Complexity of computation in Finite Fields

Sergey B. Gashkov[*], Igor S. Sergeev[†]

Аннотация

Review of some works about the complexity of implementation of arithmetic operations in finite fields by boolean circuits.

Keywords: Galois field, multiplication, inversion, Boolean circuit, depth, complexity.

Introduction

Efficient implementation of arithmetic in finite fields is of primary importance for cryptography, coding theory, digital signal processing etc. (see, for example [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]). In this survey we consider only Boolean circuits for arithmetic operations in finite fields. Another term: 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 two, but some fields of odd characteristic are also involved. In the last 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. *Depth* of a given circuit is the length of the longest directed path, connecting primary input and output of the circuit. *Complexity* of a given circuit (in other words, 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 [11, 12, 13]. Minimization of the depth and the complexity of circuits is one of the central and practically important problems in the complexity theory.

In practice, are often exploited so-called *circuits with memory* (i.e. finite automata). Numerous papers deal with finite fields arithmetic implementation on such circuits. This subject needs in special review and does 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 overriding an information stored on a tape by a reading 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 theoretic interest, it is also omitted in the review.

[0]This work was supported by the Russian Foundation for Basic Research (grants no. 08–01–00863 and 08–01–00632–a), Sci. Sch. (4470.2008.1), OMN RAN "Algebraic and combinatorial methods of mathematical cybernetics" program (project "Synthesis and complexity of control systems").

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

The 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*

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

(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 $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$ are linearily independent over $GF(q)$, then they form a basis

$$B^\alpha = \left\{\alpha^{q^0}, \alpha^{q^1}, \ldots, \alpha^{q^{n-1}}\right\},$$

which is called *normal basis* with generator α. (Theoretical background on finite fields may be found in [5, 14].) Complexity of implementation of multiplication and inversion in $GF(q)$ are denoted by $M(GF(q))$ and $I(GF(q))$ respectively. We also introduce notation $D_M(GF(q))$ and $D_I(GF(q))$ respectively for the depth of the operations.

Similar notation is used for other operations. Sometimes it is convenient to consider calculations over subfield $GF(p)$. For corresponding complexity and depth measures we use the same notation with upper indices (p) like $M^{(p)}(GF(q))$ or $D_I^{(p)}(GF(q))$.

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 some senses) circuits for addition or subtraction is not trivial. Various circuits are described in books and papers on computer arithmetic. We list some theoretic results below.

Complexity of addition (subtraction) of n-bit numbers (corresponding circuits are usually called adders or subtractors) is known to be $A(n) = 5n-3$, due to N.P.

Red'kin [15]. Such circuits are easy to built but the lower bound proof is rather complicated (in [15] 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 [16] allows one to build an adder of depth $\log n + \sqrt{(2 + o(1)) \log n}$ (here and further on "log" denotes binary logarithm); complexity of this circuit can be reduced to $(8 + o(1))n$ [17]. 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 [17] (including ternary method due to M.I. Grinchuk with depth bound $1.262 \log n + 2.05$).

Recently M.I. Grinchuk invented the adder with the depth $\log n + \log \log n + 6$ [18]. This adder is also the best known for small values of n.

V.M. Khrapchenko's [19] at 2007 proved the following lower bound for the depth of an adder (built of AND, OR, NOT cells):

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

1.2 Multiplication

Numeric multiplication is evidently 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 ant theoretic aspects.

Complexity of multiplication of n-bit numbers is denoted by $M(n)$. 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 standard multiplier can be constructed so that its depth reduced to $O(\log n)$ (using a method, proposed independently by G.K. Stolyarov [21], A. Avizienis [23], Yu.P. Ofman [22] and C. Wallace [24]).

Minimization of the depth of a standard multiplier is one of extensively studied problems of computer science. Essential results related to the problem was established by V.M. Khrapchenko [25].

To the best of our knowledge, the best current asymptotic upper bound is $4.44 \log n + O(1)$ (see [26, 27, 28]). More practical method leads to the depth estimate $5 \log n + 5$ [29]. This method also provides a benefit in terms of complexity.

The earliest method of reducing complexity of an integer multiplier is due to A.A. Karatsuba [22] (at that time he was a post-graduated student of Moscow State University (MSU), the problem was set by A.N. Kolmogorov). He made an interesting historical review on fast arithmetic algorithms in [30]. The recursive complexity estimate of Karatsuba's integer multiplier is

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

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

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

[1] The bound given in [22] is $M(n) = O(n^{\log 3})$. Constants in the above and below formulas were obtained by A.A. Burtsev in the degree work.

Karatsuba's multiplier has lower size than a standard one for $n \geq 17$. But its depth is $O(\log^2 n)$. Similarly to the case of standard multiplier, the depth of Karatsuba's multiplier can be reduced to $O(\log n)$ (see e.g. [13]). In [31] 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 $(10+o(1))\log n$ causing further increasing of the multiplicative constant in estimate for complexity [29]).

Asymptotically better multiplier was constructed by A.L. Toom [32] (at that time he was a student of MSU, his scientific adviser was O.B. Lupanov). Constants in Toom's estimate were subsequently refined, S. Cook in his thesis [34] adapted the method to Turing machines, A. Schönhage developed a modular method with similar complexity estimate (reader can find a more detailed review in [33]).

Toom's multiplier was improved by A. Schönhage and V. Strassen [35] (see also [37]). 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 $(9 + o(1))\log n$ can be achieved [29]). It was also claimed in [35] that the same complexity estimate is valid for Turing machine multiplication.

The best known multiplier can be constructed by M. Fürer's method [38] (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

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

In [39] a modular version of Fürer's algorithm was posed.

Evidently the last two multipliers both can not find applications in cryptography, due to large multiplicative constants in estimates. Some ways for speeding up program implementation of Schönhage—Strassen's algorithm were considered in detail in [40].

Pollard's multiplier [1, 41] 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 bounds $30634n \log n + 393n$ for the complexity and $349 \log n + 50$ for the depth of Pollard's circuit was claimed under the restriction $n < 201326604$.

Asymptotic efficiency (and practical inefficiency) of all 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 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

$$M(4n) \leq 7M(n) + 662n + 1085,$$

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

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

[2]Benefit of this construction was confirmed by V.V. Baev in the degree work.

In particular, $M(1024) \leq 1279651$. Karatsuba's method gives worse bound in this case. With the use of technique [31] the depth of Toom's multiplier can be reduced to $O(n \log n)$.

1.3 Division

The "school" division method allows to built a circuit for division $2n$-bit number by n-bit one of complexity $O(n^2)$ and depth $O(n \log n)$. Best of known in computer arithmetic analogous circuits 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 Newton—Raphson method.

The reduction was accomplished in [34] (see also [33, 47]). The complexity of Cook's circuit is asymptotically 5 times greater than of a multiplier and the depth is $O(\log^2 n)$. However if n is small, school division circuits have less complexity and slightly more depth.

The method of [42] 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 [43] circuit for division of depth $O(\log n)$ and complexity $O(n^5)$ was produced. In [44] 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. Complexity of multiplication modulo any natural p is not greater than $3M(\log p)+ O(\log p)$. To get this estimate one can perform usual multiplication, then calculate remainder of division of $\lceil 2 \log p \rceil$-bit product by $\lceil \log p \rceil$-bit number p. The latter operation may be implemented by the so-called Barret's method [45] (see also [6, 47]). It is very likely that this method originates from papers [34, 46]. For some particular modula like $p = 2^n \pm c$, $c = O(\log n)$, the above complexity estimate may be improved to $M(\log p) + O(\log p)$.

Complexity of addition (or subtraction) modulo n-bit number p can be estimated as $2A(n) + O(1)$. For Mersenne primes $p = 2^n - 1$ this bound can be reduced to $A(GF(p)) = 7n - 5$. 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 Fermat prime $p = 2^n + 1$, the complexity in this case is $A(GF(p)) = 9n + O(1)$.

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. 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 $M(C, p) \leq (l(C) - 1)A(GF(p))$. For instance, $M(17, p) \leq A(GF(p))$.

Analogously for multiplication by 2^k in Fermat prime field the following complexity

and depth estimates can be obtained:

$$M(2^k, p) \le 5A(GF(p))/9 + O(1),$$
$$D_M(2^k, p) = (1 + o(1)) \log n \le 2 \log n.$$

In the general case of multiplication by C complexity estimate takes a form $M(C, p) \le (l(C) - 1)A(GF(p)) + (5n + O(1))l(C)$. For instance, $M(3, p) \le 14A(GF(p))/9 + O(1)$.

Estimates $6n^2 - n + O(1)$ and $4.44 \log n + O(1)$ are known for complexity and depth of a standard multiplier modulo 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 $M_{q,f}(n)$ 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 $m_{q,f}(n)$—multiplicative complexity and $a_{q,f}(n)$—additive complexity (i.e. the number of multiplicative and additive operations over $GF(q)$ respectively). Then $M(GF(q^n)) \le M_{q,f}(n)M(GF(q))$ for any irreducible polynomial $f(x)$ over $GF(q)$. To be more precise,

$$M(GF(q^n)) \le m_{q,f}(n)M(GF(q)) + a_{q,f}(n)A(GF(q)).$$

We also use the notation $M_q(n)$ for the complexity over $GF(q)$ of multiplication of polynomials of degree less than n. Analogously $m_q(n)$ and $a_q(n)$ denote multiplicative and additive complexity.

Strassen's method [46] (see also [37]) implies that for any f

$$m_{q,f}(n) \le 3m_q(n), \quad a_{q,f}(n) \le 3a_q + O(n).$$

In [47] the other algorithm with the same complexity estimate was proposed. The algorithm is a polynomial analogue of Barret's algorithm (as well as the Barret's algorithm is its numeric analogue). If $f(x)$ is a sum of k monomials, then $M_{q,f}(n) \le M_q(n) + (2k + 1)n$, and if $q = 2$ then $M_{2,f}(n) \le M_2(n) + kn$. It is well-known hypothesis that one can always choose an irreducible polynomial f with $k \le 5$. Therefore

$$M_{q,f}(n) \le M_q(n)(1 + o(1)).$$

In [48] (see also [37]) is proved that estimates $m_q(n) = O(n \log n)$ and $a_q(n) = O(n \log n \log \log n)$ can be achieved simultaneously. In [49] a multiplicative constant in this estimate was refined. But both methods seem not to be applicable in cryptography or coding theory by the reason of this constant is too large.

It is known (see e.g. [41]) that in the case $2n - 1 \le q$ the multiplicative complexity of multiplication in $GF(q^n)$ is $2n - 1$. The main idea of upper bound was proposed by A.L. Toom [32] and the proof of the lower bound is due to S. Winograd (see e.g. [2]).

It was shown by brothers Chudnovsky [50] that in the general case multiplicative complexity is $O(n)$ as well. The reader can find improved estimates in [51] and in

several papers by Ballet et al., see e.g. [52]. On the other hand 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. Complexity and depth estimates of the "school" method are

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

For $n \approx 1000$ one has $M(n) \approx 2000000$, $D(n) = 11$.

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

$$M(2n) \leq 3M(n) + 7n - 3,$$
$$M(2n+1) \leq 2M(n+1) + M(n) + 7n - 1,$$

implying for $n = 2^k$, $k \geq 3$, the next relations:

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

In particular, for $n = 1024$ we have $M(n) \leq 330725, D(n) \leq 27$.

Using Schönhage's [48] FFT method a circuit for cyclic convolution with complexity $Z(2187) \leq 428351$ and depth $D_Z(2187) \leq 46$, or a circuit with bounds $Z(2187) \leq 430537$ and $D_Z(2187) \leq 34$ can be constructed. As a corollary we have

$$M(1024) \leq M(1093) \leq 430537, \quad D_M(1024) \leq D_M(1093) \leq 34.$$

In this case Karatsuba's multiplier is more efficient.

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

$$Z(2048) \leq 998216, \quad D_Z(2048) \leq 30.$$

In this case FFT method is preferable.

Another example: multiplication modulo $x^{1458} + x^{729} + 1$ can be implemented using FFT method with complexity 273850, and depth 33. In this case Karatsuba's method again plays over.

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

There also exists D. Cantor's method [53] for polynomial multiplication over finite fields. The asymptotic complexity of this method is slightly greater than FFT's (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 [54] a modification of Cantor's method and some applications to polynomial factorization were considered.

The Cantor's method can be viewed as some refinement of Toom's method. For interpolation it exploits as nodes an 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 binomial $x^n - 1$ as a nodes. It is equivalent to using 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 n-th 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 $\Omega(\log n(\log\log n)(\log\log\log n))$. Hence the total complexity of multiplication of n-degree polynomials over $GF(2)$ following the above way can be bounded as

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

This bound is not easy to achieve following this way, before n must divide $2^m - 1$, that prevents it to have a form 2^k, which is convenient to perform DFT fast, and rarely allows to have a form 3^k or 5^k. Nevertheless, sometimes this bound is achievable. Following the method [55], consider $n = 2^{p-1}$, $q = 2^p - 1$ be Mersenne prime. For multiplication of n-degree polynomials over $GF(2)$ it is sufficient to multiply this polynomials as the polynomials with integer coefficients 0,1. The last operation may be performed as the multiplication of polynomials over $GF(q)$ with the use of $2n$-point FFT over $GF(q^2)$. It is known, that $(2^{n/2} + 3^{n/2}i)^2 \in GF(q^2)$ is $2n$-th root of unity, where $i \in GF(q^2)$ is the root of irreducible polynomial $x^2 + 1$ over $GF(q)$. As was proved in [55], an $2n$-point FFT over $GF(q^2)$ may be computed using $\frac{3}{2}n \log n + O(n)$ multiplications and $6n \log n + O(n)$ additions in the field $GF(q)$ [3].

From the formula (3) [55] it follows that

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

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

$$M_2(n) = a_q(n)A(GF(q)) + m_q(n)M(GF(q)) =$$

$$= \frac{9}{2} M(p) n \log n + O(np \log n) = O(p \log p \log\log p) n \log n).$$

Note that the multiplicative constant in the above estimate is rather large owing to the large constant in $M(p)$ estimate. Number n has special form (for another n the constant is to be even more large). Moreover, it is still an open question if the set of Mersenne numbers is infinite. Let $p = 17$, $q = 2^{17} - 1$, $n \approx 2^{16}$, then

$$M_2(n) \approx 27 \cdot n \log^3(2n) = 2^{16} 17^3 27.$$

This bound is close to the complexity of standard school method of multiplication. Hence, the given method of multiplication is better than standard method only if n is greater than 70000.

Program implementation of the above method seems to be more challenging. As follows from [55], the multiplication of polynomials of degree $n < 2^{p-1}$ over

[3] In [55] the formula for primitive 8-th 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 n-point FFT was given inaccurately. The right number is $3n \log n$.

$GF(q)$, where $q = 2^p - 1$ is Mersenne prime, may be performed by $\frac{9}{2} n \log n + 58n + 1$ operations in $GF(q^2)$ (in [55] 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 $(n-1)q^2 \leq q^3/2$ for multiplication on $n-$th 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 school method. However to multiply polynomials of higher degrees one should to increase the field order (and consequently a size of computer memory).

After having read this, a reader can appreciate Schönhage's trick [48]. 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 still not so widely known — the authors know several papers with the similar of weaker results in which [48] is not cited. For instance, in [56] a later paper is followed (and DFT is used for division instead of using Strassen's trick mentioned above, which is seemingly is not known to the author of [56]).

Various aspects of program implementations of multiplication of polynomials over both binary and any field $GF(p)$, including algorithms based on methods Karatsuba, Toom, Schönhage, D.Cantor are discussed in [6, 9, 57, 58, 59, 60]. In [61] it was suggested once more algorithm for program multiplication modulo irreducible trinomial based on multiplication of Töplitz matrix by vector. It is not clear, is the algorithm [61] faster than algorithms [57], since the work [57] was not cited in [61] (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 [62, 63, 64]. Generally complexity and depth of these multipliers are estimated as $O(n^2)$ and $O(\log n)$ respectively.

It was shown in [65], 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.

Close idea was used in [66]. More exactly, there was suggested instead of given irreducible polynomial to take the trinomial divisible by this polynomial and at first to perform multiplication modulo this trinomial (in this case the field is embedded into the ring modulo the trinomial). Also in [66] was given corresponding tables of such trinomials. There is also stated that sometimes this trick is more efficient than using of irreducible pentanomials. To get it, the trinomial is to be chosen in such way that its middle term has a degree not less than a half of the leading term degree. It can be always done, because the tables for any trinomial include its reciprocal.

Similar idea, but with using $x^n - 1$ instead of trinomial was suggested in some works about so called redundant bases. Once more possibility of speeding of modular multiplication based on Montgomery method (see, for example, references in [9]).

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 [67, 68].

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

$$M(GF(2^{1024})) \le 356865, \quad D_M(GF(2^{1024})) \le 31.$$

2.3 Multiplication in Normal Bases

Numerous methods for multiplication in normal bases are known by now, e.g. [5, 69, 70, 71, 72, 73]. Let $T = (t_{i,j})$ be a matrix, whose i-th row is the vector of entries of $\alpha\alpha^{q^i} \in GF(q^n)$ with respect to normal basis B^α. A number of nonzero entries in the matrix T is called complexity of the basis B^α and is denoted $C(B^\alpha)$. 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

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

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 [69] for multiplication over normal basis B in $GF(q^n)$ requires $n(2C(B) + n - 1)$ operations over the subfield $GF(q)$. In [70] 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 the Strassen's trick for modular reduction are to be used to implement multiplication in the standard basis.

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 [11, 74]. In [75] 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 [78] with a worse depth bound.) Such transition circuits allow to perform multiplication in $GF(q^n)$ using $O(n^{1.806})$ operations over $GF(q)$ in depth $O(\log n)$. Exploiting new algorithm for Frobenius operation [76, 77], the complexity of method [75] may be also estimated

as $O(n^{1,667} + \sqrt{n}(n \log q)^{1+o(1)})$. [4] In [75] another construction for transition circuits was proposed, which implies for any normal basis B that the following estimates hold simultaneously:

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

Particularly, if B is a low complexity basis, i.e. $C(B) = O(n^{1.167})$, and q is small enough, i.e. $\log q = o(n^{0.167})$, then $M^{(q)}(GF(q^n)) = O(n^{1.667})$. 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 [80]. Any ONB belongs to one of three types. ONB of type I exists iff $n + 1 = p$ is a prime number, and q is a primitive element modulo p. Type II and III ONB exist iff $q = 2^m$, $(m, n) = 1$, $2n + 1 = p$ 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 to commutation with the basis

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

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 of complexity $O(n)$. Others various kinds of low complexity normal bases with $C(B) = O(n)$ were stated in [81, 82, 71, 83], particularly, Gaussian normal bases (GNB) which are more general than optimal. Using the method of [84] the following bound for the type k [5]Gauss normal basis may be obtained:

$$M(GF(q^n)) \leq (M_q(kn) + 7kn - 8)M(GF(q)).$$

In the particular case of $q = k = 2$ (which is the ONB case) this result was obtained later in [72] independently and was patented. For the type I ONB one has

$$M(GF(q^n)) \leq (M_q(n) + 7n - 8)M(GF(q)).$$

For the type II and III ONB the bounds

$$M^{(q)}(GF(q^n)) \leq 3M_q(n) + O(n \log n),$$
$$M(GF(2^n)) \leq 3M(n) + \frac{3n}{2} \log n + O(n)$$

[4]Frequently used constant $1,667$ is the exponent of special rectangular matrix multiplication [79].

[5] *Type-k GNB* exists in $GF(q^n)$, when $kn + 1$ is prime, and is generated by the element

$$\alpha = \zeta + \zeta^{\gamma} + \ldots + \zeta^{\gamma^{k-1}},$$

where ζ is a primitive root of order $kn + 1$ in $GF(q^{kn})$ and γ is a primitive root of order k in residue field $\mathbb{Z}_{kn+1}$, which is generated together with q multiplicative group $\mathbb{Z}_{kn+1} \setminus \{0\}$.

were proved in [73]. The corresponding construction is settled on the circuit for transition from the basis $\{\alpha_1, \ldots, \alpha_n\}$ to the basis $\{\alpha, \ldots, \alpha^n\}$, $\alpha = \alpha_1 = \zeta + \zeta^{-1}$ of complexity $O(sn \log_s n)$, where $q = s^m$, s is prime, and depth $O(\log_s n)$. Factor 3 in the above estimate occurs from the Strassen's inequality

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

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

Under certain conditions (e.g. f has a few nonzero coefficients) the latter bound may be improved. For example [9], 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

$$M(GF(2^n)) \leq M(n) + \frac{7n}{2} \log n + 4n,$$
$$D_M(GF(2^n)) \leq D(n) + 2 \log n + 2 \log \log n + O(1).$$

In particular,

$$M(GF(2^{191})) \leq 31600, \quad D(GF(2^{191})) \leq 44.$$

For comparison, a method of the paper [70] implies the bound $M(GF(2^{191})) \leq 90916$. Another mentioned above estimate

$$M(GF(q^n)) \leq (M_q(kn) + 7kn - 8)M(GF(q))$$

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

$$M(GF(2^{191})) \leq 77441.$$

Recently mentioned above algorithm [73] for transition between bases $\{\alpha_1, \ldots, \alpha_n\}$ and $\{\alpha, \ldots, \alpha^n\}$, $\alpha = \alpha_1 = \zeta + \zeta^{-1}$ was reopened in [85] (2007) and the next estimate was established:

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

Instead of reduction modulo minimal polynomial of basis B_α (as in [73]) algorithm [85] implies linear transform between redundant bases $\{\alpha, \ldots, \alpha^{2n}\}$ and $\{\alpha_1, \ldots, \alpha_{2n}\}$. The complexity of the transform is $O(sn \log_s n)$, where $q = s^m$, and the depth is $O(\log_s n)$, as it was shown in [73]. In view of equalities

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

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

$$M(GF(2^n)) \leq M(n) + 2n \log n + 10n,$$

$$D_M(GF(2^n)) \leq D(n) + 2 \log n + 4.$$

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.

Fast numeric version of this algorithm was stated by Knuth [89] and was optimized by Schönhage [86] (Knuth—Schönhage algorithm can be also viewed as a modern version of Euclid—Lehmer GCD algorithm; Lehmer algorithm ca be found in [33]). Polynomial version of this algorithm was published in [87] (see also [36]) but the algorithm works incorrectly in some cases; correct algorithm see in [88, 90, 37]).

Subsequently some modifications were introduced (see e.g. [91, 92]). Stehle—Zimmermann's algorithm [92] 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 it was constructed a circuit of the depth $O(n/\log n)$ and complexity $O(n^{1+\epsilon})$ [93]).

Usual binary GCD algorithm seems to be more efficient for small values of n. Its complexity is $O(n^2)$ (see e.g. [9, 92]). 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 has to use completely different approaches.

3.1 Addition Chain Method

A sequence of natural numbers $a_0 = 1, a_1, \ldots, a_m = n$, in which each number a_i is a sum $a_j + a_k$, where $j, k < i$ (indices j and k may coincide), is called an addition chain for n. Parameter m is called a length of the addition chain. The length of the shortest addition chain for n is denoted by $l(n)$. Comprehensive study of addition chains including all classical results may be found in [33].

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

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

The upper bound is due to A. Brauer [94] and proof of the lower bound is due to P. Erdős [95].

Evidently raising to the n-th power using only multiplications corresponds to constructing of 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 background for the use of addition chains in constructing invertors.

A.Brauer [94] 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 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 inverse fast, one can use identity $x^{-1} = y(xy)^{-1}$, as proposed in [96]. Clearly, $xy \in GF(q)$, as far as $(xy)^{q-1} = x^{q^n - 1} = 1$. For computation of $y = (x^{(q^{n-1}-1)/(q-1)})^q$ either Brauer's or Itoh—Tsujii [96]

method can be used (actually, the latter is just a special case of Brauer's method). To finish calculations one has to 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 needs only calculate $y = x^{-1}$.

Less elegant approach based on the formula

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

was followed in [97].

Let $F(GF(q^n))$ and $D_F(GF(q^n))$ denote the maximum on m of the complexity and the depth of the circuit implementing a Frobenius operation $x \to x^{q^m}$ in $GF(q^n)$, $m = 1, \ldots, n$. In standard basis Frobenius operation is equivalent to computation of polynomial $g^{q^m} \bmod f$ and may be performed as a modular composition $g(h) \bmod f$, where $h = x^{q^m} \bmod f$. Actually, 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 $d(n)$ denote the depth of a shortest addition chain for n. Using the addition chain method and a result of paper [98] a standard basis invertor with complexity and depth

$$I^{(q)}(GF(q^n)) \le (l(n-1)+1)\left(M^{(q)}(GF(q^n)) + F^{(q)}(GF(q^n))\right) + n = O(n^{1.667}),$$
$$D_I^{(q)}(GF(q^n)) \le (d(n-1)+1)\left(D_M^{(q)}(GF(q^n)) + D_F^{(q)}(GF(q^n))\right) + 1 = O(\log^2 n)$$

can be constructed [101]. The same scheme of calculations in the case of a normal basis implies the following bounds:

$$I^{(q)}(GF(q^n)) \le (l(n-1)+1)M^{(q)}(GF(q^n)) + n = O(n^{1.806}),$$
$$D_I^{(q)}(GF(q^n)) \le (d(n-1)+1)D_M^{(q)}(GF(q^n)) + 1 = O(\log^2 n),$$

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 $O(\log n)$ [75]. 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 [76] (see also [77]) proved that the complexity of modular composition is equal to $(n \log q)^{1+o(1)}$. Hence it follows that

$$I^{(q)}(GF(q^n)) \le (l(n-1)+1)(M^{(q)}(GF(q^n)) + F^{(q)}(GF(q^n))) + n = (n \log q)^{1+o(1)},$$

for standard base and for normal base

$$I^{(q)}(GF(q^n)) \le (l(n-1)+1)M^{(q)}(GF(q^n)) + n = O(n^{1,667}).$$

But this method does not give the bound $O(\log^2 n)$ for the depth.

The above estimates based on the A.Brauer's (1939) method seem to be hardly familiar to cryptographers. Some particular cases of the Brauer's method like Itoh—Tsujii method [96] or TYT-method [99] are frequently cited and exploited. These methods does not provide optimal complexity (for example, method [99] yields to the general Brauer's for $n = 24, 44, 47, \dots$). Using the Brauer's method some very recent results can be improved straightforwardly, e.g. the complexity bounds [100] for inversion in the fields $GF(2^{384})$, $GF(2^{480})$ (see details in [101]).

To minimize the depth of invertor we may use a version of right-to-left binary method (see [33, 101]). 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 [102], 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 [101].

Thereby, a standard-basis invertor of complexity

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

and depth

$$D_I^{(q)}(GF(q^n)) \le (\delta(n)+1)\left(D_M^{(q)}(GF(q^n)) + D_F^{(q)}(GF(q^n))\right) + 1$$

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

$$I^{(q)}(GF(q^n)) \le \left(\lambda(n-1) + (1+o(1))\frac{\lambda(n)}{\lambda(\lambda(n))}\right) M^{(q)}(GF(q^n)),$$
$$D_I^{(q)}(GF(q^n)) \le (\delta(n-1)+1)D_M^{(q)}(GF(q^n)) + 1.$$

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

3.2 Logarithmic Depth Method

The $GF(q^n)$ invertors of logarithmic depth (over $GF(q)$) invertors were presented in [103, 104] (the former paper considers the binary case). The authors didn't

estimate 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. Invertor in $GF(2^n)$ of the depth $(6.44 + o(1)) \log n$ and complexity $(2/3)n^4 + o(n^4)$ was constructed in [105] (the result holds for an arbitrary field basis). In the same paper a standard basis invertor with the depth $O(\log n)$ and complexity $O(n^{1.667})$ was constructed. The latter result was extended in [106] 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 [106] 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 $MM(m, GF(q^n))$ and $D_{MM}(m, GF(q^n))$ respectively. Combining ideas from [107, 44, 42] the following bounds for multiple multiplication circuit were proved in [106]:

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

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

The use of multiple multiplications rests on the following result [105, 106]: 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

$$(2r - 1)(mF(GF(q^n)) + MM(m, GF(q^n))) + (r - 1)M(GF(q^n)),$$
$$2(D_F(GF(q^n)) + D_{MM}(m, GF(q^n))) + D_M(GF(q^n))$$
$$+ (r - 2) \max\{D_F(GF(q^n)) + D_{MM}(m, GF(q^n)), D_M(GF(q^n))\}$$

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:

$$I^{(q)}(GF(q^n)) = O(rn^{1/r}(n^w + n^{1.5} \log n \log\log n)),$$
$$D^{(q)}_I(GF(q^n)) = O(r \log n),$$

where w is somewhat smaller then 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 $T(GF(q^n))$ and $D_T(GF(q^n))$ complexity and depth of a transition circuit (bilateral transition is considered). Then exploiting an 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 [106]:

$$I^{(q)}(GF(q^n)) = O\left(R^b n^{1+2/R}\right) + O\left(R \sqrt[R]{n}\right) T^{(q)}(GF(q^n)),$$
$$D^{(q)}_I(GF(q^n)) = O\left(R\left(\log n + D^{(q)}_T(GF(q^n))\right)\right),$$

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 leads over 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's why for n of order of several hundreds various versions of addition chain method are used (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 complexity is also decreases. Seemingly, the idea of applying composite fields to minimize depth initially appeared in [96]. Multiple approaches for arithmetic implementation in composite fields were proposed in [63, 108, 67, 68, 70, 109, 111].

Combining [73, 75], one can prove that if n and m are coprime, then for some normal basis

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

Particularly, if $n = \Omega(m)$, then

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

If N is an ϵ-smooth number, i.e. $N = n_1 \dots n_m$, all n_i are coprime, $n_1 + \dots + n_m = O(N^{\epsilon})$, then $M^{(q)}(GF(q^N)) = O(N^{1+0.806\epsilon})$. 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 [111] the authors considered a general construction of 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 is to be improved. Obviously, the authors [111] didn't know about work [48], where better result was obtained.

Towers in [111] look like

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

where each prime factor of P_j is either a factor of $q-1$, or p, that is characteristic of the field. Besides, P_j is an even number if $q = 1 \bmod 4$ only. On any floor of such tower one can choose a basis with minimal polynomial being binomial (similar towers were considered independently in [112] and got the name of optimal tower fields). To implement multiplication on each floor Toom's method [32] and FFT method (in the latter case primitive roots belong to the previous floors) were

used. However fundamental formulas on the page 227 [111] are questionable by the reason that they relies 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 (really, 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 [111] the following estimates were proved:

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

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 so large as one based on fast Euclidean algorithm.

In [113] 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 way that

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

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

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

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

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

All above statements from [113] may be improved at the expense of more accurate estimating of the complexity of multiplication by constants implied in FFT algorithm. More exactly, with a suitable choice of a basis multiplier and invertor in $GF(2^n)$, $n = m^s$, $s \geq s_m$, with the next complexity and depth bounds may be constructed:

$$M(GF(2^n)) = O_m(n \log n \log\log n), \quad I(GF(2^n)) = O_m(M(GF(2^n))),$$

$$D_M(GF(2^n)) = O_m(\log n), \quad D_I(GF(2^n)) = O_m(\log^2 n).$$

Sometimes the above bounds may be pointed in more precise form. For instance, if $m = p$ is a prime, 2 is a primitive root modulo p (this is exatcly the condition of existing 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

$$M(GF(2^n)) = O\left(\frac{p}{\log_2 p} n \log_p n \log_2 \log_p n\right), I(GF(2^n)) = O_p(M(GF(2^n))),$$

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

$$M(GF(2^n)) = 5\frac{5}{8} n \log_3 n \log_2 \log_3 n + O(n \log_3 n),$$

$$I(GF(2^n)) \lesssim \frac{5}{2} M(GF(2^n)),\ D_M(GF(2^n)) \sim \frac{6}{\log_2 3} \log_2 n.$$

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

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

were constructed in [110]. The method implies, for example,

$$M(GF(2^{1024})) \leq 357992, \quad I(GF(2^{1024})) \leq 538033.$$

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

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

(similar bases for $k = 1$ were considered in [114] in order to implement AES S-boxes). Then consider in the floor the standard basis $\{1, \alpha_k\}$ or normal basis $\{\alpha_k, \alpha_k^{2^{k+1}}\}$. With the use of the above construction asymptotically worse bounds

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

$D_I(GF(2^n)) = O(\log^3 n)$ may be established. However, S. Zikrin proved that for $n \leq 64$ best multipliers and invertors than in [67] may be constructed. For example, he obtained the next estimates:

$$M(GF(2^{16})) \leq 382, \ D_M(GF(2^{16})) \leq 11,$$

$$I(GF(2^{16})) \leq 479, \ D_I(GF(2^{16})) \leq 26,$$

$$M(GF(2^{32})) \leq 1233, \ D_M(GF(2^{32})) \leq 13,$$

$$I(GF(2^{32})) \leq 1714, \ D_I(GF(2^{32})) \leq 48.$$

$$M(GF(2^{64})) \leq 3943, \ D_M(GF(2^{64})) \leq 18,$$

$$I(GF(2^{64})) \leq 5609, \ D_I(GF(2^{64})) \leq 75,$$

$$M(GF(2^{128})) \leq 12728, \ D_M(GF(2^{128})) \leq 24,$$

$$I(GF(2^{128})) \leq 18587, \ D_I(GF(2^{128})) \leq 114.$$

One may compare this estimates with those from [67]:

$$M(GF(2^{128})) \leq 12476, \ D_M(GF(2^{128})) \leq 25,$$

$$I(GF(2^{128})) \leq 18316, \ D_I(GF(2^{128})) \leq 170.$$

Note that for a standard basis with the irreducible polynomial $x^{128}+x^7+x^2+x+1$ in $GF(2^{128})$ the next estimates hold: $M(GF(128)) \leq 33042$, $D_M(GF(128)) \leq 11$. Complexity and depth of Karatsuba's multiplier in this case are estimated as 12343 and 18 respectively. However the estimates for an invertor look as 200000 and not less than 200.

4.2 Minimization of Inversion Depth in Composite Fields

In this section we observe some recursive methods [115, 116] aimed at constructing depth-efficient invertors in composite binary fields for the values of n not more than several hundreds.

Suppose that n is odd, $D_M(GF(2^n)) \geq D_S(GF(2^n)) + 1$, where $S(GF(2^n))$ is the complexity of squaring in $GF(2^n)$. Applying a method from the paper [108], one can construct invertor and multiplier with the following recursive bounds on the complexity and the depth:

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

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 $GF(2^n)$, $D_M(GF(2^n)) \geq D_S(GF(2^n)) + 2$, then for multiplication in $B = B_1 \otimes B_2$ we have

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

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

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

If B_1 is a normal basis, then $S(GF(2^n)) = 0$.

If in the field tower $GF(((2^n)^2)^2)$ 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 next relations hold

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

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

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

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 next relations hold:

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

and for inversion:

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

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 $GF(2^n)$ arbitrary basis B_1 and consider the basis $B = B_1 \otimes B_2$ in $GF(2^{6n})$. Suppose that $D_M(GF(2^n)) \geq D_S(GF(2^n)) + 2$. Then for multiplication and inversion in B one has:

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

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 $\alpha_3^2 + \alpha_3 = \alpha_1\alpha_2$, B is arbitrary basis in $GF(2^n)$, then for the basis $B_2 \otimes B$ in $GF(2^{8n})$ the following relations hold:

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

If B is the normal basis, then

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

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:

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

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

$$\begin{aligned} M(GF(2^{30n})) &\leq 315M(GF(2^n)) + 1140n, \\ D_M(GF(2^{30n})) &\leq D_M(GF(2^n)) + 8, \\ I(GF(2^{30n})) &\leq I(GF(2^n)) + 566M(GF(2^n)) + 1537n, \\ D_I(GF(2^{30n})) &\leq 6D_M(GF(2^n)) + 17 + \max\{D_I(GF(2^n)) \\ &\quad + \max\{D_M(GF(2^n)), 6\}, D_M(GF(2^n)) + 8\}. \end{aligned}$$

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 pseudo-Mersenne prime number. Mersenne fields were mentioned above. Several techniques for

Таблица 1:

n	$I(GF(2^n))$	$D_I(GF(2^n))$
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

implementing multiplication in pseudo-Mersenne fields $GF(q^n)$, $n = 2^k, 3^k$, were proposed in [112, 117] aimed at the application to elliptic and hyperelliptic curve cryptography (see [118]). Special bases (so called optimal tower bases [112, 117]) were used. This bases are a special cases of bases considered in [111].

5.1 Multiplication in Optimal Towers of Pseudo-Mersenne Fields

Improving the results of [112], in [117] multipliers of complexity

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

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

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

Some effective applications of similar results in hyperelliptic cryptography were noted in [118]. In [119] some improvements were proposed for this circuits based on FFT in the case of Fermat number $q = 2^{16} + 1$.

Independently related results were obtained in [73], namely for $q = p^n$, $p = 2^{16} + 1$, the next bound was proved:

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

Using convolution modulo $x(x^{2^{k+1}}-1)/(x^2-1)$ it was proved [120] that

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

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, [2]), multiplicative constants in the terms involving $M(GF(q))$ 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 [120]:

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

Also in [120] 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 $k_0 = 0, 1$:

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

These results rely on using FFT modulo 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 [2]). Multiplication in $GF(q^n)$ was implemented using 3 FFT's and reduction modulo an irreducible binomial.

The method proposed in the paper [121] requires 2 FFT's on the 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 Montgomery method was used. For example, if 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

$$mM(GF(p)) + (m-1)M(1/m,p) + (6m^2 - 7m + O(1))A(GF(p)).$$

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

$$2mM(GF(p)) + (m-1)M(1/m,p) + (4m^2 - 4m + O(1))A(GF(p))$$

was obtained. Effective application of this results in elliptic curve cryptography was demonstrated in [122].

We remark that instead of Montgomery multiplication the use of usual modular multiplication is possible. It implies some simplifications, but corresponding algorithm for modular multiplication in the frequency domain requires roughly $2m^2$ multiplications on 2^k more than algorithm [121]. It is essential for software implementation, but it isn't so important for constructing of a circuit.

6 Multiplication in Fields of Small Characteristic

In the last years numerous papers on the so-called pairing-based cryptography were published (see e.g. [10]). A problem of primary practical importance in this research direction is the efficient implementation of pairings. In [123] an efficient algorithm was proposed for Tate pairing in some supersingular curves over fields of characteristics 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 [124, 125, 126, 127].

In [128, 129] a fast algorithm was presented for Tate pairing on 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 [123]. In [130] some improvements of Duursma—Lee (DL) algorithm for binary fields were suggested. In fact, similar improvements are possible in general case (see e.g. [10]). Another improvement of the DL algorithm was suggested in [131].

To implement the DL algorithm for 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

$$M(GF(p^{2pn})) \leq (6p - 3)M(GF(p^n)) + O(p^2 n M(GF(p))).$$

The smallest field to be 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 a method of paper [132]. The following complexity and depth estimates for multiplication in $GF(7^{14n})$ were proved in [133]:

$$M(GF(7^{14n})) \leq 13M(GF(7^{2n})) + 258nA(GF(7)),$$
$$D_M(GF(7^{14n})) \leq 11D_A(GF(7)) + D_M(GF(7^{2n})).$$

Particularly, $M(GF(7^{14 \cdot 31})) \leq 698\ 554$.

Список литературы

[1] J.H. McClellan, C.M. Rader, *Number theory in digital signal processing*, Prentice-Hall, 1979.

[2] R.E. Blahut, *Fast algorithms for digital signal processing*, Addison-Wesley, Reading, Massachusetts, USA, 1985.

[3] G.B. Agnew, R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone, An implementation for a fast public-key cryptosystem, *Journal of Cryptology* **3** (1991), 63–79.

[4] G.B. Agnew, T. Beth, R.C. Mullin, S.A. Vanstone, Arithmetic operations in $GF(2^m)$, *Journal of Cryptology* **6** (1993), 3–13.

[5] D. Jungnickel, *Finite fields. Structure and arithmetic*, Wissenschaftsverlag, Mannheim, Leipzig, Wien, Zurich, 1993.

[6] A.J. Menezes, P.C. van Oorshot, S.A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997.

[7] I. Blake, G. Seroussi, N. Smart, *Elliptic curves in cryptography*, Cambridge University Press, 1999.

[8] I. Blake, G. Seroussi, N. Smart, *Advances in elliptic curve cryptography*, Cambridge University Press, 2005.

[9] A.A. Bolotov, S.B. Gashkov, A.B. Frolov, A.A. Chasovskikh, *Elementary introduction in elliptic cryptography: algebraic and algorithmic background*, Moscow, 2006 (in Russian).

[10] A.A. Bolotov, S.B. Gashkov, A.B. Frolov, *Elementary introduction in elliptic cryptography: cryptographic protocols on elliptic curves*, Moscow, 2006 (in Russian).

[11] O.B. Lupanov *Asymptotic bounds on complexity of control systems.* Moscow University Press, Moscow, 1984 (in Russian).

[12] P.E. Dunne, *The complexity of boolean networks*, Academic Press, 1988.

[13] I. Wegener, *The Complexity of Boolean functions*, Wiley, Stuttgart, 1987.

[14] R. Lidl, H. Niederreiter, *Finite fields*, Addison-Wesley, 1983.

[15] N.P. Red'kin, Minimal realization of a binary adder, *Problemy Kibernetiki* **38** (1981), 181–216 (in Russian).

[16] V.M. Khrapchenko, Asymptotic estimation of addition time of a parallel adder, *Problemy Kibernetiki* **19** (1967), 107–122 (in Russian). English translation in *Syst. Theory Res.* **19** (1970), 105–122.

[17] S.B. Gashkov, M.I. Grinchuk, I.S. Sergeev, On constructing small depth adders, *Diskretnyi analiz i issledovanie operaciy, Ser.* 1 **14** №1 (2007), 27–44 (in Russian).

[18] M.I. Grinchuk Refinement the upper bound for the depth of adder and comparator. *Diskretnyi analiz i issledovanie operaciy, Ser.* 1 **15** №2 (2008), 12–22 (in Russian).

[19] V.M. Khrapchenko. On possibility of refinement of estimation for delay of parallel adder *Diskretnyi analiz i issledovanie operaciy, Ser.* 1 **14** №1 (2007), 27–44 (in Russian).

[20] D.J. Bernstein, *Multidigit multiplication for mathematicians*, http://cr.yp.to/papers.html#m3, 2004.

[21] G.K. Stolyarov, *Method for parallel multiplication in digital computers and device for its implementation*, Author certificate cl. 42, V. 14, №126668, 1960.

[22] A.A. Karatsuba, Yu.P. Ofman, Multiplication of multidigit numbers on automata, *DAN USSR* **145** №2 (1962), 293–294 (in Russian). Eng. transl. in *Soviet Phys. Dokl.* **7** (1963), 595–596.

[23] A. Avizienis, Signed-digit number representation for fast parallel arithmetic, *IEEE Trans. Elect. Comput.* **10** (1961), 389–400.

[24] C.S. Wallase, A suggestion for a fast multiplier, *IEEE Trans. Elect. Comput.* **13** (1964), 14–17.

[25] Khrapchenko V. M. Some estimates on the time of multiplication. // Problemy kibernetiki. T. 33. — Moscow, Nauka, 1978. — 221–227.

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

[27] M. Paterson, N. Pippenger, U. Zwick, Optimal carry save networks. // LMS Lecture Notes Series. — V. 169. Boolean function Complexity. — Cambridge University Press, 1992. — P. 174–201.

[28] E.Grove, Proofs with potential. — Ph.D. thesis, U.C. Berkeley, 1993.

[29] I.S. Sergeev, On the depth of circuits for multiple addition and multiplication of numbers, *Proc. VI scientific school on discrete mathematics and its applications (Moscow, IPM RAN, April 2007)* **II** (2007), 40–45 (in Russian).

[30] Karatsuba A.A. Complexity of computations.// Trudy Mat. Instituta RAN.1995. V. 211, 1–17.

[31] A.V. Chashkin, Fast multiplication and addition of integer numbers, *"Discrete mathematics and applications"* MGU, 2001, 91–110 (in Russian).

[32] A.L. Toom, The complexity of a scheme of functional elements realizing the multiplication of integers, *DAN USSR* **150** №3 (1963), 496–498 (in Russian). Eng. transl. in *Soviet Math. Dokl.* **3** (1963), 714–716.

[33] D. Knuth, *The art of computer programming*, third ed., Addison-Wesley, 1998.

[34] S. Cook, *On the minimum computation time of functions*, Ph.D. thesis, Harvard Univ., 1966.

[35] A. Schönhage, V. Strassen, Schnelle multiplikation großer zahlen, *Computing* **7** (1971), 271–282.

[36] A. Aho, J. Hopkroft, J. Ullman, *The design and analysis of computer algorithms.* Addison-Wesley, 1974.

[37] J. von zur Gathen, J. Gerhard, *Modern computer algebra*, Cambridge University Press, 1999.

[38] Martin Fürer, *Faster integer multiplication*, http://www.cse.psu.edu/~furer/Papers/mult.pdf, 2007.

[39] Anindya De, Piyush P Kurur, Chandan Saha, Ramprasad Saptharishi, *Fast Integer Multiplication Using Modular Arithmetic*, arXiv: 0801.1416v2 [cs. SC] 6 May 2008

[40] P.Gaudry, A.Kruppa, P.Zimmermann, A GMP-based Implementation of Schonhage-Strassen's Large Integer Multiplication Algorithm, *ISAAC'07, Waterloo, Ontario, Canada, 2007.*

[41] P. Naudin, C. Quitte, *Algoritmique algebrique*, Masson, Paris, Milan, Barselone, Bonn 1992.

[42] J. Reif, S. Tate, Optimal size integer division circuits, *SIAM J. Comput.* **19** №5 (1990), 912–925.

[43] Beame P., Cook S., Hoover H. Log depth circuits for division and related problems. // SIAM J. Comput. — 1986. — V. 15, n. 4. — P. 994–1003.

[44] J. Hastad, T. Leighton, *Division in $O(\log n)$ depth using $O(n^{1+\epsilon})$ processors*, http://www.nada.kth.se/~yohanh/paraldivision.ps, 1986.

[45] P.D. Barret, Implementing the Rivest, Shamir and Adleman public key encryption algorithm on a standard digital signal processor, *Advances in Cryptology, Proc. Crypto-86, LNCS* **263** (1987), 311–323.

[46] V. Strassen, Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten, *Numer. Math.* **20** (1973), 238–251.

[47] S.B. Gashkov, V.N. Chubarikov, *Arithmetic. Algorithms. Complexity of computation.* Moscow, Nayka, 1996 (in Russian).

[48] A. Schönhage, Schnelle multiplikation von polynomen über körpern der charakteristik 2, *Acta Inf.* **7.** (1977), 395–398.

[49] D. Cantor, E. Kaltofen, On fast multiplication of polynomials over arbitrary algebras, *Acta Informatica* **28** (1991), 693–701.

[50] D.V. Chudnovsky, G.V. Chudnovsky, Algebraic complexities and algebraic curves over finite fields, *J. Complexity* **4** (1988), 285–316.

[51] I.E. Shparlinski, M.A. Tsfasman, S.G. Vladuts, Curves with many points and multiplication in finite fields, *LNM* **1518** (1992), 145–169.

[52] S. Ballet, R. Roland, Multiplication algorithm in a finite field and tensor rank of the multiplication, *J. Algebra* **272** №1 (2004), 173–185.

[53] D. Cantor, On arithmetic algorithms over finite fields, *J. of combinatorial theory, Series A* **50** (1989), 285–300.

[54] J. von zur Gathen, J. Gerhard, Arithmetic and factorization of polynomials over $GF(2)$, *Proc. ISSAC 96 (Zürich)* (1996), 1–9.

[55] S.B. Gashkov Remarks on fast polynomial multiplication, Fourier and Hartley transforms. // Discrete mathematics. — 2000. — v. 12, №. 3. — p. 124–153.

[56] Vasilenko O. N. Number-theoretic algorithms in cryptoraphy. M.: MCNMO, 2007.

[57] Hankerson D., López J.H., Menezes A. Software implementation of elliptic curve cryptography over binary fields. CHES 2000, Lecture Notes in Computer Science No 1965(2000), 1-23.

[58] R.P. Brent, P.Gaudry, E. Thome, P.Zimmerman. Faster multiplication in $GF(2)[x]$. Preprint INRIA №6359, 16 p. November 2007.

[59] C.Negre, T.Plantard. Prime field multiplication in adapted modular system using Lagrange representation. Preprint, 2005.

[60] J.-C. Bajard, L.Imbert, T.Plantard. Modular number systems: Beyond the Mersenne family. SAC'04: 11th International workshop on selected areas in Cryptography, pp.159-169, 2004.

[61] H. Fan, M.A. Hasan, Alternarive to the Karazuba Algorithm for software implementation of $GF(2^n)$ multiplication.

[62] E.D. Mastrovito, *VLSI architectures for computation in Galois fields*, Ph.D. Thesis, Linköping University, Dept. Electr. Eng., Sweden, 1991.

[63] J. Guajardo, T. Güneysu, S. Kumar, C. Paar, J. Pelzl, Efficient hardware implementation of finite fields with application to cryptography, *Acta Appl. Math.* **93** (2006), 75–118.

[64] S. Erdem, T. Yanik, C. Koc, Polynomial basis multiplication over $GF(2^n)$, *Acta Appl. Math.* **93** (2006), 33–55.

[65] O. Ahmadi, A. Menezes. *Irreducible polynomials of maximum weight*, preprint, 2005.

[66] C. Doche. Redundant Trinomials for Finite Fields of Characteristic 2 ACISP 2005, LNCS 3574, pp. 122-133, 2005.

[67] C. Paar, *Effective VLSI architectures for bit paralel computation in Galois fields*, Ph.D. Thesis, Universität GH Essen, Germany, 1994.

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

[69] J.L. Massey, J.K. Omura, *Apparatus for finite fields computation*, US patent 4587627, 1986.

[70] A. Reyhani-Masoleh, M.A. Hasan, On effective normal basis multiplication, *Proc. IndiaCRYPT-2000, LNCS* **1977** (2000), 213–224.

[71] J. von zur Gathen, M. Nöcker, Fast arithmetic with general Gauss periods, *Theor. Comp. Science* **315** (2004), 419–452.

[72] I. Blake, R. Roth, G. Seroussi, *Efficient arithmetic in* $GF(2^n)$ *through palindromic representation*, Hewlett-Packard, HPL–98–134, 1998.

[73] A.A. Bolotov, S.B. Gashkov, Fast multiplication in normal bases of finite fields, *Discretnaya matematika* **13** №3 (2001), 3–31 (in Russian). Eng. transl. in *Discrete Mathematics and Applications* **11** №4 (2001), 327–356.

[74] O.B. Lupanov, On rectifier and contact rectifier circuits, *DAN USSR* **111** №6 (1956), 1171–1174 (in Russian).

[75] I.S. Sergeev, On constructing circuits for transitions between polynomial and normal bases in finite fields, *Diskretnaya matematika* **19** №3 (2007), 89–101 (in Russian). Eng. transl. in *Discrete Mathematics and Applications* **17** (2007).

[76] C. Umans, *Fast polynomial factorization, modular composition, and multipoint evaluation of multivariate polynomials in small characteristic*, http://www.cs.caltech.edu/~umans/papers/U07.pdf, 2007.

[77] K.Kedlaya, C. Umans, *Fast modular composition in any characteristic*, april, 2008.

[78] E. Kaltofen, V. Shoup, Subquadratic-time factoring of polynomials over finite fields, *Math. Comput.* **67** №223 (1998), 1179–1197.

[79] Huang X., Pan V., Fast rectangular matrix multiplication and applications. *J. Complexity* (1998) **14**, 257–299.

[80] R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone, R.M Wilson, Optimal normal bases in $GF(p^n)$, *Discrete Applied Mathematics*, **22** (1988/89), 149–161.

[81] D.W. Ash, I.F. Blake, S.A. Vanstone, Low complexity normal bases, *Discrete Applied Mathematics* **25** (1989), 191–210.

[82] J.E. Seguin, Low complexity normal bases, *Discrete Applied Mathematics* **28** (1990), 309–312.

[83] S. Feisel, J. von zur Gathen, M.A. Shokrollahi, Normal bases via general gauss periods, *Math. Comp.* **68** №225 (1999), 271–290.

[84] S. Gao, J. von zur Gathen, D. Panario, Gauss periods and fast exponentiation in finite fields, *Proc. Latin'95 (Valparaiso, Chile), LNCS* **911** (1995), 311–322.

[85] J. von zur Gathen, M.A. Shokrollahy, J. Shokrollahy, Efficient multiplication using type 2 optimal normal bases, *LNCS* **4547** (2007), 55–68.

[86] A. Schönhage, Schnelle berechnung von kettenbruchentwicklungen, *Acta Inf.* **1** (1971), 139–144.

[87] Moenk R. Fast algorithm of GCD's. // Proc. 5th Ann. ACM Symp. on Theory of Computing. — 1973. — P. 142–151.

[88] Brent R., Gustavson F., Yun D. Fast solution of Toeplitz systems of equations and computation of Padé approximants. // J. Algorithms. — 1980. — V. 1. — P. 259–295.

[89] Knuth D. The analysis of algorithms. // Proc. Intern. Congress of Math. (Nice, France). — 1970. — V. 3. — P. 269–274.

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

[91] N. Möller, On Schönhhage's algorithm and subquadratic integer gcd computation.

[92] D. Stehlé, P. Zimmermann, A binary recursive GCD algorithm, *Proc. ANTS-VI (Burlington, USA, 2004), LNCS* **3076** (2004), 411–425.

[93] B. Chor, O. Goldreich, An improved parallel algorithm for integer GCD, *Algorithmica* (1990), **5**, 1–10.

[94] A. Brauer, On addition chains, *Bull. AMS* **45** (1939), 736–739.

[95] P. Erdös, Remarks on number theory III: On addition chains, *Acta Arithm.* (1960) **6**, 77–81.

[96] T. Itoh, S. Tsujii, A fast algorithm for computing multiplicative inverses in $GF(2^n)$ using normal bases, *Inform. and Comp.* **78** (1988), 171–177.

[97] J. von zur Gathen, M. Nöcker, Exponentiation in finite fields: theory and practice, *Applied Algebra, Proc. AAECC-12, LNCS* **1255** (1997), 88–113.

[98] R. Brent, H. Kung, Fast algorithms for manipulating formal power series, *J. ACM* **25** №4 (1978), 581–595.

[99] N. Takagi, J. Yoshiki, K. Takagi, A fast algorithm for multiplicative inversion in $GF(2^n)$ using normal basis, *IEEE Trans. on Comp.* **50** №5 (2005), 394–398.

[100] K. Chang, H. Kim, J. Kang, H. Cho, An extension of TYT algorithm for $GF((2^n)^m)$ using precomputation, *Inform. Proc. Letters* **92** (2004), 231–234.

[101] S.B. Gashkov, I.S. Sergeev, An application of the method of addition chains to inversion in finite fields, *Discretnaya matematika* **18** №4 (2006), 56–72 (in Russian). Eng. transl. in *Discrete Mathematics and Applications* **16** №6 (2006), 601–618.

[102] A.C. Yao, On the evaluation of powers, *SIAM J. on Comput.* **5** (1976), 100–103.

[103] B.E. Litow, G.I. Davida, $O(\log n)$ time for finite field inversion, *LNCS* **319** (1988), 74–80.

[104] J. von zur Gathen, Inversion in finite fields, *J. Symbolic Comput.* **9** (1990), 175–183.

[105] I.S. Sergeev, Circuits of logarithmic depth for inversion in finite fields of characteristic two, *Matematicheskie Voprosy kibernetiki* **15** (2006), 35–64 (in Russian).

[106] S.B. Gashkov, I.S. Sergeev, On constructing circuits of logarithmic depth for inversion in finite fields, *Discretnaya matematika* **20** №4 (2008) (in Russian). Eng. transl. in *Discrete Mathematics and Applications* **18** (2008).

[107] W. Eberly, Very fast parallel polynomial arithmetic, *SIAM J. Comput.* **18** №5 (1989), 955–976.

[108] M. Morii, M. Kasahara, Efficient construction of gate circuit for computing multiplicative inverses in $GF(2^n)$, *Trans. of IEICE* **72** №1 (1989), 37–42.

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

[110] C. Paar, J.L. Fan, *Efficient inversion in tower fields of characteristic two*, ISIT, Ulm, Germany, 1997.

[111] V.B. Afanassiev, A.A. Davydov, Finite field tower: iterated presentation and comlpexity of arithmetic, *Finite fields and their applications*, bf 8, 216—232 (2002).

[112] D.V. Bailey, C. Paar, Efficient arithmetic in finite fields extensions with application in elliptic curve cryptography, *Journal of cryptology* **14** (2001), 153–176.

[113] A.A. Burtzev, I.B. Gashkov, S.B. Gashkov, On the complexity of boolean circuits for arithmetic in some towers of finite fields, *Vestnik MGU, ser.1. Mathem. Mechan.* **5** (2006), 10–16 (in Russian).

[114] D.Canright. A very compact Rijndael S-box. Technical Report NPS-MA-04- 001, Naval Postgraduate School, September http://library.nps.navy.mil/uhtbin/hyperion-image/NPS-MA-05-001.pdf, 2004.

[115] S.B. Gashkov, R.A. Khokhlov, On the depth of logical circuits for operations in fields $GF(2^n)$, *Chebyshevsky sbornik* **4** №4(8) (2003), 59–71 (in Russian).

[116] A.A. Burtzev, On the circuits for multiplication and inversion in composite fields $GF(2^n)$, *Chebyshevsky sbornik* **7** №1(17) (2006), 172–185 (in Russian).

[117] S. Baktir, B. Sunar, Optimal tower fields. *IEEE Trans. Comp.* **53** №10 (2004), 1231–1243.

[118] S. Baktir, J. Pelzl, T. Wollinger, B. Sunar, C. Paar, Optimal tower fields for hyperelliptic curve cryptosystems, *Proc. IEEE 38th ACSSC* (2004).

[119] S. Baktir, B. Sunar, Achieving efficient polynomial multiplication in Fermat fields using fast Fourier transform, *Proc. ACMSE'06* (2006), ACM Press, 549–554.

[120] A.A. Burtzev, S.B. Gashkov, On the circuits for arithmetic in composite fields of large characteristic, *Chebyshevskyi sbornik* **7** №1(17) (2006), 186–204 (in Russian).

[121] S. Baktir, B.Sunar, Frequency domain finite field arithmetic for elliptic curve cryptography, *Proc. ISCIS 2006, LNCS* **4263** (2006), 991–1001.

[122] S. Baktir, S. Kumar, C. Paar, B. Sunar, A state-of-the-art elliptic curve cryptographic processor operating in the frequency domain, *Mobile Networks and Appl.* **12** №4 (2007), 259–270.

[123] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, M. Scott, Efficient algorithms for pairing-based cryptosystems, *Proc. Crypto-2002, LNCS* **2442** (2002), 354–368.

[124] R. Granger, D. Page, M. Stam, Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three, *IEEE Trans. on Comp.* **54** №7 (2005), 852–860.

[125] D. Page, N.P. Smart, Hardware implementation of finite fields of characteristic three, *Proc. CHES-2003*, 529–539.

[126] G. Bertoni, J. Guajardo, S. Kumar, G. Orlando, C. Paar, T. Wolinger, Efficient $GF(p^m)$ arithmetic architectures for cryptographic applications, *LNCS* **2612** (2003), 158–175.

[127] T. Kerins, W.P. Marnane, E.M. Popovici, P.S.L.M. Barreto, Efficient hardware for Tate pairing calculation in characteristic three, *Proc. CHES-2005, LNCS* **3659** (2005), 412.

[128] I. Duursma, H.-S. Lee, *Tate pairing implementation for tripartite key agreement*, Cryptology ePrint Archive, Report 2003/053. http://eprint.iacr.org/2003/053.

[129] I. Duursma, H.-S. Lee, Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$, *Proc. Asiacrypt-2003, LNCS* **2894** (2003), 111–123.

[130] 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.

[131] P.S.M.L. Barreto, S. Galbraith, C. O'Eigeartaigh, M. Scott, *Efficient pairing computation on supersingular Abelian varieties*, Cryptology ePrint Archive, Report 2004/375. http://eprint.iacr.org/2004/375.

[132] E. Lee, H.-S. Lee, 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.

[133] A.A. Burtzev, On the boolean circuits for multiplication in finite fields of odd characteristics, *Proc. VI scientific school on discrete mathematics and its applications (Moscow, IPM RAN, April 2007)* **I** (2007), 13–16 (in Russian).