МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

Изображение выглядит как текст, Шрифт, снимок экрана, логотип

Автоматически созданное описание

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

**по дисциплине: «*Программирование*»**

|  |  |
| --- | --- |
| Выполнил:  Студент гр. «АБc-324», «АВТФ»  *Деревянкин Дмитрий Александрович*  «\_\_\_» \_\_\_\_\_\_ 2025г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | Проверил:  *Доцент кафедры ЗИ*  *Архипова Анастасия Борисовна*  «\_\_\_» \_\_\_\_\_\_ 2025г  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Новосибирск 2025

СОДЕРЖАНИЕ

[Введение 3](#_Toc191586147)

[1. Теоретическая часть 4](#_Toc191586148)

[1.1 Алгоритм работы шифровки A1Z26 4](#_Toc191586149)

[1.2 Алгоритм работы шифровки Цезаря 5](#_Toc191586150)

[1.3 Алгоритм работы протокола Диффи-Хеллмана 7](#_Toc191586151)

[2. Практическая часть 10](#_Toc191586152)

[2.1 Постановка задачи 10](#_Toc191586153)

[2.2 Характеристика задачи 10](#_Toc191586154)

[2.3 Алгоритм решения 11](#_Toc191586155)

[2.4 Руководство пользователя 13](#_Toc191586156)

[2.5 Руководство системного программиста 17](#_Toc191586157)

[2.6 Контрольный пример 19](#_Toc191586158)

[2.7 Ручной расчет 21](#_Toc191586159)

[Список использованных источников 26](#_Toc191586160)

[Приложение 27](#_Toc191586161)

# ВВЕДЕНИЕ

В современном обществе все большую роль играют компьютеры, и вообще электронные средства передачи, хранения, и обработки информации. Для того чтобы информационные технологии можно было использовать в различных областях, необходимо обеспечить их надежность и безопасность. Поэтому широкое использование информационных технологий привело к бурному развитию различных методов защиты информации, из которых основными можно, пожалуй, назвать, помехоустойчивое кодирование и криптографию.

Цель работы — это разработка программы, которая будет шифровать и дешифровать текст. Задача заключается в том, чтобы разработать алгоритмы, они будут обеспечивать безопасность обработки данных, которые будут доступны тем, кто обладает специальным ключом.

Результаты при создании программного обеспечения, должен включать работу шифрования и дешифрования текстовых данных, и также применение этого программного обеспечения для работы с задачами в области информационной безопасности.

Теоретической основой написания расчетно-графической работы являлись следующие источники: Фионов А.Н. [1], Ильин М.Е. [4], И.И.Ефишов [5].

Практической основой написания расчетно-графической работы являлись следующие источники, помогающие разработать алгоритм решения задачи: Павловская Т. А. [2], Мартин Р. [3], Страуструп Б. [6], Мейерс С. [7 – 8], Саттер Г. [9], Прата С. [10].

## 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В программе используются 3 метода шифрования: Шифр A1Z26, Шифр Цезаря, Протокол Диффи-Хеллмана.

# 1.1 Алгоритм работы шифровки A1Z26

1.1.1 Введение

Шифр A1Z26 (A1Z33), по-другому моноалфавитная замена — это простой шифр замены, в котором каждая буква алфавита заменяется её порядковым номером. Этот шифр легко расшифровать, но его можно усложнить, добавляя пробелы, заменяя числа другими символами или комбинируя с другими методами шифрования. Шифр A1Z26 представляет собой простой шифр замещения, декодируемый путем замены nᵗʰ буквы алфавита на заданное число n [4].

1.1.2 Исторический контекст

Шифр A1Z26 считается древнейшим и простейшим шифром, который зародился одновременно в нескольких развитых древних государствах примерно за 400 лет до н. э [4].

1.1.3 Принцип работы

Это простая подстановка, где каждая буква заменена её порядковым номером в алфавите. Только нижний регистр.

Алфавит выглядит следующим образом:

русские: А-Я = 0-33  
английские: A-Z = 34-59  
английские: a-z = 60-85  
русские: а-я = 86-118  
специальные символы: 119-149  
арабские цифры: 150-159

Условно сообщение "Hello Мир". Нам нужно закодировать по шаблону выше у «H» код в нашем алфавите 41, у «e» – 64 и так далее. То есть для кодирования символов от данного сообщения нам нужно определить номер каждого символа в нашем алфавите.  
H = 41 (код нашего алфавита)   
e = 64  
l = 71  
....

1.1.4 Криптоанализ

Несмотря на свою простоту, шифр A1Z26 имеет ограничения. Он не обеспечивает высокую степень безопасности, и его легко разгадать, особенно если злоумышленник знает о методе шифрования. Дешифровка аналогично, только мы наше число используем как индекс в алфавите и заменяем на соответствующий символ [4].

1.1.5 Применение

В наши дни данный шифр не используется для защиты определенной информации, так как очень прост в расшифровке. Но его часто используют в развлекательных целях: головоломках, игр и даже в мультсериалах. Данный метод шифрования может помочь студентам понять азы шифрования и дешифрования [4].

# 1.2 Алгоритм работы шифровки Цезаря

1.2.1 Введение

Шифр Цезаря — это простой метод шифрования, где каждая буква заменяется на другую, сдвинутую на фиксированное число позиций в алфавите. Этот шифр легко расшифровывается, поэтому используется для базовой защиты или учебных примеров [1].

1.2.2 Исторический контекст

Шифр Цезаря назван в честь римского полководца Гая Юлия Цезаря, который использовал его для секретной переписки с военачальниками. В его варианте использовался сдвиг на 3 буквы.

Этот шифр применялся в античные времена, но с развитием криптографии стал легко взламываемым. Несмотря на простоту, он положил основу для более сложных методов шифрования. Сегодня используется в учебных целях и простых кодировках [1].

1.2.3 Принцип работы

Основная идея шифра Цезаря заключается в том, что каждая буква в открытом тексте заменяется другой буквой, которая находится на фиксированном количестве позиций дальше в алфавите. Это количество позиций называется "сдвигом" (или ключом).

Алфавит выглядит следующим образом:

русские: А-Я = 0-33  
английские: A-Z = 34-59  
английские: a-z = 60-85  
русские: а-я = 86-118  
специальные символы: 119-149  
арабские цифры: 150-159

Условно сообщение «Hello Мир». Сдвиг = 3. Тогда буква «H» в нашем алфавите имеет номер 41, и с учётом сдвига зашифрованный символ будет иметь код 41 + 3 = 44, то есть это буква K, «e» заменяем на букву h (64 + 3 = 67) и так далее.

То есть для кодирования символов от данного сообщения нам нужно определить номер каждого символа в нашем алфавите и прибавить сдвиг, а после подставить символ с новым номером.

H = 41 + 3 = 44 – K  
e = 64 + 3 = 67 – h  
l = 71 + 3 = 74 – o

…

1.2.4 Криптоанализ

Уязвимости криптографических алгоритмов могут возникать из-за слабостей в их математических основах, ошибок в реализации или использования слабых ключей. Например, использование предсказуемых или повторяющихся ключей может значительно облегчить работу злоумышленника. Кроме того, ошибки в реализации, такие как неправильное управление памятью или незащищенные процессы генерации ключей, могут создавать дополнительные возможности для атак [1].

1.2.5 Применение

Шифр Цезаря является хорошим примером для изучения основ криптографии и криптоанализа. Несмотря на свою простоту, он иллюстрирует важные концепции, такие как частотный анализ и методы перебора. Однако для защиты серьезной информации этот метод не подходит из-за своей уязвимости к различным атакам [1].

# 1.3 Алгоритм работы протокола Диффи-Хеллмана

1.3.1 Введение

Протокол Диффи-Хеллмана — это криптографический метод, который позволяет двум сторонам безопасно обмениваться ключами шифрования через незащищенный канал связи. Он был предложен в 1976 году Уитфилдом Диффи и Мартином Хеллманом и стал одним из первых примеров использования концепции обмена ключами [5].

1.3.2 Исторический контекст

Протокол Диффи — Хеллмана, разработанный Уитфилдом Диффи и Мартином Хеллманом в 1976 году, стал революционным открытием в области криптографии. Он позволяет двум участникам безопасно обмениваться секретной информацией через незащищенные каналы связи. Идея протокола основана на математической задаче дискретного логарифма.

Он является одним из основных инструментов современной криптографии и широко используется для защиты конфиденциальности данных в интернете. Некоторые эксперты указывают на возможность атаки посредника, при которой злоумышленник может перехватить передаваемые сообщения и изменить их содержимое без ведома общающихся сторон. Кроме того, протокол требует больших вычислительных ресурсов, особенно при использовании больших ключей, что может быть непрактичным для устройств с ограниченными вычислительными возможностями) [5].

1.3.3 Принцип работы

Протокол Диффи-Хеллмана позволяет двум сторонам, которые хотят безопасно обмениваться данными, создать общий секретный ключ через незащищенный канал связи. Принцип работы протокола основан на математических свойствах модульной арифметики и сложности вычисления дискретного логарифма [5].

Алиса и Боб выбирают два публичных значения: большое простое число p и основание g (генератор). Эти значения могут быть известны всем и не требуют секретности. Затем Алиса выбирает случайное секретное число a (где 1 < a < p) и вычисляет:

A = ga mod p.

Затем она отправляет значение A Бобу.

Боб выбирает случайное секретное число b (где 1 < b < p) и вычисляет:

B = gb mod p.

Затем он отправляет значение B Алисе.

Теперь Алиса имеет значение A и может вычислить общий секретный ключ K следующим образом:

K = Ba mod p.

Боб, имея значение B, также вычисляет общий ключ K:

K = Ab mod p

В результате обе стороны получают один и тот же общий секретный ключ K*K*, который может быть использован для симметричного шифрования сообщений.

1.3.4 Криптоанализ

Протокол Диффи-Хеллмана обеспечивает безопасный обмен ключами, но подвержен атакам "человек посередине", если не используется аутентификация. Уязвимости, такие как атака Logjam и квантовые угрозы, требуют применения длинных ключей (2048+ бит) или перехода на эллиптические кривые (ECDH) для повышения безопасности [5].

1.3.5 Применение

Протокол Диффи-Хеллмана широко используется в различных системах для обеспечения конфиденциальности при обмене данными, таких как протоколы TLS/SSL для защищённых веб-соединений, VPN для безопасных частных сетей и SSH для удалённого управления серверами. Он также применяется в электронной почте (PGP) и других приложениях, где требуется защищённый обмен ключами [5].

# 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

# 2.1 Постановка задачи

Необходимо сделать программу, которая будет шифровать и дешифровать данные одним из алгоритмов, которые представлены. Программа должна выполнять основные действия:

- обеспечить процедуру проверки пользовательских данных путем авторизации в системе;

- предусмотреть ввод исходного текста с клавиатуры или загрузку данных из файла;

- реализовать шифрование данных и демонстрацию полученных результатов;

- реализовать дешифрование (расшифрование) данных и демонстрацию полученных результатов (консоль, файл).

Методы шифрования реализовать в виде отдельных функций/модулей.

# 2.2 Характеристика задачи

2.2.1 Программа «*Шифровальщик*» предназначена для автоматизации шифрования и расшифрования исходных данных авторизованного в системе пользователя.

2.2.2 Программа используется пользователем для защиты персональной информации.

2.2.3 Периодичность решения задачи по запросу пользователя.

2.2.4 Прекращение автоматизированного решения задачи происходит при отключении источника электропитания ЭВМ.

2.2.5 Связь с другими задачами отсутствует.

2.2.6 Специальных ограничений на временные характеристики решения задачи не налагается.

# 2.3 Алгоритм решения

1. Запустить приложение «Шифровальщик»;

2. Вывод: «Введите пароль: »;

3. Ввод пароля;

4. Вывод меню:

«1 – шифр Цезаря»;

«2 - шифр A1Z26»;

«3 – алгоритм Диффи-Хеллмана»;

«0 – выйти из программы»;

5. Если выбран пункт «1 - шифр Цезаря»:

5.1 Вывод: «Введите сообщение: »;

5.2 Ввод сообщения;

5.3 Вывод: «Введите название файла, в который будет записано сообщение: »;

5.4 Ввод названия файла, в котором будет храниться текст;

5.5 Вывод зашифрованного сообщения «Зашифрованное сообщение: »;

5.6 Вывод: «Введите название файла, в который будет записано зашифрованное сообщение: »;

5.7 Ввод названия файла, в котором будет храниться расшифрованный текст;

5.8 Программа предлагает расшифровать текст: «Расшифровать сообщение? Для подтверждения введите /y/ или /Y/: »;

5.8.1 Если выбрано расшифрование, то ввод: «Введите название файла, в котором хранится зашифрованное сообщение: »;

5.8.2 Вывод: «Расшифрованное сообщение: »;

5.8.3 Ввод: «Введите название файла, в котором будет хранится зашифрованный файл: »;

5.8.4 Ввод названия файла, куда будет записано расшифрованное сообщение;

5.9 Ввод: «Выберите номер шифра: »;

6. Если выбран пункт «2 –шифр A1Z2»:

6.1 Вывод: «Введите сообщение: »;

6.2 Ввод сообщения;

6.3 Вывод: «Введите название файла, в который будет записано сообщение: »;

6.4 Ввод названия файла, в котором будет храниться текст;

6.5 Вывод зашифрованного сообщения «Зашифрованное сообщение: »;

6.6 Вывод: «Введите название файла, в который будет записано зашифрованное сообщение: »;

6.7 Ввод названия файла, в котором будет храниться расшифрованный текст;

6.8 Программа предлагает расшифровать текст: «Расшифровать сообщение? Для подтверждения введите /y/ или /Y/: »;

6.8.1 Если выбрано расшифрование, то ввод: «Введите название файла, в котором хранится зашифрованное сообщение: »;

6.8.2 Вывод: «Расшифрованное сообщение: »;

6.8.3 Ввод: «Введите название файла, в котором будет хранится зашифрованный файл: »;

6.8.4 Ввод названия файла, куда будет записано расшифрованное сообщение;

6.9 Ввод: «Выберите номер шифр: »;

7. Если выбран пункт «3 – алгоритм Диффи-Хеллмана»:

7.1 Вывод: «Введите секретный ключ участника A: »;

7.2 Ввод секретного ключа участника А;

7.3 Вывод: «Введите секретный ключ участника B: »;

7.4 Ввод секретного ключа участника B;

7.5 Вывод: «Общий секретный ключ: »;

7.6 Вывод меню:

«1 – шифр Цезаря»;

«2 - шифр A1Z26»;

7.7 Вывод: «Выберите шифр, с которым использовать общий секретный ключ: »;

7.8 Если выбран пункт «1 – шифр Цезаря», то переходить к пункту 5.

7.9 Если выбран пункт «2 – шифр A1Z26», то переходить к пункту 6.

8. Если выбран пункт «0 - выйти из программы»;

8.1 Программа завершает свою работу;

# 2.4 Руководство пользователя

2.4.1 Введение

2.4.1.1 Программа «Шифровальщик» предназначена для автоматизации шифрования и расшифрования исходных данных авторизованного в системе пользователя.

2.4.1.2 Программа предоставляет пользователю следующие возможности:

- полный доступ в систему;

- выбор алгоритма шифрования;

- выбор шифрования из консоли/файла;

- шифрование/расшифрование данных;

- вывод результата в консоль/файл.

2.4.1.3 Программа реализована на языке C++. Работает в любой среде совместимой с «Windows». Дисковой памяти для запуска программы требуется не менее 300 Mb. Оперативной памяти для запуска программы требуется не менее 16 Mb.

2.4.2 Описание операций

Для удобства пользователя и более легкого изучения системы большинство форм и диалогов имеют идентичный интерфейс. Далее описаны все функции системы, а также формы и диалоги для ввода и вывода информации. Для каждой формы приведены основные компоненты и их назначения.

После запуска на экран выводится окно программы (рис. 2.1). Для начала работы с программой необходимо ввести правильный пароль («2453»)

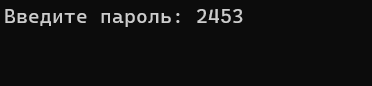


Рис. 2.1 – Окно программы

Если введен некорректный пароль программа запросит ввести его вновь (рис. 2.2).

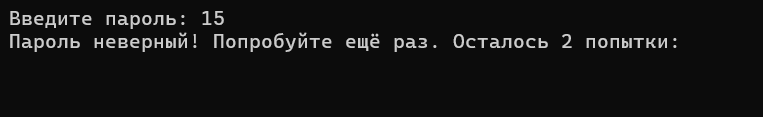


Рис. 2.2 – Ошибка «Пароль неверный! Попробуйте ещё раз»

В случае если ввести пароль неправильно второй раз программа предупредит о том, что осталась последняя попытка. После она завершится (рис. 2.3).

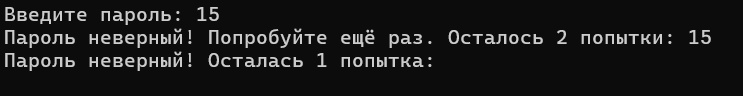


Рис. 2.3 – Ошибка «Пароль неверный! Осталась 1 попытка»

После верно введенного пароля, программа предлагает выбор алгоритмов шифрования (рис. 2.4).

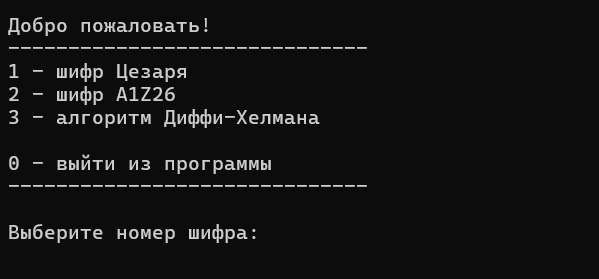


Рис. 2.4 – Вывод меню алгоритмов шифрования

При выборе «1 – шифр Цезаря» начинается диалог с пользователем (рис. 2.5). Пользователю предлагают ввести текст и записать его в файл, а после зашифровать его. После нам предлагают расшифровать текст по файлу, в котором хранится зашифрованное сообщение.

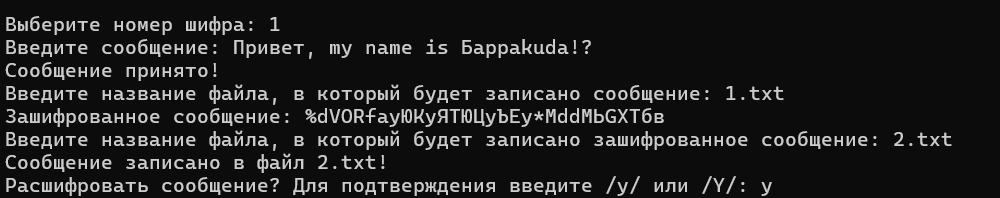


Рис. 2.5 – Вывод «1 – шифр Цезаря»

Тот же способ взаимодействия с пользователем шифра A1Z26 (рис. 2.6).

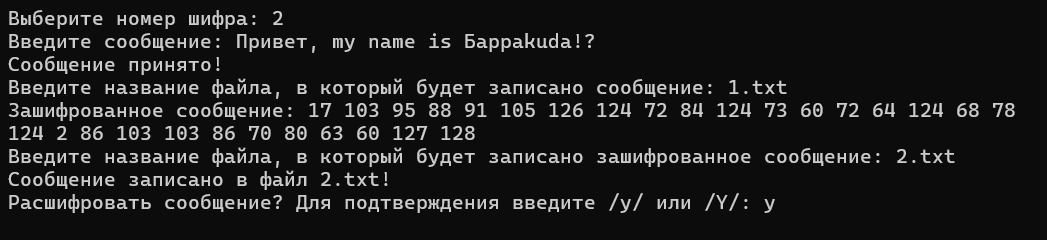


Рис. 2.6 – Вывод «2 – шифр A1Z26»

При выборе «3 – алгоритм Диффи-Хеллмана» начинается диалог с пользователем (рис. 2.7). Пользователю предлагают ввести секретные ключи участников A и B. После высчитывается общий секретный ключ и предлагается выбрать шифр, с которым дальше пользователь будет работать, используя общий секретный ключ.

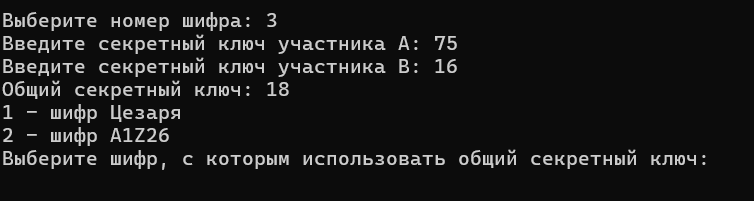


Рис. 2.7 – Вывод «3 – алгоритм Диффи-Хеллмана»

2.4.3 Сообщение пользователю

При работе с программой могут возникнуть некоторые ошибки, которые представлены в таблице 2.1.

Таблица 2.1 – Ошибки, которые могут появиться

|  |  |
| --- | --- |
| Текст сообщение программы | Ответ пользователя |
| Пароль неверный! Попробуйте ещё раз. Осталось 2 попытки:  Пароль неверный! Осталась 1 попытка: | Выводится при неверном вводе пароля. |
| Вводите только число | Выводится, если при выборе шифра, введено не число |
| Выбран неверный номер шифра! | Выводится, если был выбран некорректный пункт |
| Ошибка, введены некорректные символы: | Выводится, если происходит ввод символа/ов, который/ые не шифруется. |
| Error: Unable to open the file | Выводится при неверном пути для считывания из файла. |

2.4.4 Аварийные ситуации

К аварийным ситуациям относятся: нехватка оперативной памяти для запуска. Если исполняемый модуль программы не запускается, либо не выполнены требования условия работы программы, необходимо обратиться к разработчику программы.

# 2.5 Руководство системного программиста

Программа реализована на языке C++ в среде Visual Studio Code.

Модули программы:

main.cpp. Содержит функцию main, представляющую функционал по вводу пароля, выводу меню выбора алгоритмов шифрования, с соответствующим вызовом функций, объявленных в заголовочном файле include.h:

Объявление функции для работы с файлами

string FileInput(string filename)– функция чтения из файла;

string FileOutput(string filename, string str)– функция записи в файл;

void input\_and\_check(string& message, string choice\_shifr) – функция для ввода и проверки сообщения;

void Crypt(string choice\_shifr, int key) – функция для

шифрования и дешифрования;

const string my\_password – объявление переменной пароля;

a1z26.cpp

string A1Z26Encryption(const string& message, int key) – функция шифровки сообщения с помощью шифра A1Z26;

string A1Z26Descryption(const string& message, int key) – функция расшифровки сообщения с помощью шифра A1Z26;

vector<char> checkInputA1Z26(const string& message) – функция для проверки ввода для шифра a1z26;

caeser.cpp

string CaesarEncryption(const string& message, const int& shift, int key) – функция шифрования сообщения с помощью шифра Цезаря;

string CaesarDescryption(const string& message, const int& shift, int key) – функция расшифровки сообщения с помощью шифра Цезаря;

vector<char> checkInputCaesar(const string& message) - функция для проверки ввода для шифра Цезаря;

diffiehellman.cpp

int modExp(int base, int exp, int mod) – функция для вычисления модуля числа (baseexp % mod);

int SharedSecretKey(int a, int b) – функция для вычисления общего секретного ключа;

Связь модулей представлена на рисунке 2.7

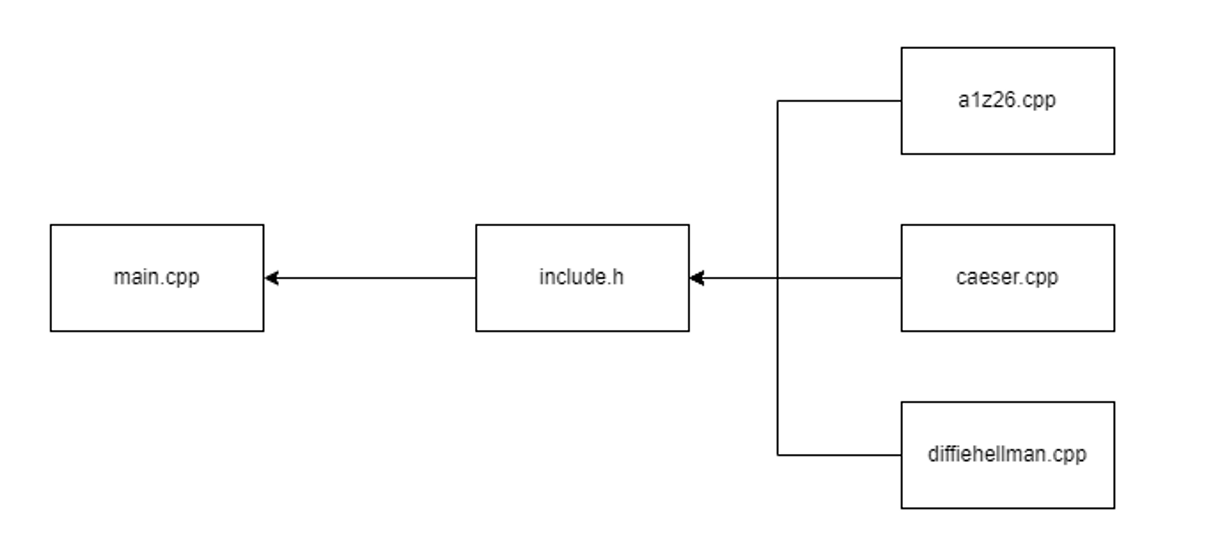


Рисунок 2.7 – Связь модулей

# 2.6 Контрольный пример

После запуска на экран выводится окно программы, где необходимо ввести правильный пароль (рис. 2.1).

После верно введенного пароля, программа предлагает выбор алгоритмов шифрования (рис. 2.4).

Если пользователь выберет шифр Цезаря, программа предложит ввести сообщение. После чего программа предложит нам сохранить текст в файл и зашифрует его, и предложит сохранить зашифрованное сообщение. Далее перед нами будет стоять выбор расшифровать сообщение, для этого нужно указать верный путь к зашифрованному сообщению, или нет. Если мы выбрали расшифровать сообщение, то на экране будет вывод нашего расшифрованного сообщения, а также нужно выбрать файл, куда будет записано расшифрованное сообщение (рис. 2.8).

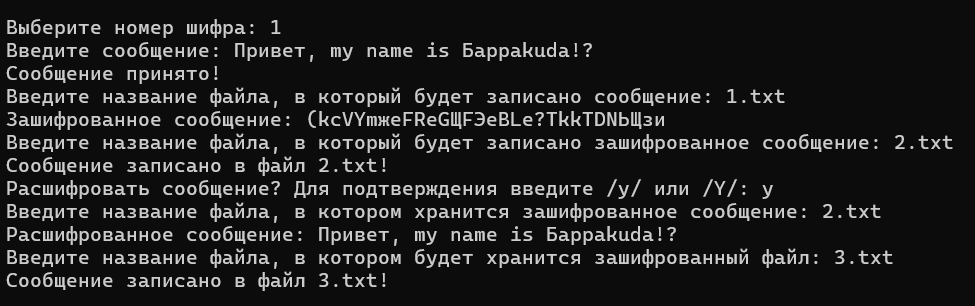


Рис. 2.8 – Работа с шифром Бекона

Если пользователь выберет шифр A1Z26, программа предложит ввести сообщение. После чего программа предложит нам сохранить текст в файл и зашифрует его, и предложит сохранить зашифрованное сообщение. Далее перед нами будет стоять выбор расшифровать сообщение, для этого нужно указать верный путь к зашифрованному сообщению, или нет. Если мы выбрали расшифровать сообщение, то на экране будет вывод нашего расшифрованного сообщения, а также нужно выбрать файл, куда будет записано расшифрованное сообщение (рис. 2.9).

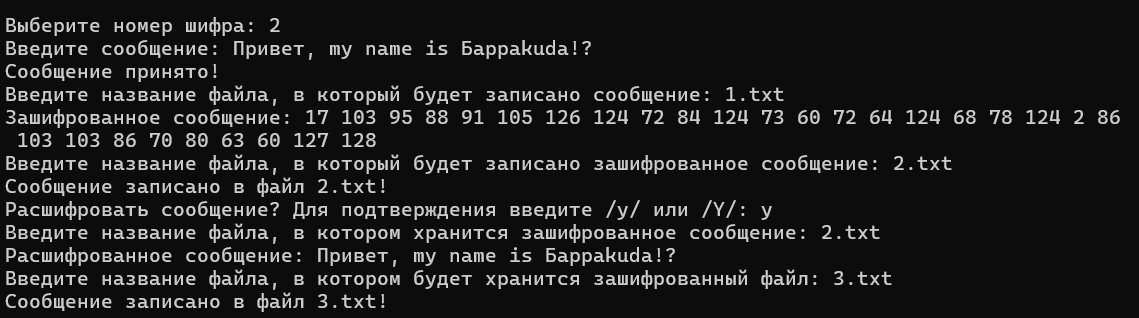


Рис. 2.9 – Работа с шифром A1Z26

Если пользователь выберет алгоритм Диффи-Хеллмана, программа предложит ввести секретные ключи участников A и B. После чего программа выведет нам общий секретный ключ. После программа предложит нам выбрать шифр, с которым будет использоваться общий секретный ключ. Далее программа предложит ввести сообщение. После чего программа предложит нам сохранить текст в файл и зашифрует его, и предложит сохранить зашифрованное сообщение. Далее перед нами будет стоять выбор расшифровать сообщение, для этого нужно указать верный путь к зашифрованному сообщению, или нет. Если мы выбрали расшифровать сообщение, то на экране будет вывод нашего зашифрованного сообщения (рис. 2.10).

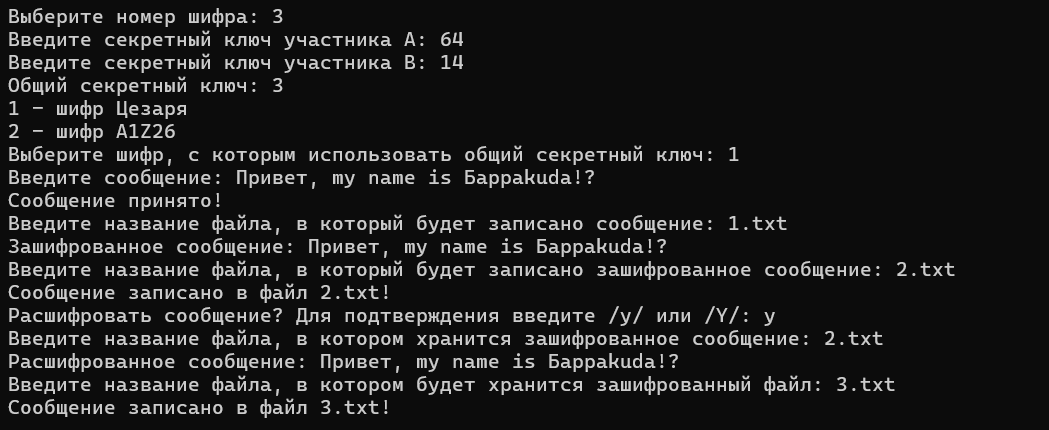


Рис. 2.10 – Работа с алгоритмом Диффи-Хеллмана

# 2.7 Ручной расчет

2.7.1 Шифровка с помощью шифра Цезаря

Текст – Привет, my name is Барраkuda!?; Сдвиг = 17;

Разделим строку на отдельные символы:

Шифровка английских символов:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Текст | m | y | n | a | e | i |
| Номер | 72 | 84 | 73 | 60 | 64 | 68 |
| После сдвига | 89 | 101 | 90 | 77 | 81 | 85 |
| Шифр | г | о(рус) | д | r | v | z |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Текст | s | k | u | d |
| Номер | 78 | 70 | 80 | 63 |
| После сдвига | 95 | 87 | 97 | 80 |
| Шифр | и | б | к | u |

Шифровка русских символов:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Текст | П | р | и | в | е | т |
| Номер | 17 | 103 | 95 | 88 | 91 | 105 |
| После сдвига | 34 | 120 | 112 | 105 | 108 | 122 |
| Шифр | A(англ) | - | щ | т | х(рус) | / |

|  |  |  |
| --- | --- | --- |
| Текст | Б | а |
| Номер | 2 | 86 |
| После сдвига | 19 | 103 |
| Шифр | С(рус) | р(рус) |

Шифровка специальных символов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Текст | пробел | , | ! | ? |
| Номер | 124 | 126 | 127 | 128 |
| После сдвига | 141 | 143 | 144 | 145 |
| Шифр | ` | ( | ) | [ |

Теперь соберем все коды Цезаря в одну строку и получаем результат ручного расчёта:

«A-щтх/(`го`дrгv`zи`Ср--рбкur)[»

Результат ручного расчета соответствует контрольному примеру.

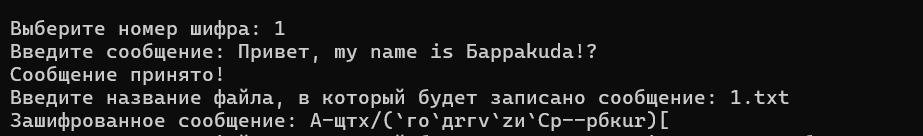


Рисунок 2.11 – Вывод контрольного примера шифра Цезаря

2.7.2 Шифровка с помощью шифра A1Z26

Текст – Привет, my name is Барраkuda!?

Подставляем вместо каждого символа его номер в нашем алфавите, и разделяем эти номера пробелами.

Шифровка русских символов.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Текст | П | р | и | в | е | т | Б | а |
| Шифр | 17 | 103 | 95 | 88 | 91 | 105 | 2 | 86 |

Шифровка английских символов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Текст | m | y | n | a | e | i |
| Шифр | 72 | 84 | 73 | 60 | 64 | 68 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Текст | s | k | u | d |
| Шифр | 78 | 70 | 80 | 63 |

Шифровка специальных символов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Текст | пробел | , | ! | ? |
| Шифр | 124 | 126 | 127 | 128 |

Получаем результат: «17 103 95 88 91 105 126 124 72 84 124 73 60 72 64 124 68 78 124 2 86 103 103 86 70 80 63 60 127 128»

Результат ручного расчета соответствует контрольному примеру.

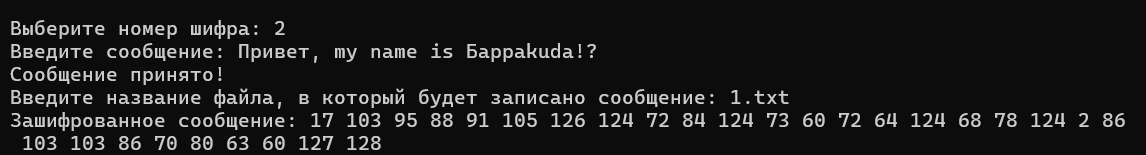


Рисунок 2.12 – Вывод контрольного примера шифра A1Z26

2.7.3 Высчитывание общего секретного ключа по алгоритму Диффи-Хеллмана

Секретный ключ участника A = 4;

Секретный ключ участника B = 6;

Для расчета у нас есть простое число p = 23 и число g = 5.

Вычисляем число A = ga mod p:

A = 54 mod 23 = 625 mod 23 = 4

Вычисляем число B = gb mod p:

B = 56 mod 23 = 15 625 mod 23 = 8

Вычисляем общий секретный ключ:

K1 = Ba mod p = 84 mod 23 = 4096 mod 23 = 2

K2 = Ab mod p = 46 mod 23 = 4096 mod 23 = 2

K1 = K2 – общие ключи совпадают. Общий секретный ключ - 2

Результат ручного расчета соответствует контрольному примеру.

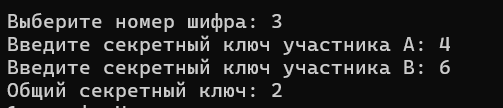


Рисунок 2.12 – Вывод контрольного примера алгоритма Диффи-Хеллмана

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Рябко Б.Я., Фионов А.Н. Криптографические методы защиты информации. 2-е издание, стереотип. – М.: Горячая линия – Телеком, 2012. – 229 c.
2. Павловская Т. А. C/C++. Программирование на языке высокого уровня: [для вузов] / Т.А.Павловская. - СПб. [и др.]: Питер, 2010. - 460 с.
3. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. — СПб.: Питер, 2013. — 464 с.
4. Ильин М.Е., Калинкина Т.И. Криптографическая защита информации в объектах информационной инфраструктуры. 46-е издание, Москва: Издательский центр «Академия», 2019. – 272 с.
5. Левина А.Б. Криптография и криптографические протоколы. Протоколы аутентификации. 2022. – 3 стр.
6. Страуструп Б. Программирование: принципы и практика с использованием С++, 2-е изд. : Пер. с англ. - М. : ООО "И.Д. Вильяме", 2016. - 1 328 с.
7. Мейерс С. Эффективное использование С++. — М.: ДМК Пресс, 2014. — 294 с.
8. Мейерс С. Эффективное использование STL. Библиотека программиста / С. Мейерс. СПб.: Питер, 2002. — 224 с.
9. Саттер Г., Александреску А. Стандарты программирования на С++. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2008. — 224 с.
10. Прата С. Язык программирования C++ (C++11). Лекции и упражнения, 6-е издание — М.: Вильямс, 2012. — 1248 с.

ПРИЛОЖЕНИЕ

Ссылка на GitHub: <https://github.com/WarpMiner/RGR>