МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА

МЕХАНИКО–МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

На правах рукописи
УДК 519.7

Сергеев Игорь Сергеевич

О РЕАЛИЗАЦИИ НЕКОТОРЫХ ОПЕРАЦИЙ В КОНЕЧНЫХ ПОЛЯХ СХЕМАМИ ЛОГАРИФМИЧЕСКОЙ ГЛУБИНЫ

01.01.09 — дискретная математика и математическая кибернетика

ДИССЕРТАЦИЯ

на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук,
профессор С. Б. Гашков

МОСКВА — 2007

Содержание

1 Введение

1.1 История вопроса

Схема из функциональных элементов является, по существу, параллельной моделью вычислений — ее быстродействие определяется задержкой в цепочке между входом и выходом, а не суммарным временем выполнения всех элементарных операций, как в последовательной модели. В теоретических работах анализ задержки обычно заменяется анализом глубины, когда все элементы имеют единичную задержку и игнорируется задержка на участках цепей между элементами. Но даже в этих допущениях, как показал В. М. Храпченко [32], глубина (максимальное число элементов в цепочке схемы) может существенно отличаться от задержки в ее физическом смысле (минимальное время, исчисляемое в единичных задержках, необходимое для установления окончательного результата на выходах схемы).

Интерес к оптимизации глубины схем для выполнения арифметических операций рос по мере развития схемотехники и уже в 60-е годы отразился в классических работах [25, 14, 31] о реализации сложения и умножения чисел. В целом же, вопросы сложности всегда имели приоритет над вопросами глубины ввиду распространения последовательных моделей вычисления, какими, например, являются компьютерные программы (из теории можно в качестве примера привести одноленточные машины Тьюринга).

Сложность, хоть она явным образом и не влияет на быстродействие схемы, все же является важной характеристикой, определяющей площадь (объем) и, как следствие, потребляемую мощность,[1] которые в реальности накладывают серьезные ограничения на способ синтеза. Поэтому если такая постановка задачи, как оптимизация сложности без учета глубины, выглядит естественной (при использовании последовательной модели вычислений, для которой глубина безразлична), то при оптимизации по глубине практически обусловленной является сопутствующая оптимизация (или хотя бы анализ) сложности.

Теоретически оптимизация по глубине с учетом сложности рассматривается в двух основных постановках: (1) заданную арифметическую операцию реализовать схемой с возможно меньшим порядком глубины и наилучшим известным порядком сложности, либо (2) построить схему с возможно меньшим порядком сложности и наилучшим известным порядком глубины. Первой постановке следует, например, обзор [40, гл. 4], а второй — рабо-

[1] Хотя, как выяснил О. М. Касим-Заде, связь между сложностью и мощностью не настолько однозначна, см., например, [15].

та [64]. В ряде случаев наилучшие известные решения совпадают для обеих постановок (как, например, для умножения). Отметим, что для операции, которая реализуется схемой с логарифмическим (относительно числа входов) порядком глубины, вторая постановка может быть переформулирована более конкретно как минимизация сложности схемы с логарифмической глубиной.

В рамках последней постановки в настоящей работе рассматриваются арифметические операции в конечных полях: умножение, инвертирование, преобразование координат при переходе между нормальными и стандартными базисами и некоторые другие.

Теория конечных полей была заложена в работах Ферма, Эйлера, Лежандра, Гаусса, Галуа, Диксона и других выдающих ученых, и до последней четверти 20-го века развивалась как область чистой математики, но в связи с развитием криптографии, как отмечается в [5], к настоящему времени превратилась едва ли не в прикладной раздел. Сегодня вопросам эффективной реализации арифметики в конечных полях посвящено несколько специальных книг (в основном зарубежных, см. [5]).

Особенность вычислений в конечном поле состоит в необходимости выбора представления элементов — от него существенно зависит способ реализации (и, как следствие, сложность и глубина схемы). При работе с числами или многочленами этот вопрос как будто бы не стоит. Несмотря на то, что потенциально возможны (и описаны в теоретических работах) разные представления, практически используются два, стандартное и нормальное, а также производные от них.

Наиболее универсальным является стандартное представление — элементы поля в нем представляются многочленами, основные арифметические операции с которыми реализуются достаточно просто.

Умножение многочленов выполняется аналогами числовых методов, наиболее известные из которых были разработаны А. А. Карацубой [14], А. Л. Тоомом [29] (уточнен Куком [48]), Шёнхаге и Штрассеном [84] в 60-е годы. На последнем методе достигаются одновременно наилучшие по порядку известные оценки глубины $O(\log n)$ и сложности $O(n \log n \log\log n)$, где n — разрядность сомножителей.[2]

Иначе дело обстоит с делением (или инвертированием, т.к. деление сводится к инвертированию и умножению). Асимптотически быстрые алго-

[2]Недавно Фюрер [55] показал, что умножение чисел можно выполнять со сложностью $n2^{O(\log^* n)} \log n$ и глубиной $O(\log n \log^* n)$, где $\log^* n$ определяется из соотношения

$$\lfloor \underbrace{\log_2 \ldots \log_2}_{\log^* n} n \rfloor = 1.$$

ритмы деления чисел[3] основаны на методе Кука [48] и имеют такую же по порядку сложность, как и умножение. Однако логарифмический порядок глубины на этих методах не достигается — наилучшая известная оценка глубины таких схем имеет вид $O(\log n \log\log n)$ [79]. Для сложности схем с глубиной $O(\log n)$ известна оценка $O(n^{1+\epsilon})$ из работы [64].

Упомянутые методы деления переносятся на степенные ряды, но не приложимы прямо к делению в конечном поле (когда деление производится по модулю неприводимого многочлена, результат вычисляется точно). Быстрый способ деления (инвертирования) в конечном поле состоит в применении алгоритма Евклида — наилучшая известная для него оценка сложности, $O(n \log^2 n \log\log n)$, достигается в методе Кнута—Шёнхаге [81]. Для глубины соответствующей схемы можно указать оценку $O(n)$. Схема сложности $O(n^w \log n)$, где $w < 1,667$ — экспонента умножения матриц размера $\sqrt{n} \times \sqrt{n}$ и $\sqrt{n} \times n$, может быть построена методом аддитивных цепочек (приведенная оценка вытекает из работ [42, 44, 67]). Глубина в этом методе оценивается как $O(\log^2 n)$.

Схемы логарифмической глубины для инвертирования в конечном поле впервые были построены в работах [73, 57] в конце 80-х годов. Сложность этих схем оценивалась авторами как $n^{O(1)}$, а глубина — как $O(\log n)$. Анализ показывает, что для сложности и глубины предложенных схем нельзя привести лучшие оценки, чем $O(n^5)$ и $15 \log_2 n$ соответственно. Улучшение этого результата являлось стимулом для настоящей работы.

В нормальном представлении конечного поля можно быстро выполнять возведение в степень определенного вида, однако другие основные операции (прежде всего, умножение) в специально разработанной для нормальных базисов технике выполняются существенно сложнее, чем в стандартных базисах (речь идет об общем случае, поскольку на практике используются конкретные базисы, в которых необходимые операции реализуются эффективно). В последнее время (например, [70, 4]) была высказана идея о том, что для ускорения реализации многих операций в нормальном представлении, а также некоторых операций в стандартном, целесообразно (как с практической точки зрения, так и для получения теоретических оценок) использовать переходы между нормальными и стандартными базисами. Оценки вида $O(n^\alpha)$, $\alpha < 2$, для сложности перехода в общем случае, по-видимому, до сих пор не были известны. Получение таких оценок также являлось стимулом для данной работы.

[3] Задача деления двух чисел состоит в нахождении частного с некоторой точностью (обычно с той же, с которой заданы делимое и делитель).

1.2 Краткое содержание работы

Основные результаты диссертации заключаются в следующем:

1. Получена новая верхняя оценка сложности инвертирования в стандартном базисе конечного поля при реализации схемами логарифмической глубины из функциональных элементов.

2. Получена новая верхняя оценка схемной глубины инвертирования в конечном поле характеристики два.

3. Получены новые верхние оценки сложности перехода между стандартными и нормальными базисами конечных полей в общем случае, в том числе, для реализации схемами логарифмической глубины. Как следствие, получены новые верхние оценки сложности умножения в нормальном базисе, проверки базисности нормальной системы в стандартном базисе и некоторых других операций в конечных полях.

Изложение построено следующим образом.

В **главе 2** даются основные определения и сведения вспомогательного характера.

В **главе 3** рассматриваются различные подходы к построению схем m-кратного умножения по модулю многочлена степени n над конечным полем с глубиной $O(\log(mn))$. Целью является минимизация порядка сложности таких схем относительно n. Основной результат главы формулируется следующим образом.

Теорема 3.4 *Пусть $j, l, r \in \mathbb{N}$, $j \leq \lceil \log_2 \log_2 m \rceil$. Тогда m-кратное умножение многочленов над $GF(q)$ по модулю многочлена степени n выполняется схемой $M_{m,n}$ сложности и глубины*

$$L(M_{m,n}) = O\left(la^j m^{1+\frac{1}{r}\left(3-\frac{1}{2^j}+\frac{2,5}{l2^j}\right)} n^{1+\frac{1}{l4^j}} \left(2^{-j}\log(mn)\log\log(mn) + l^2\right)\right),$$

$$D(M_{m,n}) = O\left((l+j)\log m + r(1+l/2^j)\log n\right),$$

где $a = 81$, если q четно, и $a = 8$, иначе.

Оценки теоремы 3.4 используются при выводе основных результатов о схемной реализации инвертирования.

В **главе 4** изучается вопрос о построении схем для инвертирования в конечном поле $GF(q^n)$ с глубиной $O(\log n)$.

Предлагается способ построения «параллельной» схемы инвертирования, состоящей из подсхем, реализующих умножения, многократные умножения и операции Фробениуса (возведения в степень вида q^k) в поле $GF(q^n)$. Доказана

Теорема 4.3 *Пусть $r \in \mathbb{N}$. Тогда инвертирование в стандартном базисе*

поля $GF(q^n)$ реализуется схемой I_n сложности и глубины

$$L(I_n) = O(rn^{1/r}(n^w + n^{1,5}\log n \log\log n)), \qquad D(I_n) = O(r\log n),$$

где w — экспонента умножения матриц размера $\sqrt{n}\times\sqrt{n}$ и $\sqrt{n}\times n$.

В частности (т.к. $w < 1,667$), можно построить схему для инвертирования сложности $O(n^{1,667})$ и глубины $O(\log n)$.

Следующая теорема позволяет получать лучшие оценки сложности для стандартных или нормальных базисов, имеющих низкую транзитивную сложность (т. е. сложность перехода в паре стандартный—нормальный базис).

Теорема 4.4 *Пусть $R \in \mathbb{N}$, $R = o(\log n/\log\log n)$. Пусть схемы T' и T'' реализуют соответственно прямой и обратный переходы между нормальным и стандартным базисами поля $GF(q^n)$. Тогда для инвертирования в любом из указанных базисов можно построить схему I_n сложности и глубины*

$$L(I_n) = O(R^b n^{1+2/R}) + O(R\sqrt[R]{n})(L(T') + L(T'));$$

$$D(I_n) = O(R(\log n + D(T') + D(T''))),$$

где $b = (4/3)\log_2 3$, если q четно, и $b = 1$, если q нечетно.

В качестве примера можно рассмотреть гауссовы нормальные базисы (ГНБ).

Утверждение 4.2 *Пусть $k = o(\log n)$ и $\epsilon > 0$, $\epsilon = \Omega(\log\log n/\log n)$. Тогда можно построить схему инвертирования в ГНБ k-го типа поля $GF(q^n)$ сложности $O(\epsilon^{-b}n^{1+\epsilon})$ и глубины $O(\epsilon^{-1}\log n)$, где b — из предыдущей теоремы.*

Далее в работе выясняется вопрос о минимизации глубины схемы инвертирования в полях характеристики два. Показано, что инвертирование в произвольном базисе поля $GF(2^n)$ можно реализовать схемой глубины асимптотически $(3+\sigma)\log_2 n$, где σ — константа глубины многократного сложения, которая определяется как наименьшее число, такое, что существует схема сложения n одноразрядных чисел, имеющая глубину $(\sigma + o(1))\log_2 n$ (известно, что $\sigma < 3,44$). Сложность построенной схемы инвертирования равна $O(n^4)$. Данный результат вытекает из следующей теоремы о сложности и глубине реализации возведения в произвольную степень в конечном поле.

Теорема 4.5 *Пусть m — вес числа E. Тогда можно построить схему $E_{m,n}$, реализующую операцию возведения в степень E в поле $GF(2^n)$, со сложностью и глубиной (при $\epsilon > 0$)*

$$L(E_{n,m}) \le (1+o(1))\frac{\log_2(mn) + C_0(\epsilon)}{\log_2(m^2n)}\cdot m^2n^2 + C_1(\epsilon)m^{2+\epsilon}n^{1+\epsilon};$$

$$D(E_{n,m}) \leq (2+\epsilon)\log_2 n + 4,44\log_2 m + O(\log^2 \log n) + C_2(\epsilon),$$

где C_i — некоторые ограниченные на любом отрезке интервала $(0,1]$ функции.

Доказательство основывается на применении дискретного логарифмирования, в отношении которого справедлива

Теорема 4.10 *Пусть $\epsilon > 0$. Тогда существует поле характеристики 2, содержащее не менее L элементов, в котором дискретное логарифмирование выполняется со сложностью (при $L \to \infty$), не превосходящей $C_1(\epsilon)L^{\epsilon}$, и глубиной $\epsilon \log_2 L + C_3(\epsilon) + O(\log\log L)$, где C_1, C_3 — некоторые ограниченные на любом отрезке интервала $(0,1]$ функции.*

В **главе 5** описывается построение схем для реализации переходов между нормальными и стандартными базисами и рассматриваются некоторые приложения.

Основным результатом главы является следующая

Теорема 5.2 *Переход между двумя любыми нормальными или стандартными базисами поля $GF(q^n)$ может быть выполнен схемой сложности $O(n^{\nu})$ и глубины $O(\log n)$, где*

$$\nu > \min_{\sigma\in[0,1]} \max\{\omega(\sigma, 1-\sigma, 1),\ \omega((1+\sigma)/2, (1+\sigma)/2, 1)\},$$

а $\omega(\alpha,\beta,\gamma)$ — экспонента умножения матриц размера $n^{\alpha}\times n^{\beta}$ и $n^{\beta}\times n^{\gamma}$.

Из данной теоремы следует (при подстановке известных оценок для матричных экспонент), что для сложности построенных схем справедлива оценка $O(n^{1,806})$. Как следствие, умножение или инвертирование в произвольном нормальном базисе поля $GF(q^n)$ может быть реализовано схемой сложности $O(n^{1,806})$ и глубины $O(\log n)$. Это примеры операций, которые выполняются асимптотически быстрее посредством перехода к стандартному базису, чем специально разработанными для нормальных базисов алгоритмами.

В качестве примера операции в стандартном базисе, которая может быть выполнена быстрее за счет перехода к нормальному представлению, приводится тест на базисность нормальной системы, т. е. задача проверки, порождает ли заданный элемент β нормальный базис в поле $GF(q^n)$. Показано, что эта операция также реализуется схемой сложности $O(n^{1,806})$ и глубины $O(\log n)$.

Ослабляя ограничения на глубину, оценки сложности переходов можно улучшить для некоторых базисов. Справедлива

Теорема 5.3 *Переход от стандартного базиса к нормальному базису B в поле $GF(q^n)$ может быть реализован схемой сложности $O(\sqrt{n}C_B) +$*

$O(n^{1,667})$, а обратный переход можно выполнить схемой сложности

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

где C_B — сложность базиса B.

Метод умножения в нормальных базисах, вытекающий из этой теоремы, предназначен для полей с малым основанием q.

Основные результаты диссертации опубликованы автором в работах [91–93]. Текст диссертации изложен на 96 страницах. Список литературы включает 93 наименования.

2 Основные определения и вспомогательные сведения

2.1 Конечные поля

Напомним, что *полем* называется кольцо с единицей, ненулевые элементы которого образуют абелеву группу относительно операции умножения. Эта группа называется мультипликативной группой поля. *Конечным полем* называется поле, содержащее конечное число элементов — это число называется *порядком* поля. Порядок конечного поля может быть только степенью простого числа (которое является характеристикой поля), и при этом все поля одного порядка изоморфны. Мультипликативная группа конечного поля — циклическая. Единственное с точностью до изоморфизма конечное поле порядка q обозначается $GF(q)$ (GF — сокращение от Galois field, т.е. поле Галуа). Иногда также используется обозначение $\mathbb{F}_q$. Подробное изложение теории конечных полей содержится в [3, 19, 69, 5]; в последних двух книгах особое внимание уделяется алгоритмическому аспекту теории.

Поле $GF(q^n)$ можно рассматривать как расширение поля $GF(q)$ (или векторное пространство над $GF(q)$) степени n — все элементы $GF(q^n)$ порождаются линейными комбинациями над $GF(q)$ базисных элементов. С различным выбором базиса связаны различные представления элементов поля (под представлением понимается способ кодирования).

При реализации операций в конечном поле $GF(q^n)$ используется два основных представления: *стандартное (или полиномиальное)*, в котором элементы поля рассматриваются как многочлены степени не выше $n-1$, а операции производятся по модулю некоторого неприводимого над $GF(q)$ многочлена $m_n(t)$ степени n, и *нормальное*, когда элементы поля рассматриваются как линейные комбинации над $GF(q)$ с базисными элементами

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

где α — порождающий элемент (или *генератор*) базиса. В обоих случаях элементы поля кодируются набором коэффициентов в разложении по соответствующему базису.

Стандартное представление соответствует разложению элементов поля в базисе

$$A = \{1, \alpha, \alpha^2, \ldots, \alpha^{n-1}\},$$

называемом *стандартным (или полиномиальным) базисом*, где α — корень $m_n(t)$. Например, можно выбрать α соответствующим одночлену t в

явной полиномиальной записи. *Нормальный базис*

$$B = \{\alpha^{q^0}, \alpha^{q^1}, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}\},$$

также порождается корнем α неприводимого многочлена степени n, однако требуется дополнительное условие, чтобы корни этого многочлена, $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$, были линейно независимы над $GF(q)$. Известно, что пара базисов (A, B) с одним и тем же порождающим элементом может быть построена в произвольном конечном поле. Необходимое обоснование можно найти в [19, 69].

В теоретических работах используются и другие представления поля, в частности, матричное (см., например, [73, 57]). Можно также рассмотреть не связанное с выбором базиса логарифмическое представление, в котором ненулевые элементы поля кодируются степенями относительно порождающего элемента мультипликативной группы. Умножение в таком представлении сводится к сложению показателей, однако трудоемкость сложения в поле сравнивается с трудоемкостью дискретного логарифмирования, поэтому на практике логарифмическое представление почти не применяется.

2.2 Схемы над $GF(q)$

В качестве модели для реализации операций в конечном поле $GF(q^n)$ рассматриваются *схемы над* $GF(q)$, которые определяются аналогично схемам из функциональных элементов (см. [23, 33]), т.е. как ориентированные графы без ориентированных циклов с вершинами-входами, которым приписаны символы переменных или константы, и функциональными элементами в других вершинах; некоторые вершины являются выходами. Входы и выходы элементов схемы принимают значения в $GF(q)$, а сами функциональные элементы реализуют функции над $GF(q)$. Понятие схемы над $GF(2)$ тождественно понятию булевой схемы.

В качестве основного схемного базиса выбирается (функционально полный) базис бинарных арифметических операций: сложения, вычитания, умножения и деления, а также констант. Случаи использования других базисов будут специально оговариваться. Так, при реализации линейных операций будет использоваться *линейный* базис из операций сложения, вычитания и скалярного умножения (умножения на константы поля $GF(q)$).

Схемы над $GF(q)$ можно превратить в обычные булевы схемы, если применить двоичное кодирование входов и заменить функциональные элементы реализующими их булевыми схемами.

Понятия сложности и глубины схем над $GF(q)$ вводятся стандартным образом, как в [23, 33]. *Сложность* схемы S определяется как количество

функциональных элементов в ней и обозначается $L(S)$, а *глубина* — как максимальное количество элементов в цепочке, ведущей от входа к выходу схемы, и обозначается $D(S)$. Сложность и глубина функции f над $GF(q)$ определяются как

$$L(f) = \min\{L(S) \mid S \text{ реализует } f\},$$

$$D(f) = \min\{D(S) \mid S \text{ реализует } f\}.$$

2.3 Основные арифметические операции в конечных полях

Операции сложения-вычитания в любом представлении поля $GF(q^n)$, использующем разложение по базису, выполняются покоординатно и, следовательно, реализуются схемой над $GF(q)$ сложности n и глубины 1.

Сложность умножения в стандартном базисе по порядку совпадает со сложностью обычного умножения многочленов степени n. Более точно, справедлива следующая лемма (из работы [86], см. также [59, гл. 9]):

Лемма 2.1 (Штрассен, 1973) *Пусть L_n и D_n — сложность и глубина схемы умножения многочленов степени $n-1$ над $GF(q)$. Тогда можно построить схему M_n умножения в стандартном базисе поля $GF(q^n)$ сложности и глубины*

$$L(M_n) \le 3L_n + n; \qquad D(M_n) \le 3D_n + 1.$$

Стандартный метод умножения многочленов имеет сложность $O(n^2)$. Эта оценка была улучшена в 1962 г. А. А. Карацубой [14], который предложил метод «деления пополам» сложности $O(n^{\log_2 3})$. Вскоре А. Л. Тоом [29] показал, что умножение может выполняться схемой почти линейной сложности $c^{\sqrt{\log n}}n$. Наилучшая на сегодняшний день оценка сложности $O(n \log n \log\log n)$ была получена в работе Шёнхаге и Штрассена [84], правда, этот метод неприменим к умножению над полями характеристики 2. В 1977 г. Шёнхаге [83] предъявил метод умножения многочленов над полями четной характеристики с таким же порядком сложности. (В действительности, в работах [14, 29, 84] изучалось умножение чисел, однако соответствующие методы переносятся на умножение многочленов.) Все указанные методы реализуются с глубиной $O(\log n)$. Более подробно об умножении многочленов см. [59, 37].

Умножение в нормальном базисе B выполняется стандартным алгоритмом Месси—Омура [74] (см. также [5, 69]) со сложностью $O(C_B n)$, где C_B —

сложность базиса — понятие, вытекающее из структуры этого алгоритма. *Сложность нормального базиса* с генератором α определяется как суммарное количество ненулевых коэффициентов в разложении элементов

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

в этом базисе. Для любого базиса B выполнено

$$2n - 1 \leq C_B < n^2.$$

Базисы, для которых достигается нижняя оценка сложности, называются *оптимальными*. Для оптимальных нормальных базисов разработаны алгоритмы умножения с теоретической оценкой сложности $O(n \log n \log\log n)$ (например, [56]), однако такие базисы существуют не во всех полях. Более подробно вопросы реализации операций в нормальных базисах рассматриваются в [5, 69].

2.4 Линейные операции

Отображение $U : GF(q)^n \to GF(q)^m$ называется *линейным*, если существует такая $m \times n$-матрица A_U над $GF(q)$, что для любого вектора $x \in GF(q)^n$ выполнено $U(x) = A_U x$. В этом случае будем говорить, что отображение U имеет размерность $m \times n$.

В стандартном методе умножение постоянной матрицы размера $m \times n$ на вектор реализуется со сложностью $O(mn)$. Асимптотически точный результат для сложности произвольного булевого линейного отображения был получен О. Б. Лупановым в работе [21] (см. также [23]; обобщение метода Лупанова для отображений над $GF(q)$ описано в [18]):

Лемма 2.2 (Лупанов, 1956) *Линейное отображение размерности $m \times n$ над $GF(q)$ реализуется схемой сложности $(1+o(1))(mn/\log_q m)$ и глубины $\lceil \log_2 n \rceil + 2$.*

Линейные отображения с транспонированными матрицами имеют примерно одинаковую сложность в силу т.н. принципа транспозиции. История многократно переоткрывавшегося принципа транспозиции восходит, в том числе, к упомянутой работе О. Б. Лупанова 1956 г. [21], в которой используется принцип транспозиции для вентильных схем. В наиболее общей формулировке, принадлежащей Фидуччиа [54], принцип транспозиции распространяется на схемы для билинейных преобразований. Более полный исторический обзор содержится в статьях [11, 39, 9].

Лемма 2.3 (Принцип транспозиции) *Если линейное отображение размерности $m \times n$ реализуется схемой над линейным базисом сложности L и глубины D, то можно построить схему, реализующую отображение с транспонированной матрицей и имеющую сложность и глубину не выше $L + m$ и $3D + \lceil \log_2 m \rceil$ соответственно.*

Оценка для глубины получается при оптимизации схемы для транспонированного отображения (см. [9]) по глубине при помощи метода [20] (или [65], см. также [8]). Это замечание о совпадении порядка глубины линейных отображений с транспонированными матрицами фактически содержится в [71].

Следует отметить, что принцип транспозиции (точнее, алгоритм построения схемы, который за ним скрывается) служит прежде всего для доказательства существования схем с заданными сложностью и глубиной и не является конструктивным в том смысле, что схема, которую он строит, обычно имеет нерегулярную структуру (не поддается разбиению на несколько простых логических блоков). Исключение составляют некоторые простые операции (обычно имеющие линейную сложность), схемы для которых могут быть построены явно.

2.5 Умножение матриц

Обзоры теории матричного умножения на русском языке можно найти в работе В. Б. Алексеева [1] и книге Кнута [16, п. 4.6.4] (основная часть материала в последней дается в виде упражнений). Из зарубежных источников наиболее полное изложение содержится в книге [45].

Стандартный метод умножения $n \times n$-матриц использует $O(n^3)$ операций, которые могут быть выполнены с глубиной $1 + \lceil \log n \rceil$. В 1968 г. Штрассен предложил метод [85], использующий $O(n^{\log_2 7})$ операций, идея которого аналогична идее «деления пополам» из метода А. А. Карацубы для обычного умножения.

Константа в показателе порядка сложности умножения матриц называется *матричной экспонентой*. Далее мы будем пользоваться определением *достижимой* экспоненты, под которой понимается любое w, для которого существует алгоритм сложности $O(n^w)$. *Абсолютная* экспонента ω определяется как нижняя грань для достижимых экспонент — алгоритм сложности $O(n^\omega)$ при этом может и не существовать (см. [59, гл. 12]).

С 1968 г. величина матричной экспоненты неоднократно уточнялась. Рекордное на сегодняшний день значение, $w \approx 2,376$, было получено Купершмитом и Виноградом [51] в конце 80-х гг. На практике все же при-

меняются варианты стандартного метода и метода Штрассена, и совсем редко — методы с лучшим порядком сложности. Принято считать, что известные алгоритмы умножения сложности $O(n^{2,77})$ не имеют прикладного значения.

По аналогии с умножением квадратных матриц определяется и экспонента для прямоугольного умножения $T_{m,n,p}$ (умножение матрицы размера $m \times n$ на матрицу размера $n \times p$). Разработанные к настоящему времени (2006 г.) алгоритмы матричного умножения являются билинейными (подробнее о билинейных алгоритмах см. [1, 16]). Известно [66, 87], что для оценки матричной экспоненты достаточно рассматривать только билинейные алгоритмы. Справедливо следствие из [26]:

Лемма 2.4 (Пан, 1972) *Матричные умножения вида $T_{n^{\sigma(\alpha)},n^{\sigma(\beta)},n^{\sigma(\gamma)}}$, где σ — произвольная функция перестановки элементов множества $\{\alpha, \beta, \gamma\}$, имеют одну и ту же экспоненту.*

Оценки сложности прямоугольного умножения матриц содержатся в работе [67]. Например, для представляющей особенный интерес операции $T_{\sqrt{n},\sqrt{n},n}$ там получена оценка $O(n^{1,667})$.

Известные алгоритмы умножения матриц имеют логарифмическую глубину по построению. Справедлив следующий общий результат [77] (см. также [40]):

Лемма 2.5 (Пан, 1987) *Матричное умножение $T_{n^\alpha,n^\beta,n^\gamma}$ может быть реализовано схемой сложности $O(n^w)$ и глубины $O(\log n)$, где w — соответствующая матричная экспонента.*

2.6 Арифметика чисел и многочленов

Цель данного параграфа состоит в том, чтобы отметить близость следующих классов арифметических операций:

— *числовые операции* (числа записываются в двоичном представлении, сложность измеряется количеством операций над битами);

— *операции с многочленами* (над некоторым коммутативным кольцом, сложность измеряется количеством операций в данном кольце);

— *операции модулярной арифметики многочленов* (аналогично предыдущему пункту, но вычисления производятся по модулю некоторого фиксированного многочлена);

— *операции со степенными рядами* (частный случай предыдущего пункта, вычисления производятся по модулю одночлена).

Алгоритм, выполняющий операцию с одним из перечисленных типов объектов, обычно имеет аналог для соответствующей операции с другим типом. В тех случаях, когда аналогия не прослеживается, можно использовать различные методы сведения операций из одного класса к операциям из другого класса.

Например, метод умножения Карацубы одинаково записывается для чисел, многочленов и степенных рядов. Для модулярного умножения многочленов можно воспользоваться леммой 2.1 и свести его к обычным умножениям многочленов.

Соответственно, известные оценки сложности для одних и тех же операций из различных классов зачастую оказываются близки (имеют тот же порядок, либо порядки различаются медленно растущим множителем): сравнивается сложность операций с одинаковым числом входов, например, с n-разрядными целыми числами с одной стороны и с многочленами степени $n-1$ — с другой.

Отчасти это обусловлено совпадением по порядку сложности выполнения фундаментальных операций сложения и умножения в перечисленных класах. Сложение выполняется с линейной сложностью, а сложность умножения составляет $O(n \log n \log\log n)$. Последняя оценка достигается в методе умножения чисел Шёнхаге—Штрассена [84] (см. также [2, гл. 7]), который модифицируется для умножения многочленов над кольцами, в которых обратим элемент 2 (сумма двух единиц кольца), см. [59]. Если элемент 2 необратим, но обратим элемент 3, то используется метод [83]. Наконец, если ни тот, ни другой элемент не обратим, то используется метод Кантора—Калтофена [47] (более подробно см. в [59, 37]). В случае модулярного умножения используется метод леммы 2.1.

В действительности, умножение многочленов над полем $\mathbb{R}$ или $\mathbb{C}$ может выполняться со сложностью $O(n \log n)$ (см., например, [2]). Однако, в соответствии с тематикой настоящей работы, для нас больший интерес представляют многочлены с коэффициентами из конечного поля $GF(q)$. Более конкретно, операции в конечном поле $GF(q^n)$ с использованием стандартного представления являются операциями арифметики многочленов по модулю $m_n(t)$, где $m_n(t)$ — (неприводимый) характеристический многочлен выбранного представления поля.

3 Многократное умножение

Под *m-кратным умножением* понимается операция умножения m многочленов (чисел или каких-либо других объектов).

Многократное умножение относится к классическим задачам теории синтеза параллельных схем. Очевидный способ, состоящий из попарных перемножений, позволяет выполнять m-кратное умножение (многочленов степени n, либо n-разрядных чисел) схемой сложности

$$O(mn(\log^2 m + \log n)\log\log(mn))$$

и глубины $O(\log m \log(mn))$, если для обычных умножений использовать методы Шёнхаге и Штрассена.

По-видимому, первая схема логарифмической глубины для многократного умножения чисел была построена Бимом, Куком и Гувером [36] в середине 80-х гг. Чуть позже Эберли [52] получил следствия для многократного умножения многочленов над различными кольцами. Сложность предложенных схем достаточно высока. Так, для n-кратного умножения n-разрядных чисел с глубиной $O(\log n)$ авторами [36] была указана оценка сложности $O(n^5 \log^2 n)$, а, например, n-кратное умножение многочленов степени n над полем $GF(2)$ в работе [52] сводится к n-кратному умножению приблизительно $n \log_2 n$-разрядных чисел.

Способ уменьшения порядка сложности схемы из [36] был указан Хастадом и Лейтоном в [64] (с его помощью авторы построили схему деления чисел сложности $O(n^{1+\epsilon})$ и глубины $O(\epsilon^{-2}\log n)$). Рейф и Тейт [79] рассмотрели возможность использования дискретного преобразования Фурье. Построенная ими схема (для возведения n-разрядного числа в степень m) имеет глубину $O(\log n + \log m \log\log m)$ и сложность $O(nm^5 \log n \log\log n)$. Это позволило построить схему деления чисел сложности $O(n \log n \log\log n)$ (т.е. такого же порядка, как для умножения) и глубины $O(\log n \log\log n)$.

Далее в этой главе рассматриваются различные подходы к построению схем многократного умножения многочленов над конечными полями с глубиной $O(\log(mn))$. Основной целью (мотивируемой, в первую очередь, приложением к инвертированию) является минимизация порядка сложности таких схем относительно n. Учитывая то, что этой задаче, насколько известно автору, даже в специальных работах практически не уделялось внимания, дается существенно больше материала, чем требуется для вывода основных результатов.

3.1 Числовое и модулярное многократное умножение

Умножение многочленов над простым конечным полем $GF(q)$, т.е. $q \in \mathbb{P}$, где $\mathbb{P}$ — множество простых натуральных чисел, можно свести к умножению q-ичных чисел, используя следующую идею.

Коэффициенты произведения $f(x)$ многочленов $f_1(x), \ldots, f_m(x)$ степени не выше $n-1$ над $GF(q)$ можно восстановить по числовому произведению

$$f(q^L) = f_1(q^L) \cdot \ldots \cdot f_m(q^L),$$

где L выбирается достаточно большим. Можно выбрать $L = m(1+\lceil \log_q n \rceil)$, тогда каждый jL-й разряд (в q-ичной системе счисления, нумерация с нуля) числа $f(q^L)$ есть коэффициент многочлена $f(x)$ при x^j, где $j = 0, \ldots, m(n-1)$. Более подробно см., например, в [52].

По всей видимости, этот способ применять не очень выгодно, т.к. он сводит умножение многочленов степени, меньшей n, к умножению $mn \log_q n$-разрядных чисел. Поэтому далее будут исследоваться специально предназначенные для умножения многочленов методы.

Рассмотрим связь между обычным и модулярным многократным умножением.

Лемма 3.1 *Пусть L_n и D_n — сложность и глубина схемы умножения многочленов степени $n-1$ над $GF(q)$. Тогда операция приведения многочлена степени не выше $kn-1$ по модулю фиксированного многочлена $g(x)$ степени n реализуется схемой сложности $(k+1)L_n + O(kn)$ и глубины $3D_n + O(\log k)$.*

Доказательство. Представим многочлен $f(x)$ степени, не большей $kn-1$, в виде

$$f(x) = f_{k-1}(x)x^{(k-1)n} + \ldots + f_1(x)x^n + f_0(x),$$

где $\deg f_i < n$. Обозначим $g_i(x) = x^{in} \bmod g(x)$. Тогда справедливо

$$f(x) \bmod g(x) = \sum_{i=0}^{k-1} f_i(x)g_i(x) \bmod g(x),$$

откуда с учетом леммы 2.1 вытекают требуемые оценки (сначала выполняются умножения f_i на g_i, затем произведения складываются, окончательно результат приводится по модулю $g(x)$).

Лемма 3.2 *Пусть $r \in \mathbb{N}$ и $s = \lceil \sqrt[r]{m} \rceil$. Пусть $L_{s,n}$ и $D_{s,n}$ — сложность и глубина схемы s-кратного умножения многочленов степени $n-1$ над*

$GF(q)$, а L_n и D_n — сложность и глубина схемы обычного умножения. Тогда можно построить схему $M_{m,n}$ m-кратного умножения по модулю многочлена $g(x)$ степени n сложности и глубины

$$L(M_{m,n}) = O(mL_{s,n}/s + mL_n), \qquad D(M_{m,n}) = O(rD_{s,n} + rD_n + \log m).$$

Доказательство. Вычисление устроено следующим образом: сомножители разбиваются на группы по s штук, для которых выполняются s-кратные умножения, а произведения приводятся по модулю $g(x)$. Эта операция затем повторяется для умножения новых многочленов (количество которых не превосходит s^{r-1}). Оценки сложности и глубины легко проверяются непосредственно или по индукции с использованием леммы 3.1.

3.2 Применение дискретного логарифмирования

Рассмотрим следующий алгоритм многократного умножения многочленов над $GF(q)$, в основе которого лежат идея интерполяции, восходящая к А. Л. Тоому [29], и идея дискретного логарифмирования из работы Эберли [52].

Напомним, что мультипликативная группа конечного поля $GF(q)$ — циклическая. Произвольный порождающий ее элемент называется *примитивным*. На мультипликативной группе определена функция *дискретного логарифма* по основанию некоторого примитивного элемента α, а именно, для любого элемента β полагается $\log_\alpha \beta = b$, где $\alpha^b = \beta$, $0 \le b \le q-1$.

Пусть требуется вычислить произведение $f(x) = f_1(x) \cdot \ldots \cdot f_m(x)$, где $\deg f_i < n$. Положим $L = m(n-1)+1$ и $k = \lceil \log_q L \rceil$. В поле $GF(q^k)$ зафиксируем набор элементов $\alpha_1, \ldots, \alpha_L$.

1. Вычислим всевозможные $f_i(\alpha_j) \in GF(q^k)$, где $i = 1, \ldots, m$, $j = 1, \ldots, L$.

2. Для каждого j вычислим произведение $f_1(\alpha_j) \cdot \ldots \cdot f_m(\alpha_j) = f(\alpha_j)$. Для этого в поле $GF(q^k)$ выберем примитивный элемент α (т.е. порождающий мультипликативную группу поля). Если $f_i(\alpha_j) \neq 0$ для всех i, тогда

2.1. Вычислим дискретные логарифмы, $\log_\alpha f_i(\alpha_j)$.

2.2. Вычислим $l_j = \sum_{i=1}^m \log_\alpha f_i(\alpha_j)$.

2.3. Вычислим $f(\alpha_j) = \alpha^{l_j}$ (эта формула корректна, поскольку $l_j \equiv \log_\alpha f(\alpha_j) \bmod (q^k - 1)$).

3. По значениям $f(\alpha_j)$, $j = 1, \ldots, L$, восстанавливается многочлен $f(t)$ степени не выше $L - 1$.

Для оценки сложности воспользуемся следующими известными фактами.

Лемма 3.3 (Лупанов, 1963) *Произвольная функция n переменных над $GF(q)$ реализуется схемой сложности $O(q^n/n)$ и глубины $O(n)$.*

Лемма 3.4 (Офман, 1962) *Сумматор m штук n-разрядных q-ичных чисел реализуется схемой сложности $O(mn)$ и глубины $O(\log(mn))$.*

Лемма 3.5 *Пусть M_n — схема умножения в поле $GF(q^n)$. Тогда экспоненцирование в $GF(q^n)$, т.е. возведение фиксированного элемента $\alpha \in GF(q^n)$ в переменную степень, записываемую m разрядами над $GF(q)$, выполняется схемой сложности $O(m(L(M_n) + qn))$ и глубины $O(\log q + D(M_n) \log m)$.*

Лемма 3.3 является обобщением на схемы над $GF(q)$ метода О. Б. Лупанова асимптотически оптимального синтеза схем для булевых функций [22] (см. также [23, 33]). Следующая лемма является аналогичным обобщением метода Ю. П. Офмана [14]. Последняя лемма вытекает из стандартного способа экспоненцирования, основанного на предвычислении всех элементов α^{iq^j}, $i = 0, \ldots, q-1$, $j = 0, \ldots, n-1$.

Для обоснования корректности оценок указанных лемм требуется расширение базиса из арифметических операций над $GF(q)$. При простом q достаточно добавить функцию минимума или максимума двух элементов поля (как чисел от 0 до $q - 1$). В общем случае, требуются функциональные элементы, реализующие необходимые арифметические операции в $\mathbb{Z}_q$ (при наличии соответствия между элементами $GF(q)$ и $\mathbb{Z}_q$ можно считать, что добавляемые функциональные элементы реализуют некоторые функции над $GF(q)$).

Теорема 3.1 *Пусть $q = (mn)^{O(1)}$. Тогда можно построить схему m-кратного умножения многочленов степени $n - 1$ над $GF(q)$ сложности $O(qm^3n^2)$ и глубины $O(\log(mn))$.*

Доказательство. Поскольку элементы α_i фиксированы, то первый и последний шаги алгоритма состоят в выполнении линейных преобразований размерности $mLk \times mn$ и $L \times Lk$ соответственно. Согласно лемме 2.2, шаг 1 реализуется со сложностью $O(m^2Lnk/\log_q(mLk)) = O(m^3n^2)$ и глубиной

$O(\log(mn))$, а шаг 3 — со сложностью $O(L^2k/\log_q L) = O(m^2n^2)$ и глубиной $O(\log(mn))$.

Рассмотрим шаг 2. Корректирующие подсхемы (вычисляющие, есть ли нули среди сомножителей, и обнуляющие результат в таком случае) реализуются со сложностью $O(Lmk) = O(m^2n\log_q(mn))$ и увеличивают глубину реализации шага не более, чем на 1 относительно схемы, построенной в предположении, что все сомножители отличны от нуля.

На этом шаге вычисляется Lm дискретных логарифмов, L штук m-кратных сумм k-разрядных чисел и L степеней элемента α.

Схему для дискретного логарифма (с k входами и k выходами) можно построить методом леммы 3.3 со сложностью $O(q^k) = O(qmn)$ и глубиной $O(k) = O(\log_q(mn))$. Следовательно, сложность шага 2.1 оценивается как $O(qLm^2n) = O(qm^3n^2)$.

Лемма 3.4 позволяет построить схему для m-кратного сложения k-разрядных чисел со сложностью $O(mk)$ и глубиной $O(\log(mk))$, следовательно, сложность шага 2.2 оценивается как $O(Lmk) = O(m^2n\log_q(mn))$.

Показатели l_j состоят не более, чем из $k + \lceil\log_q m\rceil = O(k)$ разрядов. Поэтому, согласно лемме 3.5, для возведения элемента α в степень l_j строится схема сложности $O(kM(k) + qk^2) = O(q\log_q^3(mn))$ и глубины $O(\log q + D_M(k)\log k) = O(\log(mn))$. Следовательно, сложность шага 2.3 оценивается как $O(Lq\log_q^3(mn)) = O(qmn\log_q^3(mn))$.

Суммируя оценки для всех подсхем, получаем окончательно утверждение теоремы.

Доказанная оценка сложности является довольно грубой, т.к. шаги 1 и 2.1, дающие основной вклад в эту оценку, могут быть реализованы более эффективным образом. Однако мы не будем заниматься уточнением, т.к. данный метод будет применяться при малых значениях параметров m и n.

Заметим, что если требуется выполнить многократное умножение с приведением по модулю некоторого фиксированного многочлена, то для этой операции также справедливы оценки теоремы 3.1, т.к. приведение по модулю (которое является линейной операцией) в композиции с линейным преобразованием, выполняемым на шаге 3, не приводит к увеличению размерности совокупного преобразования.

3.3 Применение китайской теоремы об остатках

Вероятно, наиболее простой путь, ведущий к схеме почти линейной сложности по n, заключается в использовании модулярной арифметики, основанной на китайской теореме об остатках (см., например, [2, 16, 24, 59]).

Опишем адаптированный для умножения многочленов метод Хаастада—Лейтона [64].

Лемма 3.6 *Пусть $h_1(x), \ldots, h_k(x)$ — попарно взаимно простые многочлены степени r над $GF(q)$. Тогда операция приведения многочлена $f(x)$ степени $kr-1$ по модулям $h_i(x)$ и обратная операция восстановления многочлена степени не выше $kr-1$ по заданным остаткам $\varphi_i(x)$ от деления на $h_i(x)$ выполняются со сложностью $O(k^2 L_r)$ и глубиной $O(D_r + \log k)$, где L_r и D_r — сложность и глубина схемы умножения многочленов степени $r-1$.*

Доказательство. Одно приведение многочлена степени $kr-1$ по модулю многочлена степени r выполняется со сложностью $O(kL_r)$ и глубиной $O(D_r + \log k)$ (см. лемму 3.1).

Для восстановления многочлена по остаткам от деления можно воспользоваться формулой Лагранжа (см., например, [2, гл. 8], [59, гл. 5]):

$$f(x) = \sum_{i=1}^{k} (\varphi_i(x)\mu_i(x) \bmod h_i(x))\lambda_i(x),$$

где $\lambda_i(x) = \prod_{j \neq i} h_j(x)$ и $\mu_i(x) = \lambda_i^{-1}(x) \bmod h_i(x)$ — фиксированные многочлены степени не выше $(k-1)r$ и $r-1$ соответственно. Требуемые оценки сложности и глубины получаются очевидным образом.

Теорема 3.2 *Пусть $l \in \mathbb{N}$. Тогда можно построить схему m-кратного умножения многочленов степени $n-1$ над $GF(q)$ сложности и глубины*

$$O\left(lm^2 n \sqrt[l]{mn}(\log(mn)\log\log(mn) + l^2)\right); \qquad O(l\log(mn)).$$

Доказательство. Известно, что можно выбрать $(1-o(1))q^d/d$ попарно взаимно простых многочленов степени d над $GF(q)$ (это следует из оценки числа неприводимых многочленов, см. [19, гл. 3], [69, гл. 1]).

Выберем такие k и d, что существует k попарно взаимно простых многочленов $h_1(x), \ldots, h_k(x)$ степени d, и $kd > m(n-1)$. Можно полагать $k = (1-o(1))mn/\log_q(mn)$ и $d = (1+o(1))\log_q(mn)$.

Построим корневое дерево из $l+1$ ярусов, каждая вершина которого соединена с порядка $k^{1/l}$ вершинами на следующем снизу ярусе, а на нижнем ярусе расположены k концевых вершин. Поставим в соответствие последним многочлены $h_i(x)$, и по индукции каждой вершине дерева — произведение многочленов, соответствующих вершинам следующего снизу яруса, с которыми данная вершина соединена ребрами. Таким образом, корню соответствует многочлен $h_1(x) \cdot \ldots \cdot h_k(x)$.

Описанная конструкция фактически является $k^{1/l}$-арным аналогом известного бинарного дерева для выполнения вычислений согласно китайской теореме (см., например, [2, гл. 8]).

Процедуру нахождения остатков от деления многочлена степени не выше $m(n-1)$ на все $h_i(x)$ можно рассматривать как движение по дереву от корня вниз — в каждой вершине вычисляется остаток от деления на ассоциированный с ней многочлен. Аналогично, восстановление многочлена с заданными остатками от деления на $h_i(x)$ рассматривается как движение от нижнего яруса к корню, где получается окончательный результат.

Вычисления в каждой вершине выполняются методом леммы 3.6, параллельно во всех вершинах одного яруса. На i-м сверху ярусе расположено $(1+o(1))k^{(i-1)/l}$ вершин, которым соответствуют многочлены степени $(1+o(1))dk^{(l-i+1)/l}$, поэтому согласно лемме 3.6 вычисления на каждом ярусе выполняются со сложностью $O(dk^{1+1/l}\log(kd)\log\log(kd))$ и глубиной $O(\log(kd))$.

Таким образом, нахождение остатков для многочлена степени не выше $m(n-1)$ и восстановление такого многочлена по известным остаткам реализуется схемой сложности $O(l(mn)^{1+1/l}\log(mn)\log\log(mn))$ и глубины $O(l\log(mn))$.

Для умножения m многочленов степени не выше $n-1$ следует вычислить их остатки от деления на многочлены $h_i(x)$, вычислить соответствующие многократные произведения по модулям $h_i(x)$ и окончательно восстановить многочлен-произведение.

Если $q > \sqrt[l]{mn}$, то степень d многочленов $h_i(x)$ не превосходит $O(l)$, и соответствующие многократные произведения (по модулям $h_i(x)$) могут быть вычислены посредством попарных перемножений со сложностью $O(kmd\log d\log\log d) = O(lm^2n)$ и глубиной $O(\log l\log m) = O(l\log m)$.

При $q \le \sqrt[l]{mn}$ произведения остатков вычисляются схемами из леммы 3.2 сложности и глубины

$$O(kmL_{s,d}/s + kmd\log d\log\log d); \qquad O(rD_{s,d} + r\log d + \log m),$$

где r — натуральный параметр, $s \approx \sqrt[r]{m}$, а $L_{s,d}$ и $D_{s,d}$ — сложность и глубина схемы s-кратного умножения многочленов степени $d-1$. Выберем $r = O(\log m/\log d)$ (или $r=1$, если $d > m$) так, чтобы выполнялось $s = O(d)$. Величину $L_{s,d}$ и $D_{s,d}$ оценим при помощи теоремы 3.1, тогда окончательно для сложности подсхемы многократных модулярных умножений имеем оценку

$$O(kmqs^2d^2) = O(m^2nq\log_q^3(mn)) = O(l^3m^2n\sqrt[l]{mn}),$$

а для глубины — $O(\log(mn))$.

Суммируя оценки по всем подсхемам, выводим утверждение теоремы.

Отметим, что комбинация метода теоремы 3.1 и основного метода теоремы 3.2 позволяет снять ограничение на порядок поля, q, и устранить q из оценки сложности.

3.4 Применение ДПФ

3.4.1 Дискретное преобразование Фурье

Напомним кратко основные сведения о ДПФ (более подробно см. в [2, гл. 7], [24, гл. 5], [59, гл. 8]).

Пусть K — коммутативное кольцо, $\zeta \in K$ — корень степени N из единицы в данном кольце, для которого выполнено условие: для любого простого $p \mid N$ элемент $\zeta^{N/p} - 1$ не является делителем нуля в K (в таком случае ζ называется *примитивным* корнем). *ДПФ порядка* N называется отображение, ставящее в соответствие набору $\gamma_0, \gamma_1, \ldots, \gamma_{N-1}$ элементов из K набор $\gamma_0^*, \gamma_1^*, \ldots, \gamma_{N-1}^*$ по правилам

$$\gamma_j^* = \sum_{i=0}^{N-1} \gamma_i \zeta^{ij}. \qquad (f)$$

Важнейшим свойством преобразования Фурье является то, что обратное преобразование (если оно существует) выглядит почти так же, как и прямое, а именно, справедливо

$$N\gamma_j = \sum_{i=0}^{N-1} \gamma_i^* \zeta^{-ij}. \qquad (f^*)$$

Если в кольце K число N (сумма N единиц) обратимо, то первоначальный набор может быть восстановлен домножением (f^*) на N^{-1}, а соответствующее отображение называется обратным преобразованием Фурье. Как видим, оно отличается от прямого постоянным множителем N^{-1} и выбором ζ^{-1} в качестве примитивного корня.

В некоторых алгоритмах (например, [47]) используется не совсем обратное ДПФ в форме (f^*). Существование N^{-1} в таком случае не требуется, и условие примитивности ζ можно несколько ослабить.

Если набор $\gamma_0, \gamma_1, \ldots, \gamma_{N-1}$ ассоциировать с коэффициентами многочлена $\Gamma(x) = \sum \gamma_i x^i$, то прямое преобразование Фурье вычисляет набор значений многочлена $\Gamma(x)$ в точках ζ^j, $j = 0, \ldots, N-1$, а именно, $\gamma_j^* = \Gamma(\zeta^j)$. Обратное преобразование Фурье восстанавливает коэффициенты единственного многочлена $G(x)$ степени не выше $N-1$, такого, что $G(\zeta^i) = \gamma_i^*$ для

всех $i = 0, \ldots, N-1$ (такой же набор значений имеет любой многочлен, совпадающий с $G(x)$ по модулю $x^N - 1$).

Это фундаментальное свойство ДПФ используется при умножении многочленов: очевидно, значение произведения двух многочленов в некоторой точке совпадает с произведением значений сомножителей. Поэтому для умножения двух многочленов (при надлежащем выборе N) достаточно применить ДПФ к коэффициентам каждого из сомножителей, затем покомпонентно перемножить полученные наборы, найти произведение при помощи обратного ДПФ.

ДПФ над алгебраически замкнутым полем F реализуется схемой сложности $O(N \log N)$ и глубины $O(\log N)$ операций сложения и умножения на степени ζ. Алгоритмы, на которых достигается эта оценка сложности, называются алгоритмами *быстрого преобразования Фурье (БПФ)*. Считается, что общая идея БПФ была высказана еще Гауссом. Однако широкое распространение в различных приложениях алгоритмы БПФ получили после 1965 г., когда Кули и Тьюки [49] опубликовали первый такой алгоритм для $N = 2^k$. Сложность базового алгоритма Кули—Тьюки составляет $1,5N \log_2 N$ операций в поле F (см., например, [59, гл. 8]), однако для некоторых полей она может быть меньше: так, в поле характеристики 2 ДПФ выполняется за $N \log_2 N$ операций. Используя метод Блюштейна [41], можно получить оценку сложности $O(N \log N)$ для ДПФ произвольного порядка N (см. также [7, 37]; другие алгоритмы БПФ можно найти в [24] и по ссылкам в [37]).

Умножение многочленов над кольцом, в котором нет необходимых корней из единицы, реализуется посредством следующей идеи [84]. Наборы коэффициентов сомножителей организуются в блоки определенной длины, которые интерпретируются как элементы некоторого кольца, в котором определено ДПФ требуемого порядка и результат умножения над которым позволяет восстановить произведение исходных многочленов. Реализации этой идеи применительно к многократному умножению посвящены следующие два пункта. Умножение над полем $GF(q)$ сводится к умножению над кольцом $GF(q)[x]/(x^{a^s} - 1)$, где $a = 2$, если q — нечетно, и $a = 3$, если q является степенью двойки (элементы такого кольца однозначно представляются многочленами степени, меньшей $(a-1)a^{s-1}$, операции с которыми производятся по модулю $(x^{a^s} - 1)/(x^{a^{s-1}} - 1)$).

3.4.2 Умножение над полем характеристики 2

Пусть $q = 2^t$. Рассмотрим кольцо $R_s^{(x)} = GF(q)[x]/(x^{2 \cdot 3^{s-1}} + x^{3^{s-1}} + 1)$ многочленов над $GF(q)$ по модулю $x^{2 \cdot 3^{s-1}} + x^{3^{s-1}} + 1$ (верхний индекс в

обозначении $R_s^{(x)}$ указывает на символ используемой переменной). В этом кольце элемент x является примитивным корнем степени 3^s из единицы, т.к. многочлены $x^{2\cdot 3^{s-1}} + x^{3^{s-1}} + 1$ и $x^{3^{s-1}} - 1$ взаимно просты. Поэтому в $R_s^{(x)}$ определено ДПФ порядка 3^s.

Операции в $R_s^{(x)}$ удобно производить по модулю $x^{3^s} - 1$, тогда элементы кольца представляются многочленами степени не выше $3^s - 1$. Такое представление неоднозначно — для получения однозначного представления требуется приведение по модулю $x^{2\cdot 3^{s-1}} + x^{3^{s-1}} + 1$ (оно выполняется со сложностью $2 \cdot 3^{s-1}$ и глубиной 1).

Заметим, что в указанном представлении прямое и обратное ДПФ (порядка $N = 3^i$) над $R_s^{(x)}$ с вычислительной точки зрения эквивалентны. Для обратного преобразования следует использовать разложение элементов по степеням x^{-1}, а оно отличается от обычного только порядком следования коэффициентов. Деление на N производить не нужно, т.к. $N \equiv 1 \bmod 2$.

Будем обозначать через $M(m,n)$ и $D_M(m,n)$ соответственно сложность и глубину (для реализации одной схемой) m-кратного умножения многочленов степени $n-1$ над $GF(q)$, а через $M'(k,s)$ и $D'_M(k,s)$ — сложность и глубину 3^k-кратного умножения в кольце $R_s^{(x)}$.

Если $m(n-1) < 2\cdot 3^{s-1}$, то m-кратное умножение многочленов степени $n-1$ можно свести к m-кратному умножению в $R_s^{(x)}$, следовательно,

$$M(m,n) \le M'(\lceil \log_3 m \rceil, \lceil \log_3(mn/2) \rceil + 1) + 3mn;$$

$$D_M(m,n) \le D'_M(\lceil \log_3 m \rceil, \lceil \log_3(mn/2) \rceil + 1) + 1, \qquad (3.1)$$

где добавочный член в оценках отвечает приведению по модулю $x^{2\cdot 3^{s-1}} + x^{3^{s-1}} + 1$.

Умножение в кольце $R_s^{(x)}$ может выполняться как умножение обычных многочленов с последующим приведением по модулю $x^{3^s} - 1$. Приведение многочлена степени $m3^s - 1$ по такому модулю равносильно сложению m многочленов степени $3^s - 1$, поэтому

$$M'(k,s) \le M(3^k, 3^s) + (3^k - 1)3^s;$$

$$D'_M(k,s) \le D_M(3^k, 3^s) + \lceil k \log_2 3 \rceil. \qquad (3.2)$$

Лемма 3.7 *Пусть $r \le s$. Тогда ДПФ порядка 3^r над $R_s^{(y)}$ реализуется схемой сложности и глубины*

$$\Phi(s,r) \le 2r3^{r+s}; \qquad D_\Phi(s,r) \le 2r.$$

Доказательство. Выберем $z = y^{3^{s-r}}$ — корень из единицы степени 3^r в $R_s^{(y)}$, и пусть $\Gamma(x) = \sum \gamma_i x^i$ — многочлен над $R_s^{(y)}$ степени не выше $3^r - 1$. Запишем,

$$\Gamma(z^l) = \sum_{i=0}^{3^r-1} \gamma_i z^{il} = \sum_{j=0}^{2} z^{jl} \sum_{i=0}^{3^{r-1}-1} \gamma_{3i+j} (z^3)^{il} = \Gamma_0(z_*^{l_1}) + z^l \Gamma_1(z_*^{l_1}) + z^{2l} \Gamma_2(z_*^{l_1}),$$

где $z_* = z^3$ — корень степени 3^{r-1} из единицы, $l_1 = l \bmod 3^{r-1}$, а

$$\Gamma_j(x) = \sum_{i=0}^{3^{r-1}-1} \gamma_{3i+j} x^i, \qquad j = 0, 1, 2.$$

Пусть необходимые значения $\Gamma_j(z_*^i)$, $j = 0, 1, 2$, $i = 0, \ldots, 3^{r-1} - 1$, найдены при помощи трех ДПФ порядка 3^{r-1}. Умножения на степени z не требуют схемных затрат (осуществляются циклическим сдвигом коэффициентов), поэтому для вычисления каждого из 3^r значений $\Gamma(z^l)$ остается выполнить два сложения в $R_s^{(y)}$ со сложностью $2 \cdot 3^s$ и глубиной 2. Получаем рекуррентные неравенства

$$\Phi(s, r) \le 2 \cdot 3^{s+r} + 3\Phi(s, r-1); \qquad D_\Phi(s, r) \le 2 + D_\Phi(s, r-1),$$

которые, если учесть, что $\Phi(s, 0) = D_\Phi(s, 0) = 0$, разрешаются как

$$\Phi(s, r) \le 2r3^{s+r}; \qquad D_\Phi(s, r) \le 2r.$$

При помощи следующей леммы можно свести m-кратное умножение в кольце $R_s^{(x)}$ к вычислениям в кольце меньшего порядка — ее доказательство, по существу, состоит в применении итерации метода Шёнхаге [83], а более точно, метода [47].

Лемма 3.8 *Пусть $k \le s - r$, тогда*

$$M'(k, r+s) \le 3^s M'(k, s) + 2(3^k + 1)s3^{2s} + (3^k - 1)3^{r+s};$$

$$D'_M(k, r+s) \le D'_M(k, s) + 4s + \lceil k \log_2 3 \rceil.$$

Доказательство. Пусть $f_i \in R_{r+s}^{(x)}$ и $f = f_1 \cdot \ldots \cdot f_{3^k}$. Запишем

$$f_i = \sum_{j=0}^{3^{r+s}-1} f_{i,j} x^j = \sum_{j=0}^{3^r-1} \left(\sum_{l=0}^{3^s-1} f_{i, l3^r+j}\, x^{l3^r} \right) x^j = \sum_{j=0}^{3^r-1} \varphi_{i,j}(\tilde{y}) x^j,$$

где $\tilde{y} = x^{3^r}$ и $\deg \varphi_{i,j} < 3^s$.

Установим соответствие между многочленами $\varphi_{i,j}(\tilde{y})$ (степени не выше $3^s - 1$) и элементами $\tilde{\varphi}_{i,j} \in R_s^{(y)}$ по правилу $\tilde{\varphi}_{i,j} = \varphi_{i,j}(y)$. Оно автоматически порождает (взаимно однозначное) соответствие между элементами $R_{r+s}^{(x)}$ и многочленами из $R_s^{(y)}[x]$ степени не выше $3^r - 1$. При этом операции в $R_s^{(y)}[x]$ соответствуют операциям в $R_{r+s}^{(x)}$ с точностью до приведения по модулю $x^{3^r} - y$. Таким образом, если обозначить через $f_i'(x)$ многочлены из $R_s^{(y)}[x]$, соответствующие f_i, и $f'(x) = f_1'(x) \cdot \ldots \cdot f_{3^k}'(x) \in R_s^{(y)}[x]$, то многочлену $f'(x) \bmod (x^{3^r} - y)$ соответствует искомый элемент $f \in R_{r+s}^{(x)}$.

Многочлен $f'(x)$ имеет степень не выше $3^k(3^r - 1)$, поэтому он однозначно восстанавливается по своим значениям в $3^s \geq 3^{k+r}$ точках $R_s^{(y)}$. Рассмотрим следующий алгоритм 3^k-кратного умножения в $R_{r+s}^{(x)}$.

А. С помощью ДПФ вычислим значения всех многочленов $f_i'(x)$ в точках $1, y, y^2, \ldots, y^{3^s-1}$ (y является примитивным корнем степени 3^s в $R_s^{(y)}$).

Б. Произведения $f'(y^j) = f_1'(y^j) \cdot \ldots \cdot f_{3^k}'(y^j)$ вычисляются при помощи 3^s экземпляров 3^k-кратных умножений в $R_s^{(y)}$.

В. Многочлен $f'(x)$ восстанавливается по своим значениям в точках $1, y, y^2, \ldots, y^{3^s-1}$ при помощи обратного ДПФ.

Г. Многочлен $f'(x)$ приводится по модулю $x^{3^r} - y$.

Оценки для подсхем, соответствующих шагам А и В, следуют из леммы 3.7, и для завершения доказательства остается показать, что заключительный шаг реализуется схемой сложности $(3^k - 1)3^{r+s}$ и глубины $\lceil \log_2 3^k \rceil$. Запишем, $f'(x) = \sum \psi_i(x) x^{i3^r}$, где $\psi_i(x) \in R_s^{(y)}[x]$ и $\deg \psi_i < 3^r$. Тогда $f'(x) \bmod (x^{3^r} - y) = \sum y^i \psi_i(x)$. Умножение на степени y в кольце $R_s^{(y)}$ осуществляется циклическим сдвигом коэффициентов. Остается просуммировать 3^k многочленов над $R_s^{(y)}$ степени не выше $3^r - 1$, что, очевидно, может быть выполнено с заявленными оценками.

Прототипом следующего алгоритма многократного умножения является метод возведения чисел в степень из работы [79]. Его идея заключается в чередовании итераций ДПФ (как в лемме 3.8), посредством которых осуществляется переход к вычислениям в кольце меньшего порядка, с итерациями, приводящими к сокращению числа сомножителей. Справедливы соотношения:

$$M'(k + l, s) \leq 3^l M'(k, s) + M'(l, s); \quad D'_M(k + l, s) \leq D'_M(k, s) + D'_M(l, s).$$

Действительно, для вычисления произведения 3^{k+l} сомножителей можно параллельно вычислить произведения в группах по 3^k штук, а затем перемножить 3^l полученных произведений.

Лемма 3.9 *Пусть* $j \in \mathbb{N}$, $j \le \lceil \log_2 k \rceil$, $k_j = \lceil k/2^j \rceil$, $s_j = \lceil \frac{s}{4^j} + \frac{1{,}5k}{2^j} \rceil + 1$. *Тогда*

$$M'(k,s) \le O(3^{3k+s+4j-3k_j-s_j}) M'(k_j, s_j) + O(s_j 3^{3k+s+4j-2k_j});$$

$$D'_M(k,s) \le 2^j (D'_M(k_j, s_j) + 50,4) + 12s + 21,2jk.$$

Доказательство. Если $k \ge 2$, то сведем 3^k-кратное умножение к 3^{k_1}-кратным как указано выше, где $k_1 = \lceil k/2 \rceil$. Затем выполним две итерации леммы 3.8. В результате получим рекуррентные соотношения:

$$M'(k,s) \le (3^{k-k_1}+1)\left(3^{r_1} M'(k_1, r_1) + O(r_1 3^{k_1+2r_1})\right) \le$$
$$\le (3^{k-k_1}+1)\left(3^{r_1+s_1} M'(k_1, s_1) + O(s_1 3^{k_1+2s_1+r_1})\right); \quad (*)$$

$$D'_M(k,s) \le 2D'_M(k_1, r_1) + 8r_1 + 2\lceil k_1 \log_2 3 \rceil \le$$
$$\le 2D'_M(k_1, s_1) + 8(r_1 + s_1) + 4\lceil k_1 \log_2 3 \rceil,$$

где $r_1 = \lceil (s + k_1)/2 \rceil$ и $s_1 = \lceil (r_1 + k_1)/2 \rceil = \lceil (s + 3k_1)/4 \rceil$.

Обозначим $k_i = \lceil k_{i-1}/2 \rceil = \lceil k/2^i \rceil$, $r_i = \lceil (s_{i-1} + k_i)/2 \rceil$ и $s_i = \lceil (s_{i-1} + 3k_i)/4 \rceil$. Рекурсивно применяя $(*)$ j раз, получаем

$$M'(k,s) = 3^{\sum_{i=1}^{j}(r_i+s_i)} M'(k_j, s_j) \prod_{i=1}^{j} B_i + O\left(\sum_{i=1}^{j} s_i 3^{s_i+k_i+\sum_{l=1}^{i}(r_l+s_l)} \prod_{l=1}^{i} B_l \right),$$

где $B_i = 3^{k_{i-1}-k_i} + 1$. Выполняется $\prod_{i=1}^{j} B_i = O(3^{k-k_j})$, в чем можно убедиться, переходя к логарифмированию:

$$\sum_{i=1}^{j} \log_3 B_i = k - k_j + \sum_{i=1}^{j} \log_3 (1 + 3^{k_i - k_{i-1}}).$$

В силу неравенства $\log_3(1+x) < x/\ln 3$, справедливого для всех $x > 0$, последняя сумма не превосходит

$$\sum_{i=1}^{j} 3^{k_i - k_{i-1}} \le \sum_{i=1}^{j} 3^{-i} < 1/2.$$

С учетом полученной оценки можно записать,

$$M'(k,s) = O(3^{k-k_j+\sum_{i=1}^{j}(r_i+s_i)}) M'(k_j, s_j) + O\left(3^k \sum_{i=1}^{j} A_i\right),$$

где $A_i = s_i 3^{s_i+\sum_{l=1}^{i}(r_l+s_l)}$. Покажем, что $\sum_{i=1}^{j} A_i = O(A_j)$. Это следует из отношения $A_l/A_{l-1} \geq 9/4$, справедливого для произвольного l:

$$A_l/A_{l-1} = s_l 3^{2s_l+r_l-s_{l-1}}/s_{l-1} > 3^{2r_l+k_l-s_{l-1}}/4 \geq 3^{2k_l}/4 \geq 9/4.$$

Получаем оценку

$$M'(k,s) = O\left(3^{k-k_j+\sum_{i=1}^{j}(r_i+s_i)}\right) M'(k_j,s_j) + O\left(s_j 3^{k+s_j+\sum_{i=1}^{j}(r_i+s_i)}\right). \quad (**)$$

Оценим возникающие в показателях суммы. Из $k_i \leq 1 + (k-1)/2^i$ следует

$$\sum_{i=1}^{j} k_i \leq (k-1)(1-2^{-j}) + j \leq k + j - k_j.$$

Далее, по индукции можно убедиться, что

$$s_i = \left\lceil \frac{s}{4^i} + \frac{3k_1}{4^i} + \frac{3k_2}{4^{i-1}} + \ldots + \frac{3k_i}{4} \right\rceil \leq \frac{s-1}{4^i} + 1 + \sum_{l=1}^{i} \frac{3k_l}{4^{i-l+1}},$$

где индуктивный переход следует из тождества $\lceil \lceil a \rceil / n \rceil = \lceil a/n \rceil$, справедливого для всех $a \in \mathbb{R}$, $n \in \mathbb{N}$. Аналогично проверяется

$$r_i = \left\lceil \frac{2s}{4^i} + \frac{6k_1}{4^i} + \ldots + \frac{6k_{i-1}}{4^2} + \frac{k_i}{2} \right\rceil \leq \frac{2(s-1)}{4^i} + 1 + \sum_{l=1}^{i} \frac{6k_l}{4^{i-l+1}} - k_i,$$

поэтому

$$\sum_{i=1}^{j}(r_i+s_i) \leq (s-1)\left(1-\frac{1}{4^j}\right) + 2j + 3\sum_{i=1}^{j} k_i \left(1 - \frac{1}{4^{j-i+1}}\right) - \sum_{i=1}^{j} k_i \leq$$

$$\leq s + 2j + 2\sum_{i=1}^{j} k_i - \left(1 + \frac{s-1}{4^j} + \sum_{i=1}^{j} \frac{3k_i}{4^{j-i+1}}\right) \leq s + 4j + 2k - 2k_j - s_j.$$

Подставляя полученную оценку в $(**)$, имеем

$$M'(k,s) \leq O(3^{3k+s+4j-3k_j-s_j}) M'(k_j,s_j) + O(s_j 3^{3k+s+4j-2k_j}).$$

Для глубины, как следует из $(*)$, справедлива оценка

$$D'_M(k,s) \leq 2^j D'_M(k_j,s_j) + \sum_{i=1}^{j} \left(2^{i+2}(s_i+r_i) + 2^{i+1}\lceil k_i \log_2 3 \rceil\right). \quad (***)$$

Оценим величину под знаком суммы:

$$2^{i+2}(s_i + r_i) + 2^{i+1}\lceil k_i \log_2 3\rceil < \frac{3s}{2^{i-2}} + 2^{i+3} + 2^i \sum_{l=1}^{i} \frac{9k_l}{4^{i-l}} + 2^{i+1}(1 + k_i \log_2 3).$$

Заменяя k_l на $1 + k/2^l$, получаем оценку сверху в виде

$$\frac{3s}{2^{i-2}} + 2^{i+3} + \frac{9}{2^i}\sum_{l=1}^{i}(4^l + 2^l k) + 2^{i+1}\log_2 6 + k\log_2 6 < \frac{3s}{2^{i-2}} + 12,6\cdot 2^{i+1} + 21,2k.$$

Суммируя по $i = 1, \ldots, j$ и подставляя в $(***)$, окончательно имеем

$$D'_M(k,s) \le 2^j(D'_M(k_j, s_j) + 50,4) + 12s + 21,2jk.$$

Из оценок доказанной леммы и соотношений (3.1), (3.2) вытекает

Следствие 3.1 *Пусть $q = 2^t$ и $j \in \mathbb{N}$, $j \le \lceil \log_2 \log_3 m\rceil$. Тогда для m-кратного умножения многочленов степени $n-1$ над $GF(q)$ можно построить схему сложности и глубины*

$$M(m,n) = O\left(81^j m^{4-\frac{4,5}{2^j}} n^{1-\frac{1}{4^j}}\right) M\left(\lceil m^{\frac{1}{2^j}}\rceil, \lceil m^{\frac{1,5}{2^j}} n^{\frac{1}{4^j}}\rceil\right) + \\ + O\left((81/2)^j m^{4-\frac{2}{2^j}} n \log(mn)\right);$$

$$D_M(m,n) \le 2^j D\left(\lceil m^{\frac{1}{2^j}}\rceil, \lceil m^{\frac{1,5}{2^j}} n^{\frac{1}{4^j}}\rceil\right) + 12\log_2 n + (21,2j + 13,6)\log_2 m + O(2^j).$$

3.4.3 Умножение над полем нечетной характеристики

Умножение в поле нечетной характеристики может выполняться в известной степени более простым алгоритмом — основанным на двоичном, а не на троичном преобразовании Фурье (см. [59]). Опишем модификацию алгоритма из предыдущего пункта для многократного умножения над полем $GF(q)$, где $q = p^t$, $p \in \mathbb{P} \setminus \{2\}$.

Рассмотрим кольцо $R_s^{(x)} = GF(q)[x]/(x^{2^s} + 1)$ многочленов над $GF(q)$ по модулю $x^{2^s} + 1$. Элемент x является примитивным корнем степени 2^{s+1} из единицы в силу взаимной простоты многочленов $x^{2^s} + 1$ и $x^{2^s} - 1$. Таким образом, в кольце $R_s^{(x)}$ определено ДПФ порядка 2^{s+1}. Для обратного преобразования требуется разложение по степеням x^{-1}, которое получается изменением порядка следования коэффициентов и умножением их на -1, т.к.

$$c_0 + \sum_{i=1}^{2^s-1} c_i x^i \equiv c_0 - \sum_{i=1}^{2^s-1} c_{2^s-i}(x^{-1})^i \bmod x^{2^s} + 1,$$

при этом умножение на -1 можно совместить с умножением на $N^{-1} = 2^{-(s+1)} \bmod p$ в конце алгоритма.[4]

Если $\Phi(s,r)$ и $D_\Phi(s,r)$ обозначают сложность и глубину схемы ДПФ порядка 2^r над $R_s^{(x)}$ $(r \le s+1)$, то для обратного преобразования можно построить схему сложности $\Phi(s,r) + 2^s$ и глубины $D_\Phi(s,r) + 1$.

Лемма 3.10 *Пусть $r \le s$. Тогда ДПФ порядка 2^r над $R_s^{(y)}$ реализуется схемой сложности и глубины*

$$\Phi(s,r) \le r2^{r+s}; \qquad D_\Phi(s,r) \le r.$$

Доказательство. Пусть $z = y^{2^{s-r+1}}$ — корень из единицы степени 2^r в $R_s^{(y)}$, и пусть $\Gamma(x) = \sum \gamma_i x^i$ — многочлен над $R_s^{(y)}$ степени $2^r - 1$. Запишем,

$$\Gamma(z^l) = \sum_{i=0}^{2^r-1} \gamma_i z^{il} = \sum_{i=0}^{2^{r-1}-1} \gamma_{2i}(z^2)^{il} + z^l \sum_{i=0}^{2^{r-1}-1} \gamma_{2i+1}(z^2)^{il} = \Gamma_0(z_*^{l_1}) + z^l \Gamma_1(z_*^{l_1}),$$

где $z_* = z^2$ — корень степени 2^{r-1} из единицы, $l_1 = l \bmod 2^{r-1}$, а

$$\Gamma_j(x) = \sum_{i=0}^{2^{r-1}-1} \gamma_{2i+j} x^i, \qquad j = 0, 1.$$

Пусть необходимые значения $\Gamma_j(z_*^i)$, $j = 0,1$, $i = 0, \ldots, 2^{r-1}-1$, найдены при помощи двух ДПФ порядка 2^{r-1}. Умножения на степени z не требуют вычислительных затрат, поэтому для вычисления каждого из 2^r значений $\Gamma(z^l)$ остается выполнить сложение в $R_s^{(y)}$ со сложностью 2^s и глубиной 1. Имеем рекуррентные неравенства

$$\Phi(s,r) \le 2^{s+r} + 2\Phi(s,r-1); \qquad D_\Phi(s,r) \le 1 + D_\Phi(s,r-1),$$

которые, с учетом $\Phi(s,0) = D_\Phi(s,0) = 0$, разрешаются как

$$\Phi(s,r) \le r2^{s+r}; \qquad D_\Phi(s,r) \le r.$$

Через $M'(k,s)$ и $D'_M(k,s)$ обозначим соответственно сложность и глубину 2^k-кратного умножения в кольце $R_s^{(x)}$. Очевидно, справедливы соотношения

$$M(m,n) \le M'(\lceil \log_2 m \rceil, \lceil \log_2 (mn) \rceil);$$
$$D_M(m,n) \le D'_M(\lceil \log_2 m \rceil, \lceil \log_2 (mn) \rceil).$$

[4] В случае характеристики три схемную сложность обратного ДПФ можно принять совпадающей со сложностью прямого ДПФ, поскольку сложность умножения на -1 в поле $GF(3)$ можно полагать равной 0 (если элементы $GF(3)$ записываются в двухбитовом двоичном коде: 1 = [01], 2 = [10], 0 = [00] или [11], то умножение на -1 осуществляется перестановкой разрядов).

В другую сторону, верно

$$M'(k,s) \le M(2^k, 2^s) + 2^s(2^k - 1); \quad D'(k,s) \le D(2^k, 2^s) + k,$$

т.к. приведение многочлена степени $2^k(2^s-1)$ (который, например, является произведением 2^k многочленов степени 2^s-1) по модулю многочлена $x^{2^s}+1$ выполняется сложением (с вычитаниями) 2^k многочленов степени 2^s-1.

Лемма 3.11 *Пусть $k \le s-r+1$, тогда*

$$M'(k, r+s) \le 2^{s+1} M'(k,s) + (2^k+1)(s+1)2^{2s+1} + 2^{k+r+s};$$

$$D'_M(k, r+s) \le D'_M(k,s) + 2s + k + 3.$$

Доказательство. Пусть $f_i \in R^{(x)}_{r+s}$ и $f = f_1 \cdot \ldots \cdot f_{2^k}$. Как и в лемме 3.8, используем представление

$$f_i = \sum_{j=0}^{2^{r+s}-1} f_{i,j} x^j = \sum_{j=0}^{2^r-1} \varphi_{i,j}(\tilde{y}) x^j,$$

где $\tilde{y} = x^{2^r}$ и $\deg \varphi_{i,j} < 2^s$, и устанавливаем соответствие между $R^{(x)}_{r+s}$ и $R^{(y)}_s[x]$. Таким образом, обозначая через $f'_i(x)$ многочлены из $R^{(y)}_s[x]$, соответствующие f_i, и $f'(x) = f'_1(x) \cdot \ldots \cdot f'_{2^k}(x) \in R^{(y)}_s[x]$, получаем соответствие между $f \in R^{(x)}_{r+s}$ и $f'(x) \bmod (x^{2^r} - y)$.

Многочлен $f'(x)$ степени не выше $2^k(2^r-1)$ однозначно восстанавливается по значениям в $2^{s+1} \ge 2^{k+r}$ точках. Рассмотрим следующий алгоритм 2^k-кратного умножения в $R^{(x)}_{r+s}$.

А. С помощью ДПФ вычислим значения многочленов $f'_i(x)$ в точках $1, y, y^2, \ldots, y^{2^{s+1}-1}$ (y — примитивный корень степени 2^{s+1} в $R^{(y)}_s$).

Б. Произведения $f'(y^j) = f'_1(y^j) \cdot \ldots \cdot f'_{2^k}(y^j)$ вычисляются посредством 2^{s+1} 2^k-кратных умножений в $R^{(y)}_s$.

В. Многочлен $f'(x)$ восстанавливается по своим значениям в точках $1, y, y^2, \ldots, y^{2^{s+1}-1}$ при помощи обратного ДПФ.

Г. Многочлен $f'(x)$ приводится по модулю $x^{2^r} - y$.

Приведение по модулю на последнем шаге сводится к сложению (с вычитаниями) 2^k многочленов над $R^{(y)}_s$ степени не выше 2^r-1 (т.к. умножение на степени y в $R^{(y)}_s$ является циклическим сдвигом со сменой знака у некоторых коэффициентов).

Лемма 3.12 *Пусть* $j \in \mathbb{N}$, $j \le \lceil \log_2 k \rceil$, $k_j = \lceil k/2^j \rceil$, $s_j = \lceil \frac{s}{4^j} + \frac{1{,}5k}{2^j} \rceil$. *Тогда*

$$M'(k,s) \le O(2^{3k+s+3j-3k_j-s_j}) M'(k_j, s_j) + O(s_j 2^{3k+s+3j-2k_j});$$

$$D'_M(k,s) \le 2^j (D'_M(k_j, s_j) + 20) + 6s + 9jk.$$

Доказательство. Как и в доказательстве леммы 3.9, будем чередовать сведение к произведениям с меньшим числом сомножителей с двумя итерациями леммы 3.11. Первому шагу отвечают соотношения

$$M'(k+l,s) \le 2^l M'(k,s) + M'(l,s); \quad D'_M(k+l,s) \le D'_M(k,s) + D'_M(l,s),$$

а объединенному:

$$M'(k,s) \le (2^{k-k_1}+1)\left(2^{r_1+s_1+2} M'(k_1,s_1) + O(s_1 2^{k_1+2s_1+r_1})\right);$$

$$D'_M(k,s) \le 2D'_M(k_1,s_1) + 4(k_1+r_1+s_1) + 12, \qquad (\star)$$

где $k_1 = \lceil k/2 \rceil$, $r_1 = \lceil \frac{s+k_1-1}{2} \rceil$ и $s_1 = \lceil \frac{r_1+k_1-1}{2} \rceil = \lceil \frac{s+3k_1-3}{4} \rceil$.

Применим $(\star)$ рекурсивно j раз. Обозначим $k_i = \lceil k_{i-1}/2 \rceil = \lceil k/2^i \rceil$, $r_i = \lceil \frac{s_{i-1}+k_i-1}{2} \rceil$ и $s_i = \lceil \frac{s_{i-1}+3k_i-3}{4} \rceil$. В отношении сложности получаем (вывод аналогичен выводу оценки $(**)$ в лемме 3.9)

$$M'(k,s) = O\left(2^{k-k_j+\sum_{i=1}^{j}(r_i+s_i)+2j}\right) M'(k_j,s_j) + O\left(2^k \sum_{i=1}^{j} A_i\right),$$

где $A_i = s_i 2^{s_i+2i+\sum_{l=1}^{i}(r_l+s_l)}$. Справедливо $\sum_{i=1}^{j} A_i = O(A_j)$, т.к. для произвольного $l < j$ выполняется

$$A_l/A_{l-1} \ge 4 s_l 2^{2s_l+r_l-s_{l-1}}/s_{l-1} \ge 2^{2r_l+(k_l-1)-s_{l-1}} \ge 2^{2(k_l-1)} \ge 4.$$

Следовательно,

$$M'(k,s) = O\left(2^{k-k_j+\sum_{i=1}^{j}(r_i+s_i)+2j}\right) M'(k_j,s_j) + \\ + O\left(s_j 2^{k+\sum_{i=1}^{j}(r_i+s_i)+s_j+2j}\right). \qquad (\star\star)$$

Параметры k_i определяются так же, как и в лемме 3.9. Поэтому справедливо $\sum_{i=1}^{j} k_i \le k + j - k_j$. Также по индукции проверяется, что

$$s_i = \left\lceil \frac{s+1}{4^i} + \frac{3k_1}{4^i} + \frac{3k_2}{4^{i-1}} + \ldots + \frac{3k_i}{4} - 1 \right\rceil \le \frac{s}{4^i} + \sum_{l=1}^{i} \frac{3k_l}{4^{i-l+1}};$$

$$r_i = \left\lceil \frac{2(s+1)}{4^i} + \frac{6k_1}{4^i} + \ldots + \frac{6k_{i-1}}{4^2} + \frac{k_i}{2} - 2 \right\rceil \le \frac{2s}{4^i} - 1 + \sum_{l=1}^{i} \frac{6k_l}{4^{i-l+1}} - k_i.$$

Следовательно,

$$\sum_{i=1}^{j}(r_i + s_i) \le s\left(1 - \frac{1}{4^j}\right) - j - \sum_{i=1}^{j} k_j + 3\sum_{i=1}^{j} k_i \left(1 - \frac{1}{4^{j-i+1}}\right) \le$$

$$\le s - j + 2\sum_{i=1}^{j} k_i - s_j \le s + j + 2k - 2k_j - s_j.$$

Подставляя эту оценку в $(\star\star)$, имеем

$$M'(k,s) \le O(2^{3k+s+3j-3k_j-s_j})M'(k_j,s_j) + O(s_j 2^{3k+s+3j-2k_j}).$$

В отношении глубины справедлива оценка

$$D'_M(k,s) \le 2^j D'_M(k_j,s_j) + \sum_{i=1}^{j} 2^{i+1}(k_i + r_i + s_i + 3). \qquad (\star\star\star)$$

Величина под знаком суммы оценивается следующим образом:

$$2^{i+1}(k_i + r_i + s_i + 3) \le \frac{3s}{2^{i-1}} + 2^{i+2} + 9\sum_{l=1}^{i} \frac{4^l k_l}{2^{i+1}} <$$

$$< \frac{3s}{2^{i-1}} + 2^{i+2} + 9\sum_{l=1}^{i} \frac{4^l + 2^l k}{2^{i+1}} < \frac{3s}{2^{i-1}} + 5 \cdot 2^{i+1} + 9k,$$

т.к. $k_l < 1 + k/2^l$. Подставляя эту оценку в $(\star\star\star)$, окончательно выводим

$$D'_M(k,s) \le 2^j(D'_M(k_j,s_j) + 20) + 6s + 9jk.$$

Следствие 3.2 *Пусть q — нечетно и $j \in \mathbb{N}$, $j \le \lceil \log_2 \log_2 m \rceil$. Тогда для m-кратного умножения многочленов степени $n-1$ над $GF(q)$ можно построить схему сложности и глубины*

$$M(m,n) = O\left(8^j m^{4-\frac{4,5}{2^j}} n^{1-\frac{1}{4^j}}\right) M\left(\lceil m^{\frac{1}{2^j}} \rceil, \lceil m^{\frac{1,5}{2^j}} n^{\frac{1}{4^j}} \rceil\right) +$$

$$+ O\left(4^j m^{4-\frac{2}{2^j}} n \log(mn)\right);$$

$$D(m,n) \le 2^j D\left(\lceil m^{\frac{1}{2^j}} \rceil, \lceil m^{\frac{1,5}{2^j}} n^{\frac{1}{4^j}} \rceil\right) + 6\log_2 n + (9j+7)\log_2 m + O(2^j).$$

3.4.4 Умножение с логарифмической глубиной

С использованием максимального числа $j \approx \log_2 k$ итераций в алгоритмах лемм 3.9 и 3.12, строятся схемы многократного умножения сложности $O(nm^4 \log^c m \log n \log\log n)$ и глубины $O(\log n + \log m \log\log m)$. Глубину можно довести до $O(\log(mn))$, если после нескольких итераций использовать метод из §3.3. При этом, если отталкиваться от метода теоремы 3.2, применение ДПФ ведет к относительному уменьшению мультипликативного коэффициента в оценке глубины. Итак, справедлива

Теорема 3.3 *Пусть $j, l \in \mathbb{N}$, $j \le \lceil \log_2 \log_2 m \rceil$. Тогда m-кратное умножение многочленов степени не выше $n-1$ над $GF(q)$ выполняется схемой сложности и глубины*

$$M(m,n) = O\left(la^j m^{4-\frac{1}{2^j}+\frac{2,5}{l2^j}} n^{1+\frac{1}{l4^j}} \left(2^{-j}\log(mn)\log\log(mn) + l^2\right)\right);$$

$$D_M(m,n) = O\left((l+j)\log m + (1+l/2^j)\log n\right),$$

где $a = 81$, если q четно, и $a = 8$, иначе.

Доказательство. Согласно следствиям из лемм 3.9 и 3.12 сложность многократного умножения $M(m,n)$ над $GF(q)$ оценивается как

$$O\left(a^j m^{4-\frac{4,5}{2^j}} n^{1-\frac{1}{4^j}}\right) M\left(\lceil m^{\frac{1}{2^j}}\rceil, \lceil m^{\frac{1,5}{2^j}} n^{\frac{1}{4^j}}\rceil\right) + O\left((a/2)^j m^{4-\frac{2}{2^j}} n \log(mn)\right).$$

Оценивая $M\left(\lceil m^{\frac{1}{2^j}}\rceil, \lceil m^{\frac{1,5}{2^j}} n^{\frac{1}{4^j}}\rceil\right)$ при помощи теоремы 3.2 как

$$O\left(lm^{\left(3,5+\frac{2,5}{l}\right)\frac{1}{2^j}} n^{\left(1+\frac{1}{l}\right)\frac{1}{4^j}} \left(2^{-j}\log(mn)\log\log(mn) + l^2\right)\right),$$

получаем

$$M(m,n) = O\left(la^j m^{4-\frac{1}{2^j}+\frac{2,5}{l2^j}} n^{1+\frac{1}{l4^j}} \left(2^{-j}\log(mn)\log\log(mn) + l^2\right)\right).$$

В отношении глубины справедливо

$$D_M(m,n) \le 2^j D_M\left(\lceil m^{\frac{1}{2^j}}\rceil, \lceil m^{\frac{1,5}{2^j}} n^{\frac{1}{4^j}}\rceil\right) + O(\log(m^j n)) + O(2^j).$$

Подставляя оценку теоремы 3.2, получаем окончательно

$$D_M(m,n) = 2^j O\left(\frac{l}{2^j}\log\left(mn^{\frac{1}{2^j}}\right)\right) + O(\log(m^j n)) + O(2^j) = \\ = O((l+j)\log m + (1+l/2^j)\log n).$$

Для модулярного многократного умножения из доказанной теоремы и леммы 3.2 получаем следующий результат.

Теорема 3.4 *Пусть* $j, l, r \in \mathbb{N}$, $j \leq \lceil \log_2 \log_2 m \rceil$. *Тогда* m*-кратное умножение многочленов над* $GF(q)$ *по модулю многочлена степени* n *выполняется схемой сложности и глубины*

$$M(m,n) = O\left(la^j m^{1+\frac{1}{r}\left(3-\frac{1}{2^j}+\frac{2,5}{l2^j}\right)} n^{1+\frac{1}{l4^j}} \left(2^{-j}\log(mn)\log\log(mn) + l^2\right)\right);$$

$$D_M(m,n) = O\left((l+j)\log m + r(1+l/2^j)\log n\right),$$

где $a = 81$, *если* q *четно, и* $a = 8$, *иначе.*

3.5 О применении метода Д. Кантора

Концепция аддитивных подгрупп конечного поля и основанного на них аддитивного аналога дискретного преобразования Фурье, позволяющего построить асипмтотически быстрый метод умножения многочленов над конечными полями, была предложена Д. Кантором в [46]. Впоследствии, в [58], метод Кантора был обобщен на более широкий класс полей, и одновременно упрощено изложение. Приведем необходимые сведения о применении аддитивного преобразования Фурье, опираясь на [58].

Пусть в поле $GF(q^k)$ выбран базис $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$. Положим $W_0 = \{0\}$ и обозначим $W_i =< \alpha_1, \ldots, \alpha_i >$ — линейное подпространство поля $GF(q^k)$, т.е. W_i является аддитивной подгруппой. Очевидно, для $i < k$ выполняется

$$W_{i+1} = \bigcup_{c \in GF(q)} \{c\alpha_{i+1} + W_i\}.$$

Определим многочлены $S_i(x)$ как

$$S_i(x) = \prod_{\beta \in W_i} (x - \beta).$$

Имеют место следующие легко проверяемые свойства многочленов $S_i(x)$ (см. [58]):

(1) $S_0(x) \mid S_1(x) \mid \ldots \mid S_k(x) = x^{q^k} - x$;

(2) $S_{i+1}(x) = S_i(x)(S_i^{q-1}(x) - S_i^{q-1}(\alpha_{i+1}))$;

(3) Многочлен $S_i(x)$ может иметь ненулевые коэффициенты только при степенях x^{q^j}, $0 \leq j \leq i$;

(4) Выполнено $S_i(\beta + \gamma) = S_i(\beta) + S_i(\gamma)$ для любых $\beta, \gamma \in GF(q^k)$.

Указанные свойства позволяют сформулировать аддитивный аналог алгоритма быстрого дискретного преобразования Фурье. Обозначим через n_i количество ненулевых коэффициентов многочлена $S_i(x)$. Из (3) следует, что $n_i \le i+1$. Нам также потребуется следующая лемма.

Лемма 3.13 *Пусть $\alpha \in GF(q^k)$, $g(x) \in GF(q^k)[x]$ и $\deg g \le q^{i+1}-1$. Для любого $c \in GF(q)$ обозначим $g_c(x) = g(x) \bmod S_i(x - c\alpha_{i+1} - \alpha)$. Тогда*

(а) *Для всех $\beta \in \alpha + W_i$ выполнено $g(\beta + c\alpha_{i+1}) = g_c(\beta + c\alpha_{i+1})$;*

(б) $g(x) = -S_i^{1-q}(\alpha_{i+1}) \sum_{c \in GF(q)} g_c(x) \frac{S_{i+1}(x-\alpha)}{S_i(x-c\alpha_{i+1}-\alpha)}$, *а для поля характеристики 2:* $g(x) = S_i^{-1}(\alpha_{i+1})(g_0(x) S_i(x - \alpha_{i+1} - \alpha) + g_1(x) S_i(x-\alpha))$;

(в) *Если известен многочлен $g(x)$, то многочлены $g_c(x)$ вычисляются со сложностью $O(n_i q^{i+2} M(k))$ и глубиной $O(q(D_M(k) + \log n_i))$;*

(г) *Если известны многочлены $g_c(x)$, то многочлен $g(x)$ вычисляется со сложностью $O(n_i q^{i+1} M(k))$ и глубиной $O(q(D_M(k) + \log n_i))$;*

Доказательство. Утверждение (а) очевидно. Равенство (б) многочленов степени не выше $q^{i+1}-1$ следует из совпадения их значений в q^{i+1} точках $\beta \in \alpha + W_{i+1}$.

Многочлен $S_i(x)$ является n_i-членом степени q^i со следующим после старшего ненулевым коэффициентом при x^{q^j}, где $j \le i-1$. Многочлен вида $S_i(x-\beta)$ отличается от $S_i(x)$ постоянным членом $S_i(\beta)$. Приведение по модулю такого многочлена (или умножение на него) можно свести к вычитаниям (сложениям).

Алгоритм приведения многочлена $g(x)$ по модулю $S_i(x-\beta)$ состоит в последовательных приведениях по модулям $S_i(x-\beta)x^{j(q^i-q^{i-1})}$, $j = q-1, \ldots, 1$. Каждое из приведений требует $O(q^i n_i)$ умножений на коэффициенты многочлена $S_i(x-\beta)$ и $O(q^i n_i)$ сложений над $GF(q^k)$ (для приведения подобных). Эти сложения выполняются с глубиной $O(\log n_i)$, т.к. при каждой степени x образуется $O(n_i)$ подобных членов. Окончательно, одно приведение по модулю $S_i(x-\beta)$ выполняется со сложностью $O(n_i q^{i+1} M(k))$ и глубиной $O(q(D_M(k) + \log n_i))$, что доказывает (в).

Умножение $g_c(x)$ на многочлен $-S_i^{1-q}(\alpha_{i+1}) S_{i+1}(x-\alpha)$, имеющий $n_{i+1}+1$ ненулевых коэффициентов, требует $O(q^i n_{i+1})$ умножений и $O(q^i n_{i+1})$ сложений в $GF(q^k)$. Сложения выполняются с глубиной $O(\log n_{i+1})$, т.к. при каждой степени x имеется $O(n_{i+1})$ подобных членов. Последующее деление на многочлен $S_i(x - c\alpha_{i+1} - \alpha)$ выполняется алгоритмом предыдущего абзаца с тем отличием, что находится не остаток, а частное от деления; также,

за счет того, что исходный многочлен может иметь степень $q^{i+1}+q^i-1$, требуется, вообще говоря, не $q-1$, а q промежуточных операций приведения по модулю. После вычисления всех многочленов-слагаемых в формуле (б), как указано выше, остается выполнить сложение q многочленов степени не выше $q^{i+1}-1$, откуда, учитывая $n_{i+1} \le 2n_i$, получаем (г).

Лемма 3.14 *Вычисление всех $g(\beta)$, $\beta \in \alpha+W_l$, где $g \in GF(q^k)[x]$, $\deg g \le q^l-1$, $\alpha \in GF(q^k)$, выполняется схемой сложности и глубины*

$$O(l^2 q^{l+1} M(k)); \qquad O(ql(D_M(k)+\log l)).$$

С такой же сложностью и глубиной восстанавливается единственный многочлен $g(x)$ степени не выше q^l-1, принимающий заданные значения на множестве $\alpha+W_l$.

Доказательство. Воспользуемся индукцией. Для $l=0$ утверждение леммы очевидно. Пусть оно справедливо для всех $l \le L-1$. Рассмотрим переход к $l=L$.

Согласно п. (а) леммы 3.13, вычисление всех значений многочлена $g(x)$ степени не выше q^L-1 на множестве $\alpha+W_L$ равносильно вычислению значений многочленов $g_c(x)$ степени не выше $q^{L-1}-1$ на соответствующих множествах $\alpha+c\alpha_L+W_{L-1}$, где $c \in GF(q)$. Из п. (в) следует, что многочлены $g_c(x)$ вычисляются со сложностью $O(n_{L-1}q^{L+1}M(k))$ и глубиной $O(q(D_M(k)+\log n_i))$. По предположению индукции, вычисление значений многочленов g_c на соответствующих подмножествах выполняется схемой сложности $O(q^{L+1}M(k))\sum_{i=0}^{L-2} n_i$ и глубины $O(q(L-1)D_M(k)) + O(q)\sum_{i=0}^{L-2}\log n_i$. Суммируя указанные оценки и замечая, что из $n_i \le i+1$ следует

$$\sum_{i=0}^{l-1} n_i \le C_l^2; \qquad \sum_{i=0}^{l-1} \log n_i \le \log l! = O(l\log l),$$

получаем первое утверждение леммы. Второе утверждение доказывается аналогично.

Замечание. Вычисление значений многочлена степени $n < q^l-1$ на множестве W_l методом леммы 3.14 выполняется схемой со сложностью $O(q^{l+1}M(k)\log_q^2 n)$ и глубиной $O(q\log_q n(D_M(k)+\log\log_q n))$, т.к. нет необходимости вычислять остатки от деления на $S_i(x-\beta)$, если $q^i \ge n$.

Теорема 3.5 *Пусть $q=O(mn)$. Тогда умножение m многочленов степени не выше $n-1$ над $GF(q)$ выполняется схемой сложности и глубины*

$$M(m,n) = O(m^2q^2nM(\log_q(mn))\log_q^2 n);$$

$$D_M(m,n) = O(\, q \log_q(mn)(\, D_M(\log_q(mn)) + \log\log_q(mn))\,).$$

Доказательство. Пусть перемножаются многочлены $f_1(x), \ldots, f_m(x) \in GF(q)[x]$ степени не выше $n-1$. Выберем $k = \lceil \log_q(m(n-1)+1) \rceil$, так что любой многочлен степени $m(n-1)$ может быть восстановлен по своим значениям в поле $GF(q^k)$. Примем, что $f_i(x) \in GF(q^k)[x]$, и обозначим $f(x) = f_1(x) \cdot \ldots \cdot f_m(x)$. Рассмотрим следующую схему вычислений.

A. Вычисляются значения всех многочленов $f_i(x)$ в точках поля $GF(q^k)$.

B. Вычисляются значения $f(x)$ на элементах поля $GF(q^k)$ непосредственным перемножением значений многочленов-сомножителей.

C. Многочлен $f(x)$ восстанавливается по известным значениям.

Согласно замечанию к лемме 3.14, шаг A реализуется схемой сложности $O(mq^{k+1}M(k)\log_q^2 n)$ и глубины $O(q\log_q n(D_M(k)+\log\log_q n))$. Шаг B состоит в выполнении q^k m-кратных умножений в $GF(q^k)$ и реализуется со сложностью $O(q^k m M(k))$ и глубиной $O(D_M(k)\log m)$. Шаг C, согласно лемме 3.14, может быть выполнен схемой сложности $O(k^2 q^{k+1} M(k))$ и глубины $O(qk(D_M(k)+\log k))$. В оценке сложности доминирует сложность реализации шага A, а в оценке глубины — глубина реализации шага C, откуда, учитывая $q^k < qmn$, получаем окончательный результат.

Оценки доказанной теоремы могут быть уточнены при специальном выборе базиса (см. [46]). Например, при $q = 2$ они могут быть приведены к виду $O(m^2 n \log^{\log_2 6}(mn))$ для сложности и $O(\log(mn)\log\log(mn))$ для глубины. Аналогичным образом, как и в п. 3.4.4, может быть построена схема глубины $O(\log(mn))$.

4 Инвертирование

Ряд асимптотически быстрых способов инвертирования в стандартном базисе конечного поля основывается на расширенном алгоритме Евклида (базовый алгоритм Евклида предназначен для вычисления НОД элементов некоторого кольца). Если $m_n(t)$ — характеристический многочлен некоторого стандартного представления поля $GF(q^n)$, то по заданному ненулевому элементу поля, представленному многочленом $f(t)$, используя расширенный алгоритм Евклида, можно найти многочлен $u(t)$ (степени обоих многочленов меньше n), такой, что выполняется $f(t)u(t) = 1 \bmod m_n(t)$, следовательно, $u(t)$ представляет обратный к $f(t)$ элемент. Более подробно об алгоритме Евклида см. [2, п. 8.8], [5, п. 1.2.3], [16, разд. 4.5].

Для варианта этого алгоритма, принадлежащего Кнуту [72] и оптимизированного Шёнхаге [81], справедлива оценка сложности $O(n \log^2 n \log\log n)$ и глубины $O(n)$. Алгоритм Шёнхаге работает с числами, переложение на многочлены было выполнено Монком [75], однако в его работе имеются неточности, исчерпывающий и корректный анализ этих алгоритмов был проведен в работах [43, 88] (см. также [59, гл. 11]).

Указанные алгоритмы предназначены для реализации программами или, например, схемами из автоматных элементов. При реализации схемой из функциональных элементов приходится использовать более громоздкие построения, но добиться тех же порядку оценок сложности и глубины возможно.

Из известных методов синтеза схем над $GF(q)$, по-видимому, наиболее общим является метод аддитивных цепочек, который позволяет строить схемы сложности $O(n^w \log n)$ и глубины $O(\log^2 n)$, где $w < 1,667$ — экспонента умножения матриц размера $\sqrt{n} \times \sqrt{n}$ и $\sqrt{n} \times n$. Подробный анализ этого метода см. в §4.1. В §4.2 предложен способ построения схем с логарифмической глубиной, основанный на использовании многократных умножений: показано, что для любого $r \in \mathbb{N}$ можно построить схему сложности $O(rn^{w+1/r})$ и глубины $O(r \log n)$ (в частном случае $r \sim \log n$ получаются оценки из метода аддитивных цепочек).

Для конкретных полей со специальными базисами разработаны практически более эффективные методы: некоторые из этих методов имеют теоретические оценки сложности $O(n^{1+\epsilon})$ и одновременно глубины, меньшей $\log_2^2 n$ (но такие схемы строятся не для всех n), см., например, работы [13, 30, 6] и по ссылкам к ним. В §4.3 показано, как для некоторых базисов (в частности, гауссовых нормальных базисов) строить схемы инвертирования сложности $O(\epsilon^{-b} n^{1+\epsilon})$ и глубины $O(\epsilon^{-1} \log n)$, где $b < 2,12$.

Далее, в §4.4 выясняется вопрос о минимизации глубины схемы инвер-

тирования в поле по основанию два. Показано, что инвертирование в поле $GF(2^n)$ можно реализовать схемой глубины асимптотически $(3+\sigma)\log_2 n$, где σ — константа глубины многократного сложения, которая определяется как наименьшее число, такое, что существует схема сложения n одноразрядных чисел, имеющая глубину $(\sigma + o(1))\log_2 n$. Сложность схемы инвертирования из §4.4 достаточно высока, порядка n^4. Похожий вид, $(1+\sigma)\log_2 n$, имеет наилучшая известная асимптотическая оценка глубины умножения n-разрядных чисел. Известно [63], что $\sigma < 3,44$. Умножение чисел в методе [63] (основанном на [78]) выполняется схемой сложности $O(n^2)$ и глубины асимптотически $4,44\log_2 n$.

Через $M(n)$ будет обозначаться порядок сложности умножения многочленов степени $n-1$ над $GF(q)$ с глубиной $O(\log n)$. Известно, что $M(n) = O(n\log n\log\log n)$ (см. п. 2.3).

4.1 Метод аддитивных цепочек

4.1.1 Аддитивные цепочки

Аддитивной цепочкой для числа n называется любая начинающаяся с 1 последовательность натуральных чисел $a_0 = 1,\ a_1,\ \ldots,\ a_m = n$, в которой каждое число является суммой каких-то двух предыдущих чисел (возможно совпадающих), т.е. для всех $i \geq 1$ выполнено $a_i = a_j + a_k,\ j,k < i$. Под *длиной* цепочки $a_0,\ a_1,\ \ldots,\ a_m$ понимается число m. Через $l(n)$ обозначается длина *кратчайшей* аддитивной цепочки для n. Теория аддитивных цепочек подробно изложена в [16, п. 4.6.3], [5, п. 4.6].

Возведение в степень может быть интерпретировано как построение аддитивной цепочки для показателя степени. *Удвоения* в аддитивной цепочке соответствуют возведениям в квадрат, а прочие сложения — умножениям.

Аддитивная цепочка называется *линейной*, если каждый ее элемент равен сумме предыдущего элемента и какого-то еще, т.е. для всех $i \geq 1$ выполнено $a_i = a_{i-1} + a_j,\ j < i$. Длина кратчайшей линейной цепочки обозначается через $l^*(n)$.

Метод Брауэра [42] (см. также [16] или [5]) позволяет по заданной линейной цепочке $1,\ a_1,\ \ldots,\ a_m = n$ построить аддитивную цепочку для $2^n - 1$ следующим образом. Выписывается последовательность

$$1 = 2^1 - 1,\ 2^{a_1} - 1,\ \ldots,\ 2^{a_m} - 1 = 2^n - 1,$$

затем в промежутки между числами вставляются последовательности удвоений. Так, между соседними числами $2^{a_i} - 1$ и $2^{a_{i+1}} - 1$, где $a_{i+1} = a_i + a_j$,

помещается последовательность

$$2(2^{a_i}-1),\ 2^2(2^{a_i}-1),\ \ldots,\ 2^{a_{i+1}-a_i}(2^{a_i}-1)=2^{a_{i+1}}-2^{a_j}.$$

Поскольку $2^{a_{i+1}}-1=(2^{a_{i+1}}-2^{a_j})+(2^{a_j}-1)$, то итоговая последовательность является линейной аддитивной цепочкой длины $n+m-1$, состоящей из $n-1$ удвоений и m сложений.

Метод Брауэра можно использовать и с произвольной цепочкой для n. При этом, если $a_i=a_j+a_k$, то для вычисления $2^{a_i}-1$ последовательность удвоений необходимо вести от $2^{a_j}-1$ или от $2^{a_k}-1$. Количество сложений в построенной цепочке также будет совпадать с длиной исходной цепочки для n, но количество удвоений может быть больше, чем $n-1$. Цепочки, для которых методом Брауэра строится цепочка ровно с $n-1$ удвоениями, называются цепочками Ханзена (линейные цепочки являются частным случаем цепочек Ханзена, см. [16]). В цепочке, построенной методом Брауэра с исходной цепочкой Ханзена, все нелинейные шаги являются удвоениями.

Из метода Брауэра следует, что

$$l^*(2^n-1)\le n+l^*(n)-1.$$

Эта теорема обобщается также на цепочки Ханзена, но гипотеза Шольца—Брауэра, утверждающая, что верно неравенство $l(2^n-1)\le n+l(n)-1$, по-видимому, до сих пор не доказана и не опровергнута. Также, видимо, не доказано неравенство $l(n)\ge\lambda(n)+\lambda(\nu(n))$, где $\lambda(n)=\lfloor\log_2 n\rfloor$, а $\nu(n)$ — вес числа n, т.е. количество единиц в его двоичной записи, хотя А. Шёнхаге в 1975 г. [82] доказал, что $l(n)\ge\lambda(n)+\lambda(\nu(n))-O(1)$ (см. также [16]).

Известно (это тоже теорема Ханзена), что для любого c существуют n, такие, что $l(n)<l^*(n)-c$, но практически требование линейности не накладывает существенных ограничений, поскольку для всех $n\le 12508$ выполняется $l(n)=l^*(n)$, т.е. для разумных значений n всегда существует линейная аддитивная цепочка минимальной длины (см. [16]).

Почти для всех n минимальная цепочка имеет длину асимптотически $\lambda(n)(1+1/\lambda(\lambda(n)))$. Для произвольного n цепочка такой длины может быть построена 2^k-арным методом Брауэра [42] (см. также [16]). Более точно, цепочка Брауэра имеет длину

$$\lambda(n)\left(1+\frac{1}{\lambda(\lambda(n))}+O\left(\frac{\log\log\log n}{\log^2\log n}\right)\right).$$

Такая же оценка может быть получена методом Яо [90]. Интересно, что методы Брауэра и Яо фактически являются двойственными друг к другу в

смысле принципа транспозиции (см. [38]). Разработанные позднее алгоритмы построения асимптотически минимальных аддитивных цепочек в той или иной степени основаны на методах Брауэра и Яо (соответствующие обзоры можно найти в [61, 38]).

Понятие *глубины* для аддитивной цепочки вводится аналогично как и для схем: в действительности, аддитивная цепочка моделируется схемой над базисом из элементов сложения и удвоения. Очевидно, минимальная глубина аддитивной цепочки для n равна $\lceil \log_2 n \rceil$. Цепочку такой глубины и длины $\lambda(n) + \nu(n) - 1$ можно построить вариантом бинарного метода. При этом для некоторых n не существует цепочек минимальной глубины и длины, меньшей, чем $2\lambda(n)$. Однако, вариант метода Яо позволяет построить цепочку глубины $\lceil \log_2 n \rceil + 1$ и асимптотически минимальной длины $\lambda(n)\left(1 + 1/\lambda(\lambda(n)) + O(\log\log\log n / \log^2 \log n)\right)$. В действительности, для всех $n \leq 228$ существуют цепочки минимальной длины $l(n)$ и глубины, не превосходящей $\lceil \log_2 n \rceil + 1$. Более подробное изложение соответствующих вопросов имеется в [12].

4.1.2 Метод Брауэра

Идея применения аддитивных цепочек к задачам возведения в степень и инвертирования высказывалась явно во многих работах, и во многих работах использовалась неявно. Она реализуется в методе Брауэра, однако этот метод практически не упоминается в выполненных после 1939 года работах. Так, часто цитируемый (например, [34]) метод инвертирования Ито—Цуйи [68] является частным случаем метода Брауэра (см. [61, 12]). Многие модификации метода Ито—Цуйи, например, [89], также накрываются методом Брауэра (см. [12]).

Инвертирование в произвольном конечном поле $GF(q^n)$ можно свести к возведению в степень специального вида [68] (см. также [5, п. 4.6.3]). Согласно теореме Ферма, для всех $x \in GF(q^n)$ справедливо $x^{q^n} = x$. Обозначим $Q = (q^n - q)/(q - 1)$. Тогда

$$x^{-1} = x^Q (x^{Q+1})^{-1}.$$

Следует заметить, что $x^{Q+1} \in GF(q)$, т.к. $(Q + 1)(q - 1) = q^n - 1$. Таким образом, инвертирование сведено к возведению в степень Q, умножению x^Q на x (оно выполняется проще, чем умножение вообще, поскольку известно, что результат принадлежит полю $GF(q)$, и достаточно вычислить только одну компоненту произведения) и делению x^Q на $x^{Q+1} \in GF(q)$. Если $q = 2$, то инвертирование просто совпадает с возведением в степень $Q = 2^n - 2$.

Метод Брауэра сводит задачу возведения в степень Q в поле $GF(q^n)$ к построению кратчайшей аддитивной цепочки для $n-1$:

$$1 = a_0,\ \ldots,\ a_{l(n-1)} = n-1.$$

Затем строится цепочка для $Q/q = (q^{n-1}-1)/(q-1)$, состоящая из чисел

$$(q^{a_i}-1)/(q-1),$$

в промежутки между которыми вставлены операции умножения на числа q^{a_j} (аналогично случаю $q=2$, см. п. 4.1.1). В силу

$$q^{a_j}(q^{a_i}-1)/(q-1) + (q^{a_j}-1)/(q-1) = (q^{a_i+a_j}-1)/(q-1),$$

построенная (вообще говоря, не аддитивная) цепочка состоит из $l(n-1)$ сложений и такого же количества умножений на степени q — последние операции соответствуют преобразованиям Фробениуса.

В нормальном базисе операции Фробениуса выполняются с нулевой схемной сложностью, поэтому справедлива

Теорема 4.1 *Пусть M_n — схема умножения в нормальном базисе поля $GF(q^n)$. Тогда для инвертирования можно построить схему I_n сложности и глубины*

$$L(I_n) \le (l(n-1)+1)L(M_n) + n; \qquad D(I_n) \le (d(n-1)+1)D(M_n) + 1,$$

где $d(n-1)$ — глубина минимальной аддитивной цепочки для $n-1$. Также можно построить схему сложности и глубины

$$\lambda(n)\left(1 + \frac{1}{\lambda(\lambda(n))} + O\left(\frac{\log\log\log n}{\log^2 \log n}\right)\right) L(M_n);$$

$$(\lceil \log_2(n-1) \rceil + 2)D(M_n) + 1.$$

При использовании стандартного базиса операции Фробениуса могут выполняться со сложностью $O(n^w) + O(\sqrt{n}M(n))$ и глубиной $O(\log n)$ методом Брента—Кунга [44], где $w < 1,667$ — экспонента матричного умножения $T_{\sqrt{n},\sqrt{n},n}$ (более подробно о методе Брента—Кунга см. далее в §5.1). Справедлива

Теорема 4.2 *Пусть M_n — схема умножения в стандартном базисе поля $GF(q^n)$; сложность и глубина схемы, реализующей произвольную операцию Фробениуса, удовлетворяют оценкам $S(n)$ и $D_S(n)$ соответственно.*

Тогда для инвертирования можно построить схему I_n сложности и глубины

$$L(I_n) \le (l(n-1)+1)(L(M_n)+S(n))+n;$$
$$D(I_n) \le (d(n-1)+1)(D(M_n)+D_S(n))+1,$$

где $d(n-1)$ — глубина минимальной аддитивной цепочки для $n-1$. В частности, можно построить схему сложности и глубины

$$O(n^w \log n + \sqrt{n} M(n) \log n); \qquad O(\log^2 n),$$

где w — экспонента умножения матриц размера $\sqrt{n} \times \sqrt{n}$ и $\sqrt{n} \times n$.

Таким образом, для схемы инвертирования в стандартном базисе поля $GF(q^n)$ имеем оценки сложности и глубины $O(n^{1,667})$ и $O(\log^2 n)$. Оценки для инвертирования в нормальном базисе зависят от метода умножения. Из результатов главы 5 следует, что можно построить схему инвертирования в произвольном нормальном базисе сложности $O(n^{1,806})$ и глубины $O(\log^2 n)$. Однако, например, для оптимальных нормальных базисов (см. ниже) известная оценка сложности умножения по порядку такая же, как для стандартных базисов, и для инвертирования в таких базисах методом аддитивных цепочек строится схема сложности $O(n \log^2 n \log\log n)$ и глубины $O(\log^2 n)$, т.е. с такими же по порядку оценками, как у алгоритма Евклида для стандартных базисов (но вопрос о схемной сложности последнего остается открытым).

В работе [61] для инвертирования в поле $GF(q^n)$, $q > 2$, применяется схема, не использующая деления в $GF(q)$ и имеющая несколько большую сложность. Эта схема основана на использовании формулы

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

Вычисление $x^{q^{n-1}-1}$ выполняется методом Брауэра, как указано выше.

Более подробно о применении метода аддитивных цепочек к инвертированию в конечных полях см. [61, 12].

4.2 К построению параллельных схем

В схемах, описанных в предыдущем параграфе, используется $O(\log n)$ умножений и операций Фробениуса, выполняемых последовательно, что приводит к оценке глубины $O(\log^2 n)$. В настоящем параграфе рассматривается способ организации вычислений, в котором в каждой цепочке выполняется $O(1)$ многократных умножений и операций Фробениуса, что позволяет реализовать инвертирование схемой глубины $O(\log n)$.

Стандартный (параллельный) способ возведения в некоторую степень

$$P = a_1 q^{e_1} + \ldots + a_r q^{e_k}, \qquad 0 < a_i < q,$$

состоит в следующем. Параллельно реализуются k операций Фробениуса $x_i = x^{q^{e_i}}$, $i = 1, \ldots, k$. Затем выполняется a-кратное умножение, где $a = \sum a_i$:

$$x^P = x_1^{a_1} \cdot \ldots \cdot x_k^{a_k}.$$

Как и в п. 4.1.2, вместо инвертирования в поле $GF(q^n)$ мы рассматриваем возведение в степень

$$Q = (q^n - q)/(q-1) = q^{n-1} + q^{n-2} + \ldots + q.$$

Стандартный способ возведения в такую степень требует выполнение $n-1$ операций Фробениуса и $n-1$-кратного умножения, что приводит к схеме слишком большой сложности, если сравнивать со схемой из теоремы 4.2. Далее излагается метод, позволяющий существенно уменьшить порядок сложности.

Лемма 4.1 *Пусть $m = \lceil \sqrt[r]{n} \rceil$, $r \in \mathbb{N}$. Пусть M_n и $M_{m,n}$ — соответственно схемы обычного и m-кратного умножения в стандартном базисе поля $GF(q^n)$; сложность и глубина схемы, реализующей произвольную операцию Фробениуса, удовлетворяют оценкам $S(n)$ и $D_S(n)$ соответственно. Тогда для операции возведения в степень $Q = (q^n - q)/(q-1)$ в поле $GF(q^n)$ можно построить схему I'_n сложности и глубины*

$$L(I'_n) \le (2r-1)(mS(n) + L(M_{m,n})) + (r-1)L(M_n);$$

$$D(I'_n) \le 2(D_S(n)+D(M_{m,n}))+D(M_n)+(r-2)\max\{D_S(n)+D(M_{m,n}), D(M_n)\}.$$

Доказательство. Пусть $n-1 = [n_r, n_{r-1}, \ldots, n_1]_m$ в системе счисления с основанием $m = \lceil \sqrt[r]{n} \rceil$.

Для $i = 1, \ldots, r-1$ положим

$$N_i = 1 + q^{m^{i-1}} + q^{2m^{i-1}} + \ldots + q^{(m-1)m^{i-1}} = \frac{q^{m^i} - 1}{q^{m^{i-1}} - 1},$$

так что выполняется

$$N_1 \cdot N_2 \cdot \ldots \cdot N_i = q^{m^i} - 1.$$

Далее, пусть $P_i = 0$, если $n_i = 0$, и

$$P_i = q^{[n_r, \ldots, n_{i+1}, 0, \ldots, 0]_m + 1} \left(1 + q^{m^{i-1}} + \ldots + q^{(n_i - 1)m^{i-1}}\right)$$

в противном случае (в квадратных скобках записано r-разрядное число).

Заметим, что возведение в любую из степеней P_i, N_i выполняется не сложнее, чем за m операций Фробениуса и одно m-кратное умножение. Суммируя тождества

$$N_1 \cdot \ldots \cdot N_{i-1} P_i = (q^{m^{i-1}} - 1) P_i = q(q^{[n_r,\ldots,n_i,0,\ldots,0]_m} - q^{[n_r,\ldots,n_{i+1},0,\ldots,0]_m})$$

по всем $i = 1, \ldots, r-1$, выводим

$$Q = P_1 + N_1(\ldots(P_{r-1} + N_{r-1}P_r)\ldots).$$

В соответствии с этой формулой можно построить такую схему для возведения элемента $x \in GF(q^n)$ в степень Q. На первом уровне x возводится в степени P_1 и N_1; на следующем x^{N_1} возводится в степени P_2 и N_2; на третьем одновременно с возведением $x^{N_1 N_2}$ в степени P_3 и N_3 можно выполнить умножение x^{P_1} на $x^{N_1 P_2}$ и т.д. На i-м уровне производится возведение $x^{N_1 \cdot \ldots \cdot N_{i-1}}$ в степени P_i и N_i, и выполняется умножение $x^{N_1 \cdot \ldots \cdot N_{i-2} P_{i-1}}$ на $x^{P_1 + N_1(\ldots(P_{i-3} + N_{i-3}P_{i-2})\ldots)}$. На $(r+1)$-м уровне вычисляется x^Q как произведение вычисленных на предыдущем уровне $x^{P_1 + N_1(\ldots(P_{r-2} + N_{r-2}P_{r-1})\ldots)}$ и $x^{N_1 \cdot \ldots \cdot N_{r-1} P_r}$.

Всего в данной процедуре используется $r-1$ умножений и $2r-1$ возведений в степени P_i, N_i. На первых двух уровнях производятся только возведения в степень, на последнем — только умножения, а на остальных операции обоих видов выполняются параллельно.

В рассмотренной только что схеме используется

$$(r-1)m + n_1 + \ldots + n_r \leq (2r-1)\sqrt[r]{n} + O(r)$$

операций Фробениуса, такова же суммарная кратность операций многократного умножения (величина $O(r)$ складывается из погрешностей округлений). Эта оценка поддается незначительному уточнению.

Рассмотрим разложение числа $n-1$ в смешанной системе счисления с основанием $[m_{r-1}, \ldots, m_1]$, которое устроено следующим образом:

$$n - 1 = m_{r-1}(m_{r-2}(\ldots(m_1 n_0 + n_1)\ldots) + n_{r-2}) + n_{r-1},$$

где $0 \leq n_i < m_i$. С использованием такого разложения можно построить алгоритм возведения в степень Q, аналогичный алгоритму леммы 4.1, использующий

$$\sum_{i=1}^{r-1} m_i + \sum_{i=0}^{r-1} n_i$$

операций Фробениуса (и многократные умножения с такой суммарной кратностью). Покажем по индукции, что основание системы счисления можно подобрать таким образом, что указанная величина не превосходит

$$2r\sqrt[r]{n/2} + O(r) = (2r - \ln 4 + O(1/r))\sqrt[r]{n} + O(r).$$

В случае $r = 1$ утверждение очевидно. Предположим, что оно доказано для всех $r \le k$. Рассмотрим случай $r = k + 1$.

Положим $m_k = \lceil \mu\sqrt[r]{n} \rceil$ (параметр μ будет выбран позже) и запишем

$$n - 1 = m_k n' + n_k,$$

где также выполнены неравенства $n_k < \mu\sqrt[r]{n}$ и $n' \le \mu^{-1}\sqrt[r]{n^k}$. Тогда, по предположению, n' может быть представлено в виде

$$m_{k-1}(m_{k-2}(\ldots(m_1 n_0 + n_1)\ldots) + n_{k-2}) + n_{k-1},$$

где

$$\sum_{i=1}^{k-1} m_i + \sum_{i=0}^{k-1} n_i \le 2k\sqrt[k]{n'/2} + O(k) \le \frac{2k\sqrt[r]{n}}{\sqrt[k]{2\mu}} + O(k),$$

откуда следует

$$\sum_{i=1}^{k} m_i + \sum_{i=0}^{k} n_i \le \left(2\mu + \frac{2k}{\sqrt[k]{2\mu}}\right)\sqrt[r]{n} + O(r).$$

При условии $\mu > 0$ величина в скобках достигает своего наименьшего значения $2r/\sqrt[r]{2}$, когда $\mu = 1/\sqrt[r]{2}$.

Основным результатом параграфа является следующая теорема, уточняющая вывод предшествующих работ [73, 57]:

Теорема 4.3 *Пусть $r \in \mathbb{N}$. Тогда инвертирование в стандартном базисе поля $GF(q^n)$ реализуется схемой I_n сложности и глубины*

$$L(I_n) = O(rn^{1/r}(n^w + n^{1,5}\log n \log\log n)), \qquad D(I_n) = O(r\log n),$$

где w — экспонента матричного умножения $T_{\sqrt{n},\sqrt{n},n}$.

Доказательство. Такая схема строится методом леммы 4.1, в котором операции Фробениуса реализуются схемами сложности $O(n^w + \sqrt{n}M(n))$ и глубины $O(\log n)$, а m-кратные умножения — схемами из следствия 3.4 сложности $O(m^a n^b)$, где $a + b \le 2,5$, и глубины $O(\log(mn))$. Поскольку $m < n$ и $w \ge 1,5$, в оценке сложности доминируют операции Фробениуса.

Из доказанной теоремы, в частности, вытекает, что можно построить схему сложности $O(n^{1,667})$ и глубины $O(\log n)$.

4.2.1 О методах Литоу—Давида и фон цур Гатена

Инвертирование в поле $GF(2^n)$ по методу Литоу—Давида [73], основанное на матричном представлении элементов поля, включает восстановление коэффициентов многочлена степени n по его корням, которые кодируются $O(n^2)$ битами, что означает вычисление элементарных симметрических функций от n чисел, содержащих $O(n^2)$ разрядов каждое. В частности, вычисляется произведение этих n чисел.

Метод фон цур Гатена [57], также использующий матричное представление, предназначен для конечных полей наиболее общего вида $GF(q^n)$. В случае $q = 2$ элементарный алгоритм инвертирования можно непосредственно вывести из результата работы [52]. Как и выше, запишем

$$x^{-1} = x^{2^n-2} = x^2 \cdot x^{2^2} \cdot \ldots \cdot x^{2^{n-1}}.$$

Произведение соответствующих элементам x^{2^i} многочленов $f_i(t)$ сводится, согласно [52], к вычислению числового произведения

$$f_1(2^L) \cdot f_2(2^L) \cdot \ldots \cdot f_{n-1}(2^L),$$

где L приблизительно равно $n \log n$. Таким образом, требуется перемножить $n - 1$ чисел, содержащих порядка $n^2 \log n$ разрядов.

Как в [73], так и в [52] предлагается использовать результат из [45]. Сложность схемы, выполняющей с логарифмической глубиной перемножение n чисел, кодирующихся n битами, в работе [45] оценивается как $O(n^5 \log^2 n)$. Мультипликативный коэффициент в оценке глубины не приводится, но элементарный анализ показывает, что он не меньше 15.

4.3 Инвертирование в базисах с низкой транзитивной сложностью

Асимптотически основная сложность построенной в §4.2 схемы инвертирования в стандартном базисе содержится в подсхеме, выполняющей операции Фробениуса. Однако эти операции выполняются «бесплатно» (при схемной реализации), если вычисления производятся в нормальном базисе. Таким образом, для стандартного базиса, допускающего сравнительно несложный переход к нормальному базису и обратно (под переходом подразумевается преобразование координат элемента поля), оценка теоремы 4.3 может быть улучшена.

Следуя [4], *транзитивной сложностью* стандартного (нормального) базиса поля $GF(q^n)$ назовем сумму сложностей схем прямого и обратно-

го перехода к нормальному (стандартному) базису.[5] Будем обозначать ее через $T(n)$. Через $D_T(n)$ обозначим сумму глубин соответствующих схем.

Таким образом, справедливы очевидные оценки

$$S(n) \le T(n); \qquad D_S(n) \le D_T(n).$$

Теорема 4.4 *Пусть $R \in \mathbb{N}$, $R = o(\log n / \log\log n)$. Пусть схемы T' и T'' реализуют соответственно прямой и обратный переходы между нормальным и стандартным базисами поля $GF(q^n)$. Тогда для инвертирования в любом из указанных базисов можно построить схему I_n сложности и глубины*

$$L(I_n) = O(R^b n^{1+2/R}) + O(R\sqrt[R]{n})(L(T') + L(T''));$$

$$D(I_n) = O(R(\log n + D(T') + D(T''))),$$

где $b = (4/3)\log_2 3$, если q четно, и $b = 1$, если q нечетно.

Доказательство. Используем схему из леммы 4.1 со следующей стратегией: операции Фробениуса реализуем в нормальном базисе, а умножения — в стандартном, выполняя переходы между базисами при необходимости.

Схему m-кратного умножения $M_{m,n}$ построим методом следствия 3.4. Для нее справедливы оценки:

$$L(M_{m,n}) = O\left(la^j m^{1+\frac{1}{r}\left(3-\frac{1}{2^j}+\frac{2,5}{l2^j}\right)} n^{1+\frac{1}{l4^j}} \left(2^{-j}\log(mn)\log\log(mn) + l^2\right)\right);$$

$$D(M_{m,n}) = O\left((l+j)\log m + r(1+l/2^j)\log n\right),$$

где l, j, r — натуральные параметры, а $a \in \{8,\ 81\}$.

Пусть $m = O(\sqrt[R]{n})$. Определим параметры $r = O(1)$, $l = O(\sqrt[3]{R})$, $j \sim (1/3)\log_2 R$ так, чтобы выполнялось

$$1 + \frac{1}{l4^j} + \frac{1}{R}\left(1 + \frac{1}{r}\left(3 - \frac{1}{2^j} + \frac{2,5}{l2^j}\right)\right) \le 1 + \frac{1,5}{R}.$$

Учитывая, что при условии $R = o(\log n / \log\log n)$ также выполнено

$$2^j l^2,\ \log(mn)\log\log(mn) = o(n^{0,5/R}),$$

имеем

$$L(M_{m,n}) = O(a^{(1/3)\log_2 R} n^{1+2/R}); \qquad D(M_{m,n}) = O(\log n).$$

[5] Уточним, что здесь не требуется, чтобы оба базиса порождались одним и тем же элементом.

Подставляя эти оценки в лемму 4.1, получаем окончательно утверждение теоремы.

Из доказанной теоремы вытекает, что в базисах с почти линейной транзитивной сложностью инвертирование также выполняется с почти линейной сложностью. При этом, если соответствующие переходы между базисами выполняются с глубиной $O(\log n)$, то может быть построена схема инвертирования глубины $O(\log n)$.

4.3.1 Оптимальные нормальные базисы

Оптимальные нормальные базисы впервые были описаны в [76]. Напомним, что *оптимальным* называется базис минимально возможной сложности $2n-1$ (см. п. 2.3). Различают три типа оптимальных нормальных базисов (ОНБ), а теорема Гао—Ленстры (см. в [69, гл. 5]) утверждает, что других ОНБ не существует.

ОНБ I типа существуют в полях $GF(q^n)$, когда число $n+1$ — простое, а q — примитивный корень по модулю $n+1$. В качестве генератора ОНБ может быть выбран какой-либо из примитивных корней степени $n+1$ из единицы в этом поле.

ОНБ II и III типа возникают, когда $q = 2^m$ и $(m, n) = 1$, число $2n+1$ — простое, и либо 2 является примитивным корнем по модулю $2n+1$ (II тип), либо n — нечетно, а примитивным корнем по модулю $2n+1$ является -2 (III тип). В качестве генератора выбирается элемент $\alpha = \zeta + \zeta^{-1}$, где ζ — примитивный корень степени $2n+1$ из единицы в поле $GF(q^{2n})$.

Умножение в ОНБ реализуется методом Месси—Омура со сложностью $O(n^2)$. Однако, как следует из работы [4], для выполнения умножения в ОНБ еще большее значение имеет то, что они обладают малой транзитивной сложностью. Более точно, для базисов I типа $T(n) = O(n)$, а для базисов II и III типов $T(n) = O(n \log n)$.

ОНБ I типа с точностью до перестановки элементов совпадает с базисом $\{\alpha, \alpha^2, \ldots, \alpha^n\}$, и для перехода к стандартному базису (или обратно) остается выразить α^n (или 1) из соотношения $\alpha^n + \ldots + \alpha + 1 = 0$. В этом случае $T(n) = O(n)$ и $D_T(n) = O(\log n)$.

ОНБ II или III типа является перестановкой элементов $\{a_1, \ldots, a_n\}$, где $a_i = \zeta^i + \zeta^{-i}$. Справедливо соотношение $a_k a_m = a_{k+m} + a_{k-m}$, из которого следует, что при любом i базисные подмножества $\{a_1, \ldots, a_i\}$ и $\{\alpha, \ldots, \alpha^i\}$ выражаются друг через друга (обозначим сложность соответствующего

преобразования координат через L_i). Пусть $2^k \le n < 2^{k+1}$, тогда

$$x = \sum_{i=1}^{n} x_i a_i = a_{2^k} \sum_{i=1}^{n-2^k} x_{2^k+i} a_i + \sum_{i=1}^{2^k} (x_{2^{k+1}-i} + x_i) a_i,$$

где $x_j \equiv 0$ при $j > n$. Учитывая, что $a_{2^k} = \alpha^{2^k}$, можно вывести рекуррентное неравенство $L_n \le L_{2^k} + L_{n-2^k} + n - 2^k$, откуда $L_n = O(n \log n)$ и, следовательно, $T(n) = O(n \log n)$ (а $D_T(n) = O(\log n)$) с учетом соотношения $m(\alpha) = 0$, позволяющего выражать α^n или 1 через остальные базисные элементы, где m — минимальный многочлен для α. Более подробное изложение см. в [4], [5, гл. 5].

Таким образом, для сложности умножения в этих базисах справедлива оценка $O(n \log n \log\log n)$, а практически приемлемые схемы можно строить методом Карацубы. В отношении инвертирования, как следует из теоремы 4.4, справедливо

Утверждение 4.1 *Для любого $\epsilon > 0$, $\epsilon = \Omega(\log\log n / \log n)$, существует схема инвертирования в ОНБ поля $GF(q^n)$ сложности $O(\epsilon^{-b} n^{1+\epsilon})$ и глубины $O(\epsilon^{-1} \log n)$.*

4.3.2 Гауссовы нормальные базисы

Оптимальные нормальные базисы являются частным случаем гауссовых нормальных базисов (ГНБ)[6], которые впервые были рассмотрены в [35][7].

ГНБ 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 $\mathbb{Z}_{kn+1} \setminus \{0\}$.

Известно, что сложность ГНБ k-го типа не превосходит $k(n+1)-1$ (см., например, [5, п. 5.3]), а в некоторых частных случаях можно получить еще лучшие оценки (см. [69, гл. 3]).

---

[6] ГНБ получили название от своих порождающих элементов, известных как *гауссовы периоды*, которые рассматривал Гаусс при решении задачи о построении правильных многоугольников при помощи циркуля и линейки.

[7] В работе [35] рассматривались только поля характеристики 2 — обобщение для произвольных полей появилось несколько позже.

Для ускорения реализации операций в ГНБ, как показано в [56], можно использовать переход к стандартному (или почти стандартному) базису в поле $GF(q^{kn})$, расширении поля $GF(q^n)$. Действительно, элементы ГНБ являются суммами различных элементов базиса $\{\zeta, \zeta^2, \ldots, \zeta^{kn}\}$ поля $GF(q^{kn})$ (см. [5]). Для перехода к стандартному базису $\{1, \zeta, \ldots, \zeta^{kn-1}\}$ остается выразить ζ^{kn} из условия $1+\zeta+\ldots+\zeta^{kn} = 0$, что можно выполнить со сложностью $O(kn)$ и глубиной $\log(kn) + O(1)$ (подробнее см. в [5]).[8]

Таким образом, умножение в ГНБ выполняется схемой сложности $O(kn \cdot \log(kn) \log\log(kn))$. Для инвертирования справедливо следствие из теоремы 4.4 (с учетом того, что понятие транзитивной сложности распространяется на переход к базису в расширении исходного поля):

Утверждение 4.2 *Пусть $k = o(\log n)$ и $\epsilon > 0$, $\epsilon = \Omega(\log\log n / \log n)$. Тогда можно построить схему инвертирования в ГНБ k-го типа поля $GF(q^n)$ сложности $O(\epsilon^{-b} n^{1+\epsilon})$ и глубины $O(\epsilon^{-1} \log n)$.*

В [53] предложено обобщение конструкции гауссовых нормальных базисов, которое имеет практическое значение (см. [61]). Такие базисы также допускают сравнительно простой переход к стандартному базису в некотором расширении поля, но удовлетворительных теоретических оценок для схем перехода в общем случае, кажется, пока не получено.

4.4 Глубина инвертирования в поле $GF(2^n)$

В данном параграфе будет рассмотрен вопрос о минимизации мультипликативной постоянной при главном члене в оценке глубины схемы инвертирования — при этом анализ будет ограничен полями с основанием два. Будет показано, что для инвертирования в поле $GF(2^n)$ можно построить схему глубины $(6,44 + o(1)) \log_2 n$.

Рассмотрим метод из §3.2, модифицированный для возведения в степень. Пусть задан элемент x в стандартном представлении, требуется вычислить x^E, где $E = 2^{e_1} + 2^{e_2} + \ldots + 2^{e_m}$. Можно считать, что $E < 2^n - 1$ (т.к. справедливо тождество Ферма $x^{2^n} = x$) и, следовательно, что $m < n$.

1. Вычислим степени $x^{2^{e_1}}, \ldots, x^{2^{e_m}}$. Пусть элементу $x^{2^{e_i}}$ соответствует многочлен $f_i(t)$ в представлении поля. Пусть далее $f(t) = f_1(t) \cdot \ldots \cdot f_m(t)$. Положим $L = m(n-1)+1$, выберем поле $GF(2^k)$, содержащее не менее L элементов; в нем выберем набор элементов $\alpha_1, \ldots, \alpha_L$.

[8] В действительности, для выполнения операций не обязательно избавляться от ζ^{kn} — в таком случае операции следует производить с многочленами степени kn без свободных членов (в частности, то же верно для ОНБ).

2. Вычислим всевозможные $f_i(\alpha_j) \in GF(2^k)$, где $i = 1, \ldots, m$, $j = 1, \ldots, L$.

3. Для всех j вычислим произведения $f_1(\alpha_j) \cdot \ldots \cdot f_m(\alpha_j) = f(\alpha_j)$. Для этого в поле $GF(2^k)$ выберем примитивный элемент α. Если $f_i(\alpha_j) \neq 0$ для всех i, тогда

3.1. Вычислим дискретные логарифмы, $\log_\alpha f_i(\alpha_j)$.

3.2. Вычислим $\sum_{i=1}^{m} \log_\alpha f_i(\alpha_j) \bmod (2^k - 1) = \log_\alpha f(\alpha_j)$.

3.3. Вычислим $f(\alpha_j) = \alpha^{\log_\alpha f(\alpha_j)}$.

4. По известным значениям $f(\alpha_j)$, $j = 1, \ldots, L$, восстанавливается многочлен $f(t)$ степени не выше $L - 1$.

5. Элементу x^E соответствует многочлен $f(t) \bmod m_n(t)$.

Теорема 4.5 *Пусть m — вес числа E. Тогда можно построить схему $E_{m,n}$, реализующую операцию возведения в степень E в поле $GF(2^n)$, со сложностью и глубиной (при $\epsilon > 0$)*

$$L(E_{n,m}) \leq (1 + o(1)) \frac{\log_2(mn) + C_0(\epsilon)}{\log_2(m^2 n)} \cdot m^2 n^2 + C_1(\epsilon) m^{2+\epsilon} n^{1+\epsilon};$$

$$D(E_{n,m}) \leq (2 + \epsilon) \log_2 n + 4,44 \log_2 m + O(\log^2 \log n) + C_2(\epsilon),$$

где C_i — некоторые ограниченные на любом отрезке интервала $(0, 1]$ функции.

Доказательство. Положим $k = \log_2(mn) + C_0(\epsilon)$, где вид функции C_0 будет определен позже.

Как и при доказательстве теоремы 3.1 заметим, что шаги 1, 2, 4, 5 являются линейными преобразованиями. Композиция преобразований, выполняемых на шагах 1–2, имеет размерность $mLk \times n$, а преобразований, выполняемых на шагах 4–5 — размерность $n \times Lk$. Согласно лемме 2.2 указанные преобразования реализуются схемами сложности и глубины $(1 + o(1))mLkn/\log_2(mLk)$, $\log_2 n + O(1)$ и $(1 + o(1))nLk/\log_2 n$, $\log_2(Lk) + O(1)$ соответственно.

Рассмотрим шаг 3, который заключается в выполнении L операций m-кратного умножения в поле $GF(2^k)$. Алгоритм многократного умножения, описанный шагами 3.1–3.3, корректно вычисляет произведение ненулевых элементов. Поэтому дополнительно требуется проверка, есть ли нулевой элемент среди сомножителей, и обнуление результата в таком случае. Соответствующая корректирующая схема имеет линейную сложность $O(mk)$ и асимптотически минимальную глубину $\log_2(mk) + O(1)$.

Дискретное логарифмирование (шаг 3.1) будет подробно рассмотрено в §4.4.1, где будет показано как выбрать функцию $C_0(\epsilon)$ (и соответственно

$k = \log_2 L + C_0(\epsilon)$) так, что существует схема дискретного логарифмирования в поле $GF(2^k)$ сложности и глубины

$$C_1(\epsilon)O(L^\epsilon); \qquad \epsilon \log_2 L + C_3(\epsilon) + O(\log\log L).$$

На шаге 3.2 выполняется m-кратное сложение k-разрядных чисел по модулю $2^k - 1$. Метод многократного сложения, разработанный Г. К. Столяровым [28], Ю. П. Офманом [14] и несколькими зарубежными авторами (см. [78]) в начале 60-х гг., основан на использовании схем, называемых *компрессорами*. Схема-компрессор служит для преобразования нескольких слагаемых к меньшему их числу с сохранением суммы, и при этом обладает константной, не зависящей от числа разрядов, глубиной.

Простейшим примером такой схемы является (3,2)-компрессор. Если даны три k-разрядных числа: $a = [a_{k-1}, \ldots, a_0]$, $b = [b_{k-1}, \ldots, b_0]$, $c = [c_{k-1}, \ldots, c_0]$ (старшинство разрядов возрастает справа налево), то сумму $a_i + b_i + c_i$ можно представить в виде $2u_i + v_i$, где

$$v_i = a_i \oplus b_i \oplus c_i, \quad u_i = a_i \cdot b_i \oplus b_i \cdot c_i \oplus a_i \cdot c_i.$$

Так количество слагаемых сокращается с трех до двух: $a + b + c = u + v$, где $u = [u_{k-1}, \ldots, u_0, 0]$, $v = [v_{k-1}, \ldots, v_0]$. Пара разрядов (u_i, v_i) вычисляется со сложностью 5 и глубиной 3, следовательно, сложность компрессора равна $5k$, а глубина — 3.

Из компрессоров можно построить схему, которая с глубиной $O(\log m)$ преобразует m чисел на входах в $O(1)$ чисел на выходах с сохранением суммы. Окончательно, полученные числа могут быть сложены посредством обычных сумматоров.

Если (как в нашем случае) k-разрядные числа складываются по модулю $2^k - 1$, то получаемые в процессе вычислений старшие разряды следует перемещать на место младших. Например, модулярный (3,2)-компрессор должен возвращать числа $u' = [u_{k-2}, \ldots, u_0, u_{k-1}]$ и $v = [v_{k-1}, \ldots, v_0]$, где u_i, v_i определяются как выше.

Вероятно, наилучшая известная оценка глубины схемы компрессоров, сводящей m-кратное сложение к сложению двух чисел, $3,44 \log_2 m + O(1)$, получена в работе [63] методом из работы [78]. Сложность такой схемы $O(mk)$. Константы под знаком «O» в этих оценках достаточно велики — практически, можно строить схемы глубины $3,71 \log_2 m$ и сложности $5mk$ из (3,2)-компрессоров.

Обыкновенный k-разрядный сумматор можно реализовать схемой линейной сложности и глубины $\log_2 k + O(\sqrt{\log k})$ методом В. М. Храпченко [31]. Однако на практическом интервале значений k лучше работают другие методы, например, троичный метод М. И. Гринчука (см. в [10]).

Сложение двух k-разрядных чисел по модулю $2^k - 1$ можно свести к обычному сложению $2k$-разрядных чисел. Действительно, если $a, b \leq 2^k - 1$, то $a + b \bmod (2^k - 1) = c + d$, где $a + b = c2^k + d$ и $c + d \leq 2^k - 1$. Результат $a + b \bmod (2^k - 1)$ содержится в разрядах с k-го по $(2k - 1)$-й (нумерация с 0) суммы чисел $(2^k + 1)a$ и $(2^k + 1)b$.

Таким образом, можно построить схему m-кратного k-разрядного сумматора по модулю $2^k - 1$ сложности и глубины

$$O(mk); \qquad 3,44 \log_2 m + O(\log k).$$

Для экспоненцирования, выполняемого на шаге 3.3, лемма 3.5 позволяет построить схему сложности $O(kM(k))$ и глубины $O(\log^2 k)$.

Окончательно, суммируя оценки по всем шагам, получаем для сложности схемы оценку

$$L(E_{m,n}) \leq (1 + o(1)) \frac{\log_2(mn) + C_0(\epsilon)}{\log_2(m^2 n)} \cdot m^2 n^2 + C_1(\epsilon) m^{2+\epsilon} n^{1+\epsilon},$$

которая определяется сложностью линейного преобразования, выполняемого на шаге 1–2, а также, при определенных значениях ϵ, подсхемой дискретных логарифмов. Для глубины имеем оценку

$$\begin{aligned} D(E_{n,m}) \leq \log_2 n + \log_2(Lk) + \epsilon \log_2 L + C_3(\epsilon) + 3,44 \log_2 m + O(\log^2 k) = \\ = (2 + \epsilon) \log_2 n + 4,44 \log_2 m + C_2(\epsilon) + O(\log^2 \log n). \end{aligned}$$

Теорема 4.6 *Инвертирование и деление в поле $GF(2^n)$ реализуются схемами сложности $(2/3 + o(1))n^4$ и глубины $(6,44 + o(1)) \log_2 n$.*

Доказательство. Инвертирование в поле $GF(2^n)$ совпадает с возведением в степень $2^n - 2$ веса $n - 1$. Схема инвертирования строится методом теоремы 4.5, где ϵ выбирается в пределах погрешности округления константы из работы [63] до $3,44$.

При вычислении y/x умножение на y можно интегрировать в схему, реализующую x^{-1}: для этого используется алгоритм из начала параграфа с параметром $m = n$, где вместо одного из многочленов $f_i(t)$ подставляется многочлен $g(t)$, представляющий элемент y. Оценки теоремы 4.5 остаются в силе и при такой модификации.

Замечание. Теорема 4.5 и следствие 4.6 доказывались для стандартного представления конечного поля, однако их оценки справедливы в любом базисе, т.к. переход между базисами является линейным преобразованием, композиция с которым не изменяет размерности преобразований, выполняемых на начальном и заключительном шагах рассмотренных алгоритмов.

Отметим, что мультипликативная постоянная в асимптотике глубины построенных схем инвертирования и деления в действительности имеет вид $3+\sigma$, где $\sigma < 3,44$ — константа многократного сложения.

4.4.1 Дискретное логарифмирование

В этом параграфе рассматривается (несколько более подробно, чем требуется для обоснования результата теоремы 4.5) способ построения схемы дискретного логарифма в поле $GF(2^k)$, который фактически является реализацией алгоритма Сильвера—Полига—Хеллмана (см., например, [17]) в виде схемы из функциональных элементов.

Будем придерживаться следующих обозначений. Пусть α — основание логарифма. Входом схемы является элемент β, отличный от нуля. Результат вычислений есть двоичное число $b < 2^k - 1$, такое, что $\beta = \alpha^b$.

Пусть $q \cdot r = 2^k - 1$, $(q, r) = 1$. На подгруппе Z_r корней r-й степени из единицы поля $GF(2^k)$ может быть корректно определена операция логарифмирования по основанию порождающего элемента подгруппы, которым является α^q. Заметим, что

$$\beta^q = (\alpha^b)^q = (\alpha^q)^{b \bmod r},$$

иначе говоря, $\log_{\alpha^q} \beta^q = b \bmod r$, где $b = \log_\alpha \beta$.

Теперь пусть известно некоторое разложение $2^k - 1$ на взаимно простые сомножители

$$2^k - 1 = r_1 \cdot r_2 \cdot \ldots \cdot r_s.$$

Введем обозначение $q_i = (2^k - 1)/r_i$, где $i = 1, \ldots, s$. Справедливо

$$\beta^{q_i} = (\alpha^{q_i})^{b \bmod r_i}.$$

Таким образом, сравнивая β^{q_i} со всевозможными степенями элемента α^{q_i} (достаточно выполнить r_i таких сравнений), можно определить остаток $b_i = b \bmod r_i$. По набору остатков $b_i, i = 1 \ldots s$, число b восстанавливается однозначно. Рассмотрим следующую схему вычислений.

АЛГОРИТМ DL.

1. Вычислим все $\beta_i = \beta^{q_i}$, $i = 1, \ldots, s$.

2. Для всех i среди $j = 0, \ldots, r_i - 1$ покоэффициентным сравнением элементов β_i и α^{jq_i} (последние вычислены предварительно) найдем b_i, удовлетворяющее $\beta_i = \alpha^{b_i q_i}$.

3. Число $b = \log_\alpha \beta$ восстанавливается по своим остаткам $b_i = b \bmod r_i$.

Лемма 4.2 *Операция вычисления набора $\beta^{q_1}, \ldots, \beta^{q_s}$, $\beta \in GF(2^k)$, реализуется схемой сложности $O(skM(k))$ и глубины $O(\log^2 k)$.*

Доказательство. Сначала вычисляются все степени вида β^{2^l}, $l = 0, \ldots, k-1$ (далее в п. 5.5.4 показано, что такая операция реализуется со сложностью $O(kM(k))$ и глубиной $O(\log k)$), а затем выполняется s многократных умножений кратности не выше $k-1$ (одно многократное умножение реализуется схемой сложности $O(kM(k))$ и глубины $O(\log^2 k)$, если использовать стандартный метод).

Указанные оценки можно получить также на основе теории аддитивных цепочек (см., например, [16]). Кроме того, с использованием материала главы 3, можно построить схему глубины $O(\log k)$.

Рассмотрим шаг 3, состоящий в применении китайской теоремы об остатках.

Лемма 4.3 *Операция восстановления числа b, $0 \le b < 2^k - 1$ по заданным остаткам $b_i = b \bmod r_i$, $i = 1, \ldots, s$ реализуется схемой сложности $O(k^2)$ и глубины $O(\log k)$.*

Доказательство. Согласно китайской теореме об остатках (см., например, [2, 16, 24])

$$b = b_1 \cdot c_1 + b_2 \cdot c_2 + \ldots + b_s \cdot c_s \mod 2^k - 1,$$

$$c_i = \nu_i r_1 \cdot \ldots \cdot r_{i-1} \cdot r_{i+1} \cdot \ldots \cdot r_s = \frac{\nu_i (2^k - 1)}{r_i},$$

где нормирующий коэффициент $\nu_i \in [1, r_i - 1]$ подбирается, исходя из условия $c_i = 1 \bmod r_i$. По построению, числа c_i состоят не более чем из k двоичных разрядов.

Рассмотрим следующий способ вычисления (близкий к [36]). Пусть $b_i = (b_{i,j-1}, b_{i,j-2}, \ldots, b_{i,0})$ в двоичном представлении, $j = \lceil \log_2 r_i \rceil$, тогда

$$b_i \cdot c_i = b_{i,0} \cdot c_i + 2b_{i,1} \cdot c_i + \ldots + 2^{j-1} b_{i,j-1} \cdot c_i.$$

Вычисление слагаемых в приведенной формуле осуществляется «бесплатно» — также «бесплатно» выполняется приведение их по модулю $2^k - 1$ (старшие разряды подставляются на место младших). Поступим так с каждым из произведений $b_i \cdot c_i$, $i = 1, \ldots, s$. Количество вновь образованных слагаемых оценивается как

$$\sum_{i=1}^{s} \lceil \log_2 r_i \rceil < s + \sum_{i=1}^{s} \log_2 r_i = s + \log_2 (2^k - 1) < k + s.$$

Задача сведена к суммированию не более чем $k+s$ экземпляров k-разрядных чисел по модулю 2^k-1, для сложности и глубины которого с учетом очевидного неравенства $s<k$ справедливы оценки $O(k^2)$ и $O(\log k)$ соответственно (см. доказательство теоремы 4.5).

В работе [64] предложен метод построения схемы сложности $O(k^{1+\epsilon})$ и глубины $O(\epsilon^{-1}\log k)$, где $\epsilon>0$ (вариант этого метода, применяемый к умножению многочленов, описан в §3.3). По всей видимости, метод [64] не превосходит стандартный метод леммы 4.3 в части глубины.

В рассматриваемой схеме дискретного логарифмирования самым трудоемким оказывается блок, соответствующий шагу 2. Справедлива

Лемма 4.4 *Дискретное логарифмирование в подгруппе Z_r поля $GF(2^k)$ можно реализовать схемой $\Lambda_{k,r}$ сложности и глубины*

$$L(\Lambda_{k,r})<r\left(2+\frac{k}{\log_2 r}\cdot\frac{\log_2 r+6}{\log_2 r-\log_2\log_2 r}\right);$$

$$D(\Lambda_{k,r})\le\lceil\log_2 k\rceil+\lceil\log_2 r\rceil+1.$$

Порождающий элемент группы Z_r обозначим через α_r. Пусть требуется вычислить $c=\log_{\alpha_r}\beta_r$, где $\beta_r\in Z_r$. Схема строится из следующих двух подсхем. Первая подсхема сравнивает β_r со всевозможными элементами α_r^l, $l=0,\ldots,r-1$, составляющими всю группу Z_r (фактически реализуется набор компараторов). Вторая подсхема имеет входами выходы компараторов первой подсхемы и выдает номер l компаратора, в котором произошло сравнение — такая схема называется *шифратором*.

Компаратор k-разрядного элемента β_r с фиксированным элементом поля $GF(2^k)$ есть некоторая обобщенная конъюнкция разрядов β_r (под компаратором здесь понимается схема, определяющая совпадение или несовпадение двух наборов).

Разобьем набор из k переменных (которыми кодируется β_r) на поднаборы, содержащие не более s переменных. Для каждого поднабора построим схему, реализующую все возможные обобщенные конъюнкции этой группы переменных (она называется *дешифратором*). Чтобы получить необходимые r конъюнкций k переменных, требуется еще не более $r(\lceil k/s\rceil-1)$ конъюнкций, соединяющих соответствующие выходы дешифраторов. Следующая лемма фактически содержится в [23].

Лемма 4.5 *Можно построить схему K_s дешифратора s переменных со сложностью и глубиной*

$$L(K_s)<2^s+3{,}81\cdot 2^{\frac{s}{2}};\quad D(K_s)\le\lceil\log_2 s\rceil+1.$$

Доказательство. Разбиваем множество переменных на две части: они равны, когда s четно, и отличаются на 1 в нечетном случае. Пусть для них построено два дешифратора. Тогда с помощью 2^s конъюнкций объединяем всевозможными способами выходы этих подсхем.

Участвующие в этой конструкции дешифраторы меньшего порядка устроены точно так же. Дешифратор одной переменной включает в себя лишь один функциональный элемент отрицания: $L(K_1) = 1$, $D(K_1) = 1$. Оценим сложность простейших дешифраторов:

$$L(K_2) = 2^2 + 2L(K_1) = 6, \quad L(K_3) = 2^3 + L(K_1) + L(K_2) = 15,$$

$$L(K_4) = 2^4 + 2L(K_2) = 28, \quad L(K_5) = 2^5 + L(K_2) + L(K_3) = 53,$$

$$L(K_6) = 2^6 + 2L(K_3) = 94, \quad L(K_7) = 2^7 + L(K_3) + L(K_4) = 171,$$

$$L(K_8) = 2^8 + 2L(K_4) = 312.$$

В этих случаях заявленная оценка сложности выполняется; константа 3,81 получается при $s = 7$.

Проверку утверждения при $s > 8$ проведем по индукции. Отметим, что приводимая ниже выкладка корректна как для четного ($\delta = 0$), так и для нечетного случая ($\delta = 0.5$).

$$\begin{aligned} L(K_s) &\le 2^s + L(K_{\frac{s}{2}-\delta}) + L(K_{\frac{s}{2}+\delta}) < \\ &< 2^s + 2^{\frac{s}{2}}(2^\delta + 2^{-\delta}) + 3,81 \cdot 2^{\frac{s}{4}}(2^{\frac{\delta}{2}} + 2^{-\frac{\delta}{2}}) < \\ &< 2^s + \frac{3}{\sqrt{2}}2^{\frac{s}{2}} + 3,81 \cdot \frac{1+\sqrt{2}}{\sqrt[4]{2}}2^{\frac{s}{4}} < 2^s + 3,81 \cdot 2^{\frac{s}{2}}, \end{aligned}$$

если $2^{\frac{s}{4}} > 4,6$, что выполняется при $s \ge 9$.

Легко проверяется, что глубина построенной схемы дешифратора равна $\lceil \log_2 s \rceil + 1$.

Следствие 4.1 *Набор из r k-разрядных компараторов с общим входом реализуется схемой $Q_{k,r}$ сложности и глубины*

$$L(Q_{k,r}) < \frac{kr}{\log_2 r} \cdot \frac{\log_2 r + 6}{\log_2 r - \log_2 \log_2 r}; \quad D(Q_{k,r}) \le \lceil \log_2 k \rceil + 2.$$

Доказательство. Пользуясь доказанной леммой, оценим общую сложность схемы сравнения как

$$L(Q_{k,r}) \le \frac{k}{s}(L(K_s) + r) \le \frac{k}{s}\left(2^s + 3,81 \cdot 2^{\frac{s}{2}}\right) + \frac{k}{s}r.$$

Положим $s = \lceil \log_2 r - \log_2 \log_2 r \rceil$, тогда оценка примет вид:

$$L(Q_{k,r}) < \frac{k}{\log_2 r - \log_2 \log_2 r} \left(\frac{2r}{\log_2 r} + 3,81 \sqrt{\frac{2r}{\log_2 r}} + r \right) <$$
$$< \frac{kr}{\log_2 r (\log_2 r - \log_2 \log_2 r)} \left(2 + 3,81 \sqrt{\frac{2 \log_2 r}{r}} + \log_2 r \right) <$$
$$< \frac{kr}{\log_2 r} \cdot \frac{\log_2 r + 6}{\log_2 r - \log_2 \log_2 r},$$

так как $2 \log_2 r \leq r$.

Глубина отдельного компаратора и, следовательно, всей схемы, не превосходит $\lceil \log_2 k \rceil + 2$.

Следующая схема по r входам (выходы компараторов), только один из которых может принимать значение 1, вычисляет номер данного входа.

Сопоставим выходу каждой схемы сравнения β_r и α_r^l номер l, точнее, его двоичную запись. Назовем *частной дизъюнкцией разряда* h дизъюнкцию всех входов с номерами, h-й разряд которых равен 1. Заметим, что частная дизъюнкция выходов компараторов произвольного разряда h вычисляет h-й разряд числа $c = \log_{\alpha_r} \beta_r$. Действительно, получая на входе элемент β_r, только один компаратор принимает значение 1, а именно тот, который отмечен номером c. Если h-й разряд c равен 1, то выход соответствующего компаратора участвует в частной дизъюнкции разряда h, она, следовательно, равна 1. Иначе, если h-й разряд c равен 0, выход компаратора не подается на вход данной дизъюнкции, которая поэтому принимает значение 0. Таким образом, вычисление всего набора частных дизъюнкций дает двоичное представление $\lceil \log_2 r \rceil$-разрядного числа c. Иначе говоря, выходы шифратора реализуют частные дизъюнкции всех разрядов, вплоть до $\lceil \log_2 r \rceil$-го.

Лемма 4.6 *Шифратор с r входами реализуется схемой V_r сложности и глубины*

$$L(V_r) \leq 2r - 2\lceil \log_2 r \rceil - 2; \qquad D(V_r) = \lceil \log_2 r \rceil - 1.$$

Доказательство. Вначале докажем утверждение в случае $r = 2^s$, т.е. индуктивно построим схему V_{2^s}, для которой выполняются оценки сложности и глубины

$$L(V_{2^s}) \leq 2^{s+1} - 2s - 2; \qquad D(V_{2^s}) = s - 1.$$

При $s = 1$ входы схемы кодируются одним битом; единственная дизъюнкция совпадает в данном случае со входом, отмеченным единицей, т. е. $L(V_{2^1}) = 0, D(V_{2^1}) = 0$.

Рассмотрим переход от s к $s+1$. В зависимости от значения старшего $(s+1)$-го разряда кода (0 или 1) все входы можно разбить на две группы, каждая содержит по 2^s входов. Для каждой из групп реализуем систему s частных дизъюнкций младших разрядов. Соединив соответствующие выходы этих подсхем элементами дизъюнкций получим все правильные частные дизъюнкции для полного набора 2^{s+1} входов, за исключением дизъюнкции $(s+1)$-го разряда.

Частная дизъюнкция старшего разряда объединяет все входы одной из подгрупп. Для ее вычисления можно использовать результаты предшествующих построений. Заметим, что частная дизъюнкция s-го разряда для данного множества входов уже получена; она вычисляет дизъюнкцию половины из входов рассматриваемой подгруппы. Заметим далее, что дизъюнкция половины из оставшихся входов как частная дизъюнкция $(s-1)$-го разряда группы входов с двумя фиксированными старшими разрядами 1 и 0, также уже вычислена (ведь схемы для частных дизъюнкций меньших порядков строились по тому же индуктивному принципу), и так далее.

Итак, шифратор с 2^{s+1} входами получается из двух шифраторов с 2^s входами, кроме того s элементов требуется дополнительно для вычисления частной дизъюнкции старшего разряда и по одному — для остальных частных дизъюнкций. Отсюда по индукции имеем

$$L(V_{2^{s+1}}) \le 2L(V_{2^s}) + 2s \le 2(2^{s+1} - 2s - 2) + 2s = 2^{s+2} - 2(s+1) - 2.$$

Также по индукции проверяется, что глубина построенной схемы равна s. Глубина частной дизъюнкции $(s+1)$-го разряда равна s по построению. Глубина выходов частных дизъюнкций других разрядов на 1 больше глубины шифратора с 2^s входами, откуда следует, что их глубина также s.

Рассмотрим теперь случай $2^{s+1} \ge r = 2^s + r'$, где $r' > 0$. Доказательство проведем индукцией по r.

Схема будет устроена так же, как и в частном случае. Множество входов разобьем на два: 2^s входов с нулевым старшим разрядом и r' — с единичным. Вычислим частные дизъюнкции на этих подмножествах (второе из них кодируется, вообще говоря, $\lceil \log_2 r' \rceil$ младшими разрядами исходного кода).

Далее, $\lceil \log_2 r' \rceil$ функциональных элементов необходимо, чтобы получить дизъюнкции младших разрядов. Еще столько же — для вычисления дизъюнкции $(s+1)$-го разряда. Это приводит к рекуррентному соотноше-

нию

$$L(V_r) \le L(V_{2^s}) + L(V_{r'}) + 2\lceil \log_2 r' \rceil \le$$
$$\le (2^{s+1} - 2s - 2) + (2r' - 2\lceil \log_2 r' \rceil - 2) + 2\lceil \log_2 r' \rceil = 2r - 2(s+1) - 2.$$

Глубина схемы равна $s = \lceil \log_2 r \rceil - 1$.

Доказательство леммы 4.4. Доказательство получается суммированием оценок из следствия 4.1 и леммы 4.6:

$$L(\Lambda_{k,r}) \le L(Q_{k,r}) + L(V_r) < \frac{kr}{\log_2 r} \cdot \frac{\log_2 r + 6}{\log_2 r - \log_2 \log_2 r} + 2r;$$

$$D(\Lambda_{k,r}) \le D(Q_{k,r}) + D(V_r) \le (\lceil \log_2 k \rceil + 2) + (\lceil \log_2 r \rceil - 1).$$

Из алгоритма DL и лемм 4.2–4.4 вытекает

Теорема 4.7 *Пусть $2^k - 1 = r_1 \cdot r_2 \cdot \ldots \cdot r_s$, где сомножители r_i попарно взаимно просты, $\rho = \max_i \log_2 r_i$. Тогда можно построить схему Λ_k дискретного логарифмирования в поле $GF(2^k)$, такую, что при $k \to \infty$ выполняются оценки:*

$$L(\Lambda_k) \le \sum_{i=1}^{s} (2 + (1 + o(1))k/\log r_i) r_i + O(skM(k)); \quad D(\Lambda_k) \le \rho + O(\log^2 k).$$

4.4.2 Выбор вспомогательного поля

В методе теоремы 4.7 эффективность логарифмирования зависит от *гладкости* разложения числа $2^k - 1$ в произведение взаимно простых сомножителей. (Разложение — *гладкое*, если все его сомножители малы. Также гладким называют число, допускающее гладкое разложение на множители (см., например, [18]).) Схема тем проще, чем меньше максимальный из сомножителей.

Для интерполяции требуется поле, содержащее не менее L элементов. Практически, среди нескольких полей со степенями $k \ge \lceil \log_2 L \rceil$ необходимо выбрать поле с возможно более гладким порядком мультипликативной группы.

Например, поле $GF(2^9)$ менее гладкое, чем $GF(2^{10})$, так как $2^9 - 1 = 7 \cdot 73$, а $2^{10} - 1 = 3 \cdot 11 \cdot 31$. Но еще более гладким является поле $GF(2^{12})$, поскольку $2^{12} - 1 = 5 \cdot 7 \cdot 9 \cdot 13$.

Для того, чтобы оценить эффективность операции логарифмирования, рассмотрим несколько способов выбора гладкого поля при произвольном значении L.

В первом из них выбирается наименьшее четное из подходящих значений k, $k = 2l \geq \lceil \log_2 p \rceil$, и используется разложение $2^{2l} - 1$ на всегда взаимно простые сомножители $2^l - 1$ и $2^l + 1$. Заметим, что одно из этих чисел делится на 3, откуда имеем

$$2^{2l} - 1 = \begin{cases} 3^d \cdot \frac{2^l - 1}{3^d} \cdot (2^l + 1), & l \text{ — четно;} \\ 3^d \cdot \frac{2^l + 1}{3^d} \cdot (2^l - 1), & l \text{ — нечетно,} \end{cases}$$

где 3^d — максимальная из степеней тройки, на которую делится $2^{2l} - 1$. Используя теорему 4.7, получаем следующие оценки для схемы вычисления дискретного логарифма в поле $GF(2^{2l})$: сложность по порядку $(8 + o(1))2^l \lesssim (16 + o(1))\sqrt{L}$, глубина — $l + o(l) \lesssim (1/2)\log_2 L$.

Более эффективная схема получается при выборе поля $GF(2^k)$, где $k = 6l \geq \lceil \log_2 L \rceil$ (выбирается наименьшее из возможных значений k). Взаимно простые сомножители $2^{3l} - 1$ и $2^{3l} + 1$ допускают дальнейшее разложение: $2^{3l} \pm 1 = (2^l \pm 1)(2^{2l} \mp 2^l + 1)$. Так как $2^{2l} \pm 2^l + 1 = 3 \bmod (2^l \mp 1)$, то 3 — это единственный общий делитель, который могут иметь множители в указанном разложении.

$$2^{6l} - 1 = \begin{cases} 3^{d+1} \cdot \frac{2^l - 1}{3^d} \cdot (2^l + 1) \cdot \frac{2^{2l} + 2^l + 1}{3} \cdot (2^{2l} - 2^l + 1), & l \text{ — четно;} \\ 3^{d+1} \cdot \frac{2^l + 1}{3^d} \cdot (2^l - 1) \cdot \frac{2^{2l} - 2^l + 1}{3} \cdot (2^{2l} + 2^l + 1), & l \text{ — нечетно.} \end{cases}$$

Сложность логарифмирования в поле $GF(2^{6l})$ оценивается как $(20/3 + o(1))2^{2l} \lesssim (80/3 + o(1))\sqrt[3]{L}$, глубина — $2l + o(l) \lesssim (1/3)\log_2 L$. Для $L > 2^8$ (что отражает пример поля $GF(2^{12})$) оценки, полученные вторым способом, лучше, чем в первом случае.

Дальнейшее развитие идеи использования разложения многочленов вида $x^k - 1$ на неприводимые многочлены над $\mathbb{Z}$ приводит к построению схемы логарифмирования сложности $C(\epsilon)O(L^\epsilon)$ и глубины $(\epsilon + o(1))\log_2 L$, где $\epsilon > 0$.

Рассмотрим поле $GF(2^{k_v l})$, где $k_v = p_1 \cdot p_2 \cdot \ldots \cdot p_v$, $\{p_i\}$ — возрастающая последовательность простых натуральных чисел.

Теорема 4.8 *Пусть $l \in \mathbb{N}$. Тогда число $2^{k_v l} - 1$ представимо в виде произведения попарно взаимно простых сомножителей $r_1, r_2, \ldots, r_s$, при этом*

$$\max_i r_i \leq 2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l},$$

где $\varphi(k)$ — функция Эйлера.

Доказательство теоремы предварим несколькими вспомогательными утверждениями. Сначала обратимся к теории круговых многочленов (подробное ее изложение см. в [19]).

Пусть $d \in \mathbb{N}$. Многочлен $F_d \in \mathbb{C}[x]$ минимально возможной степени со старшим коэффициентом 1 такой, что его корнями являются все первообразные корни степени d из единицы, называется *d-м круговым многочленом.*

Для нас существенны следующие свойства круговых многочленов:
(1) $F_d \in \mathbb{Z}[x]$;
(2) $\deg F_d = \varphi(d)$;
(3) многочлены $\{F_d\}$ попарно взаимно просты;
(4) $x^h - 1 = \prod_{d|h} F_d(x)$.

Лемма 4.7 *Пусть $x \geq 1$, тогда $F_d(x) \leq x^{\varphi(d)} e^{\varphi(d)/x}$.*

Доказательство. Корни многочлена $F_d(x)$ по модулю равны 1, обозначим их через ξ_i, тогда

$$F_d(x) = \prod_{i=1}^{\varphi(d)} (x - \xi_i) < \prod_{i=1}^{\varphi(d)} (x + |\xi_i|) =$$

$$= (x+1)^{\varphi(d)} = x^{\varphi(d)} \left(1 + \frac{1}{x}\right)^{\varphi(d)} \leq x^{\varphi(d)} e^{\varphi(d)/x}.$$

Заметим, что если $x \to \infty$, то $F_d(x) = O(x^{\varphi(d)})$. Нам понадобится еще одна лемма о делимости чисел (она приведена в [27, задача 12]).

Лемма 4.8 *Пусть p — простое число, $a \in \mathbb{Z}$, тогда*

$$\text{НОД}\left(a-1, \frac{a^p-1}{a-1}\right) = \text{НОД}\left((a-1)^2, \frac{a^p-1}{a-1}\right) = \text{НОД}(a-1, p).$$

Доказательство. Дважды разделим многочлен $x^{p-1} + \ldots + 1 = \frac{x^p-1}{x-1}$ на $x-1$ с остатком:

$$x^{p-1} + x^{p-2} + \ldots + 1 =$$
$$= (x-1)^2 \left(x^{p-3} + 3x^{p-4} + \ldots + \frac{(p-1)(p-2)}{2}\right) + (x-1)\frac{p(p-1)}{2} + p.$$

Подставим a вместо x. Последующая проверка отношений делимости не представляет труда.

Доказательство теоремы 4.8. Многочлен $x^{k_v}-1$ раскладывается в произведение круговых многочленов (свойство (4))

$$x^{k_v}-1=\prod_{d|k_v}F_d(x).$$

Количество сомножителей в этом произведении равно 2^v — оно соответствует количеству делителей числа k_v.

Если вместо переменной x подставить число 2^l, то получится разложение

$$2^{k_v l}-1=\prod_{d|k_v}F_d(2^l) \qquad (*)$$

числа $2^{k_v l}-1$ на множители, не превосходящие $2^{\varphi(k_v)l}e^{\varphi(k_v)/2^l}$, что следует из леммы 4.7, так как $\varphi(k_v)$ — максимальная из степеней многочленов F_d. Исследуем далее возможность преобразования данного разложения в произведение *взаимно простых* сомножителей, не превосходящих $2^{\varphi(k_v)l}e^{\varphi(k_v)/2^l}$.

Рассмотрим следующие разложения многочлена $x^{k_v}-1$ в произведение двух сомножителей (для краткости введем обозначение $y_i=x^{k_v/p_i}$):

$$x^{k_v}-1=y_i^{p_i}-1=(y_i-1)(y_i^{p_i-1}+\ldots+1). \qquad (i)$$

Из свойства (4) круговых многочленов следует, что

$$y_i-1=\prod_{d|k_v,\,p_i\nmid d}F_d(x), \qquad y_i^{p_i-1}+\ldots+1=\prod_{d|k_v,\,p_i|d}F_d(x),$$

поэтому любой многочлен $F_d(x)$, $d\mid k_v$, целиком делит какой-либо из двух сомножителей в правой части каждой из формул (i).

Покажем, что общими делителями значений двух круговых многочленов (при подстановке 2^l) могут быть только простые числа p_i, $i=2,\ldots,v$.

Рассмотрим произвольную пару $F_{d_1}(2^l)$ и $F_{d_2}(2^l)$, где $d_1,d_2\mid k_v$, пусть при этом $d_1<d_2$. Тогда обязательно найдется такое число p_i, что $p_i\mid d_2$ и $p_i\nmid d_1$. Рассмотрим разложение (i). Многочлен $F_{d_1}(x)$ делит первый из сомножителей, а $F_{d_2}(x)$ — второй. Из леммы 4.8 следует, что

$$\text{НОД}(\,(y_i-1)\,|_{x=2^l},\ (y_i^{p_i-1}+\ldots+1)\,|_{x=2^l}\,)\in\{1,p_i\}.$$

Следовательно,

$$\text{НОД}(F_{d_1}(2^l),\,F_{d_2}(2^l))\in\{1,p_i\}.$$

Выделив в отдельные сомножители $p_i^{c_i}$, $i=2,\ldots,v$, где c_i — кратность p_i в произведении, получим разложение на взаимно простые множители. Осталось показать, что вновь образованные множители $p_i^{c_i}$ удовлетворяют оценке $2^{\varphi(k_v)l}e^{\varphi(k_v)/2^l}$.

Если p_i делит только один из сомножителей исходного разложения $(*)$, то доказывать нечего. Рассмотрим случай, когда p_i делит два сомножителя разложения $(*)$ $F_{d_1}(2^l)$ и $F_{d_2}(2^l)$, $d_1 \neq d_2$. Из леммы 4.8 следует, что в любом разложении (j), где $j \neq i$, многочлены $F_{d_1}(x)$ и $F_{d_2}(x)$ делят один и тот же сомножитель в правой части (иначе p_i не может быть общим делителем). Покажем, что в разложении (i) они делят различные сомножители.

Заметим, что d, произвольный делитель k_v, однозначно определяется, если про любое число p_s, $s = 1, \ldots, v$, известно, делит оно d или нет. Поэтому и многочлен $F_d(x)$ единственным образом определяется по своей принадлежности к одному из сомножителей в каждом разложении (s).

Из этого замечания вытекает, что если бы в разложении (i) (как и во всех остальных) многочлены $F_{d_1}(x)$ и $F_{d_2}(x)$ делили один и тот же сомножитель, то они бы совпадали, что противоречило бы условию $d_1 \neq d_2$. Следовательно, в разложении (i) они делят различные сомножители.

Из приведенного рассуждения также следует, что p_i не может делить более двух сомножителей из набора $F_d(2^l)$, $d \mid k_v$. При этом, как следует из леммы 4.8, если p_i делит ровно два сомножителя в разложении $(*)$, то один из сомножителей (а именно тот, который соответствует многочлену, делящему $y_i - 1$ в разложении (i)) делится на $p_i^{c_i - 1}$. Следовательно, выделенный множитель $p_i^{c_i}$ может, самое большее, в p_i раз превосходить любой из сомножителей $F_d(2^l)$ исходного разложения, для которого выполнено: $d \mid k_v$ и $p_i \nmid d$. При этом $\varphi(d) \leq \varphi(k_v)/(p_i - 1)$. Используя лемму 4.7, имеем

$$p_i^{c_i} \leq p_i \cdot \max_{d \mid k_v,\, p_i \nmid d} F_d(2^l) \leq p_i 2^{\varphi(k_v) l/(p_i-1)} e^{\varphi(k_v)/(2^l(p_i-1))}.$$

Покажем, что

$$p_i 2^{\varphi(k_v) l/(p_i-1)} e^{\varphi(k_v)/(2^l(p_i-1))} < 2^{\varphi(k_v) l} e^{\varphi(k_v)/2^l},$$

где $1 < i \leq v$, $l \geq 1$. Отдельно рассмотрим случай $i = v = 2$, $l = 1$ (т. е. $p_i = 3$, $k_v = 6$). После подстановки параметров в неравенство, оно принимает вид $6\sqrt{e} < 4e$, что верно. Неравенство тем более остается верным при увеличении параметров v и (или) l. Если $i > 2$, тогда выполнено $p_i < 2^{\varphi(k_i)/2}$, что проверяется, например, следующим образом по индукции. При $i = 3$ неравенство выполнено в силу $5 < 2^4$. Если верно, что $p_{i-1} < 2^{\varphi(k_{i-1})/2}$, то

$$p_i < 2p_{i-1} < p_{i-1}^{p_i - 1} < 2^{\varphi(k_{i-1})(p_i-1)/2} = 2^{\varphi(k_i)/2}.$$

Мы воспользовались известным неравенством $p_i < 2p_{i-1}$ (постулат Бертра-

на). Из доказанного промежуточного неравенства выводим

$$p_i 2^{\varphi(k_v)l/(p_i-1)} e^{\varphi(k_v)/(2^l(p_i-1))} < 2^{\varphi(k_i)/2+\varphi(k_v)l/(p_i-1)} e^{\varphi(k_v)/2^l} < \\ < 2^{\varphi(k_v)l(1/2+1/(p_i-1))} e^{\varphi(k_v)/2^l} < 2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l}.$$

Тем самым теорема полностью доказана.

Заметим, что количество сомножителей в построенном разложении не превосходит $2^v + v - 1$, где 2^v — количество сомножителей в исходном разложении $(*)$, плюс дополнительно выделяется не более $v-1$ сомножителей.

Для иллюстрации рассмотрим пример $v = 3$, $k_3 = 30$.

$$x^{30} - 1 = F_1 F_2 F_3 F_5 F_6 F_{10} F_{15} F_{30};$$

$$F_1(x) = x - 1, \quad F_2(x) = x + 1, \quad F_3(x) = x^2 + x + 1,$$
$$F_5(x) = x^4 + x^3 + x^2 + x + 1, \quad F_6(x) = x^2 - x + 1,$$
$$F_{10}(x) = x^4 - x^3 + x^2 - x + 1, \quad F_{15}(x) = x^8 - x^7 + x^5 - x^4 + x^3 - x + 1,$$
$$F_{30}(x) = x^8 + x^7 - x^5 - x^4 - x^3 + x + 1.$$

Пусть $l = 1$. При подстановке $x = 2$ получается разложение на множители для числа $2^{30} - 1$ (порядок сомножителей сохранен)

$$2^{30} - 1 = 1 \cdot 3 \cdot 7 \cdot 31 \cdot 3 \cdot 11 \cdot 151 \cdot 331.$$

Выделим в отдельные множители вхождения в произведение двух первых нечетных простых чисел: 3 (встречается два раза, $F_2(2) = F_6(2) = 3$) и 5 (нет). Окончательно, что гарантируется теоремой, имеем удовлетворяющее всем требованиям разложение

$$2^{30} - 1 = 7 \cdot 9 \cdot 11 \cdot 31 \cdot 151 \cdot 331,$$

которое в данном случае совпадает с каноническим разложением $2^{30} - 1$ на простые множители.

Используя результат теоремы 4.8, можно показать, что в поле $GF(2^{30l})$ логарифмирование выполняется со сложностью $O(2^{8l})$, что при $2^{30l} > L \geq 2^{30(l-1)}$ соответствует оценке $O(L^{4/15})$. Однако мультипликативная константа в этом случае слишком велика.

Далее мы будем пользоваться следующим фактом:

$$\frac{c_1}{\log(v+1)} < \frac{\varphi(k_v)}{k_v} = \prod_{i=1}^{v} \frac{p_i - 1}{p_i} < \frac{c_2}{\log v} \to 0, \quad \text{при } v \to \infty.$$

Легко показать, что $c_1 > e^{-5/2}$, а $c_2 < \ln 2 = 0.693\ldots$ Более точные оценки приведены в работе [80][9].

Теорема 4.9 *Пусть $v \in \mathbb{N}$, $L \geq 2^{k_v}$. Тогда существует поле характеристики 2, содержащее не менее L элементов, в котором сложность логарифмирования (при $L \to \infty$) не превосходит*

$$\log_2(v+1)2^{\varphi(k_v)+v+4}e^{\varphi(k_v)L^{-1/k_v}}L^{\varphi(k_v)/k_v} + O(2^v \log^3 L),$$

а глубина не превосходит

$$(\varphi(k_v)/k_v)\log_2 L + 2\varphi(k_v) + O(\log^2 \log L).$$

Доказательство. Для заданного L рассмотрим поле $GF(2^{k_v l})$, где l удовлетворяет соотношению $2^{k_v(l-1)} < L \leq 2^{k_v l}$. Согласно теореме 4.8, число $2^{k_v l} - 1$ раскладывается в произведение взаимно простых сомножителей, не превосходящих $2^{\varphi(k_v)l}e^{\varphi(k_v)/2^l}$.

С помощью теоремы 4.7 оценим глубину схемы логарифмирования в поле $GF(2^{k_v l})$ как

$$\begin{aligned}&\log_2(2^l e^{1/2^l})^{\varphi(k_v)} + O(\log^2(k_v l)) < \\ &< \varphi(k_v)(l+1) + O(\log^2 \log L) < \varphi(k_v)(2 + (\log_2 L)/k_v) + O(\log^2 \log L) < \\ &< (\varphi(k_v)/k_v)\log_2 L + 2\varphi(k_v) + O(\log^2 \log L).\end{aligned}$$

Та же теорема 4.7 позволяет оценить порядок сложности схемы как

$$\begin{aligned}&(2^v + v)(2 + k_v/\log_2 r_{max})r_{max} + O(2^v k_v l M(k_v l)) \leq \\ &\leq (2^v + v)(2 + k_v/\varphi(k_v))2^{\varphi(k_v)l}e^{\varphi(k_v)/2^l} + O(2^v \log^3 L),\end{aligned}$$

где r_{max} — максимальный из сомножителей в используемом разложении числа $2^{k_v l} - 1$, который оценивается при помощи теоремы 4.8.

Справедливо неравенство:

$$(2^v + v)(2 + k_v/\varphi(k_v)) < 2^{v+4}\log_2(v+1),$$

которое проверяется при $v = 1, \ldots, 4$ непосредственной подстановкой, а при $v \geq 5$ можно воспользоваться оценкой $k_v/\varphi(k_v) < e^{5/2}\log_2(v+1)$, тогда

$$(2^v + v)(2 + k_v/\varphi(k_v)) < \log_2(v+1)2^v\left[\left(1 + \frac{v}{2^v}\right)\left(e^{5/2} + \frac{2}{\log_2(v+1)}\right)\right].$$

[9] Согласно теореме Мертенса это выражение имеет асимптотику $\frac{e^\gamma}{\ln v}$, где γ — постоянная Эйлера (см. также [80]).

Выражение в квадратных скобках является монотонно убывающей функцией от v, и из того, что при $v = 5$ его значение меньше 16, следует заявленное неравенство.

Далее,

$$2^{\varphi(k_v)l} = 2^{\varphi(k_v)}\big(2^{k_v(l-1)}\big)^{\varphi(k_v)/k_v} < 2^{\varphi(k_v)} L^{\varphi(k_v)/k_v}.$$

Наконец, так как $2^l \geq L^{-1/k_v}$, имеем

$$e^{\varphi(k_v)/2^l} \leq e^{\varphi(k_v)L^{-1/k_v}}.$$

Комбинируя все указанные неравенства, получаем требуемый результат.

Теорема 4.10 *Пусть $\epsilon > 0$. Тогда существует поле характеристики 2, содержащее не менее L элементов, в котором дискретное логарифмирование выполняется со сложностью (при $L \to \infty$), не превосходящей $C_1(\epsilon)L^{\epsilon}$, и глубиной $\epsilon \log_2 L + C_3(\epsilon) + O(\log\log L)$, где C_1, C_3 — некоторые ограниченные на любом отрезке интервала $(0, 1]$ функции.*

Доказательство. Выберем v, такое, что $\frac{\varphi(k_v)}{k_v} \leq \epsilon$ (например, подойдет $v = \lceil 2^{0.7/\epsilon} \rceil$) и применим теорему 4.9.

5 Переход между нормальными и стандартными базисами

Основной мотивировкой перехода от нормального к полиномиальному базису является ускорение операции умножения, а обратно — операции возведения в степень (см. [5]). В общем случае такой переход может быть выполнен методом О. Б. Лупанова для линейных операторов со сложностью $O(n^2/\log_q n)$ [23]. В работе [4] показано, что для оптимальных нормальных базисов справедлива оценка $O(n \log n)$. Оценка сложности перехода фактически является оценкой сложности умножения в произвольном нормальном базисе.

Идея об использовании перехода между двумя типами базисов для ускорения некоторых операций, по-видимому, впервые была явно высказана в работах [70, 4].

В работе [70] показано, что переход от нормального базиса к стандартному можно выполнить со сложностью $O(n^{1,815})$. Там же построен вероятностный алгоритм обратного перехода с такой же оценкой сложности.

В §5.1 будет описан базовый метод Брента—Кунга [44]. С его помощью в §5.2 и§5.3 будут построены схемы перехода соответственно к нормальному и к стандартному базису сложности $O(n^{1,834})$ и глубины $O(\log n)$. В §5.4 будет показано, что оценку сложности этих схем можно уточнить до $O(n^{1,806})$. Для схемы перехода к стандартному базису эта оценка фактически установлена в [70] (см. п. 5.5.1). В §5.5 будут рассмотрены некоторые приложения.

5.1 Метод Брента—Кунга

Рассмотрим задачу вычисления x^{q^k}, где $x \in GF(q^n)$, в стандартном базисе поля. Оценка сложности $O(n^2/\log_q n)$ может быть получена методом Лупанова [23] (т.к. операция Фробениуса является линейным преобразованием). В 1978 г. в работе [44] был предложен метод, имеющий меньший порядок сложности (см. также [5]). Опишем его применительно к данной задаче. Пусть $x = f(t)$ в стандартном представлении, $\deg f < n$,

$$f^{q^k}(t) \bmod m_n(t) = f(t^{q^k}) \bmod m_n(t) = \\ = f(t^{q^k} \bmod m_n(t)) \bmod m_n(t) = f(\xi(t)) \bmod m_n(t),$$

где $\xi(t) = t^{q^k} \bmod m_n(t)$. Пусть $n \le rp$. Запишем,

$$f(t) = f_0(t) + f_1(t)t^r + \ldots + f_{p-1}(t)t^{(p-1)r}, \qquad (5.1)$$

где $\deg f_i < r$. Пусть $f_i(t) = f_{i,0} + f_{i,1}t + \ldots + f_{i,r-1}t^{r-1}$.

АЛГОРИТМ 1.

0. Считаем, что многочлены

$$\xi(t),\ \xi^2(t),\ \ldots,\ \xi^{r-1}(t);\ \ \xi^r(t),\ \xi^{2r}(t),\ \ldots,\ \xi^{(p-1)r}(t)$$

(все — по модулю $m_n(t)$) вычислены заранее.

1. Вычисляются многочлены

$$f_i(\xi(t)) = f_{i,0} + f_{i,1}\xi(t) + \ldots + f_{i,r-1}\xi^{r-1}(t).$$

Заметим, что выполнено

$$\begin{bmatrix} f_0(\xi(t)) \\ f_1(\xi(t)) \\ \cdots \\ f_{p-1}(\xi(t)) \end{bmatrix}_{p\times n} = \begin{bmatrix} f_0(t) \\ f_1(t) \\ \cdots \\ f_{p-1}(t) \end{bmatrix}_{p\times r} \cdot \begin{bmatrix} 1 \\ \xi(t) \\ \cdots \\ \xi^{r-1}(t) \end{bmatrix}_{r\times n}, \qquad (5.2)$$

где в строчках матриц выписаны коэффициенты соответствующих многочленов, а снизу подписаны размерности. Сложность шага оценивается как сложность умножения $p \times r$-матрицы на $r \times n$-матрицу.

2. Вычисляются произведения $f_i(\xi(t)) \cdot \xi^{ir}(t)$. Сложность шага оценивается как $O(pM(n))$.

3. Результаты предыдущего шага (многочлены степени $2n-2$) складываются покомпонентно со сложностью не более чем $2np$.

4. Окончательно, результат приводится по модулю $m_n(t)$ со сложностью $O(M(n))$ (см. п. 2.3).

ОЦЕНКА СЛОЖНОСТИ.

Выберем $r, p \sim \sqrt{n}$. Сложность шагов **2**–**4** оценивается как $O(pM(n)) = O(n^{1,5}\log n \log\log n)$. Сложность шага **1** есть $O(n^w)$, где $w < 1,667$ — экспонента матричного умножения $T_{\sqrt{n},\sqrt{n},n}$ (т.е. умножения матрицы размера $\sqrt{n} \times \sqrt{n}$ на матрицу размера $\sqrt{n} \times n$). Таким образом, алгоритм может быть реализован схемой сложности $O(n^{1,667})$ и глубины $O(\log n)$.

5.2 Переход к нормальному базису

Рассмотрим задачу преобразования координат элемента $x \in GF(q^n)$ из стандартного базиса $A = \{1, t, t^2, \ldots, t^{n-1}\}$ в нормальный базис $B = \{\alpha_0 = t, \alpha_1 = t^q, \alpha_2 = t^{q^2}, \ldots, \alpha_{n-1} = t^{q^{n-1}}\}$. Пусть $n \leq ms$.

АЛГОРИТМ 2.

0. Считаем, что известны частичные базисные разложения (относительно первых m элементов базиса B) для $1, t, t^2, \ldots, t^{n-1}$, т.е.

$$t^i = t_{i,0}\alpha_0 + t_{i,1}\alpha_1 + \ldots + t_{i,m-1}\alpha_{m-1} + \beta_i,$$

где $\beta_i \in < \alpha_m, \ldots, \alpha_{n-1} >$.

1. При помощи алгоритма 1 вычисляются

$$x^{q^m},\ x^{q^{2m}},\ \ldots,\ x^{q^{m(s-1)}}$$

в стандартном базисе со сложностью $O(sn^w + s\sqrt{n}M(n))$.

2. Вычисляются частичные разложения для $x^{q^{mi}}$ подстановкой известных разложений для $1, t, t^2, \ldots, t^{n-1}$. Справедливо

$$\begin{bmatrix} x \\ x^{q^m} \\ \ldots \\ x^{q^{m(s-1)}} \end{bmatrix}^B_{s\times m} = \begin{bmatrix} x \\ x^{q^m} \\ \ldots \\ x^{q^{m(s-1)}} \end{bmatrix}^A_{s\times n} \cdot \begin{bmatrix} 1 \\ t \\ \ldots \\ t^{n-1} \end{bmatrix}^B_{n\times m},$$

где в строчках матриц выписаны коэффициенты в разложениях по базисам, указанным в верхних индексах матриц (для нормального базиса B имеется в виду частичное разложение).

Частичное разложение $x^{q^{mi}}$ определяет m координат из полного разложения $x = x_0\alpha_0 + \ldots + x_{n-1}\alpha_{n-1}$ в базисе B, а именно

$$x^{q^{mi}} = x_{n-mi}\alpha_0 + x_{n-mi+1}\alpha_1 + \ldots + x_{n-m(i-1)-1}\alpha_{m-1} + \ldots$$

Здесь для $j \notin [0, n-1]$ полагаем $x_j \equiv x_{j \bmod n}$. Т.к. все $x^{q^{mi}}$ вычислены, то получено полное разложение x в нормальном базисе. Сложность шага оценивается как сложность умножения $s \times n$-матрицы на $n \times m$-матрицу.

ОЦЕНКА СЛОЖНОСТИ.

Сложность шага **2** оценим как сложность s умножений матрицы размера $s \times m$ на $m \times m$-матрицу. Известно [50], что матричное умножение $T_{m^h,m,m}$, где $h \leq 0,294$, выполняется со сложностью $O(m^{2+\epsilon})$ при любом $\epsilon > 0$. Положим $s \sim n^{1-w/2}, m \sim n^{w/2}$. При $w = 1,667$ выполнено условие $s = o(m^{0,294})$, поэтому сложность обоих шагов и алгоритма в целом оценивается как $O(n^{1+w/2}) \sim O(n^{1,834})$. В отношении глубины справедливо то же замечание, что и для алгоритма 1.

5.3 Переход к стандартному базису

Рассмотрим задачу перехода от нормального представления элемента поля $x = x_0\alpha_0 + \ldots + x_{n-1}\alpha_{n-1}$ к стандартному. Пусть $n \leq ms$. Можно записать,

$$x = \gamma_0 + \gamma_1^{q^m} + \ldots + \gamma_{s-1}^{q^{m(s-1)}},$$

где $\gamma_i = x_{mi}\alpha_0 + \ldots + x_{mi+m-1}\alpha_{m-1}$. Если $j \geq n$, полагаем $x_j \equiv 0$.

АЛГОРИТМ 3.

0. Считаем, что для $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}$ известны представления в стандартном базисе.

1. По заданным представлениям для первых m элементов нормального базиса вычисляются стандартные представления для γ_i,

$$\begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \cdots \\ \gamma_{s-1} \end{bmatrix}^A_{s\times n} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \cdots \\ \gamma_{s-1} \end{bmatrix}^B_{s\times m} \cdot \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \cdots \\ \alpha_{m-1} \end{bmatrix}^A_{m\times n}$$

в прежних обозначениях. Сложность шага оценивается как сложность умножения $s \times m$-матрицы на $m \times n$-матрицу.

2. Вычисляются $\gamma_i^{q^{mi}}$ алгоритмом 1 со сложностью $O(sn^w + s\sqrt{n}M(n))$ и складываются покомпонентно со сложностью, не превосходящей ns.

Оценивание производится в точности как в алгоритме 2. Таким образом, показано, что переход между базисами может выполняться схемой над $GF(q)$ со сложностью $O(n^{1,834})$ и глубиной $O(\log n)$.

5.4 Уточнение оценки сложности

В этом параграфе будет рассмотрен модифицированный способ реализации шага **1** в алгоритме 2 и шага **2** в алгоритме 3, который позволит для обоих алгоритмов получить оценку сложности $O(n^{1,806})$. Для этого нам потребуется распространить метод Брента—Кунга из §5.1 на решение следующей задачи.

Пусть заданы элементы $x_1, x_2, \ldots, x_l \in GF(q^n)$ своими представлениями в стандартном базисе поля и натуральные параметры s и m. Требуется вычислить все $x_i^{q^{jm}}$, $j = 1, \ldots, s$, также в стандартном базисе. Иначе говоря, рассматривается одновременная реализация s последовательных степеней операции Фробениуса $x \to x^{q^m}$ на l элементах поля.

Через w_τ будем далее обозначать экспоненту матричного умножения T_{n,n,n^τ}.

Лемма 5.1 *Пусть элементы $x_1, \ldots, x_l \in GF(q^n)$ заданы в стандартном базисе, где $l \sim n^\lambda$, $\lambda \in [0,1]$. Пусть также $\delta > 0$ и $s \sim n^\sigma$, $\sigma \in [0,1]$. Обозначим $\tau = 2(1+\sigma)/(1+\lambda)$. Тогда вычисление всех $x_i^{q^{jm}}$, $i = 1, \ldots, l$, $j = 1, \ldots, s$, может быть реализовано схемой над $GF(q)$ сложности*

$$O\left(n^{(1+\lambda)w_\tau/2} + n^{\sigma+(1+\lambda)/2} M(n)\right)$$

и глубины $O(\log n)$.

Доказательство. Обозначим $\xi_j(t) = t^{q^{jm}} \bmod m_n(t)$. Пусть элементу x_i соответствует многочлен $g_i(t)$. Представим $g_i(t)$ в виде (5.1) (см. §5.1). Матричное тождество (5.2) (см. алгоритм 1), соответствующее вычислению $x_i^{q^{jm}}$, т.е. композиции многочленов $g_i(\xi_j)$, запишем сокращенно $X_{i,j} = Y_i \cdot C_j$. Тогда выполнено

$$\begin{bmatrix} X_{1,1} & \cdots & X_{1,s} \\ \cdots & \cdots & \cdots \\ X_{l,1} & \cdots & X_{l,s} \end{bmatrix}_{lp \times sn} = \begin{bmatrix} Y_1 \\ \cdots \\ Y_l \end{bmatrix}_{lp \times r} \cdot \begin{bmatrix} C_1 & \cdots & C_s \end{bmatrix}_{r \times sn},$$

где $n \le rp$.

Все $x_i^{q^{jm}}$ вычисляются при помощи ls экземпляров алгоритма 1, у которых шаг **1** выполняется совместно — посредством умножения $lp \times r$-матрицы на $r \times sn$-матрицу, как записано выше. Для шагов **2**–**4**, считая что они выполняются независимо, имеем оценку сложности $O(lspM(n))$.

Если $l \le n$, то параметры p и r можно выбрать таким образом, что $lp \sim r \sim \sqrt{ln}$. Пусть $l \sim n^\lambda$ и $s \sim n^\sigma$, где $\lambda, \sigma \in [0,1]$. Тогда $p \sim n^{(1-\lambda)/2}$,

$r \sim n^{(1+\lambda)/2}$, и сложность совмещенного шага **1** оценивается как сложность умножения матриц размера $n^{(1+\lambda)/2} \times n^{(1+\lambda)/2}$ и $n^{(1+\lambda)/2} \times n^{1+\sigma}$. Обозначив $\tau = 2(1+\sigma)/(1+\lambda)$, получаем для этого шага оценку сложности $O(n^{(1+\lambda)w_\tau/2})$.

Заметим, что второй член в оценке сложности алгоритма леммы 5.1 с точностью до логарифмического множителя имеет порядок $n^{1+\sigma+(1+\lambda)/2} = n^{(1+\lambda)(1+\tau)/2} \leq n^{(1+\lambda)w_\tau/2}$ в силу очевидного неравенства $w_\tau \geq \tau + 1$. Таким образом, сложность алгоритма определяется сложностью матричного умножения на первом шаге.

В работе [67] показано, что w_τ не превосходит минимума по $\theta \in [0,1]$, $h \in \mathbb{N}$ величины

$$\frac{(2+\tau)\log(h+2)+\theta\log\theta+(1+\tau)(1-\theta)\log((1+\tau)(1-\theta))+(1+\tau\theta)\log(1+\tau\theta)-(2+\tau)\log(2+\tau)}{(1-\theta)\log h}. \qquad (5.3)$$

Следующая лемма мотивируется тем, что оценка сложности умножения матриц размера $\sqrt{n} \times \sqrt{n}$ и $\sqrt{n} \times n^{1+\sigma}$, вытекающая из (5.3), хуже, чем аналогичная оценка для умножения матриц размера $n^{(1+\sigma)/2} \times n^{(1+\sigma)/2}$ и $n^{(1+\sigma)/2} \times n$.

Лемма 5.2 *Пусть элемент $x \in GF(q^n)$ задан в стандартном базисе и $\delta > 0$. Далее, пусть $s \sim n^\sigma$, где $\sigma \in [0,1]$. Обозначим $\tau = 2/(1+\sigma)$. Тогда вычисление всех $x^{q^{jm}}$, $j = 0, \ldots, s-1$, может быть выполнено схемой над $GF(q)$ сложности $O(n^{\delta+w_\tau/\tau})$ и глубины $O(\delta^{-1}\log n)$.*

Идею доказательства леммы поясним на простом примере. Пусть $s = r^2$. Тогда вычисление x, $x^{q^m}, \ldots, x^{q^{(r^2-1)m}}$ можно выполнить в два этапа. Сначала вычисляется набор x, $x^{q^{rm}}, x^{q^{2rm}} \ldots, x^{q^{(r-1)rm}}$, а на втором шаге к нему применяются операции Фробениуса со степенями $q^m, q^{2m}, \ldots, q^{(r-1)m}$.

Доказательство. Пусть $k \in \mathbb{N}$ и $s \leq s_1 \cdot \ldots \cdot s_k$, где все $s_i \sim \sqrt[k]{s}$. Обозначим $u_i = s_1 \cdot \ldots \cdot s_i$, $m_i = m s_{i+1} \cdot \ldots \cdot s_k$, полагаем $m_k = m$. Рассмотрим следующий порядок вычислений.

На первом шаге вычисляются все $x^{q^{jm_1}}$, $j = 0, \ldots, s_1 - 1$.

Пусть $2 \leq i \leq k$. После $(i-1)$-го шага вычислен набор $x^{q^{jm_{i-1}}}$, $j = 0, \ldots, u_{i-1} - 1$. На i-м шаге вычисляется результат применения операций Фробениуса со степенями q^{jm_i}, $j = 1, \ldots, s_i - 1$, к этому набору. Получаем все $x^{q^{jm_i}}$, $j = 0, \ldots, u_i - 1$.

В частности, после k-го шага известны все $x^{q^{jm}}$, $j = 0, \ldots, s-1$.

Пусть каждый шаг выполняется алгоритмом леммы 5.1. Как следует из этой леммы, сложность i-го шага определяется умножением $n^{(k+(i-1)\sigma)/2k} \times n^{(k+(i-1)\sigma)/2k}$-матрицы на $n^{(k+(i-1)\sigma)/2k} \times n^{1+\sigma/k}$-матрицу.

Сложность этой операции можно оценить как сложность $n^{\frac{(k-i+1)\sigma}{k(1+\sigma)}+\frac{\sigma}{k}}$ умножений матриц с набором линейных размеров $n^{(k+(i-1)\sigma)/2k}$, $n^{(k+(i-1)\sigma)/2k}$ и $n^{\tau(k+(i-1)\sigma)/2k}$ (это следует из леммы 2.4). Для экспоненты матричного умножения имеем оценку

$$w_\tau\left(\frac{1}{2}+\frac{(i-1)\sigma}{2k}\right)+\frac{(k-i+1)\sigma}{k(1+\sigma)}+\frac{\sigma}{k}=$$
$$=\frac{w_\tau}{2}+\frac{\sigma}{2k}\Big((i-1)w_\tau+(k-i+1)\tau\Big)+\frac{\sigma}{k}=$$
$$=\frac{w_\tau}{\tau}-\frac{\sigma}{2k}(k-i+1)(w_\tau-\tau)+\frac{\sigma}{k}\le\frac{w_\tau}{\tau}+\frac{\sigma}{2k}-\frac{(k-i)\sigma}{2k},$$

где последний переход следует из неравенства $w_\tau \ge \tau+1$.

Суммарная сложность вычислений по всем шагам оценивается из формулы для суммы геометрической прогрессии со знаменателем $n^{-\sigma/2k}$ как

$$O\left(n^{w_\tau/\tau+O(1/k)}/(1-n^{-\sigma/2k})\right).$$

Окончательно, утверждение леммы следует из выбора $k \sim c/\delta$, где c — подходящая константа.

Лемма 5.3 *Пусть элементы $x_0,\dots,x_{s-1}\in GF(q^n)$ заданы в стандартном базисе, $\delta>0$ и $s\sim n^\sigma$, где $\sigma\in[0,1]$. Обозначим $\tau=2/(1+\sigma)$. Тогда сумма $\sum x_j^{q^{jm}}$ может быть вычислена схемой над $GF(q)$ сложности $O(n^{\delta+w_\tau/\tau})$ и глубины $O(\delta^{-1}\log n)$.*

Доказательство предварим рассмотрением примера, в котором $s=r^2$, а сумма

$$x_0+x_1^{q^m}+\ldots+x_{r^2-1}^{q^{\left(r^2-1\right)m}}$$

вычисляется в два этапа. Сначала операции Фробениуса $x\to x^{q^{im}}$, $i=0,\dots,r-1$ выполняются на соответствующих наборах $\{x_j \mid j=i \bmod r\}$, и вычисляются суммы

$$x_{1,j}=x_{jr}+x_{jr+1}^{q^m}+\ldots+x_{jr+r-1}^{q^{(r-1)m}},\qquad j=0,\dots,r-1.$$

Окончательно, искомая сумма получается как

$$x_{1,0}+x_{1,1}^{q^{rm}}+\ldots+x_{1,r-1}^{q^{(r-1)rm}}.$$

Доказательство. Выберем $k \in \mathbb{N}$ и $s_i \sim \sqrt[k]{s}$ так, что $s \le s_1 \cdot \ldots \cdot s_k$. Обозначим $m_i = ms_1 \cdot \ldots \cdot s_i$ и $n_i = s_{i+1} \cdot \ldots \cdot s_k$, где $n_k = 1$ и $m_0 = m$. Положим $x_{0,j} = x_j$, если $j < s$, и $x_{0,j} = 0$ иначе. Запишем,

$$\sum_{j=0}^{s-1} x_j^{q^{jm}} = \sum_{j=0}^{n_0-1} x_{0,j}^{q^{jm}} = \sum_{j=0}^{n_1-1} x_{1,j}^{q^{jm_1}} = \ldots = \sum_{j=0}^{n_i-1} x_{i,j}^{q^{jm_i}} = \ldots = \sum_{j=0}^{n_{k-1}-1} x_{k-1,j}^{q^{jm_{k-1}}},$$

где

$$x_{i,j} = x_{i-1,js_i} + x_{i-1,js_i+1}^{q^{m_{i-1}}} + \ldots + x_{i-1,js_i+s_i-1}^{q^{(s_i-1)m_{i-1}}}.$$

Рассмотрим алгоритм, в котором на каждом i-м шаге, $1 \le i \le k$, вычисляются все $x_{i,j}$, $j = 0, \ldots, n_i - 1$. Результат последнего шага, $x_{k,0}$, совпадает с искомой суммой.

На i-м шаге выполняются s_i операций Фробениуса $x \to x^{q^{lm_{i-1}}}$ на соответствующих наборах x_{i-1,js_i+l}, $j = 0, \ldots, n_i - 1$, где $l = 0, \ldots, s_i - 1$.

Количество сложений на всех шагах — в точности $s - 1$, что дает вклад $O(sn)$ в сложность алгоритма.

Сложность i-го шага при использовании алгоритма леммы 5.1 определяется $n^{\sigma/k}$ умножениями матриц размера $n^{(k+(k-i)\sigma)/2k} \times n^{(k+(k-i)\sigma)/2k}$ и $n^{(k+(k-i)\sigma)/2k} \times n$.

Заключительное рассуждение повторяет завершение доказательства леммы 5.2 (сложность i-го шага здесь оценивается так же, как и сложность $(k-i+1)$-го шага в лемме 5.2).

Замечание. Если порядок глубины не является принципиальным, то в леммах 5.2, 5.3 можно выбрать $k = \lceil \log_2 s \rceil$ и все $s_i = 2$. Оценка для сложности примет вид $O(n^{w_\tau/\tau} + n^{1/\tau} M(n))$, а для глубины — $O(\sigma \log^2 n)$.

Теорема 5.1 *Переход между соответствующими нормальным и стандартным базисами поля $GF(q^n)$ может быть выполнен схемой над $GF(q)$ сложности $O(n^{1,806})$ и глубины $O(\log n)$.*

Доказательство. Как следует из лемм 5.2 и 5.3, порядок сложности алгоритмов 2 и 3 определяется умножением матриц размера $n^\sigma \times n^{1-\sigma}$ и $n^{1-\sigma} \times n$ (на шагах **2** и **1** соответственно), и умножением матриц размера $n^{(1+\sigma-O(\delta))/2} \times n^{(1+\sigma-O(\delta))/2}$ и $n^{(1+\sigma-O(\delta))/2} \times n^{1+O(\delta)}$ (на шагах **1** и **2**), где $\sigma \in [0, 1]$.

Положим $\sigma = 0,195$. Т.к. $\sigma < 0,294(1-\sigma)$, то первое умножение выполняется со сложностью $O(n^{2-\sigma+\epsilon})$. Также выполнено $w_\tau/\tau < 1,806$, что следует из оценки (5.3) при выборе $\theta = 0,02$ и $h = 8$. Выбирая ϵ и δ достаточно малыми, имеем для обоих умножений оценку $O(n^{1,806})$.

Замечание. Заметим, что в действительности оценка сложности доказанной теоремы имеет вид $O(n^\nu)$, где

$$\nu > \min_{\sigma\in[0,1]} \max\{\omega(\sigma, 1-\sigma, 1),\ \omega((1+\sigma)/2, (1+\sigma)/2, 1)\},$$

где $\omega(\alpha, \beta, \gamma)$ — экспонента матричного умножения $T_{n^\alpha, n^\beta, n^\gamma}$.

Следствием доказанной теоремы является

Утверждение 5.1 *Для умножения и инвертирования в нормальном базисе поля $GF(q^n)$ можно построить схемы сложности $O(n^{1,806})$ и глубины $O(\log n)$.*

Отметим, что константы под знаком «O» в этих оценках очень велики — во-первых, из-за большого (порядка тысячи) числа итераций, и во-вторых, из-за больших мультипликативных констант в оценках для матричных умножений.

5.5 Дополнение

5.5.1 О методе Калтофена—Шаупа

Алгоритм 3 перехода к стандартному базису изложен в работе [70] с той разницей, что шаг **2** реализуется в схеме Горнера с m последовательно выполняемыми операциями Фробениуса. Оценка сложности получается такой же, что и в §5.3, однако оценка глубины имеет вид $O(m \log n)$ и является полиномиальной относительно n.

Авторами [70] также доказано существование алгоритма, имеющего сложность $O(n^{1,815})$ (результат [67] позволяет уточнить эту оценку до $O(n^{1,806})$). Доказательство основано на построении алгоритма такой сложности для в определенном смысле двойственной операции с использованием принципа транспозиции (см. п. 2.4) и не является конструктивным. Конструктивное доказательство содержится в §5.4.[10]

5.5.2 О вычислениях в произвольном стандартном базисе

Переход между произвольными стандартными базисами поля $GF(q^n)$ выполняется методом Брента—Кунга не сложнее, чем за $O(n^{1,667})$ операций. Действительно, пусть $x \in GF(q^n)$ представляется многочленом $f(\beta)$ в

[10] В известных автору работах после 1998 г., посвященных реализации операций в конечных полях, этот результат из [70] не упоминается, что отчасти объясняется отсутствием аналогичного алгоритма для перехода к нормальному базису.

стандартном базисе A_β с генератором β. Далее, пусть β представляется многочленом $\xi(\alpha)$ в стандартном базисе A_α с генератором α — корнем неприводимого многочлена $m_n(t)$. В таком случае $x = f(\xi(\alpha))$, т.е. в базисе A_α элемент x представляется многочленом $f(\xi(t)) \bmod m_n(t)$. Этот многочлен можно вычислить алгоритмом 1.

В частности, со сложностью $O(n^{1,667})$ и глубиной $O(\log n)$ можно перейти от произвольного стандартного базиса к стандартному базису, генератор которого также порождает нормальный базис поля — и обратно. Таким образом, из теоремы 5.1 автоматически вытекает следующий результат.

Теорема 5.2 *Переход между двумя произвольными нормальными или стандартными базисами поля $GF(q^n)$ может быть выполнен схемой над $GF(q)$ сложности $O(n^\nu)$ и глубины $O(\log n)$, где ν — из замечания к теореме 5.1.*

Для полноты картины отметим, что переход между двумя нормальными базисами можно выполнять быстрее, со сложностью порядка $M(n)$. Пусть элемент $x = x'_0\beta_0 + \ldots + x'_{n-1}\beta_{n-1}$ задан в нормальном базисе B_β с генератором β, и требуется найти представление $x = x_0\alpha_0 + \ldots + x_{n-1}\alpha_{n-1}$ в нормальном базисе B_α с генератором α. Пусть $\beta = b_0\alpha_0 + \ldots + b_{n-1}\alpha_{n-1}$. Легко проверить, что

$$\begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_{n-1} \end{bmatrix}^{B_\alpha} = \begin{bmatrix} b_0 & b_{n-1} & \cdots & b_1 \\ b_1 & b_0 & \cdots & b_2 \\ \cdots & \cdots & \cdots & \cdots \\ b_{n-1} & b_{n-2} & \cdots & b_0 \end{bmatrix}_{n\times n} \cdot \begin{bmatrix} x'_0 \\ x'_1 \\ \cdots \\ x'_{n-1} \end{bmatrix}^{B_\beta} .$$

Матрица перехода является *циркулянтной*, т.е. все ее строки получаются из первой строки последовательными циклическими сдвигами. Умножение на такую матрицу, как известно, сводится к умножению многочленов (см., например, [5]). А именно, все x_i являются коэффициентами многочлена

$$(b_{n-1} + b_{n-2}y + \ldots + b_0y^{n-1})(x'_{n-1} + x'_{n-2}y + \ldots + x'_0y^{n-1}) \mod y^n - 1,$$

который вычисляется со сложностью $O(M(n))$ и глубиной $O(\log n)$.[11]

[11] Такое умножение на самом деле есть *циклическая свертка*, которая фактически является частью алгоритмов умножения Штрассена и Шёнхаге (см., например, [59, гл. 8]).

5.5.3 Об умножении в нормальных базисах

В этом пункте будет рассмотрена еще одна пара алгоритмов перехода между нормальными и стандартными базисами, связанная со стандартным методом умножения в нормальных базисах. При определенных ограничениях для этих алгоритмов можно получить лучшую оценку сложности, чем для рассмотренных в основном тексте.

Умножение в нормальном базисе B выполняется стандартным алгоритмом Месси—Омура (см., например, [69, 4]) со сложностью $O(C_B n)$, где C_B — сложность базиса (см. п. 2.3). Непосредственно из определения вытекает, что сложность операции умножения на элемент базиса B не превосходит $2C_B - n$ при глубине $\lceil \log_2 n \rceil + 1$.

Как следует из теоремы 5.1, оценки сложности умножения в нормальном базисе стандартным алгоритмом и алгоритмом с использованием перехода к стандартному базису находятся в отношении $O(C_B/n^{0,806})$. Далее будет показано, что для базисов низкой сложности это отношение имеет больший порядок при условии, что q не очень велико.

Лемма 5.4 *Переход к нормальному базису B в поле $GF(q^n)$ может быть реализован схемой со сложностью $O(\sqrt{n}C_B) + O(n^{1,667})$ и глубиной*

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

Доказательство. Пусть элемент поля в стандартном базисе представлен многочленом $f(t)$ степени не выше $n-1$. Пусть $n \le rp$, где $p, r \sim \sqrt{n}$. Запишем,

$$f(t) = f_0(t^r) + t f_1(t^r) + \ldots + t^{r-1} f_{r-1}(t^r),$$

где $\deg f_i < p$.

АЛГОРИТМ 4.

0. Считаем, что нормальные представления для t^{ir}, $i = 0, \ldots, p-1$, вычислены предварительно.

1. Нормальное представление элементов $f_i(t^r)$ получается умножением $r \times p$-матрицы коэффициентов многочленов f_i на $p \times n$-матрицу представления элементов t^{ir} в нормальном базисе.

2. Далее, пусть $q_0 q^s < r \le (q_0+1)q^s$, где $0 < q_0 < q$ и $s \ge 0$. Для всех $i = (q_0-1)q^s, \ldots, r - q^s - 1$ вычислим элементы $g_i = t^{q^s} f_{i+q^s}(t^r) + f_i(t^r)$ в нормальном представлении (заметим, что $t^{q^s} = \alpha_s$, поэтому

сложность одной такой операции не превосходит $2C_B$) и положим $g_i = f_i(t^r)$ для $i < (q_0 - 1)q^s$ и $i \geq r - q^s$. Выполняется

$$f(t) = g_0 + tg_1 + \ldots + t^{r_1-1}g_{r_1-1}, \tag{5.4}$$

где $r_1 = q_0 q^s$. Повторяем указанную процедуру для формулы (5.4) и т.д. Для нахождения окончательного результата требуется всего $q_0 + (q-1)\lfloor \log_q(r-1) \rfloor \sim \min\{\sqrt{n}, q \log_q \sqrt{n}\}$ таких шагов, на которых выполняется в совокупности $r-1$ операций, состоящих из умножения на элемент нормального базиса и сложения.

Лемма 5.5 *Переход к стандартному базису в поле $GF(q^n)$ можно выполнить схемой сложности $O(n^{1,667}) + O(n^{1,5} \log q \log n \log\log n)$ и глубины $O(\sqrt{n} \log q \log n)$.*

Доказательство. Пусть $n \leq rp$ и $p, r \sim \sqrt{n}$. Элемент поля в нормальном базисе представим в виде

$$\gamma_0 + \gamma_1^q + \gamma_2^{q^2} + \ldots + \gamma_{r-1}^{q^{r-1}},$$

где все γ_i являются линейными комбинациями базисных элементов α_{jr}, $j = 0, \ldots, p-1$.

АЛГОРИТМ 5.

0. Считаем известными стандартные представления элементов α_{jr}, $j = 0, \ldots, p-1$.

1. Стандартное представление элементов γ_i получается умножением $r \times p$-матрицы коэффициентов γ_i в частичном базисе из элементов α_{jr} на $p \times n$-матрицу представления элементов α_{jr} в стандартном базисе.

2. Воспользуемся вариантом схемы Горнера. Положим $g_{r-1} = \gamma_{r-1}$. Далее вычисляем $g_{r-2} = \gamma_{r-2} + g_{r-1}^q$ и т.д. по формулам $g_i = \gamma_i + g_{i+1}^q$. Всего для завершения вычислений нужно выполнить последовательно $r-1$ действий, состоящих из возведения в степень q и сложения со сложностью $O(M(n) \log q)$ каждое (т.к. возведение в степень q выполняется в аддитивной цепочке длины $O(\log q)$).

Объединяя две доказанных леммы, получаем следующий результат.

Теорема 5.3 *Переход от стандартного базиса к нормальному базису B в поле $GF(q^n)$ может быть реализован схемой сложности $O(\sqrt{n}C_B) + O(n^{1,667})$, а обратный переход можно выполнить схемой сложности*

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

где C_B — сложность базиса B.

Окончательно, если $\log q = O(n^{0,167})$, то рассматриваемое отношение оценок сложности умножения для алгоритма Месси—Омура и алгоритма с переходом к стандартному базису принимает вид $O(C_B/n^{0,667})$, когда $C_B = O(n^{1,167})$, и $O(\sqrt{n})$ в противном случае (при $C_B \gtrsim n^{1,306}$ приоритет имеет исходная оценка $O(C_B/n^{0,806})$).

5.5.4 О реализации всех автоморфизмов Фробениуса

То, что выполнение некоторых операций в нормальном базисе может быть ускорено за счет перехода к стандартному базису (и затем, обратно), кажется правдоподобным в свете уже доказанного. Рассмотрим операцию вычисления всех x^{q^i}, $i = 0, \ldots, n-1$, которая в нормальном базисе не требует схемных затрат, и следует ожидать, что при ее реализации в стандартном базисе можно извлечь выгоду от перехода к нормальному базису.

Оценку $O(M(n^2)\log n) = O(n^2 \log^2 n \log\log n)$ можно получить методом работы [62] (см. также [59, гл. 14]).[12] В основе метода [62] лежит способ быстрого вычисления значений многочлена степени $n-1$ в n точках, которым определяется глубина метода $O(\log^2 n)$ в общем случае.

Однако, в известной степени более простым способом можно получить даже несколько лучшую оценку сложности, а именно, $O(nM(n)) = O(n^2 \cdot \log n \log\log n)$ при глубине $O(\log n)$.

Действительно, любым методом сложности $O(n^2)$ найдем разложение $x = x_0\alpha_0 + \ldots + x_{n-1}\alpha_{n-1}$ в некотором нормальном базисе B. По свойству нормального базиса наборы координат элементов x^{q^i} совпадают с точностью до циклического сдвига. Поэтому для завершения вычислений остается перейти обратно к стандартному представлению, т.е. вычислить произведение циркулянтной матрицы

$$X = \begin{bmatrix} x \\ x^q \\ \cdots \\ x^{q^{n-1}} \end{bmatrix}_{n\times n}^{B} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{n-1} \\ x_{n-1} & x_0 & \cdots & x_{n-2} \\ \cdots & \cdots & \cdots & \cdots \\ x_1 & x_2 & \cdots & x_0 \end{bmatrix}$$

[12] На самом деле, в работе [62] рассмотрена более общая операция вычисления всех $x^{q^i} \bmod f$, где многочлен f также является входом алгоритма.

на матрицу перехода C размера $n \times n$.

Компоненты вектора-произведения матрицы X на произвольный столбец $[c_0, c_1, \ldots, c_{n-1}]^T$ матрицы C вычисляются со сложностью $O(M(n))$ и глубиной $O(\log n)$ (см. п. 5.5.2), откуда и следуют заявленные оценки.[13]

5.5.5 О проверке линейной независимости нормальной системы

В приложениях часто встречается задача проверить, порождает ли выбранный элемент β нормальный базис в поле $GF(q^n)$, иначе говоря, является ли нормальная система $\{\beta, \beta^q, \ldots, \beta^{q^{n-1}}\}$ линейно независимой над $GF(q)$. Обычно предполагается, что β задан в стандартном представлении поля.

В работе [60] предложен алгоритм проверки, для которого, с учетом замечания из [62], справедлива оценка сложности $O(M^2(n)\log n) = O(n^2 \cdot \log^3 n \log^2 \log n)$. Глубина имеет порядок $O(\log^2 n)$, однако в методе [60] используется алгоритм Евклида, поэтому его затруднительно реализовать в виде схемы. В работе [70] предложен вероятностный алгоритм для той же задачи со сложностью (с учетом [67]) $O(n^{1,806})$. Покажем, что проверку линейной независимости нормальной системы можно выполнить схемой сложности $O(n^{1,806})$ и глубины $O(\log n)$.

Пусть известно, что некоторый элемент α является генератором нормального базиса B_α в поле $GF(q^n)$. Со сложностью $O(n^{1,806})$ и глубиной $O(\log n)$ можно получить представление $\beta = b_0\alpha_0 + \ldots + b_{n-1}\alpha_{n-1}$ в этом базисе. Нетрудно показать (см., например, [69, гл. 3]), что система $\{\beta, \beta^q, \ldots, \beta^{q^{n-1}}\}$ является базисом тогда и только тогда, когда обратима циркулянтная матрица

$$\begin{bmatrix} b_0 & b_{n-1} & \cdots & b_1 \\ b_1 & b_0 & \cdots & b_2 \\ \cdots & \cdots & \cdots & \cdots \\ b_{n-1} & b_{n-2} & \cdots & b_0 \end{bmatrix}_{n\times n}.$$

Обратимость такой матрицы, в свою очередь, эквивалентна тому, что многочлен

$$b(y) = b_{n-1} + b_{n-2}y + \ldots + b_0 y^{n-1}$$

обратим по модулю $y^n - 1$ (о свойствах циркулянтных матриц более подробно см. в [40, гл. 2]). Для обратимости многочлена по модулю другого необходимо и достаточно, чтобы они были взаимно просты.

[13] При помощи стандартного рассуждения оценку сложности можно привести к виду $O(n^2 \log n \log\log n / \log\log\log_q n)$, имеющему смысл при $q = o(n)$.

Чтобы установить, являются ли многочлены, один из которых, $y^n - 1$, фиксирован, взаимно простыми, можно воспользоваться вариантом метода Брента—Кунга. Пусть известно разложение

$$y^n - 1 = \varphi_1^{e_1}(y) \cdot \ldots \cdot \varphi_k^{e_k}(y)$$

на неприводимые множители. Обозначим $\deg \varphi_i = d_i$. Очевидно, $\sum d_i \leq n$. Пусть $n \leq rp$. Запишем,

$$b(y) = f_0(y) + f_1(y)y^r + \ldots + f_{p-1}(y)y^{(p-1)r},$$

где $\deg f_i < r$.

АЛГОРИТМ 6.

0. Считаем, что все остатки $\psi_{i,j}(y) = y^{ir} \bmod \varphi_j(y)$ заранее вычислены.

1. Операция вычисления остатков от деления многочлена степени $r-1$ на многочлены с постоянными коэффициентами $\varphi_j(y)$ является линейным $r \times d$-оператором (где $d \leq n$). Таким образом, нахождение всех остатков $\rho_{i,j}(y) = f_i(y) \bmod \varphi_j(y)$ сводится к умножению $p \times r$-матрицы, составленной из коэффициентов многочленов f_i, на $r \times d$-матрицу этого оператора.

2. Вычисляются все произведения $\rho_{i,j}(y)\psi_{i,j}(y)$. Сложность шага можно оценить как $O(pM(n))$, т.к. вычисление k произведений многочленов степени d_i по порядку не сложнее, чем умножение многочленов степени $n-1$ (на этом шаге выполняется p таких операций).

3. Поскольку

$$b(y) \bmod \varphi_j(y) = \sum_{i=0}^{p-1} \rho_{i,j}(y)\psi_{i,j}(y) \bmod \varphi_j(y),$$

то для завершения вычислений остается при каждом j сложить полученные на предыдущем шаге произведения $\rho_{i,j}(y)\psi_{i,j}(y)$ и привести результат по модулю $\varphi_j(y)$. Сложность этого шага также не превосходит $O(pM(n))$.

Если все остатки, найденные указанным алгоритмом, отличны от нуля, то β является генератором нормального базиса, и не является в противном случае. Выбирая $p, r \sim \sqrt{n}$ и учитывая, что $d \leq n$, имеем оценку сложности

$O(n^{1,667})$ и глубины $O(\log n)$ для алгоритма 6,[14] и $O(n^{1,806})$, $O(\log n)$ — для сложности и глубины алгоритма проверки линейной независимости в целом.

Для случая программной реализации отметим, что в предложенном методе требуется, чтобы один нормальный базис в поле $GF(q^n)$ уже был построен. Для нахождения нормального базиса можно воспользоваться каким-нибудь вероятностным алгоритмом, например, [60, 70].

В заключение автор выражает глубокую благодарность научному руководителю С. Б. Гашкову за постановку задач, а также коллективам кафедры дискретной математики механико-математического факультета Московского государственного университета имени М. В. Ломоносова и отдела теоретической кибернетики Института прикладной математики имени М. В. Келдыша РАН за всесторонние помощь и поддержку.

[14] При программной реализации и отсутствии ограничения на глубину, для нахождения НОД двух многочленов, конечно, можно воспользоваться алгоритмом Евклида, имеющим оценку сложности $O(n \log^2 n \log\log n)$ и глубины $O(\log^2 n)$, и не требующим разложения многочлена $y^n - 1$ на множители.

Список литературы

[1] Алексеев В. Б., Сложность умножения матриц. // Кибернетический сборник. Вып. 25. — М.: Мир, 1988. — С. 189–236.

[2] Ахо А., Хопкрофт Дж., Ульман Дж. Проектирование и анализ вычислительных алгоритмов. — М.: Мир, 1979.

[3] Берлекемп Э. Алгебраическая теория кодирования. — М.: Мир, 1971.

[4] Болотов А. А., Гашков С. Б. О быстром умножении в нормальных базисах конечных полей. // Дискретная математика. — 2001. — Вып. 13, №3. — С. 3–31.

[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., Гашков И. Б., Гашков С. Б. О сложности булевых схем для арифметики в некоторых башнях конечных полей // Вестник МГУ. Серия 1. Математика. Механика. — 2006. — №5. — С. 10–16.

[7] Гашков С. Б. Замечания о быстром умножении многочленов, преобразовании Фурье и Хартли.// Дискретная математика. — 2000. — Вып. 12, №3. — С. 124–153.

[8] Гашков С. Б. Замечание о минимизации глубины булевых схем. // Вестник МГУ. Серия 1. Математика. Механика. — 2007. — №2.

[9] Гашков С. Б., Гашков И. Б. О сложности вычисления дифференциалов и градиентов. // Дискретная математика. — 2005. — Вып. 17, №3. — С. 45–67.

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

[11] Гашков С. Б., Кочергин В. В. Об аддитивных цепочках векторов, вентильных схемах и сложности вычисления степеней. // Методы дискретного анализа в теории графов и сложности. — 1992. — Том 52. — С. 22–40.

[12] Гашков С. Б., Сергеев И. С. О применении метода аддитивных цепочек к инвертированию в конечных полях. // Дискретная математика. — 2006. — Вып. 18, №4. — С. 56–72.

[13] Гашков С. Б., Хохлов Р. А. О глубине логических схем для операций в полях $GF(2^n)$. // Чебышевский сборник. — 2003. — Т. 4, вып. 4(8). — С. 59–71.

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

[15] Касим-Заде О. М. Об одной мере сложности схем из функциональных элементов. // Проблемы кибернетики. Вып. 38. — М.: Наука, 1981. — С. 117–179.

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

[17] Коблиц Н. Курс теории чисел и криптографии. — М.: ТВП, 2001.

[18] Коновальцев И. В. Об одном алгоритме решения линейных уравнений в конечных полях. // Проблемы кибернетики. Вып. 19. — М.: Наука, 1967. — С. 269–274.

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

[20] Ложкин С. А. О связи между глубиной и сложностью эквивалентных формул и о глубине монотонных функций алгебры логики. // Проблемы кибернетики. Вып. 38. — М.: Наука, 1981. — С. 269–271.

[21] Лупанов О. Б. О вентильных и контактно-вентильных схемах. // Доклады АН СССР. — 1956. — Т. 111(6). — С. 1171–1174.

[22] Лупанов О. Б. О синтезе некоторых классов управляющих систем. // Проблемы кибернетики. Вып. 10. — М.: Физматлит, 1963. — С. 63–97.

[23] Лупанов О. Б. Асимптотические оценки сложности управляющих систем. — М.: Изд-во МГУ, 1984.

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

[25] Офман Ю. П. Алгоритмическая сложность дискретных функций. // Доклады АН СССР. — 1962. — Т. 145(1). — С. 48–51.

[26] Пан В. Я. О схемах вычисления произведений матриц и обратной матрицы. // Успехи мат. наук. — 1972. — 27, №5. — С. 249–250.

[27] Серпинский В. 250 задач по элементарной теории чисел. — М.: Просвещение, 1968.

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

[29] Тоом А. Л. О сложности схемы из функциональных элементов, реализующей умножение целых чисел. // Доклады АН СССР. — 1963. — Т. 150(3). — С. 496–498.

[30] Хохлов Р. А. Реализация логическими схемами операций умножения и инвертирования в конечных полях характеристики два. — Канд. дисс., МГУ, 2005.

[31] Храпченко В. М. Об асимптотической оценке времени сложения параллельного сумматора. // Проблемы кибернетики. Вып. 19. — М.: Наука, 1967. — С. 107–120.

[32] Храпченко В. М. Различие и сходство между задержкой и глубиной. // Проблемы кибернетики. Вып. 35. — М.: Наука, 1979. — С. 141–168.

[33] Яблонский С. В. Введение в дискретную математику. — М.: Высшая школа, 2002.

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

[35] Ash D., Blake I., Vanstone S. Low complexity normal bases. // Discrete Applied Math. — 1989. — V. 25. — P. 191–210.

[36] Beame P., Cook S., Hoover H. Log depth circuits for division and related problems. // SIAM J. Comput. — 1986. — V. 15, №4. — P. 994–1003. [Русский перевод: Бим П., Кук С., Гувер Г. Схемы логарифмической глубины для деления и связанных с ним проблем. // Кибернетический сборник. Вып. 28. М.: Мир, 1991. С. 134–150.]

[37] Bernstein D. J. Multidigit multiplication for mathematicians. — 2001. — http://cr.yp.to/papers.html#m3.

[38] Bernstein D. J. Pippenger's exponentiation algorithm. — 2002. — http://cr.yp.to/papers.html#pippenger.

[39] Bernstein D. J. The transposition principle. — http://cr.yp.to/transposition.html.

[40] Bini D., Pan V. Polynomial and matrix computations. Vol. 1. — Boston: Birkhäuser, 1994.

[41] Bluestein L. A linear filtering approach to the computation of the discrete Fourier transform. // IEEE Northeast El. Res. Eng. Meet. — 1968. — V. 10. — P. 218–219.

[42] Brauer A. On addition chains. // Bull. AMS. — 1939. — V. 45. — P. 736–739.

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

[44] Brent R., Kung H. Fast algorithms for manipulating formal power series. // J. ACM. — 1978. — V. 25, №4. — P. 581–595.

[45] Bürgisser P., Clausen M., Shokrollahi M. A. Algebraic complexity theory. — Berlin—Heidelberg: Springer-Verlag, 1997.

[46] Cantor D. On arithmetical algorithms over finite fields. // J. Comb. Theory. — 1989. — V. A50. — P. 285–300.

[47] Cantor D., Kaltofen E. On fast multiplication of polynomials over arbitrary algebras. // Acta Inf. — 1991. — V. 28, №7. — P. 693–701.

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

[49] Cooley J., Tukew J. An algorithm for the machine calculation of complex Fourier series. // Math. Comp. — 1965. — V. 19. — P. 297–301.

[50] Coppersmith D. Rectangular matrix multiplication revisited. // J. Complexity. — 1997. — V. 13. — P. 42–49.

[51] Coppersmith D., Winograd S. Matrix multiplication via arithmetic progressions. // J. Symb. Comput. — 1990. — V. 9. — P. 251–280.

[52] Eberly W. Very fast parallel polynomial arithmetic. // SIAM J. Comput. — 1989. — V. 18, №5. — P. 955–976.

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

[54] Fiduccia C. M. On the algebraic complexity of matrix multiplication. — Ph. D. thesis, Brown Univ., 1973.

[55] Fürer M. Faster integer multiplication. — 2007. — http://www.cse.psu.edu/~furer/Papers/mult.pdf.

[56] Gao S., von zur Gathen J., Panario D. Gauss periods and fast exponentiation in finite fields. // Proc. Latin'95 (Valparaiso, Chile). Lecture Notes in Comp. Sci. — 1995. — V. 911. — P. 311–322.

[57] von zur Gathen J. Inversion in finite fields using logarithmic depth. // J. Symb. Comput. — 1990. — V. 9. — P. 175–183.

[58] von zur Gathen J., Gerhard J. Arithmetic and factorization of polynomials over $\mathbb{F}_2$. // Proc. ISSAC'96 (Zürich, 1996). — P. 1–9.

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

[60] von zur Gathen J., Giesbrecht M. Constructing normal bases in finite fields. // Symb. Comp. — 1990. — V. 10. — P. 547–570.

[61] von zur Gathen J., Nöcker M. Exponentiation in finite fields: theory and practice. // Applied Algebra, AAECC-12, Lecture Notes in Comp. Sci. — 1997. — V. 1255. — P. 88–113.

[62] von zur Gathen J., Shoup V. Computing Frobenius maps and factoring polynomials. // Comput. Complexity. — 1992. — V. 2. — P. 187–224.

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

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

[65] Hoover H., Klawe M., Pippenger N. Bounding fan-out in logical networks. // J. ACM. — 1984. — V. 31, №1. — P. 13–18.

[66] Hopcroft J. E., Kerr L. R. On minimizing the number of multiplications necessary for matrix multiplication. // SIAM J. Appl. Math. — 1971. — V. 20, №1. — P. 30–36.

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

[68] Itoh T., Tsujii S. A fast algorithm for computing multiplicative inverses in $GF(2^n)$ using normal basis. // Inform. and Comp. — 1988. — V. 78. — P. 171–177.

[69] Jungnickel D. Finite fields: structure and arithmetics. — Mannheim: Wissenschaftsverlag, 1995.

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

[71] Kaltofen E., Singer M. Size efficient parallel algebraic circuits for partial derivatives. // IV ICCAPR Conf. (Singapore, 1991). — P. 133–145.

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

[73] Litow B., Davida G. $O(\log n)$ parallel time finite field inversion. // Proc. Aegean Workshop on Computing. Lecture Notes in Comp. Sci. — 1988. — V. 319. — P. 74–80.

[74] Massey J. L., Omura J. K., Apparatus for finite fields computation. // US patent application. — 1986. — №4587627.

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

[76] Mullin R., Onyszchuk I., Vanstone S., Wilson R. Optimal normal bases in $GF(p^n)$. // Discrete Applied Math. — 1988/89. — V. 22. — P. 149–161.

[77] Pan V. Y. Complexity of parallel matrix computations. // Theor. Comp. Sci. — 1987. — V. 54. — P. 65–85.

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

[79] Reif J., Tate S. Optimal size integer division circuits. // SIAM J. Comput. — 1990. — V. 19, №5. — P. 912–925.

[80] Rosser J., Schoenfeld L. Approximate formulas for some functions of prime numbers. // Ill. J. Math. — 1962. — V. 6 — P. 64–94.

[81] Schönhage A. Schnelle berechnung von kettenbruchentwicklungen. // Acta Inf. — 1971. — V. 1. — P. 139–144.

[82] Schönhage A. A lower bound for the length of addition chains. // Theor. Comp. Sci. — 1975. — V. 1. — P. 1–12.

[83] Schönhage A. Schnelle multiplikation von polynomen über körpern der charakteristik 2. // Acta Inf. — 1977. — V. 7. — P. 395–398.

[84] Schönhage A., Strassen V. Schnelle multiplikation großer zahlen. // Computing. — 1971. — V. 7. — P. 271–282. [Русский перевод: Шёнхаге А., Штрассен В. Быстрое умножение больших чисел. // Кибернетический сборник. Вып. 10. М.: Мир, 1973. С. 87–98.]

[85] Strassen V. Gaussian elimination is not optimal. // Numer. Math. — 1969. — B. 13, №4. — P. 354–356. [Русский перевод: Штрассен Ф. Алгоритм Гаусса не оптимален. // Кибернетический сборник. Вып. 7. М.: Мир, 1971. С. 67–70.]

[86] Strassen V. Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten. // Numer. Math. — 1973. — B. 20 — P. 238–251.

[87] Strassen V. Vermeidung von divisionen. // J. reine u. angew. Math. — 1973. — V. 264. — P. 182–202.

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

[89] Takagi N., Yoshiki J., Takagi K. A fast algorithm for multiplicative inversion in $GF(2^n)$ using normal basis. // IEEE Trans. on Comp. — 2001. — V. 50, №5. — P. 394–398.

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

Работы автора по теме диссертации

[91] Сергеев И. С. О реализации некоторых операций конечных полей характеристики 2 схемами логарифмической глубины. // Материалы XVI Международной школы-семинара «Синтез и сложность управляющих систем» (Санкт-Петербург, 26–30 июня 2006 г.). — М.: Изд-во мех.-матем. факультета МГУ. — 2006. — с. 101–103.

[92] Сергеев И. С. Об инвертировании в конечных полях характеристики 2 с логарифмической глубиной. // Вестник МГУ. Серия 1. Математика. Механика. — 2007. — №1. — С. 28–33.

[93] Сергеев И. С. О схемах логарифмической глубины для инвертирования в конечных полях характеристики два. // Математические вопросы кибернетики. Вып. 15. — М.: Физматлит, 2006. — С. 35–64.