–133. <http://bodrato.it/papers/#WAIFI2007>.
- [19]. Cenk M, Koç ÇK, and Ozbudak F, "Polynomial multiplication over finite fields using field extensions and interpolation," in Proc. 19th IEEE Int. Symp. Comput. Arithmetic, 2009, pp. 84–91.
- [20]. Fan H, Sun J, Gu M, and Lam K-Y, "Overlap-free Karatsuba-Ofman polynomial multiplication algorithms," IET Inf. Secur, vol. 4, no. 1, pp. 8–14, 3 2010.
- [21]. Zhou G and Michalik H, "Comments on "a new architecture for a parallel finite field multiplier with low complexity based on composite field"," IEEE Trans. Comput, vol. 59, no. 7, pp. 1007–1008, 7 2010.
- [22]. Zhou G, Michalik H, and Hinsenkamp L, "Complexity analysis and efficient implementations of bit parallel finite field multipliers based on Karatsuba-Ofman algorithm on FPGAs," IEEE Trans. Very Large Scale Integr. Syst, vol. 18, no. 7, pp. 1057–1066, 7 2010.
- [23]. Dyka Z, Langendoerfer P, and Vater F, "Combining multiplication methods with optimized processing sequence for polynomial multiplier in $GF(2^k)$," in Proc. 4th Western Eur. Conf. Res. Cryptology, 2011, pp. 137–150.
- [24]. Chang NS, Kim CH, Park Y-H, and Lim J, "A non-redundant and efficient architecture for Karatsuba-Ofman algorithm," in Proc. Int. Conf. Inf. Secur., 2005, pp. 288–299.
- [25]. von zur Gathen J and Shokrollahi J, "Fast arithmetic for polynomials over F_2 in hardware," in Proc. IEEE Inf. Theory Workshop, 2006, pp. 107–111.

- [26]. Toom AL, "The complexity of a scheme of functional elements realizing the multiplication of integers," Soviet Math. Doklady, vol. 3, no. 4, pp. 714–716, 1963.
- [27]. Cenk M, Hasan MA, and Nègre C, "Efficient subquadratic space complexity binary polynomial multipliers based on block recombination," IEEE Trans. Comput, vol. 63, no. 9, pp. 2273–2287, 9 2014.
- [28]. Montgomery P, "Six-term Karatsuba-variant calculator," U.S. Patent 7 363 336, 4 22, 2008.
- [29]. Koç Ç and Erdem S, "Lean multiplication of multi-precision numbers over GF(2m)," U.S. Patent 7 447 310, 11 4, 2008.
- [30]. Boyar J and Find MG, "Cancellation-free circuits in unbounded and bounded depth," Theoretical Comput. Sci, vol. 590, pp. 17–26, 2015.
- [31]. Jukna S and Sergeev I, "Complexity of linear boolean operators," Found. Trends Theoretical Comput. Sci, vol. 9, no. 1, pp. 1–123, 2013.
- [32]. Håstad J, "Tensor rank is NP-complete," J. Algorithms, vol. 11, no. 4, pp. 644–654, 1990.
- [33]. Boyar J, Matthews P, and Peralta R, "Logic minimization techniques with applications to cryptology," J. Cryptology, vol. 26, no. 2, pp. 280–312, 2013.
- [34]. Boyar J, Find MG, and Peralta R, "Small low-depth circuits for cryptographic applications," Cryptography Commun, 3 2018, 10.1007/s12095-018-0296-3.
- [35]. Brown MR and Dobkin DP, "An improved lower bound on polynomial multiplication," IEEE Trans. Comput, vol. C-29, no. 5, pp. 337–340, 5 1980.
- [36]. Lempel A, Seroussi G, and Winograd S, "On the complexity of multiplication in finite fields," Theoretical Comput. Sci, vol. 22, pp. 285–296, 1983.
- [37]. Kaminski M, "A lower bound for polynomial multiplication," Theoretical Comput. Sci, vol. 40, pp. 319–322, 1985.
- [38]. Grassl M, "Bounds on the minimum distance of linear codes and quantum codes," 2007 [Online]. Available: <http://www.codetables.de>. Accessed on: Sep. 10, 2015.
- [39]. Peralta R, "Circuit minimization work," 1 2014 [Online]. Available: <http://cs-www.cs.yale.edu/homes/peralta/circuitstuff/cmt.html>

Fig. 1.

Circuit $\mathcal{C}$ computing the product of two polynomials of degree 1. This circuit has $M_{\wedge}(\mathcal{C}) = 3$ and $M_{\oplus}(\mathcal{C}) = M^{\oplus}(\mathcal{C}) = 2$.

TABLE 1

Known Recurrence Relations Based on Karatsuba's Method

alg	recurrence		reference
0	$M(n+1)$	$M(n) + 4n$	Schoolbook
1	$M(2n)$	$3M(n) + 7n - 3$	[1]
2	$M(2n-1)$	$2M(n) + M(n-1) + 7n - 8$	[4, Eq. 3]
-	$M(2n-2)$	$2M(n) + M(n-2) + 7n - 16$	[4, Eq. 3]
5	$M(3n)$	$6M(n) + 18n - 6$	[27]
-	$M(4n)$	$9M(n) + 34n - 11$	[4, Eq. 18]
3	$M(4n)$	$M(2n) + 6M(n) + 27n - 8$	[1], [4, Eq. 9]
3	$M(4n-1)$	$M(2n) + 5M(n) + M(n-1) + 27n - 18$	[1], [4, Eq. 10]
3	$M(4n-2)$	$M(2n) + 5M(n) + M(n-2) + 27n - 34$	[1], [4, Eq. 10]
4	$M(5n)$	$13M(n) + 55n - 17$	[4, Eq. 18]
4	$M(5n-1)$	$12M(n) + M(n-1) + 55n - 26$	[4, Eq. 20]
-	$M(6n)$	$17M(n) + 96n - 34$	[1], [27]

TABLE 2

New Recurrence Relations for Karatsuba-Like Algorithms

alg	recurrence		reference
6	$M(4n)$	$9M(n) + 34n - 12$	Eq. (10)
7	$M(5n)$	$13M(n) + 54n - 19$	Eq. (11)
8	$M(6n)$	$17M(n) + 85n - 29$	Eq. (12)
9	$M(7n)$	$22M(n) + 107n - 33$	Eq. (13)

NIST Author Manuscript

NIST Author Manuscript

NIST Author Manuscript

TABLE 3Depths of Circuits for Matrices T, R, E

n	T	R	E
4	2	4	4
5	3	3	5
6	4	9	15
7	5	5	6

NIST Author Manuscript

NIST Author Manuscript

NIST Author Manuscript

TABLE 4Upper and Lower Bounds on the Multiplicative Complexity of n -Term Polynomial Multiplication over $\mathbb{F}_2$

n	lower	(l, d, w)	upper
2	3	(2, 2, 2)	3
3	6	(5, 3, 3)	6
4	9	(8, 5, 3)	9
5	13	(12, 5, 5)	13
6	15	(14, 6, 6)	17
7	18	(17, 7, 7)	22

NIST Author Manuscript

NIST Author Manuscript

NIST Author Manuscript

TABLE 5

Sizes and Depths of Circuits Based on Karatsuba Recurrences

n	s_{CH}	s_K	d_{CH}	d_K	alg.
2	5	5	2	2	0
3	13	13	3	3	0
4	25	25	4	3	0
5	41	41	5	4	0
6	57	57	6	6	1
7	81	81	7	6	0
8	100	100	7	6	1
9	126	126	7	6	5
10	155	154	8	9	7
11	186	186	7	8	*
12	207	207	7	8	6
13	255	255	8	8	0
14	289	289	10	9	1
15	317	312	16	10	7
16	349	349	8	8	6
17	407	406	10	9	2
18	438	438	10	9	1
19	498	495	11	12	2
20	527	522	8	11	7
21	596	573	11	13	9
22	632	632	10	10	1
23	676	675	10	11	2
24	702	702	10	11	1
25	791	784	18	11	7
26	853	853	11	11	1
27	912	912	11	10	5
28	956	944	15	14	9
29	1020	1009	19	13	2
30	1053	1038	19	13	1
31	1119	1113	19	12	2
32	1156	1156	11	11	1
33	1274	1271	13	12	2
34	1335	1333	13	12	4
35	1393	1392	15	11	3
36	1429	1428	15	11	6
37	1559	1552	14	15	2
38	1616	1604	13	14	4
39	1680	1669	13	14	3
40	1718	1703	11	14	1

NIST Author Manuscript

NIST Author Manuscript

NIST Author Manuscript

n	s_{CH}	s_K	d_{CH}	d_K	alg.
41	1858	1806	14	16	2
42	1929	1862	13	16	9
43	1996	1982	15	15	2
44	2037	2036	15	12	6
45	2116	2105	20	14	7
46	2182	2179	15	14	3
47	2229	2228	15	13	3
48	2260	2259	15	13	6
49	2451	2436	21	14	2
50	2545	2523	21	17	7
51	2668	2663	16	14	2
52	2726	2725	16	13	6
53	2858	2841	14	24	8
54	2922	2878	14	24	8
55	3006	2987	20	17	2
56	3060	3022	20	17	9
57	3191	3145	22	15	3
58	3256	3212	22	17	4
59	3304	3273	20	15	3
60	3334	3306	20	15	6
61	3500	3472	22	15	2
62	3571	3553	22	15	1
63	3632	3626	21	14	6
64	3674	3673	16	13	6
65	3927	3920	16	15	2
66	4048	4041	16	15	1
67	4159	4152	18	14	3
68	4228	4220	18	14	3
69	4356	4353	18	14	2
70	4420	4417	20	14	3
71	4494	4478	20	25	8
72	4535	4510	20	25	8
73	4798	4782	18	18	2
74	4892	4815	29	18	7
75	4929	4847	29	18	7
76	5109	5075	18	17	1
77	5241	5198	16	16	3
78	5297	5255	16	16	3
79	5359	5329	29	16	6
80	5400	5366	21	16	6
81	5713	5593	17	19	2
82	5854	5702	16	19	1

n	s_{CH}	s_K	d_{CH}	d_K	alg.
83	5983	5769	18	18	9
84	6064	5804	18	18	9
85	6209	6118	23	18	2
86	6284	6224	20	19	4
87	6369	6344	20	18	3
88	6415	6413	20	15	1
89	6576	6516	23	28	8
90	6660	6550	23	28	8
91	6794	6776	23	17	2
92	6851	6842	20	16	3
93	6944	6929	23	18	3
94	7013	7010	18	16	1
95	7076	7073	20	15	3
96	7112	7110	20	16	1
97	7496	7465	21	17	2
98	7684	7636	24	20	4
99	7859	7801	26	19	3
100	7934	7847	21	19	7