МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра математического моделирования и анализа данных

АЛИФЕРОВИЧ РОСТИСЛАВ МИХАЙЛОВИЧ

ГЕНЕРАЦИЯ ПРОСТЫХ ЧИСЕЛ И АРИФМЕТИКА В ПОЛЯХ ХАРАКТЕРИСТИКИ 2 В КРИПТОГРАФИИ

Отчет о преддипломной практике
студента 5 курса 9 группы

"Допустить к защите"	**Руководитель**
Руководитель работы	Агиевич Сергей Валерьевич
с предварительной оценкой ______	канд. физ.-мат. наук

____

"___"__________ 2015 г

МИНСК 2015

АННОТАЦИЯ

Алиферович Р.М. Эффективная реализация криптографических алгоритмов: Генерация простых чисел и арифметика в полях характеристики 2 в криптографии. Дипломная работа / Минск: БГУ, 2015. - 14 с.

Оптимизация алгоритмов для арифметики в нормальных и оптимальных нормальных базисах в полях характеристики 2.

АНАТАЦЫЯ

Аліфяровіч Р.М. Эффектыўная рэалізацыя крыптаграфічных алгарытмаў: Генерацыя простых лікаў і арыфметыка ў палях характарыстыкі 2 у крыптаграфіі. Дыпломная работа / Мінск: БДУ, 2015. - 14 с.

Аптымізацыя алгарытмаў для арыфметыкі ў нармальных і аптымальных нармальных базісах у палях характарыстыкі 2.

ANNOTATION

Aliferovich R.M. Effective implementation of cryptographic algorithms: generation of prime numbers and arithmetics in the fields of characteristic 2 in cryptography. Final thesis / Minsk: BSU, 2015. - 14 p.

Optimization of algorithms for arithmetic in normal and optimal normal bases in the fields of characteristic 2.

Содержание

Введение

Современные криптографические системы с открытым ключом существенно используют вычисления с большими целыми числами. Более того, во многих криптосистемах (например, RSA) используются большие простые числа. Генерация простых чисел является важным компонентом таких систем. Обычно генерация простых чисел выполняется по следующей схеме: выбрать число-кандидат, проверить делимость числа на малые простые, проверить простоту с помощью теста Рабина Ц Миллера. Также современная криптография немыслима без использования эллиптических кривых. Этим математическим объектам посвящен целый раздел в криптографии. Для реализации достаточно надежных эллиптических кривых используются расширения поля $GF(2)$. Важно быстро уметь проводить операции над объектами этих полей, чему собственно и посвящен раздел по ускорению вычислений с использованием оптимальных нормальных базисов. Их преимущество в возможности быстро возводить в степень, кратную двум. Дипломная работа посвящена ускорению проверки делимости числа на малые простые и оптимизации простейших арифметических операций в полях характеристики 2.

1 Постановка задач

В этой работе для проверки простоты числа будем использовать тест Рабина-Миллера. Цель данной курсовой работы заключается в выработке некоторого предварительного этапа, после которого вероятность теста Рабина-Миллера была бы выше, чем без него. Речь пойдет о следующем:

1.1 Фактор-множество

На любом конечном интервале натуральных чисел почти половина чисел делится на 2, примерно треть на три и так далее. Таким образом, проверив делимость числа на малые простые, мы многократно повысим вероятность того, что оно простое. Поэтому есть смысл ее проверить. Создадим множество простых последовательных чисел, назовем его фактор-множеством, и обозначим F, где $F = \{x_i \mid x_i \in PRIMES\}$. Ниже будет описан алгоритм быстрого нахождения остатка от деления на малое число ModDigit. В нем есть некоторый порог m, который означает, что можно быстро найти остаток от деления числа n на число, не большее m. При построении фактор-множества придется это учесть.

При увеличении мощности фактор-множества растет вероятность успеха теста Миллера-Рабина, и в предельном случае когда фактор-множество максимально, вероятность теста равна 1, но асимптотика времени работы - экспоненциальна. Таким образом задача состоит в оценке мощности фактор-множества.

1.2 Разбиение

Существует также следующая оптимизация: фактор-множество можно разбить на подмножества, внутри которых произведение чисел меньше порога для алгоритма ModDigit. Пусть $F = \{F_i'\}, F_i' = \{x_{ij}\}, \prod x_{ij} < m, m$ - порог алоритма ModDigits. Очевидно, что мощность множества $\{F_i'\}$ меньше мощности множества $\{x_i\}$. В этом случае количество вызовов алгоритма ModDigits меньше. Тогда множество остатков нужно считать немного по-другому: сначала находим все $n_i' = n \ (mod \ \prod x_{ij})$, затем $n_{ij}' = n_i' \ (mod \ x_{ij})$.

$R_n = \{n'_{ij}\}$. Таким образом стоит задача оптимального разбиения фактор-множества.

1.3 Увеличение порога

Предположим, что решив задачу оценки мощности фактор-множества, обнаружим, что этот порог настолько большой, что в фактор-множестве найдутся числа, большие порога для алгоритма быстрого деления. Тогда встает еще одна задача: можно ли модифицировать алгоритм ModDigits таким образом, чтобы его порог можно было увеличить в достаточное число раз?

1.4 Множество кандидатов

Составим множество остатков $R_n = \{n_i = n \ (mod\ x_i) \mid x_i \in F\}$. Если хотя бы одно из n_i равно нулю, то число n - составное. Отметим, чтобы составить множество R_{n+2} не нужны приведения по модулю больших чисел, а всего лишь сложение по модулю для маленьких $R_{n+2} = \{n_i + 2 \ (mod\ x_i) \mid n_i \in R_n\}$. Таким образом можно быстро проверить делимость на малые простые некоторого множества кандидатов, назовем его K. Понятно, что когда-либо в множество кандидатов добавится число, взаимно простое с фактор-множеством. Только при этом может статься, что мощность множества кандидатов велика настолько, что проверка этого множества требует больше времени, чем тест Миллера-Рабина для случайных чисел. Поэтому стоит задача оценки мощности множества кандидатов.

2 Оптимизация проверки простоты

2.1 Деление на малый разряд

При пробных делениях на простые делитель много меньше основания системы счисления $b = 2^{\omega}$. Этим можно воспользоваться и заменить при вычислениях медленное деление разрядов на быстрое их умножение. Такая замена реализована в следующем алгоритме.

Алгоритм 1 ModDigit

Вход: $u = (u_{n-1} \dots u_1 u_0)_b$ — делимое, m — модуль, $0 < m < \sqrt{b}$.
Выход: $u_b \mod m$.
Шаги:

1. $(r_1 r_0)_b \leftarrow 0$.
2. $m^* \leftarrow b \mod m$.
3. Для $i = n-1, \dots, 0$:
   (a) $(r_1 r_0)_b \leftarrow (r_1 m^* + r_0) m^* + u_i$.
4. Пока $r_1 \neq 0$:
   (a) $(r_1 r_0)_b \leftarrow r_1 m^* + (r_{0_b} \mod m)$.
5. Возвратить $r_{0_b} \mod m$.

Корректность: На шагах алгоритма неявно вычисляется сумма

$$\begin{aligned} r &= u_{n-1}(m^*)^{n-1} + u_{n-2}(m^*)^{n-2} + \dots + u_1 m^* + u_0 \\ &= (\dots(u_{n-1} m^* + u_{n-2}) m^* + \dots + u_1) m^* + u_0, \end{aligned}$$

которая сравнима с u по модулю m. Контролируется регистр $(r_1 r_0)_b$, содержимое которого сравнимо с r по модулю m.
В регистр $(r_1 r_0)_b$ на шаге 3.1 помещается значение

$$u_i + r_0 m^* + r_1 (m^*)^2 \leq (b-1)(1 + (m-1) + (m-1)^2) = (b-1)(m^2 - m + 1) < b^2.$$

Полученная оценка означает, что переполнение при записи не произойдет.
Сложность. Требуется $2+d$ деления разрядов, $2n+d$ умножений и столько же сложений. Здесь d — число дополнительных итераций на шаге 4.
Оценим d. По окончании первой дополнительной итерации

$$(r_1 r_0)_b \leq (b-1)(m-1) + (m-1) = bm,$$

а по окончании второй

$$(r_1 r_0)_b \leq m(m-1) + (m-1) = m^2 - 1 < b.$$

Таким образом, $d \leq 2$.
Окончательно получаем: $C_n(ModDigit) = 4D + (2n+2)M + (2n+2)A$.

Таким образом видим, что перед запуском более трудоемкого теста Рабина-Миллера или теста Ферма, можно предварительно проверить делимость длинного числа на малые простые. С имеющимся у нас алгоритмом малые простые числа можно брать в диапазоне $[2...\sqrt{b}]$. Известна оценка количества простых чисел от 1 до n: $\pi(x) \to \frac{x}{lnx}$. Понятно, что брать x, близкий к числу n невыгодно, так как время работы алгоритма перебора будет экспоненциальным. Зато в диапазоне $[2...\sqrt{b}]$ можно сделать перебор по простым делителям. Если число n делится хоть на одно из них, то n однозначно не является простым.

2.2 Деление на несколько малых разрядов

Следующий алгоритм может использоваться для деления сразу на несколько малых простых.

Алгоритм 2 ModDigits

Вход: $u = (u_{n-1}...u_0)_b$ - делимое, $m_1, ..., m_s$ - взаимно простые модули, $\prod_{i=1}^{s} m_i < \sqrt{b}$.
Выход: $(u \ mod \ m_1, ..., u \ mod \ m_s)$.
Шаги:

1. $m \leftarrow 1$.
2. Для $i = 1, ..., s$:
3. $\quad m \leftarrow m \cdot m_i$.
4. $m \leftarrow ModDigit(u, m)$.
5. Возвратить $(m \ mod \ m_1, ..., m \ mod \ m_s)$.

Сложность: $C_n(ModDigit) + (s-1)M + sD = (4+s)D + (2n+4)(1M+1A) + (s-1)M$. Для сравнения, расчет вычетов $u \ mod \ m_i$ по отдельности выполняется за время $4sD + s(2n+4)(1M+1A)$.

Имея этот алгоритм, можно несколько соптимизировать предварительное деление n на малые простые в диапазоне $[2...\sqrt{b}]$. Числа из этого диапазона можно разбить на группы чисел$(p_{i_1}, p_{i_2}...p_{i_n})$, произведение которых меньше $\sqrt{b}$. Тогда для этих групп можно провести алгоритм ModDigits. Задача состоит в оптимальности разбиения диапазона на группы. Отметим, что для всех простых битовой длины $\ln(\sqrt{b}) - 1$ нельзя подобрать такого малого простого, чтобы их можно было объединить в одну группу.Таких чисел примерно $\frac{\sqrt{b}}{\ln(\sqrt{b})} - \frac{\sqrt{b}}{2(\ln(\sqrt{b})-1)}$, что довольно много.

2.3 Тест Миллера-Рабина

Алгоритм 3 Miller-Rabin

Вход: n - число для проверки на простоту, $base$ - базовое множество, r - количество раундов (мощность $base$).

Выход: Ответ "составное"или "вероятно простое".

Шаги:

1. Представить $n-1$ в виде $2^s t, t$ - нечетно.
2. r раундов:
3. $\quad a \leftarrow base[r]$
4. $\quad a \leftarrow a^t \mod n$
5. $\quad$ Если ($x = 1$или $x = n-1$)
6. $\qquad r \leftarrow r-1$, перейти на шаг 2.
7. $\quad$ Цикл $s-1$ раз:
8. $\qquad x \leftarrow x^2 \mod n$
9. $\qquad$ Если ($x = n-1$)
10. $\qquad\quad r \leftarrow r-1$, перейти на шаг 2.
11. $\quad$ Вернуть "составное".
12. Вернуть "вероятно простое".

Сложность: $O(\log_2^3 n)$. В программе будет использовано базовое множество $base = \{2, 7, 61\}$, которое гарантирует безошибочность теста на числах до 4759123141.

3 Базисы конечных полей

Здесь приведем основные сведения о стандартных и нормальных (в том числе и оптимальных) базисах описанные в [1].

3.1 Стандартные и нормальные базисы

Через $GF(q^n)$ обозначим конечное поле порядка q^n, рассматриваемое как расширение степени n поля $GF(q)$ порядка q. В качестве представления элементов поля $GF(q^n)$ используем многочлены степени не более $n-1$ с коэффициентами из поля $GF(q)$. Если многочлены записаны в стандартном базисе

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

(в этом случае элемент α называем генератором базиса), то сложение элементов поля $GF(q^n)$ сводится к покомпонентному сложению в поле $GF(q)$ векторов коэффициентов, соответствующих данным многочленам, а умножение элементов поля есть умножение соответсвующих многочленов над полем $GF(q)$, выполняемое по модулю неприводимого над полем $GF(q)$ многочлена $g(x)$, определяющего рассматриваемое представление поля.

Иногда вместо стандартного базиса удобнее так называемый нормальный базис, то есть базис вида

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

который порождается генератором α стандартного базиса - корнем неприводимого над полем $GF(q)$ многочлена $g(x)$ в своем поле разложения $GF(q^n)$. Нормальный базис существует для любого n, но порождается не всяким неприводимым многочленом $g(x)$, так как составляющие его степени элемента α должны быть линейно независимыми над полем $GF(q)$.

Если система степеней

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

образует нормальный базис, то любой элемент ζ поля $GF(q^n)$ однозначно представляется в виде

$$\zeta = x_0\alpha + x_1\alpha^q + x_2\alpha^{q^2} + ... + x_{n-1}\alpha^{q^{n-1}}$$

где $x_0, ..., x_{n-1}$ - коэффициенты из поля $GF(q)$.

Сложение в нормальном базисе, как и в стандартном, есть покомпонентное сложение векторов коэффициентов в поле $GF(q^n)$.

Возведение в степень q (а значит, и в любую степень q^m) в нормальном базисе представляет собой циклический сдвиг коэффициентов, так как

$$\zeta^q = x_{n-1}\alpha + x_0\alpha^q + x_1\alpha^{q^2} + ... + x_{n-2}\alpha^{q^{n-1}}$$

Рассмотрим умножение в нормальных базисах.

Сложностью C_B произвольного нормального базиса

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

называется число ненулевых элементов в матрице T, i-я строка которой есть вектор коэффициентов элемента $\alpha\alpha^{q^i}$ поля $GF(q^n)$ относительно базиса B, то есть

$$\alpha\alpha^{q^i} = \sum_{j=0}^{n-1} t_{i,j}\alpha^{q^i}$$

Это определение мотивируется алгоритмом Месси-Омуры умножения в нормальном базисе B.

Пусть

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

- произвольные элементы поля $GF(q^n)$, разложенные по нормальному базису B. Тогда их произведение можно вычислить по формуле

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

где разность $i-j$ вычисляется по модулю n, а так как

$$\alpha^{(q^{i-j}+1)q^j} = (\alpha^{q^{i-j}+1})^{q^j} = (\sum_{k=0}^{n-1} t_{i-j,k}\alpha^{q^k})^{q^j} = \sum_{k=0}^{n-1} t_{i-j,k}\alpha^{q^{k+j}} = \sum_{m=0}^{n-1} t_{i-j,m-j}\alpha^{q^m},$$

где разность $m-j$ и сумма $k+j$ тоже вычисляются по модулю n, то

$$\pi = \sum_{m=0}^{n-1} p_m\alpha^{q^m},$$

где

$$p_m = \sum_{i,j=0}^{n-1} t_{i-j,m-j}x_i y_j.$$

Определив матрицу A равенствами $a_{i,j} = t_{i-j,-j}$, где $i-j$ и $m-j$ вычисляются по модулю n, замечаем, что предыдущую формулу можно переписать в виде

$$p_m = \sum_{i,j=0}^{n-1} t_{i-j,m-j} x_i y_j = \sum_{k,l=0}^{n-1} t_{k-l,-l} x_{k+m} y_{l+m} = \sum_{i,j=0}^{n-1} a_{i,j} x_{i+m} y_{j+m} =$$
$$= \sum_{i,j=0}^{n-1} a_{i,j} S^m(x_i) S^m(y_j),$$

где S^m - операция циклического сдвига координат вектора на m компонент, а

$$A(x,y) = \sum_{i,j=0}^{n-1} a_{i,j} x_i y_j$$

- билинейная форма, связанная с матрицей A.

Матрица A симметрическая и число C_B ее ненулевых элементов, а также сумма элементов такие же, как и у матрицы T. Для вычисления билинейной формы $A(x,y)$ достаточно выполнить $2C_B + n - 1$ сложений и умножений в поле $GF(q)$. Если пренебречь временем выполнения циклических сдвигов, то сложность выполнения умножения над нормальным базисом поля $GF(q^n)$ оценивается сверху как $n(2C_B + n - 1)$ операций в поле $GF(q)$, что видно из следующей основной формулы:

$$\xi\zeta = A(\xi,\zeta)\alpha + A(\xi^{q^{n-1}}, \zeta^{q^{n-1}})\alpha^q + A(\xi^{q^{n-2}}, \zeta^{q^{n-2}})\alpha^{q^2} + ... + A(\xi^q, \zeta^q)\alpha^{q^{n-1}}.$$

Таким образом, арифметическая сложность умножения зависит только от количества ненулевых элементов C_B в матрице A. Верхняя граница сложности умножения в произвольном нормальном базисе является кубической. Матрица A (таблица умножения в базисе B) однозначно определяет операцию умножения в рассматриваемом поле.

О сложности нормальных базисов известно следующее:

Теорема 3.1. *Для любого нормального базиса B поля $GF(q^n)$ его сложность C_B не меньше $2n-1$. Более того, если $q = 2$, то сложность нечетна.*

Нормальные базисы, для которых достигается эта граница, называют оптимальными.

Для вычисления обратного элемента в оптимальном нормальном базисе используется следующая формула: $x^{-1} = x^{2n-2}, x \neq 0$. Существует эффективный алгоритм подсчета правой части этой формулы.

3.2 Оптимальные нормальные базисы и нормальные базисы низкой сложности

Оптимальные нормальные базисы были описаны в [4]. Они удачно могут быть использованы в мультиплере Масси-Омуры. Впоследствии в [5] было показано, что других оптимальных базисов, кроме найденных в [4], не существует.

Поскольку оптимальные нормальные базисы существуют не во всех полях, то необходимо заранее выбирать поле для операций. В украинском стандарте ДСТУ 4145-2002 для алгоритмов над эллиптическими кривыми в случае использования оптимальных нормальных базисов рекомендуют использовать следующие поля:

Таблица 1 — Допустимые основные поля с оптимальным нормальным базисом

Степень поля n	173	179	191	233	239	251	281
Степень поля n	293	359	419	431	443	491	509

Все степени, приведенные в таблице, являются простыми Софи-Жермен, т.е. для них выполняется условие: если p - простое Софи Жермен, то $2p+1$ - простое число.

Различают три типа оптимальных нормальных базисов в поле $GF(q^n)$ по типу их построения.

Первый тип оптимальных нормальных базисов можно построить, когда $n+1=p$ - простое число, а q - примитивный корень по модулю p. В этом случае генератором оптимального нормального базиса будет один из примитивных корней p-й степени из единицы в поле $GF(q^n)$.

Второй тип оптимальных нормальных базисов возникает, когда $2n+1=p$ - простое число, а элемент q, как и в первом случае, - примитивный корень по модулю p. Генератором этого базиса служит элемент $\alpha=\zeta+\zeta^{-1}$, где ζ - примитивный корень p-й степени из единицы в поле $GF(q^{2n})$.

Третий тип оптимальных нормальных базисов порождается, когда $2n+1=p$ - простое число, $p\equiv 3 \mod 4$, а q - квадратичный вычет по модулю p и любой квадратичный вычет представляется в виде степени q по модулю p (или, другими словами, порядок элемента q по модулю p равен n). Как и в случае базиса второго типа в качестве порождающего элемента базиса третьего типа берется $\alpha=\zeta+\zeta^{-1}$, где ζ - примитивный корень p-й степени из единицы в поле $GF(q^{2n})$.

Далее будем рассматривать только степени полей, предложенные стандартом ДСТУ 4145-2002. Все предложенные степени образуют либо ОНБ 2-го, либо ОНБ 3-го типа. [2]

Таблица 2 — Типы оптимальных нормальных базисов

Степень поля n	173	179	191	233	239	251	281
Тип ОНБ	2	3	3	2	3	3	2
Степень поля n	293	359	419	431	443	491	509
Тип ОНБ	2	3	3	3	3	3	2

4 Арифметика в нормальных базисах

4.1 Алгоритм генерации оптимальных нормальных базисов второго типа

Базис второго типа возникает, когда $2n+1 = p$ - простое число, q - является примитивным корнем из 1 по модулю p. Элемент ζ тогда будет примитивным корнем степени p из 1 в поле $GF(q^{2n})$, в качестве порождающего элемента оптимального нормального базиса нужно взять $\alpha = \zeta + \zeta^{-1}$.

4.1.1 Доказательство оптимальности.

Так как $q^n \equiv -1 \mod p$, то

$$\alpha^{q^n} = \zeta^{q^n} + \zeta^{-q^n} = \zeta + \zeta^{-1} = \alpha,$$

значит, $\alpha^{q^n} = \alpha$, поэтому α принадлежит подполю $GF(q^n)$ поля $GF(q^{2n})$. Система

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

линейно независима, так как

$$\alpha^{q^k} = \zeta^{q^k} + \zeta^{-q^k} = \zeta^{q^k} + \zeta^{q^{k+n}}$$

в силу эквивалентности $q^{k+n} \equiv -q^k \mod p$, а система

$$\{\zeta, \zeta^q, \zeta^{q^2}, ..., \zeta^{q^{2n-1}}\}$$

линейно независима. Можно проверить, что при $n > k > 0$

$$\alpha\alpha^{q^k} = (\zeta + \zeta^{-1})(\zeta^{q^k} + \zeta^{-q^k}) = \zeta^{1+q^k} + \zeta^{-1-q^k} + \zeta^{1-q^k} + \zeta^{-1+q^k} = \alpha^{q^s} + \alpha^{q^t},$$

значит, соответствующая этому базису матрица T при $q = 2$ содержит $2n - 1$ единицу, потому что при $k = 0$ разложение $\alpha\alpha^{q^0}$ по базису состоит из одного слагаемого, ведь $\alpha^2 = (\zeta + \zeta^{-1})^2 = \zeta^2 + \zeta^{-2} = \alpha^{q^s}$.

Для выполнения указанной проверки заметим, что согласно определению последовательность $q^k \mod p, k = 0, ..., 2n-1$ является перестановкой $\pi(1), ..., \pi(2n)$ множества чисел $\{1, ..., 2n\}$ в силу того, что q-примитивный корень по модулю $p = 2n + 1$, причем в силу соотношения

$$q^{k+n} \equiv -q^k \mod p, k = 0, ..., n-1,$$

справедливо равенство

$$\pi(k) + \pi(k+n) = p, k = 0, ..., n-1,$$

а так как все неупорядоченные пары

$$(1 + q^k \mod p, -1 - p^k \mod p)$$

при любом $k, k = 0, ...n-1$, отличны от пары $(0, 0)$, то их последовательность состоит из пар

$$\pi(k), \pi(k+n)), k = 0, ..., n-1,$$

т.е. из неупорядоченных пар вида $(u, p-u), 0 < u \geq n$, значит существует такое отображение $(\sigma(1), ..., \sigma(n))$ множества чисел $1, ..., n$ в себя, что

$$\zeta^{1+q^k} + \zeta^{-1-q^k} = \zeta^{q^{\sigma(k)}} + \zeta^{-q^{\sigma(k)}} = \alpha^{q^{\sigma(k)}}.$$

Аналогично определяется отображение $\mu(1), ..., \mu(n)$ множества $1, ..., n$ в себя, такое, что

$$\zeta^{-1-q^k} + \zeta^{-1+q^k} = \zeta^{q^{\mu(k)}} + \zeta^{-q^{\mu(k)}} = \alpha^{q^{\mu(k)}}$$

при любом $n \geq k > 0$.

Так как

$$1 - q^k \neq 1 + q^k, 1 - q^k \neq -1 - q^k$$

по модулю p, то при любом $n > k > 0$ справедливо неравенство

$$\mu(k) \neq \sigma(k).$$

Вспоминая определение матрицы T посредством равенств

$$\alpha\alpha^{q^i} = \sum_{j=0}^{n-1} t_{i,j}\alpha^{q^j}$$

и сравнивая их с полученными равенствами

$$\alpha\alpha^{q^i} = \alpha^{q^{\sigma(i)}} + \alpha^{q^{\mu(i)}}, i > 0,$$

$$\alpha\alpha^{q^0} = \zeta^{1+q^0} + \zeta^{-1-q^0} + \zeta^{1-q^0} + \zeta^{-1+q^0} = \zeta^{1+q^0} + \zeta^{-1-q^0} + \zeta^0 + \zeta^0 =$$
$$= \zeta^{1+q^0} + \zeta^{-1-q^0} = \alpha^{q^{\sigma(1)}},$$

имеем

$$t_{i,j} = \delta_{\sigma(i+1),j} + \delta_{\mu(i),j}, i \neq 0, t_0, j = \delta_{\sigma(1),j},$$

где $\sigma_{k,s}$ - дельта-символ Кронекера, равный 1 при $k = s$, и нулю в противном случае.

4.1.2 Свойства матрицы T.

Для оптимизации вычислений при построении оптимального нормального базиса второго типа в поле $GF(2^n)$ докажем два утверждения:

Утверждение 4.1. *Для матрицы T оптимального нормального базиса второго типа в поле $GF(2^n)$ равенство $t_{i,j} = 1$ верно тогда и только тогда, когда выполняется одно из четырех соотношений*

$$2^i \pm 2^j \equiv \pm 1 \mod 2n + 1$$

Действительно, согласно определению матрицы T ее элемент $t_{i,j} = 1$, тогда и только тогда, когда в разложении $\alpha\alpha^{2^i}$. Но в силу равенства

$$\alpha\alpha^{2^i} = (\xi + \xi^{-1})(\xi^{2^i} + \xi^{-2^i}) = (\xi^{2^i - 1} + \xi^{-2^i + 1}) + (\xi^{2^i + 1} + \xi^{-2^i - 1})$$

и того факта, что $\xi^p = 1, p = 2n + 1$, так как ξ - первообразный корень p-й степени из единицы, это возможно, тогда и только тогда, когда $2^i \pm 1 \equiv \pm 2^j \mod 2n + 1$. Последняя эквивалентность равносильна $\pm 2^i \pm 2^j \equiv \pm 1 \mod 2n + 1$ и эквивалентности $2^i \pm 2^j \equiv \pm 1 \mod 2n + 1$.

Утверждение 4.2. *В случае оптимального нормального базиса второго типа в поле $GF(2^n)$ матрицы T и A совпадают.*

Действительно, так как $a_{i,j} = t_{i-j,-j}$, то согласно предыдущему утверждению $a_{i,j} = 1$, тогда и только тогда, когда $\pm 2^{i-j} \pm 2^{-j} \equiv \pm 1 \mod p$, где $i-j$ и $-j$ вычисляются по модулю n. Поэтому вместо $-j$ можно взять $n-j$, а вместо $i-j$ при $i-j < 0$ можно взять $n+i-j$. Умножая обе части эквивалентности на 2^j по модулю p, получаем равносильное соотношение $\pm 2^{i+n} \pm 2^n \equiv \pm 2^j \mod p$, которое в силу эквивалентности $2^n \equiv \pm 1 \mod p$ равносильно соотношению $\pm 2^{i+n} \pm 2^n \equiv \pm 2^j \mod p$. Эквивалентность $2^n \equiv \pm 1 \mod p$ вытекает из малой теоремы Ферма $2^{2n} \equiv 1 \mod p$.

Алгоритм 4 ONB2GEN

Вход: степень n расширения поля $GF(2)$.

Выход: вектор $b = (b_1, ..., b_n)$ коэффициентов неприводимого многочлена $b_0 + b_1 X + ... + b_{n-1} X^{n-1} + b_n X^n$ с корнем x - генератором ОНБ 2-го типа, таблица умножения A в ОНБ, порождаемом генератором x (в виде матрицы или списка коордат единичных элементов) или сообщение "ОНБ второго типа не существует".

Шаги:

1. Вычислить $p = 2n + 1$, проверить условия ОНБ второго типа:
   а) p - простое число;
   б) 2 - является примитивным корнем 1 по модулю p:
   для каждого простого делителя δ числа $p - 1$ выполняется $2^{(p-1)/\delta} \neq 1$
   Если хотя бы одно из этих условий не выполняется, то вернуть сообщение "ОНБ второго типа не существует".

2. Вычислить вектор b:
   для $j = 0, ..., n$: вычислить двоичные коды N и M чисел $\lfloor\frac{(n+j)}{2}\rfloor$ и j,
   принять $b_j = 1$, если $\neg(\neg M \vee N) = 0$ и 0 в остальных случаях.

3. Образовать нулевую $n \times n$ матрицу A (или образовать пустой список A).
   Заполнить ненулевые элементы:
   для $k = 0, ..., 2n$ вычислить $\pi(k) = 2^k \mod p$;
   для $k = 1, ..., 2n$ вычислить $\pi^{-1}(k) = dlog_2 k \mod p$;
   для $k = 0, ..., n - 1$
   вычислить $\sigma(k) = (\pi^{-1}(1 + 2^k) \mod p) \mod n = (dlog_2(1 + 2^k) \mod p) \mod n$;
   для $k = 1, ..., n - 1$
   вычислить $\mu(k) = (\pi^{-1}(-1 + 2^k) \mod p) \mod n = (dlog_2(-1 + 2^k) \mod p) \mod n$;
   для $k = 0, ..., n - 1$
   включить пару $(k, \sigma(k))$ в список A;
   для $k = 1, ..., n - 1$
   включить пару $(k, \mu(k))$ в список A;

4. Вернуть b и A.

4.2 Алгоритм генерации оптимальных нормальных базисов третьего типа

Базис третьего типа возникает, когда n нечетно, $2n + 1 = p$ - простое число, а условие на q заменяется на то, что $q^n \equiv 1 \mod p$, и при любом $0 < k < n, q^k \not\equiv 1 \mod p$ (другими словами, число q имеет по модулю p порядок n, а не $2n$, как во втором случае, и тогда автоматически существует такое r, что $q = r^2 \mod p$, т.е. q - квадратичный вычет по модулю p, и поэтому все его степени $q^k \mod p, k = 0, ..., n - 1$ образуют перестановку множества всех квадратичных вычетов по модулю p, так как их ровно n штук).

Поскольку p равно 3 по модулю 4, то -1 является квадратичным невычетом по модулю p, потому что в противном случае существовало бы такое число r, что $-1 \equiv r^2 \mod p$, и тогда получилось бы противоречие с малой теоремой Ферма:

$$r^{p-1} = (r^2)^{\frac{(p-1)}{2}} \equiv -1 \mod p$$

.

Поэтому из того, что произведение вычета на невычет является невычетом, следует, что последовательность $-q^k \mod p, k = 0, ..., n-1$ образует перестановку множества всех квадратичных невычетов по модулю p.

Как и в случае базиса второго типа, в качестве элемента ζ берется примитивный корень степени p из 1 в поле $GF(q^{2n})$, а в качестве порождающего элемента оптимального нормального базиса - элемент $\alpha = \zeta + \zeta^{-1}$.

4.2.1 Доказательство оптимальности.

Доказательство линейной независимости системы $\{\alpha, \alpha^q, \alpha^{q^2}, ..., \alpha^{q^{n-1}}\}$ отличается от случая базиса второго типа. Допустим противное, т.е.

$$\sum_{j=0}^{n-1} a_j \alpha^{q^j} = 0$$

для некоторого ненулевого вектора (α_j) с координатами из поля $GF(q)$. Подставляя равенства

$$\alpha^{q^i} = \zeta^{q^i} + \zeta^{-q^i},$$

получаем, что

$$\sum_{j=0}^{n-1} a_j(\zeta^{q^i} + \zeta^{-q^i}) = 0$$

Последнее равенство можно переписать в виде

$$\sum_{i=0}^{2n-1} b_i \zeta^i = 0$$

, если определить b_i как a_j, где $i = \pm q^j \mod p$ (как было указано выше, последнее условие определяет число j однозначно; знак плюс соответствует случаю, когда число i - вычет, а знак минус - случаю, когда i является квадратичным невычетом по модулю p), значит многочлен

$$f(X) = \sum_{i=0}^{2n-1} b_i X^i$$

имеет корни ζ и ζ^{-1} в поле $GF(q^{2n})$, а так как эти элементы имеют минимальные многочлены степени n каждый, причем они взаимно просты (в указанном поле корнями первого из них являются $\zeta^{q^i}, i = 0, ..., n-1$, а корнями второго - элементы $\zeta^{-q^i}, i = 0, ..., n-1$), то многочлен $f(X)$ над полем $GF(q)$ должен делиться на их произведение, что невозможно, так как его степень меньше $2n$.

Проверка оптимальности построенного базиса почти ничем не отличается от второго случая. Достаточно проверить, что при $k < n$

$$1 + q^k = \pm q^{\sigma(k+1)} \mod p$$

и

$$1 - q^k = \pm q^{\mu(k)} \mod p$$

(в последнем случае только при $k > 0$), тогда

$$\zeta^{1+q^k} + \zeta^{-1-q^k} = \zeta^{q^{\sigma(k+1)}} + \zeta^{-q^{\sigma(k+1)}} = \alpha^{q^{\sigma(k+1)}},$$

$$\zeta^{1-q^k} + \zeta^{-1+q^k} = \zeta^{q^{\mu(k)}} + \zeta^{-q^{\mu(k)}} = \alpha^{q^{\mu(k)}}, k > 0.$$

Так же, как и в случае второго типа, при $n > k > 0$ имеем

$$\alpha\alpha^{q^k} = (\zeta + \zeta^{-1})(\zeta^{q^k} + \zeta^{-q^k}) = \zeta^{1+q^k} + \zeta^{-1-q^k} + \zeta^{1-q^k} + \zeta^{-1+q^k} = \\ = \alpha^{q^{\sigma(k+1)}} + \alpha^{q^{\mu(k)}},$$

а при $k = 0$

$$\alpha\alpha^{q^0} = \zeta^{1+q^0} + \zeta^{-1-q^0} + \zeta^{1-q^0} + \zeta^{-1+q^0} = \zeta^{1+q^0} + \zeta^{-1-q^0} + \zeta^0 + \zeta^0 = \\ = \zeta^{1+q^0} + \zeta^{-1-q^0} = \alpha^{q^{\sigma(1)}}.$$

4.2.2 Вычисление матрицы T.

Как и в случае второго типа, выписывая равенства

$$\alpha\alpha^{q^i} = \sum_{j=0}^{n-1} t_{i,j}\alpha^{q^j}$$

и сравнивая их с полученными равенствами

$$\alpha\alpha^{q^i} = \alpha^{q^{\sigma(i+1)}} + \alpha^{q^{\mu(i)}}, i > 0, \alpha\alpha^{q^0} = \alpha^{q^{\sigma(1)}},$$

имеем

$$t_{i,j} = \delta_{\sigma(t+1),j} + \delta_{\mu(i),j}, i \neq 0, t_{0,j} = \delta_{\sigma(1),j},$$

где $\delta_{k,s}$ - дельта-символ Кронекера.

Отметим, что в случае базиса третьего типа матрицы T и A идентичны.

Алгоритм 5 ONB3GEN

Вход: степень n расширения поля $GF(2)$.

Выход: вектор $b = (b_1, ..., b_n)$ коэффициентов многочлена $b_0 + b_1X + ... + b_{n-1}X^{n-1} + b_nX^n$ с корнем x - генератором ОНБ 3-го типа, таблица умножения A в ОНБ, порождаемом генератором x (в виде матрицы или списка коордат единичных элементов) или сообщение "ОНБ третьего типа не существует".

Шаги:

1. Вычислить $p = 2n + 1$, проверить условия ОНБ третьего типа:
   а) p - простое число;
   б) $2^n \equiv 1 \mod p$;
   в) для каждого простого делителя δ числа n выполняется $2^{\frac{n}{\delta}} \mod p \neq 1$.
   Если хотя бы одно из этих условий не выполняется, то вернуть сообщение "ОНБ третьего типа не существует".

2. Вычислить вектор b:
   для $j = 0, ..., n$: вычислить двоичные коды N и M чисел $\lfloor \frac{(n+j)}{2} \rfloor$ и j,
   принять $b_j = 1$, если $\neg(\neg M \vee N) = 0$ и 0 в остальных случаях.

3. Образовать нулевую $n \times n$ матрицу A (или образовать пустой список A).
   Заполнить ненулевые элементы:
   для $k = 0, ..., 2n$ вычислить $\pi(k) = 2^k \mod p$;
   для $k = 1, ..., 2n$ вычислить $\pi^{-1}(k) = dlog_2 k \mod p$;
   для $k = 0, ..., n - 1$
   вычислить $\sigma(k) = (\pi^{-1}(\pm(1+2^k)) \mod p) \mod n = (dlog_2(\pm(1+2^k)) \mod p) \mod n$;
   для $k = 1, ..., n - 1$
   вычислить $\mu(k) = (\pi^{-1}(\pm(-1 + 2^k)) \mod p) \mod n = (dlog_2(\pm(-1 + 2^k)) \mod p) \mod n$;
   для $k = 0, ..., n - 1$
   включить пару $(k, \sigma(k))$ в список A;
   для $k = 1, ..., n - 1$
   включить пару $(k, \mu(k))$ в список A;

4. Вернуть b и A.

5 Оптимизация преобразований базисов

5.1 О комбинированном использовании полиномиального и нормального базисов

Программные реализации умножения в нормальных базисах оказались медленнее алгоритмов умножения в стандартных базисах даже в полях небольших размерностей. Сравнение двух типов базисов, стандартного и нормального, наводит на мысль об ускорении арифметики в конечных полях за счет использования выгодных сторон каждого из них. Действительно, умножение быстрее производить в стандартном представлении поля $GF(2^n)$, а возведение в степень - в нормальном представлении.

Для реализации этой идеи понадобятся матрицы перехода от нормального базиса к стандартному и обратно, которые могут оказаться не разряженными, а плотными, и тогда сложность перехода от одного базиса к другому в худшем случае будет $O(n^2/logn)$(если использовать для умножения матрицы на вектор метод "четырех русских").

Но в удачном случае сложность перехода может оказаться даже не выше линейной, например в случае, если число ненулевых элементов в матрицах переходов будет $O(n)$. Таким образом, возникает задача поиска нормальных базисов с "простыми"матрицами переходов к стандартным базисам и обратно.

Эта задача легко решается в случае оптимальных нормальных базисов первого типа.

Теорема 5.1. *Переход от стандартного базиса поля $GF(q^n)$ к соответствующему(с тем же генератором) оптимальному нормальному базису первого типа(если, конечно, он существует для данного n) и обратно можно выполнить с линейной сложностью.*

Доказательство. Легко видеть, что в этом случае базис $\{\zeta, ..., \zeta^n\}$(не совсем стандартный) совпадает с точностью до перестановки с оптимальным нормальным базисом

$$\{\zeta, \zeta^q, \zeta^{q^2}, \zeta^{q^{n-1}}\},$$

так как последовательность чисел $1, q, q^2, ..., q^{n-1}$, вычисленных по модулю p, совпадает с некоторой перестановкой

$$\{1, 2, ..., n\}$$

в силу того, что ζ есть примитивный корень $p-$ степени из единицы в поле $GF(q^n)$, а q - примитивный корень p-й степени из единицы в поле $GF(q^n)$, а q - примитивный корень по модулю p.

Поэтому, очевидно, переход от базиса $\{\zeta, ..., \zeta^n\}$ к нормальному базису

$$\{\zeta, \zeta^q, \zeta^{q^2}, \zeta^{q^{n-1}}\}$$

и обратно выполняется с оценкой сложности не более чем n.

Отметим попутно, что ζ является корнем неприводимого над полем $GF(q)$ многочлена

$$1 + X + ... + X^n = \frac{X^p - 1}{X - 1}$$

Стандартным базисом для перехода будет базис $\{1, \zeta, ..., \zeta^{n-1}\}$. Очевидно, он с линейной сложностью выражается через базис $\{\zeta, ..., \zeta^n\}$, поскольку

$$\zeta^n = 1 + \zeta + ... + \zeta^{n-1},$$

и обратно, базис $\{\zeta, ..., \zeta^n\}$, благодаря формуле

$$1 = \zeta + ... + \zeta^{n-1} + \zeta^n,$$

с линейной сложностью выражается через базис $\{1, \zeta, ..., \zeta^{n-1}\}$. В результате имеем, что переход от оптималього нормального базиса

$$\{\zeta, \zeta^q, \zeta^{q^2}, ..., \zeta^{q^{n-1}}\}$$

к стандартному базису $\{1, \zeta, ..., \zeta^{n-1}\}$ и обратно выполняется с линейной сложностью. $\square$

5.2 Оценка сложности перехода от оптимальных нормальных базисов второго и третьего типа к стандартным и обратно

Вначале докажем несколько вспомогательных утверждений (следующая лемма выполняется и для общего случая). Пусть $x = \alpha_1$ - генератор оптимального нормального базиса

$$\{\alpha_1, ..., \alpha_n\}$$

второго типа поля $GF(q^n)$, он же генератор соответствующего стандартного базиса $\{x^0, ..., x^{n-1}\}$, т.е. $\alpha_1 = x^{q^{k-1}} = \zeta^{q^{k-1}} + \zeta^{-q^{k-1}}$, и при любом $k \leq n$

$$\alpha_k = x^{q^{k-1}} = \zeta^{q^{k-1}} + \zeta^{-q^{k-1}},$$

где ζ - примитивный корень из 1 степени $p = 2n + 1$ в поле $GF(q^{2n})$.

Рассмотрим вспомогательную последовательность $a_0, a_1, ...$, порожденную элементом x. Положим $a_0 = 2$ (в случае $p = 2$, $a_0 = 0$) и, далее, $a_k = \zeta^k + \zeta^{-k}, k = 1, 2, ..., a_1 = \alpha_1$.

Элементы $a_1, a_2, ..., a_n$ этой последовательности образуют почти нормальный базис, являющийся перестановкой элментов нормального базиса.

Лемма 5.1. *Для любого $k \geq 1$ имеют место следующие рекуррентные соотношения:*
1. $a_{k+1} = a_k a_1 - a_{k-1}$
2. $a_{p^k} = x^{p^k}$.
3. $a_{p^k+i} = a_i x^{p^k} - a_{p^k-i}$.

Доказательство. Для доказательства первой формулы заметим, что

$$(\zeta^k + \zeta^{-k})(\zeta + \zeta^{-1}) = \zeta^{k-1} + \zeta^{-k+1} + \zeta^{k+1} + \zeta^{-k-1},$$

т.е. $a_k a_1 - a_{k-1} = a_{k+1}$.

Вторая формула очевидна.

Третья формула проверяется непосредственно:

$$a_{p^k+i} = \zeta^{p^k+i} + \zeta^{-p^k-i} = (\zeta^{p^k} + \zeta^{-p^k})(\zeta^i + \zeta^- i) - (\zeta^{p^k-i} + \zeta^{-p^k+i}) =$$
$$= a_{p^k} a_i - a_{p^k-i} = x^{p^k} a_i - a_{p^k-i}.$$

□

Рекуррентные соотношения, доказанные в лемме, позволяют записать формулы перехода от почти стандартного базиса $\{x^1, ..., x^n\}$ к почти нормальному базису.

В случае $p = 2$ вычитание в лемме можно заменить на сложение.

Из леммы следует, что для любого $i \geq 1$ элемент a_i почти нормального базиса α_i выражается в виде значения некоторого многочлена степени i над полем $GF(q)$, т.е.

$$a_i = f_i(x) = \sum_{j=1}^{i} f_{i,j} x^j.$$

Выразив таким образом все элементы a_i почти нормального базиса в почти стандартном базисе, получим матрицу перехода $F_n = (f_{i,j})$ от почти стандартного базиса к почти нормальному. В случае, когда $p = 2$ плотность $S(F_n)$(количество ненулевых элементов) матрицы F_n оценивается следующим образом:

Теорема 5.2. *Плотность матрицы перехода от почти стандартного базиса поля $GF(2^n)$ к почти нормальному базису второго или третьего типа равна $O(n^{\log_2 3})$.*

Из этой теоремы видно, что матрица перехода является разряженной, и это позволяет осуществлять быстрое умножение в этом базисе. Приведем доказательство этой теоремы.

Доказательство. Согласно формулам предыдущей леммы (5.1), матрица F_{2n+1}, построенная с их помощью, имеет при $n = 2^k - 1$ вид:

$$F_{2n+1} = \begin{pmatrix} F_n & o_n & O_n \\ 0...0 & 1 & 0...0 \\ G_n & o_n & F_n \end{pmatrix} \tag{5.1}$$

где o_n - нулевой вектор-столбец высоты n, O_n - нулевая $n \times n$ матрица, матрица G_n есть симметрическое отражение матрицы F_n относительно средней строки, т.е. $G_n = I_n F_n$, где $I_n = (\delta_{i,n-i+1})$ - матрица с единицами на побочной диагонали и нулями в остальных местах. Например, матрица F_7 выглядит так:

$$F_7 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \tag{5.2}$$

Если разбить матрицу на 4 квадратных подматрицы, удалив среднюю строку и средний столбец, получим, что левый верхний квадрат 3×3 симметричен относительно горизонтали левому нижнему и равен нижнему правому квадрату. Средняя строка и средний столбец содержат ровно одну единицу, лежащую на их пересечении. Матрица является нижнетреугольной с единицами на главной диагонали. Действительно, в общем случае согласно лемме (5.1), при $0 \le i \le 2^k$

$$\sum_{j=1}^{2^k+i} f_{2^k+i,j} x^j = a_{2^k+i} = a_i x^{2^k} + a_{2^k-i} = \sum_{j=1}^{i} f_{i,j} x^{2^k+j} + \sum_{j=1}^{2^k-i} f_{2^k-i,j} x^j,$$

откуда имеем $f_{2^k+i,j} = f_{2^k-i,j}$, при $0 \le j \le 2^k$, и $f_{2^k+i,2^k+j} = f_{i,j}$ при $1 \le j \le 2^k$.

Опираясь на это представление, можно получить следующую рекуррентную формулу для вычисления плотности последовательности матриц F_n

$$S(F_{2n+1}) = 3S(F_n) + 1, n \ge 3, S(F_3) = 4,$$

из которой вытекает рекуррентная формула

$$S(F_{2^k-1}) = 3S(F_{2^{k-1}-1}) + 1, k \ge 2, S(F_3) = 4.$$

Полагая $l(k) = S(F_{2^k-1})$, перейдем к линейному рекуррентному соотношению $l(k) = 3l(k-1)+1, l(2) = 4$, решением которого будет $l(k) = (3^k-1)/2$. Обозначив $n = 2^k - 1$ и выразив $3^k = (2^k)^{\log_2 3} = (n+1)^{\log_2 3}$, получим, что

$$S(F_n) = l(k) = \frac{(n+1)^{\log_2 3} - 1}{2} = O(n^{log_2 3})$$

В общем случае равенство $S(F_n) = O(n^{\log_2 3})$ сохраняется, так как выбрав k таким образом, что $2^k - 1 < n \leq 2^{k+1} - 1$, можно заметить, что матрица F_n является главной подматрицей матрицы F_m, $m = 2^k - 1$, откуда имеем

$$S(F_n) \leq S(F_m) = O(m^{log_2 3}) = O(n^{\log_2 3}).$$

Аналогично можно оценить плотность матрицы $F'_n = F_n \cdot B_n$, и непосредственно проверяется, что в матрице B_n все элементы нулевые, кроме элементов $b_{i,i+1} = 1$ над диагональю и некоторых элементов нижней строки $b_{n,j}$ и справедливы следующие равенства для элементов матрицы F'_n :

$$f'_{i,1} = f_{i,n}b_{n,1} = \delta_{i,n}b_{n,1}, f'_{i,j} = f_{i,j-1} + f_{i,n}b_{n,j} = f_{i,j-1} + \delta_{i,n}b_{n,j},$$

так как в силу того, что матрица F_n нижнетреугольная, для ее элементов имеем $f_{i,n} = \delta_{i,n}$, где $\delta_{i,n}$ - дельта-символ Кронекера. Складывая эти равенства, получим, что

$$S(F'_n) = \sum_{i=1,j=1}^{n,n-1} f_{i,j} + \sum_{j=1}^{n} b_{n,j} = S(F_n) - 1 + \sum_{j=1}^{n} b_{n,j} \leq S(F_n) - 1 + n = O(n^{\log_2 3}).$$

□

Обозначим через $L(F_n)$ сложность линейного преобразования, определяемого матрицей F_n (в данном случае - это наименьшее число операций сложения по модулю два, необходимых для вычисления этого преобразования).

Теорема 5.3. $L(F_n) \leq \frac{n}{2}\log_2 n + 2n = O(n \log_2 n)$

Доказательство. Нам будет удобно оценивать сложность преобразования, задаваемого транспонированной матрицей F_n^T, которая равна $L(F_n)$ согласно известной лемме о взаимосвязи сложности транспонированных матриц (см. например [3]). Впрочем, для дальнейшего нам нужно будет оценить сложность перехода от координат в нормальном базисе

$$\sum_{i=1}^{n} x_i \alpha_i$$

к координатам в почти стандартном базисе

$$\sum_{i=1}^{n} y_i x^i,$$

которая определяется как раз матрицей F_n^T. Пусть $2^k \le n \le 2^{k+1}$, тогда с использованием формул из леммы имеем

$$\sum_{i=1}^{n} y_i x^i = \sum_{i=1}^{n} x_i a_i = \sum_{i=1}^{2^k} x_i a_i + \sum_{i=2^k+1}^{n} x_i a_i =$$

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

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

где $a_0 = 0$. Далее, определяя векторы-столбцы

$$X = \{x_1, ..., x_n\}^T,$$

$$X_1 = \{x_1, ..., x_{2^{k+1}-n-1}, x_{2^{k+1}-n} + x_n, ..., x_{2^k-1} + x_{2^k+1}, x_{2^k}\}^T,$$

где, естественно, в случае $n = 2^{k+1} - 1$

$$X_1 = \{x_1 + x_n, ..., x_{2^k-1} + x_{2^k+1}, x_{2^k}\}^T,$$

а в случае $n = 2^{k+1}$

$$X_1 = \{x_1 + x_{n-1}, ..., x_{2^k-1} + x_{2^k+1}, x_{2^k}\}^T,$$

и во всех случаях

$$X_2 = \{x_{1+2^k,...,x_n}\}^T,$$

и вектора-строки

$$Y_1 = \{y_1, ..., y_{2^k}\}, Y_2 = \{y_{1+2^k}, ..., y_n\},$$

получим

$$\sum_{i=1}^{n} y_i x^i = \sum_{i=1}^{n} x_i a_i = \sum_{i=1}^{2^k} X_{1,i} a_i + x^{2^k} \sum_{i=1}^{n-2^k} X_{2,i} a_i =$$

$$= \sum_{i=1}^{2^k} Y_{1,i} a^i + \sum_{i=1}^{n-2^k} Y_{2,i+2^k} a_{i+2^k},$$

значит

$$F_n^T \otimes X = (y_i' ... y_n') = (Y_1, Y_2) = (F_{2^k}^T \otimes X_1, F_{n-2^k}^T \otimes X_2),$$

где $\otimes$ - операция умножения матрицы на вектор в поле $GF(2)$. Разумеется, последнее равенство можно было получить, основываясь только на структуре матрицы F_n. Осталось индуктивно оценить сложность преобразования координат, определяемого матрицей F_n^T. Согласно последнему равенству

$$L(2^{m+1}) \le 2^m - 1 + 2L(2^m), m \le k-1, L(2) = 2$$

По индукции непосредственно проверяется, что

$$L(2^m) = 2^{m-1}m + 1$$

Для произвольного n в пределах $2^k < n < 2^{k+1}$ получим

$$L(n) \le 2^{k_s-1}k_s + ... + 2^{k_1-1}k_1 + s - 1 + (n - 2^{k_s}) + ...$$
$$... + (n - 2^{k_s - ... - 2^{k_2}}) \le \frac{n}{2}\log_2 n + c\frac{n}{2},$$

где

$$c = 1\frac{2}{2} + \frac{3}{4} + \frac{4}{8} + \frac{5}{16} + ... < 4$$

□

Теорема 5.4. *Сложность $B(n)$ перехода в поле $GF(2^n)$ от оптимального нормального базиса второго или третьего типа к соответсвтующему стандартному и наоборот удовлетворяет неравенству*

$$B(n) \le \frac{n}{2}\log_2 n + 3n$$

Доказательство. Обозначим $x = \zeta + \zeta^{-1}$ генератор стандартного базиса

$$A = \{1, x^1, ..., x^{n-1}\}$$

и базиса второго или третьего типа

$$B = \{x^{2^0}, x^{2^1}, ..., x^{2^{n-1}}\}.$$

Переход от нормального базиса к стандартному. Переход будет осуществляться через цепочку из четырех базисов

$$B \to B' \to A' \to A$$

и преобразования координат элемента f поля $GF(2^n)$ в этих базисах

$$f = \sum_{i=1}^{n} x_i x^{2^{i-1}} = \sum_{i=1}^{n} x_i' a_i = \sum_{i=1}^{n} y_i' x^i = \sum_{i=1}^{n} y_i x^{i-1},$$

где a_i обозначает $\zeta^i + \zeta^{-i}, B' = \{a_1, ..., a_n\}, A' = \{x^1, ..., x^n\}$.

Переход от базиса B к базису B' и преобразование координат $\tilde{x}$ к координатам $\tilde{x}'$. Этот переход осуществляется перестановкой базисных элементов, и действительно, существует такая перестановка $\pi(i)$ чисел $\{1, ..., n\}$, что для любого $i = 1, ..., 2n$ выполняется равенство

$$2^i \mod p = \pm\pi(i) \in \{1, ..., n\}. \tag{5.3}$$

В самом деле, для базиса второго типа последовательность степеней

$$1, 2, 2^2, ..., 2^{2n-1},$$

вычисленных по модулю p, совпадает с некоторой перестановкой

$$\pi(1), ..., \pi(2n)$$

множества чисел $\{1, ..., 2n\}$ в силу того, что 2 - примитивный корень по модулю p. А в силу эквивалентности

$$2^{k+n} \equiv -2^k \mod p,$$

(по теореме Ферма $2^{2n} \equiv 1 \mod p$, значит, $2^n \equiv -1 \mod p$) окончательно получим формулу (5.3)

В случае же базиса третьего типа число 2 является квадратичным вычетом по модулю p, поэтому все степени $2^k \mod p, k = 1, ..., n-1$ образуют перестановку множества всех квадратичных вычетов по модулю p, так как их ровно n штук. Добавив к этому тот факт, что p равно 3 по модулю 4, и поэтому -1 является квадратичным невычетом по модулю p, так как в противном случае существовало бы такое число r, что $-1 \equiv r^2 \mod p$, а это приводило к противоречию с теоремой Ферма:

$$r^{p-1} = (r^2)^{\frac{p-1}{2}} \equiv -1 \mod p,$$

и тот факт, что произведение вычета на невычет является невычетом, получаем, что последовательность $-2^k \mod p, k = 0, ..., n-1$ образует перестановку множества всех квадратичных невычетов по модулю p, в итоге получаем, что и в этом случае верна формула (5.3) □

6 Реализация

В полях с оптимальным базисом 2 типа были реализованы следующие арифметические операции: генерация базиса, генерация таблицы умножения, умножение, поиск обратного элемента, возведение в квадрат, деление. Имплементированные алгоритмы были протестированы на процессоре Intel Core i5-3230M 2.6 GHz.

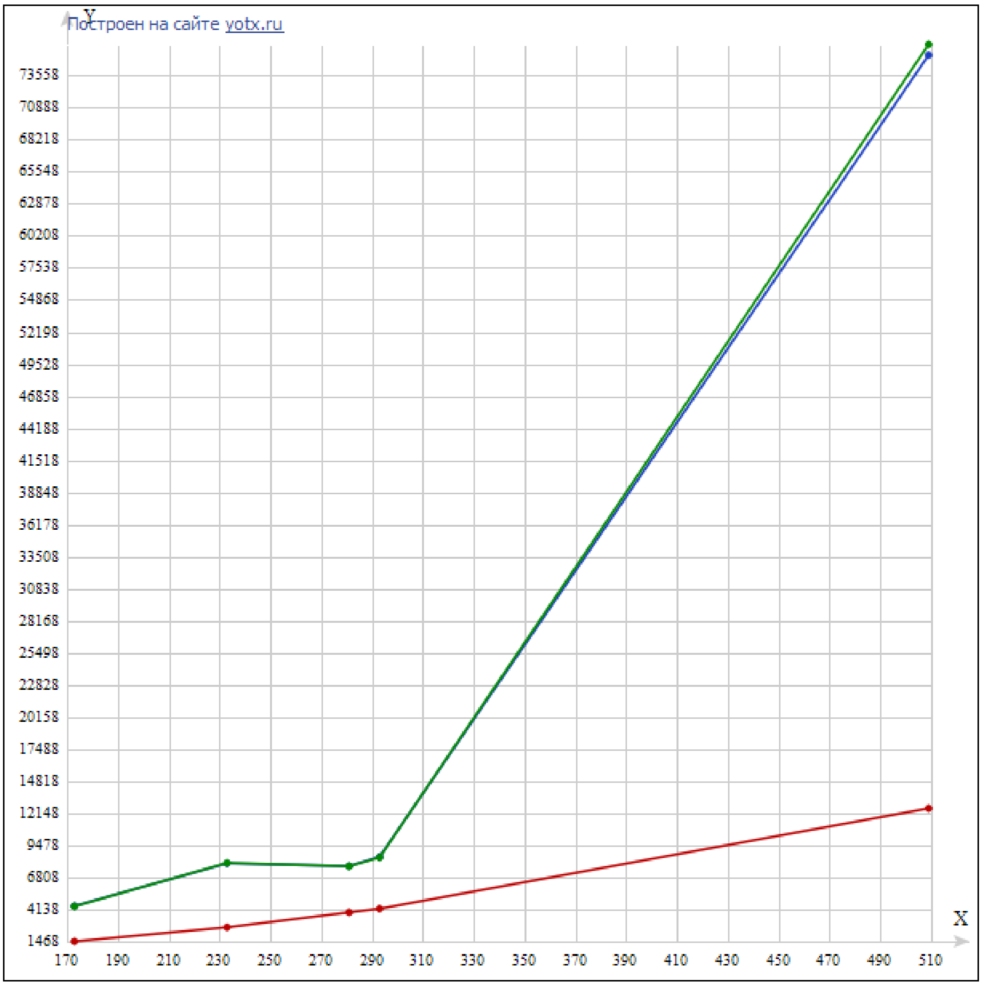
Был получен следующий результат:

Таблица 1 — Сравнение скоростей выполнения арифметических операций в оптимальном нормальном базисе

Алгоритм	m=173	m=233	m=281	m=293	m=509
Генерация таблицы умножения	4415	7985	7731	8494	75243
Умножение	1468	2641	3880	4187	12536
Обращение	4383	7990	7715	8444	76134

В ячейках таблицы указано среднее число тысяч тиков процессора для выполнения соответствующей операции в поле. Как видно из таблицы, самыми дорогими операциями являются генерация таблицы умножения и обращение элемента. Так как деление использует обращение делителя, то оно также имеет такую же сложность. Так как таблицу умножения для каждого поля можно сгенерировать в самом начале, то ее сложностью можно пренебречь. Можно заметить, что на приведенной выборке из степеней расширения умножение примерно в 4 раза быстрее обращения.

Также можно отметить, что умножение быстрее выполняется в стандартном базисе, а возведение в степень, кратную q быстрее в нормальном базисе. Поэтому можно сказать, что можно ускорить вычисления за счет использования преимуществ обоих базисов. Для этого нужно уметь быстро выполнять переход от одного базиса к другому. Можно показать, что это делается за линейное время. Реализация этой идеи будет продолжением этой работы.



На рисунке обращение изображено зеленым цветом, генерация таблицы - синим, умножение - красным.

6.1 Множество кандидатов

Будем решать задачу генерации простого числа заданной длины. Для этого выберем случайное $n \xleftarrow{R} [2^{\log_2 n}, 2^{\log_2 n+1} - 1]$. Также у нас есть некоторое фактор-множество $F = \{x_i \mid x_i \in PRIMES\}$, которое состоит из последовательных малых простых чисел, исключая число 2. Составим систему остатков $R_n = \{n_i = n \ (mod \ x_i) \mid x_i \in F\}$. Если среди них есть хотя бы один ноль, то число n делится на соответствующее малое простое, следовательно n является составным. Заметим, что можно просто построить систему R_{n+2}, зная заранее R_n. Нужно просто к остаткам прибавить двойку и, если нужно, привести по модулю. Таким образом можно просто построить любую другую систему вида R_{n+k}, причем имеет смысл k выбирать всегда четным, иначе $n + k$ получится четным, следовательно составным. Метод состоит в том, чтобы строить R_n, и в случае того, если R_n содержит ноль, то строить следующе R_{n+2}, и так далее, пока не найдем такой R_{n+2i}, что он не содержит нуля, т.е. взаимно прост со всеми числами фактор-множества. Недостаток метода в том, что при большой цепочке подряд идущих составных числах, может оказаться, что i, при этом необходимо на каждой итерации заново пересчитывать все суммы.

В связи с этим придумана оптимизация этого метода. Рассмотрим конкретное число из факторной базы $p = 5$, и $n = 29$. $n \mod p = 29 \mod 5 = 4$. Чтобы получить число, делящееся на 5, к 29 необходимо добавить минимальное четное число 6, а потом добавлять всегда по 10. Таким образом после постройки $R_{n=29}$ не нужно строить R_{n+6}, R_{n+16}, и т.д. Такие числа как $6, 16, 26, ...$ для $n = 29$ будем называть запрещенными. Таким образом можно построить множество D, состоящее из запрещенных чисел для n и всех чисел из факторной базы. Таким образом, если в множестве D не будет содержаться, например число 4, то R_{n+4} будет взаимно просто со всеми числами факторной базы. При реализации этого метода ограничим число i некоторым порогом (например максимальным числом из факторной базы $+ 1$), чтобы иметь гарантированно константное использование памяти. При этом необходимо предварительно посчитать $|F|$ раз остаток от деления n на каждое простое число. При составлении множества запрещенных чисел на каждой итерации будет добавляться примерно $\frac{|F|}{p_i}$ чисел. Таким образом, генерация множества запрещенных чисел имеет асимптотику

$$O(|F| \sum_{p \in F} \frac{1}{p}) = O(|F| \ln \ln p_{max}),$$

где p_{max} - максимальное число из факторной базы. Если при этом вспомнить оценку числа простых чисел от 1 до p, то

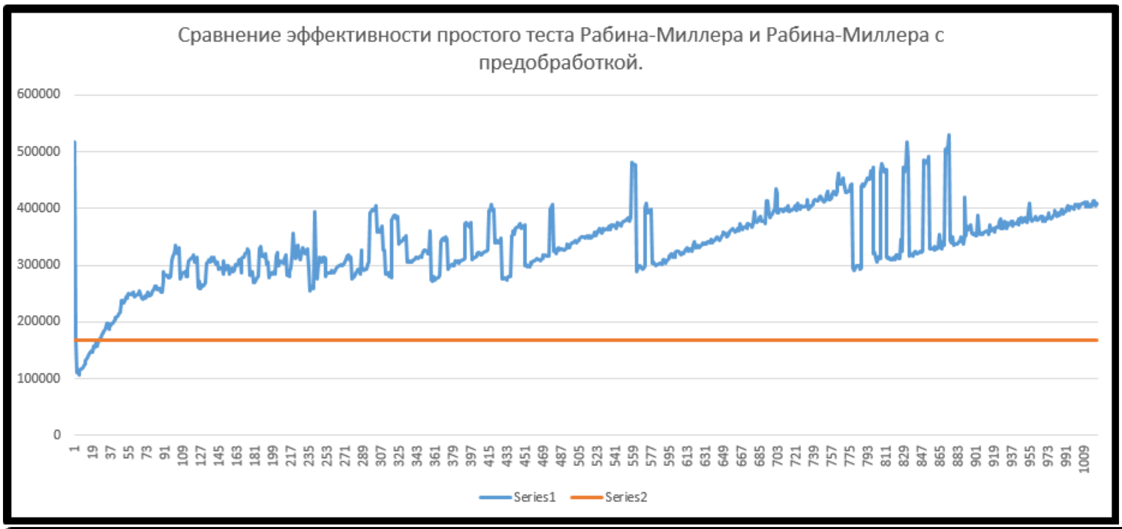
$$|F| \approx \frac{p_{max}}{\ln p_{max}}.$$

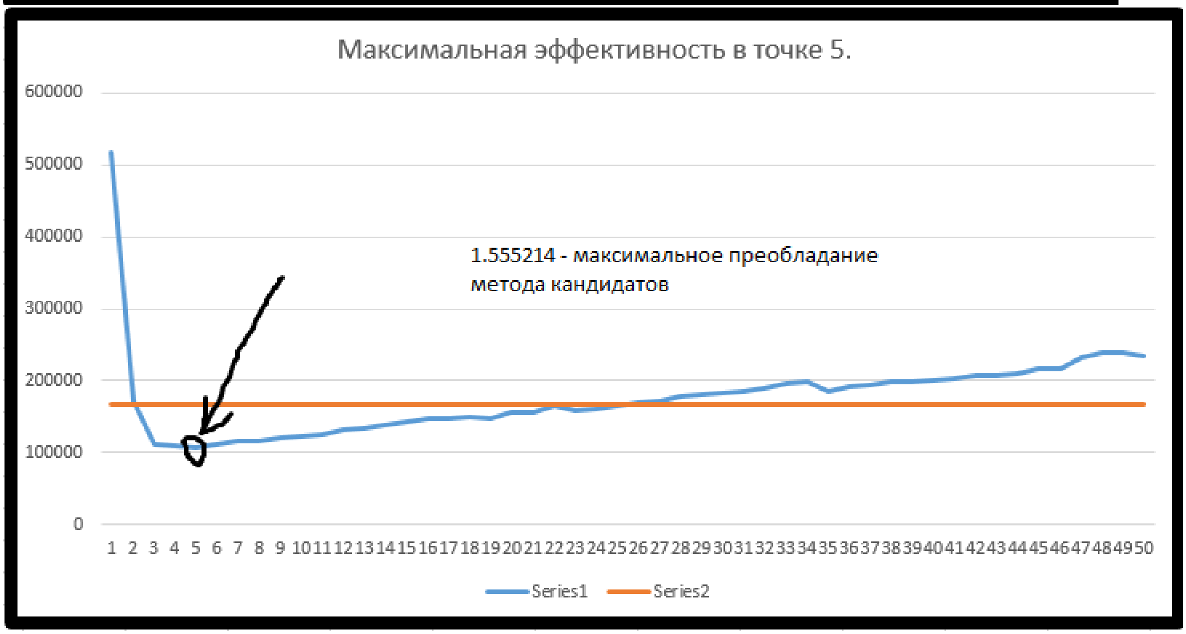
Следовательно генерация множества запрещенных чисел имеет асимптотику

$$O(\frac{p_{max} \ln \ln p_{max}}{\ln p_{max}}).$$

Этап построения R_n имеет асимптотику $O(|F| \log_2 n)$, так как выполняется $|F|$ операций приведения по малому модулю. Итого асиптотика равна

$$O(\frac{p_{max}(\ln \ln p_{max} + \log_2 n)}{\ln p_{max}}).$$





Таким образом можем заключить, что метод себя оправдывает при небольшой мощности фактор-множества. В частности, при генерации простых чисел длины не более 32 бита, удалось добится прироста производительности в 1.5 раза при мощности фактор-множества, равной шести (соответствует пятерке на графике). При этом в алгоритме Миллера-Рабина использованы базовые числа, которые позволяют точно проверять на простоту числа вплоть до 4759123141. Вполне возможно, что для генерации простых чисел большей длины понадобится фактор-множество большего размера, но это предмет следующих исследований.

Литература

1. Фролов А.Б. Болотов А.А., Гашков С.Б. *Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы.* КомКнига, 2006.

2. Фролов А.Б. Болотов А.А., Гашков С.Б. *Элементарное введение в эллиптическую криптографию. Протоколы криптографии на эллиптических кривых.* КомКнига, 2006.

3. Кочергин В.В Гашков С.Б. Об аддитивных цепочках векторов, вентильных схемах и сложности вычисления степеней. *Дискретная математика*, (52):22–40, 1992.

4. R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone, and R.M. Wilson. Optimal normal bases in gf(pn). *Discrete Applied Mathematics*, 22(2):149 – 161, 1988Ц1989.

5. H.W Lenstra Jr. S Gao. Optimal normal bases. *Designs, Codes and Cryptography*, 22(2):149 – 161, 1992.