Journal of Mathematical Sciences, Vol. 191, No. 5, June, 2013

COMPLEXITY OF COMPUTATION IN FINITE FIELDS

S. B. Gashkov and I. S. Sergeev UDC 512.624

ABSTRACT. We give a review of some works on the complexity of implementation of arithmetic operations in finite fields by Boolean circuits.

Introduction

Efficient implementation of arithmetic in finite fields is of primary importance for cryptography, coding theory, digital signal processing, etc. (see, e.g., [2, 3, 24, 26, 27, 29, 30, 83, 103, 104]). In this survey, we consider only Boolean circuits for arithmetic operations in finite fields. Another term is bit-parallel circuits. Boolean circuits for multiplication and inversion in finite fields are implemented physically on chips and are tailored for particular applications. These circuits are usually called multipliers and invertors. In practice, the main interest lies in fields of characteristic 2, but some fields of odd characteristic are also involved. In the latter case elements of a field are coded by binary strings. Boolean circuits are composed from Boolean two-input cells (or gates) AND, NAND, OR, NOR, XOR, XNOR, connected by wires. The *depth* of a given circuit is the length of the longest directed path, connecting a primary input and an output of the circuit. The *complexity* of a given circuit (in other words, the size of a circuit) is the number of cells in it. This notion is very close to the notion of bit complexity of computation (program). All necessary definitions may be found in [50, 99, 138]. Minimization of the depth and the complexity of circuits is a central and practically important problem in complexity theory.

In practice, often the so-called *circuits with memory* (i.e., finite automata) are exploited. Numerous papers deal with the implementation of finite field arithmetic on such circuits. This subject needs a special review and is not included in the survey.

In some theoretical papers on computer arithmetic, *Turing machines* are used as a computational model. They function via reading and overwriting information stored on a tape by a read/write head (i.e. as an automaton). Various types of Turing machines are known: multitape, pointer, with memory, etc. As far as this concept is mainly of theoretical interest, it is also omitted in the review.

A computer program is a popular model to implement finite field arithmetic. If a program does not include cycles and conditional jumps, it appears in fact to be a nonbranching program. The latter notion can be formalized in such a manner that it will turn out to be identical with the notion of circuit. The execution time of the program can be roughly estimated by the complexity of the corresponding circuit. To be more accurate, one must keep in mind that the execution time of different primitive operations on a computer differs. Further in the review, some results concerning program implementation are also mentioned, though it can be the subject of an individual review.

A field of order q is denoted by $GF(q)$. Elements of $GF(q^n)$ may be represented by polynomials over $GF(q)$ of degree at most $n-1$. If elements of $GF(q^n)$ are represented in the *standard basis*

$$B_\alpha = \{\alpha^0, \alpha^1, \dots, \alpha^{n-1}\}$$

(the element $\alpha \in GF(q^n)$ is called the generator of B_α), then multiplication in B_α amounts to polynomial multiplication modulo an irreducible polynomial $g(x)$ over $GF(q)$ such that $g(\alpha) = 0$. If the conjugate elements $\alpha, \alpha^q, \alpha^{q^2}, \dots, \alpha^{q^{n-1}}$ are linearly independent over $GF(q)$, then they form a basis

$$B^\alpha = \{\alpha^{q^0}, \alpha^{q^1}, \dots, \alpha^{q^{n-1}}\},$$

Translated from Fundamentalnaya i Prikladnaya Matematika, Vol. 17, No. 4, pp. 95–131, 2011/12.

which is called a *normal basis* with generator α. (Theoretical background on finite fields may be found in [83, 96].) The complexity of implementation of multiplication and inversion in $GF(q)$ are denoted by $\mathrm{M}\big(GF(q)\big)$ and $\mathrm{I}\big(GF(q)\big)$, respectively. We also introduce the notation $\mathrm{D}_{\mathrm{M}}\big(GF(q)\big)$ and $\mathrm{D}_{\mathrm{I}}\big(GF(q)\big)$, respectively, for the depth of the operations.

Similar notation is used for other operations. Sometimes it is convenient to consider calculations over a subfield $GF(p)$. For corresponding complexity and depth measures we use the same notation with superscript (p) like $\mathrm{M}^{(p)}\big(GF(q)\big)$ or $\mathrm{D}_{\mathrm{I}}^{(p)}\big(GF(q)\big)$.

1. Integer Arithmetic

Circuits implementing elementary numeric operations (namely, operations modulo p, where p is prime) are used as building blocks for circuits implementing operations in finite fields (of order p^n). This is why we discuss also some issues related to implementation of integer arithmetic.

1.1. Addition. At first sight it may be striking, but even the problem of synthesis of efficient (in one or another sense) circuits for addition or subtraction is not trivial. Various circuits are described in books and papers on computer arithmetic. We list some theoretical results below.

The complexity of addition (subtraction) of n-bit numbers (corresponding circuits are usually called adders or subtractors) is known to be $\mathrm{A}(n) = 5n - 3$, due to N. P. Red'kin [118]. Such circuits are easy to build but the lower bound proof is rather complicated (in [118], the tight complexity of an adder built from conjunctions, disjunctions and negations is also found).

The problem of minimization of the depth of an adder appears to be complicated even in constructing aspect. A method due to V. M. Khrapchenko [89] allows one to build an adder of depth $\log n + \sqrt{\big(2 + o(1)\big)\log n}$ (here and further on "log" denotes binary logarithm); the complexity of this circuit can be reduced to $\big(8 + o(1)\big)n$ [62]. In practice, when n is no more than several thousands, other methods result in better circuits.

Some techniques for building such adders are presented in [62] (including a ternary method due to M. I. Grinchuk with depth bound $1.262 \log n + 2.05$).

Recently Grinchuk invented an adder with depth $\log n + \log\log n + 6$ [76]. This adder is also the best known for small values of n.

Khrapchenko [91] at 2007 proved the following lower bound for the depth of an adder (built of AND, OR, and NOT cells):

$$\log n + \big(1 - o(1)\big) \log\log\log n.$$

1.2. Multiplication. Numeric multiplication is evidently a more complex operation than addition. The reader can find a comprehensive analysis of theoretical aspects of implementation of multiplication in [20]. In the present paper, we briefly consider both practical and theoretical aspects.

The complexity of multiplication of n-bit numbers is denoted by $\mathrm{M}(n)$. It is well known that the complexity of a standard multiplier is $6n^2 - 8n + O(1)$ (clearly, one should use binary, not decimal, version of the algorithm).

It is less evident that a standard multiplier can be constructed so that its depth is reduced to $O(\log n)$ (using a method proposed independently by G. K. Stolyarov [130], A. Avizienis [7], Yu. P. Ofman [86], and C. Wallace [137]).

Minimization of the depth of a standard multiplier is one of the extensively studied problems of computer science. Essential results related to the problem have been established by V. M. Khrapchenko [90]. A trick from [48] allows one to reduce the multiplier complexity to $5.5n^2 - 6.5n$.

To the best of our knowledge, the best current asymptotic upper bound is $4.44 \log n + O(1)$ (see [77, 116, 117]). A more practical method leads to the depth estimate $5 \log n + 5$ [127]. This method also provides a benefit in terms of complexity.

The earliest method of reducing the complexity of an integer multiplier is due to A. A. Karatsuba [86] (at that time he was a post-graduate student of Moscow State University; the problem was set by A. N. Kolmogorov). He made an interesting historical review on fast arithmetic algorithms in [85]. The recursive complexity estimate of Karatsuba's integer multiplier is

$$\mathrm{M}(2n) \le 3\mathrm{M}(n) + 52n - 9.$$

The upper bound for $n = 2^s$ is[1]

$$\mathrm{M}(n) \le \frac{1463}{54} \cdot n^{\log 3} - 52n + 4.5.$$

Karatsuba's multiplier has lower size than a standard one for $n \ge 17$. But its depth is $O(\log^2 n)$. Similarly to the case of a standard multiplier, the depth of Karatsuba's multiplier can be reduced to $O(\log n)$ (see, e.g., [138]). In [43], a somewhat better construction was presented,[2] but in any case multiplicative constants in estimates for depth and complexity are exceedingly large for practical applications (the depth of Karatsuba's multiplier can be further reduced to $\bigl(10 + o(1)\bigr) \log n$, causing a further increase of the multiplicative constant in the complexity estimate [127]).

An asymptotically better multiplier was constructed by A. L. Toom [134] (at that time he was a student of Moscow State University; his scientific adviser was O. B. Lupanov). Constants in Toom's estimate were subsequently refined; S. Cook in his thesis [46] adapted the method to Turing machines; A. Schönhage developed a modular method with a similar complexity estimate (the reader can find a more detailed review in [93]).

Toom's multiplier was improved by A. Schönhage and V. Strassen [123] (see also [70]). The complexity of the last multiplier is $O(n \log n \log\log n)$, and the depth is $O(\log n)$ (more precisely, a bound $\bigl(9 + o(1)\bigr) \log n$ can be achieved [127]). It was also claimed in [123] that the same complexity estimate is valid for Turing machine multiplication.

The best known multiplier can be constructed by M. Fürer's method [58] (2007); its complexity is

$$n \log n 2^{\log^* n},$$

but its depth is $O(\log n \log^* n)$ (worse than in Schönhage–Strassen's method). Here $\log^* n$ is a very slowly growing function defined by

$$\Bigl\lfloor \underbrace{\log \ldots \log}_{\log^* n} n \Bigr\rfloor = 1.$$

In [47], a modular version of Fürer's algorithm was posed.

Evidently, neither of the last two multipliers can find applications in cryptography, due to large multiplicative constants in the estimates. Some ways for speeding up program implementation of Schönhage–Strassen's algorithm were considered in detail in [74].

Pollard's multiplier [103, 109] seems to have more chances for finding practical applications, but also could not be used in cryptography. It was noted by Ya. V. Vegner that the complexity of Pollard's multiplier is less than Karatsuba's one only for $n > 2^{22}$. In that paper, the bounds $30\,634 n \log n + 393n$ for the complexity and $349 \log n + 50$ for the depth of Pollard's circuit were claimed under the restriction $n < 201\,326\,604$.

Asymptotic efficiency (and practical inefficiency) of all the above methods (except Karatsuba's and Toom's ones) relies on multiple implementation of Fast Discrete Fourier Transforms (either under the complex field or under Fermat residue rings).

From the practical point of view, Toom's method is the best known. Using Toom's method, A. A. Burtzev has built a multiplier with recursive estimate of complexity

$$\mathrm{M}(4n) \le 7M(n) + 662n + 1085,$$

[1] The bound given in [86] is $\mathrm{M}(n) = O(n^{\log 3})$. Constants in the above and below formulas were obtained by A. A. Burtzev in his degree work.

[2] The benefit of this construction was confirmed by V. V. Baev in his degree work.

which leads, for $n = 4^s$, $s \geq 4$, to the upper complexity bound

$$\mathrm{M}(n) \leq 402.5n^{\log_4 7} - \frac{662}{3}n - \frac{1085}{6}.$$

In particular, $M(1024) \leq 1\,279\,651$. Karatsuba's method gives a worse bound in this case. With the use of technique from [43], the depth of Toom's multiplier can be reduced to $O(n \log n)$.

1.3. Division. The "school" division method allows one to build a circuit for division of a $2n$-bit number by an n-bit one of complexity $O(n^2)$ and depth $O(n \log n)$. Best of the analogous circuits known in computer arithmetic have the same complexity, but depth $O(n)$.

Efficient implementation of division (including depth minimization) seems to be an even more complicated problem than multiplication. However, it can be reduced to multiplication via the Newton–Raphson method.

The reduction was accomplished in [46] (see also [61,93]). The complexity of Cook's circuit is asymptotically five times greater than that of a multiplier, and the depth is $O(\log^2 n)$. However, if n is small, then school division circuits have less complexity and slightly more depth.

The method of [119] allows one to reduce it to $O(\log n \log\log n)$. The size is of the same order as for $O(\log n)$-depth multipliers in both cases. Employing Fürer's technique leads to a circuit with somewhat higher estimated depth.

In [19], a circuit for division of depth $O(\log n)$ and complexity $O(n^5)$ was produced. In [80], division circuits of depth $O(\epsilon^{-2} \log n)$ and complexity $O(n^{1+\epsilon})$ for any positive parameter ϵ were constructed.

However, all proposed methods except for the first one seem to be of academic interest only.

1.4. Prime Field Arithmetic. Arithmetic in a finite field of prime order p is just the integer arithmetic modulo p. The complexity of multiplication modulo any natural p is not greater than $3\mathrm{M}(\log p) + O(\log p)$. To get this estimate one can perform the usual multiplication, then calculate the remainder of division of the $\lceil 2 \log p \rceil$-bit product by the $\lceil \log p \rceil$-bit number p. The latter operation may be implemented by the so-called Barret's method [16] (see also [61,104]). It is very likely that this method originates from the papers [46,131]. For some particular moduli like $p = 2^n \pm c$, $c = O(\log n)$, the above complexity estimate may be improved to $\mathrm{M}(\log p) + O(\log p)$.

The complexity of addition (or subtraction) modulo an n-bit number p can be estimated as $2\mathrm{A}(n) + O(1)$. For Mersenne primes $p = 2^n - 1$ this bound can be reduced to $A\big(GF(p)\big) = 7n - 5$. The depth in the last case is the same up to $O(1)$ as the depth of integer addition-subtraction. The same depth bound holds also for a Fermat prime $p = 2^n + 1$; the complexity in this case is $A\big(GF(p)\big) = 9n + O(1)$.

Multiplication by 2^k in the Mersenne prime field for any integer k amounts to the cyclic shift, which costs nothing in terms of circuit complexity. The complexity of multiplication by integer C, where $C \bmod p$ can be represented as a sum of $l(C)$ powers of two, can be estimated as $\mathrm{M}(C,p) \leq (l(C) - 1)A\big(GF(p)\big)$. For instance, $\mathrm{M}(17,p) \leq A\big(GF(p)\big)$.

Analogously for multiplication by 2^k in Fermat prime field the following complexity and depth estimates can be obtained:

$$\mathrm{M}(2^k, p) \leq \frac{5\mathrm{A}(GF(p))}{9} + O(1),$$
$$\mathrm{D_M}(2^k, p) = \big(1 + o(1)\big) \log n \leq 2 \log n.$$

In the general case of multiplication by C, the complexity estimate takes the form

$$\mathrm{M}(C,p) \leq (l(C) - 1)A\big(GF(p)\big) + \big(5n + O(1)\big)l(C).$$

For instance, $\mathrm{M}(3,p) \leq 14A\big(GF(p)\big)/9 + O(1)$.

The estimates $6n^2 - n + O(1)$ and $4.44 \log n + O(1)$ are known for complexity and depth of a standard multiplier modulo a Mersenne prime p. In the Fermat case, analogous estimates are $6n^2 + 11n + O(1)$ and $4.44 \log n + O(1)$.

2. Multiplication in General Finite Fields

Let $\mathrm{M}_{q,f}(n)$ be the total number of operations over $GF(q)$ (or the complexity over $GF(q)$) required for multiplication of polynomials modulo f, $\deg f = n$. Similarly one can define multiplicative complexity $\mathrm{m}_{q,f}(n)$ and additive complexity $\mathrm{a}_{q,f}(n)$ (i.e., the number of multiplicative and additive operations over $GF(q)$, respectively). Then

$$\mathrm{M}\big(GF(q^n)\big) \le \mathrm{M}_{q,f}(n)\mathrm{M}\big(GF(q)\big)$$

for any irreducible polynomial $f(x)$ over $GF(q)$. To be more precise,

$$\mathrm{M}\big(GF(q^n)\big) \le \mathrm{m}_{q,f}(n)\mathrm{M}\big(GF(q)\big) + \mathrm{a}_{q,f}(n)A\big(GF(q)\big).$$

We also use the notation $\mathrm{M}_q(n)$ for the complexity over $GF(q)$ of multiplication of polynomials of degree less than n. Analogously $\mathrm{m}_q(n)$ and $\mathrm{a}_q(n)$ denote multiplicative and additive complexity.

Strassen's method [131] (see also [70]) implies that for any f

$$\mathrm{m}_{q,f}(n) \le 3\mathrm{m}_q(n), \quad \mathrm{a}_{q,f}(n) \le 3\mathrm{a}_q + O(n).$$

In [61], another algorithm with the same complexity estimate was proposed. The algorithm is a polynomial analogue of Barret's algorithm (as well as Barret's algorithm is its numeric analogue). If $f(x)$ is a sum of k monomials, then

$$\mathrm{M}_{q,f}(n) \le \mathrm{M}_q(n) + (2k+1)n,$$

and if $q = 2$ then

$$\mathrm{M}_{2,f}(n) \le \mathrm{M}_2(n) + kn.$$

It is a well-known hypothesis that one can always choose an irreducible polynomial f with $k \le 5$. Therefore,

$$\mathrm{M}_{q,f}(n) \le \mathrm{M}_q(n)\big(1 + o(1)\big).$$

In [122] (see also [70]), it is proved that the estimates

$$\mathrm{m}_q(n) = O(n \log n), \quad \mathrm{a}_q(n) = O(n \log n \log\log n)$$

can be achieved simultaneously. In [41], a multiplicative constant in this estimate was refined. But both methods seem not to be applicable in cryptography or coding theory for the reason that this constant is too large.

It is known (see, e.g., [109]) that in the case $2n-1 \le q$ the multiplicative complexity of multiplication in $GF(q^n)$ is $2n-1$. The main idea of the upper bound was proposed by A. L. Toom [134] and the proof of the lower bound is due to S. Winograd (see, e.g., [24]).

It was shown by the Chudnovsky brothers [45] that in the general case the multiplicative complexity is $O(n)$ as well. More accurate estimates were obtained in [128]. However, as it is known now, both papers contain shortcomings in the proofs. One can find correct proofs and improved estimates in several papers by Ballet et al. (see paper [15] and the references therein). On the other hand, the additive complexity of these methods is not that low. Therefore, the above methods seem to have no practical applications.

2.1. Polynomial Multiplication. First, consider the case of binary polynomial multiplication. The complexity and depth estimates of the "school" method are

$$\mathrm{M}(n) = n^2 + (n-1)^2, \quad \mathrm{D_M}(n) = 1 + \lceil \log_2 n \rceil.$$

For $n \approx 1000$ one has $\mathrm{M}(n) \approx 2\,000\,000$, $\mathrm{D}(n) = 11$.

The recursive complexity estimates for Karatsuba's method look as follows:

$$\mathrm{M}(2n) \le 3\mathrm{M}(n) + 7n - 3, \quad \mathrm{M}(2n+1) \le 2\mathrm{M}(n+1) + \mathrm{M}(n) + 7n - 1,$$

implying for $n = 2^k$, $k \ge 3$, the following relations:

$$\mathrm{M}(n) \le \frac{103}{18}3^k - 7n + \frac{3}{2}, \quad \mathrm{D_M}(n) \le 3k - 3.$$

In particular, for $n = 1024$ we have $\mathrm{M}(n) \le 330\,725$, $\mathrm{D}(n) \le 27$.

Using Schönhage's [122] FFT method, a circuit for cyclic convolution with complexity $Z(2187) \leq 428\,351$ and depth $\mathrm{D}_Z(2187) \leq 46$, or a circuit with bounds $Z(2187) \leq 430\,537$ and $\mathrm{D}_Z(2187) \leq 34$ can be constructed. As a corollary we have

$$\mathrm{M}(1024) \leq \mathrm{M}(1093) \leq 430\,537, \quad \mathrm{D}_{\mathrm{M}}(1024) \leq \mathrm{D}_{\mathrm{M}}(1093) \leq 34.$$

In this case, Karatsuba's multiplier is more efficient.

On the other hand, Karatsuba's method for convolution allows one to build circuits with

$$Z(2048) \leq 998\,216, \quad D_Z(2048) \leq 30.$$

In this case, the FFT method is preferable.

Another example: multiplication modulo $x^{1458} + x^{729} + 1$ can be implemented using the FFT method with complexity 273 850 and depth 33. In this case, Karatsuba's method again comes into play.

So, the point where Schönhage's multiplier becomes advantage over Karatsuba's one lies somewhere after $n = 1000$.

In [21], one can find accurate estimates of the complexity of the multipliers of binary polynomials of small degrees based on the methods of Karatsuba and Toom.

There also exists D. Cantor's method [40] for polynomial multiplication over finite fields. The asymptotic complexity of this method is slightly greater than the FFT method (e.g., $O(n \log^{1.59} n)$ for multiplication over $GF(2)$ and $O(n \log^2 n)$ for multiplication over any finite field), but for some medium-sized fields Cantor's method may be preferable. In [69], a modification of Cantor's method and some applications to polynomial factorization were considered. An improved version of the algorithm was proposed by Gao and Mateer (see [102]).

Cantor's method can be viewed as some refinement of Toom's method. For interpolation it exploits as nodes elements of affine subspaces over $GF(2)$ of appropriate extension field $GF(2^n)$. Here the polynomial whose roots are the above nodes has few nonzero coefficients. So the interpolation polynomial is easy to calculate. Certainly, it is more efficient to choose the roots of the binomial $x^n - 1$ as nodes. This is equivalent to using a Discrete Fourier Transform (DFT) of order n. To implement an n-point FFT, one requires $O(n \log n)$ operations in the minimal field $GF(2^m)$ containing all nth roots of unity. It is clear that $2^m > n$. Therefore, the complexity of multiplication in the field $GF(2^m)$ (the best known Schönhage's method is used) is greater than $\Omega\big(\log n (\log\log n)(\log\log\log n)\big)$. Hence the total complexity of multiplication of n-degree polynomials over $GF(2)$ following the above way can be bounded as

$$\Omega\big(n \log^2 n (\log\log n)(\log\log\log n)\big).$$

This bound is not easy to achieve following this way, because n must divide $2^m - 1$, which prevents it from having the form 2^k, which is convenient to perform DFT fast, and rarely allows one to have the form 3^k or 5^k. Nevertheless, sometimes this bound is achievable. Following the method from [60], consider $n = 2^{p-1}$, where $q = 2^p - 1$ is a Mersenne prime. For multiplication of n-degree polynomials over $GF(2)$ it is sufficient to multiply these polynomials as polynomials with integer coefficients 0, 1. The last operation may be performed as the multiplication of polynomials over $GF(q)$ with the use of a $2n$-point FFT over $GF(q^2)$. It is known that $(2^{n/2} + 3^{n/2} i)^2 \in GF(q^2)$ is a $2n$th root of unity, where $i \in GF(q^2)$ is the root of the irreducible polynomial $x^2 + 1$ over $GF(q)$. As was proved in [60], a $2n$-point FFT over $GF(q^2)$ may be computed using $(3/2) n \log n + O(n)$ multiplications and $6n \log n + O(n)$ additions in the field $GF(q)$.[3]

From [60, formula (3)] it follows that

$$m_q(n) \leq \frac{9}{2} n \log n + O(n), \quad a_q(n) \leq 18 n \log n + O(n),$$

[3] In [60], the formula for a primitive 8th root of unity was printed incorrectly. The proper one is $\epsilon = 2^{-(p+1)/4}(1 + i)$. Also the number of additions required for the computation of an n-point FFT was given inaccurately. The right number is $3n \log n$.

whence the complexity of multiplication of n-degree polynomials over $GF(2)$ is

$$\mathrm{M}_2(n) = a_q(n)A\big(GF(q)\big) + m_q(n)\mathrm{M}\big(GF(q)\big) = \frac{9}{2}\mathrm{M}(p)n\log n + O(np\log n) = O\big((p\log p\log\log p)n\log n\big).$$

Note that the multiplicative constant in the above estimate is rather large owing to the large constant in the estimate of $\mathrm{M}(p)$. The number n has a special form (for other n the constant would be even larger). Moreover, it is still an open question whether the set of Mersenne numbers is infinite. Let $p = 17$, $q = 2^{17} - 1$, $n \approx 2^{16}$. Then

$$\mathrm{M}_2(n) \approx 27 \cdot n\log^3(2n) = 2^{16}17^3 27.$$

This bound is close to the complexity of standard school method of multiplication. Hence, the given method of multiplication is better than the standard method only if n is greater than 70 000.

Program implementation of the above method seems to be more challenging. As follows from [60], the multiplication of polynomials of degree $n < 2^{p-1}$ over $GF(q)$, where $q = 2^p - 1$ is a Mersenne prime, may be performed by $(9/2)n\log n + 58n + 1$ operations in $GF(q^2)$ (in [60] the last bound was printed incorrectly). If multiplication and addition tables for $GF(q^2)$ are stored in a computer memory (it is enough to keep only the table of volume $(n-1)q^2 \le q^3/2$ for multiplication on nth roots of unity, because each of n general multiplications may be performed using 6 operations modulo q), then for $q = 127$ the given method of multiplication for 63-degree polynomials uses almost the same time as the school method. However, to multiply polynomials of higher degrees, one should increase the field order (and consequently the size of computer memory).

After having read this, the reader can appreciate Schönhage's trick [122]. Schönhage's technique for polynomial multiplication involves FFT in the ring $GF(2)[X]/(x^n + 1)$ and leads to the complexity bound $O(n\log n\log\log n)$. Strangely enough, this estimate is still not so widely known: the authors know several papers with similar of weaker results in which [122] is not cited. For instance, in [136] a later paper followed (and DFT is used for division instead of using Strassen's trick mentioned above, which seems unknown to the author of [136]).

Various aspects of program implementations of multiplication of polynomials over both binary and any field $GF(p)$, including algorithms based on methods of Karatsuba, Toom, Schönhage, and D. Cantor, are discussed in [9, 30, 32, 79, 104, 110]. In [56], there was suggested one more algorithm for program multiplication modulo an irreducible trinomial based on multiplication of a Toeplitz matrix by vector. It is not clear whether the algorithm [56] is faster than the algorithms [79], since the work [79] was not cited in [56] (though the authors of both papers work in the same institute).

2.2. Multiplication in Standard Bases. Various architectures of multipliers for standard bases were proposed in [54, 78, 101]. Generally, the complexity and depth of these multipliers are estimated as $O(n^2)$ and $O(\log n)$, respectively.

In [4], it was shown that sometimes using a standard basis with irreducible polynomials of maximum weight, i.e., polynomials of the form

$$1 + x + \cdots + x^{m-1} + x^{m+1} + \cdots + x^n,$$

offers a benefit.

A close idea was used in [49]. More exactly, it was suggested that, instead of a given irreducible polynomial, one takes a trinomial divisible by this polynomial and then perform multiplication modulo this trinomial (in this case the field is embedded into the ring modulo the trinomial). Also in [49] were given corresponding tables of such trinomials. It is also stated that sometimes this trick is more efficient than using irreducible pentanomials. To get it, the trinomial is to be chosen in such a way that its middle term has a degree not less than a half of the leading term's degree. This can always be done, because the tables for any trinomial include its reciprocal.

A similar idea, but using $x^n - 1$ instead of a trinomial, was suggested in some works about the so-called redundant bases. One more possibility of speeding up modular multiplication is based on the Montgomery method (see, for example, references in [30]).

Multipliers of asymptotic complexity $O(n^{\log 3})$ can be constructed following Karatsuba's method. Some aspects of application of Karatsuba's method to multiplication in $GF(2^n)$ are discussed in [111,113].

For example, multiplication in $GF(2^{1024})$, when an irreducible polynomial is taken to be

$$x^{1024} + x^{19} + x^6 + x + 1,$$

can be implemented by a circuit with

$$\mathrm{M}\big(GF(2^{1024})\big) \le 356\,865, \quad \mathrm{D_M}\big(GF(2^{1024})\big) \le 31.$$

2.3. Multiplication in Normal Bases. Numerous methods for multiplication in normal bases are known by now, e.g., [25,28,72,83,100,120]. Let $T = (t_{i,j})$ be a matrix whose ith row is the vector of entries of $\alpha\alpha^{q^i} \in GF(q^n)$ with respect to a normal basis B^α. The number of nonzero entries in the matrix T is called the complexity of the basis B^α and is denoted $\mathrm{C}(B^\alpha)$. If

$$\xi = \sum_{i=0}^{n-1} x_i \alpha^{q^i}, \quad \zeta = \sum_{j=0}^{n-1} y_j \alpha^{q^j}$$

are some elements of $GF(q^n)$, then the product $\pi = \xi\zeta$ may be computed by the formula

$$\pi = \sum_{m=0}^{n-1} p_m \alpha^{q^m}, \quad p_m = \sum_{i,j=0}^{n-1} t_{i-j,m-j} x_i y_j = A\big(S^m(x), S^m(y)\big),$$

where $S^m(v)$ is the cyclic shift of a given vector v by m positions, $A(u,v)$ is the bilinear form associated with the matrix $A = (a_{i,j})$, with the condition $a_{i,j} = t_{i-j,-j}$, and indices $i-j$ and $-j$ are handled modulo n. This Massey–Omura algorithm [100] for multiplication over normal basis B in $GF(q^n)$ requires $n(2C(B) + n - 1)$ operations over the subfield $GF(q)$. In [120], a more efficient algorithm with the bound $n(C(B) + 3n - 2)/2$ was proposed. But both these bounds are at best quadratic in n, and cubic in the worst case.

Alternatively, an idea of transition to the standard basis representation of the field elements may be exploited. The asymptotically fast polynomial multiplication algorithm with Strassen's trick for modular reduction are to be used to implement multiplication in the standard basis.

The usual method for implementation of such transition rests on the fact that transition is a linear operator over the subfield $GF(q)$. Thus, the transition can be implemented by a circuit of $O(n^2/\log_q n)$ complexity and $O(\log n)$ depth. This is a corollary to a classical result due to O. B. Lupanov [98, 99]. In [126], circuits for transition between standard and normal bases with complexity $O(n^{1.806})$ and depth $O(\log n)$ were constructed. (The same estimate for the complexity of single-direction transition had been proven earlier in [84] with a worse depth bound.) Such transition circuits allow one to perform multiplication in $GF(q^n)$ using $O(n^{1.806})$ operations over $GF(q)$ in depth $O(\log n)$. Exploiting a new algorithm for the Frobenius operation [87,135], Sergeev [126] estimated the complexity of the method as $O\big(n^{1.667} + \sqrt{n}(n\log q)^{1+o(1)}\big)$.[4] In [126], another construction for transition circuits was proposed, which implies for any normal basis B that the following estimates hold simultaneously:

$$\mathrm{M}^{(q)}\big(GF(q^n)\big) = O(\sqrt{n}C(B) + n^{1.667} + n^{1.5}\log q \log n \log\log n),$$

$$\mathrm{D}_{\mathrm{M}}^{(q)}\big(GF(q^n)\big) = O(\sqrt{n}\log q \log n).$$

In particular, if B is a low complexity basis, i.e., $\mathrm{C}(B) = O(n^{1.167})$, and q is small enough, i.e., $\log q = o(n^{0.167})$, then $\mathrm{M}^{(q)}\big(GF(q^n)\big) = O(n^{1.667})$. But multiplicative constants in the estimates above are pessimistic.

For some special but important cases, better bounds are known. Normal bases in $GF(q^n)$ of the minimal complexity $2n-1$ are called optimal normal bases (ONB). All these bases were enumerated in [108]. Any ONB belongs to one of three types. ONB of type I exists if and only if $n + 1 = p$ is

[4] The frequently used constant 1.667 is the exponent of special rectangular matrix multiplication [81].

a prime number and q is a primitive element modulo p. Type II and III ONB exist if and only if $q = 2^m$, $(m, n) = 1$, $2n + 1 = p$ is a prime, and either 2 is a primitive element modulo p (type II) or n is odd and -2 is a primitive element modulo p (type III). The type II or III basis is generated by the element $\alpha = \zeta + \zeta^{-1}$, where $\zeta \in GF(q^{2n})$, $\zeta^p = 1$, $\zeta \neq 1$, and coincides with commutation with the basis

$$\{\alpha_1, \dots, \alpha_n\}, \quad \alpha_k = \zeta^k + \zeta^{-k}, \quad k = 1, \dots, n.$$

The type II and III bases construction may be generalized for $q \neq 2^m$, but in this case the complexity of the bases is larger than $2n - 1$, so they are not optimal, though the bases are of complexity $O(n)$. Various other kinds of low complexity normal bases with $\mathrm{C}(B) = O(n)$ were stated in [6, 57, 72, 124], in particular, Gaussian normal bases (GNB), which are more general than optimal. Using the method of [59], one can obtain the following bound for the type-k[5] Gauss normal basis:

$$\mathrm{M}\big(GF(q^n)\big) \leq (\mathrm{M}_q(kn) + 7kn - 8)\mathrm{M}\big(GF(q)\big).$$

In the particular case of $q = k = 2$ (which is the ONB case), this result was obtained later in [25] independently and was patented. For the type I ONB one has

$$\mathrm{M}\big(GF(q^n)\big) \leq (\mathrm{M}_q(n) + 7n - 8)\mathrm{M}\big(GF(q)\big).$$

For the type II and III ONB the bounds

$$\mathrm{M}^{(q)}\big(GF(q^n)\big) \leq 3\mathrm{M}_q(n) + O(qn \log_q n), \quad \mathrm{M}\big(GF(2^n)\big) \leq 3\mathrm{M}(n) + \frac{3n}{2} \log n + O(n)$$

were proved in [28]. The corresponding construction is settled on the circuit for transition from the basis $\{\alpha_1, \dots, \alpha_n\}$ to the basis $\{\alpha, \dots, \alpha^n\}$, $\alpha = \alpha_1 = \zeta + \zeta^{-1}$ of complexity $O(sn \log_s n)$, where $q = s^m$, s is prime, and depth $O(\log_s n)$. The factor 3 in the above estimate comes from the Strassen's inequality

$$\mathrm{M}^{(q)}\big((GF(q^n)\big) \leq 3\mathrm{M}_q(n) + O(n).$$

This relation implies that the complexity of reduction modulo minimal polynomial f of the standard basis $B_\alpha = \{1, \dots, \alpha^{n-1}\}$ is estimated by $2\mathrm{M}_q(n) + O(n)$.

Under certain conditions (e.g., f has few nonzero coefficients) the latter bound may be improved. For example [30], if $n = 3 \cdot 2^k - 1$ and ONB of type II or III exists, then for the complexity and the depth of multiplication in this basis we have

$$\mathrm{M}\big(GF(2^n)\big) \leq \mathrm{M}(n) + \frac{7n}{2} \log n + 7n + O(\log n),$$
$$\mathrm{D}_{\mathrm{M}}\big(GF(2^n)\big) \leq D(n) + 2 \log n + 2 \log \log n + O(1).$$

In particular,

$$\mathrm{M}\big(GF(2^{191})\big) \leq 31\,600, \quad \mathrm{D}\big(GF(2^{191})\big) \leq 44.$$

For comparison, the method of paper [120] implies the bound $\mathrm{M}\big(GF(2^{191})\big) \leq 90\,916$. Another above-mentioned estimate

$$\mathrm{M}\big(GF(q^n)\big) \leq (\mathrm{M}_q(kn) + 7kn - 8)\mathrm{M}\big(GF(q)\big)$$

for $q = 2 = k$, $n = 191$ with the use of Karatsuba's method leads to the inequality

$$\mathrm{M}\big(GF(2^{191})\big) \leq 77\,441.$$

The recently mentioned algorithm [28] for transition between bases $\{\alpha_1, \dots, \alpha_n\}$ and $\{\alpha, \dots, \alpha^n\}$, $\alpha = \alpha_1 = \zeta + \zeta^{-1}$ was rediscovered in [73] (2007), and the following estimate was established:

$$\mathrm{M}^{(q)}\big(GF(q^n)\big) \leq \mathrm{M}_q(n) + O(qn \log_q n).$$

[5] *Type-k GNB* exists in $GF(q^n)$, when $kn+1$ is prime, and is generated by the element $\alpha = \zeta + \zeta^{\gamma} + \dots + \zeta^{\gamma^{k-1}}$, where ζ is a primitive root of order $kn+1$ in $GF(q^{kn})$ and γ is a primitive root of order k in the residue field $\mathbb{Z}_{kn+1}$, which generates together with q the multiplicative group $\mathbb{Z}_{kn+1} \setminus \{0\}$.

Instead of reduction modulo a minimal polynomial of the basis B_α (as in [28]), the algorithm from [73] implies linear transform between redundant bases $\{\alpha, \dots, \alpha^{2n}\}$ and $\{\alpha_1, \dots, \alpha_{2n}\}$. The complexity of the transform is $O(sn \log_s n)$, where $q = s^m$, and the depth is $O(\log_s n)$, as was shown in [28]. In view of the equalities

$$\alpha_{k+n} = \zeta^{k+n} + \zeta^{-k-n} = \zeta^{k+n-p} + \zeta^{p-k-n} = \zeta_{k-n-1} + \zeta_{n+1-k} = \alpha_{n+1-k}, \quad k = 1, \dots, n,$$

the transition to the basis $\{\alpha_1, \dots \alpha_n\}$ may be performed with the complexity n and the depth 1. As a consequence,

$$\mathrm{M}\big(GF(2^n)\big) \le \mathrm{M}(n) + 2n \log n + 10n, \quad \mathrm{D}_\mathrm{M}\big(GF(2^n)\big) \le \mathrm{D}(n) + 2 \log n + 4.$$

Analogous estimates were obtained in [22].

3. Inversion in Finite Fields

The best known asymptotic complexity estimate for inversion in a standard basis of $GF(q^n)$ over $GF(q)$ is $O(n \log^2 n \log\log n)$. The corresponding algorithm can be derived from the fast extended Euclidean GCD (greatest common divisor) algorithm.

A fast numeric version of this algorithm was stated by Knuth [92] and was optimized by Schönhage [121] (the Knuth–Schönhage algorithm can also be viewed as a modern version of the Euclid–Lehmer GCD algorithm; Lehmer's algorithm can be found in [93]). A polynomial version of this algorithm was published in [105] (see also [5]) but the algorithm works incorrectly in some cases; the correct algorithm can be found in [33, 70, 132]).

Subsequently some modifications were introduced (see, e.g., [106, 129]). Stehle–Zimmermann's algorithm [129] shows considerable promise for polynomial multiplication over $GF(2)$. In practice, all the algorithms are implemented in software because of the great depth of the corresponding circuits, which is $O(n)$ (for numeric version there was constructed a circuit of depth $O(n/\log n)$ and complexity $O(n^{1+\epsilon})$ [44]).

The usual binary GCD algorithm seems to be more efficient for small values of n. Its complexity is $O(n^2)$ (see, e.g., [30, 129]). However, the circuit version has several times greater complexity and the depth $O(n \log n)$. That is why to construct a small depth invertor one must use completely different approaches.

3.1. Addition Chain Method. A sequence of natural numbers $a_0 = 1, a_1, \dots, a_m = n$ in which each number a_i is the sum $a_j + a_k$, where $j, k < i$ (indices j and k may coincide), is called an addition chain for n. The parameter m is called the length of the addition chain. The length of the shortest addition chain for n is denoted by $\mathrm{l}(n)$. Comprehensive study of addition chains including all classical results may be found in [93].

Put $\lambda(n) = \lfloor \log n \rfloor$. It is known that

$$\mathrm{l}(n) = \lambda(n) + \big(1 + o(1)\big)\frac{\lambda(n)}{\lambda(\lambda(n))}.$$

The upper bound is due to A. Brauer [31] and a proof of the lower bound is due to P. Erdős [55].

Evidently raising to the nth power using only multiplications corresponds to constructing an addition chain for n. Fermat's identity $x = x^{q^n}$ for any $x \in GF(q^n)$ implies that inversion in $GF(q^n)$ is equivalent to raising to the power $q^n - 2$. This forms the background for the use of addition chains in constructing invertors.

A. Brauer [31] proposed an appropriate way to build an addition chain for $2^n - 1$ starting from an addition chain for n. His method easily extends to the calculation of $(q^n - 1)/(q - 1)$, where multiplications by q are used instead of doubling steps.

Denote $y = x^{(q^n - q)/(q-1)}$. To calculate the inverse fast, one can use the identity $x^{-1} = y(xy)^{-1}$, as proposed in [82]. Clearly, $xy \in GF(q)$, as far as $(xy)^{q-1} = x^{q^n - 1} = 1$. For the computation of $y = (x^{(q^{n-1}-1)/(q-1)})^q$ either Brauer's method or the Itoh–Tsujii [82] method can be used (actually, the latter is just a special case of Brauer's method). To finish calculations, one must multiply x by y (it is simpler than in the general case, due to the fact that the product belongs to subfield) and divide by $xy \in GF(q)$. In the case $q = 2$, one only needs to calculate $y = x^{-1}$.

A less elegant approach based on the formula

$$x^{-1} = x^{(q^{n-1}-1)q}x^{q-2}$$

was followed in [71].

Let $\mathrm{F}\bigl(GF(q^n)\bigr)$ and $\mathrm{D_F}\bigl(GF(q^n)\bigr)$ denote the maximum on m of the complexity and the depth of the circuit implementing a Frobenius operation $x \to x^{q^m}$ in $GF(q^n)$, $m = 1, \dots, n$. In the standard basis, the Frobenius operation is equivalent to the computation of the polynomial $g^{q^m} \bmod f$ and may be performed as a modular composition $g(h) \bmod f$, where $h = x^{q^m} \bmod f$. Indeed, if

$$g(x) = \sum_{i=0}^{s} a_i x^i,$$

then

$$g^{q^m}(x) = \sum_{i=0}^{s} a_i^{q^m} x^{q^m i} = \sum_{i=0}^{s} a_i x^{q^m i} \bmod f = \sum_{i=0}^{s} a_i h^i \bmod f = g(h) \bmod f.$$

Let $\mathrm{d}(n)$ denote the depth of a shortest addition chain for n. Using the addition chain method and a result of paper [34], Gashkov and Sergeev [64] constructed a standard basis invertor with complexity and depth

$$\mathrm{I}^{(q)}\bigl(GF(q^n)\bigr) \le (\mathrm{l}(n-1)+1)\Bigl(\mathrm{M}^{(q)}\bigl(GF(q^n)\bigr) + \mathrm{F}^{(q)}\bigl(GF(q^n)\bigr)\Bigr) + n = O(n^{1.667}),$$

$$\mathrm{D}_{\mathrm{I}}^{(q)}\bigl(GF(q^n)\bigr) \le (\mathrm{d}(n-1)+1)\Bigl(\mathrm{D}_{\mathrm{M}}^{(q)}\bigl(GF(q^n)\bigr) + \mathrm{D}_{\mathrm{F}}^{(q)}\bigl(GF(q^n)\bigr)\Bigr) + 1 = O(\log^2 n).$$

The same scheme of calculations in the case of a normal basis implies the following bounds:

$$\mathrm{I}^{(q)}\bigl(GF(q^n)\bigr) \le (\mathrm{l}(n-1)+1)\mathrm{M}^{(q)}\bigl(GF(q^n)\bigr) + n = O(n^{1.806}),$$

$$\mathrm{D}_{\mathrm{I}}^{(q)}\bigl(GF(q^n)\bigr) \le (\mathrm{d}(n-1)+1)\mathrm{D}_{\mathrm{M}}^{(q)}\bigl(GF(q^n)\bigr) + 1 = O(\log^2 n),$$

since the Frobenius operation is simply a cyclic shift of a field element coefficients in the normal basis, which has zero complexity, and multiplication in any normal basis can be implemented with complexity $O(n^{1.806})$ and depth $O(\log n)$ [126]. Additive terms n in both complexity bounds and 1 in both depth bounds can be omitted in the case $q = 2$.

The complexity of Brent–Kung method may be estimated as $O(n^{1.667})$. In 2007, Umans [135] proved that the complexity of modular composition is equal to $n^{1+o(1)}$ if the field $GF(q)$ has characteristic $n^{o(1)}$. (The claim from [87] that the estimate $n^{1+o(1)}$ is also valid in any characteristic seems inapplicable to implementation by circuits.) Hence it follows that

$$\mathrm{I}^{(q)}\bigl(GF(q^n)\bigr) \le (\mathrm{l}(n-1)+1)\Bigl(\mathrm{M}^{(q)}\bigl(GF(q^n)\bigr) + \mathrm{F}^{(q)}\bigl(GF(q^n)\bigr)\Bigr) + n = (n \log q)^{1+o(1)}$$

for the standard base and

$$\mathrm{I}^{(q)}\bigl(GF(q^n)\bigr) \le (\mathrm{l}(n-1)+1)\mathrm{M}^{(q)}\bigl(GF(q^n)\bigr) + n = O(n^{1.667})$$

for a normal base in the case where the characteristic is small, in particular in the case of binary fields. But this method does not give the bound $O(\log^2 n)$ for the depth.

The above estimates based on A. Brauer's (1939) method seem hardly familiar to cryptographers. Some particular cases of Brauer's method, like the Itoh–Tsujii method [82] or the TYT-method [133], are frequently cited and exploited. These methods do not provide optimal complexity (for example, the method from [133] yields to the general Brauer method for $n = 24, 44, 47, \dots$). Using Brauer's method, some very recent results can be improved straightforwardly, e.g., the complexity bounds [42] for inversion in the fields $GF(2^{384})$ and $GF(2^{480})$ (see details in [64]).

To minimize the depth of an invertor we may use a version of the right-to-left binary method (see [64,93]). The method allows one to build a minimal depth $\delta(n) = \lceil \log_2 n \rceil$ addition chain for n with the length $\lambda(n) + \nu(n) - 1$, where $\nu(n)$ is the number of 1's in the binary representation of n. The length of such a chain is at most $2\lambda(n)$; this bound is tight.

Using a modified Yao's method [139], an addition chain for n with the depth $\delta(n) + 1$ and asymptotically minimal length

$$\lambda(n) + \frac{\lambda(n)}{\lambda(\lambda(n))} + \frac{O(\lambda(n)\lambda(\lambda(\lambda(n))))}{(\lambda(\lambda(n)))^2}$$

was constructed in [64].

Thereby, a standard-basis invertor of complexity

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) \le \left(\lambda(n-1) + \big(1 + o(1)\big)\frac{\lambda(n)}{\lambda(\lambda(n))}\right)\Big(\mathrm{M}^{(q)}\big(GF(q^n)\big) + \mathrm{F}^{(q)}\big(GF(q^n)\big)\Big)$$

and depth

$$\mathrm{D}_{\mathrm{I}}^{(q)}\big(GF(q^n)\big) \le (\delta(n-1)+1)\Big(\mathrm{D}_{\mathrm{M}}^{(q)}\big(GF(q^n)\big) + \mathrm{D}_{\mathrm{F}}^{(q)}\big(GF(q^n)\big)\Big) + 1$$

can be constructed. Analogous bounds for a normal basis take the form

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) \le \left(\lambda(n-1) + \big(1 + o(1)\big)\frac{\lambda(n)}{\lambda(\lambda(n))}\right)\mathrm{M}^{(q)}\big(GF(q^n)\big),$$

$$\mathrm{D}_{\mathrm{I}}^{(q)}\big(GF(q^n)\big) \le (\delta(n-1)+1)\mathrm{D}_{\mathrm{M}}^{(q)}\big(GF(q^n)\big) + 1.$$

Indeed, for any $n \le 228$ there exists a minimal length chain of depth at most $\delta(n)+1$. For any $n \le 1024$ there exists a minimal length chain of depth at most $\delta(n)+2$.

3.2. Logarithmic Depth Method. The $GF(q^n)$ invertors of logarithmic depth (over $GF(q)$) were presented in [68,97] ([97] considers the binary case). The authors did not estimate the complexity and depth of the circuits more tightly than $n^{O(1)}$ and $O(\log n)$, respectively. In fact, the multiplicative constants involved are rather large. An invertor in $GF(2^n)$ of depth $\big(6.44 + o(1)\big)\log n$ and complexity $(2/3)n^4 + o(n^4)$ was constructed in [125] (the result holds for an arbitrary field basis). In the same paper, a standard basis invertor with depth $O(\log n)$ and complexity $O(n^{1.667})$ was constructed. The latter result was extended in [65] to the case of the general field $GF(q^n)$. As a corollary, a normal basis invertor of complexity $O(n^{1.806})$ and depth $O(\log n)$ can be constructed.

This method [65] looks like a parallel version of addition chain method. It involves multiple multiplications. We denote complexity and depth of multiplication of m elements in the field $GF(q^n)$ by $\mathrm{MM}\big(m, GF(q^n)\big)$ and $\mathrm{D}_{\mathrm{MM}}\big(m, GF(q^n)\big)$, respectively. Combining ideas from [53,80,119], the following bounds for a multiple multiplication circuit were proved in [65]:

$$\mathrm{MM}^{(q)}\big(m, GF(q^n)\big) = O\big(l^c m^{1+\epsilon} n^{1+l^{-3}}(\log(mn)\log\log(mn) + l^3)\big)$$

$$\mathrm{D}_{\mathrm{MM}}^{(q)}\big(m, GF(q^n)\big) = O(l \log m + \epsilon^{-1}\log n),$$

where l is a natural parameter, ϵ is a positive parameter, and c is a certain constant.

The use of multiple multiplications rests on the following result [65,125]: let $m = \lceil \sqrt[r]{n} \rceil$, $r \in \mathbb{N}$. Then raising to the power $(q^n - q)/(q-1)$ in $GF(q^n)$ can be implemented by a circuit with complexity and depth

$$(2r-1)\Big(m\mathrm{F}\big(GF(q^n)\big) + \mathrm{MM}\big(m, GF(q^n)\big)\Big) + (r-1)\mathrm{M}\big(GF(q^n)\big),$$

$$2\Big(\mathrm{D}_{\mathrm{F}}\big(GF(q^n)\big) + \mathrm{D}_{\mathrm{MM}}\big(m, GF(q^n)\big)\Big) + \mathrm{D}_{\mathrm{M}}\big(GF(q^n)\big)$$
$$+ (r-2)\max\big\{\mathrm{D}_{\mathrm{F}}\big(GF(q^n)\big) + \mathrm{D}_{\mathrm{MM}}\big(m, GF(q^n)\big), \mathrm{D}_{\mathrm{M}}\big(GF(q^n)\big)\big\},$$

respectively. As before, two more operations are required to finish inversion. Finally, for any $r \in \mathbb{N}$ a standard basis invertor with the following complexity and depth estimates can be constructed:

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) = O\big(rn^{1/r}(n^w + n^{1,5}\log n \log\log n)\big), \quad \mathrm{D}_{\mathrm{I}}^{(q)}\big(GF(q^n)\big) = O(r\log n),$$

where w is somewhat smaller than 1.667. One can set r to be large enough to obtain a logarithmic depth circuit of complexity $O(n^{1.667})$.

Better bounds in both standard and normal cases may be obtained if the transition between the bases is performed fast. Denote by $\mathrm{T}\big(GF(q^n)\big)$ and $\mathrm{D}_{\mathrm{T}}\big(GF(q^n)\big)$ the complexity and depth of a transition circuit (bilateral transition is considered). Then, by exploiting the idea that multiplication is faster in standard bases and Frobenius operation is faster in normal bases, the following bounds for the inversion in either of the bases could be obtained [65]:

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) = O(R^b n^{1+2/R}) + O(R\sqrt[R]{n})\mathrm{T}^{(q)}\big(GF(q^n)\big),$$
$$\mathrm{D}_{\mathrm{I}}^{(q)}\big(GF(q^n)\big) = O\Big(R\Big(\log n + \mathrm{D}_{\mathrm{T}}^{(q)}\big(GF(q^n)\big)\Big)\Big),$$

where $b < 2.12$ and R is a natural parameter which is either constant or some very slowly growing function with respect to n. Therefore, if a transition circuit of almost linear complexity and logarithmic depth exists, then a logarithmic depth invertor of almost linear complexity can be constructed.

For instance, an invertor in type k Gauss normal basis of the field $GF(q^n)$ of complexity $O(\epsilon^{-b}n^{1+\epsilon})$ and depth $O(\epsilon^{-1}\log n)$, where $\epsilon > 0$, can be constructed under the condition $k = o(\log n)$.

4. Arithmetic in Composite Fields

All the above logarithmic depth circuits outdo addition chain circuits only when n is large enough ($n > 500$) but their complexity in that case (even when $n \approx 100$) is too high for applications. That is why for n of order of several hundreds various versions of the addition chain method are used (the Itoh–Tsujii method as usual). Some depth reduction is possible for composite degree fields of characteristic 2 if not standard bases but bases evolved from field tower representation are used. It is essential that the complexity also decreases. It seems that the idea of applying composite fields to minimize depth first appeared in [82]. Multiple approaches for arithmetic implementation in composite fields were proposed in [1, 78, 107, 111, 113, 114, 120].

Combining [28, 126], one can prove that if n and m are coprime, then for some normal basis

$$\mathrm{M}^{(q)}\big(GF(q^{nm})\big) = O\big(nm(m^{0.806} + n^{0.806})\big).$$

In particular, if $n = \Omega(m)$, then

$$\mathrm{M}^{(q)}\big(GF(q^N)\big) = O(N^{1.403}), \quad N = nm.$$

If N is an ϵ-smooth number, i.e., $N = n_1 \cdots n_m$, all n_i are coprime, $n_1 + \cdots + n_m = O(N^\epsilon)$, then $\mathrm{M}^{(q)}\big(GF(q^N)\big) = O(N^{1+0.806\epsilon})$. But the depth of this circuit is prohibitively high.

4.1. Multiplication and Inversion in Towers of Fields. *Tower* is a consequence of fields embedded one into another. In [1], the authors considered a general construction of a field tower, and for multiplication complexity in the corresponding fields $GF(2^n)$ they established the estimate $O(n\log^2 n)$. So the best possible estimate of straightforward DFT implementation way has to be improved. Obviously, the authors [1] did not know about work [122], where a better result was obtained.

Towers in [1] look like

$$GF(q) \subset K_1 \subset \cdots \subset K_h, \quad K_j = GF(q^{P_1 \dots P_j}), \quad j = 1, \dots, h,$$

where each prime factor of P_j is either a factor of $q-1$ or p, which is the characteristic of the field. Moreover, P_j is an even number if $q = 1 \bmod 4$ only. On any floor of such a tower one can choose a basis with the minimal polynomial being binomial (similar towers were considered independently in [8] and got the name optimal tower fields). To implement multiplication on each floor, Toom's method [134] and the FFT method (in the latter case primitive roots belong to the previous floors) were used. However, fundamental formulas [1, p. 227] are questionable for the reason that they rely on the fact that multiplication of an element from K_j by an element of K_{j-2} has the same order of complexity as addition in K_j (indeed, each of P_jP_{j-1} coordinates is to be multiplied by the element of K_{j-2}, so the complexity of multiplication is to be estimated as $P_jP_{j-1}M(K_{j-2})$). In the particular case $P_j = p^j$, where $p \mid q-1$, $n = P_h$, in [1] the following estimates were proved:

$$\mathrm{M}^{(q)}\big(GF(q^n)\big) = O(n^{1+1/\log p}), \quad \mathrm{I}^{(q)}\big(GF(q^n)\big) = O(n^{1+1/\log p}).$$

Estimates of multiplication complexity are worse than for Schönhage's standard basis multipliers, but the invertor has the same order of complexity and depth not as large as one based on the fast Euclidean algorithm.

In [37], it was proved that for any $\epsilon > 0$ and any natural $m > 1$ one may choose a basis in the field $GF(2^n)$, $n = m^s$, $s \geq s_\epsilon$, in such a way that

$$\mathrm{M}\big(GF(2^n)\big) < n^{1+\epsilon/2}, \quad \mathrm{I}\big(GF(2^n)\big) < n^{1+\epsilon}.$$

In particular, the following asymptotic complexity bounds for $n = 8 \cdot 3^k$ were obtained:

$$\mathrm{I}\big(GF(2^n)\big) = O\big(n^{\log_3 5}\big), \quad \mathrm{M}\big(GF(2^n)\big) = O\big(n^{\log_3 5}\big).$$

Further, for $n = 2 \cdot 3^k$ the following bounds were established:

$$\mathrm{M}\big(GF(2^n)\big) < n(\log_3 n)^{(\log_2 \log_3 n)/2 + O(1)}, \quad \mathrm{I}\big(GF(2^n)\big) < n(\log_3 n)^{(\log_2 \log_3 n)/2 + O(1)}.$$

All the above statements from [37] may be improved at the expense of more accurate estimation of the complexity of multiplication by constants implied in the FFT algorithm. More exactly, with a suitable choice of a basis multiplier and invertor in $GF(2^n)$, $n = m^s$, $s \geq s_m$, the following complexity and depth bounds may be constructed:

$$\mathrm{M}\big(GF(2^n)\big) = O_m(n \log n \log\log n), \quad \mathrm{I}\big(GF(2^n)\big) = O_m\Big(\mathrm{M}\big(GF(2^n)\big)\Big)$$
$$\mathrm{D}_{\mathrm{M}}\big(GF(2^n)\big) = O_m(\log n), \quad \mathrm{D}_{\mathrm{I}}\big(GF(2^n)\big) = O_m(\log^2 n).$$

Sometimes the above bounds may be pointed in a more precise form [66, 67]. For instance, if $m = p$ is a prime, 2 is a primitive root modulo p (this is exactly the condition of the existence of ONB of type I in $GF(2^{p-1})$), and $p^2 \nmid 2^{p-1} - 1$ (it is known that the last condition is fair for $p < 10^{12}$), then for some basis in $GF(2^n)$, $n = (p-1)p^s$, the equalities

$$\mathrm{M}\big(GF(2^n)\big) = O(n \log n \log\log n), \quad \mathrm{I}\big(GF(2^n)\big) = O_p\Big(\mathrm{M}\big(GF(2^n)\big)\Big)$$

are valid. In particular, for $p = 3$

$$\mathrm{M}\big(GF(2^n)\big) = 5n \log_3 n \log_2 \log_3 n + O(n \log n),$$
$$\mathrm{I}\big(GF(2^n)\big) \lesssim \frac{5}{2}\mathrm{M}\big(GF(2^n)\big), \quad \mathrm{D}_{\mathrm{M}}\big(GF(2^n)\big) \lesssim \frac{6}{\log_2 3}\log_2 n.$$

The corresponding multiplication method is, in fact, a modification of Schönhage's algorithm [122] for multiplication of binary polynomials.

For the tower of fields $GF(2^n)$, $n = 2^k$, multiplier and invertor of complexity

$$\mathrm{M}\big(GF(2^n)\big) = O(n^{1.58}), \quad \mathrm{I}\big(GF(2^n)\big) = O(n^{1.58})$$

were constructed in [112]. The method implies, for example,

$$\mathrm{M}\big(GF(2^{1024})\big) \leq 357\,992, \quad \mathrm{I}\big(GF(2^{1024})\big) \leq 538\,033.$$

But the depth of the invertor is $\Omega(\log^3 n)$.

Consider ONB $\{\xi, \xi^2, \xi^4, \xi^8\}$ in $GF(2^4)$ and select in each floor the element $\alpha_k \in GF(2^{2^{k+2}})$ such that

$$\alpha_k^2 + \alpha_k = \xi\alpha_1 \cdots \alpha_{k-1}$$

(similar bases for $k = 1$ were considered in [39] in order to implement AES S-boxes). Then consider in the floor the standard basis $\{1, \alpha_k\}$ or the normal basis $\{\alpha_k, \alpha_k^{2^{k+1}}\}$. With the use of the above construction, asymptotically worse bounds

$$\mathrm{M}\big(GF(2^n)\big) = O(n^{\log 3} \log n), \quad \mathrm{I}\big(GF(2^n)\big) = O(n^{\log 3} \log n), \quad \mathrm{D_I}\big(GF(2^n)\big) = O(\log^3 n),$$

may be established. However, S. Zikrin proved that for $n \le 64$ one can construct better multipliers and invertors than in [111]. For example, he obtained the following estimates:

$$\mathrm{M}\big(GF(2^{16})\big) \le 382, \quad \mathrm{D_M}\big(GF(2^{16})\big) \le 11,$$
$$\mathrm{I}\big(GF(2^{16})\big) \le 479, \quad \mathrm{D_I}\big(GF(2^{16})\big) \le 26,$$

$$\mathrm{M}\big(GF(2^{32})\big) \le 1233, \quad \mathrm{D_M}\big(GF(2^{32})\big) \le 13,$$
$$\mathrm{I}\big(GF(2^{32})\big) \le 1714, \quad \mathrm{D_I}\big(GF(2^{32})\big) \le 48,$$

$$\mathrm{M}\big(GF(2^{64})\big) \le 3943, \quad \mathrm{D_M}\big(GF(2^{64})\big) \le 18,$$
$$\mathrm{I}\big(GF(2^{64})\big) \le 5609, \quad \mathrm{D_I}\big(GF(2^{64})\big) \le 75,$$

$$\mathrm{M}\big(GF(2^{128})\big) \le 12\,728, \quad \mathrm{D_M}\big(GF(2^{128})\big) \le 24,$$
$$\mathrm{I}\big(GF(2^{128})\big) \le 18\,587, \quad \mathrm{D_I}\big(GF(2^{128})\big) \le 114.$$

One may compare these estimates with those from [111]:

$$\mathrm{M}\big(GF(2^{128})\big) \le 12\,476, \quad \mathrm{D_M}\big(GF(2^{128})\big) \le 25,$$
$$\mathrm{I}\big(GF(2^{128})\big) \le 18\,316, \quad \mathrm{D_I}\big(GF(2^{128})\big) \le 170.$$

Note that for a standard basis with the irreducible polynomial $x^{128} + x^7 + x^2 + x + 1$ in $GF(2^{128})$ the following estimates hold:

$$\mathrm{M}\big(GF(128)\big) \le 33\,042, \quad \mathrm{D_M}\big(GF(128)\big) \le 11.$$

The complexity and depth of Karatsuba's multiplier in this case are estimated as 12 343 and 18, respectively. However, the estimates for an invertor look like 200 000 and not less than 200.

4.2. Minimization of Inversion Depth in Composite Fields. In this section, we observe some recursive methods [35, 63] aimed at constructing depth-efficient invertors in composite binary fields for values of n not more than several hundreds.

Suppose that n is odd, $\mathrm{D_M}\big(GF(2^n)\big) \ge \mathrm{D_S}\big(GF(2^n)\big) + 1$, where $\mathrm{S}\big(GF(2^n)\big)$ is the complexity of squaring in $GF(2^n)$. Applying a method from [107], one can construct invertor and multiplier with the following recursive bounds on the complexity and the depth:

$$\mathrm{M}\big(GF(2^{2n})\big) \le 3\mathrm{M}\big(GF(2^n)\big) + 4n, \quad \mathrm{D_M}\big(GF(2^{2n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 2,$$
$$\mathrm{I}\big(GF(2^{2n})\big) \le \mathrm{I}\big(GF(2^n)\big) + 3\mathrm{M}\big(GF(2^n)\big) + \mathrm{S}\big(GF(2^n)\big) + 2n,$$
$$\mathrm{D_I}\big(GF(2^{2n})\big) \le \mathrm{D_I}\big(GF(2^n)\big) + 2\mathrm{D_M}\big(GF(2^n)\big) + 1.$$

Suppose that $(n, 3) = 1$, $B_2 = \{\alpha, \alpha^2, \alpha^4\}$ is the ONB in $GF(2^3)$, where $\alpha^3 = \alpha^2 + 1$, and B_1 is any basis in $GF(2^n)$, $\mathrm{D_M}\big(GF(2^n)\big) \ge \mathrm{D_S}\big(GF(2^n)\big) + 2$. Then for multiplication in $B = B_1 \otimes B_2$ we have

$$\mathrm{M}\big(GF(2^{3n}\big) \le 6\mathrm{M}\big(GF(2^n)\big) + 12n, \quad \mathrm{D_M}\big(GF(2^{3n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 3.$$

Further, for inversion in B we have the following recursions:

$$\mathrm{I}\big(GF(2^{3n})\big) \le \mathrm{I}\big(GF(2^n)\big) + 9\mathrm{M}\big(GF(2^n)\big) + 3\mathrm{S}\big(GF(2^n)\big) + 8n,$$
$$\mathrm{D_I}\big(GF(2^{3n})\big) \le \mathrm{D_I}\big(GF(2^n)\big) + 3\mathrm{D_M}\big(GF(2^n)\big) + 1.$$

If B_1 is a normal basis, then $\mathrm{S}\big(GF(2^n)\big) = 0$.

If in the field tower $GF\Big(\big((2^n)^2\big)^2\Big)$ the ONB $\{\alpha_1, \alpha_1^2\}$ and the standard basis $\{1, \alpha_2\}$, where $\alpha_1^2+\alpha_1 = 1$, $\alpha_2^2 + \alpha_2 = \alpha_1$, are chosen, then for the complexity and the depth of a multiplier the following relations hold:

$$\mathrm{M}\big(GF(2^{4n})\big) \le 9\mathrm{M}\big(GF(2^n)\big) + 20n, \quad \mathrm{D_M}\big(GF(2^{4n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 4.$$

If we choose a normal basis in $GF(2^n)$, then one can construct an invertor with the following recursive relations for complexity and depth:

$$\mathrm{I}\big(GF(2^{4n})\big) \le 14\mathrm{M}\big(GF(2^n)\big) + 14n + \mathrm{I}\big(GF(2^n)\big),$$
$$\mathrm{D_I}\big(GF(2^{4n})\big) \le 3\mathrm{D_M}\big(GF(2^n)\big) + 2 + \max\big\{\mathrm{D_I}\big(GF(2^n)\big), 2\big\}.$$

Suppose that $(n, 5) = 1$, $B_2 = \{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}\}$, where $\alpha^5 = \alpha^4+\alpha^2+\alpha+1$, B_1 is any normal basis in $GF(2^n)$, and $B = B_1 \otimes B_2$. Then for the multiplication in the basis B the following relations hold:

$$\mathrm{M}\big(GF(2^{5n})\big) \le 15\mathrm{M}\big(GF(2^n)\big) + 40n, \quad \mathrm{D_M}\big(GF(2^{5n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 4,$$

and for inversion we have

$$\mathrm{I}\big(GF(2^{5n})\big) \le \mathrm{I}\big(GF(2^n)\big) + 91\mathrm{M}\big(GF(2^n)\big) + 117n,$$
$$\mathrm{D_I}\big(GF(2^{5n})\big) \le \mathrm{D_I}\big(GF(2^n)\big) + 3\mathrm{D_M}\big(GF(2^n)\big) + 1 + \max\big\{\mathrm{D_M}\big(GF(2^n)\big), 6\big\}.$$

The field $GF(2^{6n})$ can be represented as an extension of $GF(2^n)$ of degree 6. We choose in $GF(2^6)$ an ONB $B_2 = \{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}\}$, where $\alpha^6 = \alpha^5 + \alpha^4 + \alpha + 1$. Also we choose in $GF(2^n)$ an arbitrary basis B_1 and consider the basis $B = B_1 \otimes B_2$ in $GF(2^{6n})$. Suppose that $\mathrm{D_M}\big(GF(2^n)\big) \ge \mathrm{D_S}\big(GF(2^n)\big) + 2$. Then for multiplication and inversion in B one has

$$\mathrm{M}\big(GF(2^{6n})\big) \le 21\mathrm{M}\big(GF(2^n)\big) + 60n, \quad \mathrm{D_M}\big(GF(2^{6n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 4,$$
$$\mathrm{I}\big(GF(2^{6n})\big) \le \mathrm{I}\big(GF(2^n)\big) + 42\mathrm{M}\big(GF(2^n)\big) + 5\mathrm{S}\big(GF(2^n)\big) + 65n,$$
$$\mathrm{D_I}\big(GF(2^{6n})\big) = 4\mathrm{D_M}\big(GF(2^n)\big) + 4 + \max\big\{\mathrm{D_I}\big(GF(2^n)\big), 4\big\}.$$

Suppose that $(n, 2) = 1$, $B_1 = \{\alpha_1, \alpha_1^2\} \otimes \{1, \alpha_2\}$, where $\alpha_1^2 + \alpha_1 = 1$, $\alpha_2^2 + \alpha_2 = \alpha_1$ and $B_2 = B_1 \otimes \{1, \alpha_3\}$, where $\alpha_3^2 + \alpha_3 = \alpha_1\alpha_2$, and B is an arbitrary basis in $GF(2^n)$. Then for the basis $B_2 \otimes B$ in $GF(2^{8n})$ the following relations hold:

$$\mathrm{M}\big(GF(2^{8n})\big) \le 27\mathrm{M}\big(GF(2^n)\big) + 80n, \quad \mathrm{D_M}\big(GF(2^{8n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 7.$$

If B is the normal basis, then

$$\mathrm{I}\big(GF(2^{8n})\big) \le \mathrm{I}\big(GF(2^n)\big) + 45\mathrm{M}\big(GF(2^n)\big) + 101n,$$
$$\mathrm{D_I}\big(GF(2^{8n})\big) \le 4\mathrm{D_M}\big(GF(2^n)\big) + 8 + \max\big\{\mathrm{D_I}\big(GF(2^n)\big), 6\big\}.$$

If we choose in $GF(2^4)$ an ONB $B_1 = \{\alpha, \alpha^2, \alpha^4, \alpha^8\}$, where $\alpha^4 = \alpha^3 + \alpha^2 + \alpha + 1$, then in $GF(2^8)$ there exists a basis $B_2 = B_1 \otimes \{1, \beta\}$ such that $\beta^2 + \beta = \alpha$. One can choose in $GF(2^n)$ a normal basis B and consider the basis $B_2 \otimes B$ in $GF(2^{8n})$. For the chosen basis in $GF(2^{8n})$ the following bounds for the complexity and the depth are valid:

$$\mathrm{M}\big(GF(2^{8n})\big) \le 30\mathrm{M}\big(GF(2^n)\big) + 82n, \quad \mathrm{D_M}\big(GF(2^{8n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 5,$$
$$\mathrm{I}\big(GF(2^{8n})\big) \le \mathrm{I}\big(GF(2^n)\big) + 52\mathrm{M}\big(GF(2^n)\big) + 88n,$$
$$\mathrm{D_I}\big(GF(2^{8n})\big) \le 4\mathrm{D_M}\big(GF(2^n)\big) + 6 + \max\big\{\mathrm{D_I}\big(GF(2^n)\big), 2\big\}.$$

Let $(n, 30) = 1$. Then in $GF(2^{30n})$ a normal basis can be chosen and multiplier and invertor can be constructed to prove the relations

$$\mathrm{M}\big(GF(2^{30n})\big) \le 315\mathrm{M}\big(GF(2^n)\big) + 1140n,$$
$$\mathrm{D_M}\big(GF(2^{30n})\big) \le \mathrm{D_M}\big(GF(2^n)\big) + 8,$$

Table 1

n	$\mathrm{I}\big(GF(2^n)\big)$	$\mathrm{D_I}\big(GF(2^n)\big)$
10	220	14
12	293	16
15	590	20
16	499	23
20	905	21
24	1 162	24
30	1 925	29
36	4 438	30
40	3 355	30
120	36 230	54
210	88 000	67
330	171 009	71
690	712 655	101

$$\mathrm{I}\big(GF(2^{30n})\big) \le \mathrm{I}\big(GF(2^n)\big) + 566\mathrm{M}\big(GF(2^n)\big) + 1537n,$$
$$\mathrm{D_I}\big(GF(2^{30n})\big) \le 6\mathrm{D_M}\big(GF(2^n)\big) + 17 + \max\big\{\mathrm{D_I}\big(GF(2^n)\big) + \max\big\{\mathrm{D_M}\big(GF(2^n)\big), 6\big\}, \mathrm{D_M}\big(GF(2^n)\big) + 8\big\}.$$

Table 1 shows bounds on the depth and complexity of inversion in certain fields of characteristic 2 obtained by the above methods.

5. Arithmetic in Pseudo-Mersenne Fields

A prime number q of the form $2^n \pm c$, where c is small, is called a pseudo-Mersenne prime number. Mersenne fields were mentioned above. Several techniques for implementing multiplication in pseudo-Mersenne fields $GF(q^n)$, $n = 2^k, 3^k$, were proposed in [8, 12], aimed at the application to elliptic and hyperelliptic curve cryptography (see [11]). Special bases (the so-called optimal tower bases [8, 12]) were used. These bases are a special case of the bases considered in [1].

5.1. Multiplication in Optimal Towers of Pseudo-Mersenne Fields. Improving the results of [8], Baktir and Sunar [12] constructed multipliers of complexity

$$\mathrm{M}\big(GF(q^{2^k})\big) \le 3^k\mathrm{M}\big(GF(q)\big) + 5(3^k - 2^k)\mathrm{A}\big(GF(q)\big) + \frac{1}{2}(3^k - 1)\mathrm{M}(\alpha_0, q),$$
$$\mathrm{M}\big(GF(q^{3^k})\big) \le 6^k\mathrm{M}\big(GF(q)\big) + 5(6^k - 3^k)\mathrm{A}\big(GF(q)\big) + \frac{2}{5}(6^k - 1)\mathrm{M}(\alpha_0, q),$$

where $x^2 - \alpha_0$, $x^3 - \alpha_0$ are irreducible binomials over $GF(q)$, $\alpha_0 \in GF(q)$, and $\mathrm{M}(\alpha_0, q)$ is the complexity of multiplication by α_0 in $GF(q)$. As a consequence,

$$\mathrm{M}\big(GF(q^4)\big) \le 9\mathrm{M}\big(GF(q)\big) + 25\mathrm{A}\big(GF(q)\big) + 4\mathrm{M}\big(3, q\big),$$
$$\mathrm{M}\big(GF(q^8)\big) \le 27\mathrm{M}\big(GF(q)\big) + 95\mathrm{A}\big(GF(q)\big) + 13\mathrm{M}(3, q),$$
$$\mathrm{M}\big(GF(q^{32})\big) \le 243\mathrm{M}\big(GF(q)\big) + 1055\mathrm{A}\big(GF(q)\big) + 121\mathrm{M}(3, q).$$

Some effective applications of similar results in hyperelliptic cryptography were noted in [11]. In [13], some improvements were proposed for these circuits based on FFT in the case of the Fermat number $q = 2^{16} + 1$.

Independently related results were obtained in [28], namely for $q = p^n$, $p = 2^{16} + 1$, the following bound was proved:

$$\mathrm{M}\big(GF(q^{2^k})\big) \le 2^{k+1}\mathrm{M}\big(GF(q)\big) + 2^{k+1}(3k+1)\mathrm{A}\big(GF(q)\big) + (3(2^k(k-1)+1) + k + 2)\mathrm{M}(2^s, q).$$

Using convolution modulo $x\big(x^{2^{k+1}} - 1\big)/(x^2-1)$, Burtzev and Gashkov [38] proved that

$$\begin{gathered}\mathrm{M}\big(GF(q^4)\big) \le 7\mathrm{M}\big(GF(q)\big) + 59\mathrm{A}\big(GF(q)\big) + 3\mathrm{M}(3,p),\\ \mathrm{M}\big(GF(q^8)\big) \le 15\mathrm{M}\big(GF(q)\big) + 193\mathrm{A}\big(GF(q)\big) + 7\mathrm{M}(3,p),\\ \mathrm{M}\big(GF(q^{16})\big) \le 31\mathrm{M}\big(GF(q)\big) + 558\mathrm{A}\big(GF(q)\big) + 15\mathrm{M}(3,p),\\ \mathrm{M}\big(GF(q^{32})\big) \le 63\mathrm{M}\big(GF(q)\big) + 1525\mathrm{A}\big(GF(q)\big) + 31\mathrm{M}(3,p).\end{gathered}$$

Construction of the circuit on which the latter bound was achieved rests on the existence of the primitive root $\sqrt{2} = 2^4(2^8-1)$ of order 64 in the field $GF(p)$. As follows from Winograd's theorem (see, for example, [24]), multiplicative constants in the terms involving $\mathrm{M}\big(GF(q)\big)$ in the above estimates are minimal.

For $q = p^n$, $p = 2^{13} - 1$, $n = 2^{k_0} \cdot 3^{k_1} \cdot 5^{k_2} \cdot 7^{k_3} \cdot 13^{k_4}$, where $k_0 = 0, 1$, the following relations were proved in [38]:

$$\begin{gathered}\mathrm{M}\big(GF(q^7)\big) \le 13\mathrm{M}\big(GF(q)\big) + 344\mathrm{A}\big(GF(q)\big) + 6\mathrm{A}\big(GF(p)\big),\\ \mathrm{M}\big(GF(q^{13})\big) \le 26\mathrm{M}\big(GF(q)\big) + 1026\mathrm{A}\big(GF(q)\big) + 12\mathrm{A}\big(GF(p)\big).\end{gathered}$$

Also in [38] analogous bounds were proved for $q = p^n$, $p = 2^{17} - 1$, $n = 2^{k_0} \cdot 3^{k_1} \cdot 5^{k_2} \cdot 17^{k_3}$, where $k_0 = 0, 1$:

$$\begin{gathered}\mathrm{M}\big(GF(q^9)\big) \le 17\mathrm{M}\big(GF(q)\big) + 578\mathrm{A}\big(GF(q)\big) + 6\mathrm{A}\big(GF(p)\big),\\ \mathrm{M}\big(GF(q^{18})\big) \le 35\mathrm{M}\big(GF(q)\big) + 1825\mathrm{A}\big(GF(q)\big) + 17\mathrm{A}\big(GF(p)\big).\end{gathered}$$

These results rely on using FFT modulo a Mersenne prime p corresponding to primitive roots ± 2 of order p or $2p$. In the last case, FFT is performed by the Good–Thomas method (see [24]). Multiplication in $GF(q^n)$ was implemented using three FFT's and reduction modulo an irreducible binomial.

The method proposed in [14] requires two FFT's on average when batch calculation of sufficiently many multiplications in $GF(q^n)$ is performed. This method was called modular multiplication in the frequency domain since all the operations are performed over Fourier-images of input data. For modular multiplication, the Montgomery method was used. For example, if the binomial $x^n - 2$ is irreducible over $GF(p)$, $p = 2^m - 1$, $2n - 1 \le m$, then the complexity of modular multiplication in the frequency domain is

$$m\mathrm{M}\big(GF(p)\big) + (m-1)\mathrm{M}\left(\frac{1}{m}, p\right) + \big(6m^2 - 7m + O(1)\big)\mathrm{A}\big(GF(p)\big).$$

In the case where $2n - 1 < 2m$, the complexity bound

$$2m\mathrm{M}\big(GF(p)\big) + (m-1)\mathrm{M}\left(\frac{1}{m}, p\right) + \big(4m^2 - 4m + O(1)\big)\mathrm{A}\big(GF(p)\big)$$

was obtained. Effective application of these results in elliptic curve cryptography was demonstrated in [10].

We remark that, instead of Montgomery multiplication, the use of usual modular multiplication is possible. It implies some simplifications, but the corresponding algorithm for modular multiplication in the frequency domain requires roughly $2m^2$ multiplications by 2^k more than the algorithm from [14]. This is important for software implementation but it is not as important for the construction of a circuit.

6. Multiplication in Fields of Small Characteristic

In recent years, numerous papers on the so-called pairing-based cryptography were published (see, e.g., [29]). A problem of primary practical importance in this research direction is the efficient implementation of pairings. In [18], an efficient algorithm was proposed for Tate pairing in some supersingular curves over fields of characteristic 3. The performance of this algorithm depends on the efficient implementation of arithmetic in $GF(3^n)$. Various approaches to this problem were developed in [23, 75, 88, 115].

In [51, 52], a fast algorithm was presented for Tate pairing on the hyperelliptic curve $y^2 = x^p - x + d$, $d = \pm 1$, over the field $GF(p^n)$. In the case $p = 3$, this algorithm is more efficient than that of the paper [18]. In [94], some improvements of the Duursma–Lee (DL) algorithm for binary fields were suggested. In fact, similar improvements are possible in the general case (see, e.g., [29]). Another improvement of the DL algorithm was suggested in [17].

To implement the DL algorithm for the general case, one needs a circuit for arithmetic in $GF(p^{2pn})$, $(2p, n) = 1$, $p = 4k + 3$.

For this purpose one can use a multiplier with complexity estimate

$$\mathrm{M}\big(GF(p^{2pn})\big) \le (6p-3)\mathrm{M}\big(GF(p^n)\big) + O\Big(p^2 n M\big(GF(p)\big)\Big).$$

The smallest field of some interest is the field $GF(7^{14n})$, which corresponds to the case $p = 7$. Efficient implementation of arithmetic in this field leads to improvements in the method of [95]. The following complexity and depth estimates for multiplication in $GF(7^{14n})$ were proved in [36]:

$$\mathrm{M}\big(GF(7^{14n})\big) \le 13\mathrm{M}\big(GF(7^{2n})\big) + 258n\mathrm{A}\big(GF(7)\big),$$
$$\mathrm{D_M}\big(GF(7^{14n})\big) \le 11\mathrm{D_A}\big(GF(7)\big) + \mathrm{D_M}\big(GF(7^{2n})\big).$$

In particular,

$$\mathrm{M}\big(GF(7^{14\cdot 31})\big) \le 698\,554.$$

This work was supported by the Russian Foundation for Basic Research (grants Nos. 11-01-00508 and 11-01-00792), OMN RAN "Algebraic and combinatorial methods of mathematical cybernetics" program (project "Problems of optimal synthesis of control systems").

REFERENCES

1. V. B. Afanassiev and A. A. Davydov, "Finite field tower: iterated presentation and complexity of arithmetic," *Finite Fields Appl.*, **8**, 216–232 (2002).
2. G. B. Agnew, T. Beth, R. C. Mullin, and S. A. Vanstone, "Arithmetic operations in $GF(2^m)$," *J. Cryptol.*, **6**, 3–13 (1993).
3. G. B. Agnew, R. C. Mullin, I. M. Onyszchuk, and S. A. Vanstone, "An implementation for a fast public-key cryptosystem," *J. Cryptol.*, **3**, 63–79 (1991).
4. O. Ahmadi and A. Menezes, *Irreducible Polynomials of Maximum Weight*, Preprint (2005).
5. A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading (1974).
6. D. W. Ash, I. F. Blake, and S. A. Vanstone, "Low complexity normal bases," *Discrete Appl. Math.*, **25**, 191–210 (1989).
7. A. Avizienis, "Signed-digit number representation for fast parallel arithmetic," *IEEE Trans. Electron. Comput.*, **10**, 389–400 (1961).
8. D. V. Bailey and C. Paar, "Efficient arithmetic in finite fields extensions with application in elliptic curve cryptography," *J. Cryptol.*, **14**, 153–176 (2001).
9. J.-C. Bajard, L. Imbert, and T. Plantard, "Modular number systems: Beyond the Mersenne family," in: *SAC'04. 11th Int. Workshop on Selected Areas in Cryptography* (2004), pp. 159–169.
10. S. Baktir, S. Kumar, C. Paar, and B. Sunar, "A state-of-the-art elliptic curve cryptographic processor operating in the frequency domain," *Mobile Networks Appl.*, **12**, No. 4, 259–270 (2007).

11. S. Baktir, J. Pelzl, T. Wollinger, B. Sunar, and C. Paar, “Optimal tower fields for hyperelliptic curve cryptosystems,” in: *Proc. IEEE 38th ACSSC* (2004).
12. S. Baktir and B. Sunar, “Optimal tower fields,” *IEEE Trans. Comput.*, **53**, No. 10, 1231–1243 (2004).
13. S. Baktir and B. Sunar, “Achieving efficient polynomial multiplication in Fermat fields using fast Fourier transform,” in: *Proc. ACMSE'06*, ACM Press (2006), pp. 549–554.
14. S. Baktir and B. Sunar, “Frequency domain finite field arithmetic for elliptic curve cryptography,” in: *Proc. ISCIS 2006*, Lect. Notes Comput. Sci., Vol. 4263, Springer, Berlin (2006), pp. 991–1001.
15. S. Ballet, J. Chaumine, J. Pieltant, and R. Rolland, *On the Tensor Rank of Multiplication in Finite Extensions of Finite Fields*, `arXiv:1107.1184` (2011).
16. P. D. Barret, “Implementing the Rivest, Shamir and Adleman public key encryption algorithm on a standard digital signal processor,” in: *Advances in Cryptology, Proc. Crypto-86*, Lect. Notes Comput. Sci., Vol. 263, Springer, Berlin (1987), pp. 311–323.
17. P. S. M. L. Barreto, S. Galbraith, C. O'Eigeartaigh, and M. Scott, *Efficient Pairing Computation on Supersingular Abelian Varieties*, Cryptology ePrint Archive, Report 2004/375, `http://eprint.iacr.org/2004/375`.
18. P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, “Efficient algorithms for pairing-based cryptosystems,” in: *Proc. Crypto-2002*, Lect. Notes Comput. Sci., Vol. 2442, Springer, Berlin (2002), pp. 354–368.
19. P. Beame, S. Cook, and H. Hoover, “Log depth circuits for division and related problems,” *SIAM J. Comput.*, **15**, No. 4, 994–1003 (1986).
20. D. J. Bernstein, *Multidigit Multiplication for Mathematicians*, `http://cr.yp.to/papers.html#m3` (2004).
21. D. J. Bernstein, “Batch binary Edwards,” in: S. Halevi, ed., *Advances in Cryptology—CRYPTO 2009. 29th Annual Int. Cryptology Conf., Santa Barbara, CA, USA, August 16–20, 2009. Proceedings*, Lect. Notes Comput. Sci., Vol. 5677, Springer, Berlin (2009), pp. 317–336.
22. D. J. Bernstein and T. Lange, “Type-II optimal polynomial bases,” in: *Arithmetic of Finite Fields. Third International Workshop, WAIFI 2010, Istanbul, Turkey, June 27–30, 2010. Proceedings*, Lect. Notes Comput. Sci., Vol. 6087, Springer, Berlin (2010), pp. 41–61.
23. G. Bertoni, J. Guajardo, S. Kumar, G. Orlando, C. Paar, and T. Wolinger, “Efficient $GF(p^m)$ arithmetic architectures for cryptographic applications,” in: *CT-RSA'03 Proc. of the 2003 RSA Conf. on the Cryptographers' Track*, Lect. Notes Comput. Sci., Vol. 2612, Springer, Berlin (2003), pp. 158–175.
24. R. E. Blahut, *Fast Algorithms for Digital Signal Processing*, Addison-Wesley, Reading (1985).
25. I. Blake, R. Roth, and G. Seroussi, *Efficient Arithmetic in $GF(2^n)$ through Palindromic Representation*, Hewlett-Packard, HPL-98-134 (1998).
26. I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, Cambridge Univ. Press, Cambridge (1999).
27. I. Blake, G. Seroussi, and N. Smart, *Advances in Elliptic Curve Cryptography*, Cambridge Univ. Press, Cambridge (2005).
28. A. A. Bolotov and S. B. Gashkov, “Fast multiplication in normal bases of finite fields,” *Discrete Math. Appl.*, **11**, No. 4, 327–356 (2001).
29. A. A. Bolotov, S. B. Gashkov, and A. B. Frolov, *Elementary Introduction in Elliptic Cryptography*: *Cryptographic Protocols on Elliptic Curves* [in Russian], KomKniga, Moscow (2006).
30. A. A. Bolotov, S. B. Gashkov, A. B. Frolov, and A. A. Chasovskikh, *Elementary Introduction in Elliptic Cryptography*: *Algebraic and Algorithmic Background* [in Russian], KomKniga, Moscow (2006).
31. A. Brauer, “On addition chains,” *Bull. Am. Math. Soc.*, **45**, 736–739 (1939).
32. R. P. Brent, P. Gaudry, E. Thome, and P. Zimmerman, *Faster Multiplication in $GF(2)[x]$*, Preprint INRIA No. 6359 (2007).

33. R. Brent, F. Gustavson, and D. Yun, "Fast solution of Toeplitz systems of equations and computation of Padé approximants," *J. Algorithms*, **1**, 259–295 (1980).
34. R. Brent and H. Kung, "Fast algorithms for manipulating formal power series," *J. ACM*, **25**, No. 4, 581–595 (1978).
35. A. A. Burtzev, "On the circuits for multiplication and inversion in composite fields $GF(2^n)$," *Chebyshevskii Sb.*, **7**, No. 1 (17), 172–185 (2006).
36. A. A. Burtzev, "On the Boolean circuits for multiplication in finite fields of odd characteristics," in: *Proc. VI Sci. School on Discrete Mathematics and Its Applications* (*Moscow, IPM RAN, April 2007*) [in Russian], Vol. I (2007), pp. 13–16.
37. A. A. Burtzev, I. B. Gashkov, and S. B. Gashkov, "On the complexity of Boolean circuits for arithmetic in some towers of finite fields," *Vestn. Mosk. Univ. Ser. 1 Mat. Mekh.*, No. 5, 10–16 (2006).
38. A. A. Burtzev and S. B. Gashkov, "On the circuits for arithmetic in composite fields of large characteristic," *Chebyshevskii Sb.*, **7**, No. 1 (17), 186–204 (2006).
39. D. Canright, *A Very Compact Rijndael S-Box*, Technical Report NPS-MA-04-001, Naval Postgraduate School, `http://library.nps.navy.mil/uhtbin/hyperion-image/NPS-MA-05-001.pdf` (2004).
40. D. Cantor, "On arithmetic algorithms over finite fields," *J. Combin. Theory Ser. A*, **50**, 285–300 (1989).
41. D. Cantor and E. Kaltofen, "On fast multiplication of polynomials over arbitrary algebras," *Acta Inform.*, **28**, 693–701 (1991).
42. K. Chang, H. Kim, J. Kang, and H. Cho, "An extension of TYT algorithm for $GF\big((2^n)^m\big)$ using precomputation," *Inform. Process. Lett.*, **92**, 231–234 (2004).
43. A. V. Chashkin, "Fast multiplication and addition of integer numbers," in: *Discrete Mathematics and Applications* [in Russian], MGU (2001), pp. 91–110.
44. B. Chor and O. Goldreich, "An improved parallel algorithm for integer GCD," *Algorithmica*, **5**, 1–10 (1990).
45. D. V. Chudnovsky and G. V. Chudnovsky, "Algebraic complexities and algebraic curves over finite fields," *J. Complexity*, **4**, 285–316 (1988).
46. S. Cook, *On the Minimum Computation Time of Functions*, Ph.D. Thesis, Harvard Univ. (1966).
47. A. De, P. P. Kurur, C. Saha, and R. Saptharishi, *Fast Integer Multiplication Using Modular Arithmetic*, `arXiv:0801.1416v2` (2008).
48. E. Demenkov, A. Kojevnikov, A. S. Kulikov, and G. Yaroslavtsev, "New upper bounds on the Boolean circuit complexity of symmetric functions," *Inform. Process. Lett.*, **110**, 264–267 (2010).
49. C. Doche, "Redundant trinomials for finite fields of characteristic 2," in: *ACISP 2005*, Lect. Notes Comput. Sci., Vol. 3574, Springer, Berlin (2005), pp. 122–133.
50. P. E. Dunne, *The Complexity of Boolean Metworks*, Academic Press, London (1988).
51. I. Duursma and H.-S. Lee, "Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$," in: *Proc. Asiacrypt-2003*, Lect. Notes Comput. Sci., Vol. 2894, Springer, Berlin (2003), pp. 111–123.
52. I. Duursma and H.-S. Lee, *Tate Pairing Implementation for Tripartite Key Agreement*, Cryptology ePrint Archive, Report 2003/053, `http://eprint.iacr.org/2003/053`.
53. W. Eberly, "Very fast parallel polynomial arithmetic," *SIAM J. Comput.*, **18**, No. 5, 955–976 (1989).
54. S. Erdem, T. Yanik, and C. Koc, "Polynomial basis multiplication over $GF(2^n)$," *Acta Appl. Math.*, **93**, 33–55 (2006).
55. P. Erdős, "Remarks on number theory. III: On addition chains," *Acta Arith.*, **6**, 77–81 (1960).
56. H. Fan and M. A. Hasan, "Alternative to the Karatsuba algorithm for software implementation of $GF(2^n)$ multiplication," *Information Security, IET*, **3**, No. 2, 60–65 (2009).
57. S. Feisel, J. von zur Gathen, and M. A. Shokrollahi, "Normal bases via general Gauss periods," *Math. Comput.*, **68**, No. 225, 271–290 (1999).
58. M. Fürer, "Faster integer multiplication," in: *Proc. 39th ACM STOC 2007 Conf.*, pp. 57–66.

59. S. Gao, J. von zur Gathen, and D. Panario, "Gauss periods and fast exponentiation in finite fields," in: *Proc. Latin'95* (*Valparaiso, Chile*), Lect. Notes Comput. Sci., Vol. 911, Springer, Berlin (1995), pp. 311–322.
60. S. B. Gashkov, "Remarks on fast polynomial multiplication, Fourier and Hartley transforms," *Diskret. Mat.*, **12**, No. 3, 124–153 (2000).
61. S. B. Gashkov and V. N. Chubarikov, *Arithmetic. Algorithms. Complexity of Computation* [in Russian], Nauka, Moscow (1996).
62. S. B. Gashkov, M. I. Grinchuk, and I. S. Sergeev, "On constructing small depth adders," *Diskret. Anal. Issled. Oper. Ser. 1*, **14**, No. 1, 27–44 (2007).
63. S. B. Gashkov and R. A. Khokhlov, "On the depth of logical circuits for operations in fields $GF(2^n)$," *Chebyshevskii Sb.*, **4**, No. 4 (8), 59–71 (2003).
64. S. B. Gashkov and I. S. Sergeev, "An application of the method of addition chains to inversion in finite fields," *Discrete Math. Appl.*, **16**, No. 6, 601–618 (2006).
65. S. B. Gashkov and I. S. Sergeev, "On constructing circuits of logarithmic depth for inversion in finite fields," *Diskret. Mat.*, **20**, No. 4, 8–28 (2008).
66. S. B. Gashkov and I. S. Sergeev, "The complexity and depth of Boolean circuits for multiplication and inversion in some fields $GF(2^n)$," *Moscow Univ. Math. Bull.*, **64**, No. 4, 139–143 (2009).
67. S. B. Gashkov and I. S. Sergeev, "On the complexity and the depth of multiplication and inversion in some polynomial rings," in: *Proc. XI Int. Sem. "Discrete Math. and Its Applications"* (*Moscow, June 2012*) [in Russian].
68. J. von zur Gathen, "Inversion in finite fields," *J. Symbol. Comput.*, **9**, 175–183 (1990).
69. J. von zur Gathen and J. Gerhard, "Arithmetic and factorization of polynomials over $GF(2)$," in: *Proc. ISSAC'96*, Zürich (1996), pp. 1–9.
70. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge Univ. Press, Cambridge (1999).
71. J. von zur Gathen and M. Nöcker, "Exponentiation in finite fields: theory and practice," in: *Applied Algebra. Proc. AAECC-12*, Lect. Notes Comput. Sci., Vol. 1255, Springer, Berlin (1997), pp. 88–113.
72. J. von zur Gathen and M. Nöcker, "Fast arithmetic with general Gauss periods," *Theor. Comput. Sci.*, **315**, 419–452 (2004).
73. J. von zur Gathen, M. A. Shokrollahy, and J. Shokrollahy, "Efficient multiplication using type 2 optimal normal bases," in: C. Carlet and B. Sunar, eds., *Arithmetic of Finite Fields. First International Workshop, WAIFI 2007, Madrid, Spain, June 21–22, 2007. Proceedings*, Lect. Notes Comput. Sci., Vol. 4547, Springer, Berlin (2007), pp. 55–68.
74. P. Gaudry, A. Kruppa, and P. Zimmermann, "A GMP-based implementation of Schönhage–Strassen's large integer multiplication algorithm," in: *ISAAC'07, Waterloo, Ontario, Canada, 2007.*
75. R. Granger, D. Page, and M. Stam, "Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three," *IEEE Trans. Comput.*, **54**, No. 7, 852–860 (2005).
76. M. I. Grinchuk, "Refinement of the upper bound for the depth of adder and comparator," *Diskret. Anal. Issled. Oper. Ser. 1*, **15**, No. 2, 12–22 (2008).
77. E. Grove, *Proofs with Potential*, Ph.D. Thesis, U.C. Berkeley (1993).
78. J. Guajardo, T. Güneysu, S. Kumar, C. Paar, and J. Pelzl, "Efficient hardware implementation of finite fields with application to cryptography," *Acta Appl. Math.*, **93**, 75–118 (2006).
79. D. Hankerson, J. H. López, and A. Menezes, "Software implementation of elliptic curve cryptography over binary fields," in: *CHES 2000*, Lect. Notes Comput. Sci., Vol. 1965, Springer, Berlin (2000), pp. 1–23.
80. J. Hastad and T. Leighton, *Division in $O(\log n)$ Depth Using $O(n^{1+\epsilon})$ Processors*, `http://www.nada.kth.se/~yohanh/paraldivision.ps` (1986).
81. X. Huang and V. Pan, "Fast rectangular matrix multiplication and applications," *J. Complexity*, **14**, 257–299 (1998).

82. T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverses in $GF(2^n)$ using normal bases," *Inform. and Comput.*, **78**, 171–177 (1988).
83. D. Jungnickel, *Finite Fields. Structure and Arithmetic*, Wissenschaftsverlag, Mannheim (1993).
84. E. Kaltofen and V. Shoup, "Subquadratic-time factoring of polynomials over finite fields," *Math. Comput.*, **67**, No. 223, 1179–1197 (1998).
85. A. A. Karatsuba, "Complexity of computations," *Tr. Mat. Inst. Steklova*, **211**, 1–17 (1995).
86. A. A. Karatsuba and Yu. P. Ofman, "Multiplication of multidigit numbers on automata," *Sov. Phys. Dokl.*, **7**, 595–596 (1963).
87. K. Kedlaya and C. Umans, "Fast modular composition in any characteristic," in: *Proc. 49th IEEE Symp. on Foundations of Computer Science* (*FOCS*) (2008), pp. 146–155.
88. T. Kerins, W. P. Marnane, E. M. Popovici, and P. S. L. M. Barreto, "Efficient hardware for Tate pairing calculation in characteristic three," in: *Proc. CHES-2005*, Lect. Notes Comput. Sci., Vol. 3659, Springer, Berlin (2005), p. 412.
89. V. M. Khrapchenko, "Asymptotic estimation of addition time of a parallel adder," *Syst. Theory Res.*, **19**, 105–122 (1970).
90. V. M. Khrapchenko, "Some estimates on the time of multiplication," *Probl. Kibern.*, **33**, 221–227 (1978).
91. V. M. Khrapchenko, "On possibility of refinement of estimation for delay of parallel adder," *Diskret. Anal. Issled. Oper. Ser. 1*, **14**, No. 1, 27–44 (2007).
92. D. Knuth, "The analysis of algorithms," in: *Proc. Int. Congress of Math.* (*Nice, France*), **3**, 269–274 (1970).
93. D. Knuth, *The Art of Computer Programming*, Addison-Wesley, Reading (1998).
94. S. Kwon, *Efficient Tate Pairing Computation for Supersingular Elliptic Curves over Binary Fields*, Cryptology ePrint Archive, Report 2004/303, `http://eprint.iacr.org/2004/303`.
95. E. Lee, H.-S. Lee, and Y. Lee, *Fast Computation of Tate Pairing on General Divisors for Hyperelliptic Curves of Genus* 3, Cryptology ePrint Archive, Report 2006/125, `http://eprint.iacr.org/2006/125`.
96. R. Lidl and H. Niederreiter, *Finite Fields*, Addison-Wesley, Reading (1983).
97. B. E. Litow and G. I. Davida, "$O(\log n)$ parallel time finite field inversion," in: *VLSI Algorithms and Architectures*, Lect. Notes Comput. Sci., Vol. 319, Springer, Berlin (1988), pp. 74–80.
98. O. B. Lupanov, "On rectifier and contact rectifier circuits," *Dokl. Akad. Nauk SSSR*, **111**, No. 6, 1171–1174 (1956).
99. O. B. Lupanov, *Asymptotic Bounds on Complexity of Control Systems* [in Russian], Izd. Mosk. Univ., Moscow (1984).
100. J. L. Massey and J. K. Omura, *Apparatus for Finite Fields Computation*, US Patent 4587627 (1986).
101. E. D. Mastrovito, *VLSI Architectures for Computation in Galois Fields*, Ph.D. Thesis, Linköping Univ. (1991).
102. T. Mateer, *Fast Fourier Algorithms with Applications*, Ph.D. Thesis, Clemson Univ. (2008).
103. J. H. McClellan and C. M. Rader, *Number Theory in Digital Signal Processing*, Prentice-Hall, Englewood Cliffs (1979).
104. A. J. Menezes, P. C. van Oorshot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton (1997).
105. R. Moenck, "Fast computation of GCDs," in: *Proc. 5th Ann. ACM Symp. on Theory of Computing* (1973), pp. 142–151.
106. N. Möller, "On Schönhhage's algorithm and subquadratic integer gcd computation," *Math. Comput.*, **77**, 589–607 (2008).
107. M. Morii and M. Kasahara, "Efficient construction of gate circuit for computing multiplicative inverses in $GF(2^n)$," *Trans. IEICE*, **72**, No. 1, 37–42 (1989).
108. R. C. Mullin, I. M. Onyszchuk, S. A. Vanstone, and R. M. Wilson, "Optimal normal bases in $GF(p^n)$," *Discrete Appl. Math.*, **22**, 149–161 (1988/89).

109. P. Naudin and C. Quitte, *Algorithmique Algébrique*, Masson, Paris (1992).

110. C. Negre and T. Plantard, *Prime Field Multiplication in Adapted Modular System Using Lagrange Representation*, Preprint (2005).

111. C. Paar, *Effective VLSI Architectures for Bit Parallel Computation in Galois Fields*, Ph.D. Thesis, Universität GH Essen (1994).

112. C. Paar and J. L. Fan, *Efficient inversion in tower fields of characteristic two*, ISIT, Ulm (1997).

113. C. Paar, P. Fleischmann, and P. Roelse, "Effective multiplier architectures for Galois fields $GF(2^{4n})$," *IEEE Trans. Comput.*, **47**, No. 2, 162–170 (1998).

114. C. Paar, P. Fleischmann, and P. Soria-Rodriges, "Fast arithmetic for public-key algorithms in Galois fields with composite exponents," *IEEE Trans. Comput.*, **48**, No. 10, 1025–1034 (1999).

115. D. Page and N. P. Smart, "Hardware implementation of finite fields of characteristic three," in: *Proc. CHES-2003*, pp. 529–539.

116. M. Paterson, N. Pippenger, and U. Zwick, "Optimal carry save networks," in: *Boolean Function Complexity*, London Math. Soc. Lect. Note Ser., Vol. 169, Cambridge Univ. Press, Cambridge (1992), pp. 174–201.

117. M. Paterson and U. Zwick, "Shallow circuits and concise formulae for multiple addition and multiplication," *Comput. Complexity*, **3**, 262–291 (1993).

118. N. P. Red'kin, "Minimal realization of a binary adder," *Probl. Kibern.*, **38**, 181–216 (1981).

119. J. Reif and S. Tate, "Optimal size integer division circuits," *SIAM J. Comput.*, **19**, No. 5, 912–925 (1990).

120. A. Reyhani-Masoleh and M. A. Hasan, "On effective normal basis multiplication," in: *Proc. India-CRYPT-2000*, Lect. Notes Comput. Sci., Vol. 1977, Springer, Berlin (2000), pp. 213–224.

121. A. Schönhage, "Schnelle Berechnung von Kettenbruchentwicklungen," *Acta Inform.*, **1**, 139–144 (1971).

122. A. Schönhage, "Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2," *Acta Inform.*, **7**, 395–398 (1977).

123. A. Schönhage and V. Strassen, "Schnelle multiplikation großer Zahlen," *Computing*, **7**, 271–282 (1971).

124. J. E. Seguin, "Low complexity normal bases," *Discrete Appl. Math.*, **28**, 309–312 (1990).

125. I. S. Sergeev, "Circuits of logarithmic depth for inversion in finite fields of characteristic two," *Mat. Vopr. Kibern.*, **15**, 35–64 (2006).

126. I. S. Sergeev, "On constructing circuits for transitions between polynomial and normal bases in finite fields," *Diskret. Mat.*, **19**, No. 3, 89–101 (2007).

127. I. S. Sergeev, "On the depth of circuits for multiple addition and multiplication of numbers," in: *Proc. VI Sci. School on Discrete Math. and Its Appl.* (*Moscow, IPM RAN, April 2007*) [in Russian], Vol. II (2007), pp. 40–45.

128. I. E. Shparlinski, M. A. Tsfasman, and S. G. Vladuts, "Curves with many points and multiplication in finite fields," in: *Coding Theory and Algebraic Geometry*, Lect. Notes Math., Vol. 1518, Springer, Berlin (1992), pp. 145–169.

129. D. Stehlé and P. Zimmermann, "A binary recursive GCD algorithm," in: *Proc. ANTS-VI* (*Burlington, USA, 2004*), Lect. Notes Comput. Sci., Vol. 3076, Springer, Berlin (2004), pp. 411–425.

130. G. K. Stolyarov, *Method for Parallel Multiplication in Digital Computers and Device for Its Implementation*, Author Certificate cl. 42, **14**, No. 126668 (1960).

131. V. Strassen, "Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten," *Numer. Math.* **20**, 238–251 (1973).

132. V. Strassen, "The computational complexity of continued fractions," *SIAM J. Comput.*, **12**, 1–27 (1983).

133. N. Takagi, J. Yoshiki, and K. Takagi, "A fast algorithm for multiplicative inversion in $GF(2^n)$ using normal basis," *IEEE Trans. Comput.*, **50**, No. 5, 394–398 (2005).

134. A. L. Toom, “The complexity of a scheme of functional elements realizing the multiplication of integers,” *Sov. Math. Dokl.*, **3**, 714–716 (1963).
135. C. Umans, “Fast polynomial factorization and modular composition in small characteristic,” in: *Proc. 40th Symp. on Theory of Computing* (*STOC*) (2008), pp. 481–490.
136. O. N. Vasilenko, *Number-Theoretic Algorithms in Cryptoraphy* [in Russian], MCNMO, Moscow (2007).
137. C. S. Wallace, “A suggestion for a fast multiplier,” *IEEE Trans. Electron. Comput.*, **13**, 14–17 (1964).
138. I. Wegener, *The Complexity of Boolean Functions*, Wiley, Stuttgart (1987).
139. A. C. Yao, “On the evaluation of powers,” *SIAM J. Comput.*, **5**, 100–103 (1976).

Sergey B. Gashkov
Moscow State University, 119899 Moscow, Russia
E-mail: sbgashkov@gmail.com

Igor S. Sergeev
Moscow State University, 119899 Moscow, Russia
E-mail: isserg@gmail.com