С. Б. Гашков, И. С. Сергеев, Сложность вычислений в конечных полях, *Фундамент. и прикл. матем.*, 2012, том 17, выпуск 4, 95–131

Сложность вычислений в конечных полях

С. Б. ГАШКОВ
Московский государственный университет им. М. В. Ломоносова
e-mail: sbgashkov@gmail.com

И. С. СЕРГЕЕВ
Московский государственный университет им. М. В. Ломоносова
e-mail: isserg@gmail.com

УДК 512.624

Ключевые слова: поля Галуа, умножение, инвертирование, булевы схемы, сложность, глубина.

Аннотация

Дан обзор некоторых работ о сложности реализации арифметических операций в конечных полях логическими схемами.

Abstract

S. B. Gashkov, I. S. Sergeev, Complexity of computation in finite fields, Fundamentalnaya i prikladnaya matematika, vol. 17 (2011/2012), no. 4, pp. 95—131.

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

Введение

Эффективная реализация арифметических операций в конечных полях имеет важное значение в криптографии, кодировании, цифровой обработке сигналов и других областях (см., например, [2, 4, 5, 26, 39, 40, 61, 62, 101, 112]). В этом обзоре речь пойдёт в основном о методах реализации арифметических операций в конечных полях булевыми схемами (так мы для краткости именуем то, что в московской школе теории сложности булевых функций принято называть схемами из функциональных элементов в базисах, состоящих из функций алгебры логики [25]; полное наименование мы не используем ещё и потому, что термины «элемент» и «базис» далее будут использоваться в их обычном алгебраическом смысле). Схемы для умножения и инвертирования (т. е. вычисления мультипликативного обратного) в конечных полях далее для краткости называются *мультиплерами* и *инверторами*. На практике чаще используются поля характеристики 2, но будут рассмотрены также схемы для полей нечётной

Фундаментальная и прикладная математика, 2011/2012, том 17, № 4, с. 95—131.

характеристики. В последнем случае элементы полей представляются в некоторой двоичной кодировке. Булевы схемы строятся из функциональных элементов с двумя входами и одним выходом, реализующих стандартные булевы операции AND (конъюнкция), NAND (отрицание конъюнкции), OR (дизъюнкция), NOR (отрицание дизъюнкции), XOR (исключающая дизъюнкция, или сумма по модулю два), XNOR (отрицание суммы по модулю два). *Глубина* схемы — это число функциональных элементов в длиннейшей цепи, связывающей входы схемы с её выходами. *Сложностью* схемы называется число функциональных элементов в ней. Последнее понятие очень близко к понятию *битовой сложности* вычисления, а само понятие схемы очень близко к понятию *неветвящейся программы*. Более подробные разъяснения основных понятий теории сложности булевых функций можно найти, например, в [25, 77, 138]. Минимизация сложности и глубины схем — одна из центральных и практически важных проблем в этой теории.

В приложениях часто используются, кроме логических схем без памяти, *схемы с памятью* (т. е. конечные автоматы). Синтезу таких схем, реализующих операции в конечных полях, посвящена обширная литература, требующая специального обзора. Здесь мы этой темы касаться не будем.

В теоретических работах по компьютерной арифметике вычисления иногда моделируются на *машинах Тьюринга*, принцип работы которых заключается в считывании и перезаписывании информации на ленте при помощи подвижной головки, функционирующей как автомат. Существует множество модификаций машин Тьюринга: многоленточные, с указателями, с памятью и пр. На практике машины Тьюринга в чистом виде не используются, поэтому мы эту тему затрагивать также не будем.

Кроме логических и автоматных схем, для реализации арифметики в конечных полях часто используются также компьютерные программы. Если написать такие программы без использования циклов и условных переходов, то фактически получатся так называемые неветвящиеся программы. Понятие неветвящейся программы можно формализовать, и тогда оно станет эквивалентным понятию схемы. Время работы таких программ грубо можно оценить сложностью соответствующей схемы. Для более точной его оценки надо учитывать то, что время выполнения разных операций компьютером существенно различается. Далее мы будем упоминать работы, посвящённые программной (компьютерной) реализации арифметических операций, хотя эта тема также требует специального обзора.

Поле порядка q далее обозначается $GF(q)$. Элементы поля $GF(q^n)$ можно представить как многочлены над $GF(q)$ степени не выше $n-1$. Если для представления элементов поля $GF(q^n)$ используется *стандартный базис*

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

(элемент $\alpha \in GF(q^n)$ называется *генератором* базиса B_α), то умножение в базисе B_α представляется как полиномиальное умножение по модулю неприводимого многочлена $g(x)$ над $GF(q)$, такого что $g(\alpha) = 0$. Если *сопряжённые*

элементы $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$ линейно независимы над $GF(q)$, то они образуют базис

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

который называется *нормальным базисом* с генератором α. (Основные понятия, связанные с конечными полями, можно найти, например, в [23,101].) Сложности умножения и инвертирования в поле $GF(q)$ обозначаются $\mathrm{M}\big(GF(q)\big)$ и $\mathrm{I}\big(GF(q)\big)$ соответственно. Мы также будем использовать обозначения $\mathrm{D_M}\big(GF(q)\big)$ и $\mathrm{D_I}\big(GF(q)\big)$ соответственно для глубины этих операций.

Подобные обозначения будут использоваться и в других случаях. Иногда мы будем представлять вычисления над простым полем $GF(p)$ в виде p-ичных, а не булевых схем (т. е. схем, построенных не из двоичных, а из p-ичных функциональных элементов). Для соответствующих понятий сложности и глубины тогда будем использовать те же обозначения, но снабжённые индексом p, например $\mathrm{M}^{(p)}\big(GF(q)\big)$ или $\mathrm{D}_{\mathrm{I}}^{(p)}\big(GF(q)\big)$.

1. Целочисленная арифметика

Для построения схем, реализующих операции в произвольных конечных полях $GF(p^n)$, в качестве строительных блоков обычно используются схемы, реализующие элементарные операции по модулю p. Эти операции в свою очередь сводятся к операциям сложения и умножения обычной целочисленной арифметики. Поэтому уместно начать с обсуждения вопросов схемной реализации операций целочисленной арифметики.

1.1. Сложение

Как это ни удивительно, весьма нетривиальной оказывается даже задача синтеза эффективных в том или ином смысле схем для операции сложения (вычитания). В книгах и статьях по компьютерной арифметике описано большое количество различных подобных схем. Здесь мы упомянем о результатах, представляющих теоретический интерес.

Минимальная сложность сумматоров n-разрядных чисел равна $\mathrm{A}(n) = 5n-3$. Схемы с такой сложностью строятся просто, но весьма нетривиальным является доказательство нижней оценки, полученное Н. П. Редькиным [28] (в [28] точно найдена также минимальная сложность сумматора, построенного только из элементов конъюнкции, дизъюнкции, отрицания).

Задача минимизации глубины сумматора оказалась нетривиальной уже на этапе конструирования схемы. В. М. Храпченко [34] построил схему для сумматора с глубиной

$$\log n + \sqrt{\big(2 + o(1)\big)\log n}$$

(здесь и далее log означает двоичный логарифм); сложность такой схемы может быть уменьшена до $\big(8 + o(1)\big)n$ [13]. Схема В. М. Храпченко асимптотически

оптимальна, но она начинает превосходить схемы глубины $2\log n+O(1)$, обычно используемые на практике, лишь для n, больших нескольких тысяч.

Некоторые методы построения эффективных (хотя и асимптотически неоптимальных) схем для сумматоров при $n \leqslant 1000$ (включая «троичный метод» М. И. Гринчука с глубиной сумматора $1{,}262\log n + 2{,}05$) указаны в [13].

Недавно М. И. Гринчук построил сумматор с глубиной, не большей $\log n + \log\log n + 6$ [19]. Этот сумматор является рекордным не только в асимптотическом смысле, но и при малых значениях n.

В. М. Храпченко [36] в 2007 г. получил первую нетривиальную оценку глубины сумматора (в базисе из конъюнкции, дизъюнкции, отрицания): она не может быть меньше, чем $\log n + \bigl(1 - o(1)\bigr)\log\log\log n$.

1.2. Умножение

Умножение чисел, очевидно, более сложная операция, чем сложение. Построению различных алгоритмов умножения посвящено огромное количество публикаций. Читатель может обратиться, например, к подробному обзору асимптотически быстрых алгоритмов умножения в [56], написанному с теоретической точки зрения. Практически эффективные алгоритмы описаны в многочисленных монографиях и статьях по компьютерной арифметике. Здесь мы только кратко коснёмся этой обширной темы.

Сложность минимальной схемы для умножения n-разрядных чисел обычно обозначается $\mathrm{M}(n)$. Далее для краткости вместо «схема для умножения» будем использовать термин «мультиплер». Стандартный (школьный) мультиплер имеет, как легко проверить, сложность $\mathrm{M}(n) \leqslant 6n^2 - 8n + O(1)$ (разумеется, для вывода указанной оценки следует использовать двоичную, а не десятичную версию школьного алгоритма умножения).

Менее очевидно, что стандартный мультиплер можно реализовать схемой асимптотически такой же сложности, но глубины $O(\log n)$ (это можно сделать методами, которые независимо предложили Г.К. Столяров [32], Ю.П. Офман [20], А. Авизиенис [43] и К. Уоллес [137]).

Минимизация глубины стандартного мультиплера — это практически важная проблема компьютерной арифметики. Существенные результаты в этой области были получены В. М. Храпченко [35]. Приём из работы [75] позволяет уточнить оценку сложности мультиплера до $5{,}5n^2 - 6{,}5n$.

Наилучшая текущая асимптотическая оценка минимальной глубины n-разрядного мультиплера — $4{,}44\log n + O(1)$ (см. [95, 123, 124]). Практически эффективный (также в смысле сложности) мультиплер, описанный в [29], имеет глубину $5\log n + 5$.

Метод уменьшения сложности умножения был впервые предложен в 1961 г. (в то время аспирантом мехмата МГУ) А. А. Карацубой [20] (интересная исто-

рия развития быстрых арифметических алгоритмов им описана в [21]). Рекурсивная оценка сложности мультиплера Карацубы имеет вид

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

Из неё можно вывести следующую верхнюю оценку при n, равном степени двойки[1]:

$$\mathrm{M}(n) \leqslant \frac{1463}{54} \cdot n^{\log 3} - 52n + 5.$$

Можно проверить, что мультиплер Карацубы становится лучше стандартного начиная с $n = 17$. Однако практическому применению мультиплера Карацубы в компьютерной арифметике препятствует тот факт, что его глубина (при очевидной реализации) равна $O(\log^2 n)$. Впрочем при более изощренной реализации глубину мультиплера Карацубы можно уменьшить до $O(\log n)$ [138]. А. В. Чашкиным в [37] предложена чуть лучшая конструкция[2], но мультипликативные константы в оценках сложности и глубины в ней слишком велики для практических приложений (глубину мультиплера Карацубы можно уменьшить до $\big(10 + o(1)\big) \log n$, как было показано в [29]).

В [33] А. Л. Тоом (в то время студент мехмата МГУ, занимавшийся под руководством О. Б. Лупанова) предложил мультиплер сложности $n^{1+o(1)}$. Точнее, оценка А. Л. Тоома имела вид $n2^{O(\sqrt{\log n})}$. Впоследствии константы в оценке А. Л. Тоома были уточнены, С. Кук в своей диссертации [73] перенёс его метод на машины Тьюринга, а А. Шёнхаге предложил модулярный метод умножения с подобной же оценкой сложности (обо всём этом можно прочитать в [22]).

Принципиальное улучшение результата А. Л. Тоома было достигнуто А. Шёнхаге и Ф. Штрассеном в [129] (см. также, например, [1, 22, 89]). Сложность мультиплера, построенного по их методу, равна $O(n \log n \log\log n)$, а глубина равна $O(\log n)$ (можно получить оценку $\big(9+o(1)\big) \log n$, как указано в [29]). В [129] отмечено, что этот метод с той же оценкой сложности переносится и на тьюринговы вычисления.

Наилучший известный алгоритм умножения найден в 2007 г. М. Фюрером [85]. Сложность построенного на его основе мультиплера равна

$$n \log n 2^{O(\log^* n)},$$

где чрезвычайно медленно растущая функция сверхлогарифм $\log^* n$ определяется из отношения

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

Однако глубина этого мультиплера равна $O(\log n \log^* n)$, что хуже, чем у А. Шёнхаге и Ф. Штрассена. В [74] дана модулярная версия алгоритма Фюрера.

[1] В [20] эта оценка дана в виде $\mathrm{M}(n) = O(n^{\log 3})$. Приведённые выше константы вычислены А. А. Бурцевым в дипломной работе.

[2] То, что это действительно так, было установлено в дипломной работе В. В. Баева.

Последние два алгоритма, вероятно, не могут иметь приложений (за пределами рекордных вычислений различных констант с миллиардами знаков) из-за больших мультипликативных констант в оценках сложности. Возможности ускорения алгоритма Шёнхаге—Штрассена (с целью его программной, а не схемной реализации) подробно рассмотрены в [93].

Алгоритм Полларда (см., например, [26, 27]), возможно, имеет больше шансов на практическое применение, однако, как заметил Я. В. Вегнер, схемная сложность мультиплера Полларда становится меньше, чем сложность мультиплера Карацубы только при $n > 2^{22}$ (можно получить оценку $30\,634n\log n + 393n$ для сложности и оценку $349\log n + 50$ для глубины мультиплера Полларда при $n < 201\,326\,604$).

Асимптотическая эффективность (и практическая неэффективность) всех упомянутых выше алгоритмов (кроме методов Тоома и Карацубы) обусловлена тем, что они основаны на многократном применении быстрого дискретного преобразования Фурье (иногда над полем комплексных чисел, иногда над кольцом вычетом по модулю чисел Ферма).

С практической точки зрения алгоритм Тоома выглядит более предпочтительно. Например, как заметил А. А. Бурцев, методом Тоома можно построить мультиплер с рекурсивной оценкой сложности

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

которая при $n = 4^s$, $s \geqslant 4$, влечёт оценку

$$\mathrm{M}(n) \leqslant 402{,}5n^{\log_4 7} - \frac{662}{3}n - \frac{1085}{6}.$$

В частности, из неё следует, что $\mathrm{M}(1024) \leqslant 1\,279\,651$. Мультиплер Карацубы имеет в этом случае примерно на 20 процентов большую сложность. Используя приём, указанный в [37], можно и методом Тоома построить мультиплер глубины $O(\log n)$, однако ценой увеличения мультипликативной константы в оценке сложности.

1.3. Деление

Школьный алгоритм деления $2n$-разрядного числа на n-разрядное, очевидно, позволяет построить схему, вычисляющую одновременно частное и остаток со сложностью $O(n^2)$ и глубиной $O(n\log n)$. В компьютерной арифметике известно много подобных схем, наилучшие из которых имеют при той же сложности глубину $O(n)$.

Эффективная реализация деления (с одновременной минимизацией глубины) выглядит более сложной проблемой, чем умножение, однако деление с помощью метода Ньютона—Рафсона можно свести к умножению.

Это было сделано в [73] (см. также, например, [18, 22]). Глубина построенной схемы равна $O(\log^2 n)$, а сложность асимптотически в пять раз больше

сложности умножения. Однако при малых n эта схема имеет бóльшую сложность, чем схемы, построенные на основе модификаций школьного алгоритма, и не намного меньшую глубину.

Метод [125] позволяет уменьшить глубину схемы для деления до $O(\log n \log\log n)$, сохраняя по порядку ту же сложность, которую имеет мультиплер с глубиной $O(\log n)$. Применение мультиплера Фюрера позволяет уменьшить сложность, но увеличивает глубину.

В [55] построена схема для деления глубины $O(\log n)$ и сложности $O(n^5)$. В [98] дан набросок построения схемы глубины $O(\varepsilon^{-2}\log n)$ и сложности $O(n^{1+\varepsilon})$ для любого заданного параметра ε.

Однако все перечисленные методы, кроме школьного и метода Ньютона, имеют, вероятно, только теоретический интерес.

1.4. Реализация арифметики простых полей

Арифметика в поле простого порядка p есть просто целочисленная арифметика по модулю p. Сложность умножения по произвольному (не обязательно простому) модулю p можно оценить как $3\mathrm{M}(\log p) + O(\log p)$. Для этого нужно выполнить обычное умножение и найти остаток от деления полученного $\lceil 2\log p\rceil$-разрядного результата на $\lceil \log p\rceil$-разрядное число p. Последняя операция выполняется с помощью так называемого алгоритма Баррета [52] (см. также, например, [18, 112]). Метод Баррета, возможно, появился под влиянием работ [73, 133]. Для специальных модулей, например для чисел вида $p = 2^n \pm c$, $c = O(\log n)$, приведённую выше оценку можно усилить до $\mathrm{M}(\log p) + O(\log p)$.

Сложность сумматора по модулю n-разрядного числа p, очевидно, оценивается как $2\mathrm{A}(n) + O(1)$. Для специальных модулей эту оценку тоже можно улучшить. Например, для модулей Мерсенна $p = 2^n - 1$ оценка улучшается до $\mathrm{A}\big(GF(p)\big) = 7n - 5$. Глубина при этом на $O(1)$ отличается от глубины обычного сумматора. Такая же оценка глубины справедлива для сложения по модулю Ферма $p = 2^n + 1$, сложность при этом оценивается как $\mathrm{A}\big(GF(p)\big) = 9n + O(1)$.

Умножение на константу вида 2^k в простом поле по модулю Мерсенна для любого целого k сводится к циклическому сдвигу битов, который реализуется схемой нулевой сложности. Сложность умножения на произвольную константу C, которая представляется в виде суммы $l(C)$ степеней двойки по модулю p, оценивается как $\mathrm{M}(C, p) \leqslant (l(C) - 1)\mathrm{A}\big(GF(p)\big)$. Например, $\mathrm{M}(17, p) \leqslant \leqslant \mathrm{A}\big(GF(p)\big)$.

Аналогично для сложности и глубины умножения на 2^k в простом поле по модулю Ферма получаются оценки

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

В общем случае умножения на константу C оценка сложности имеет вид

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

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

Для сложности и глубины стандартного мультиплера в поле Мерсенна можно получить оценки $6n^2 - n + O(1)$ и $4{,}44\log n + O(1)$ (с очень большой константой в оценке глубины). Для поля Ферма аналогичные оценки имеют вид $6n^2 + 11n + O(1)$ и также $4{,}44\log n + O(1)$.

2. Умножение в произвольных конечных полях

Пусть $\mathrm{M}_{q,f}(n)$ есть общее число операций над полем $GF(q)$ (или сложность над $GF(q)$), требующихся для умножения многочленов по модулю f, $\deg f = n$. Аналогичным образом обозначим $\mathrm{m}_{q,f}(n)$ мультипликативную сложность и $\mathrm{a}_{q,f}(n)$ — аддитивную сложность (т. е. число умножений и сложений-вычитаний в $GF(q)$ соответственно) такого модулярного умножения. Тогда

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

для любого неприводимого многочлена $f(x)$ над $GF(q)$. Более точно,

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

Также будет использоваться обозначение $\mathrm{M}_q(n)$ для сложности над $GF(q)$ умножения многочленов степени, меньшей n. Аналогично вводятся обозначения $\mathrm{m}_q(n)$ и $\mathrm{a}_q(n)$ для мультипликативной и аддитивной сложности.

Метод Штрассена [133] (см. также, например, [89]) приводит для любого f к оценке

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

В [18] описан другой алгоритм с такой же оценкой сложности, являющийся полиномиальным аналогом алгоритма Баррета (в той же мере, как и сам алгоритм Баррета является его числовым аналогом). Если $f(x)$ — сумма k мономов, тогда очевидно, что $\mathrm{M}_{q,f}(n) \leqslant \mathrm{M}_q(n)+(2k+1)n$, а если $q=2$, то $\mathrm{M}_{2,f}(n) \leqslant \mathrm{M}_2(n)+kn$. Хорошо известно, что неприводимые полиномы f с $k \leqslant 5$ мономами почти всегда существуют (вероятно, просто всегда существуют, но это, кажется, пока не доказано). Для таких полиномов, очевидно,

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

А. Шёнхаге в [128] (см. также, например, [89]) доказал, что $\mathrm{m}_q(n) = O(n\log n)$ одновременно с $\mathrm{a}_q(n) = O(n\log n\log\log n)$. В [69] получены более точные мультипликативные константы в этих оценках. Но оба метода выглядят малопригодными для применений в кодировании и криптографии, так как эти константы велики.

Известно (см., например, [27]), что в случае $2n-1 \leqslant q$ мультипликативная сложность над $GF(q)$ умножения в поле $GF(q^n)$ равна $2n-1$. Основная идея

доказательства верхней оценки принадлежит А. Л. Тоому [33], а нижняя оценка была доказана С. Виноградом (см., например, [2]).

Братья Чудновские доказали в [72], что мультипликативная сложность и в общем случае равна $O(n)$. Более точные оценки были получены С. Г. Влэдуцем, М. А. Цфасманом и И. Е. Шпарлинским в [131]. Однако, как выяснилось позднее, обе работы содержат неточности в обосновании. Корректное обоснование и уточнённые оценки мультипликативной сложности получены в серии статей С. Балле и соавторов (см. работу [51] и ссылки в ней). Но аддитивная сложность в этих методах не оценивалась, и, вероятно, она не так мала. По этой причине упомянутые алгоритмы, по-видимому, не имеют практических применений.

2.1. Полиномиальное умножение

Рассмотрим умножение многочленов над $GF(2)$ и оценим возможность применения асимптотически быстрых алгоритмов. Очевидно, сложность и глубина стандартного школьного мультиплера равна

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

При $n \approx 1000$ отсюда имеем, что $\mathrm{M}(n) \approx 2\,000\,000$, $\mathrm{D}(n) = 11$.

Из рекурсивных оценок сложности полиномиального мультиплера Карацубы

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

при $n = 2^k$, $k \geqslant 3$, вытекают следующие неравенства для сложности и глубины:

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

В частности, для $n = 1024$ мы имеем, что $\mathrm{M}(n) \leqslant 330\,725$, $\mathrm{D}(n) \leqslant 27$.

Используя уже упоминавшийся метод Шёнхаге [128], можно реализовать конволюцию многочленов степени, меньшей 2187, схемой сложности 428 351 и глубины 46 или же схемой сложности 430 537 и глубины 34 (конволюция многочленов степени, меньшей n, — это их произведение по модулю $x^n - 1$). Как следствие, имеем оценки

$$\mathrm{M}(1024) \leqslant \mathrm{M}(1093) \leqslant 430\,537, \quad \mathrm{D_M}(1024) \leqslant \mathrm{D_M}(1093) \leqslant 34.$$

Очевидно, что метод Карацубы в рассматриваемом случае лучше.

С другой стороны, адаптация метода Карацубы для вычисления конволюции многочленов степени, меньшей 2048, приводит к схеме сложности 998 216 и глубины 30. В этом случае метод Шёнхаге предпочтительнее.

Другой пример: умножение по модулю $x^{1458} + x^{729} + 1$ может быть схемно реализовано методом Шёнхаге со сложностью 273 850 и глубиной 33. В этом случае метод Карацубы опять выигрывает.

Таким образом, граница, начиная с которой мультиплер Шёнхаге имеет меньшую сложность в сравнении с мультиплером Карацубы, пролегает где-то за $n = 1000$.

Аккуратные оценки сложности схем для умножения двоичных многочленов малых степеней, основанных на методах Карацубы и Тоома, приведены в [57].

Для полиномиального умножения известен также метод Д. Кантора [68]. Асимптотически его сложность выше, чем у метода Шёнхаге (например, $O(n\log^{1,59} n)$ для умножения над $GF(2)$ и $O(n\log^2 n)$ над произвольным конечным полем), но для для некоторых полей среднего размера метод Кантора может быть предпочтительнее. В [88] приведена модификация метода Кантора и рассмотрены некоторые приложения к задачам полиномиальной факторизации. Улучшенная версия алгоритма построена Ш. Гао и Т. Матиром (см. [111]).

Метод Кантора в определённом смысле уточняет метод Тоома, а именно для интерполяции многочленов использует в качестве узлов элементы подходящего расширения $GF(2^n)$, лежащие в его аффинных подпространствах над полем $GF(2)$. При этом многочлен, корни которого являются этими узлами, оказывается так называемым *линеаризованным* многочленом и имеет мало ненулевых коэффициентов, благодаря чему существенно уменьшается сложность вычисления интерполяционного многочлена. Конечно, ещё выгоднее брать в качестве узлов корни двучлена $x^n - 1$. Указанный метод равносилен применению дискретного преобразования Фурье порядка n. Но для реализации дискретного преобразования Фурье нужно выполнить $O(n\log n)$ операций в наименьшем поле $GF(2^m)$, содержащем корни n-й степени из единицы. Очевидно, что $2^m > n$, поэтому даже сложность аддитивной операции в этом поле по порядку не меньше $\log n$, а сложность умножения не меньше $\log n(\log\log n)(\log\log\log n)$ (даже с учётом возможностей метода Шёнхаге). Поэтому такое применение преобразования Фурье для умножения многочленов степени, меньшей n, над полем $GF(2)$ имеет сложность по порядку не меньше $n\log^2 n(\log\log n)(\log\log\log n)$. Даже такой оценки этим способом достигнуть затруднительно, так как n должно делить 2^m-1, т. е. не может иметь вид 2^k, удобный для быстрого вычисления преобразования Фурье, и редко когда может иметь вид 3^k или, скажем, 5^k, также пригодный для быстрого преобразования Фурье. Всё же иногда этой оценки можно достичь, действуя, например, следующим образом [11]. Пусть $n = 2^{p-1}$, $q = 2^p - 1$ — простое число Мерсенна. Для умножения многочленов степени, меньшей n, достаточно их перемножить как многочлены с целыми коэффициентами, равными 0 или 1, а это можно сделать, перемножая их как многочлены над полем $GF(q)$ с помощью преобразования Фурье порядка $2n$. Корень такой степени из единицы можно найти в поле $GF(q^2)$, которое можно рассматривать как расширение поля $GF(q)$, порождённое корнем i многочлена x^2+1, неприводимого над $GF(q)$. В качестве подходящего корня степени $2n$ можно взять, как известно, $(2^{n/2} + 3^{n/2}i)^2$. В [11] показано, что преобразование Фурье порядка $2n$ можно вычислить при помощи $(3/2)n\log n + O(n)$ операций умножения и $6n\log n + O(n)$ операций сложения в поле $GF(q)$[1].

[1] В [11] неверно указана в доказательстве формула для примитивного корня восьмой степени из единицы. На самом деле $\varepsilon = 2^{-(p+1)/4}(1+i)$. Также допущена неточность в оценке числа сложений, которая для дискретного преобразования Фурье порядка n на самом деле имеет вид $3n\log n$.

Из [11, формула (3)] следует, что

$$\mathrm{m}_q(n) \leqslant \frac{9}{2} n \log n + O(n), \quad \mathrm{a}_q(n) \leqslant 18 n \log n + O(n),$$

откуда следует, что сложность умножения многочленов степени, меньшей n, в поле $GF(2)$ равна

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

Заметим, что мультипликативный множитель в этой оценке велик, так как он велик в оценке $\mathrm{M}(p)$, число n имеет специальный вид (для других n величина этого множителя ещё вырастет) и до сих пор неизвестно, бесконечно ли количество чисел Мерсенна. Если взять $p = 17$, $q = 2^{17} - 1$, то для $n \approx 2^{16}$ имеем оценку

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

которая почти совпадает со сложностью школьного метода. Значит, этот метод превзойдёт школьный лишь начиная с n порядка 70 000.

Для программной реализации перспективы у этого метода менее пессимистичны. Из результатов [11] вытекает, что для умножения многочленов степени, меньшей $n = 2^{p-1}$, в поле $GF(q)$, где $q = 2^p - 1$ — число Мерсенна, достаточно выполнить $(9/2)n \log n + 58n + 1$ операций в поле $GF(q^2)$ (в [11] по недосмотру указана несколько иная оценка). Если оперативная память машины в состоянии вместить таблицы умножения и сложения в этом поле (впрочем, достаточно иметь только таблицы умножения на корни n-й степени из 1, а их суммарный объём равен $(n-1)q^2 = q^3/2$, так как умножений общего вида только n штук и их можно выполнить, сведя к шести операциям по модулю q, иначе уже при $p = 7$ нужна память гигабайтного размера) и процедура извлечения результата из памяти выполняется не медленнее умножения по модулю $q = 127$, то указанный алгоритм для умножения полиномов 63-й степени требует приблизительно столько же времени, сколько и школьный. Однако для умножения полиномов большей степени нужно увеличивать размеры поля (и соответственно размер используемой памяти).

Прочитав всё это, читатель оценит трюк Шёнхаге [128], предложившего для умножения многочленов использовать преобразование Фурье не в поле, а в кольце полиномов с операциями по модулю многочлена $x^n + 1$ и получившего оценку $O(n \log n \log\log n)$. Интересно, что она остаётся до сих пор малоизвестной, так как авторы видели несколько статей, в которых получались более слабые или такие же результаты, но без всяких ссылок на [128]. В [10], например, вместо [128] излагается более поздняя работа с аналогичным результатом (а для быстрого деления используется преобразование Фурье вместо упоминавшегося выше трюка Штрассена, который, видимо, автору [10] также не знаком).

В заключение отметим, что подробное описание нескольких алгоритмов для программного умножения многочленов как в бинарном поле, так и в произвольном простом поле, в том числе и основанных на использовании методов Карацубы, Тоома, Шенхаге, Д. Кантора, можно найти в [5, 45, 64, 97, 112, 117]. В [83] предложен ещё один алгоритм программного умножения бинарных многочленов по модулю данного трёхчлена, который даёт алгоритм умножения в бинарном поле в полиномиальном базисе, отвечающем неприводимому трёхчлену. Этот алгоритм использует умножение *теплицевой матрицы* на вектор и несколько превосходит по скорости алгоритм, использующий метод Карацубы, как утверждают авторы (но оценка сложности по порядку остаётся такой же). Как соотносится разработанный ими алгоритм по скорости с алгоритмами, предложенными в [97], неясно, так как в [83] работа [97] вообще не упоминается (хотя авторы обеих статей работают в одном университете).

2.2. Умножение в стандартных базисах

Известно много разных архитектур мультиплеров для стандартных базисов (см., например, [81, 96, 110]). Обычно в них используются базисы, минимальные многочлены которых являются трёхчленами или пятичленами. Сложность и глубина таких мультиплеров (как, впрочем, и в случае произвольных стандартных базисов) оценивается как $O(n^2)$, $O(\log n)$ соответственно.

В [41] было отмечено, что иногда выгоднее использовать стандартные базисы с минимальными полиномами максимального веса, т. е. полиномами вида

$$1 + x + \ldots + x^{m-1} + x^{m+1} + \ldots + x^n.$$

В [76] предложено для неприводимого многочлена данной степени подбирать ближайший к нему по степени кратный ему трёхчлен и вместо умножения по модулю неприводимого многочлена выполнять умножение по модулю этого трёхчлена (вкладывая поле в кольцо многочленов по модулю этого трёхчлена). В [76] также даны соответствующие таблицы и утверждается, что иногда указанный приём эффективнее использования неприводимых пятичленов. Нужно выбирать трёхчлен так, чтобы его средний член имел степень, не меньшую половины степени трёхчлена. Это всегда можно сделать, так как таблицы содержат вместе с каждым трёхчленом взаимный к нему.

Подобная же идея, только с заменой трёхчлена на двучлен $x^n - 1$, предлагалась ранее в ряде работ по так называемым *редундантным базисам*. Ещё одна возможность ускорения модулярного умножения и модулярного экспоненцирования как чисел, так и многочленов связана с использованием *метода Монтгомери* (подробнее см. [5] и указанную там литературу).

Мультиплеры для стандартных базисов сложности $O(n^{\log 3})$ могут быть сконструированы методом Карацубы [118, 120].

Например, умножение в $GF(2^{1024})$ при выборе базиса, соответствующего неприводимому многочлену

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

может быть реализовано схемой с характеристиками

$$\mathrm{M}\big(GF(2^{1024})\big) \leqslant 356\,865, \quad \mathrm{D}_{\mathrm{M}}\big(GF(2^{1024})\big) \leqslant 31.$$

2.3. Умножение в нормальных базисах

Известно несколько алгоритмов умножения в нормальных базисах (см., например, [3,60,91,101,109,126]). Опишем кратко алгоритм Месси—Омуры. Пусть $T = (t_{i,j})$ — матрица, в которой i-я строка является вектором координат элемента $\alpha\alpha^{q^i} \in GF(q^n)$ при любом i в *нормальном базисе*

$$B^\alpha = \{\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}\}.$$

Число ненулевых элементов в матрице T называется *сложностью* $\mathrm{C}(B^\alpha)$ базиса B^α. Если

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

произвольные элементы поля $GF(q^n)$, то их произведение $\pi = \xi\zeta$ задаётся формулой

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

где $S^m(v)$ — циклический сдвиг координат данного вектора v на m позиций, $A(u,v)$ — билинейная форма, ассоциированная с матрицей $A = (a_{i,j})$ с элементами $a_{i,j} = t_{i-j,-j}$, где индексы $i-j$ и $-j$ вычисляются по модулю n. Данный алгоритм для умножения в произвольном нормальном базисе B поля $GF(q^n)$, предложенный в [109], требует $n(2C(B) + n - 1)$ операций над подполем $GF(q)$. В [126] предложен более эффективный алгоритм с оценкой сложности $n(C(B) + 3n - 2)/2$. Но обе эти оценки в лучшем случае квадратические и в худшем случае кубические. Для их улучшения можно использовать идею перехода от нормального базиса к стандартному и обратно, а для умножения в стандартном базисе использовать асимптотически быстрый алгоритм умножения многочленов и упоминавшийся выше алгоритм Штрассена быстрого приведения по модулю данного многочлена.

Преобразование координат в одном базисе в координаты в другом базисе является линейным преобразованием над подполем $GF(q)$ и может быть реализовано схемой сложности $O(n^2/\log_q n)$ и глубины $O(\log n)$, по существу, с помощью метода О. Б. Лупанова, опубликованного впервые в 1956 г. для случая так называемых вентильных схем [24, 25] (фактическое применение этого метода к умножению булевых матриц в [1] называется «алгоритмом четырёх русских», см. также [5]). В [30] построены схемы для перехода между произвольным нормальным и произвольным стандартным базисом и обратно, которые имеют сложность $O(n^{1,806})$ и глубину $O(\log n)$ (такая же оценка сложности перехода в одну

сторону другим методом была ранее получена в [102] с худшей оценкой глубины). Эти схемы перехода позволяют выполнять умножение в нормальных базисах поля $GF(q^n)$, используя $O(n^{1,806})$ операций над $GF(q)$ со схемной глубиной $O(\log n)$. С учётом нового способа реализации операции Фробениуса [103, 136] сложность метода [30] можно оценить как $O\bigl(n^{1,667}+\sqrt{n}(n\log q)^{1+o(1)}\bigr)$[1]. В [30] дана также другая конструкция для схем перехода, которая для произвольного нормального базиса B даёт схему над $GF(q)$ сложности

$$O(\sqrt{n}C(B)+n^{1,667}+n^{1,5}\log q\log n\log\log n)$$

и глубины

$$O(\sqrt{n}\log q\log n).$$

В частности, если B — *базис низкой сложности*, т. е. $C(B)=O(n^{1,167})$, и q достаточно мало, т. е. $\log q=o(n^{0,167})$, то $M^{(q)}\bigl(GF(q^n)\bigr)=O(n^{1,667})$. Но мультипликативные константы в этих оценках велики.

Для некоторых специальных нормальных базисов могут быть построены лучшие мультиплеры. Нормальный базис поля $GF(q^n)$ минимальной возможной сложности $2n-1$ называется *оптимальным нормальным базисом*. Все такие базисы найдены в [116]. Оптимальный нормальный базис типа I существует, только если $n+1=p$ — простое и q — первообразный корень по модулю p. Оптимальные нормальные базисы типов II, III существуют, только если $q=2^m$, $(m,n)=1$, $2n+1=p$ — простое и 2 — первообразный корень по модулю p (тип II) или n нечётно и -2 — первообразный корень по модулю p (тип III). Базис типа II или III порождается элементом $\alpha=\zeta+\zeta^{-1}$, где $\zeta\in GF(q^{2n})$, $\zeta^p=1$, $\zeta\ne1$, и с точностью до перестановки совпадает с базисом

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

Конструкции базисов типов II и III допускают обобщение для $q\ne2^m$, однако в этом случае указанные базисы имеют сложность больше $2n-1$, поэтому формально не являются оптимальными, но являются базисами сложности $O(n)$. В [42, 84, 91, 130] были найдены другие типы нормальных базисов сложности $\mathrm{C}(B)=O(n)$, в частности гауссовы нормальные базисы, которые являются обобщением оптимальных нормальных базисов. Используя метод [86], можно для гауссовых нормальных базисов произвольного типа k^2 построить мультиплер сложности

$$\mathrm{M}\bigl(GF(q^n)\bigr)\leqslant(\mathrm{M}_q(kn)+7kn-8)\mathrm{M}\bigl(GF(q)\bigr).$$

В частности, в случае $q=k=2$ (случая оптимального нормального базиса) последний результат был независимо получен в [60]. Для оптимального

[1]Неоднократно упоминаемая далее константа 1,667 происходит из быстрого метода умножения прямоугольных матриц [99].

[2]*Гауссов нормальный базис k-го типа* существует в поле $GF(q^n)$, если число $kn+1$ простое, и порождается элементом $\alpha=\zeta+\zeta^{\gamma}+\ldots+\zeta^{\gamma^{k-1}}$, где ζ — примитивный корень степени $kn+1$ в поле $GF(q^{kn})$, а γ — примитивный корень степени k в поле вычетов $\mathbb{Z}_{kn+1}$, который вместе с q порождает всю мультипликативную группу $\mathbb{Z}_{kn+1}\setminus\{0\}$.

нормального базиса типа I имеем, что

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

Для оптимальных нормальных базисов типов II, III в [3] было доказано, что

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

Конструкция соответствующих мультиплеров была основана на построенной в [3] схеме перехода от базиса $\{\alpha_1, \ldots, \alpha_n\}$ к базису $\{\alpha, \ldots, \alpha^n\}$, $\alpha = \alpha_1 = \zeta + \zeta^{-1}$, сложности $O(sn\log_s n)$, где $q = s^m$, s простое, и глубины $O(\log_s n)$. Множитель 3 в указанной оценке появился из-за упоминавшегося выше неравенства Штрассена

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

Это соотношение подразумевает, что сложность приведения по модулю минимального многочлена f стандартного базиса $B_\alpha = \{1, \ldots, \alpha^{n-1}\}$ оценивается как $2\mathrm{M}_q(n) + O(n)$.

В некоторых случаях этот многочлен содержит мало ненулевых коэффициентов и оценки сложности могут быть улучшены. Например, в [5] показано, что если $q = 2$, $n = 3 \cdot 2^k - 1$ и оптимальный нормальный базис типа II или III существует, то для этого базиса

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

В частности, используя для оценки $\mathrm{M}(191)$ метод Карацубы, имеем, что

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

Для сравнения, алгоритм [126] даёт оценку $\mathrm{M}\big(GF(2^{191})\big) \leqslant 90\,916$. Другая приведённая выше оценка

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

при $q = 2 = k$, $n = 191$ и использовании метода Карацубы даёт

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

В 2007 г. в [92] был переоткрыт упоминавшийся выше алгоритм перехода между базисами $\{\alpha_1, \ldots, \alpha_n\}$ и $\{\alpha, \ldots, \alpha^n\}$, $\alpha = \alpha_1 = \zeta + \zeta^{-1}$, найденный в [3], и с помощью него получена оценка

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

Улучшение было достигнуто за счёт более точной оценки сложности приведения результата умножения многочленов, представляющих элементы поля $GF(q^n)$, разложенные по базису B_α, по модулю минимального многочлена этого базиса. Вместо деления с остатком на этот многочлен, которое можно рассматривать

как линейное преобразование из пространства размерности $2n-1$ в пространство размерности n над полем $GF(q)$, в [92] было предложено воспользоваться линейным преобразованием между редундантными базисами $\{\alpha,\dots,\alpha^{2n}\}$ и $\{\alpha_1,\dots,\alpha_{2n}\}$, которое реализуется со сложностью $O(sn\log_s n)$, где $q=s^m$, и глубиной $O(\log_s n)$. Тот факт, что эти базисы не настоящие, а редундантные (т. е. эти системы элементов в поле $GF(q^n)$ линейно зависимы), никак не сказывается на справедливости предыдущих оценок. После указанного преобразования от координат в первом редундантном базисе к координатам во втором останется заметить, что второй редундантный базис в силу равенств

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

представляет собой «сдвоенный» базис $\{\alpha_1,\dots,\alpha_n\}$ и переход к координатам в последнем базисе делается со сложностью n и глубиной 1. Отсюда можно вывести оценки

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

Аналогичные оценки получены в [58].

3. Инвертирование в конечных полях

Наилучшая известная асимптотическая оценка сложности инвертирования (так для краткости далее называется операция вычисления мультипликативного обратного) в стандартном базисе поля $GF(q^n)$ над подполем $GF(q)$ равна $O(n\log^2 n\log\log n)$. Соответствующий алгоритм может быть получен с помощью быстрого расширенного алгоритма Евклида для вычисления наибольшего общего делителя.

Быстрый числовой вариант этого алгоритма принадлежит Д. Кнуту [105] и оптимизирован А. Шёнхаге [127] (впрочем, как отмечалось в литературе, алгоритм Кнута—Шёнхаге можно рассматривать как современную версию алгоритма Евклида—Лехмера для вычисления НОД больших целых чисел; алгоритм Лехмера можно найти в [22]). Полиномиальная версия алгоритма была предложена Р. Мунком [113] (см. также [1]), однако она оказалась в некоторых случаях некорректной. Корректные алгоритмы были построены в работах [65, 134] (см. также [89, гл. 11]).

Впоследствии было предложено ещё несколько модификаций алгоритма для НОД (см., например, [114, 132]). Алгоритм Штеле—Циммермана [132] обещает быть перспективным применительно к умножению многочленов над $GF(2)$. На практике эти алгоритмы используются только для компьютерных вычислений, потому что схемная реализация подобных алгоритмов затруднительна и приводит к схемам большой глубины (порядка n; для чисел известны схемы глубины $O(n/\log n)$ и сложности $O(n^{1+\varepsilon})$ [71]).

При вычислении НОД многочленов небольшой степени n, видимо, более эффективно работает известный бинарный алгоритм с оценкой сложности $O(n^2)$ (см., например, [5,132]). Однако опять же при переложении на схемы сложность увеличивается в несколько раз, а глубину удаётся оценить только как $O(n \log n)$. Поэтому для построения инверторов малой глубины используются совершенно другие алгоритмы.

3.1. Метод аддитивных цепочек

Аддитивной цепочкой, вычисляющей число n, называется любая последовательность натуральных чисел $a_0 = 1, a_1, \ldots, a_m = n$, в которой каждое число равно сумме двух встречающихся в ней ранее чисел (которые могут и совпадать, тогда говорят не о сложении, а об удвоении). Число m называется *длиной* цепочки. Длина кратчайшей цепочки для n обычно обозначается $\mathrm{l}(n)$. Подробное изложение известных результатов об аддитивных цепочках имеется в [22].

Мы будем использовать далее обозначение $\lambda(n) = \lfloor \log n \rfloor$. Известно, что

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

Верхняя оценка принадлежит А. Брауэру [63], а нижняя — П. Эрдёшу [82].

Очевидно, что вычисление n-й степени произвольной переменной с помощью одних только умножений соответствует вычислению числа n в аддитивной цепочке. Из тождества Ферма $x = x^{q^n}$ для любого $x \in GF(q^n)$ следует, что инвертирование в $GF(q^n)$ эквивалентно возведению в степень $q^n - 2$. Это даёт возможность использовать аддитивные цепочки для конструирования инверторов.

А. Брауэр [63] предложил метод построения аддитивной цепочки для $2^n - 1$, отправляясь от аддитивной цепочки для n (см. также [22]). Этот метод легко расширяется до вычисления $(q^n - 1)/(q - 1)$, где умножение на q используется вместо удвоения.

Положим $y = x^{(q^n - q)/(q-1)}$ и затем воспользуемся тождеством $x^{-1} = y(xy)^{-1}$ (вероятно, этот элегантный способ вычислений был впервые указан в [100]). Очевидно, что $xy \in GF(q)$, так как $(xy)^{q-1} = x^{q^n - 1} = 1$. Для вычисления $y = (x^{(q^{n-1} - 1)/(q-1)})^q$ можно использовать метод Брауэра или метод Ито—Цуйи [100] (последний на самом деле есть специальный случай метода Брауэра). В случае $q = 2$ справедливо $y = x^{-1}$. В общем случае для завершения вычислений требуется выполнить умножение x на y (которое выполняется проще, чем в общем случае, так как произведение принадлежит подполю) и деление на $xy \in GF(q)$.

Подход, основанный на несколько отличной формуле

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

и методе Брауэра, рассматривался в [90].

Обозначим $\mathrm{F}\big(GF(q^n)\big)$ и $\mathrm{D}_{\mathrm{F}}\big(GF(q^n)\big)$ максимальные по m сложность и глубину схемной реализации автоморфизма Фробениуса $x \to x^{q^m}$ в $GF(q^n)$, $m = 1, \ldots, n$. В стандартном представлении элементов поля операция Фробениуса состоит в вычислении многочлена $g^{q^m} \bmod f$ и может быть выполнена как модулярная композиция $g(h) \bmod f$, где $h = x^{q^m} \bmod f$. Действительно, если

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

то

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

Обозначим $\mathrm{d}(n)$ глубину кратчайшей аддитивной цепочки для n. Используя аддитивные цепочки и алгоритм Брента—Кунга [66] для модулярной композиции многочленов в [14] можно построить инвертор в стандартном базисе поля $GF(q^n)$ над подполем $GF(q)$ со сложностью и глубиной

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) \leqslant (\mathrm{l}(n-1)+1)\Big(\mathrm{M}^{(q)}\big(GF(q^n)\big) + \mathrm{F}^{(q)}\big(GF(q^n)\big)\Big) + n = O(n^{1{,}667}),$$

$$\mathrm{D}_{\mathrm{I}}^{(q)}\big(GF(q^n)\big) \leqslant (\mathrm{d}(n-1)+1)\Big(\mathrm{D}_{\mathrm{M}}^{(q)}\big(GF(q^n)\big) + \mathrm{D}_{\mathrm{F}}^{(q)}\big(GF(q^n)\big)\Big) + 1 = O(\log^2 n).$$

Инвертор для нормального базиса можно построить со сложностью и глубиной

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) \leqslant (\mathrm{l}(n-1)+1)\mathrm{M}^{(q)}\big(GF(q^n)\big) + n = O(n^{1{,}806}),$$

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

благодаря тому что любая операция Фробениуса в нормальном базисе есть просто циклический сдвиг координат элемента в этом базисе и поэтому имеет нулевую схемную сложность, а сложность умножения в любом нормальном базисе согласно [30] равна $O(n^{1{,}806})$ при глубине $O(\log n)$. Слагаемое n в обеих оценках сложности и 1 в обеих оценках глубины можно опустить в случае $q = 2$.

Сложность метода Брента—Кунга оценивается как $O(n^{1{,}667})$. В 2007 г. К. Уманс [136] показал, что модулярную композицию можно выполнить со сложностью $n^{1+o(1)}$, если поле $GF(q)$ имеет характеристику $n^{o(1)}$. (Утверждение в [103] о том, что оценка $n^{1+o(1)}$ справедлива и без ограничения на характеристику, по-видимому, не относится к реализации схемами.) Отсюда вытекает улучшение оценки для сложности инвертирования в стандартных базисах до

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) = n^{1+o(1)}$$

и в нормальных базисах до

$$\mathrm{I}^{(q)}\big(GF(q^n)\big) = O(n^{1{,}667})$$

в случае полей малой характеристики, в частности для двоичных полей. Однако при этом не получаются оценки для глубины $O(\log^2 n)$.

Упомянутые выше алгоритмы, основанные на методе Брауэра 1939 г., по-видимому, малоизвестны. Поэтому часто используются несколько худшие алгоритмы, предложенные в [100] или в [135] (например, алгоритм [135] уступает методу Брауэра при $n = 24, 44, 47, \ldots$). При использовании метода Брауэра могут быть улучшены некоторые недавно опубликованные алгоритмы, например оценки сложности [70] для инвертирования в полях $GF(2^{384})$, $GF(2^{480})$ (детали см. в [14]).

Для минимизации глубины инвертора можно использовать вариант бинарного метода построения аддитивных цепочек (см. [14, 22]). Этот метод позволяет построить цепочку для n с глубиной $\delta(n) = \lceil \log n \rceil$ и длиной $\lambda(n) + \nu(n) - 1$, где $\nu(n)$ — число единиц в двоичной записи n. Длина такой цепочки $2\lambda(n)$ в худшем случае.

В [14] с помощью модифицированного метода А. Яо [139] построена аддитивная цепочка для n с глубиной $\delta(n) + 1$ и асимптотически минимальной длиной

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

Используя последний результат, можно построить инвертор для стандартного базиса сложности

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

и глубины

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

Аналогичные оценки для нормального базиса имеют вид

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

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

Заметим, что для любого $n \leqslant 228$ существует цепочка минимальной длины с глубиной, не большей $\delta(n) + 1$. Для любого $n \leqslant 1024$ существует цепочка минимальной длины с глубиной, не превосходящей $\delta(n) + 2$.

3.2. Инверторы логарифмической глубины

Впервые инверторы логарифмической глубины были построены в работах [87,108] (в [108] рассмотрен только случай поля характеристики 2). В этих работах авторы оценили сложность и глубину построенных схем только как $n^{O(1)}$ и $O(\log n)$ соответственно. В действительности константы в этих оценках довольно велики. В [31] был построен инвертор для $GF(2^n)$ глубины $\big(6{,}44 + o(1)\big)\log n$ и сложности $(2/3)n^4 + o(n^4)$, причём этот результат верен

для произвольного базиса в данном поле. В то же время для стандартного базиса в [31] построен инвертор глубины $O(\log n)$ и сложности $O(n^{1,667})$. Последний результат переносится на случай произвольного поля $GF(q^n)$ в [15]. Как следствие, для нормального базиса может быть построен инвертор сложности $O(n^{1,806})$ и глубины $O(\log n)$.

Метод, описанный в [15], является параллельной версией метода аддитивных цепочек. Он основан на использовании схемы для многократного умножения. Сложность и глубина такой схемы умножения m элементов поля $GF(q^n)$ обозначается далее $\mathrm{MM}\big(m, GF(q^n)\big)$ и $\mathrm{D}_{\mathrm{MM}}\big(m, GF(q^n)\big)$ соответственно. В [15] с использованием идей из [80,98,125] построена схема для многократного умножения сложности

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

и глубины

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

где l — произвольный натуральный и ε — произвольный положительный параметры, c — некоторая константа.

Использование схемы для многократного умножения приводит к следующему результату [15, 31]. Пусть $m = \lceil \sqrt[r]{n} \rceil$, где $r \in \mathbb{N}$ — произвольный параметр. Тогда возведение в степень $(q^n - q)/(q-1)$ в поле $GF(q^n)$ может быть реализовано схемой сложности и глубины

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

$$2\Big(\mathrm{D_F}\big(GF(q^n)\big) + \mathrm{D_{MM}}\big(m, GF(q^n)\big)\Big) + \mathrm{D_M}\big(GF(q^n)\big) + \\ + (r-2)\max\big\{\mathrm{D_F}\big(GF(q^n)\big) + \mathrm{D_{MM}}\big(m, GF(q^n)\big), \mathrm{D_M}\big(GF(q^n)\big)\big\}$$

соответственно. Как было показано выше, после такого возведения в степень остаётся ещё две операции для вычисления инверсии. Тогда для любого $r \in \mathbb{N}$ можно построить инвертор в стандартном базисе со следующими оценками сложности и глубины:

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

где $w < 1{,}667$. Выбирая r достаточно большим, можно получить инвертор логарифмической глубины и сложности $O(n^{1,667})$.

Лучшие оценки для некоторых стандартных и нормальных базисов можно получить, используя быстрый переход между ними. Обозначим $\mathrm{T}\big(GF(q^n)\big)$ и $\mathrm{D_T}\big(GF(q^n)\big)$ сложность и глубину схемы перехода между стандартным и нормальным базисами (переход рассматривается в обоих направлениях). Тогда, пользуясь тем, что умножение можно быстро выполнять в стандартном базисе, а операцию Фробениуса — в нормальном базисе, можно получить следующие оценки для сложности и глубины инвертирования в любом из этих

базисов [15]:

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

где $b < 2{,}12$ и R — натуральный параметр, который или постоянен, или медленно растёт с ростом n. Поэтому если существуют схемы перехода с почти линейной сложностью и логарифмической глубиной, то в этом случае можно построить и инвертор с почти линейной сложностью и логарифмической глубиной.

В частности, для гауссова нормального базиса типа k поля $GF(q^n)$ при любом $k = o(\log n)$ и любом $\varepsilon > 0$ можно построить инвертор сложности $O(\varepsilon^{-b} n^{1+\varepsilon})$ и глубины $O(\varepsilon^{-1}\log n)$.

4. Реализация арифметики в композитных бинарных полях

Схемы для инвертирования, о которых шла речь в предыдущем пункте, имеют наименьшую возможную по порядку глубину, однако они превосходят схемы, построенные методом аддитивных цепочек, только для полей большой размерности ($n > 500$), но их сложность при этом (уже при $n \approx 100$) слишком велика для практического применения. Поэтому при n порядка нескольких сотен для минимизации глубины применяют различные разновидности метода аддитивных цепочек (обычно метод Ито—Цуйи). Некоторого уменьшения глубины можно добиться для полей характеристики 2 и составной размерности, если использовать не стандартные базисы, а базисы, возникающие из представления таких полей в виде башни полей. Существенно также, что при этом уменьшается сложность. По-видимому впервые идея использования композитных бинарных полей появилась в [100]. Реализации арифметики в композитных бинарных полях посвящены работы [38, 96, 115, 118, 120, 121, 126].

Комбинируя [3, 30], можно построить при взаимно простых n, m для некоторого нормального базиса мультиплер сложности

$$\mathrm{M}^{(q)}\big(GF(q^{nm})\big) = O\big(nm(m^{0{,}806} + n^{0{,}806})\big)$$

и глубины $O(\log nm)$. В частности, если $n = \Omega(m)$, то сложность равна

$$\mathrm{M}^{(q)}\big(GF(q^N)\big) = O(N^{1{,}403}), \quad N = nm,$$

а глубина равна $O(\log N)$. Применяя метод аддитивных цепочек, получаем что

$$\mathrm{I}^{(q)}\big(GF(q^N)\big) = O(N^{1{,}403}), \quad \mathrm{D}_{\mathrm{I}}^{(q)}\big(GF(q^N)\big) = O(\log^2 N).$$

Если N — ε-гладкое число, т. е. $N = n_1 \cdots n_m$, где все n_i попарно взаимно просты, $n_1 + \ldots + n_m = O(N^{\varepsilon})$, то $\mathrm{M}^{(q)}\big(GF(q^N)\big) = O(N^{1+0{,}806\varepsilon})$. Но оценки для глубины становятся существенно хуже.

4.1. Умножение и инвертирование в башнях полей

Башней называется последовательность вложенных друг в друга полей. В работе [38] (которая завершает цикл работ авторов с 1991 по 2002 г.) рассмотрена весьма общая конструкция башни полей, для умножения и инвертирования в которой приведены оценки сложности. В частности, утверждается, что для соответствующих полей $GF(2^n)$ сложность умножения оценивается как $O(n\log^2 n)$, и тем самым улучшается оценка сложности умножения, на которую можно рассчитывать, прямолинейным образом применяя преобразование Фурье для умножения в стандартном базисе. Вероятно, авторы [38] не знают о работе Шёнхаге [128], в которой получена лучшая оценка.

Башни, которые рассматриваются в [38], имеют вид

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

где каждое P_j может иметь только простые делители, общие с $q-1$, а также p — характеристику поля, причём P_j может быть чётным, только если $q=1 \bmod 4$. У таких башен на каждом этаже можно выбрать базис, минимальный многочлен которого является двучленом (подобные башни под названием оптимальных башен независимо рассматривались в [44]). Для умножения на каждом этаже башни используется полиномиальное умножение по модулю указанного двучлена. Для выполнения полиномиального умножения используется фактически метод Тоома [33] и метод преобразования Фурье, в котором требуемые корни из единицы выбираются из предыдущих этажей башни (та же идея применялась и в [8]), а для инвертирования фактически используется метод Ито—Цуйи. Полученные оценки сложности громоздки и поэтому здесь не приводятся, тем более что основные формулы [38, с. 227] вызывают сомнение, так как при их выводе предполагается, что умножение в поле K_j на элемент подполя K_{j-2} выполняется со сложностью, по порядку равной сложности сложения в K_j (на самом деле каждую из P_jP_{j-1} координат надо умножать на элемент из K_{j-2} и сложность будет оцениваться как $P_jP_{j-1}M(K_{j-2})$). В частном случае $P_j=p^j$, где $p\mid q-1$, $n=P_h$, в [38] фактически получены для сложности умножения и инвертирования оценки

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

Оценка для умножения хуже оценки Шёнхаге для мультиплеров в стандартном базисе, но зато в этом базисе можно построить инвертор по порядку той же сложности и не такой большой глубины, как инверторы, которые можно построить с помощью быстрого алгоритма Евклида.

В [8] показано, что для любого $\varepsilon>0$ и любого натурального $m>1$ в башне полей $GF(2^n)$, $n=m^s$, $s\geqslant s_\varepsilon$, можно выбрать базис, в котором выполняются соотношения

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

В частности, при $n=8\cdot 3^k$ доказано, что

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

В [8] для $n = 2 \cdot 3^k$ получены также асимптотические оценки

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

Все указанные выше утверждения из [8] могут быть усилены за счёт более точного оценивания сложности умножения на константы при выполнении преобразования Фурье. А именно, можно построить мультиплер и инвертор в башне полей $GF(2^n)$, $n = m^s$, $s \geqslant s_m$, имеющие оценки для сложности

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

при подходящем выборе базиса, при этом получаются следующие оценки для глубины:

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

Иногда указанные оценки можно уточнить (см. [16, 17]). Например, если $m = p$ простое, 2 — первообразный корень по модулю p (это в точности условие существования оптимального нормального базиса первого типа в поле $GF(2^{p-1})$) и $2^{p-1} - 1$ не кратно p^2 (известно, что последнее условие заведомо выполняется при $p < 10^{12}$), то при выборе подходящего базиса в башне полей $GF(2^n)$, $n = (p-1)p^s$, имеем

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

В частности, при $p = 3$

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

Соответствующий метод умножения фактически является модификацией метода Шёнхаге умножения двоичных многочленов [128].

В [119] для башни полей $GF(2^n)$, $n = 2^k$, при специальном выборе базиса на каждом её этаже построены мультиплер и инвертор сложности

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

соответственно. В частности, получаются оценки

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

Глубина инвертора (как установил в дипломной работе В. Волынин) равна $\Omega(\log^3 n)$.

Если в поле $GF(2^4)$ выбрать оптимальный нормальный базис $\{\xi, \xi^2, \xi^4, \xi^8\}$ и на каждом этаже башни выбирать элемент $\alpha_k \in GF(2^{2^{k+2}})$, удовлетворяющий соотношениям

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

(подобные базисы предлагались при $k = 1$ в [67] для минимизации размера S-блоков в схемной реализации стандарта шифрования AES), и в качестве базиса взять на этом этаже либо стандартный базис $\{1, \alpha_k\}$, либо нормальный базис $\{\alpha_k, \alpha_k^{2^{k+1}}\}$, то получаются худшие оценки

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

однако (как показал в дипломной работе С. Зикрин) при $n \leqslant 64$ можно построить лучшие мультиплеры и инверторы, чем в [118]. Например, построены схемы с оценками сложности и глубины

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

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

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

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

Для сравнения приведём оценки сложности и глубины конструкций, предложенных в [118]:

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

Заметим, что если выбрать в поле $GF(2^{128})$ стандартный базис с неприводимым многочленом $x^{128} + x^7 + x^2 + x + 1$, то для него

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

Сложность и глубина схемы умножения, построенной методом Карацубы, оцениваются как 12 343 и 18 соответственно. Однако инвертирование в стандартном базисе имеет сложность порядка 200 000 и глубину не менее 200.

4.2. Минимизация глубины инверторов в некоторых композитных полях

В этом разделе мы кратко рассмотрим, следуя [6,12], некоторые рекурсивные методы построения инверторов в композитных бинарных полях, минимизирующие глубину в диапазоне значений n в пределах первых нескольких сотен.

Пусть n нечётно, $\mathrm{D_M}\bigl(GF(2^n)\bigr) \geqslant \mathrm{D_S}\bigl(GF(2^n)\bigr) + 1$, где $\mathrm{S}\bigl(GF(2^n)\bigr)$ — сложность операции возведения в квадрат в поле $GF(2^n)$. Применяя идею из [115], можно построить инвертор и мультиплер в некотором базисе поля $GF(2^{2n})$ со следующими оценками сложности и глубины:

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

Пусть $(n, 3) = 1$, $B_2 = \{\alpha, \alpha^2, \alpha^4\}$ — оптимальный нормальный базис в поле $GF(2^3)$, где $\alpha^3 = \alpha^2 + 1$, и B_1 — произвольный базис в поле $GF(2^n)$ Пусть также $\mathrm{D_M}\bigl(GF(2^n)\bigr) \geqslant \mathrm{D_S}\bigl(GF(2^n)\bigr) + 2$. Тогда для умножения в базисе $B = B_1 \otimes B_2$, который является произведением базисов B_i, можно построить схему, сложность и глубина которой удовлетворяют неравенствам

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

Для инвертирования в базисе B можно построить схему, удовлетворяющую следующим неравенствам:

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

Если B_1 — нормальный базис, то $\mathrm{S}\bigl(GF(2^n)\bigr) = 0$.

Если в башне полей $GF\Bigl(\bigl((2^n)^2\bigr)^2\Bigr)$ выбраны оптимальный нормальный базис $\{\alpha_1, \alpha_1^2\}$ и стандартный базис $\{1, \alpha_2\}$, где $\alpha_1^2 + \alpha_1 = 1$, $\alpha_2^2 + \alpha_2 = \alpha_1$, то можно построить мультиплер, сложность и глубина которого удовлетворяют неравенствам

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

Если выбрать в подполе $GF(2^n)$ нормальный базис, то можно построить инвертор со следующими оценками для сложности и глубины:

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

Пусть $(n, 5) = 1$, $B_2 = \{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}\}$ — оптимальный нормальный базис в поле $GF(2^5)$, где $\alpha^5 = \alpha^4 + \alpha^2 + \alpha + 1$, B_1 — произвольный нормальный базис в поле $GF(2^n)$ и $B = B_1 \otimes B_2$ — произведение этих базисов. Тогда для умножения в базисе B можно построить такой мультиплер, что

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

и такой инвертор, что

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

Поле $GF(2^{6n})$ можно рассматривать как расширение шестой степени поля $GF(2^n)$. Выберем в поле $GF(2^6)$ оптимальный нормальный базис $B_2 = \{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32}\}$, где $\alpha^6 = \alpha^5 + \alpha^4 + \alpha + 1$. Для любого базиса B_1 в поле $GF(2^n)$ построим произведение базисов $B = B_1 \otimes B_2$, которое является базисом поля $GF(2^{6n})$. Пусть $\mathrm{D_M}\big(GF(2^n)\big) \geqslant \mathrm{D_S}\big(GF(2^n)\big) + 2$. Тогда для умножения и инвертирования в базисе B можно построить схемы, для которых

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

Пусть $(n, 2) = 1$, $B_1 = \{\alpha_1, \alpha_1^2\} \otimes \{1, \alpha_2\}$, где $\alpha_1^2 + \alpha_1 = 1$, $\alpha_2^2 + \alpha_2 = \alpha_1$, $B_2 = B_1 \otimes \{1, \alpha_3\}$, $\alpha_3^2 + \alpha_3 = \alpha_1\alpha_2$, B — произвольный базис в $GF(2^n)$. Тогда для базиса $B_2 \otimes B$ в поле $GF(2^{8n})$

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

Если B — нормальный базис, то

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

Приведём другую конструкцию мультиплера и инвертора в поле $GF(2^{8n})$. Пусть в $GF(2^4)$ выбран оптимальный нормальный базис $B_1 = \{\alpha, \alpha^2, \alpha^4, \alpha^8\}$, где $\alpha^4 = \alpha^3 + \alpha^2 + \alpha + 1$, и в $GF(2^8)$ выбран базис $B_2 = B_1 \otimes \{1, \beta\}$, где $\beta^2 + \beta = \alpha$. Для произвольного нормального базиса B_3 в поле $GF(2^n)$ рассмотрим произведение базисов $B = B_2 \otimes B_3$. Тогда для базиса B в поле $GF(2^{8n})$ можно построить такие мультиплер и инвертор, что

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

Пусть $(n, 30) = 1$. Тогда в поле $GF(2^{30n})$ можно выбрать нормальный базис и построить для него мультиплер и инвертор, для которых

$$\mathrm{M}\big(GF(2^{30n})\big) \leqslant 315\mathrm{M}\big(GF(2^n)\big) + 1140n,$$
$$\mathrm{D_M}\big(GF(2^{30n})\big) \leqslant \mathrm{D_M}\big(GF(2^n)\big) + 8,$$
$$\mathrm{I}\big(GF(2^{30n})\big) \leqslant \mathrm{I}\big(GF(2^n)\big) + 566\mathrm{M}\big(GF(2^n)\big) + 1537n,$$
$$\mathrm{D_I}\big(GF(2^{30n})\big) \leqslant 6\mathrm{D_M}\big(GF(2^n)\big) + 17 + \\ + \max\big\{\mathrm{D_I}\big(GF(2^n)\big) + \max\big\{\mathrm{D_M}\big(GF(2^n)\big), 6\big\}, \mathrm{D_M}\big(GF(2^n)\big) + 8\big\}.$$

В таблице 1 приведены оценки для сложности и глубины наилучших известных авторам инверторов в некоторых бинарных полях. Эти оценки получены методами, упомянутыми выше.

Таблица 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

5. Реализация арифметики в псевдо-мерсенновских полях

Простое число p вида $2^n \pm c$, где c сравнительно мало, называется псевдо-мерсенновским простым. Реализация арифметики в мерсенновских полях была рассмотрена выше. В [44, 48] предложена техника, позволяющая строить эффективные мультиплеры и инверторы в псевдо-мерсенновских полях $GF(p^n)$, $n = 2^k, 3^k$, с целью применения в криптографии на эллиптических и гиперэллиптических кривых (см. [47]). Для этого используются специальные базисы (названные в [44, 48] базисами оптимальных башен). Эти базисы являются частными случаями базисов, рассмотренных в [38].

5.1. Умножение в оптимальных башнях псевдо-мерсенновских полей

В [48] (с улучшением результатов [44]) были построены алгоритмы умножения, позволяющие конструировать мультиплеры сложности

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

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

где $x^2-\alpha_0$, $x^3-\alpha_0$ — неприводимые двучлены над $GF(q)$, $\alpha_0\in GF(q)$, $\mathrm{M}(\alpha_0,q)$ — сложность умножения на константу α_0 в поле $GF(q)$. Как следствие, получились оценки

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

В [49] были предложены улучшения этих алгоритмов, основанные на применении быстрого преобразования Фурье в поле по модулю числа Ферма $q=2^{16}+1$.

Независимо подобные же результаты были получены в [3], а именно для $q=p^n$, $p=2^{16}+1$, были доказаны следующие оценки:

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

С использованием схемы для вычисления конволюции по модулю $x\bigl(x^{2^{k+1}}-1\bigr)/(x^2-1)$ в [9] были получены также оценки

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

Конструкция последней схемы основана на использовании примитивного корня $\sqrt{2}=2^4(2^8-1)$ порядка 64 из единицы в поле $GF(p)$. Как следует из теоремы Винограда (см., например, [2]), мультипликативные константы перед $\mathrm{M}\bigl(GF(q)\bigr)$ в указанных выше оценках минимальные возможные.

В [9] также доказаны при $q=p^n$, $p=2^{13}-1$, $n=2^{k_0}\cdot 3^{k_1}\cdot 5^{k_2}\cdot 7^{k_3}\cdot 13^{k_4}$, где $k_0=0,1$, неравенства

$$\mathrm{M}\bigl(GF(q^7)\bigr)\leqslant 13\mathrm{M}\bigl(GF(q)\bigr)+344\mathrm{A}\bigl(GF(q)\bigr)+6\mathrm{A}\bigl(GF(p)\bigr),$$
$$\mathrm{M}\bigl(GF(q^{13})\bigr)\leqslant 26\mathrm{M}\bigl(GF(q)\bigr)+1026\mathrm{A}\bigl(GF(q)\bigr)+12\mathrm{A}\bigl(GF(p)\bigr)$$

и аналогичные оценки при $q=p^n$, $p=2^{17}-1$, $n=2^{k_0}\cdot 3^{k_1}\cdot 5^{k_2}\cdot 17^{k_3}$, где $k_0=0,1$:

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

Эти результаты получены с использованием преобразования Фурье по модулю простого числа Мерсенна p, соответствующего примитивному корню ± 2 порядка p или $2p$. В последнем случае преобразование Фурье вычисляется методом Гуда—Томаса (см., например, [2]). Умножение в $GF(q^n)$ выполняется через трёхкратное применение преобразования Фурье и приведение по модулю неприводимого двучлена.

В [50] показано, как можно при последовательном выполнении достаточно большого числа операций в $GF(q^n)$ выполнять в среднем два преобразования Фурье на одно умножение. Этот метод авторы [50] называют модулярным умножением в частотной области, поскольку все операции выполняются над фурье-образами. Для ускорения модулярного умножения используется метод Монтгомери. Показано, например, что если двучлен $x^n - 2$ неприводим над $GF(p)$, $p = 2^m - 1$, $2n - 1 \leqslant m$, то сложность модулярного умножения в частотной области равна

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

В случае $2n - 1 < 2m$ получается оценка сложности

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

В [46] указаны применения этих алгоритмов в криптографии на эллиптических и гиперэллиптических кривых.

Заметим, что в этих алгоритмах вместо умножения Монтгомери можно использовать обычное модулярное умножение. Это упрощает понимание алгоритма, но увеличивает его сложность за счёт приблизительно $2m^2$ умножений на константы вида 2^k в частотной области. Такое усложнение имеет некоторое значение при программной реализации, но несущественно для схемной реализации.

6. Умножение в некоторых полях малой характеристики

В последние годы большое количество публикаций в области криптографии с открытым ключом посвящено так называемой криптографии, основанной на спариваниях (см., например, [4]). Основной практической проблемой здесь является эффективная реализация спариваний. В [54] предложен эффективный алгоритм для реализации спаривания Тейта для суперсингулярных кривых над полями характеристики 3. Скорость этого алгоритма зависит от эффективной реализации арифметики в полях $GF(3^n)$, различные подходы к которой развиты в [59, 94, 104, 122].

В [78, 79] предложен быстрый алгоритм для спаривания Тейта, частично переносимый и на гиперэллиптические кривые $y^2 = x^p - x + d$, $d = \pm 1$, над полем $GF(p^n)$. В случае $p = 3$ этот алгоритм превосходит [54]. В [106] предложены некоторые усовершенствования алгоритма Дуурсма—Ли (DL-алгоритма) для случая бинарных полей. Подобные усовершенствования возможны и в указанном выше общем случае (см. [4]). Другие улучшения DL-алгоритма описаны в [53].

Для реализации DL-алгоритма в общем случае нужно реализовать арифметику в полях $GF(p^{2pn})$, $(2p, n) = 1$, $p = 4k + 3$.

Для этих полей можно построить мультиплер, для которого

$$\mathrm{M}\bigl(GF(p^{2pn})\bigr) \leqslant (6p-3)\mathrm{M}\bigl(GF(p^n)\bigr) + O\Bigl(p^2 n M\bigl(GF(p)\bigr)\Bigr).$$

Вероятно, практический интерес представляет только поле $GF(7^{14n})$, соответствующее случаю $p = 7$. Быстрые схемы для арифметики в таких полях могли бы способствовать, например, эффективной реализации алгоритма [107]. Одним из шагов в этом направлении является работа [7], в которой для поля $GF(7^{14n})$ построен мультиплер с оценками сложности и глубины

$$\mathrm{M}\bigl(GF(7^{14n})\bigr) \leqslant 13\mathrm{M}\bigl(GF(7^{2n})\bigr) + 258n\mathrm{A}\bigl(GF(7)\bigr),$$
$$\mathrm{D}_{\mathrm{M}}\bigl(GF(7^{14n})\bigr) \leqslant 11\mathrm{D}_{\mathrm{A}}\bigl(GF(7)\bigr) + \mathrm{D}_{\mathrm{M}}\bigl(GF(7^{2n})\bigr).$$

Например, в конкретном случае $n = 31$ указанная конструкция приводит к оценке

$$\mathrm{M}\bigl(GF(7^{14\cdot 31})\bigr) \leqslant 698\,554.$$

Работа выполнена при финансовой поддержке РФФИ, проекты 11-01-00508 и 11-01-00792, и программы фундаментальных исследований Отделения математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики и информационные системы нового поколения» (проект «Задачи оптимального синтеза управляющих систем»).

Литература

[1] Ахо А., Хопкрофт Д., Ульман Д. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979.

[2] Блейхут Р. Э. Быстрые алгоритмы цифровой обработки сигналов. — М.: Мир, 1989.

[3] Болотов А. А., Гашков С. Б. О быстром умножении в нормальных базисах конечных полей // Дискрет. мат. — 2001. — Т. 13, № 3. — С. 3—31.

[4] Болотов А. А., Гашков С. Б., Фролов А. Б. Элементарное введение в эллиптическую криптографию: криптографические протоколы на эллиптических кривых. — М.: КомКнига, 2006.

[5] Болотов А. А., Гашков С. Б., Фролов А. Б., Часовских А. А. Элементарное введение в эллиптическую криптографию: алгебраические и алгоритмические основы. — М.: КомКнига, 2006.

[6] Бурцев А. А. О схемах для умножения и инвертирования в композитных полях // Чебышёвский сб. — 2006. — Т. 7, № 1 (17). — С. 172—185.

[7] Бурцев А. А. О сложности булевых схем для умножения в конечных полях нечётной характеристики // Материалы VI молодёжной научной школы по дискретной математике и её приложениям (Москва, ИПМ РАН, апрель 2007). Т. I. — 2007. — С. 13—16.

[8] Бурцев А. А., Гашков И. Б., Гашков С. Б. О сложности булевых схем для арифметики в некоторых башнях конечных полей // Вестн. Моск. ун-та. Сер. 1. Математика, механика. — 2006. — № 5. — С. 10—16.

[9] Бурцев А. А., Гашков С. Б. О схемах для арифметики в композитных полях большой характеристики // Чебышёвский сб. — 2006. — Т. 7, № 1 (17). — С. 186—204.

[10] Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. — М.: Изд-во МЦНМО, 2007.

[11] Гашков С. Б. Замечания о быстром умножении многочленов, преобразовании Фурье и Хартли // Дискрет. мат. — 2000. — Т. 12, № 3. — С. 124—153.

[12] Гашков С. Б., Хохлов Р. А. О глубине логических схем для операций в полях $GF(2^n)$ // Чебышёвский сб. — 2003. — Т. 4, № 4 (8). — С. 59—71.

[13] Гашков С. Б., Гринчук М. И., Сергеев И. С. О построении схем сумматоров малой глубины // Дискретный анализ и исследование операций. Сер. 1. — 2007. — Т. 14, № 1. — С. 27—44.

[14] Гашков С. Б., Сергеев И. С. О применении метода аддитивных цепочек к инвертированию в конечных полях // Дискрет. мат. — 2006. — Т. 18, № 4. — С. 56—72.

[15] Гашков С. Б., Сергеев И. С. О построении схем логарифмической глубины для инвертирования в конечных полях // Дискрет. мат. — 2008. — Т. 20, № 4. — С. 8—28.

[16] Гашков С. Б., Сергеев И. С. О сложности и глубине схем для умножения и инвертирования в некоторых полях $GF(2^n)$ // Вестн. Моск. ун-та. Сер. 1. Математика, механика. — 2009. — № 4. — С. 3—7.

[17] Гашков С. Б., Сергеев И. С. О сложности умножения и инвертирования в некоторых кольцах многочленов // Материалы XI Междунар. семинара «Дискретная математика и её приложения» (Москва, июнь 2012).

[18] Гашков С. Б., Чубариков В. Н. Арифметика. Алгоритмы. Сложность вычислений. — М.: Наука, 1996.

[19] Гринчук М. И. Уточнение верхней оценки глубины сумматора и компаратора // Дискретный анализ и исследование операций. Сер. 1. — 2008. — Т. 15, № 2. — С. 12—22.

[20] Карацуба А. А., Офман Ю. П. Умножение многозначных чисел на автоматах // ДАН СССР. — 1962. — Т. 145 (2). — С. 293—294.

[21] Карацуба А. А. Сложность вычислений // Тр. Мат. ин-та РАН. — 1995. — Т. 211. — С. 1—17.

[22] Кнут Д. Искусство программирования. Т. 2. Получисленные алгоритмы. — М.: Вильямс, 2004.

[23] Лидл Р., Нидеррайтер Х. Конечные поля. — М.: Мир, 1988.

[24] Лупанов О. Б. О вентильных и контактно-вентильных схемах // ДАН СССР. — 1956. — Т. 111, № 6. — С. 1171—1174.

[25] Лупанов О. Б. Асимптотические оценки сложности управляющих систем. — М.: Изд-во Моск. ун-та, 1984.

[26] Маклеллан Дж. Х., Редер Ч. М. Применение теории чисел в цифровой обработке сигналов. — М.: Радио и связь, 1983.

[27] Ноден П., Китте К. Алгебраическая алгоритмика. — М.: Мир, 1999.

[28] Редькин Н. П. О минимальной реализации двоичного сумматора // Проблемы кибернетики. — 1981. — Вып. 38. — С. 181—216.

[29] Сергеев И. С. О глубине схем для многократного сложения и умножения чисел // Материалы VI молодёжной научной школы по дискретной математике и её приложениям (Москва, ИПМ РАН, апрель 2007). Т. II. — 2007. — С. 40—45.

[30] Сергеев И. С. О построении схем для перехода между полиномиальными и нормальными базисами конечных полей // Дискрет. мат. — 2007. — Т. 19, № 3. — С. 89—101.

[31] Сергеев И. С. Об инвертировании в конечных полях характеристики 2 с логарифмической глубиной // Вестн. Моск. ун-та. Сер. 1. Математика, механика. — 2007. — № 1. — С. 28—33.

[32] Столяров Г. К. Способ параллельного умножения в цифровых вычислительных машинах и устройство для осуществления способа: Авт. свид-во кл. 42, т. 14, № 126668. — 1960.

[33] Тоом А. Л. О сложности схемы из функциональных элементов, реализующей умножение целых чисел // ДАН СССР. — 1963. — Т. 150. — С. 496—498.

[34] Храпченко В. М. Об асимптотической оценке времени сложения параллельного сумматора // Проблемы кибернетики. — 1967. — Вып. 19. — С. 107—120.

[35] Храпченко В. М. Некоторые оценки для времени умножения // Проблемы кибернетики. — 1978. — Вып. 33. — С. 221—227.

[36] Храпченко В. М. Об одной из возможностей уточнения оценок для задержки параллельного сумматора // Дискретный анализ и исследование операций. Сер. 1. — 2007. — Т. 14, №1. — С. 87—93.

[37] Чашкин А. В. Быстрое умножение и сложение целых чисел // Дискретная математика и её приложения. Т. II. — М.: Изд-во ЦПИ при мех.-мат. ф-те МГУ, 2001. — С. 91—110.

[38] Afanassiev V. B., Davydov A. A. Finite field tower: iterated presentation and comlpexity of arithmetic // Finite Fields Appl. — 2002. — Vol. 8. — P. 216—232.

[39] Agnew G. B., Beth T., Mullin R. C., Vanstone S. A. Arithmetic operations in $GF(2^m)$ // J. Cryptol. — 1993. — Vol. 6. — P. 3—13.

[40] Agnew G. B., Mullin R. C., Onyszchuk I. M., Vanstone S. A. An implementation for a fast public-key cryptosystem // J. Cryptol. — 1991. — Vol. 3. — P. 63—79.

[41] Ahmadi O., Menezes A. Irreducible polynomials of maximum weight: Preprint. — 2005.

[42] Ash D. W., Blake I. F., Vanstone S. A. Low complexity normal bases // Discrete Appl. Math. — 1989. — Vol. 25. — P. 191—210.

[43] Avizienis A. Signed-digit number representation for fast parallel arithmetic // IEEE Trans. Electron. Comput. — 1961. — Vol. 10. — P. 389—400.

[44] Bailey D. V., Paar C. Efficient arithmetic in finite fields extensions with application in elliptic curve cryptography // J. Cryptol. — 2001. — Vol. 14. — P. 153—176.

[45] Bajard J.-C., Imbert L., Plantard T. Modular number systems: Beyond the Mersenne family // SAC'04: 11th Int. Workshop on Selected Areas in Cryptography. — 2004. — P. 159—169.

[46] Baktir S., Kumar S., Paar C., Sunar B. A state-of-the-art elliptic curve cryptographic processor operating in the frequency domain // Mobile Networks Appl. — 2007. — Vol. 12, no. 4. — P. 259—270.

[47] Baktir S., Pelzl J., Wollinger T., Sunar B., Paar C. Optimal tower fields for hyperelliptic curve cryptosystems // Proc. IEEE 38th ACSSC. — 2004.

[48] Baktir S., Sunar B. Optimal tower fields // IEEE Trans. Comput. — 2004. — Vol. 53. — P. 1231—1243.

[49] Baktir S., Sunar B. Achieving efficient polynomial multiplication in Fermat fields using fast Fourier transform // Proc. ACMSE'06. — ACM Press, 2006. — P. 549—554.

[50] Baktir S., Sunar B. Frequency domain finite field arithmetic for elliptic curve cryptography // Proc. ISCIS 2006. — Berlin: Springer, 2006. — (Lect. Notes Comput. Sci.; Vol. 4263). — P. 991—1001.

[51] Ballet S., Chaumine J., Pieltant J., Rolland R. On the tensor rank of multiplication in finite extensions of finite fields. — 2011. — arXiv:1107.1184.

[52] Barret P. D. Implementing the Rivest, Shamir and Adleman public key encryption algorithm on a standard digital signal processor // Advances in Cryptology. Proc. Crypto-86. — Berlin: Springer, 1987. — (Lect. Notes Comput. Sci.; Vol. 263). — P. 311—323.

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

[54] Barreto P. S. L. M., Kim H. Y., Lynn B., Scott M. Efficient algorithms for pairing-based cryptosystems // Proc. Crypto-2002. — Berlin: Springer, 2002. — (Lect. Notes Comput. Sci.; Vol. 2442). — P. 354—368.

[55] Beame P., Cook S., Hoover H. Log depth circuits for division and related problems // SIAM J. Comput. — 1986. — Vol. 15, no. 4. — P. 994—1003.

[56] Bernstein D. J. Multidigit Multiplication for Mathematicians. — 2004. — http://cr.yp.to/papers.html#m3.

[57] Bernstein D. J. Batch binary Edwards // Advances in Cryptology — CRYPTO 2009. 29th Annual Int. Cryptology Conf. Santa Barbara, CA, USA, August 16—20, 2009. Proceedings / S. Halevi, ed. — Berlin: Springer, 2009. — (Lect. Notes Comput. Sci.; Vol. 5677). — P. 317—336.

[58] Bernstein D. J., Lange T. Type-II optimal polynomial bases // Arithmetic of Finite Fields. Third International Workshop, WAIFI 2010. Istanbul, Turkey, June 27—30, 2010. Proceedings. — Berlin: Springer, 2010. — (Lect. Notes Comput. Sci.; Vol. 6087). — P. 41—61.

[59] Bertoni G., Guajardo J., Kumar S., Orlando G., Paar C., Wolinger T. Efficient $GF(p^m)$ arithmetic architectures for cryptographic applications // CT-RSA'03 Proc. of the 2003 RSA Conf. on the Cryptographers' Track. — Berlin: Springer, 2003. — (Lect. Notes Comput. Sci.; Vol. 2612). — P. 158—175.

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

[61] Blake I., Seroussi G., Smart N. Elliptic Curves in Cryptography. — Cambridge: Cambridge Univ. Press, 1999.

[62] Blake I., Seroussi G., Smart N. Advances in elliptic curve cryptography. — Cambridge: Cambridge Univ. Press, 2005.

[63] Brauer A. On addition chains // Bull. Am. Math. Soc. — 1939. — Vol. 45. — P. 736—739.

[64] Brent R. P., Gaudry P., Thome E., Zimmerman P. Faster multiplication in $GF(2)[x]$: Preprint INRIA no. 6359. — 2007.

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

[66] Brent R., Kung H. Fast algorithms for manipulating formal power series // J. ACM. — 1978. — Vol. 25. — P. 581—595.

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

[68] Cantor D. On arithmetic algorithms over finite fields // J. Combin. Theory Ser. A. — 1989. — Vol. 50. — P. 285—300.

[69] Cantor D., Kaltofen E. On fast multiplication of polynomials over arbitrary algebras // Acta Inform. — 1991. — Vol. 28. — P. 693—701.

[70] Chang K., Kim H., Kang J., Cho H. An extension of TYT algorithm for $GF\big((2^n)^m\big)$ using precomputation // Inform. Process. Lett. — 2004. — Vol. 92. — P. 231—234.

[71] Chor B., Goldreich O. An improved parallel algorithm for integer GCD // Algorithmica. — 1990. — Vol. 5. — P. 1—10.

[72] Chudnovsky D. V., Chudnovsky G. V. Algebraic complexities and algebraic curves over finite fields // J. Complexity. — 1988. — Vol. 4. — P. 285—316.

[73] Cook S. On the minimum computation time of functions: Ph. D. Thesis. — Harvard Univ., 1966.

[74] De A., Kurur P. P., Saha C., Saptharishi R. Fast integer multiplication using modular arithmetic // Proc. 40th ACM Symp. on Theory of Computing. — 2008. — P. 499—506.

[75] Demenkov E., Kojevnikov A., Kulikov A. S., Yaroslavtsev G. New upper bounds on the Boolean circuit complexity of symmetric functions // Inform. Process. Lett. — 2010. — Vol. 110 (7). — P. 264—267.

[76] Doche C. Redundant trinomials for finite fields of characteristic 2 // ACISP 2005. — Berlin: Springer, 2005. — (Lect. Notes Comput. Sci.; Vol. 3574). — P. 122—133.

[77] Dunne P. E. The Complexity of Boolean Networks. — London: Academic Press, 1988.

[78] Duursma I., Lee H.-S. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$ // Proc. Asiacrypt-2003. — Berlin: Springer, 2003. — (Lect. Notes Comput. Sci.; Vol. 2894). — P. 111—123.

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

[80] Eberly W. Very fast parallel polynomial arithmetic // SIAM J. Comput. — 1989. — Vol. 18. — P. 955—976.

[81] Erdem S., Yanik T., Koc C. Polynomial basis multiplication over $GF(2^n)$ // Acta Appl. Math. — 2006. — Vol. 93. — P. 33—55.

[82] Erdős P. Remarks on number theory. III: On addition chains // Acta Arith. — 1960. — Vol. 6. — P. 77—81.

[83] Fan H., Hasan M. A. Alternative to the Karatsuba algorithm for software implementation of $GF(2^n)$ multiplication // Information Security. — 2009. — Vol. 3, no. 2. — P. 60—65.

[84] Feisel S., von zur Gathen J., Shokrollahi M. A. Normal bases via general Gauss periods // Math. Comput. — 1999. — Vol. 68, no. 225. — P. 271—290.

[85] Fürer M. Faster integer multiplication // Proc. 39th ACM STOC 2007 Conf. — P. 57—66.

[86] Gao S., von zur Gathen J., Panario D. Gauss periods and fast exponentiation in finite fields // Proc. Latin'95 (Valparaiso, Chile). — Berlin: Springer, 1995. — (Lect. Notes Comput. Sci.; Vol. 911). — P. 311—322.

[87] Von zur Gathen J. Inversion in finite fields // J. Symbolic Comput. — 1990. — Vol. 9. — P. 175—183.

[88] Von zur Gathen J., Gerhard J. Arithmetic and factorization of polynomials over $GF(2)$ // Proc. ISSAC'96 (Zürich). — 1996. — P. 1—9.

[89] Von zur Gathen J., Gerhard J. Modern Computer Algebra. — Cambridge: Cambridge Univ. Press, 1999.

[90] Von zur Gathen J., Nöcker M. Exponentiation in finite fields: theory and practice // Applied Algebra. Proc. AAECC-12. — Berlin: Springer, 1997. — (Lect. Notes Comput. Sci.; Vol. 1255). — P. 88—113.

[91] Von zur Gathen J., Nöcker M. Fast arithmetic with general Gauss periods // Theor. Comput. Sci. — 2004. — Vol. 315. — P. 419—452.

[92] Von zur Gathen J., Shokrollahy M. A., Shokrollahy J. Efficient multiplication using type 2 optimal normal bases // Arithmetic of Finite Fields. First International Workshop, WAIFI 2007. Madrid, Spain, June 21—22, 2007. Proceedings / C. Carlet, B. Sunar, eds. — Berlin: Springer, 2007. — (Lect. Notes Comput. Sci.; Vol. 4547). — P. 55—68.

[93] Gaudry P., Kruppa A., Zimmermann P. A GMP-based implementation of Schönhage—Strassen's large integer multiplication algorithm // ISSAC'07. Waterloo, Ontario, Canada, 2007.

[94] Granger R., Page D., Stam M. Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three // IEEE Trans. Comput. — 2005. — Vol. 54, no. 7. — P. 852—860.

[95] Grove E. Proofs with potential: Ph. D. Thesis. — U. C. Berkeley, 1993.

[96] Guajardo J., Güneysu T., Kumar S., Paar C., Pelzl J. Efficient hardware implementation of finite fields with application to cryptography // Acta Appl. Math. — 2006. — Vol. 93. — P. 75—118.

[97] Hankerson D., López J. H., Menezes A. Software implementation of elliptic curve cryptography over binary fields // CHES 2000. — Berlin: Springer, 2000. — (Lect. Notes Comput. Sci.; Vol. 1965). — P. 1—23.

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

[99] Huang X., Pan V. Fast rectangular matrix multiplication and applications // J. Complexity. — 1998. — Vol. 14. — P. 257—299.

[100] Itoh T., Tsujii S. A fast algorithm for computing multiplicative inverses in $GF(2^n)$ using normal bases // Inform. and Comput. — 1988. — Vol. 78. — P. 171—177.

[101] Jungnickel D. Finite Fields. Structure and Arithmetic. — Mannheim: Wissenschaftsverlag, 1993.

[102] Kaltofen E., Shoup V. Subquadratic-time factoring of polynomials over finite fields // Math. Comput. — 1998. — Vol. 67, no. 223. — P. 1179—1197.

[103] Kedlaya K., Umans C. Fast modular composition in any characteristic // Proc. 49th IEEE Symp. on Foundations of Comput. Sci. (FOCS). — 2008. — P. 146—155.

[104] Kerins T., Marnane W. P., Popovici E. M., Barreto P. S. L. M. Efficient hardware for Tate pairing calculation in characteristic three // Proc. CHES-2005. — Berlin: Springer, 2005. — (Lect. Notes Comput. Sci.; Vol. 3659). — P. 412.

[105] Knuth D. The analysis of algorithms // Proc. Int. Congress of Math. (Nice, France). Vol. 3. — 1970. — P. 269—274.

[106] Kwon S. Efficient Tate pairing computation for supersingular elliptic curves over binary fields. — Cryptology ePrint Archive. Report 2004/303. — http://eprint.iacr.org/2004/303.

[107] Lee E., Lee H.-S., Lee Y. 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.

[108] Litow B. E., Davida G. I. $O(\log n)$ parallel time finite field inversion // VLSI Algorithms and Architectures. — Berlin: Springer, 1988. — (Lect. Notes Comput. Sci.; Vol. 319). — P. 74—80.

[109] Massey J. L., Omura J. K. Apparatus for finite fields computation: US patent 4587627. — 1986.

[110] Mastrovito E. D. VLSI architectures for computation in Galois fields: Ph. D. Thesis. — Linköping Univ., 1991.

[111] Mateer T. Fast Fourier algorithms with applications: Ph. D. Thesis. — Clemson Univ., 2008.

[112] Menezes A. J., van Oorshot P. C., Vanstone S. A. Handbook of Applied Cryptography. — Boca Raton: CRC Press, 1997.

[113] Moenck R. Fast computation of GCDs // Proc. 5th Ann. ACM Symp. on Theory of Computing. — 1973. — P. 142—151.

[114] Möller N. On Schönhhage's algorithm and subquadratic integer gcd computation // Math. Comput. — 2008. — Vol. 77. — P. 589—607.

[115] Morii M., Kasahara M. Efficient construction of gate circuit for computing multiplicative inverses in $GF(2^n)$ // Trans. IEICE. — 1989. — Vol. 72, no. 1. — P. 37—42.

[116] Mullin R. C., Onyszchuk I. M., Vanstone S. A., Wilson R. M. Optimal normal bases in $GF(p^n)$ // Discrete Appl. Math. — 1988/1989. — Vol. 22. — P. 149—161.

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

[118] Paar C. Effective VLSI architectures for bit paralel computation in Galois fields: Ph. D. Thesis. — Universität GH Essen, 1994.

[119] Paar C., Fan J. L. Efficient Inversion in Tower Fields of Characteristic Two. — Ulm: ISIT, 1997.

[120] Paar C., Fleischmann P., Roelse P. Effective multiplier architectures for Galois fields $GF(2^{4n})$ // IEEE Trans. Comput. — 1998. — Vol. 47, no. 2. — P. 162—170.

[121] Paar C., Fleischmann P., Soria-Rodriges P. Fast arithmetic for public-key algorithms in Galois fields with composite exponents // IEEE Trans. Comput. — 1999. — Vol. 48, no. 10. — P. 1025—1034.

[122] Page D., Smart N. P. Hardware implementation of finite fields of characteristic three // Proc. CHES-2003. — P. 529—539.

[123] Paterson M., Pippenger N., Zwick U. Optimal carry save networks // Boolean Function Complexity. — Cambridge: Cambridge Univ. Press, 1992. — (London Math. Soc. Lect. Note Ser.; Vol. 169). — P. 174—201.

[124] Paterson M., Zwick U. Shallow circuits and concise formulae for multiple addition and multiplication // Comput. Complexity. — 1993. — Vol. 3. — P. 262—291.

[125] Reif J., Tate S. Optimal size integer division circuits // SIAM J. Comput. — 1990. — Vol. 19, no. 5. — P. 912—925.

[126] Reyhani-Masoleh A., Hasan M. A. On effective normal basis multiplication // Proc. IndiaCRYPT-2000. — Berlin: Springer, 2000. — (Lect. Notes Comput. Sci.; Vol. 1977). — P. 213—224.

[127] Schönhage A. Schnelle Berechnung von Kettenbruchentwicklungen // Acta Inform. — 1971. — Vol. 1. — P. 139—144.

[128] Schönhage A. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2 // Acta Inform. — 1977. — Vol. 7. — P. 395—398.

[129] Schönhage A., Strassen V. Schnelle Multiplikation großer Zahlen // Computing. — 1971. — Vol. 7. — P. 271—282.

[130] Seguin J. E. Low complexity normal bases // Discrete Appl. Math. — 1990. — Vol. 28. — P. 309—312.

[131] Shparlinski I. E., Tsfasman M. A., Vladuts S. G. Curves with many points and multiplication in finite fields // Coding Theory and Algebraic Geometry. — Berlin: Springer, 1992. — (Lect. Notes Math.; Vol. 1518). — P. 145—169.

[132] Stehlé D., Zimmermann P. A binary recursive GCD algorithm // Proc. ANTS-VI (Burlington, USA, 2004). — Berlin: Springer, 2004. — (Lect. Notes Comput. Sci.; Vol. 3076). — P. 411—425.

[133] Strassen V. Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten // Numer. Math. — 1973. — Vol. 20. — P. 238—251.

[134] Strassen V. The computational complexity of continued fractions // SIAM J. Comput. — 1983. — Vol. 12. — P. 1—27.

[135] Takagi N., Yoshiki J., Takagi K. A fast algorithm for multiplicative inversion in $GF(2^n)$ using normal basis // IEEE Trans. Comput. — 2005. — Vol. 50, no. 5. — P. 394—398.

[136] Umans C. Fast polynomial factorization and modular composition in small characteristic // Proc. 40th Symp. on Theory of Computing (STOC). — 2008. — P. 481—490.

[137] Wallace C. S. A suggestion for a fast multiplier // IEEE Trans. Electron. Comput. — 1964. — Vol. 13. — P. 14—17.

[138] Wegener I. The Complexity of Boolean Functions. — Stuttgart: Wiley, 1987.

[139] Yao A. C. On the evaluation of powers // SIAM J. Comput. — 1976. — Vol. 5. — P. 100—103.