Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Вычислительной математики и информационных технологий

Отчет

производственной практики

(технологическая (проектно-технологическая) практика)

Обучающийся Гусев Виталий Евгеньевич гр.09-335 \_\_\_\_\_\_\_

(ФИО студента) (Группа) (Подпись)

Научный руководитель

доцент КСАИТ Мубараков Б.Г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность, ученое звание) (ФИО) (подпись)

Руководитель практики от КФУ:

доцент кафедры САИТ Андрианова А.А.

Оценка за практику \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

Дата сдачи отчета «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2025 г.

Казань – 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1. Разработка модификаций методов дискретного логарифмирования 4

2. Модификация алгоритма Шенкса 5

3. Модификация алгоритма Полига-Хеллмана 7

4. Модификация алгоритма ро-метод Полларда 10

5. Модификация алгоритма Адлемана 12

6. Модификация алгоритма COS 14

7. Модификация решета числового поля 17

ЗАКЛЮЧЕНИЕ 21

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22

ПРИЛОЖЕНИЕ 24

ВВЕДЕНИЕ

Производственная практика (технологическая (проектно-технологическая) практика) проходила на кафедре системного анализа и информационных технологий Института вычислительной математики и информационных технологий КФУ с 01 ноября 2024 года по 08 февраля 2025 года.

Целью практики является исследование, разработка и тестирование модификаций алгоритмов дискретного логарифмирования с экспоненциальной и субэкспоненциальной сложностью.

В процессе практики будут исследованы, реализованы и протестированы модификации алгоритмов: алгоритм Шенкса, алгоритм Полига-Хеллмана, ро-метод Полларда, алгоритм Адлемана, алгоритм COS, решето числового поля. Для тестирования модификаций данных алгоритмов будет использован генератор параметров Диффи-Хеллмана и возведение в степень по модулю. Также для тестирования алгоритмов будет реализован замер времени выполнения алгоритма и количество потраченной памяти на выполнение алгоритма.

1. Разработка модификаций методов дискретного логарифмирования

В процессе практики были изучены и реализованы модификации методов дискретного логарифмирования на языке программирования C# на .NET8 в Windows Forms (рисунок 1). Для тестирования данных алгоритмов был реализован генератор параметров Диффи-Хеллмана и возведение числа в степень по модулю [1]. Также для тестирования данных алгоритмов был реализован замер времени выполнения алгоритма и количество потраченной памяти на выполнение алгоритма.

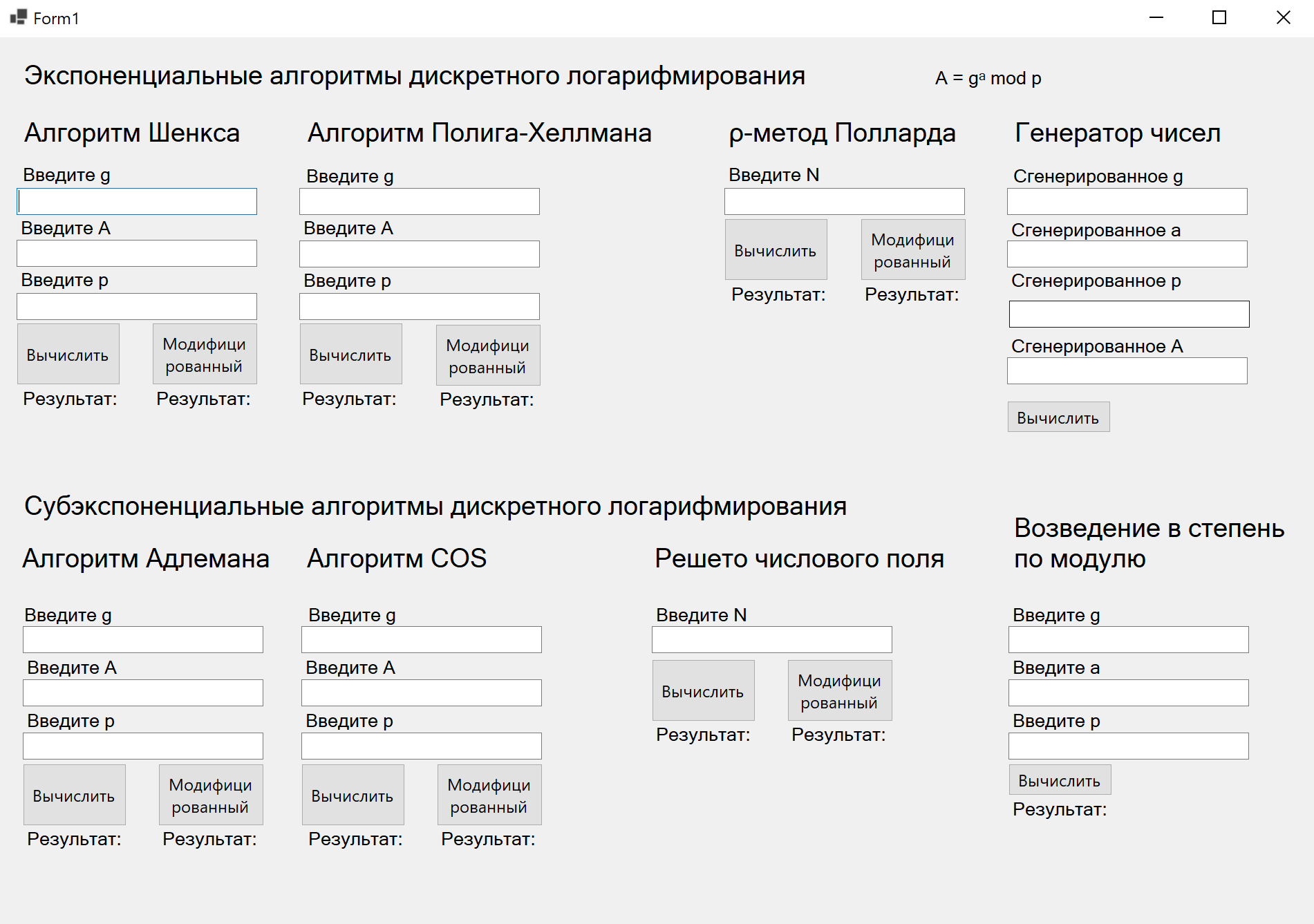


Рисунок 1 - Реализованная программа

Были реализованы модификации экспоненциальных алгоритмов дискретного логарифмирования: алгоритм Шенкса [2], алгоритм Полига-Хеллмана [3], ро-метод Полларда [4], а также модификации субэкспоненциальных алгоритмов дискретного логарифмирования: алгоритм Адлемана [5], алгоритм COS [6], решето числового поля [7].

Разработанная программа позволяет вносить в текстовые поля необходимые значения параметров возведения чисел в степень по модулю: g, a, p, A [8, 9], либо целых чисел N для разложения на простые множители [10] и выводить результат вычисления. В процессе практики были добавлены кнопки для вычисления модификаций, а также результаты модифицированных алгоритмов, совместно с временем вычисления алгоритма и затраченной памятью вычисления алгоритма (рисунок 2).

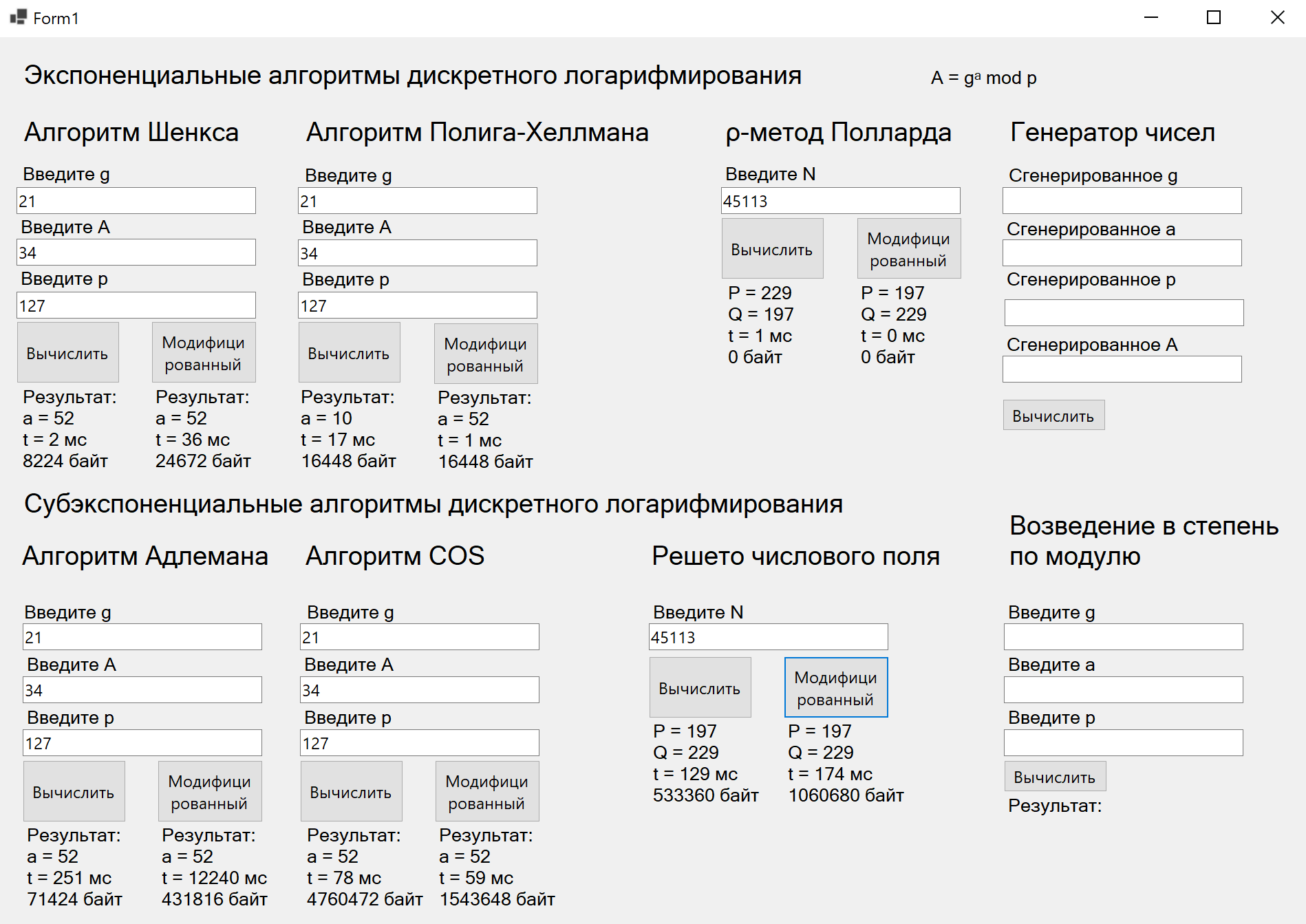


Рисунок 2 - Вычисление модифицированных алгоритмов

2. Модификация алгоритма Шенкса

Была реализована модификация экспоненциального алгоритма «Шаг младенца — шаг великана» — в теории групп детерминированный алгоритм дискретного логарифмирования в мультипликативной группе кольца вычетов по модулю простого числа. Начальный алгоритм был предложен советским математиком Александром Гельфондом в 1962 году и Дэниелом Шенксом в 1972 году. Метод теоретически упрощает решение задачи дискретного логарифмирования, на вычислительной сложности которой построены многие криптосистемы с открытым ключом. Относится к методам встречи посередине. Это был один из первых методов, который показал, что задача вычисления дискретного логарифма может быть решена значительно быстрее, чем методом перебора. Идея алгоритма состоит в выборе оптимального соотношения времени и памяти, а именно в усовершенствованном поиске показателя степени.

Пусть задано сравнение , необходимо найти натуральное число , удовлетворяющее данному сравнению.

Начальный алгоритм реализован следующим образом:

1) Сначала берутся два целых числа и , такие, что . Как правило .

2) Вычисляются два ряда чисел:

,

.

Все вычисления проводятся по модулю .

3) Идёт поиск таких и , для которых выполняется равенство . То есть ищется во втором ряду такое число, которое присутствует и в первом ряду. Запоминаются показатели степени и , при которых данные числа получались.

4) В результате работы алгоритма неизвестная степень вычисляется по формуле .

Была реализована модификация алгоритма, состоящая в распараллеливании 2 и 3 шага алгоритма. На 2 шаге алгоритма параллельно вычисляются два ряда чисел. На 3 шаге был сделан параллельный поиск результата с начала и с конца ряда. Данная модификация предположительно должна увеличить скорость работу с рядами чисел.

Для проверки работы модифицированного алгоритма были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 81 мс и затратил памяти 3947976 байт, а модифицированный за 120 мс и затратил 2987840 байт (рисунок 3).

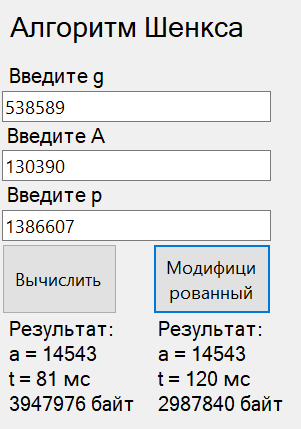


Рисунок 3 - Результат работы модифицированного алгоритма Шенкса

3. Модификация алгоритма Полига-Хеллмана

Был реализован модифицированный детерминированный алгоритм дискретного логарифмирования с экспоненциальной сложностью Полига-Хеллмана в кольце вычетов по модулю простого числа. Одной из особенностей алгоритма является то, что для простых чисел специального вида можно находить дискретный логарифм за полиномиальное время. Данный алгоритм был придуман американским математиком Роландом Сильвером, но впервые был опубликован другими двумя американскими математиками Стивеном Полигом и Мартином Хеллманом в 1978 году в статье «An improved algorithm for computing logarithms over GF(p) and its cryptographic significance», которые независимо от Роланда Сильвера разработали данный алгоритм.

Пусть задано сравнение , необходимо найти натуральное число , удовлетворяющее данному сравнению.

Шаги выполнения алгоритма:

1) Идёт разложение числа на простые множители.

2) Составляется таблица значений ,

где .

3) Вычисляется .

Для от 1 до :

Пусть ,

где .

Тогда верно сравнение:

.

С помощью таблицы, составленной на шаге 1, находится .

Для от 0 до рассматривается сравнение

.

Решение находится по таблице

Конец цикла по .

Конец цикла по .

4) Найдя для всех , происходит поиск

по китайской теореме об остатках.

Была реализована модификация алгоритма, состоящая в том, что на 1 шаге алгоритма число было разложено на простые множители и данные простые множители были возведены в свои степени, чтобы на 2 шаге была составлена таблица из единичных значений без степеней. Данная модификация предположительно должна будет уменьшить расход памяти при составлении таблицы на 2 шаге и ускорить вычисления на 3 и 4 шаге алгоритма.

Для проверки работы модифицированного алгоритма были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 7 мс и затратил памяти 2426080 байт, а модифицированный за 6 мс и затратил 3806344 байт (рисунок 4).

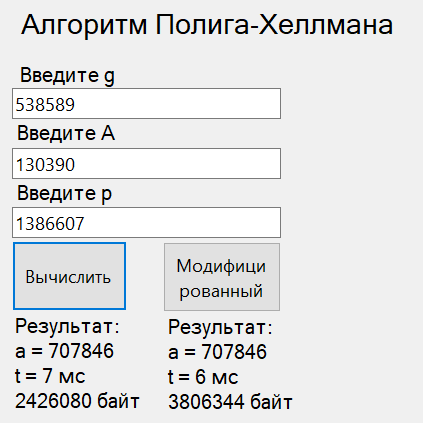


Рисунок 4 - Результат работы модифицированного алгоритма Полига-Хеллмана

4. Модификация алгоритма ро-метод Полларда

Была реализована модификация экспоненциального алгоритма дискретного логарифмирования ро-метод Полларда для факторизации (разложения на множители) целых чисел. Данный алгоритм основывается на алгоритме Флойда поиска длины цикла в последовательности и некоторых следствиях из парадокса дней рождения. Ро-метод Полларда строит числовую последовательность, элементы которой образуют цикл, начиная с некоторого номера , что может быть проиллюстрировано, расположением чисел в виде греческой буквы ρ, что послужило названием семейству алгоритмов.

Шаги выполнения алгоритма:

1) Генерируется случайно число между и .

2) Инициализируются числа , , .

3) В цикле вычисляется до тех пор, пока не будет равен 1.

4) Если равен , то присваивается и присваивается . Далее и .

5) После завершения цикла на 3 шаге возвращается результат, равный .

Была реализована модификация алгоритма, состоящая в том, что на 4 шаге алгоритма увеличилась степень вычисляемого . При вычислении степень полинома увеличилась до 3. Данная модификация предположительно увеличит скорость вычисления алгоритма в цикле на 3 шаге, но увеличит количество затрачиваемой памяти.

Для проверки работы модифицированного алгоритма было сгенерировано число . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 1 мс и затратил памяти 7968 байт, а модифицированный за 1 мс и затратил 8224 байт (рисунок 5).

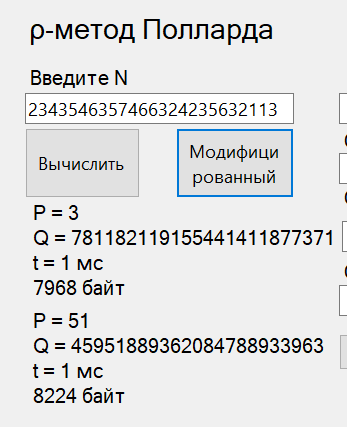


Рисунок 5 - Результат работы модифицированного алгоритма ро-метод Полларда

5. Модификация алгоритма Адлемана

Была реализована модификация алгоритма Адлемана, который является первым субэкспоненциальным алгоритмом дискретного логарифмирования в кольце вычетов по модулю простого числа. Алгоритм был предложен Леонардом Максом Адлеманом в 1979 году. Леонард Макс Адлеман — американский учёный-теоретик в области компьютерных наук, профессор компьютерных наук и молекулярной биологии в Университете Южной Калифорнии. Он известен как соавтор системы шифрования RSA и ДНК-вычислений. RSA широко используется в приложениях компьютерной безопасности, включая протокол HTTPS.

Пусть задано сравнение , необходимо найти натуральное число , удовлетворяющее данному сравнению.

Описание алгоритма:

1) Сформировывается факторная база, состоящая из всех простых чисел :

.

2) С помощью перебора идёт поиск натуральных чисел таких, что

,

то есть раскладывается по факторной базе. Отсюда следует, что .

3) Набрав достаточно много соотношений из 2 шага, решается получившаяся система линейных уравнений относительно неизвестных дискретных логарифмов элементов факторной базы .

4) С помощью некоторого перебора ищется одно значение , для которого , где – простые числа «средней» величины, то есть , где – также некоторая субэкспоненциальная граница, .

5) С помощью вычислений, аналогичных этапам 2 и 3 ищутся дискретные логарифмы .

6) Определяется искомый дискретный логарифм:

.

Была реализована модификация алгоритма, состоящая в том, что на 1 шаге алгоритма был изменён показатель степени при вычислении числа, тем самым повысив факторную базу. Данная модификация предположительно должна увеличить таблицу на 2 шаге для большего перебора результата алгоритма.

Для проверки работы модифицированного алгоритма были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 268 мс и затратил памяти 48040 байт, а модифицированный за 11612 мс и затратил 532072 байт (рисунок 6).

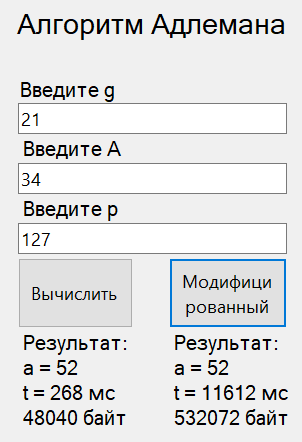


Рисунок 6 - Результат работы модифицированного алгоритма Адлемана

6. Модификация алгоритма COS

Была реализована модификация алгоритма COS (Копперсмит, Одлыжко, Шреппель), который является первым субэкспоненциальным алгоритмом дискретного логарифмирования в кольце вычетов по модулю простого числа.

Пусть задано сравнение , необходимо найти натуральное число , удовлетворяющее данному сравнению.

Описание алгоритма:

1) Задаётся . Сформировывается множество , где и – простые величины, .

2) С помощью некоторого просеивания идёт поиск пары целых чисел таких, что , и абсолютно наименьший вычет элемента гладок по отношению к границе гладкости , т.е.

.

При этом, поскольку , то

, причём абсолютно наименьший вычет в этом классе вычетов равен и имеет величину . Поэтому вероятность его гладкости выше, чем для произвольных чисел на отрезке . Логарифмируя по основанию , получается соотношение

*.*

Это однородное уравнение относительно неизвестных величин . Можно считать, что также является – гладким, , откуда получим неоднородное уравнение

*.*

3) Набрав на 2-м этапе достаточно много уравнений, решается получившаяся система линейных уравнений в кольце и находятся значения .

4) Для нахождения конкретного логарифма мы введём новую границу гладкости . Случайным перебором находим одно значение такое, что

.

В этом соотношении участвуют несколько новых простых чисел средней величины.

5) С помощью методов, аналогичных 2 и 3 этапам, мы находим логарифмы нескольких простых чисел средней величины, возникших на 4 этапе.

6) Находим ответ

.

Конец алгоритма.

Была реализована модификация алгоритма, состоящая в том, что на 2 шаге был увеличен наименьший вычет, добавив значение , чтобы увеличить разложение чисел при формировании СЛАУ. Данная модификация предположительно должна увеличить таблицу на 3 шаге для большего перебора результата алгоритма.

Для проверки работы модифицированного алгоритма были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 129 мс и затратил памяти 907296 байт, а модифицированный за 81 мс и затратил 4747168 байт (рисунок 7).

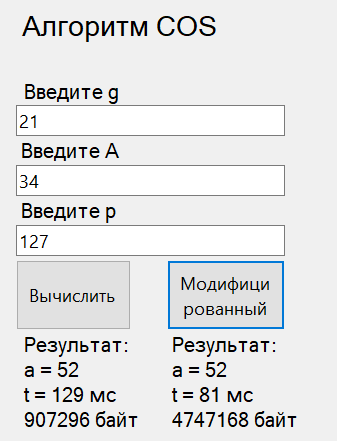


Рисунок 7 - Результат работы модифицированного алгоритма COS

7. Модификация решета числового поля

Была реализована модификация алгоритма решета числового поля, который является методом факторизации целых чисел.

Описание алгоритма:

1) Пусть — нечетное составное число, которое требуется факторизовать.

2) Выберем степень неприводимого многочлена (при не будет выигрыша в сравнении с методом квадратичного решета).

3) Выберем целое такое, что , и разложим по основанию :

.

4) Свяжем с разложением из 3 шага неприводимый в кольце полиномов с целыми коэффициентами многочлен

.

5) Определим полином просеивания как однородный многочлен от двух переменных и :

.

6) Определим второй полином и соответствующий однородный многочлен .

7) Выберем два положительных числа и , определяющих область просеивания:

.

8) Пусть  — корень . Рассмотрим кольцо полиномов . Определим множество, называемое алгебраической факторной базой , состоящее из многочленов первого порядка вида с нормой шага 5, являющейся простым числом. Эти многочлены — простые неразложимые в кольце алгебраических целых поля . Ограничим абсолютные значения норм полиномов из константой

9) Определим рациональную факторную базу , состоящую из всех простых чисел, ограниченных сверху константой .

10) Определим множество , называемое факторной базой квадратичных характеров. Это множество полиномов первого порядка , норма которых - простое число. Должно выполняться условие .

11) Выполним просеивание многочленов по факторной базе и целых чисел по факторной базе . В результате получим множество , состоящее из гладких пар , то есть таких пар , что НОД= 1, полином и число и раскладываются полностью по и соответственно.

12) Найдём такое подмножество , что

.

13) Определим многочлен

, где – производная .

14) Многочлен является полным квадратом в кольце полиномов . Пусть тогда есть корень из и — корень из .

15) Строим отображение , заменяя полином числом . Это отображение является кольцевым гомоморфизмом кольца алгебраических целых чисел в кольцо , откуда получаем соотношение:

.

16) Пусть . Найдём пару чисел таких, что . Тогда найдём делитель числа , вычисляя НОД.

Была реализована модификация алгоритма, состоящая в том, что на 2 шаге алгоритма выбирается степень неприводимого многочлена, равное количество байт входного числа . Данная модификация предположительно поможет эффективно выбирать не случайным образом степень полинома для дальнейших вычислений и повысит скорость вычисления алгоритма.

Для проверки работы модифицированного алгоритма было сгенерировано число . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 540 мс и затратил памяти 2014208 байт, а модифицированный за 235 мс и затратил 792064 байт (рисунок 8).

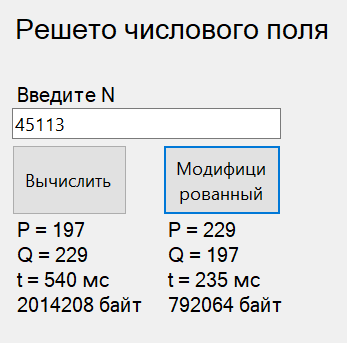


Рисунок 8 - Результат работы модифицированного решета числового поля

ЗАКЛЮЧЕНИЕ

В результате практики были исследованы, реализованы и протестированы модификации алгоритмов: алгоритм Шенкса, алгоритм Полига-Хеллмана, ро-метод Полларда, алгоритм Адлемана, алгоритм COS, решето числового поля. Для тестирования модификаций данных алгоритмов был использован генератор параметров Диффи-Хеллмана и возведение в степень по модулю. Также для тестирования алгоритмов был реализован замер времени выполнения алгоритма и количество потраченной памяти на выполнение алгоритма.

За период практики были приобретены следующие компетенции:

|  |  |  |
| --- | --- | --- |
| Компетенция | Расшифровка компетенции | Описание приобретенных знаний, умений и навыков |
| УК-6 | Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки | Приобретена способность реализовывать приоритеты собственной деятельности для разработки методов дискретного логарифмирования на основе самооценки |
| ПК-3 | Управление проектами в области информационных технологий малого и среднего уровня сложности в условиях неопределенностей, порождаемых запросами на изменения, с применением формальных инструментов управления рисками и проблемами проекта | Получен навык управления проектами в области информационной безопасности в условиях неопределённости, порождаемых запросами на изменения, с применением формальных инструментов управления рисками и проблемами проекта |
| ПК-4 | Управление проектами в области информационных технологий любого масштаба в условиях высокой неопределенности, вызываемой запросами на изменения и рисками, и с учетом влияния организационного окружения проекта; разработка новых инструментов и методов управления проектами в области информационных технологий | Получен навык управления проектами в области информационной безопасности любого масштаба в условиях высокой неопределенности, вызываемой запросами на изменения и рисками, и с учетом влияния организационного окружения проекта; разработка новых инструментов и методов управления проектами в области информационной безопасности |

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Теоретический минимум и алгоритмы цифровой подписи / Молдовян Н. А. – Текст: непосредственный // Книжный Дом «ЛИБРОКОМ», 2010. — С. 304.

2) The infrastructure of a real quadratic field and its applications. Proceedings of the Number Theory Conference. / D. Shanks. – Текст: непосредственный // University of Colorado, Boulder, 1972. — С. 217-224.

3) An Improved Algorithm for Computing Logarithms Over GF(p) and its Cryptographic Significance (англ.) / S. C. Pohlig and M. E. Hellman. - Текст: непосредственный // IEEE Transactions on Information Theory. — 1978. — Vol. 1, no. 24. — С. 106-110.

4) Theorems on factorization and primality testing / Pollard J.M. - Текст: непосредственный // Mathematical Proceedings of the Cambridge Philosophical Society. — 1974. — Т. 76, вып. 03. — С. 521–528.

5) A subexponential algorithm for discrete logarithms over all finite fields / Adleman L. M., Demarrais J. - Текст: непосредственный // Mathematics of computation. — 1993.

6) Теоретико-числовые алгоритмы в криптографии. / Василенко О.Н. - Текст: непосредственный // N— М.: МЦНМО, 2003. — C. 328.

7) Методы факторизации натуральных чисел. / Ишмухаметов Ш. Т. - Текст: непосредственный // — Казань: Казан. ун.. — 2011. — C. 190.

8) Applied Cryptography: Protocols, Algorithms, and Source Code in C. / Schneier, Bruce – Текст: непосредственный // Second Edition. — 2nd. — Wiley, 1996.

9) Методы факторизации натуральных чисел. / Ишмухаметов Ш. Т. - Текст: непосредственный // — Казань: Казан. ун.. — 2011. — C. 10.

10) Методы факторизации натуральных чисел. / Ишмухаметов Ш. Т. - Текст: непосредственный // — Казань: Казан. ун.. — 2011. — C. 52.

ПРИЛОЖЕНИЕ

**ИССЛЕДОВАНИЕ МЕТОДОВ ДИСКРЕТНОГО ЛОГАРИФМИРОВАНИЯ**

***Гусев В.Е.***

*Научный руководитель - канд. техн. наук Мубараков Б.Г.*

*Казанский (Приволжский) федеральный университет, Институт Вычислительной математики и информационных технологий*

*viegusev@stud.kpfu.ru*

В процессе практики были изучены и реализованы модификации методов дискретного логарифмирования на языке программирования C# на .NET8 в Windows Forms. Также для тестирования данных алгоритмов был реализован генератор параметров Диффи-Хеллмана, возведение числа в степень по модулю [1], замер времени выполнения алгоритма и количество потраченной памяти на выполнение алгоритма.

Были реализованы модификации экспоненциальных алгоритмов дискретного логарифмирования: алгоритм Шенкса [2], алгоритм Полига-Хеллмана [3], ро-метод Полларда [4], а также модификации субэкспоненциальных алгоритмов дискретного логарифмирования: алгоритм Адлемана [5], алгоритм COS [6], решето числового поля [7]. Для разработки всех алгоритмов были реализованы вспомогательные математические функции.

Была реализована функция быстрого возведения в степень по модулю. Возведение в степень по модулю — это операция над натуральными числами возведения в степень, выполняемая по модулю. Находит применение в информатике, особенно, в области криптографии с открытым ключом. Возведение в степень по модулю — это вычисление остатка от деления натурального числа (основание), возведенного в степень (показатель степени), на натуральное число (модуль). Обозначается: [8, 9].

Для проверки чисел на простоту был реализован тест Миллера-Рабина. Данный тест является вероятностным полиномиальным тестом простоты. Тест Миллера-Рабина, наряду с тестом Ферма и тестом Соловея-Штрассена, позволяет эффективно определить, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера-Рабина часто используется в криптографии для получения больших случайных простых чисел.

Была реализована модификация экспоненциального алгоритма «Шаг младенца — шаг великана» — в теории групп детерминированный алгоритм дискретного логарифмирования в мультипликативной группе кольца вычетов по модулю простого числа. Был предложен советским математиком Александром Гельфондом в 1962 году и Дэниелом Шенксом в 1972 году. Метод теоретически упрощает решение задачи дискретного логарифмирования, на вычислительной сложности которой построены многие криптосистемы с открытым ключом. Относится к методам встречи посередине. Это был один из первых методов, который показал, что задача вычисления дискретного логарифма может быть решена значительно быстрее, чем методом перебора. Идея алгоритма состоит в выборе оптимального соотношения времени и памяти, а именно в усовершенствованном поиске показателя степени.

Была реализована модификация алгоритма «Шаг младенца — шаг великана», состоящая в распараллеливании 2 и 3 шага алгоритма. На 2 шаге алгоритма параллельно вычисляются два ряда чисел. На 3 шаге был сделан параллельный поиск результата с начала и с конца ряда. Данная модификация предположительно должна увеличить скорость работу с рядами чисел.

Для проверки работы модифицированного алгоритма «Шаг младенца — шаг великана» были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 81 мс и затратил памяти 3947976 байт, а модифицированный за 120 мс и затратил 2987840 байт.

Была реализована модификация детерминированного алгоритма дискретного логарифмирования с экспоненциальной сложностью Полига-Хеллмана в кольце вычетов по модулю простого числа. Одной из особенностей алгоритма является то, что для простых чисел специального вида можно находить дискретный логарифм за полиномиальное время. Данный алгоритм был придуман американским математиком Роландом Сильвером, но впервые был опубликован другими двумя американскими математиками Стивеном Полигом и Мартином Хеллманом в 1978 году в статье «An improved algorithm for computing logarithms over GF(p) and its cryptographic significance», которые независимо от Роланда Сильвера разработали данный алгоритм.

Была реализована модификация алгоритма Полига-Хеллмана, состоящая в том, что на 1 шаге алгоритма число было разложено на простые множители и данные простые множители были возведены в свои степени, чтобы на 2 шаге была составлена таблица из единичных значений без степеней. Данная модификация предположительно должна будет уменьшить расход памяти при составлении таблицы на 2 шаге и ускорить вычисления на 3 и 4 шаге алгоритма.

Для проверки работы модифицированного алгоритма Полига-Хеллмана были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 7 мс и затратил памяти 2426080 байт, а модифицированный за 6 мс и затратил 3806344 байт.

Была реализована модификация экспоненциального алгоритма дискретного логарифмирования ро-метод Полларда для факторизации (разложения на множители) целых чисел [10]. Данный алгоритм основывается на алгоритме Флойда поиска длины цикла в последовательности и некоторых следствиях из парадокса дней рождения. Ро-метод Полларда строит числовую последовательность, элементы которой образуют цикл, начиная с некоторого номера , что может быть проиллюстрировано, расположением чисел в виде греческой буквы ρ, что послужило названием семейству алгоритмов.

Была реализована модификация алгоритма ро-метод Полларда, состоящая в том, что на 4 шаге алгоритма увеличилась степень вычисляемого . При вычислении x степень полинома увеличилась до 3. Данная модификация предположительно увеличит скорость вычисления алгоритма в цикле на 3 шаге, но увеличит количество затрачиваемой памяти.

Для проверки работы модифицированного алгоритма ро-метод Полларда было сгенерировано число . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 1 мс и затратил памяти 7968 байт, а модифицированный за 1 мс и затратил 8224 байт.

Была реализована модификация алгоритма Адлемана, который является первым субэкспоненциальным алгоритмом дискретного логарифмирования в кольце вычетов по модулю простого числа. Алгоритм был предложен Леонардом Максом Адлеманом в 1979 году. Леонард Макс Адлеман — американский учёный-теоретик в области компьютерных наук, профессор компьютерных наук и молекулярной биологии в Университете Южной Калифорнии. Он известен как соавтор системы шифрования RSA и ДНК-вычислений. RSA широко используется в приложениях компьютерной безопасности, включая протокол HTTPS.

Была реализована модификация алгоритма Адлемана, состоящая в том, что на 1 шаге алгоритма был изменён показатель степени при вычислении числа , тем самым повысив факторную базу. Данная модификация предположительно должна увеличить таблицу на 2 шаге для большего перебора результата алгоритма.

Для проверки работы модифицированного алгоритма Адлемана были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 268 мс и затратил памяти 48040 байт, а модифицированный за 11612 мс и затратил 532072 байт.

Была реализована модификация алгоритма COS (Копперсмит, Одлыжко, Шреппель), который является первым субэкспоненциальным алгоритмом дискретного логарифмирования в кольце вычетов по модулю простого числа. В 1986 г. Копперсмит, Одлыжко и Шреппель предложили алгоритм дискретного логарифмирования с эвристической оценкой сложности арифметических операций.

Была реализована модификация алгоритма COS, состоящая в том, что на 2 шаге был увеличен наименьший вычет, добавив значение , чтобы увеличить разложение чисел при формировании СЛАУ. Данная модификация предположительно должна увеличить таблицу на 3 шаге для большего перебора результата алгоритма.

Для проверки работы модифицированного алгоритма COS были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 129 мс и затратил памяти 907296 байт, а модифицированный за 81 мс и затратил 4747168 байт.

Была реализована модификация алгоритма решето числового поля, который является методом факторизации целых чисел. Метод решета числового поля (как специальный, так и общий) можно представить как усовершенствование более простого метода — метода рационального решета либо метода квадратичного решета. Подобные им алгоритмы требуют нахождения гладких чисел порядка . Размер этих чисел экспоненциально растёт с ростом . Метод решета числового поля, в свою очередь, требует нахождения гладких чисел субэкспоненциального относительно размера. Благодаря тому, что эти числа меньше, вероятность того, что число такого размера окажется гладким, выше, что и является причиной эффективности метода решета числового поля. Для достижения ускорения вычислений в рамках метода проводятся в числовых полях, что усложняет алгоритм, по сравнению с более простым рациональным решетом.

Была реализована модификация алгоритма решето числового поля, состоящая в том, что на 2 шаге алгоритма выбирается степень неприводимого многочлена, равное количество байт входного числа . Данная модификация предположительно поможет эффективно выбирать не случайным образом степень полинома для дальнейших вычислений и повысит скорость вычисления алгоритма.

Для проверки работы модифицированного алгоритма решето числового поля было сгенерировано число . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 540 мс и затратил памяти 2014208 байт, а модифицированный за 235 мс и затратил 792064 байт.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) *Молдовян Н. А.* Теоретический минимум и алгоритмы цифровой подписи / Молдовян Н. А. – Книжный Дом «ЛИБРОКОМ», 2010. — 304 с.

2) *D. Shanks.* The infrastructure of a real quadratic field and its applications. Proceedings of the Number Theory Conference. / D. Shanks. – University of Colorado, Boulder, 1972. — С. 217-224.

3) An Improved Algorithm for Computing Logarithms Over GF(p) and its Cryptographic Significance (англ.) / S. C. Pohlig, M. E. Hellman. // IEEE Transactions on Information Theory. — 1978. — Vol. 1, no. 24. — С. 106-110.

4) *Pollard J.M.* Theorems on factorization and primality testing / Pollard J.M. // Mathematical Proceedings of the Cambridge Philosophical Society. — 1974. — Т. 76, вып. 03. — С. 521–528.

5) A subexponential algorithm for discrete logarithms over all finite fields / Adleman L. M., Demarrais J. // Mathematics of computation. — 1993.

6) *Василенко О.Н.* Теоретико-числовые алгоритмы в криптографии. / Василенко О.Н. // N— М.: МЦНМО, 2003. — 328 с.

7) *Ишмухаметов Ш. Т.* Методы факторизации натуральных чисел. / Ишмухаметов Ш. Т. // — Казань: Казан. ун.. — 2011. — 190 с.

8) Applied Cryptography: Protocols, Algorithms, and Source Code in C. / Schneier, Bruce // Second Edition. — 2nd. — Wiley, 1996.

9) *Ишмухаметов Ш. Т.* Методы факторизации натуральных чисел. / Ишмухаметов Ш. Т. // — Казань: Казан. ун.. — 2011. — 10 с.

10) *Ишмухаметов Ш. Т.* Методы факторизации натуральных чисел. / Ишмухаметов Ш. Т. // — Казань: Казан. ун.. — 2011. — 52 с.