Searching for best Karatsuba recurrences

Çağdaş Çalık[1] and Morris Dworkin[2] Nathan Dykas[3] Rene Peralta[4]

[1] Computer Security Division, NIST, USA
cagdas.calik@nist.gov
[2] Computer Security Division, NIST , USA
dworkin@nist.gov
[3] Mathematics Department, University of Maryland
ndykas@math.umd.edu
[4] Computer Security Division, NIST, USA
peralta@nist.gov **

Abstract. Efficient circuits for multiplication of binary polynomials use what are known as Karatsuba recurrences. These methods divide the polynomials of size (i.e. number of terms) $k \cdot n$ into k pieces of size n. Multiplication is performed by treating the factors as degree-$(k-1)$ polynomials, with multiplication of the pieces of size n done recursively. This yields recurrences of the form $M(kn) \leq \alpha M(n) + \beta n + \gamma$, where $M(t)$ is the number of binary operations necessary and sufficient for multiplying two binary polynomials with t terms each. Efficiently determining the smallest achievable values of (in order) α, β, γ is an unsolved problem. We describe a search method that yields improvements to the best known Karatsuba recurrences for k = 6,7 and 8. This yields improvements on the size of circuits for multiplication of binary polynomials in a range of practical interest.

1 Introduction

Polynomials over $\mathbb{F}_2$ are called *binary polynomials*. They have a number of applications, including in cryptography (see [2,5] and the references therein) and in error correcting codes. Let A, B be binary polynomials. We seek small circuits, over the basis $(\wedge, \oplus, 1)$ (that is, arithmetic over $\mathbb{F}_2$), that compute the polynomial $A \cdot B$. In addition to size, i.e. number of gates, we also consider the depth of such circuits, i.e. the length of critical paths.

Notation: We let $M(t)$ denote the number of gates necessary and sufficient to multiply two binary polynomials of size t.

Suppose the polynomials A, B are of odd degree $2n-1$. Karatsuba's algorithm ([11]) splits A, B into polynomials A_0, A_1 (B_0, B_1 resp.) of size n. Then it recursively computes the product $C = A \cdot B$ as shown in Figure 1. Careful counting of operations leads to the *2-way Karatsuba recurrence* $M(2n) \leq 3M(n) + 7n - 3$ (see [9], equation (4)).

** Corresponding author.

$$
\begin{aligned}
A &= (a_0 + a_1 X + \cdots a_{n-1} X^{n-1}) + X^n \cdot (a_n + a_{n+1} X + \cdots a_{2n-1} X^{n-1}) \\
A &= A_0 + X^n A_1 \\
B &= (b_0 + b_1 X + \cdots b_{n-1} X^{n-1}) + X^n \cdot (b_n + b_{n+1} X + \cdots b_{2n-1} X^{n-1}) \\
B &= B_0 + X^n B_1 \\
U &\leftarrow A_0 \cdot B_0 \\
V &\leftarrow A_1 \cdot B_1 \\
W &\leftarrow (A_0 + A_1) \cdot (B_0 + B_1) + U + V \\
C &\leftarrow U + X^n W + X^{2n} V.
\end{aligned}
$$

Fig. 1. Karatsuba's algorithm

The product C is $A_0 B_0 + X^n (A_0 B_1 + A_1 B_0) + X^{2n} A_1 B_1$. The constant 3 in the 2-way Karatsuba recurrence comes from the fact that 3 multiplications are necessary and sufficient to calculate the three terms $A_0 B_0, A_0 B_1 + A_1 B_0$, and $A_1 B_1$ from A_0, A_1, B_0, B_1. The term $7n - 3$ counts the number of $\mathbb{F}_2$ additions necessary and sufficient to produce the term W and then combine the terms U, V, W into the result C (see [9]).

The generalized Karatsuba method takes two polynomials with kn terms, splits each into k pieces $A_0, \ldots, A_{k-1}$, $B_0, \ldots, B_{k-1}$, computes the polynomials

$$C_m = \sum_{m=i+j} A_i B_j$$

and finally combines the C_i's by summing the overlapping terms.

Karatsuba recurrences have been studied for some time. The paper [12] gives recurrences for the cases $n = 5, 6$, and 7. These recurrences have been improved over the years. The state of the art is [9].

The work [9] provides a unifying description of the generalized Karatsuba method, allowing for a systematic search for such recurrences. The steps in the search are outlined in Figure 2. Steps 1 and 4 involve solving computationally hard problems. We rely on experimental methods to gain reasonable assurance that we have found the best Karatsuba recurrences in the defined search space.

2 Finding minimum-size spanning bilinear forms

In this section we describe the method for computing (or finding upper bounds on) the constant α in the Karatsuba recurrence.

1. find sets of bilinear forms of minimum size α from which the target C_i's can be computed via additions only.
2. as per [9], each set of bilinear forms determines three matrices T, R, E over $\mathbb{F}_2$.
3. the matrices T, R, E define linear maps L_T, L_R, L_E.
4. let the number of additions necessary for each of the maps be μ_T, μ_R, μ_E, respectively.
5. then the maps yield the recurrence

$$M(kn) \leq \alpha M(n) + \beta n + \gamma$$

with $\beta = 2\mu_T + \mu_E$ and $\gamma = \mu_R - \mu_E$.
6. pick the best recurrence.

Fig. 2. Methodology

2.1 Description of the problem

Consider the two n-term (degree $n-1$) binary polynomials

$$f(x) = \sum_{i=0}^{n-1} a_i x^i, \quad g(x) = \sum_{i=0}^{n-1} b_i x^i \quad \in \mathbb{F}_2[x]$$

with $(2n-1)$-term product

$$h(x) := (fg)(x) = \sum_{k=0}^{2n-2} c_k x^k = \sum_{k=0}^{2n-2} \sum_{i+j=k} a_i b_j x^k$$

We wish to describe the *target coefficients* $c_k = \sum_{i+j=k} a_i b_j$ as linear combinations of bilinear forms of the form

$$\Big(\sum_{i \in S} a_i\Big)\Big(\sum_{i \in S'} b_i\Big), \quad S, S' \subseteq [n-1] = \{0, 1, \ldots, n-1\}$$

Each such bilinear form represents one field multiplication, and the smallest number required to express the target coefficients equals the *multiplicative complexity* of the polynomial multiplication.

Finding these sets of bilinear forms involves searching a space that is doubly exponential in n. Because of this, we will mostly restrict our attention to the *symmetric bilinear forms*, those for which $S = S'$. Two justifications for this simplification are that heuristically they stand a good chance of efficiently generating the target coefficients, which are themselves symmetric, and also that in practice all known cases admit an optimal solution consisting solely of symmetric bilinear forms. However it should be noted that there do exist optimal solutions containing non-symmetric bilinear forms.

2.2 Method for finding spanning sets of bilinear forms

Barbulescu et al. [1] published a method for finding minimum-size sets of bilinear forms that span a target set. Their method, which substantially reduces the search space, is described below in the context of Karatsuba recurrences.

The first step is to guess the size of the smallest set of symmetric bilinear forms that spans the target polynomials. Call this guess θ. If θ is too low, then no solution will be found. For the cases of $6, 7, 8$-terms θ is $17, 22, 26$, respectively.

We now assume that the target polynomials are contained in a space spanned by θ of the $(2^n - 1)^2$ symmetric bilinear vectors. Checking all spanning sets of size θ is of complexity $\Omega\binom{(2^n-1)^2}{\theta}$, and even if we restrict attention to symmetric bilinear forms as explained above, this is of complexity $\Omega\binom{(2^n-1)}{\theta}$, which is still prohibitively large, even for $n = 7, \theta = 22$ (for $n = 6, \theta = 17$, this is about 2^{50} and thus close to the limit of what we can compute in practice).

The Barbulescu et al. method is as follows: Let $\mathcal{B}$ be the collection of $(2^n - 1)$ symmetric bilinear products and $\mathcal{T}$ the collection of $2n - 1$ target vectors. For a subset $\mathcal{S} \subset \mathcal{B}$ of size $\theta - (2n - 1)$, let $\mathcal{G} = \mathcal{T} \cup \mathcal{S}$ be a generating set of vectors of size θ and let $\mathbf{C}$ be the candidate subspace generated by $\mathcal{G}$.

We compute the intersection $\mathcal{B} \cap \mathbf{C}$ by applying the rank test to all B in $\mathcal{B}$:

$$B \in \mathbf{C} \iff \theta = \mathrm{rank}(\mathbf{C}) = \mathrm{rank}(\langle \mathbf{C}, B \rangle)$$

which can be computed efficiently via Gausian elimination.

Now let $\mathbf{C}' := \langle \mathcal{B} \cap \mathbf{C} \rangle$ be the subspace spanned by the intersection. In order to determine $\mathcal{T} \cap \mathbf{C}'$, the collection of target vectors in $\mathbf{C}'$, we again apply a rank test to all T in $\mathcal{T}$:

$$T \in \mathbf{C}' \iff \mathrm{rank}(\mathbf{C}') = \mathrm{rank}(\langle \mathbf{C}', T \rangle)$$

If all the target vectors are spanned, i.e. if $\mathcal{T}' = \mathcal{T}$, then each set of θ independent vectors in $\mathcal{B} \cap \mathbf{C}$ is a solution.

We iterate through the different choices of $\mathcal{S}$ until a solution is found. This reduces the complexity to $O\left(2^n \binom{(2^n-1)}{\theta-(2n-1)}\right)$, which in the cases of $n = 6, 7, 8$ transforms the problem from computationally infeasible to feasible. For details, see [1].

This method generates a potentially large number of solutions with the target multiplicative complexity. Each such solution allows one to produce an arithmetic circuit that computes the product of two n-term polynomials. [9] describes a way to translate this arithmetic circuit into three $\mathbb{F}_2$-matrices T, R, E, the *top*,*main*, and *extended* matrices. The additive complexities μ_T, μ_R, μ_E, respectively, of these matrices determine the parameters α, β, γ of a recursion (see Figure 2). In the next section we describe our methods for bounding these additive complexities.

3 Finding small circuits for the linear maps determined by each bilinear form

The problem is NP-hard and MAX-SNP hard [4], implying limits to its approximability. In practice, it is not currently possible to exactly solve this problem for matrices of the size that arise in this research. SAT-solvers have been used on small matrices, but at size about 8x20 the methods begin to fail (see [10]). The sizes of the matrices T, R, E in the method of [9] are given in Table 1.

n	T	R	E
5	**13x5**	**9x13**	10x26
6	**17x6**	**11x17**	12x34
7	**22x7**	**13x22**	14x44
8	**26x8**	15x26	16x52

Table 1. Dimensions of linear optimization problems.

For small-enough matrices (those with dimensions in written in **bold**) in Table 1, we used the heuristic of [4] (henceforth the *BMP* heuristic). For the larger matrices we used the randomized algorithm of [3]. More specifically, we used the RAND-GREEDY algorithm with generalized-Paar operation, allowing less than optimal choices in the greedy step (see [3], section 3.4-3.6).

4 Experimental results

We looked for recurrences for 6,7, and 8-way Karatsuba. Only symmetric bilinear forms were considered. There exist spanning sets of bases, of optimal size, that contain one or more non-symmetric bilinear forms. However, it is believed, but has not been proven, that there always exists an optimal size spanning set containing only symmetric bilinear forms.

In the following subsections, we give the best T and R matrices found for $n = 6, 7$, and 8. In each case, the matrix E is defined as follows: letting R_i be the ith row of R, the matrix E is

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

4.1 6-way split

The search included all symmetric bilinear forms. We searched but did not find solutions with 16 multiplications. We conjecture that the multiplicative complexity of multiplying two binary polynomials of size 6 is 17. 54 solutions with 17 multiplications were found. This matches results reported in [1]. For the matrices T and R, the BMP heuristic was used. For the E matrix, RAND-GREEDY was used. The best recurrence thus obtained was

$$M(6n) \leq 17M(n) + 83n - 26.$$

The best Karatsuba recurrence known before this work was ([9])

$$M(6n) \leq 17M(n) + 85n - 29.$$

The matrices are

$$T_6 = \begin{pmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 1&1&0&0&0&0 \\ 0&1&1&0&0&0 \\ 1&1&1&0&0&0 \\ 0&0&0&1&0&0 \\ 1&0&1&0&1&0 \\ 0&1&1&0&1&0 \\ 1&1&0&1&1&0 \\ 0&0&0&0&0&1 \\ 0&1&1&0&0&1 \\ 0&1&0&1&0&1 \\ 1&0&1&1&0&1 \\ 0&1&0&0&1&1 \\ 0&1&1&0&1&1 \\ 0&1&0&1&1&1 \\ 1&1&1&1&1&1 \end{pmatrix} \quad R_6 = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&1&0&1&0&1&1&1&0&0&1&1&0&0&1 \\ 1&0&1&1&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&1&1&1&0&1&1&0&1&1&0&0&1&0&1&1&0 \\ 0&1&0&0&0&1&0&0&0&1&1&0&0&1&1&0&0 \\ 0&1&0&1&0&0&0&0&0&0&1&1&0&1&0&1&0 \\ 0&1&0&0&1&1&1&0&0&1&0&0&1&0&0&1&1 \\ 0&0&0&1&0&0&0&1&0&0&1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \end{pmatrix}$$

4.2 7-way split

The search included all symmetric bilinear forms. There are no solutions with 21 multiplications. This leads us to conjecture that the multiplicative complexity of multiplying two binary polynomials of size 7 is 22. 19550 solutions with 22 multiplications were found, which matches results reported in [1]. For the matrix T the BMP heuristic was used. For the R and E matrices, the RAND-GREEDY heuristic was used.

Both the BMP heuristic and the RAND-GREEDY are randomized algorithms. The way to use these algorithms is to run them many times and pick the best solution found. Since the linear optimization problem is NP-hard, we expect that at some value of n, we should no longer be confident that we can find the optimal solution. In practice, we aimed at running the algorithms about

100 thousand times. Since we wouldn't be able to do this for all 19550 sets of matrices, we proceeded in two rounds. In the first round, we ran the algorithms for 1000 times on each set of matrices. The results yielded four sets of matrices that implied values of the β parameter which were better than the rest. We then ran the algorithms for 100 thousand times on each of the four sets of matrices and picked the best.

The best recurrence thus obtained was

$$M(7n) \leq 22M(n) + 106n - 31.$$

The best Karatsuba recurrence known before this work was ([9])

$$M(7n) \leq 22M(n) + 107n - 33.$$

The matrices are

$$T_7 = \begin{pmatrix} 1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0 \\ 1&1&0&0&0&0&0 \\ 0&0&1&0&0&0&0 \\ 1&0&1&0&0&0&0 \\ 1&0&1&1&0&0&0 \\ 0&1&1&1&0&0&0 \\ 0&0&1&0&1&0&0 \\ 1&0&1&0&1&0&0 \\ 0&0&0&0&0&1&0 \\ 0&1&0&0&0&1&0 \\ 0&0&0&1&0&1&0 \\ 0&1&0&1&0&1&0 \\ 0&1&1&0&1&1&0 \\ 0&0&0&0&0&0&1 \\ 1&0&1&0&0&0&1 \\ 1&0&1&0&1&0&1 \\ 1&1&0&1&1&0&1 \\ 1&1&1&1&1&0&1 \\ 0&0&0&0&0&1&1 \\ 1&0&1&1&0&1&1 \\ 1&1&1&1&1&1&1 \end{pmatrix} \quad R_7 = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&1&1&1&1&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&1&1&0&1&1&0&1&1&1&0&0&0&1&1&0&1 \\ 1&0&0&1&0&0&1&0&0&0&0&1&1&0&1&1&0&1&1&0&0&0 \\ 1&1&1&0&0&0&0&1&0&0&1&1&1&1&0&1&1&0&0&0&1&1 \\ 1&1&0&0&1&0&1&1&0&0&1&0&1&0&0&0&0&1&1&0&0&0 \\ 0&1&0&0&1&1&0&0&0&1&1&0&0&0&0&1&0&0&1&0&1&1 \\ 0&0&0&0&1&0&0&0&1&1&0&0&0&0&0&1&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \end{pmatrix}$$

4.3 8-way split

It is known that the multiplicative complexity of 8-term binary polynomials is at most 26 ([8]). We were not able to improve on this, the search for solutions with multiplicative complexity 25 appears to require either a huge investment in computation time or an improvement in search methods.

For multiplicative complexity 26, we were not able to search the whole space of symmetric bilinear forms. We verified that there are no solutions with either

7 or 8 "singleton" bases (i.e. bases of the form $a_i b_i$), and there are exactly 77 solutions with 6 "singleton" bases. Additionally, we restricted the search space to sets of bases containing the bilinear forms $a_1 b_1$ and $(a_0+a_2+a_3+a_5+a_6)(b_0+b_2+b_3+b_5+b_6)$ and three among the following

$$\begin{gathered}
(a_1+a_3+a_4+a_5)(b_1+b_3+b_4+b_5)\\
(a_1+a_2+a_3+a_6)(b_1+b_2+b_3+b_6)\\
(a_2+a_4+a_5+a_6)(b_2+b_4+b_5+b_6)\\
(a_0+a_2+a_3+a_4+a_7)(b_0+b_2+b_3+b_4+b_7)\\
(a_0+a_1+a_2+a_5+a_7)(b_0+b_1+b_2+b_5+b_7)\\
(a_0+a_1+a_4+a_6+a_7)(b_0+b_1+b_4+b_6+b_7)\\
(a_0+a_3+a_5+a_6+a_7)(b_0+b_3+b_5+b_6+b_7).
\end{gathered}$$

Our search yielded 2079 solutions, including 63 of the 77 solutions with 6 singletons. For the matrix T, the BMP heuristic was used. For the R and E matrices, RAND-GREEDY was used. Among these 2079 solutions, we found one for which T_8 could be computed with 24 gates, R_8 with 59 gates and E_8 with 99 gates.

The matrices are

$$T_8 = \begin{pmatrix}
1&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0\\
1&1&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0\\
1&0&1&0&0&0&0&0\\
0&1&0&1&1&1&0&0\\
0&0&0&0&0&0&1&0\\
0&1&1&1&0&0&1&0\\
0&0&1&1&1&0&1&0\\
1&0&1&0&0&1&1&0\\
1&0&1&1&0&1&1&0\\
0&1&0&0&1&1&1&0\\
0&0&1&0&1&1&1&0\\
0&0&0&0&0&0&0&1\\
1&0&1&0&0&0&0&1\\
1&1&1&0&1&0&0&1\\
1&0&1&1&1&0&0&1\\
1&1&1&0&0&1&0&1\\
0&1&1&0&1&1&0&1\\
1&0&0&1&1&1&0&1\\
0&0&0&0&0&0&1&1\\
1&1&0&1&0&0&1&1\\
1&1&0&0&1&0&1&1\\
1&1&0&1&1&0&1&1\\
1&0&1&0&0&1&1&1\\
1&1&1&1&1&1&1&1
\end{pmatrix}
\quad
R_8 = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&1&0&1&1&0&0&1&1&0&1&0&1&0&0&0&0&1&0&1&1&1&0&1\\
0&0&1&1&0&0&0&1&1&1&0&0&1&0&1&1&0&1&1&0&1&1&1&1&0&0\\
0&0&0&0&1&1&0&0&1&1&0&1&1&1&1&0&1&1&0&0&1&0&0&0&0&1\\
0&0&0&0&0&0&1&1&0&0&0&1&1&1&0&1&0&1&0&1&1&0&1&0&1&1\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&0&1&0&1&1&0&1&1\\
1&1&1&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&0&1&0&0&1&0&0&1\\
1&1&0&1&1&1&0&1&0&0&0&1&0&0&0&1&0&1&0&0&0&1&0&0&0&1\\
0&1&0&1&0&0&1&1&1&1&1&0&1&0&1&1&1&0&0&1&0&0&0&1&1&0\\
0&0&1&1&0&0&0&0&0&1&0&1&0&0&1&1&1&1&1&1&1&0&1&1&0&1\\
0&0&0&0&1&0&0&0&0&1&0&0&0&1&1&0&0&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}$$

This yields the recurrence

$$M(8n) \leq 26M(n) + 147n - 40.$$

The new recurrence for 8-way Karatsuba may be of practical interest. The smallest known Karatsuba-based circuit for multiplying two polynomials of size 96 has 7110 gates ([9]). Using the new recurrence, along with $M(12) \leq 207$, yields

$$M(96) = M(8 \cdot 12) \leq 26 \cdot 207 + 147 \cdot 12 - 40 = 7106.$$

5 Implications for the circuit complexity of binary polynomial multiplication

This work yielded three new Karatsuba recurrences:

$$M(6n) \leq 17M(n) + 83n - 26$$
$$M(7n) \leq 22M(n) + 106n - 31$$
$$M(8n) \leq 26M(n) + 147n - 40.$$

As per [9], the circuits for these recurrences can be leveraged into circuits for multiplication of binary polynomials of various sizes. Doing this, we found that the new recurrences improve known results for Karatsuba multiplication starting at size 28. The circuits were generated automatically from the circuits for each set of matrices for $n = 2, \ldots, 8$ (the cases $n = 6, 7, 8$ are reported in this work). We generated the circuits up to $n = 100$. The circuits were verified by generating and validating the algebraic normal form of each output. Table 2 compares the new circuit sizes and depths to the state of the art as reported in [9]. The table starts at the first size in which the new recurrences yield a smaller number of gates. The circuits have not been optimized for depth. The circuits will be posted at `cs-www.cs.yale.edu/homes/peralta/CircuitStuff/CMT.html`.

A different approach to gate-efficient circuits for binary polynomial multiplication is to use interpolation methods. These methods can yield smaller circuits than Karatsuba multiplication at the cost of higher depth (see, for example, [6,7]). An interesting open question is to characterize the depth/size tradeoff of Karatsuba versus interpolation methods for polynomials of sizes of practical interest. In elliptic curve cryptography, multiplication of binary polynomials with thousands of bits is used.

References

1. Barbulescu, R., Detrey, J., Estibals, N., Zimmermann, P.: Finding optimal formulae for bilinear maps. In: Arithmetic of Finite Fields, pp. 168–186. Springer (2012)
2. Bernstein, D.J.: Batch binary Edwards. In: Halevi, S. (ed.) Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5677, pp. 317–336. Springer (2009). https://doi.org/10.1007/978-3-642-03356-8_19, `http://dx.doi.org/10.1007/978-3-642-03356-8_19`

n	size in [9]	new size	depth [9]	new depth	n	size in [9]	new size	depth in [9]	new depth
28	944	943	14	15	64	3673	3673	13	13
29	1009	1009	13	13	65	3920	3920	15	15
30	1038	1038	13	13	66	4041	4041	15	15
31	1113	1113	12	12	67	4152	4152	14	14
32	1156	1156	11	11	68	4220	4220	14	14
33	1271	1271	12	12	69	4353	4353	14	14
34	1333	1333	12	12	70	4417	4417	14	14
35	1392	1392	11	11	**71**	4478	4456	25	20
36	1428	1428	11	11	**72**	4510	4489	25	20
37	1552	1552	15	15	73	4782	4782	18	18
38	1604	1604	14	14	74	4815	4815	18	18
39	1669	1669	14	14	75	4847	4847	18	18
40	1703	1703	14	14	76	5075	5075	17	17
41	1806	1806	16	17	77	5198	5198	16	16
42	1862	1859	16	17	78	5255	5255	16	16
43	1982	1982	15	16	79	5329	5329	16	16
44	2036	2036	12	12	80	5366	5366	16	16
45	2105	2105	14	14	81	5593	5593	19	20
46	2179	2179	14	14	**82**	5702	5697	19	19
47	2228	2228	13	13	**83**	5769	5760	18	19
48	2259	2259	13	13	**84**	5804	5795	18	19
49	2436	2436	14	14	**85**	6118	6115	18	19
50	2523	2523	17	17	**86**	6224	6221	19	20
51	2663	2663	14	14	87	6344	6344	18	19
52	2725	2725	13	13	88	6413	6413	15	15
53	2841	2825	24	19	**89**	6516	6488	28	23
54	2878	2863	24	19	**90**	6550	6523	28	23
55	2987	2984	17	18	91	6776	6776	17	17
56	3022	3017	17	18	92	6842	6842	16	16
57	3145	3145	15	15	93	6929	6929	18	19
58	3212	3211	17	18	94	7010	7010	16	16
59	3273	3273	15	15	**95**	7073	7071	15	25
60	3306	3306	15	15	**96**	7110	7106	16	25
61	3472	3472	15	15	97	7465	7465	17	17
62	3553	3553	15	15	98	7636	7636	20	20
63	3626	3626	14	14	99	7801	7801	19	19

Table 2. New circuit sizes and depths for $n = 28$ to 99. Values of n for which we obtained and improvement in size are in **bold**.

3. Boyar, J., Find, M.G., Peralta, R.: Small low-depth circuits for cryptographic applications. Cryptography and Communications (Mar 2018).

https://doi.org/10.1007/s12095-018-0296-3, https://doi.org/10.1007/s12095-018-0296-3
4. Boyar, J., Matthews, P., Peralta, R.: Logic minimization techniques with applications to cryptology. J. Cryptology **26**(2), 280–312 (2013)
5. Brent, R.P., Gaudry, P., Thomé, E., Zimmermann, P.: Faster multiplication in GF(2)[x]. In: van der Poorten, A.J., Stein, A. (eds.) Algorithmic Number Theory, 8th International Symposium, ANTS-VIII, Banff, Canada, May 17-22, 2008, Proceedings. Lecture Notes in Computer Science, vol. 5011, pp. 153–166. Springer (2008). https://doi.org/10.1007/978-3-540-79456-1_10, http://dx.doi.org/10.1007/978-3-540-79456-1_10
6. Cenk, M., Hasan, M.A.: Some new results on binary polynomial multiplication. Journal of Cryptographic Engineering **5**, 289–303 (2015)
7. De Piccoli, A., Visconti, A., Rizzo, O.G.: Polynomial multiplication over binary finite fields: new upper bounds. Journal of Cryptographic Engineering (Apr 2019). https://doi.org/10.1007/s13389-019-00210-w, https://doi.org/10.1007/s13389-019-00210-w
8. Fan, H., Hasan, M.A.: Comments on "five, six, and seven-term karatsuba-like formulae". IEEE Trans. Computers **56**(5), 716–717 (2007)
9. Find, M.G., Peralta, R.: Better circuits for binary polynomial multiplication. IEEE Transactions on Computers **68**(4) (April 2018). https://doi.org/10.1109/TC.2018.2874662
10. Fuhs, C., Schneider-Kamp, P.: Optimizing the AES S-box using SAT. In: Proc. International Workshop on Implementation of Logics (IWIL) (2010)
11. Karatsuba, A.A., Ofman, Y.: Multiplication of multidigit numbers on automata. Soviet Physics Doklady **7**, 595–596 (1963), available at: http://cr.yp.to/bib/entries.html#1963/karatsuba
12. Montgomery, P.L.: Five, six, and seven-term Karatsuba-like formulae. IEEE Trans. Computers **54**(3), 362–369 (2005). https://doi.org/10.1109/TC.2005.49, http://doi.ieeecomputersociety.org/10.1109/TC.2005.49