МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра систем информационной безопасности

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой к.п.н., доцент Р. Ш. Бердибаев
_____ «____» _____2019 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Реализация элементов технологии криптографической защиты
Специальность 5В100200 - "Системы информационной безопасности"
Выполнил Талипов Мажит Махамбетулы Группа СИБ-15-2
Научный руководитель Турганбаев Ерик Сулейменович

Консультант:
по экономической части:
к.э.н., профессор Аренбаева Пс.Г.
(ученая степень, звание, Ф.И.О)
«07» июня 2019 г.
(подпись)

по безопасности жизнедеятельности:
Ст. преподователь Бондарев И.И
(ученая степень, звание, Ф.И.О)
«11» июня 2019 г.
(подпись)

по применению вычислительной техники:
д, ф-м. н., профессор ВАК, доцент Турганбаев Е.С.
(ученая степень, звание, Ф.И.О)
«5» июня 2019 г.
(подпись)

Нормоконтролер:
Ст. преподователь Макарова Л.Т.
(ученая степень, звание, Ф.И.О)
«11» июня 2019 г.
(подпись)

Рецензент:
К.т.н., ассистент-профессор Сейлова И.А.
(ученая степень, звание, Ф.И.О)
«07» 06 2019 г.
(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра систем информационной безопасности

Специальность 5В100200 - "Системы информационной безопасности"

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Талипову Мажиту Махамбетулы

Тема проекта: Реализация элементов технологии криптографической защиты

Утверждена приказом по университету № 124 от «26» Октября 2018 г.

Срок сдачи законченного проекта «12» июня 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): проект подразумевает разработку программы шифрования.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: дипломный проект включает в себя 4 глав, разделенных на под главы, каждая из которых освещает определенную тематику.

В первой главе дипломного проекта представлена общая теоритическая информация о криптосистемах.

Во второй главе подробно описывается разработка программы по сети Фейстеля.

В третьей главе приводится технико-экономическое обоснование проекта.

В четвертой главе рассматриваются необходимые условия для комфортной разработки программного обеспечения.

Перечень графического материала (с точным указанием обязательных чертежей):

1) схемы криптографических систем защиты;
2) таблицы сравнения оценки алгоритмов;
3) рисунки общих схем;

Основная рекомендуемая литература:

1) Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем. 2-е изд. -Спб.: Питер, 2011.

2) Е.Ж. Айтхожаева, С.Т. Тынымбаев, Аспекты аппаратного приведения по модулю в асимметричной криптографии: Вестник НАНРК №5, 2014. Алматы. с 88-93

3) Рябко Б.Я, Фионов А.И. Основы современной криптографии для специалистов в информационных технологиях. - М.: Научный мир, 2004.

4) Аманжолова К. Б., Алибаева С. А. Экономика предприятии телекоммуникации: Учебное пособие. - Алматы: АИЭС, 2003

Конструкции по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономика	Алибаева К.Г.	04.03-07.06	
Безопасность	Бекбасаров Ж.Ж.	04.03-07.06	
Вычислительная тех.	Гуламбаев Е.С	04.03.07.06	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Изучение принципов работы Кри	07.04.2019	
Изучение алгоритмов шифро	10.04.2019	
Изучение шифрования Сети Фей	16.04.2019	
Разработка приложений	18.04.2019	
Изучение подходящего язы	25.04.2019	
Изучение графического приложен	01.05.2019	
Изучение Qt	07.05.2019	
Тестирование приложен	14.05.2019	
Экономическая часть	16.05.2019	
Безопасность жизнедеятельн	12.05.2019	

Дата выдачи задания «04» Марта 2018 г.

Заведующий кафедрой _____ (_____)
(Подпись) (Ф.И.О)

Научный руководитель проекта _____ (Тулкамбаев Е.С.)
(Подпись) (Ф.И.О)

Задание принял к исполнению студент _____ (Талипов М.М.)
(Подпись) (Ф.И.О)

АҢДАТПА

Бұл дипломдық жобаның мақсаты – заманауи симметриялы криптографияның негіздерін, ақпаратты рұқсатсыз қол жеткізуден қорғау әдістері мен құралдарын зерттеу. Зерттеу нәтижесі симметриялық блокты шифрлаудың қазіргі заманғы бағдарламасын әзірлеу болып табылады. Бағдарлама іске асырылған шифрлау алгоритмі криптоанализге төзімділіктің белгілі бір деңгейіне ие болады. Бағдарлама web-технологиялардың көмегімен сипатталған пайдаланушының заманауи, қарапайым және түсінікті интерфейсі болады. Сонымен қатар, бағдарламаны пайдалану ыңғайлылығы үшін кез келген ДК-де кроссплатформ іске асырылады.

АННОТАЦИЯ

Цель данного дипломного проекта – это изучение основ современной симметричной криптографии, методов и средств защиты информации от несанкционированного доступа. Результатом изучения будет являться разработка современной программы симметричного блочного шифрования. Алгоритм шифрования реализованный в программе будет иметь определённый уровень стойкости к криптоанализу. Программа будет иметь современный, простой и понятный интерфейс пользователя, описанный с помощью web-технологий. Так же для удобства использования программы на любых ПК будет реализована кроссплатформенность.

ANNOTATION

The purpose of this diploma project is to study the basics of modern symmetric cryptography, methods and means of protecting information from unauthorized access. The result of the study will be the development of a modern program of symmetric block encryption. The encryption algorithm implemented in the program will have a certain level of resistance to cryptanalysis. The program will have a modern, simple and intuitive user interface, described using web-technologies. Also for the convenience of using the program on any PC will be implemented cross-platform.

Содержание

Введение

В современном мире компьютеры обычно играют все более важную роль в качестве электронных средств передачи, хранения и обработки информации. Чтобы использовать информационные технологии в различных областях, необходимо обеспечить их надежность и безопасность. Под безопасностью (в широком смысле) понимается способность информационной системы поддерживать свою целостность и работоспособность при случайных или преднамеренных внешних воздействиях. Таким образом, широкое использование информационных технологий привело к быстрому развитию различных методов защиты информации, в которых защита от помех кодирование и шифрование можно назвать основными методами.

Самые простые методы шифрования использовались в течение длительного времени, но развитие методов научных исследований и методов шифрования впервые появилось в прошлом (20-й век). На сегодняшний день шифрование включает в себя различные результаты (теоремы, алгоритмы), такие как основы и приложения. Шифрование невозможно без строгой математической подготовки. Требуются специальные знания в области дискретной математики, теории чисел, абстрактной алгебры, теории алгоритмов. Однако имейте в виду, что криптографические методы в основном используются в реальных приложениях.

Теоретически сильные алгоритмы могут подвергаться атакам, не предусмотренным математическими моделями. Целью данного дипломного проекта является изучение основ современной симметричной криптографии и защита информации от методов и средств несанкционированного доступа. Результатом этого исследования станет новейшая программа для разработки симметричного блочного шифрования. Алгоритм шифрования, реализованный в программе, несколько устойчив к дешифрованию. Программа будет иметь современный, простой и интуитивно понятный пользовательский интерфейс, написанный с использованием веб-технологий. Кроме того, он реализован на разных платформах для облегчения использования программы на любом ПК.

1. **Предмет криптографии**

Шифрование является важным инструментом для защиты информации в вычислительных системах. Он используется во всем мире и ежедневно используется миллиардами людей во всем мире. Он используется для защиты данных отдыха и данных упражнений. Система шифрования является частью стандартного протокола, в частности протокола TLS, и относительно легко реализовать сильное шифрование в различных приложениях.

Шифрование очень полезно, но очень уязвимо. Самая безопасная технология шифрования. Система может быть не совсем безопасной из-за одной спецификации или ошибки программирования. Существует не так много модульных тестов для обнаружения дыр в безопасности в криптографических системах.

Вместо этого, чтобы утверждать, что криптографическая система является безопасной, мы полагаемся на математическое моделирование и доказательства, чтобы доказать, что конкретная система соответствует назначенным ей атрибутам безопасности. Нам часто нужно делать некоторые разумные предположения для продвижения нашей теории безопасности.

Эта книга об этом: построение практической криптографической системы, мы можем требовать безопасности при разумных допущениях. Эта книга охватывает множество различных задач в области шифрования. Для каждой задачи определите конкретную цель безопасности, которую вы пытаетесь достичь, и продемонстрируйте схему, которая достигает этой цели. Чтобы проанализировать структуру, мы разработали унифицированную структуру для криптографических доказательств. Читатели, которые освоили эту структуру, могут применить ее к новым структурам, которые эта книга не может решить.

Эта книга показывает пример того, как много систем развертывания работают. Описывает типичные ошибки, которых следует избегать, и атаки на реальные системы.

Важность строгости в криптографии Завершает каждую главу забавным приложением и неожиданно применяет идеи из этой главы.

Предположим, что у Алисы и Боба есть общий ключ k, и Алиса пытается отправить сообщение m Бобу по сети, поддерживая конфиденциальность при наличии перехвата с другой стороны. В этой главе мы начнем разрабатывать базовый подход к решению этой проблемы. В дополнение к отправке сообщений по сети, эти же методы позволяют Алисе сохранять файлы на диск, чтобы те, кто не имеет доступа к диску, могли читать файлы, но Алиса Вы можете прочитать файл самостоятельно позже.

Методы, которые мы разработали для решения этой основной проблемы, важны и интересны, но следует подчеркнуть, что они не решают всех проблем, связанных с ними.

"

Секрет в том, чтобы обеспечить конфиденциальность, только если Алиса отправит одно сообщение. Для ключа. Если Алиса хочет тайно отправить несколько сообщений, используя один и тот же ключ, ей следует использовать этот метод.

Эти методы не гарантируют целостность сообщения. Если злоумышленник может изменить бит зашифрованного текста при переходе от Алисы к Бобу, Боб не узнает, что это произошло, и примет сообщение, отличное от сообщения, отправленного Алисой. Описывает гарантию целостности сообщения.

Этот метод не предоставляет механизм, который позволяет Алисе и Бобу делиться.

Секретный ключ отображается первым. Возможно, они могут сделать это в какой-то момент через какую-то безопасную сеть (или физическую конференцию лицом к лицу), и сообщения отправляются позже, чем Алиса и Боб должны общаться через незащищенную сеть. вы. Однако существуют также протоколы, которые позволяют Алисе и Бобу обмениваться секретными ключами даже в незащищенных сетях, если используется соответствующая инфраструктура.

Существуют и другие угрозы для информации, защищаемой нелегальными пользователями: подстановка, имитация и т. Д., Которые описаны ниже. Эта ситуация может быть представлена следующей диаграммой. На рисунке1.1.

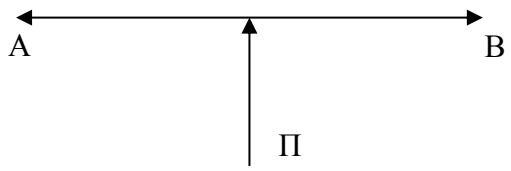


Рисунок 1.1 – Угроза разглашения информации

Здесь A и B являются удаленными законными пользователями защищенной информации, они хотят обмениваться информацией через общедоступные каналы связи, P является нелегальным пользователем (оппозиция), и это формальное решение Может рассматриваться как типичная модель случая с использованием методов криптографической защиты информации. Исторически некоторые военные термины были дополнены криптографией («враг», «пароль атаки» и т. Д.).

Основной механизм шифрования сообщений с использованием общего ключа называется паролем (или методом шифрования). В этом разделе

представлена немного упрощенная концепция шифрования. Это называется паролем Шеннона. Пароль Шеннона - это пара функций E = (E, D).

Функция E (функция шифрования) получает ключ k и сообщение m (также называемое простым текстом) в качестве входных данных и выводит зашифрованный текст c. Этос = E (к, м) (1)

и мы говорим, что с шифрование м под к.

– Функция D (функция дешифрования) принимает в качестве ввода ключ k и зашифрованный текст c, и выдает сообщение м. То есть,

$$m = D (k, c), \quad (1.1)$$

и мы говорим, что m дешифрование с под к.

– Мы требуем, чтобы дешифрование «отменяло» шифрование; то есть шифр должен удовлетворять следующему свойство корректности: для всех ключей k и всех сообщений m имеем

$$D (k, E (k, m)) = m. \quad (1.2)$$

Чтобы быть немного более формальным, давайте предположим, что K - это набор всех ключей (пространство ключей), M – это набор всех сообщений (пространство сообщений), и что C является набором всех зашифрованных текстов (зашифрованный текст пространство). C помощью этой записи мы можем написать:

$$E: K \rightarrow M! C, \quad (1.3)$$

$$D: K \rightarrow C! M. \quad (1.4)$$

Более того, можно сказать, что E определяется как (K, M, C).

Предположим, что Алиса и Боб хотят использовать такой пароль, чтобы Алиса могла отправить сообщение Бобу. Идея состоит в том, что Алиса и Боб должны заранее договориться о ключе k 2K. E (k, m) 2 C и отправить его Бобу через определенную сеть связи. Боб, который получает с расшифровку с под k и атрибут корректности, гарантирует, что D (k, c) идентична исходному сообщению Алисы. Конечно, цель интуитивно понятна, чтобы показать, что подслушивающие, которые могут получать с в командировке, мало знают о послании Алисы - это интуитивная концепция, которая принимает формальные определения безопасности.

На самом деле ключи, сообщения и зашифрованный текст обычно являются байтовыми последовательностями. Ключи обычно имеют фиксированную длину. Например, 16-байтовый (то есть 128-битный) ключ очень распространен. Сообщения и зашифрованный текст могут представлять собой последовательность байтов фиксированной или переменной длины. Например, сообщение может представлять собой видеофайл размером 1 ГБ,

музыкальный файл объемом 10 МБ, сообщение электронной почты объемом 1 КБ или один голос или голосование за электронные выборы.

Другими словами, законные пользователи должны учитывать это в своей стратегии защиты. Если есть очевидные недостатки, нет смысла делать ссылки очень мощными («тот же принцип силы защиты»). Еще одна важная проблема, о которой следует помнить, это соотношение цены и цены, затраты на защиту и затраты на добычу. На уровне развития современных технологий развитие средств коммуникации само по себе, а средства перехвата информации и средства защиты информации очень дороги.

Прежде чем защищать свою информацию, пожалуйста, задайте следующие два вопроса.

Превышает ли ценность врага стоимость атаки.

Превышает ли ваша ценность стоимость защиты.

Эти соображения имеют решающее значение при выборе подходящих средств защиты (физическая, стеганографическая, шифрование и т. Д.). Некоторые понятия криптографии объясняются на исторических примерах, поэтому я приведу некоторые исторические объяснения. В течение долгого времени, использование шифрования было много странных компаньонов [6].

Среди них талантливые ученые, дипломаты и священники. Даже шифрование можно рассматривать с помощью черной магии. Криптография как период развития искусства длилась с прошлого до начала 20 века, пока не появились первые машины шифрования.

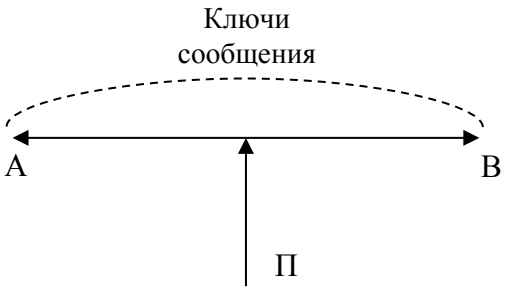
После работы известного американского ученого К. Шеннона он начал понимать математическую природу криптографии для решения проблем в середине 20-го века. Многие известные исторические личности оставили свой след в истории криптографии.

Первое сообщение об использовании криптографии в вооруженных силах касается имени спартанского командира Лизандера, Scytal Cipher. Этот кодекс известен с тех пор, как Спарта сражалась против Афин в 5 веке до нашей эры. Для реализации используется стержень, имеющий цилиндрическую форму. Тонкая полоса папируса (без зазоров или внахлест) оборачивается вокруг катушки от катушки, и текст записывается на ленту вдоль оси седалищной кости.

Лента была расширена, и (для начинающих) было обнаружено, что некоторые буквы были написаны не по порядку на другой стороне ленты. Затем лента отправляется получателю. Получатель читает сообщение по центру и таким же образом оборачивает полученную ленту. В пароле преобразование из открытого текста в шифрование включает в себя определенное расположение символов открытого текста. Следовательно, криптографический класс, на который ссылается пароль Scytal, называется заменяющим шифром. В ответ Цезарь использовал исторически исчезший пароль в качестве шифра Цезаря.

Этот пароль выполняет следующее преобразование открытого текста: каждая буква в открытом тексте заменяется третьей буквой в алфавите, которая считается кружком. Буква «я» сопровождается буквой «а». Цезарь заменил третью букву после нее, но вы можете заменить другие. Главное, что человек, отправляющий зашифрованное сообщение, знает значение передачи. Криптографический класс, к которому относится криптография Цезаря, называется замещающей криптографией. Поэтому тип криптографии, также известный как криптография Stsital, называется замещающей криптографией.

Пароль Цезаря Cipher реализует следующее преобразование открытого текста: каждый символ в открытом тексте заменяется третьим символом после алфавита. Это считается написанным по кругу. Обратите внимание, что



Цезарь заменил третью букву.

Рисунок 1.2 – Формальное описание криптографии

Теперь мы замечаем, что не существует единого пароля для каждой ситуации. Выбор метода шифрования зависит от характера информации, ее ценности и способности владельца защитить эту информацию. Во-первых, выделите различные виды защищенной информации, такие как документальные фильмы, телефоны, телевизоры и компьютеры.Каждый тип информации имеет уникальные функции, которые значительно влияют на выбор способа шифрования информации. Большое значение имеет объем и скорость передачи зашифрованной информации. Выбор типов паролей A, B и P-ключей и их параметров [P, A] зависит главным образом от характера защищенного секрета или секретов. Некоторые секреты (например, государственные, военные и т. Д.) Должны храниться десятилетиями, а некоторые секреты (например, торговля) могут быть раскрыты в течение нескольких часов.

Вы также должны учитывать способность человека защитить эту информацию. Это одна из вещей, чтобы противостоять одиночеству и даже жестоким. Еще одна сильная национальная структура. Способность пароля противостоять различным атакам называется надежностью пароля.

Атакуя пароль, они понимают попытку открыть этот пароль. Понятие надежности пароля является центральным для шифрования. Хотя это легко понять качественно, получение точных и доказуемых оценок надежности для каждого конкретного пароля является открытой проблемой.

Это можно объяснить тем, что пока нет математических результатов, необходимых для решения этой проблемы. Следовательно, надежность конкретного пароля может быть открыта только различными попытками и зависит от того, может ли криптоанализатор атаковать пароль.

Этот процесс также называется испытанием на долговечность. Важным подготовительным шагом к проверке надежности пароля является рассмотрение возможности различных предположений о том, что злоумышленник может атаковать пароль. Появление таких способностей у противника обычно не зависит от шифрования, что является некоторыми внешними подсказками, которые существенно влияют на надежность пароля.

Поэтому оценки надежности пароля всегда включают в себя предположения о целях и способностях противника, и эти оценки получены в соответствии с этими предположениями. Во-первых, как упоминалось выше, обычно считается, что оппонент сам знает пароль и имеет возможность провести предварительный опрос.

Оппонент также знает некоторые особенности открытого текста, такие как общая тема, стиль, стандарт и формат сообщения. Более конкретно, мы приводим три примера способностей противника:

-Враг может перехватить все зашифрованные сообщения, но не соответствующий открытый текст.

-Враг может перехватить все зашифрованные сообщения и извлечь соответствующий открытый текст.

- Злоумышленник может получить доступ к паролю (не может получить доступ к ключу), но может зашифровать и расшифровать любую информацию [7].

На протяжении веков споры о надежности паролей и возможности создания абсолютно надежных паролей не исчезали среди экспертов. Этот вопрос будет возвращен позже. В конце этого раздела добавьте еще одно предложение о термине. В последнее время слово «шифрование» часто используется в слове «шифрование», но их взаимосвязь не всегда понимается правильно. Окончательное формирование этих научных областей в настоящее время ведется, и их темы и проблемы проясняются.

1.2 **Основные задачи**

Шифрование родилось из науки о методах шифрования и долгое время было зашифровано (то есть защищало передаваемые или хранимые данные от

несанкционированного чтения). Однако в последнее время с быстрым развитием информационных технологий появилось много новых приложений, которые не имеют прямого отношения к сокрытию секретной информации.

Необходимость использования методов шифрования вытекает из условий хранения и обмена информацией. В современных информационных системах данные обычно обмениваются группами, члены которых не доверяют друг другу.

Примеры включают подписание контрактов и других документов, финансовые сделки и совместные решения. В этом случае необходимо принять меры, чтобы гарантировать, что информация не будет искажена или полностью заменена в процессе обмена или хранения.

Эта гарантия может быть достигнута только с помощью научных методов шифрования. Поэтому целью использования шифрования является защита информационной системы от вражеских целей и разрушительных воздействий (атак). Средства защиты существенно зависят от конкретной ситуации. Враги возможны в зависимости от типа угрозы, которую вам нужно защищать. Основное назначение шифрования

Обеспечение конфиденциальности данных (предотвращение несанкционированного доступа к данным). Это одна из основных задач шифрования, и для ее решения используется шифрование данных. Выполняя такое преобразование, только законные пользователи с соответствующим ключом могут их прочитать.

Обеспечение целостности данных гарантирует, что неавторизованные пользователи не будут изменять данные во время передачи или хранения. Модификация относится к вставке, удалению или замене информации, а также к повторной передаче ранее перехваченного текста.

Обеспечение аутентификации - это аутентификация субъекта (стороны, обменивающейся данными, автора документа и т. Д.) Или самой информации. Частным случаем аутентификации является идентификация - процедура, используемая для доказательства того, что это действительно тот факт, о котором говорится. Во многих случаях субъект X должен не только доказать свои права, но и предотвратить использование в будущем информации, полученной субъектом проверки (Y), для представления себя в качестве X. Такие доказательства называются «доказательствами неразглашения».

Обеспечение невозможности отклонения автора - не позволяет субъекту отвергнуть совершенное им поведение (как правило, возможность отказаться от подписания документов). Эта работа неотделима от двойственности - для обеспечения невозможности авторской атрибуции. Самый очевидный пример работы - это контракт между двумя или более людьми, которые не доверяют друг другу. В этом случае все будущие подписанты не смогут сначала отказаться от своих подписей, а затем никто не может изменять, заменять или создавать новые документы. Утверждают, что этот конкретный файл был подписан.

Основным способом решения этой проблемы является использование цифровой подписи. В дополнение к основным задачам, указанным выше, вам необходимо вызывать больше вещей, таких как электронное голосование, подбрасывание, секретный обмен (обмен секретной информацией между несколькими темами, чтобы их можно было использовать только вместе).

1.3 **Теория информации и ее ключевые понятия**

Слово «информация» происходит от латинского слова «informatio», что означает прояснение сознания. Эта концепция является одной из ключевых концепций кибернетики и понимается как любой сбор информации или данных.

В то же время, эта концепция относится к базовой исходной концепции общего предельного уровня, и, как и многие из этих концепций, не существует общепринятого строгого научного определения.

Современная наука о природе информационных атрибутов и информационных процессов называется теорией информации. Содержание понятия «информация» можно раскрыть на двух примерах первого метода измерения количества информации в истории: методов Хартли и Шеннона. Первый основан на теории множеств и комбинаторике, а второй - на теории вероятностей.

Основа всей теории информации была открыта Р. Хартли в 1928 году, и информация является количественной. 1948, К. Шеннон дает полное и полное представление о теории. Русский ученый А.Н. внес большой вклад в дальнейшее развитие и универсализацию теории информации. Колмогоров А.А. Харкевич Р.Л. Стратанович.

Недавно немецкие исследователи из Советского архива сообщили, что эта теория, называемая сегодня теорией Шеннона, была создана А.Н. Колмогоров еще в 1938 году, но был классифицирован как используемый для военной разработки.

Подход Р. Хартли основан на очень базовой теории множеств, которая является основой комбинации, а также на нескольких интуитивно понятных и очень очевидных предположениях.

Рассмотрим эти предположения. Мы предполагаем, что если имеется много элементов и один из них выбран, сообщается или генерируется определенное количество информации. Сообщение заключается в том, что если вы не знаете, какой элемент будет выбран до выбора, он станет известен после выбора.

Давайте найдем форму функции, которая связывает объем информации, получаемой при выборе элемента из коллекции, с количеством элементов в этой коллекции, то есть по его мощности.

Если набор элементов, делающих выбор, состоит из одного элемента, то очевидно, что его выбор предопределен, т. Е. В выборе нет неопределенности. Поэтому, если мы обнаружим, что выбрали этот единственный элемент, то,

очевидно, мы не получим никакой новой информации, то есть мы получим нулевую информацию.

Если коллекция состоит из двух элементов, неопределенность выбора минимальна. В этом случае минимальное значение - это также объем информации, которую мы получаем, когда знаем, какой из элементов выбран.

Чем больше элементов в коллекции, тем больше неопределенность выбора и тем больше информации мы получаем, когда знаем, какой элемент мы выбираем.

Исходя из этих очевидных соображений, первое требование заключается в следующем: информация является монотонной функцией мощности исходного набора.

Рассмотрим набор чисел в двоичной системе счисления, длина которой является двоичным числом. В этом случае каждое число может принимать только значения 0 и 1, как показано в таблице 1.1.

Таблица 1.1 — К выводу формулы количества информации по Р.Хартли

Количество двоичных разрядов (i)	Количество состояний, которое можно пронумеровать i-разрядными двоичными числами (N)	Основание системы счисления		
		0	6	2
1	2	1	1	1
2	4	1 2 3	1 2 3	0 01 10 11
3	8	1 2 3 4 5 6 7	1 2 3 4 5 6 7	00 001 010 011 100 101 110 111

Продолжение таблицы 1.1

4	16	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	0 1 2 3 4 5 6 7 8 9 A B C D E F	0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
...	...			
i	2^i			

Очевидно, количество этих чисел (элементов) в множестве равно:

$$N = 2^i \qquad (1.1)$$

Рассмотрим процесс выбора чисел из рассмотренного множества. До выбора вероятность выбрать любое число одинакова. Существует объективная неопределенность в вопросе о том, какое число будет выбрано. Эта неопределенность тем больше, чем больше N – количество чисел в множестве, а чисел тем больше – чем больше разрядность i этих чисел.

Примем, что выбор одного числа дает нам следующее количество информации:

$$i = \mathrm{Log}_2(N). \qquad (1.2)$$

Таким образом, количество информации, содержащейся в двоичном числе, равно количеству двоичных разрядов в этом числе.

Это выражение и представляет собой формулу Хартли для количества информации. Отметим, что оно полностью совпадает с выражением для энтропии (по Эшби), которая рассматривалась им как количественная мера степени неопределенности состояния системы.

При увеличении длины числа в два раза количество информации в нем также должно возрасти в два раза, несмотря на то, что количество чисел в множестве возрастает при этом по показательному закону (в квадрате, если числа двоичные), т.е. если

$$N_2 = (N_1)^2 \quad (1.3)$$

то

$$I_2 = 2 \cdot I_1 \quad (1.4)$$

$$F(N_1 \cdot N_1) = F(N_1) + F(N_1). \quad (1.5)$$

Это невозможно, если количество информации выражается линейной функцией от количества элементов в множестве. Но известна функция, обладающая именно таким свойством: это Log:

$$Log_2(N_2) = Log_2(N_1)^2 = 2 \cdot Log_2(N_1). \quad (1.6)$$

Это второе требование называется требованием аддитивности . Таким образом, логарифмическая мера информации, предложенная Хартли, одновременно удовлетворяет условиям монотонности и аддитивности. Сам Хартли пришел к своей мере на основе эвристических соображений, подобных только что изложенным, но в настоящее время строго доказано, что логарифмическая мера для количества информации однозначно следует из этих двух постулированных им условий.

Минимальное количество информации получается при выборе одного из двух равновероятных вариантов. Это количество информации принято за единицу измерения и называется «бит».

Подход К. Шеннона. Клод Шеннон основывается на теоретико – вероятностном подходе. Это связано с тем, что исторически шенноновская теория информации выросла из потребностей теории связи, имеющей дело со статистическими характеристиками передаваемых сообщений и каналов связи.

Пусть существует некоторое конечное множество событий (состояний системы): $X = \{x_1, x_2, ..., x_N\}$, которые могут наступать с вероятностями: $p(x_i)$, соответственно, причем множество вероятностей удовлетворяет естественному условию нормировки:

$$\sum p(x_i) = 1 \quad (1.7)$$

Исходное множество событий характеризуется некоторой неопределенностью, т.е. энтропией Хартли, зависящей, как мы видели выше, только от мощности множества. Но Шеннон обобщает это понятие, учитывая, что различные события в общем случае не равновероятны. Например, неопределенность системы событий: {монета упала «орлом», монета упала «решкой»}, значительно выше, чем неопределенность событий: {монета упала «орлом», монета упала «ребром»}, так как в первом случае варианты

равновероятны, а во втором случае вероятности вариантов сильно отличаются:

$$H(X) = -\sum p(x_i)\cdot Log_2(p(x_i)) \quad (1.8)$$

Если измерять количество информации изменением степени неопределенности, то шенноновское количество информации численно совпадает с энтропией исходного множества:

$$I(X) = -\sum p(x_i)\cdot Log_2(p(x_i)) \quad (1.9)$$

Следуя , приведем вывод выражения Шеннона (1.9) непосредственно из выражения Хартли для количества информации: $I = Log_2(N)$.

Пусть события исходного множества мощности N равновероятны:

$$p(x_i) = 1/N, \quad (1.10)$$

тогда учитывая, что

$$Log(1/N) = Log(1)\cdot Log(N) = Log(N), \quad (1.11)$$

$$\sum 1/N = 1 \quad (1.12)$$

непосредственно из формулы Хартли получаем

$$I(X) = -\sum 1/N\cdot Log_2(1/N) = -\sum p(x_i)\cdot Log_2(p(x_i)). \quad (1.13)$$

До сих пор предполагается, что это выражение также верно для событий, где событие не равно вероятности. Эта гипотеза включает в себя обобщение Клода Шеннона, который составляет всю эпоху развития современной теории информации.

Очень важный и основополагающий факт заключается в том, что при построении метрики Хартли используется только понятие «разнообразие», которое накладывает только одно условие (ограничение) на элементы оригинальной коллекции: оно должно уметь отличать эти элементы друг от друга.

Теория Шеннона в основном использует статистику и предполагает, что случайные события (состояния системы) распределяются по обычным правилам.

Следовательно, различие между методами Хартли и Шеннона и построением теории информации соответствует разнице между непараметрическими и параметрическими методами в статистике.

Более конкретно, ясно, что до тех пор, пока вероятность всех событий (состояний) равна, мера Шеннона постепенно входит в измерение Хартли.

Статистика доказывает основные свойства энтропии случайного процесса. Характерной особенностью является то, что при нормальном распределении и достаточно больших условиях выборки весь набор событий можно разделить на две категории:

Событие, которое крайне вероятно может произойти (считается заслуживающим изучения);

Событие, которое вряд ли произойдет (считается не заслуживающим особого внимания).

Кроме того, одинаково возможно иметь события высокой вероятности с высокой точностью. По мере увеличения размера выборки доля «заметных» событий уменьшается без ограничений, и метрика Шеннона асимптотически преобразуется в метрику Хартли.

Поэтому можно предположить, что для больших выборок нормального распределения метрика Хартли является разумным упрощением метрики Шеннона.

1.4 **Формальная модель и классификация шифров**

Наука Шифрования разработала алгоритмы, которые обеспечивают шифрование и аутентификацию. Некоторые из этих алгоритмов могут быть официально проверены, а другие алгоритмы использовались в течение десятилетий, не выявляя серьезных недостатков. Из этих строительных блоков протоколы безопасности могут достигать таких целей, как обмен секретными ключами и справедливость, и их безопасность также может быть продемонстрирована с использованием правильности криптографических алгоритмов. Но все эти соображения касаются только уровня математики. В системах безопасности встроенные алгоритмы создаются и улучшаются в программном и аппаратном обеспечении, вводя много новых аспектов и побочных эффектов, не охватываемых математической моделью абстракции. Например, считается, что чип-карта защищена, потому что аппаратное обеспечение недоступно. Но на самом деле очень успешная атака на эти чипы путем непосредственного измерения их энергопотребления зависит от его расчетов. Также успешные методы включают измерение времени работы электромагнитного излучения и дистанционную регистрацию, что позволяет восстановить работу монитора и клавиатуры в течение нескольких метров. Третий пример - скрытый канал, который может передавать точную информацию злоумышленнику.

Многие системы шифрования предполагают открытый текст, а зашифрованный текст и ключи являются целыми числами. Поскольку числовые функции хорошо изучены, это предположение облегчает построение и демонстрацию алгоритмов шифрования и дешифрования. Однако это не ограничивает область применения таких алгоритмов, поскольку любой текст, написанный с использованием букв (например, русских букв), всегда можно

представить в виде целого числа. Как правило, для этой цели каждый символ алфавита кодируется набором 0 и 1 (например, в соответствии с таблицей ASCII), а текст представляется в виде последовательности кодов соответствующих символов, записанных один за другим. Результирующий ноль и одна последовательность представляют собой цифровое представление текста.

В каждой распределенной системе должно быть что-то, что отличает законного получателя от всех других участников. В системе шифрования эта функция представляет собой понимание определенного секрета. В прошлом люди обычно договаривались о нераскрытом алгоритме.

Хочешь общаться безопасно. Даже сегодня это случается время от времени. Но у этого подхода есть несколько недостатков:

 Большинство алгоритмов легко взломать, потому что очень сложно создать действительно безопасный алгоритм;

 Если вы нанимаете стороннюю компанию (например, криптографа) для выполнения этой работы, то он сталкивается с потенциальными рисками. Из-за этого многие дееспособные криптографы «случайно» исчезли;

Невозможно тщательно проанализировать алгоритм, потому что каждый, кто сможет это сделать, наверняка постигнет та же участь, что и в предыдущем абзаце;

 По мере увеличения числа участников становится очень сложно придумать достаточное количество алгоритмов. Мы видим, что лучше всего использовать только один тип публичных записей и доступных алгоритмов, поэтому его можно тщательно изучить, стандартизировать и внедрить в массы.

Однако такой алгоритм должен иметь некоторый параметр (называемый «ключом»), чтобы обеспечить уникальную особенность, которая отличает участников.

Эти ключи должны иметь некоторые важные атрибуты:

Creation их создание должно основываться на реальных случайных числах;

Number Количество всех возможных разных ключей должно быть намного больше, чем количество участников (фактор 2 100 не слишком много). Это предотвращает случайное совпадение двух ключей и сводит к минимуму возможность правильного угадывания;

Key Ключевые отношения с владельцем не могут быть защищены каким-либо образом, но должны быть проверены лично. Используя открытый алгоритм, каждый может создать ключ с любым именем;

Основной принцип заключается в том, что вся система не будет хуже (или хуже), чем этот исходный ключ.

1.5 **Симметричные системы шифрования**

До появления асимметричных систем в 1976 году вся криптография была симметричной. Их история насчитывает тысячи лет; Одним из самых

ранних известных паролей является планшет Цезаря, который работает путем преобразования простых букв в заранее определенное количество символов. Наиболее распространенной системой, используемой в настоящее время, является система одноразовой маскировки заполнения (также известная как Vernam Chiffre) и коллег, код аутентификации, вездесущий DES (Data En-

Стандарт шифрования) и его указанный преемник AES (Advanced Encryption Standard). Подробные инструкции можно найти по адресу:

Легенды и символы

X - группа всех текстовых сообщений

C - коллекция всего зашифрованного текста

S - набор всех подписей

K - набор всех симметричных ключей

kAB∈K - это определенный ключ, принадлежащий A и B

Симметричные системы характеризуются выполнением шифрования и дешифрования или подписью и тестированием с использованием одного и того же ключа соответственно.

камуфляж

алгоритм:

Шифрование: $X \times K \rightarrow C$

сертификация

алгоритм:

Символ: $X \times K \rightarrow S$

Расшифровка: $C \times K \rightarrow X$

Симметрия и условия впрыска:

∈ k∈K, x∈X. Расшифровать (encrypt (x, k), k) = x

Отправить зашифрованные сообщения от A до Б:

 Шифрование: A выбирает сообщение x∈X и вычисляет c = crypt (x, k AB)

Передача: отправлено с настоящего момента получателю (возможно, также наблюдателям и злоумышленникам)

 Расшифровка: поскольку - кроме A - только B знает k AB, только он может вычислить x = расшифровать (c, k AB)

Отправить подписанное сообщение от A до Б:

IgnПодпись: A выбирает сообщение x∈X и вычисляет s = знак (x, k AB)

 перевод: x; s теперь отправляется получателю (и, возможно, злоумышленнику)

Поступление: B получил сообщение x; с хакером или оригиналом или модифицированным)

 Тест: B теперь вычисляет s = знак (x, k AB); если s = s, сообщение действительно. Поскольку - кроме A - только B знает k AB, никто не может изменить сообщение на x и подделать прямую подпись.

Прежде чем два участника смогут использовать эти алгоритмы, они должны договориться об общем симметричном ключе.

Используйте между ними. Если участники - два человека и имеют возможность встретиться друг с другом, чтобы заменить дискету или CD-ROM, то этот процесс не важен. Если участники не могут встретиться, будут проблемы. Они могут не знать друг друга лично, жизнь слишком далека, или участники просто машины. Тогда для сторонней поддержки требуется сторона, которой доверяют обе стороны. Обмен должен быть:

 Секрет: только две стороны (возможно, доверенная третья сторона) могут знать ключевые значения;

AlReal: протокол должен гарантировать, что ключ не может быть изменен во время процесса или после получения ключа, и участник должен проверить равенство впоследствии.

Соглашение для выполнения этой задачи - протокол Needham-Schroeder-Secret-Key (NSSK). Это также решает критическую проблему взрыва: в системе с n участниками, в худшем случае, вам нужно не более n (n-1) ключей (если каждый хочет общаться с другими и использовать его один раз). Панель пола), затем количество ключей увеличится до квадрата. Открытый текст записывает геометрию (обычно прямоугольник) вдоль траектории, а затем записывает символы из траектории по различным траекториям для получения зашифрованного текста. один пример. Мы записываем фразу «Это устройство маршрутизации» в прямоугольный стол 3 x 9, перемещаясь слева направо, перемещаясь слева направо, как показано на рисунке 1.1.3.

э	т	о	м	а	р	ш	р	у
т	н	а	я	п	е	р	е	с
т	а	н	о	в	к	а		

Рисунок 1.3 – Пример маршрутной перестановки

Первая статья о таких системах была опубликована в 1976 году Диффи и Хеллманом. Они описывают протокол, который позволяет двум участникам специально получать общий ключ от него.

публичная информация. В 1978 году Рональд Ривест, Ади Шамир и Леонард Адлеман опубликовали еще одну статью, описывающую знаменитый алгоритм RSA. Все асимметричные системы основаны на математических «односторонних» функциях, то есть операции могут быть легко выполнены в одном направлении, но их обратные (гипотетические) вычисления фактически неразрешимы.

Фактически, только две факторизации были использованы и хорошо изучены: легко получить два разных больших простых числа p и q и вычислить их произведение n = p • q. Однако, несмотря на столетия исследований, не известен эффективный алгоритм для вычисления двух простых факторов n. Дискретный логарифм. Легко взять показатель степени в конечном поле, то есть с учетом двух чисел a, x и простого p легко вычислить

y = a x (mod p). С другой стороны, однако, нет известного алгоритма, и данные y, a и p, x могут быть эффективно определены.

Но следует помнить, что эти два факта только полностью предположены и не могут быть доказаны или опровергнуты. Эти предположения могут быть использованы для построения криптосистемы с открытым ключом. Там каждый новый участник генерирует два ключа: секретный ключ, который не следует никому сообщать, и открытый ключ, который можно и нужно распространять как можно шире. Открытый ключ может быть использован для шифрования сообщения или проверки его подлинности и может быть расшифрован или подписан только с использованием закрытого ключа. Появление такой системы можно рассматривать как прорыв в «массовой криптографии», потому что почти все проблемы с симметричным обменом ключами (но проверка правильности отношения ключ-владелец) исчезли.

Первой программой с открытым исходным кодом с открытым исходным кодом для асимметричного шифрования был Фил Зим, «Pretty Good Privacy» (PGP) от Mermann, чья новая версия, к сожалению, была коммерциализирована сегодня. Партнером с открытым исходным кодом является GNU Privacy Guard (GPG), который использует PGP и открытый исходный код, поэтому любой может проверить его точность.

Если шифрование с открытым ключом имеет много преимуществ, почему бы не забыть эти «старые» символы? Метрическая система? К сожалению, асимметричные системы также имеют некоторые недостатки, которые до сих пор (и всегда) доказывают существование симметричных систем:

Они требуют много математических расчетов, примерно на 10 3 10 5 больше, чем симметричные системы. Поэтому они не подходят для небольших встроенных систем с очень ограниченными вычислениями памяти, скорости и / или мощности;

Обладая достаточной вычислительной мощностью, они легко взламываются и даже не перехватывают «реальное» сообщение: поскольку злоумышленник знает открытый ключ, он может зашифровать столько сообщений, сколько необходимо, и ищет секрет для его расшифровки. ключ. Аналогичный подход применяется к аутентификации;

Истина предположения, использованного □, не подтвердилась. Напротив, разработка новых и более быстрых алгоритмов намного быстрее, чем развитие вычислительной мощности.

Как правило, хеш-функция отображает вход любой длины на выход фиксированной длины. В криптографии приложение - это вычисление, необходимое для сокращения цифровых подписей: вместо подписи всего сообщения (если вам нужно подписать двоичный файл, такой как видео или программа, это может быть несколько мегабайт или даже ГБ). Сначала вычислите хэш-код сообщения и затем подпишите этот хэш-код.

Хорошо известной простой хеш-функцией является бит четности и сумма CRC. Они идеальны, если им нужно только обнаружить случайное

повреждение из-за технических недостатков или шумных каналов связи. Но поскольку сообщение легко изменить, оно имеет то же значение четности / CRC, что и исходное сообщение, поэтому его нельзя использовать для шифрования. Хеш-функции имеют свойство находить два сообщения, которые создают один и тот же хеш-код, поэтому сложнее найти другое сообщение для заданного хеш-значения, которое называется «антиколлизия».

Из-за необходимости понять этот метод для использования определенных протоколов, эта идея кратко описана. Как упоминалось ранее, это позволяет двум участникам вычислять открытые симметричные секретные ключи только из общедоступной информации и, разумеется, на основе своих собственных знаний. Алгоритм использует дискретные логарифмические допущения и экспоненциальный обмен.

Все расчеты сделаны в последнем поле. Модуль p и примитивный элемент a хорошо известны и могут быть одинаковыми для всех участников. (Оригинальный элемент - это элемент, который генерирует все поле, непрерывно увеличивая мощность.)Определить $P = C = K = (Z\ 26\ )\ m$.
Для ключа $K = (k\ 1\ ,\ k\ 2\ ,\ldots,\ k\ m\ )$ определим

$$e\ K\ (x\ 1\ ,\ x\ 2\ ,\ldots,\ x\ m\ ) = (x\ 1\ +\ k\ 1\ ,\ x\ 2\ +\ k\ 2\ ,\ldots,\ x\ m\ +\ k\ m\ ) \quad (1.14)$$

а также

$$d\ K\ (y\ 1\ ,\ y\ 2\ ,\ldots,\ y\ m\ ) = (y\ 1\ -k\ 1\ ,\ y\ 2\ -k\ 2\ ,\ldots,\ y\ m\ –k\ m\ ) \quad (1.15)$$

где все операции выполняются в Z 26 ».

Алгоритм блочного шифрования представляет собой шифр с симметричным ключом. Шифрование с симметричным ключом - это шифрование на основе ключей, в котором используется тот же ключ, который используется для шифрования конфиденциальных данных, используемых для расшифровки конфиденциальных данных. Недостаток использования шифрования с симметричным ключом состоит в том, что для получения ключа требуется только тот, кто хочет взломать шифрование. Поэтому ключ должен быть защищен и защищен. Ключ часто называют ключом.

Процесс отправки защищенной передачи с использованием симметричного алгоритма становится утомительным, потому что вы не можете позволить ключу быть распознанным. Вам необходимо безопасно получить ключ для получателя устройства безопасности, например, доставить ключ вручную. Как правило, метод шифрования с открытым ключом используется для передачи ключа, сгенерированного алгоритмом симметричного ключа. Когда получатель владеет ключом, вы можете зашифровать сообщение, а получатель может расшифровать сообщение, используя ключ, предоставивший ключ. Когда они хотят ответить, они могут использовать тот же ключ для шифрования своих сообщений, а затем вы можете использовать ключ для расшифровки сообщения.

Многие алгоритмы симметричного ключа используют конструкцию, называемую полем замены, также называемую S-блоком. Целью S-блока является защита алгоритма от линейного и дифференциального криптоанализа путем скрытия зашифрованного текста от простого текста, что также увеличивает диффузионные характеристики алгоритма. Под диффузией понимается изменение количества выходных битов при изменении входного бита. Алгоритм симметричного ключа делится на алгоритм блочного и потокового шифра. Алгоритм блочного шифрования работает, разделяя сообщение на меньшие блоки и шифруя каждый блок отдельно. Размер блока обычно составляет 64 бита. С другой стороны, потоковый шифр шифрует один бит за раз. Преимущество потоковых шифров состоит в том, что они намного быстрее алгоритмов блочных шифров, обычно в несколько раз быстрее. Алгоритмы блочного шифрования могут использовать блочные шифры во многих режимах.

Алгоритм может быть:

Итерация блочного шифра;

 электронный код книги кодов;

Chain цепочка шифров;

Feedback пароль обратной связи;

 выходной обзор;

Итеративный блочный шифр - это блочный шифр, который работает, выполняя одинаковое преобразование в одном и том же блоке. Итерации часто называют раундами.

Итеративная секция блочного шифра - Feistel Ciphers. Пароль Feistel - это пароль, который выполняет то же преобразование в том же блоке, чтобы

получить шифрование в виде простого текста. Пароль Фейстеля также известен как пароль DES. Благодаря этому методу для алгоритма используется стандарт шифрования данных. Прочность итеративного блочного шифра можно увеличить, увеличив количество раундов. Расходы на это со временем приведут к снижению производительности при шифровании и дешифровании данных. Обычно компромиссы не стоят того, потому что для некоторых алгоритмов требуется слишком много циклов, чтобы усложнить шифрование, поэтому разработчики алгоритмов часто взвешивают эффекты увеличения скорости из-за силы или увеличения силы из-за скорости.

Пароль E-Codebook - это пароль, который шифрует каждый блок независимо друг от друга. Электронно зашифрованные книги работают быстрее, чем итеративные блочные шифры и шифры Фейстеля. Преимущество шифрования каждого блока друг с другом означает, что вы можете шифровать и дешифровать данные параллельно.

Цепочка блоков шифров начинается с заполнения случайного значения и его первого блока XOR. Это значение затем шифруется и становится первым блоком зашифрованного текста. Этот зашифрованный блок также используется для следующего блока XOR. Затем значение шифруется, и процесс продолжается до тех пор, пока не останется больше блоков.

Преимущество состоит в том, что все скрыто в процессе XOR'ing. Любой случайный блок не будет указывать на какой-либо другой блок. Обратная связь с паролем аналогична ссылке на зашифрованный блок, но вместо шифрования блока XORed она начинается с зашифрованного начального значения, а затем XOR с первым блоком. Он становится первым блоком зашифрованного текста, а затем шифрует и XOR со вторым блоком. Повторите процесс, пока нет блоков. Выходная обратная связь аналогична обратной связи по паролю. Выходная обратная связь начинается с шифрования начального числа и XOR-значения с первым блоком открытого текста, чтобы получить первый блок зашифрованного текста. Затем зашифрованное начальное число снова шифруется и затем используется для XOR со вторым блоком. Повторяйте этот процесс, пока не останется больше блоков.

DES является одним из самых популярных стандартов шифрования для блочного шифрования. DES расшифровывается как Data Encryption Standard и был выпущен IBM в 1974 году, когда Министерству торговли потребовался универсальный стандарт шифрования. DES работает путем деления простого текста на 64-б

1.5.1 История , конструкция сети Фейстеля

Сеть Фейстел (англ. Feistel Network) (Фейстел Дизайн) (англ. Feistel Password) является одним из методов построения блочных шифров. Сеть - это повторяющаяся итерационная (итерационная) структура, называемая единицей Фейстеля. При переходе из одной ячейки в другую, ключ меняется, и выбор ключа зависит от конкретного алгоритма. Операции шифрования и дешифрования, выполняемые на каждом этапе, очень просты, и с некоторым

улучшением они перекрываются, что требует только обратного порядка используемых ключей. Сеть Feistel проста в программном и аппаратном обеспечении, обеспечивая широкий спектр приложений. Большинство современных блочных шифров используют сеть Фейстеля в качестве основы. Альтернативой сети Фейстеля является сеть договоренностей о замене.

В 1971 году Хорст Фейстел (Horst Feistel) запатентовал два устройства, которые реализовали различные алгоритмы шифрования, и назвал его «Люцифер» («Lucifer»). Одно из устройств использовало конструкцию, которая впоследствии была названа сетью «Feistel Password» («Сеть Фейстеля»). Вместе с Доном Копперсмитом Фейстел стремится создать новую криптосистему в стене IBM. Проект Lucifer является экспериментальным, но он является основой алгоритма DES (стандарт шифрования данных). В 1973 году Хорст Фейстель опубликовал статью под названием «Криптография и компьютерная безопасность (криптография и компьютерная конфиденциальность)» в журнале «Scientific American», в которой он раскрыл многие важные аспекты шифрования и описал проект Luisfer. Одна версия проекта не использовала сеть Фейстеля. В 1977 году правительство США приняло стандарт FIPS 46-3, который признал алгоритм DES в качестве стандартного алгоритма шифрования данных. DES уже давно используется в криптосистемах. Итерационная структура алгоритма позволяет создавать простые программные и аппаратные реализации. По некоторым данным из Советского Союза, в 1970-х годах КГБ разработал блочный шифр с использованием сети Фейстеля, который, вероятно, будет принят в качестве ГОСТ 28147-89 в 1990 году. [19]

В 1987 году были разработаны алгоритмы FEAL и RC2. Сеть Фейстеля была широко доступна в 1990-х годах - в эпоху таких алгоритмов, как Blowfish (1993), TEA (1994), RC5 (1994), CAST-128 (1996), XTEA (1997), XXTEA (1998), RC6 (1998) и так далее.

2 января 1997 года NIST объявил конкурс на создание нового алгоритма шифрования данных для замены DES. Новый блочный шифр был одобрен 26 мая 2002 года под названием AES (Advanced English Encryption Standard). В AES вместо сети Фейстеля используется сеть перестановок-перестановок [9].

шифрование. Позволяет зашифровать некоторую информацию, представленную в двоичном виде (в виде серии нулей и единицы) и сохранить ее в памяти компьютера или другого устройства (например, в файле).

Вся информация алгоритма шифрования делится на блоки фиксированной длины. Если длина входного блока меньше размера данного алгоритма (размера блока) шифрования, блок каким-то образом расширяется. Как правило, длина блока равна степени двух, например: 64 бита, 128 бит. Кроме того, мы рассмотрим операции, которые происходят только с одним блоком, потому что те же операции выполняются с другими блоками во время процесса шифрования.

Выбранный блок делится на два равных субблока - «Левый» (L_0) и «Правый» (R_0). «Левый подблок» L_0 модифицируется функцией f (L_0,

K_0) в зависимости от круглой клавиши K_0, после чего он добавляет «правый подблок» R_0 к модулю 2s.

Результат сложения назначается новому левому подблоку L_1, который будет составлять половину входных данных следующего раунда, а «левый подблок» L_0 выделяется без изменения нового правого подблока R_1 (см. рисунок), где будет другая половина.

После этого операция повторяется N-1 раз. В этом случае между переходами из одной фазы в другую, согласно некоторым математическим правилам, изменяется ключ округления (от K_0 до K_ {1} и т. Д.), Где N - количество раундов в данном алгоритме. Структура и алгоритм шифрования показаны на рисунке. 1.4.

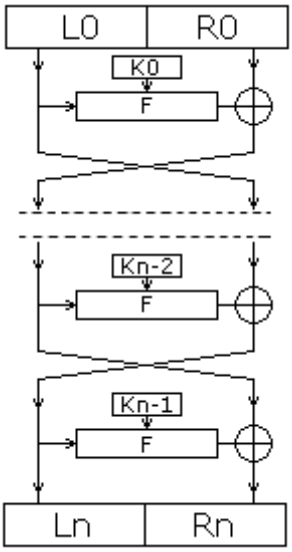


Рисунок 1.4 - Шифрование

Дешифрование информации аналогично шифрованию, за исключением того, что ключи расположены в обратном порядке, то есть не с первого по N-й, а с N-го по первый. Декодирование как показано. 1.5.

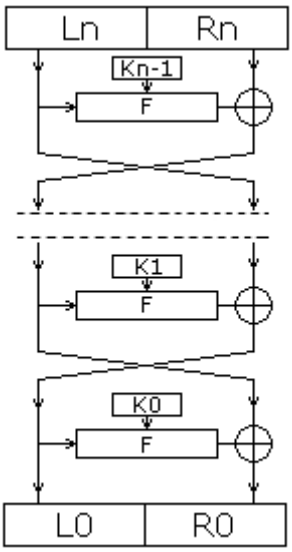


Рисунок 1.5 – Расшифровка

1.6 Законодательное обоснование средств криптографической защиты информации

СКЗИ предназначены для:

– сохранения конфиденциальности данных при помощи шифра;

– аутентификации, в том числе контроля целостности данных, при помощи имитовставки и (или) ЭЦП;

– генерации, формирования, распределения и (или) управления ключами.

СКЗИ, соответствующие требованиям настоящего стандарта, рассматриваются как технологически завершенные (работоспособные) аппаратные, программные или аппаратно-программные средства.

В зависимости от криптографической стойкости для СКЗИ устанавливаются 4 уровня безопасности:

СКЗИ первого уровня безопасности предназначены для защиты информации, ущерб от разглашения, навязывания, или несанкционированного изменения которой в объеме, защищенном с использованием одного и того же ключа (одних и тех же ключей), не превышает 100 минимальных расчетных показателей;

СКЗИ второго уровня безопасности предназначены для защиты информации, ущерб от изменения которой в объеме, защищенном с использованием одного и того же ключа (одних и тех же ключей), не превышает 10 000 минимальных расчетных показателей;

СКЗИ третьего уровня безопасности предназначены для защиты информации, ущерб от изменения которой в объеме, защищенном с использованием одного и того же ключа (одних и тех же ключей), не превышает 1 000 000 минимальных расчетных показателей;

СКЗИ четвертого уровня безопасности предназначены для защиты информации, ущерб от изменения которой в объеме, защищенном с

использованием одного и того же ключа (одних и тех же ключей), не превышает 100 000 000 минимальных расчетных показателей.

СКЗИ не могут быть признаны соответствующими первому, второму, третьему или четвертому уровню безопасности, если вычислительная сложность существующих алгоритмов вскрытия криптографической защиты, обеспечиваемой ими, составляет менее 250, 280, 2^{120} или 2^{160} соответственно.

Средства криптографической защиты информации должны соответствовать требованиям настоящего стандарта и технической документации, утвержденной в установленном порядке.

Общие требования к СКЗИ. Генерируемые СКЗИ ключи (кроме открытых ключей) должны представлять собой последовательности случайных чисел, формируемые с помощью физических генераторов шума (например, тепловых, диодных, радиационных, импульсных), либо последовательности псевдослучайных чисел, формируемые с использованием случайных событий (например, системных параметров ЭВМ, движений мыши, нажатий клавиатуры, состояния таймера).

СКЗИ, использующие распределение ключей по незащищенным каналам связи, должны обеспечивать криптографическую защиту ключей в целях предотвращения разглашения и несанкционированного изменения этих ключей (кроме разглашения открытых ключей), а также навязывания ложных ключей.

Любой используемый СКЗИ ключ должен применяться только одним алгоритмом криптографического преобразования, например, только для шифрования или только для формирования электронной цифровой подписи.

Должна обеспечиваться защита от несанкционированного изменения СКЗИ, в том числе от модификации или подмены их элементов и модулей, с целью исключения влияния на криптографическую стойкость СКЗИ.

Требования к технической документации СКЗИ. Техническая документация (конструкторская, технологическая и программная документация, в зависимости от вида СКЗИ) должна содержать полное описание реализованных в СКЗИ алгоритмов криптографических преобразований, генерации, формирования, распределения и управления ключами.

Если в СКЗИ реализованы алгоритмы криптографических преобразований, определенные государственными и межгосударственными стандартами или другими нормативными документами по стандартизации, действующими или применяемыми в Республике Казахстан в установленном порядке, то в технической документации вместо их полного описания допускается делать ссылки на данные документы.

СКЗИ должны реализовывать алгоритмы криптографических преобразований в точном соответствии с их описанием, приведенным в технической документации.

В каждый комплект СКЗИ должна входить эксплуатационная документация, которая полно и адекватно описывает все возможные режимы

их использования и содержит перечень всех организационных и технических мер, необходимых для обеспечения безопасности обрабатываемой информации, включая порядок и частоту смены ключей, порядок технического обслуживания СКЗИ и действия, которые необходимо предпринять для устранения ошибок оператора и других нештатных ситуаций, возможных во время эксплуатации, а также их последствий.

Требования к СКЗИ первого уровня безопасности:

– длина ключа реализуемых СКЗИ симметричных алгоритмов криптографического преобразования должна быть не менее 60 бит;

– длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования должна быть не менее 120 бит.

Длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования, криптографическая стойкость которых основана на вычислительной сложности задачи разложения составного числа на множители или задачи дискретного логарифмирования в конечном поле, должна быть не менее 500 бит.Длина вычисляемого СКЗИ хэш-кода должна быть не менее 120 бит.Длина формируемой СКЗИ ЭЦП должна быть не менее 120 бит.

Реализуемый СКЗИ принцип генерации и формирования ключей должен обеспечивать принятие каждым битом ключа единичного значения с вероятностью из интервала (0,50 ± 0,03).

Требования к СКЗИ второго уровня безопасности:

– длина ключа реализуемых СКЗИ симметричных алгоритмов криптографического преобразования должна быть не менее 100 бит;

– длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования должна быть не менее 160 бит.

Длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования, криптографическая стойкость которых основана на вычислительной сложности задачи разложения составного числа на множители или задачи дискретного логарифмирования в конечном поле, должна быть не менее 1500 бит. Длина вычисляемого СКЗИ хэш-кода должна быть не менее 160 бит. Длина формируемой СКЗИ ЭЦП должна быть не менее 200 бит.

Реализуемый СКЗИ принцип генерации и формирования ключей должен обеспечивать принятие каждым битом ключа единичного значения с вероятностью из интервала (0,50 ± 0,01).СКЗИ должны реализовывать процедуры вычисления и проверки контрольной информации о ключах в целях предотвращения использования случайно искаженных на этапе распределения и загрузки ключей с вероятностью не менее 0,9999.При предварительном шифровании СКЗИ должны реализовывать процедуры вычисления и проверки контрольной информации о шифруемых данных в целях выявления случайно искаженных зашифрованных данных с вероятностью не менее 0,9999.СКЗИ должны информировать оператора об

установлении, сбросе, а также о невозможности установления режима шифрования.

Требования к СКЗИ третьего уровня безопасности:

– длина ключа реализуемых СКЗИ симметричных алгоритмов криптографического преобразования должна быть не менее 150 бит;

– длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования должна быть не менее 250 бит.

Длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования, криптографическая стойкость которых основана на вычислительной сложности задачи разложения составного числа на множители или задачи дискретного логарифмирования в конечном поле, должна быть не менее 4000 бит. Длина вычисляемого СКЗИ хэш-кода должна быть не менее 250 бит .Длина формируемой СКЗИ ЭЦП должна быть не менее 300 бит.

Реализуемый СКЗИ принцип генерации и формирования ключей должен обеспечивать принятие каждым битом ключа единичного значения с вероятностью из интервала (0,500 ± 0,003), при этом ключи должны быть последовательностями случайных чисел и формироваться с помощью физических генераторов шума.

СКЗИ должны реализовывать процедуры формирования и проверки имитовставок или ЭЦП для ключей в целях предотвращения использования случайно или умышленно искаженных на этапе распределения и загрузки ключей с вероятностью не менее 0,999999.

При предварительном шифровании СКЗИ должны реализовывать процедуры формирования и проверки имитовставок или ЭЦП для шифруемых данных в целях выявления случайно или умышленно искаженных зашифрованных данных с вероятностью не менее 0,999999.СКЗИ должны информировать оператора об установлении, сбросе, а также о невозможности установления режима шифрования и других нештатных ситуациях.СКЗИ должны обеспечивать иерархическую криптографическую защиту ключей на этапе их распределения и управления в целях предотвращения разглашения и несанкционированного изменения этих ключей (кроме разглашения открытых ключей), а также навязывания ложных ключей, или эксплуатационная документация СКЗИ должна содержать организационные и технические меры по обеспечению защиты от данных угроз.Реализуемые СКЗИ штатные процедуры удаления (уничтожения) ключей должны гарантировать невозможность их восстановления.

Требования к СКЗИ четвертого уровня безопасности.

– длина ключа реализуемых СКЗИ симметричных алгоритмов криптографического преобразования должна быть не менее 200 бит;

– длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования должна быть не менее 400 бит.

Длина ключа реализуемых СКЗИ асимметричных алгоритмов криптографического преобразования, криптографическая стойкость которых

основана на вычислительной сложности задачи разложения составного числа на множители или задачи дискретного логарифмирования в конечном поле, должна быть не менее 8000 бит.Длина вычисляемого СКЗИ хэш-кода должна быть не менее 400 бит.Длина формируемой СКЗИ ЭЦП должна быть не менее 400 бит.

Реализуемый СКЗИ принцип генерации и формирования ключей должен обеспечивать принятие каждым битом ключа единичного значения с вероятностью из интервала (0,500 ± 0,001), при этом ключи должны быть последовательностями случайных чисел и формироваться с помощью физических генераторов шума.

СКЗИ должны реализовывать процедуры формирования и проверки имитовставок или ЭЦП для ключей в целях предотвращения использования случайно или умышленно искаженных на этапе распределения и загрузки ключей, с вероятностью не менее 0,999999999. СКЗИ должны реализовывать процедуры формирования и проверки имитовставок или ЭЦП для шифруемых данных в целях выявления случайно или умышленно искаженных зашифрованных данных с вероятностью не менее 0,999999999.

СКЗИ должны информировать оператора об установлении, сбросе, а также о невозможности установления режима шифрования и других нештатных ситуациях, предотвращать транзит через себя открытых данных в область хранения, распределения и последующей обработки зашифрованных данных.

СКЗИ должны обеспечивать иерархическую криптографическую защиту ключей на этапе их распределения и управления в целях предотвращения разглашения и несанкционированного изменения этих ключей (кроме разглашения открытых ключей), а также от навязывания ложных ключей.

Реализуемые СКЗИ штатные процедуры удаления (уничтожения) ключей должны гарантировать невозможность их восстановления. Если СКЗИ не реализуют указанных процедур, то эти процедуры гарантированного удаления (уничтожения) ключей (кроме открытых ключей) должны быть реализованы техническими средствами, поставляемыми в комплекте с СКЗИ.

2 Общая постановка задачи

Необходимо создать приложение симметричного блочного шифрования. В котором будет реализованы все современные методы и средства криптографии.

К программе предъявлены следующие требования:

– программа должна быть кроссплатформенная, с поддержкой Windows и Linux систем;

– должно быть реализовано шифрования любых файлов и текста;

– шифр реализованный в программе должен иметь определённый уровень стойкости к криптоанализу;

– реализован генератор ключей;

– данные после расшифровки не должны быть изменены;

– программа должна иметь простой и удобный интерфейс описан с помощью веб-технологий.

2.1 Схема работы приложения

Работу приложения описывает схема показанная на рисунке 2.1. На схеме показано как реализовано приложение, принцип работы, и последовательность действий при работе с программой. Программа состоит из заставки и главного окна. В главном окне реализован весь функционал программы.

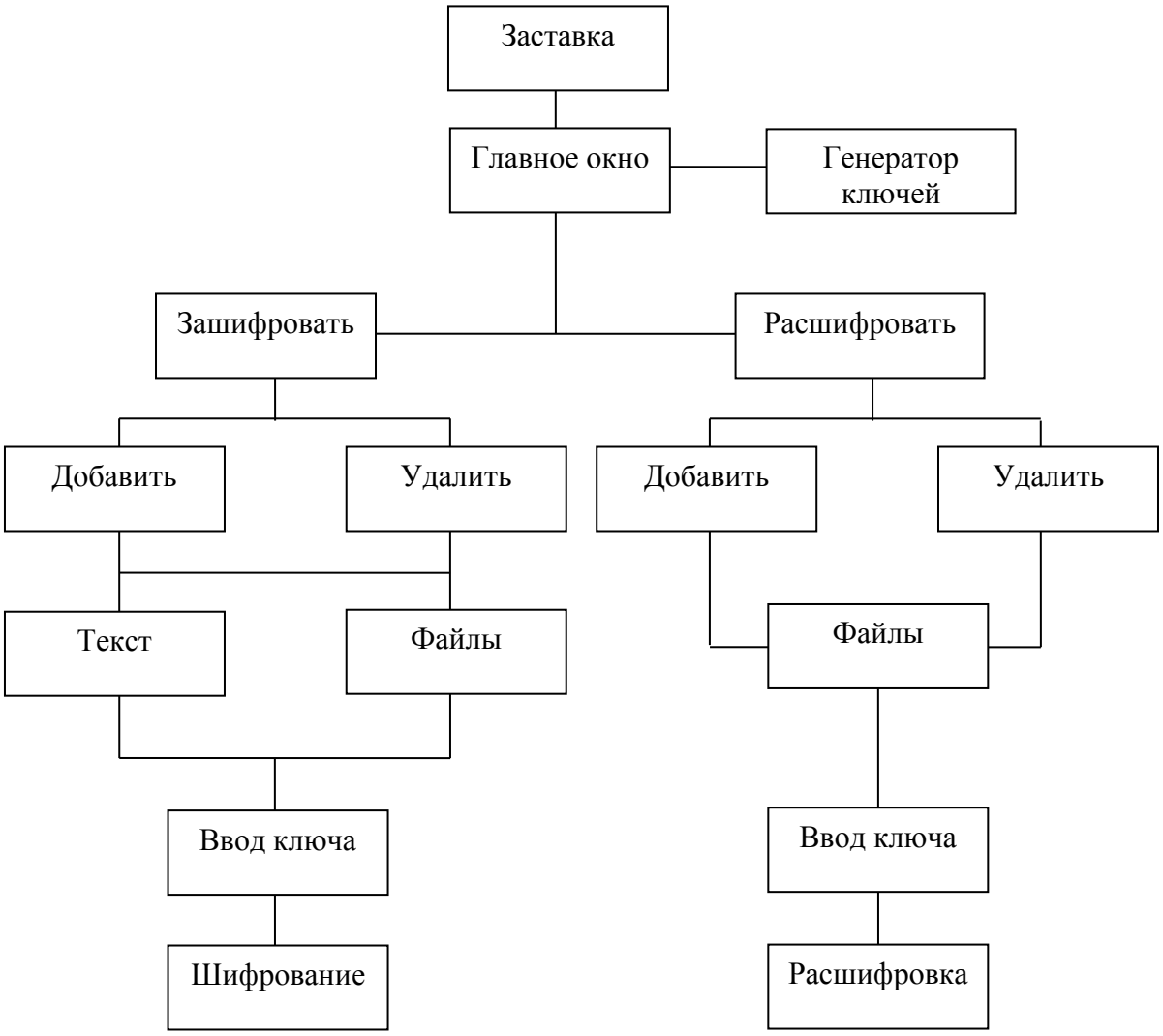


Рисунок 2.1 - Схема работы приложения

2.2 Схема шифрования данных

Схема шифрования представляет собой одну из разновидности сети Фейстеля. Сеть представляет собой определённую многократно повторяющуюся структуру, называющуюся ячейкой Фейстеля. Был использован режим шифрования CBC (Cipher Block Chaining) – сцепление блоков по шифротексту. Когда каждый блок открытого текста маскируется соответственно блоком шифротекста, полученном на предыдущем этапе. Схема шифрования представлена на рисунке 2.2.

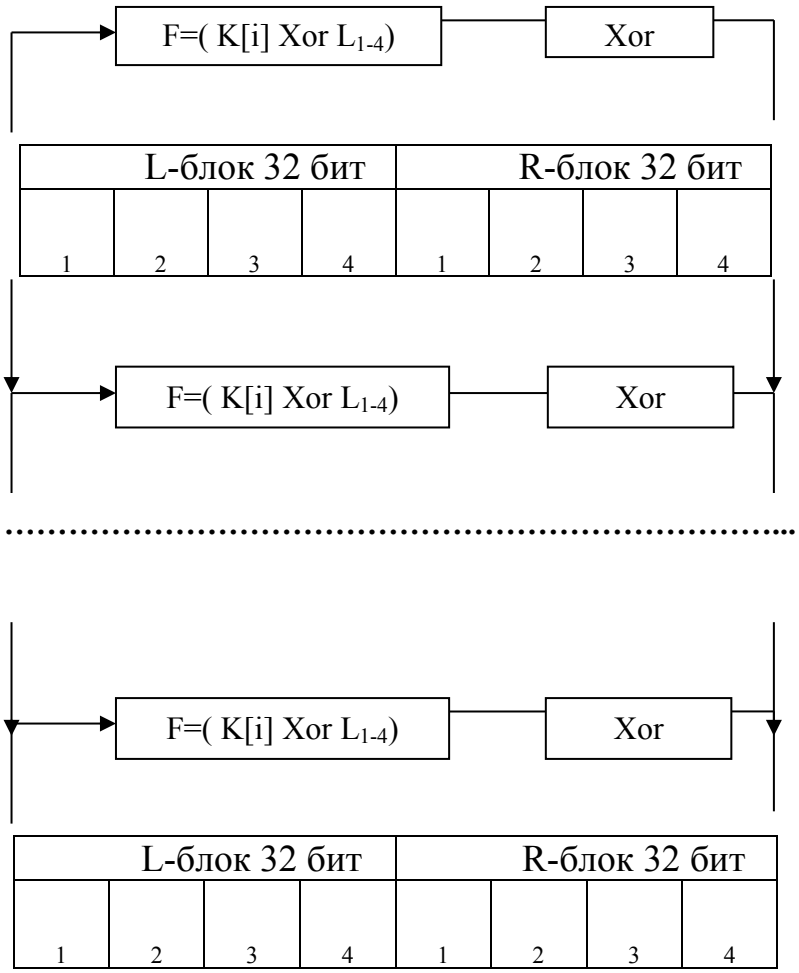


Рисунок 2.2 - Схема шифрования данных

2.3 Описание приложения

Приложение состоит из:

– Main.cpp,1 кб, главный модуль программы, вызов заставки и главного окна;

– Dialog.cpp, 0,4 кб, модуль окно «Инструкция» , инструкция по использованию программы;

– Mainwindow.cpp, 15.6 кб , модуль главного окна программы;

– Oprog.cpp, 0.3 кб, модуль окна «О программе»;

– Dialog.h, 0.4 кб, заголовочный файл окна «Инструкция»;

– Mainwindow.h, 1.2 кб , заголовочный файл «Главного окна»;

– Oprog.h, 0.3 кб , заголовочный файл окна «О программе»;

– Dialog.ui, 3 кб, файл формы окна «Инструкция»;

– Mainwindow.ui, 42 кб , файл формы «Главного окна»;

– Oprog.ui, 3 кб , файл формы окна «Опрограмме»;

– Resource.qrc , 0.8 кб , файл ресурсов программы;

– Diplom.pro, 0.5 кб , файл проекта программы.

Таблица 2.1 - Описание процедур и функций

Процедура/программа	Назначение
int main(int argc, char *argv[])	Вызов заставки и главного окна
void read_faist(char buf_data[], string key)	Шифрование, сеть Фейстеля
void open_file_inc_block(QString fil_open, QString fil_ot , string key)	Посчет и дополнение блока
void open_file_inc_block_en(QString fil_open,QString fil_ot , string key);	Подсчет количество блоков
void open_file_del_block_en(unsigned long long int col_block,QString fil_open,QString fil_ot , string key);	Удаление лишних данных в последнем блоке
void read_faist_en(char buf_data[], string key)	Расшифровка, сеть Фейстеля
on_pushButton_3_clicked()	Добавление файлов
on_commandLinkButton_2_clicked()	Переключение страницы
Процедура/программа	Назначение
on_lineEdit_textChanged(const QString &arg1)	Заполнение индикатора пароля
on_pushButton_5_clicked()	Шифрование
on_commandLinkButton_5_clicked()	Переключение страницы
on_lineEdit_3_textChanged(const QString &arg1)	Заполнение индикатора пароля
on_pushButton_7_clicked()	Добавление файлов
on_pushButton_4_clicked()	Удаление файлов из списка
on_pushButton_8_clicked()	Удаление файлов из списка
on_pushButton_9_clicked()	Шифрование текста
on_pushButton_11_clicked()	Генератор ключей

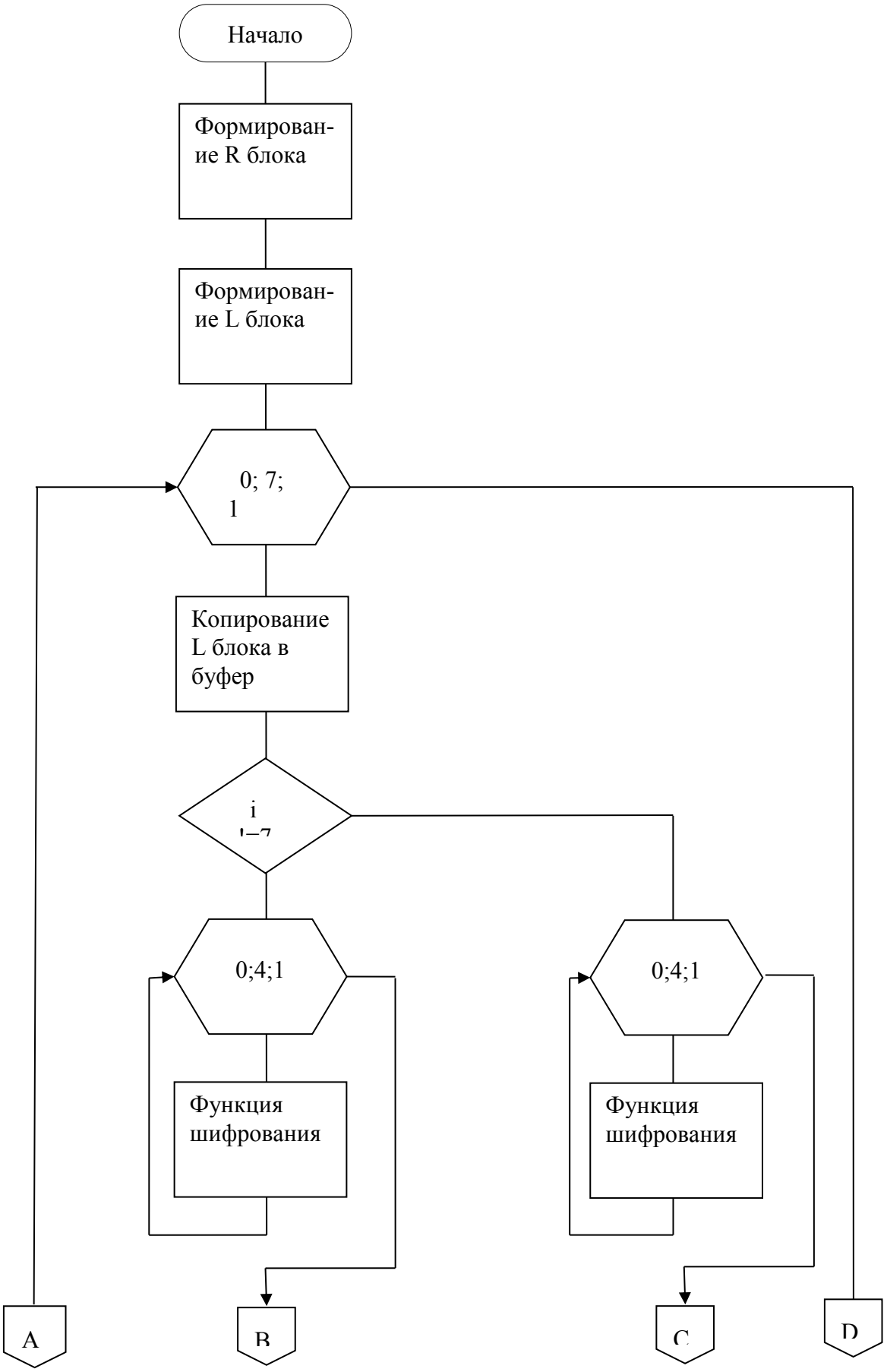
2.4 Описание программ

2.4.1 Описание программы блочного симметричного шифрования

Необходимо осуществить шифрование файла. До начало шифрования файла , файл необходимо дополнить блоком чтоб размер файла был кратным 64 битам.

2.4.1.1 Схема алгоритма программы блочного симметричного шифрования

Схема алгоритма программы блочного симметричного шифрования представлена на рисунке 2.3. Схема описывает шифрование файла с помощью сети Фейстеля.


Начало
Формирование R блока
Формирование L блока
0; 7;
1
Копирование L блока в буфер
i
0;4;1
0;4;1
Функция шифрования
Функция шифрования
A
B
C
D

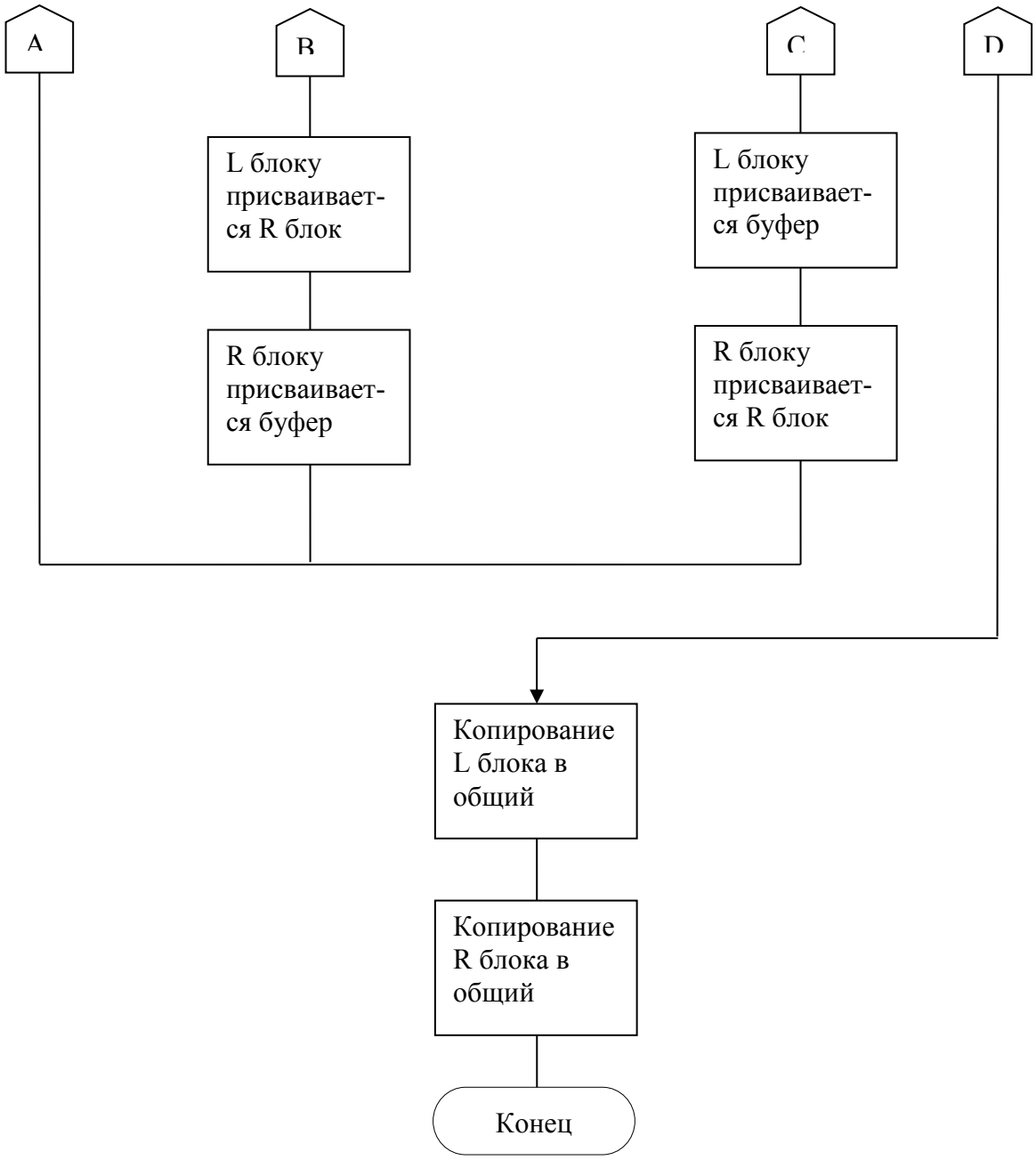


Рисунок 2.3 – Блок схема программы блочного симметричного шифрования

2.4.1.2 Таблица идентификаторов программы блочного симметричного шифрования

Идентификаторы программы блочного симметричного шифрования приведены в таблице 2.2.

Таблица 2.2 – Таблица идентификаторов программы блочного симметричного шифрования

Идентификатор	Смысловое содержание	Тип	Разрядность
l_block[4]	L блок	char	255
r_block[4]	R блок	char	255
tmp[4]	буфер	char	255

2.5 Инструкция пользователю

2.5.1 Установка приложения на ОС Linux

Чтоб не зависеть от какой либо пакетной базы ОС Linux приложение будет хранится в ZIP-архиве. Приложение работоспособно на любой платформе ОС Linux не зависимо от пакетной базы дистрибьютива буть это DEB-based дистрибьютив (Debian , Ubuntu , Xubuntu , Mint и т.д) или RPM-based дистрибьютив (Fedora , OpenSUSE и т.д). Для начало установки приложения необходимо:

Необходимо открыть архив с программой в любом менеджере архивов как показано на рисунке 2.4.

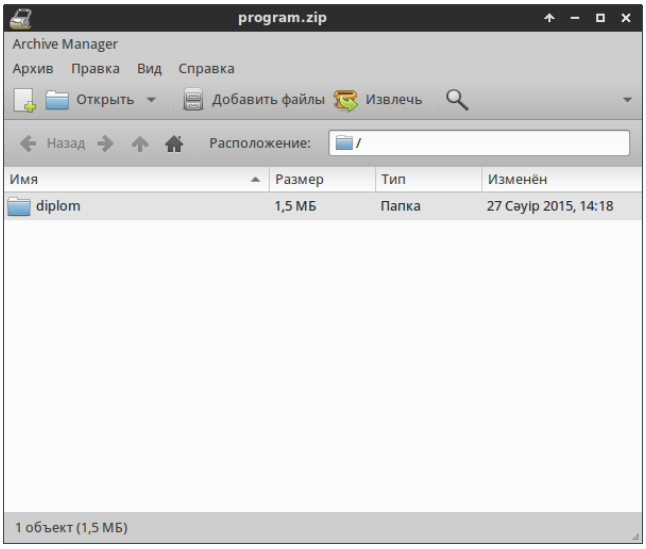


Рисунок 2.4 - Открытие ZIP-архива с программой

Нужно будет кликнуть на кнопку извлечь чтоб открыть проводник и указать папку куда будет распакована папка с программой как показано на рисунке 2.5

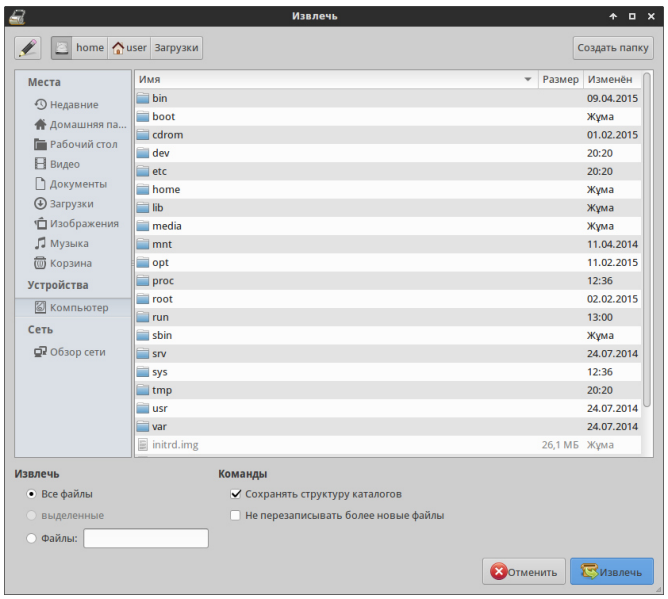


Рисунок 2.5 - Извлечение папки с программой

.Так как по соглашению ОС Linux сторонние ПО необходимо устанавливать в папку «/opt» это делается из за соображения безопасности . Папку необходимо переместить в директорию «/opt».

Для этого необходимо в терминале перейти в директорию с где лежит программа.

В терминале нужно выполнить команду cd путь до директории как показано на рисунке 2.6.

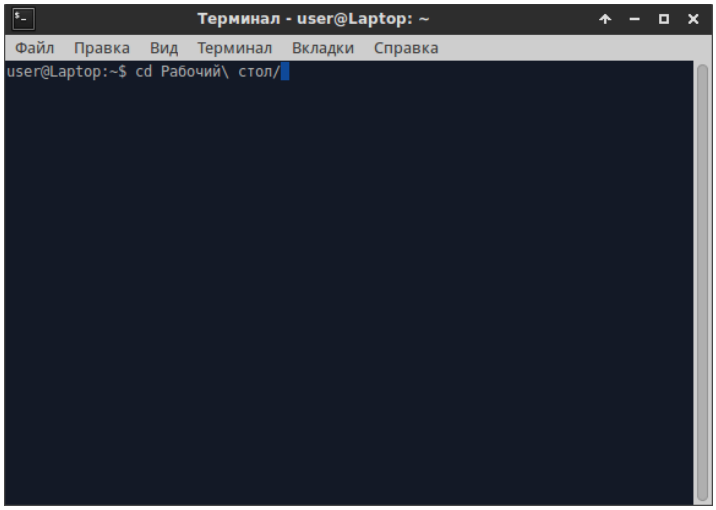


Рисунок 2.6 - Выполнение команды в терминале

После того как перешли в директорию с папкой необходимо выполнить команду перемещения папки в директорию «/opt» sudo mv diplom /opt как показано на рисунке 2.7

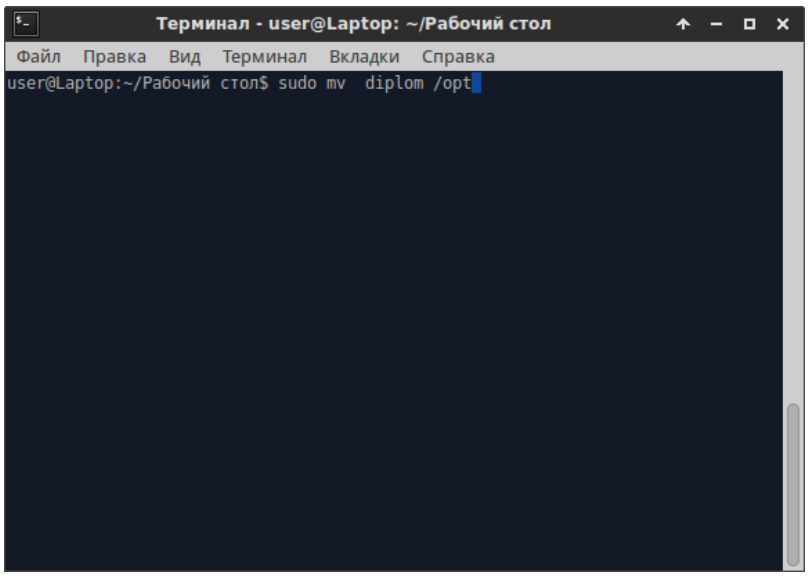


Рисунок 2.7 - Выполнение команды в терминале

После того как папка была перемещена нужно выставить ей права доступа командой sudo chmod 775 diplom как показано рисунке 2.8.

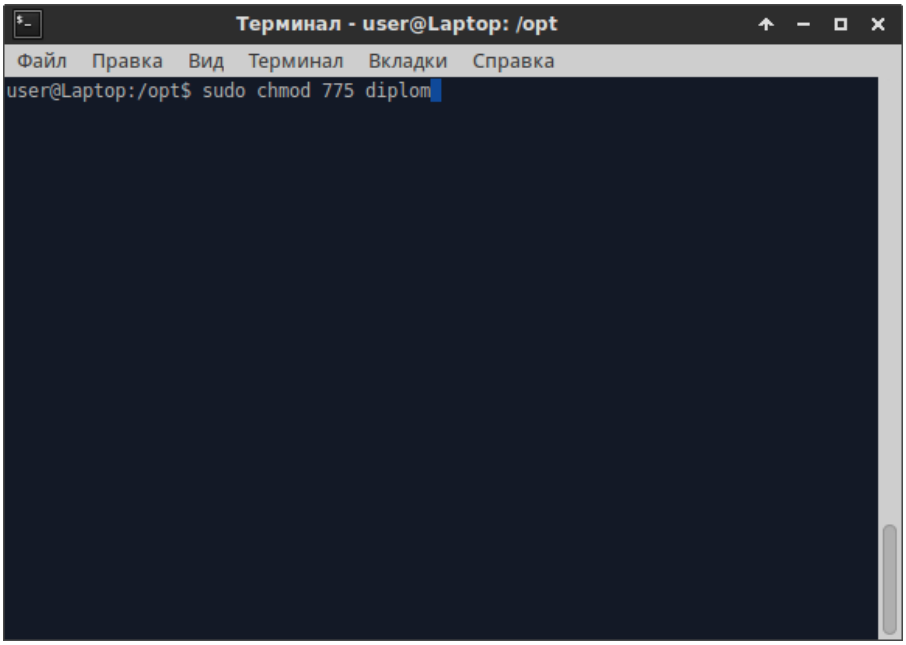


Рисунок 2.8 - Выполнение команды в терминале

Чтоб бы при использовании программы не возникало проблем также нужно сменить владельца папки командой sudo chown –R имя пользователя : группа diplom как показано на рисунке 2.9.

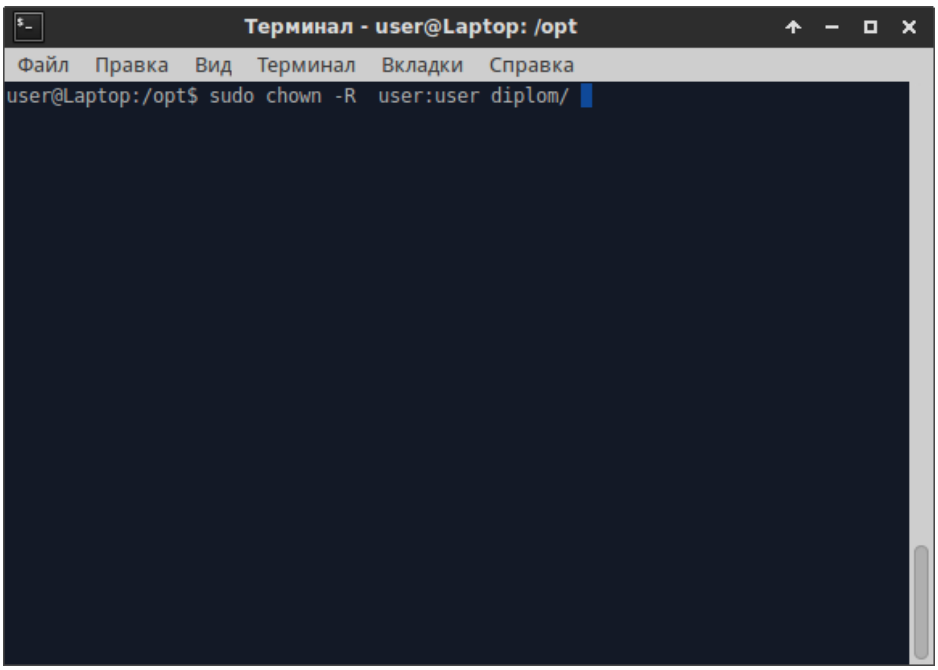


Рисунок 2.9 - Выполнение команды в терминале

После этого нужно перейти в папку с программой командой cd diplom как показано на рисунке 2.10

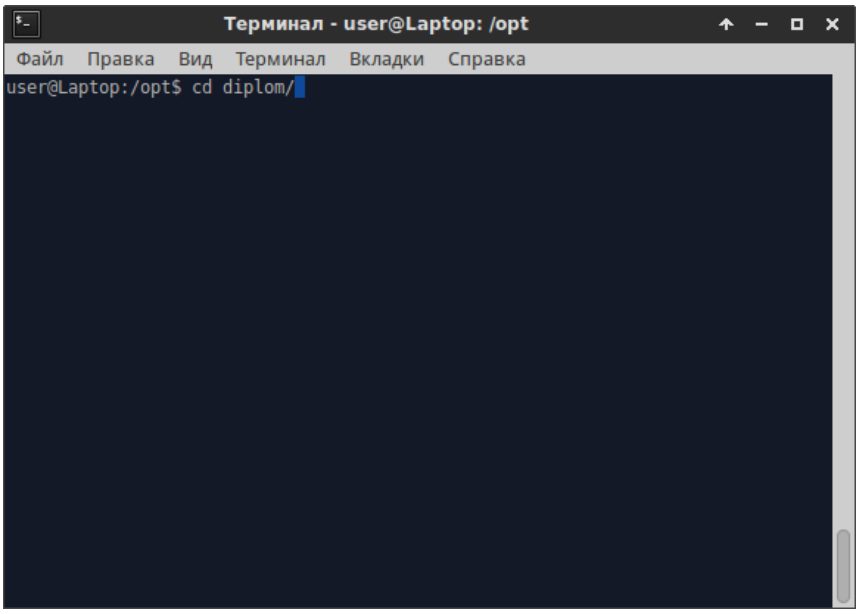


Рисунок 2.10 - Выполнение команды в терминале

Затем необходимо задать права доступа на запуск приложения командой sudo chmod +x diplom как показано на рисунке 2.11.

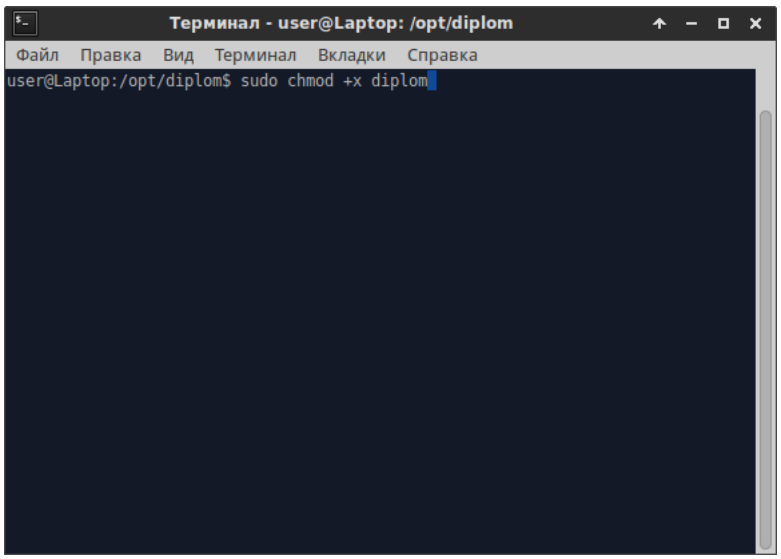


Рисунок 2.11 - Выполнение команды в терминале

Установка программы подходит к концу необходимо запустить программу на выполнение. Программа должна попросить удовлетворить зависимости и выстроит список пактов которые необходимо будет установить. В зависимости от дистрибьютива установить пакеты можно будет следящим образом :

Если это DEB-based дистрибьютив (Debian , Ubuntu , Mint) необходимо будет выполнить команду sudo apt-get install список пакетов через пробел.

Если это RPM-based дистрибьютив (Fedora , OpenSUSE) необходимо будет выполнить команду yum install список пакетов через пробел.

После того как все зависимости будут удовлетворены необходимо будет сделать символическую ссылку программы в катало «/bin» командой

sudo ln –s /opt/diplom/diplom /bin/diplom также сменить владельца и права доступа к ссылке. Программа установлена и готова к использованию.

2.5.2 Установка приложения на ОС Windows

Установка программы на ОС Windows немного отличается отличается по сравнению с установкой приложения на ОС Linux. Для того что бы установить программу, приложение необходимо извлечь из архива.

В архиве находятся три папки:

– Diplom - папка с приложением;

– Dll_qt_5_win_32 - папка с библиотеками и зависимостями программы;

– Microsoft.NET.Framework.4 – программная платформа для запуска программы;

Папку Diplom необходимо извлечь в корень диска где установлена система как показано на рисунке 2.12 . Путь до приложения не должен содержать пробелов и спец символов.

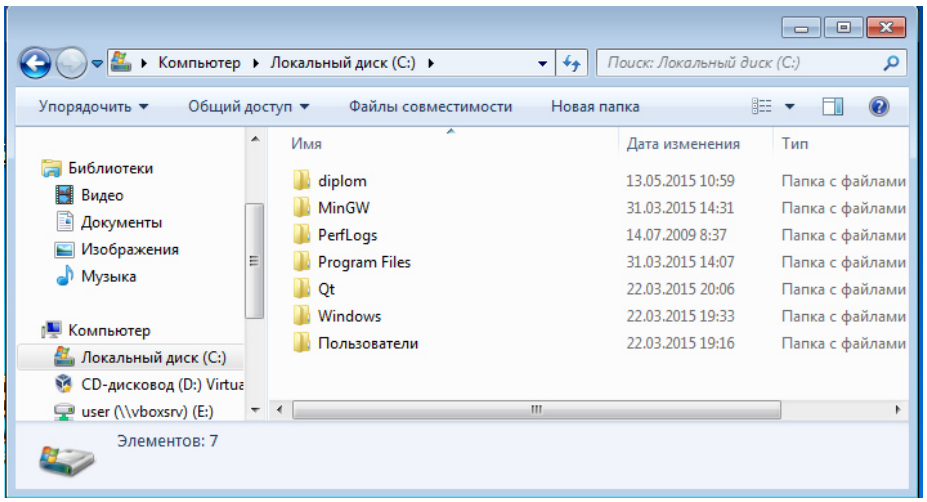


Рисунок 2.12 – Извлечение папки в корень диска

После того как папка с приложением была извлечена необходимо создать ссылку на рабочем столе.Для этого необходимо зайти в папку Diplom->debug. Там находится исполняемый файл программы . Необходимо кликнуть правой кнопкой мыши по файлу выбрать пункт Отправить-> На рабочий стол (создать ярлык). После этого необходимо из папки Dll_qt_5_win_32 копировать все файлы в папку Windows->System32 как показано на рисунке 2.13.

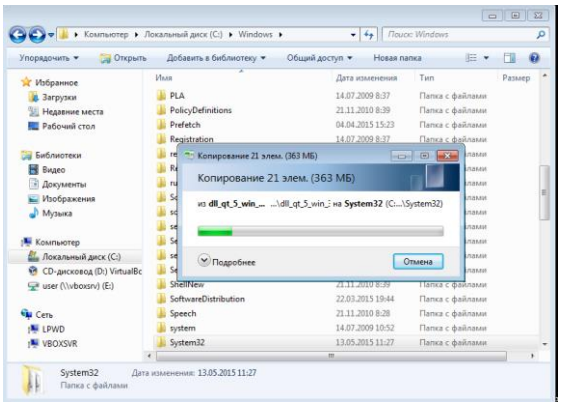


Рисунок 2.13 – Копирование файлов

Остался последний этап установки. Необходимо из папки Microsoft.NET.Framework.4 установить приложение. Приложение является программной платформой для запуска приложений написанных на разных языках программирования.

2.5.3 Использование приложения

Для запуска приложения необходимо :

– в ОС Windows кликнуть по ярлыку на рабочем столе diplom.exe;

– в ОС Linux необходимо открыть терминал и набрать команду diplom или кликнуть по ярлыку на рабочем столе;

После того как произойдет запуск приложения появится экранная заставка как продемонстрировано на рисунке 2.14.

Рисунок 2.14 – Экранная заставка приложения

После экранной заставки приложения ,откроется главное окно программы. Дальше работа с программой не будет отличатся от платформы. Главное окно имеет одинаковый вид что в ОС Windows что в ОС Linux как показано на рисунке 2.15.

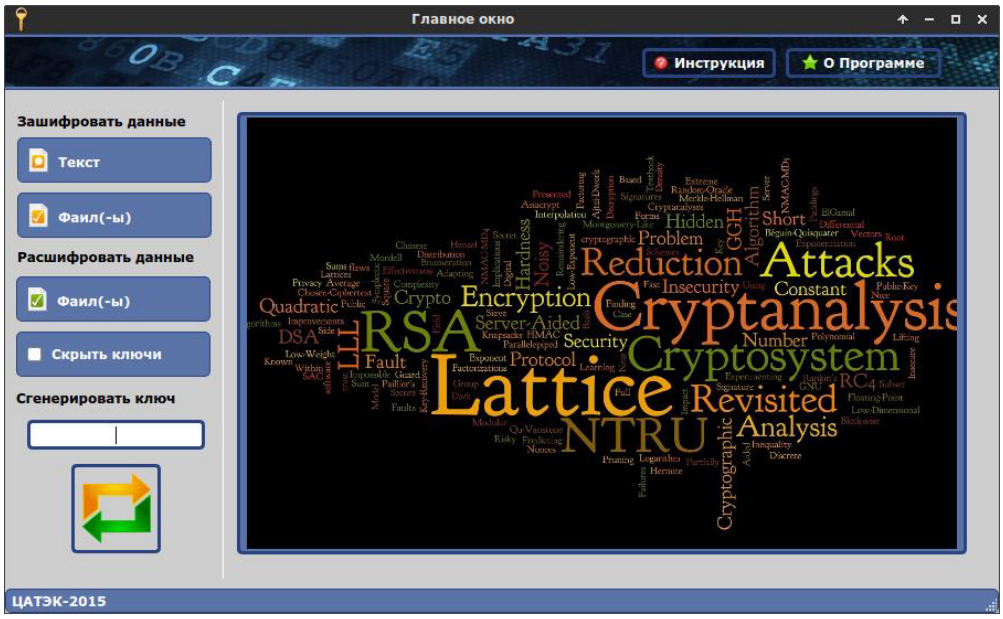


Рисунок 2.15 – Главное окно программы

В главном окне программы реализован весь функционал программы для удобства пользователя. Главное окно программы имеет простой и понятный интерфейс. Окно программы разбито на несколько блоков:

– Блок управления шифрования данных представлен на рисунке 2.16;

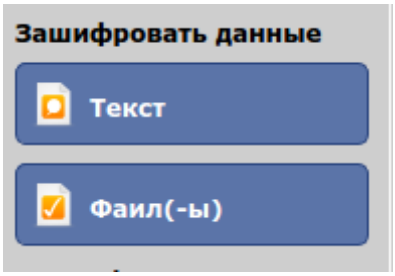


Рисунок 2.16 - Блок управления шифрования

– Блок управления расшифровки данных представлен на рисунке 2.17;

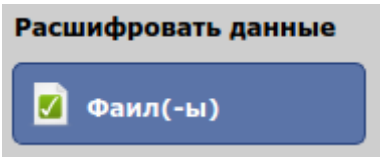


Рисунок 2.17 - Блок управления расшифровки

– Блок управления ключами представлен на рисунке 2.18;

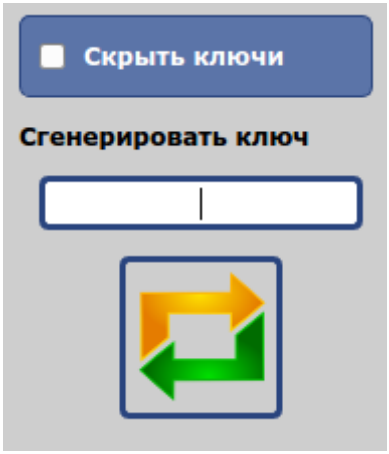


Рисунок 2.18 - Блок управления ключами

– И основная рабочая область программы, которая показана на рисунке 2.19;

Рисунок 2.19 – Основная рабочая область

Для того чтобы зашифровать данные например файлы необходимо в блоке управления шифрования данных нажать кнопку Файл(-ы) как показано на рисунке 2.20;

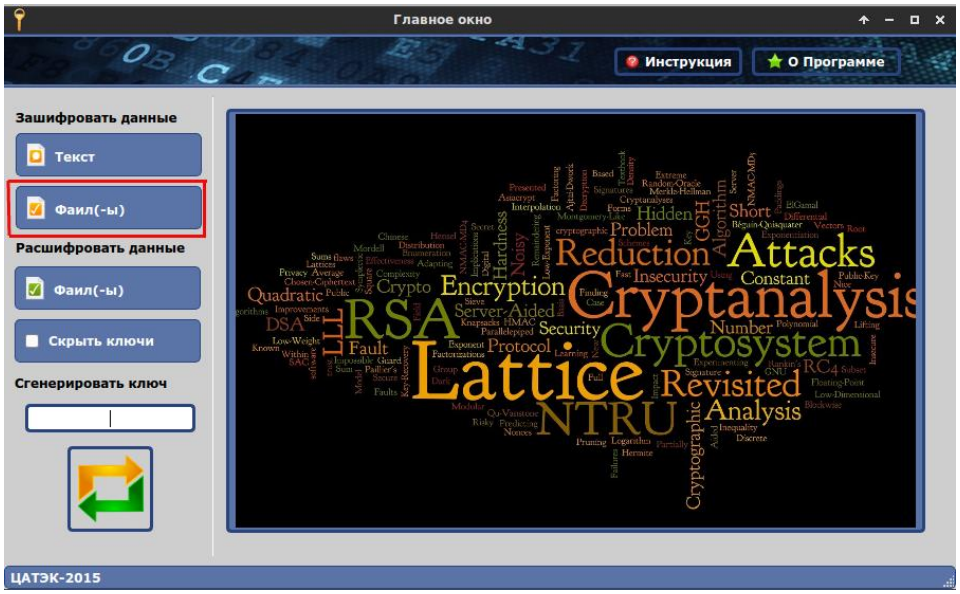


Рисунок 2.20 – Нажатие кнопки Файл(-ы)

После нажатия кнопки в основной рабочей области отобразится страница для шифрования файлов, старница содержит:

– кнопку Добавить - для добавления файлов в список шифруемых;
– кнопку Удалить - для удаления файлов из списка шифруемых;
– поле ввода ключа шифрования ;
– индикатор заполнения ввода ключа;
– таблицу шифруемых файлов;

– кнопку Шифровать.

Рабочая область шифрования файлов продемонстрирована на рисунке 2.21.

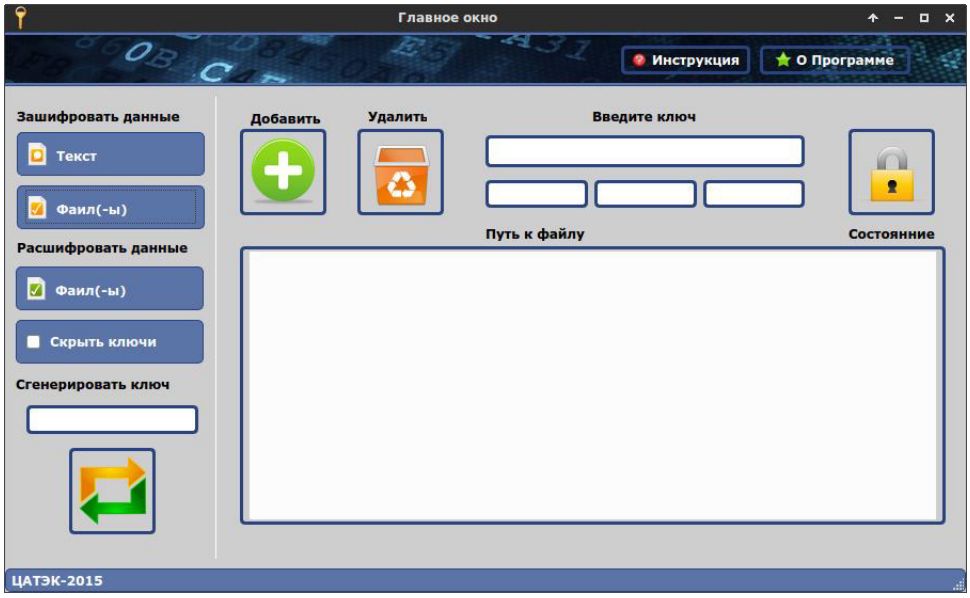


Рисунок 2.21 – Рабочая область шифрования файлов

Для добавления файлов в таблицу шифруемых необходимо нажать кнопку Добавить. После нажатия кнопки Добавить откроется проводник для выбора файлов как показано на рисунке 2.22

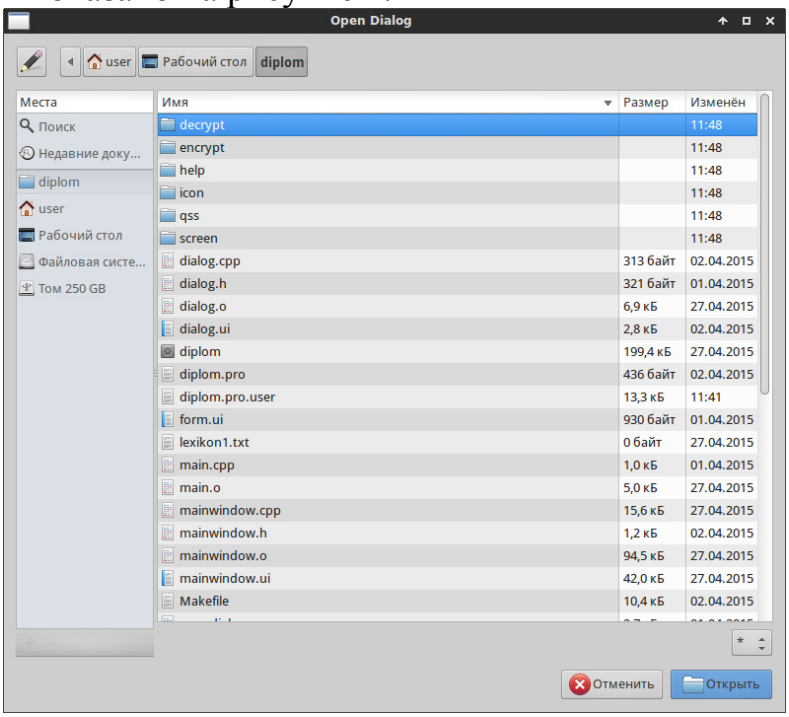


Рисунок 2.22 – Добавление фалов

В окне проводника необходимо будет выбрать файлы и нажать кнопку открыть. После этого в таблице шифруемых файлов отоброзится список . В

колонке Состояние будет выставлен статус Ожидает . Это означает что файлы еще не шифровались, как показано на рисунке 2.23

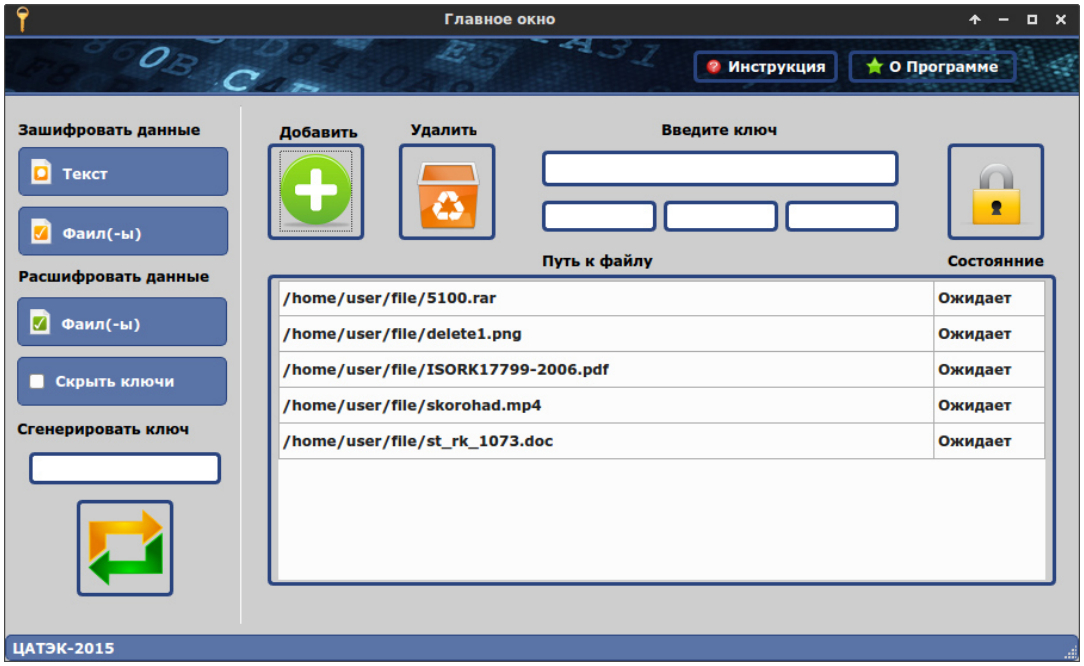


Рисунок 2.23 – Добавление файлов в список шифруемых

Если при добавление файлов , в список попал не нужный файл его можно удалить. Для этого нужно выделить его в списке и нажать кнопку Удалить. После этого необходимо ввести восьми значный ключ для шифрования файлов. Ключ не должен содержать кирилицу и спец символы. По мере заполнения ключа будет заполнятся индикатор как показано на рисунке 2.24.

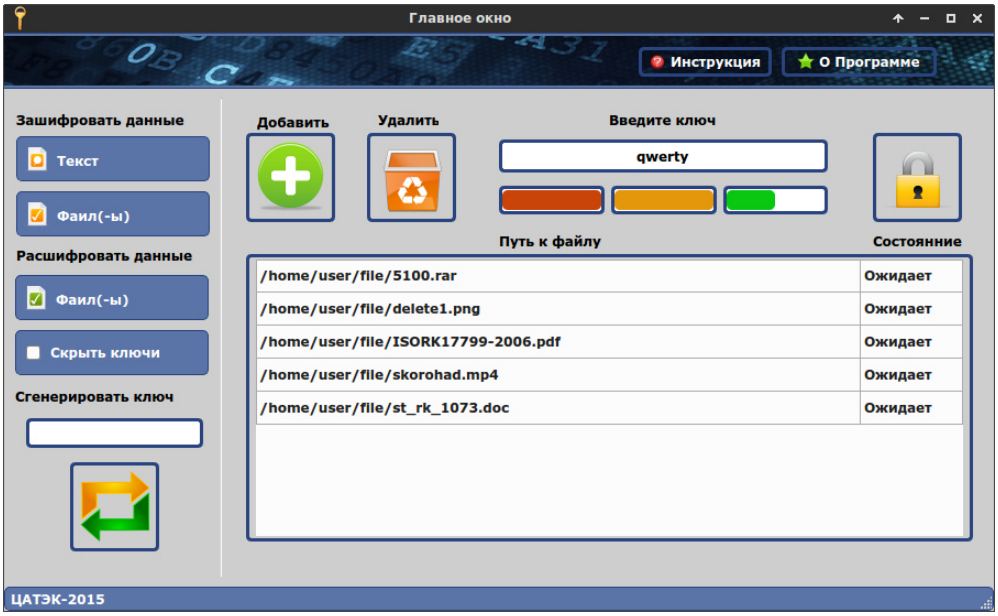


Рисунок 2.24 – Ввод пароля

Чтобы не придумывать ключ шифрования можно воспользовастся встроеным генератором ключей. Для этого необходимо нажать кнопку Сгенерировать несколько раз при этом будет случайным образом сгенирирован пароль. Так же чтоб защитить ключ от кражи (подглядывания) необходимо выставить влажок Скрыть ключи. Если ключ будет короче 8 символов и при этом пользователь попытается этим ключом зашифровать файлы вылезит окно с ошибкой предуприждающие пользователя что ключ короче 8 символов и шифрования не произойдет как показано на рисунке 2.25

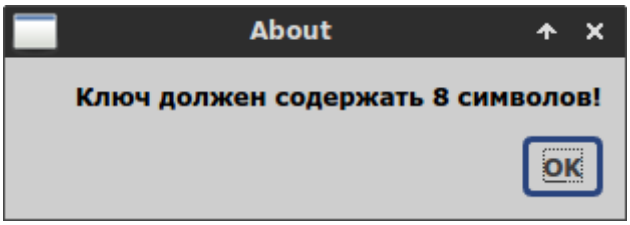


Рисунок 2.25 – Ошибка

После нажатия кнопки Шифровать произойдет шифрования файлов и Статус в таблице будет изменен на Готово как показано на рисунке 2.26.

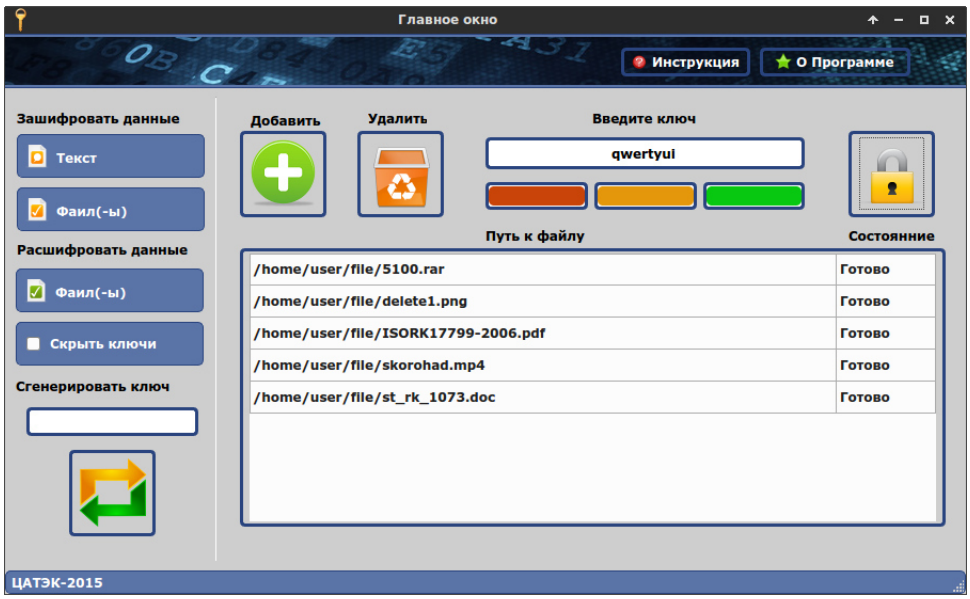


Рисунок 2.26 – Отображение зашифрованых файлов

Все зашифрованые файлы будут находится в папке encrypt как показано на рисунке 2.27. К основному расиширению файлов будет прибавлено .crypt это необходимо для того чтобы все файлы хранились в одном месте и их было легко отличить от нешифрованых файлов.

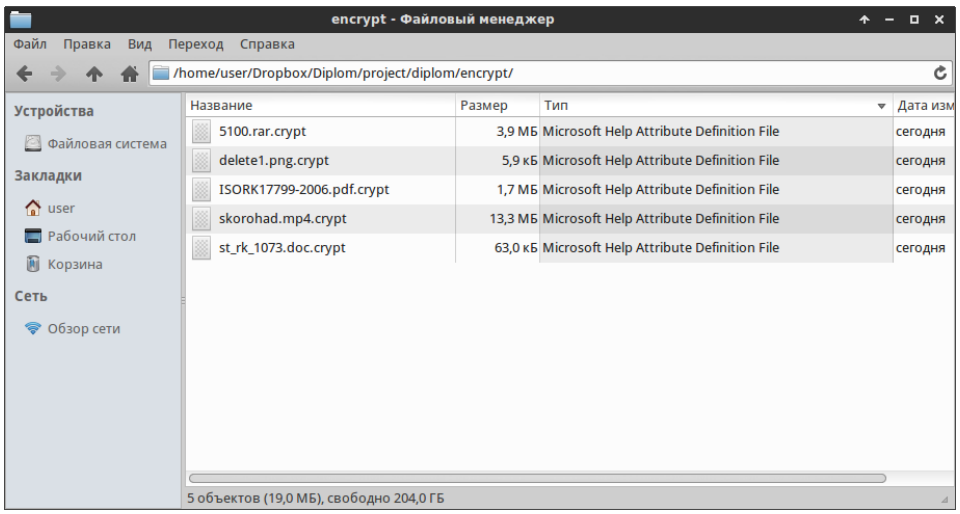


Рисунок 2.27 – Зашифрованные файлы

На этом процесс шифрования закончен. Для расшифровки файлов необходимо в блоке расшифровки данных нажать кнопку Файл(-ы). Откроется страница расшифровки файлов. Процесс расшифровки не чем не отличается от процесса шифрования файлов. Также необходимо выбрать зашифрованные файлы которые хранятся в папке encrypt. Ввести ключ который был введен при шифровании файлов. Нажать кнопку Расшифровать. После этого расшифрованные файлы появятся в папке decrypt с родным расширением как показано на рисунке 2.28.

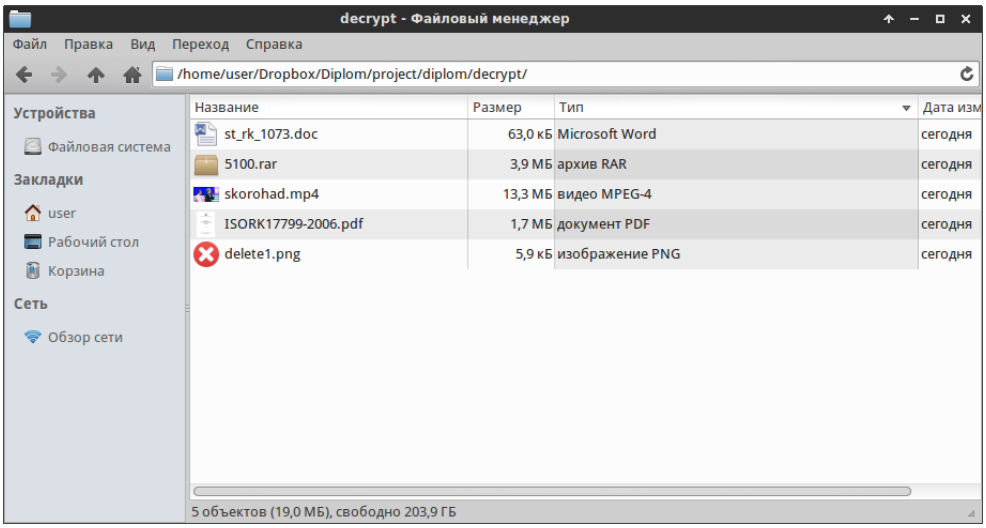


Рисунок 2.28 – Расшифрованные файлы

Процесс шифрования текста немного отличается от процесса шифрования файлов. Для шифрования текста необходимо нажать кнопку Текст. Откроется страница для ввода текста как показано на рисунке 2.29.

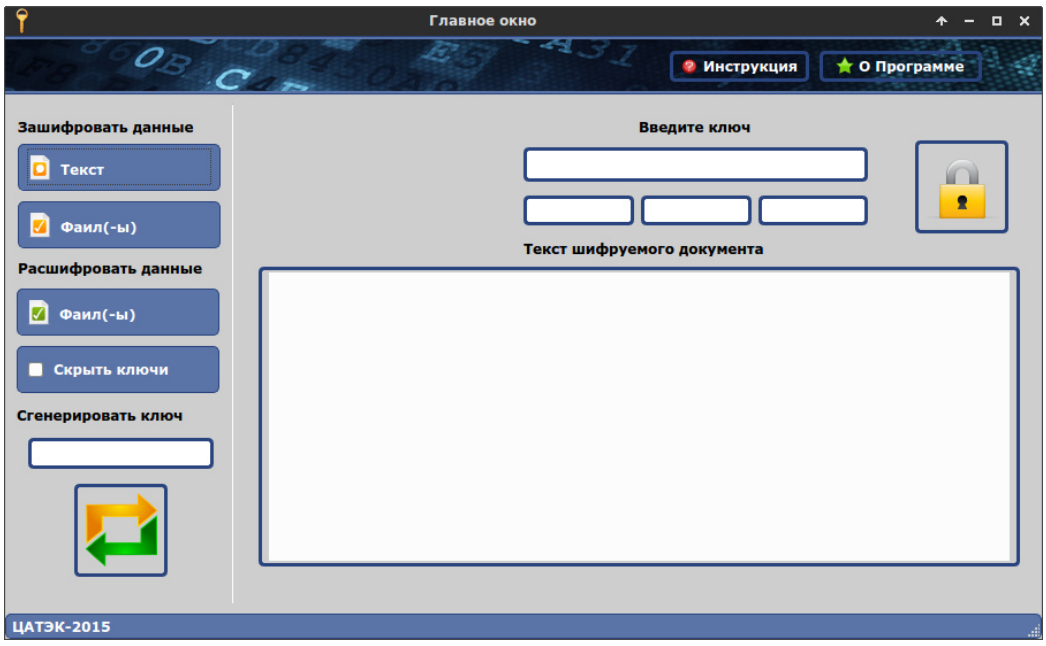


Рисунок 2.29 – Окно для ввода текста

После этого необходимо набрать или вставить текст который будет шифроваться. Ввести ключ для шифрования. Потом нажать кнопку Шифровать. Откроется проводник для сохранения файла, файл рекомендуется сохранить в папку encrypt как показано на рисунке 2.30.

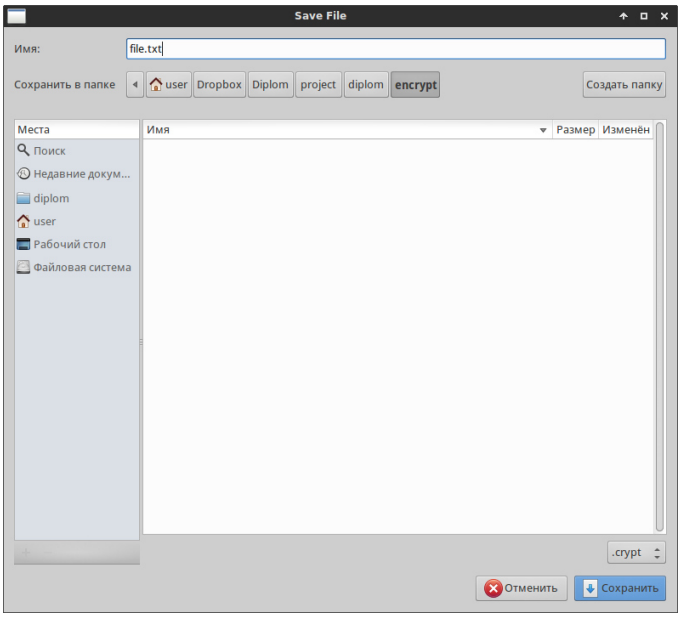


Рисунок 2.30 – Сохранения шифрованного файла

На этом процесс шифрования текста закончен. Расшифровывать текст надо будет как обычный файл.

3 Технико-экономическое обоснование

Цель данной дипломной работы заключается в разработке крипто-приложения с использованием нескольких метедов шифрования.

В данном дипломном проекте по разработке криптографической защиты будет участвовать группа специалистов, которая включает в себя: руководитель проекта, разработчик, бэкенд-программист, фронтенд-программист и тестировщик. В обязанности руководителя проекта входит соблюдение и разработка рабочих графиков, их контроль и оптимизация. В обязанности разработчика входит создание приложения и реализация шифрования. В обязанности бэкенд-программиста входит создание серверной части приложения. В обязанности фронтенд-программиста дизайн и удобства приложения. В обязанности тестировщика входит тестирование проекта на разных его стадиях для определения его качества и наличия ошибок. Технико-экономическое обоснование содержит следующие пункты:

- определение сложности разработки программного обеспечения;
- расчет затрат на разработку ПО;
- определение ценности готового продукта;
- оценка результатов работы программного обеспечения.

3.1 Определение трудоемкости разработки ПО

Последовательность создания программы и четкая проработка этапов – залог успеха всего проекта. Каждый этап создания программы – очень важный шаг, за который ответственен каждый разработчик из команды. Создание программы заключает в себе не только разработку дизайна и программу, но и детальный анализ проекта, сотрудничество с заказчиком и поиск решений для достижения поставленных целей проекта. В таблице 3.1 показана возможная поэтапная разработка сайта.

Таблица 3.1 – Этапы разработки ПО

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Построение алгоритма решения задач	15
Этап 2	Запись алгоритма на языке c++	15
Этап 3	Дизайн приложения в среде языка c++	10
Этап 4	Поиск и изучение соответсвующих программ и устройств	13
Этап 5	Поиск литературы по сети Фейстеля	30
Этап 6	Реализация программы в среде программирования	35

Этап 7	Отладка программного обеспечения	25
Этап 8	Оформление отчета	15
Этап 9	Тестирование приложения	20
Этап 10	Итог и вывод разработки программного продукта	30
Итого: трудоемкость выполнения проекта		228

228:8=29

Продолжительность рабочего дня равна 8 часам. В результате для реализации программного обеспечения необходимо 29 рабочих дней.

3.2 Расчет затрат на разработку ПО

Для определения затрат, которые необходимые при создании веб-приложения необходимо учитывать следующие элементы имеющейся сметы:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

К материальным затратам относятся затраты на материалы, энергию и другие затраты необходимые для разработки ПО. Расчет материальных затрат происходит по форме, предоставленной в таблице 3.2.

Таблица 3.2 – Затраты на материальные ресурсы

Наименование материала	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Бумага (1000 листов)	Упаковка	2	1 200	2 400
Чистящий набор для компьютера	Штук	1	1200	1200
Блокнот	Штук	1	500	500
Тонер	Штук	2	2 000	4 000
Магнитно маркерная доска 60х45см	Штук	1	6 000	6 000
Итого				14 100

Общую сумму, необходимую на материальные средства ($З_м$) можно рассчитать по следующей формуле:

$$З_м = \sum P_i * Ц_i, \quad (4.1)$$

где P_i - расход i-го вида материального ресурса, натуральные единицы;
$Ц_i$ - цена за единицу i-го вида материального ресурса, тг;
i - вид материального ресурса;
n - количество видов материальных ресурсов.

Для реализации программного обеспечения необходимы материалы на сумму 14 100 тенге.

Для разработки программного обеспечения будет использоваться ноутбук HP 250 G6, мощности ноутбука достаточно для выполнению поставленных задач.

Общую сумму, необходимую на материальные средства ($З_м$) можно рассчитать по следующей формуле:

$$З_м = \sum P_i * Ц_i, \quad (4.1)$$

где P_i - расход i-го вида материального ресурса, натуральные единицы;
$Ц_i$ - цена за единицу i-го вида материального ресурса, тг;
i - вид материального ресурса;
n - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 3.3.

Таблица 3.3 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	HP 250 G6	Штук	1	190 000	190 000
Принтер	Samsung SL-M2020 A4	Штук	1	38 670	3670
Модем	TP-Link TD-W8901N	Штук	1	8 000	8 000
Итого:					229 470

Зм = 14 100 + 229 470 = 243 570 (тг)

Для реализации программного обеспечения необходимы материалы на сумму 236 670тенге.

3.3 Расчет затрат на электроэнергию

Так как длч разработки ПП используется электрооборудование, то необходимо рассчитать затраты на электроэнергию.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$З_{эл.эн.обор.} = \sum M * K * S * T, \quad (4.3)$$

где М – потребляемая мощность, Вт;

К – коэффициент использования ($К_{исц}$ = 0,7..0,9);

Т – время работы;

S – тариф (1кВт/ч = 17,81 тг).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 3.4.

Таблица 3.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/кВтч	Сумма, тг.
Ноутбук	0,54	0,8	160	23,85	1 231
Модем	0,1	0,9	160	23,85	256,4
Принтер	0,4	0,9	16	23,85	102,6
Кондиционер	0,76	0,9	100	23,85	1 949,1
Освещение	0,3	0,7	160	23,85	598,4
Итого:					4137,5

$$З_{эл.эн.обор.} = 4137{,}5 \text{ (тенге)}$$

3.4 Расчет затрат на оплату труда

Для разработки программного обеспечения, как указывалось ранее, необходимо пять работников:

- руководитель;
- разработчик;
- бэкенд-программист;
- фронтенд-программист;
- тестировщик.

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$З_{тр} = \sum ЧС_i * T_i \quad (4.5)$$

где $ЧС_i$ - часовая ставка i-го работника, тг;

T_i - трудоемкость разработки модели, чел.×ч; i - категория работника;
n - количество работников, занятых разработкой ПП.
Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$\text{ЧС}_i = \frac{\text{ЗП}_i}{\text{ФРВ}_i} \quad (4.6)$$

где ЗП_i - месячная заработная плата i-го работника, тг;
ФРВ_i - месячный фонд рабочего времени i-го работника, час.

Месячная заработная плата руководителя равняется 215 000 тенге, месячная заработная плата дизайнера равняется 130 000 тенге, месячная заработная плата бэкенд-программиста равняется 145 000 тенге, месячная заработная плата фронтенд-программиста равняется 165 000 тенге, месячная заработная плата тестировщика равняется 100 000 тенге. Рассчитаем часовую ставку каждого работника согласно формуле (4.6):

$$\text{ЧС}_{\text{руководитель}} = \frac{215\,000}{29 * 8} = 926{,}72 \text{ тг/ч}$$

$$\text{ЧС}_{\text{дизайнер}} = \frac{130\,000}{29 * 8} = 560{,}34 \text{ тг/ч}$$

$$\text{ЧС}_{\text{бэкенд-прог.}} = \frac{145\,000}{29 * 8} = 625 \text{ тг/ч}$$

$$\text{ЧС}_{\text{фронтенд-прог.}} = \frac{165\,000}{29 * 8} = 711{,}2 \text{ тг/ч}$$

$$\text{ЧС}_{\text{тестировщик}} = \frac{100\,000}{29 * 8} = 431{,}03 \text{ тг/ч}$$

Расчеты затрат по оплате труда показаны в таблице 3.5.

Таблица 3.5. – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	Руководитель	120	926,72	100 000,00
Дизайнер	Дизайнер	60	560,34	25 000,00
Фронтенд-программист	Программист	90	625,00	56 250,00
Бэкенд-программист	Программист	90	711,20	56 250,00
Тестировщик	Программист	100	431,03	33 333,00
Итого:				270 833,00

3.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонта оплаты труда, поскольку в данном случае нет необходимости добавлять 1,5% на медицинскую страховку. Социальный налог можно рассчитать по следующей формуле:

$$С_н = (ФОТ - ПО) * 0{,}095 \quad (4.7)$$

где ПО - отчисления в пенсионный фонд, они составляют 10% от ФОТ.

$$ПО = 270\,833 * 0{,}1 = 27\,083{,}3 \text{ тенге}$$
$$С_н = (270\,833 - 27\,083{,}3) * 0{,}095 = 23\,156{,}22 \text{ тенге}$$

3.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$A_г = \frac{С_{об} * Н_а}{100} \quad (4.8)$$

где, $С_{об}$ – стоимость оборудования;
$Н_а$ – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_г = \frac{190\,000 * 25}{100} = 47\,500 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_г = \frac{47\,500 * 29}{365} = 3575{,}3 \text{ тенге}$$

Где 29 – это общее количество дней разработки. Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (3.7).

Таблица 3.7 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	190 000	25	47 500	3575,30
Принтер	38 670	25	9 667	768,06
Модем	8 000	25	2 000	158,90
Итого:			59 167	4 700,93

Смета расходов на разработку ПО.
На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице 4.8.

Таблица 4.8 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг
Затраты на оборудование и материальные расходы	243 670,00
Затраты на оплату труда	270 833,00
Социальные налоги	23 156,22
Затраты на электроэнергию	4137,50
Амортизация основных фондов	4 863,06
Прочие расходы	18 100,00
Итого по смете:	564 759,78

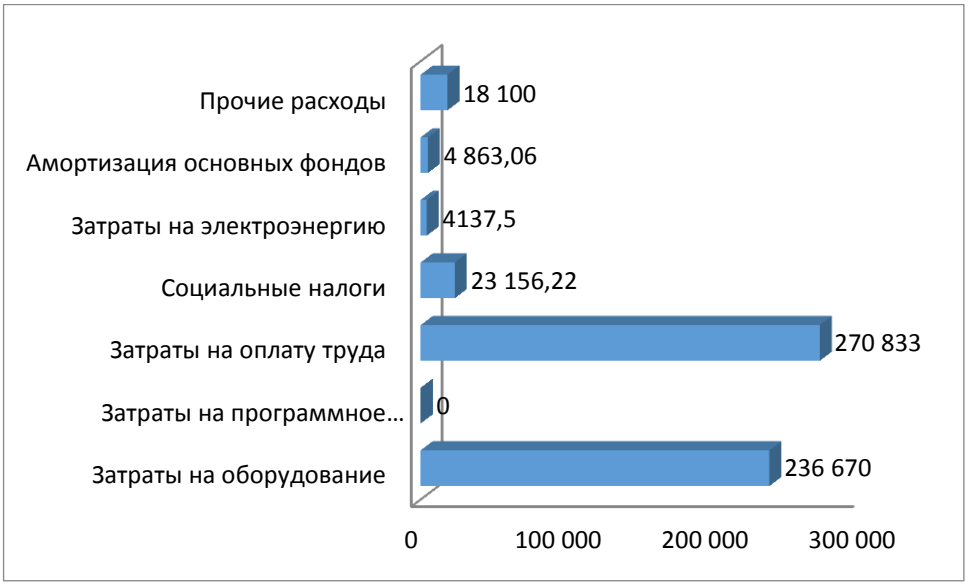


Рисунок 3.7 – Диаграмма затрат

3.7 Определение возможной (договорной) цены ПО

Прибыль программного продукта равна 25% от общей стоимости разработки приложения и может быть вычислен как:

$$П_{пр} = З_{нир} * 0,25 = \ 564\ 759,78 * 0,25 = 141\ 189,94 \text{ тенге}$$

Стоимость программного обеспечения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость $Ц_Д$ для программного обеспечения можно рассчитать по следующей формуле:

$$Ц_Д = З_{нир}\left(1 + \frac{P}{100}\right), \quad (4.9)$$

где $З_{нир}$ – затраты на разработку программного обеспечения, тг;

Р – средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

$$Ц_Д = 564\ 759,78 + 564\ 759,78 * 0,25 = 705\ 949,72 \text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации, учитывая НДС можно рассчитать по следующей формуле:

$$Ц_р = Ц_д + Ц_д * НДС, \quad (4.10)$$

$$Ц_р = 705\ 949,72\ + 705\ 949,72\ * 0,12 = 790\ 663,68 \text{ тг.}$$

В данной части дипломного проекта содержится экономические расчеты, которые показывают затраты необходимые при разработке выбранной программы по реализации криптографии. Расчеты включают в себя:

– определение трудоемкости разработки программного обеспечения;

– расчет затрат на разработку ПО;

– определение ценности готового продукта;

– оценка результатов работы программного обеспечения.

По выполненным расчетам можно сказать, что цена реализации выполненного проекта равна 790 663,683 тг с учетом НДС, что прибыль равна 141 189,95 и себестоимость проекта равна 564 759,78.

4 Безопасность жизнедеятельности

Физиология труда - это физиологическое явление, которое изучает механизмы и модели производственных процессов физиологических процессов человека, в частности принятие и регулирование трудовых процессов человека. Физиология труда ближе к здоровью и здравоохранению, чем психология труда. Знание физиологии человека проверяет машины, органы управления и контроля, принимая во внимание присутствие людей в процессе строительства. Кроме того, физиология труда дает представление о характеристиках цвета, музыки, шума, температуры и других показателей людей, что делает его более эффективным для организации безопасности.

Как известно, любой трудовой процесс можно разделить на операции, приемы, действия, движения.

4.1 **Рабочее помещение**

Основы физиологии

Как вы знаете, любой процесс труда, оборудование, деятельность, движения можно разделить на операцию.

Трудовые процессы классифицируются по следующим критериям:

1) по типу:

– Астан stalınatın;

– поддержка;

– Перед тем, как читать замены;

– что делать .

2) в соответствии с методом движения;

3) с точностью:

– это ;

– отрегулировать .

4) по функции:

– Основной;

– коррекция;

– кроме того;

– Чрезвычайное положение;

– ошибка.

Трудовая деятельность Это набор рабочих движений, например, карандаш или кусок работы, совместимый с одним или несколькими человеческими телами.

Важно учитывать коррекцию и устранение дополнительных действий при оптимизации. Трудовая деятельность делится по двум принципам.

1) по принципу универсальности:

- (приобретение, размещение, подъем);
- специальное;

2) по названию:

- сменная;
- подключи;
- управление оборудованием.

Труда и целесообразность разработки и производства рационализации отражающей единой системы саморегулирования в соответствии с законами комбинаций.

Рационализация рабочих движений является важным резервом роста производительности труда.

Любое рабочее движение можно описать с трех сторон :

– механическое движение, характеризуемое силой, скоростью, скоростью, траекторией;

– базовый, дополнительный, экстремальный, избыточный, силовой и т. д. психологические движения, которые можно охарактеризовать на основе разделения;

– Физиологическое, трудовое движение считается моторизованным рефлексом, где трудовая активность отражается в терминах рефлексов.

Двигатель может быть описан как совокупность взаимосвязанных компонентов двигательных реакций, требующих динамических и статических сил. Оценка эффективности трудовых движений учитывает их скорость, простоту внедрения и оптимальное использование энергии. Оптимизация трудовой миграции основывается на следующих принципах: активные и пассивные силы, плавность, непрерывность и ритм движения, траектория движущихся дорожек, нормальное движение движения, устранение нежелательных движений, ограничение статических напряжений и т. Д. Исследование трудового движения в первую очередь направлено на рациональное и эффективное проектирование труда, изучение и анализ темпа и ритма работы, разработку научных стандартов времени, основанных на нормировании микроэлементов.

При планировании рабочей области, нужно учитывать санитарные характеристики процессов, соблюдать нормы полезной площади для работы, так и соотношения областей техники и оборудования, и необходимой ширины проходов, обеспечения безопасной эксплуатации и удобной техники обслуживания и оборудование.

Рабочее помещение находится на втором этаже пятиэтажного здания:

а) размеры рабочего помещения: длина 4.5 м, ширина 4.5 м, высота 3м;

б) остекление помещения – 1 окно размером 2м × 2м выходящее юго-запад;

в) искусственное освещение – ЛПО 12-2×40-904 (производство PHILIPS);

г) внутренняя отделка стен – светлая;

д) помещение по зрительным условиям работы относится к IV разряду (размер различаемых при работе предметов от 1 до 10 мм и выше)

План помещения представлен на рисунке 1.

Общая площадь помещения 20,25 м2. Объем рабочего помещения равняется 60,75 м3, что обеспечивает необходимый объем на трех человек. Рассматривается рабочие помещения, расположенные в здании, которое не находится в непосредственной близости от железнодорожной магистрали или нагруженной автомагистрали, аэропорта и так далее, поэтому внешних источников шума, влияющих на процесс работы – нет.

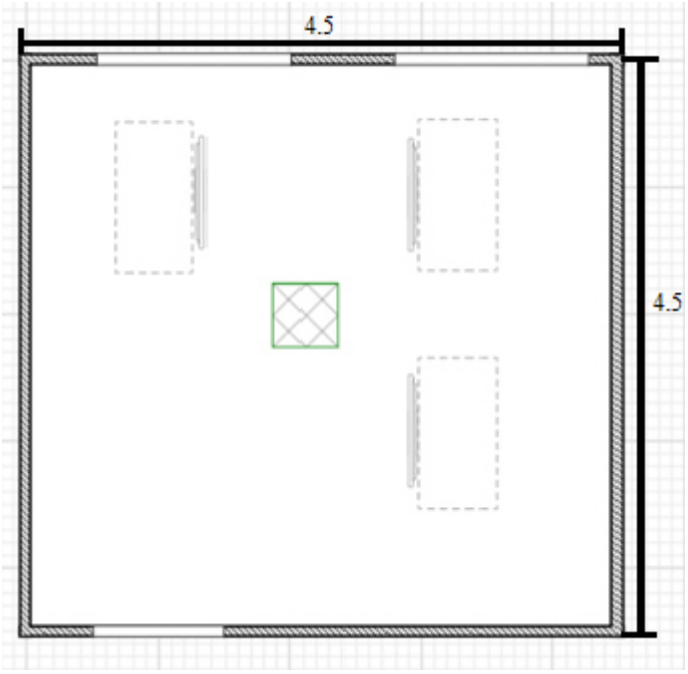


Рисунок 4.1 - План помещения

Характеристики используемого оборудования.

Модель системы создается и запускается на ноутбуке, сборка и припаивание элементов платы на паяльной станции, происходит в отдельной комнате на обычном столе. В помещении имеются 3 ноутбуки:

а) ноутбук HP Pavilion 15-p263ur (Intel Core i7 5500U 2.4Ghz/15.6"/1366x768/6Gb/750Gb/NVIDIA GeForce 840M/DVD-RW/Wi-Fi/Bluetooth/Win8.1);

б) электропитание: переменное напряжение 220-250 В, частотой 50-60 Гц., мощность 350 Вт;

в) электропитание: переменное напряжение 220 В, частотой 50 Гц., мощность 550 Вт;

г) ЛПО 12-2×40-904 (производство Samsung);

д) электропитание: переменное напряжение 220 В, частотой 50 Гц., мощность 65 Вт, напряжение на лампе 103 В.

Данное оборудование не обладает сильным шумовым воздействием. Вероятность возгорания ноутбуков или поражение током - мала, а для предотвращения пожара достаточно установленного огнетушителя.

Микроклиматические условия**.**

Микроклиматические условия действительно допустимы такие параметры микроклимата, в которых долгосрочное и систематическое

воздействие может вызвать временные изменения быстро и стандартизировать функциональное и тепловое состояние тела и напряженной работы механизма терморегуляции не выходит за пределы допустимой приспособляемости физиологические способности.

Решение этой проблемы лежит в следующих областях: управление планирования пространства и проектирования зданий, рационального расположения оборудования, механизации и автоматизации производственных процессов, дистанционного управления и мониторинга, создание процессов и эффективные производственные мощности, которая позволяет эффективная теплоизоляция оборудования, защита работников различных типов нагревательных экранов и адекватной вентиляции, рационализации труда и отдыха, использование СИЗ (средства индивидуальной защиты).

Здание относится к I степени огнестойкости (СНиП РК 2.02-05-2002) (Здания с несущими и ограждающими конструкциями из естественных или искусственных материалов, бетона или железобетона с применением листовых негорючих материалов). Операционная комната пожарной безопасности относится к классу «Д». В соответствии со стандартными правилами пожарной безопасности административного здания и отдельных помещений, а также технологических установок обеспечены первичными средствами пожаротушения в соответствии с нормами.

Естественное освещение не обеспечивает полного рабочего времени, необходимого освещения, так как погода может измениться, или может быть в более позднее время, когда она становится темно и естественное освещение может быть недостаточно, поэтому на рабочем месте предусмотрена система искусственного общего освещения, состоящий из ламп ЛПО. Климатические условия эксплуатации оборудования совпадают с климатическими условиями, нормированных для рабочего персонала.

В рабочем помещении трудятся 3 сотрудника (мужчины – 2, женщины – 1), включая меня, которые имеют служебные места.

Характеристики кабинета: длина L = 4.5 метров, ширина W= 4.5 метров, высота H = 3 метра. Присутствует окно площадью 4 м^2. Установлен старый кондиционер McQuay.

Характеристики используемого оборудования.

В помещении имеются 2 ноутбука и периферийные устройства:

а) ноутбук HP Pavilion 15-p263ur (Intel Core i7 5500U 2.4Ghz/15.6"/1366x768/6Gb/750Gb/NVIDIA GeForce 840M/DVD-RW/Wi-Fi/Bluetooth/Win8.1);

б) электропитание: переменное напряжение 220-250 В, частотой 50-60 Гц., мощность 350 Вт;

в) принтер HP LaserJet Pro P1102 (черно-белый/лазерный/настольный/349x196x233 мм);

г) электропитание: переменное напряжение 220 В, частотой 50 Гц., мощность 550 Вт;

Оборудования не представляют шумовую угрозу. Анализ условий труда показал, что слабым местом является вентиляция, в связи с этим в данном разделе производится расчет искусственной вентиляции.

Планировка помещения представлена на рисунке 4.2

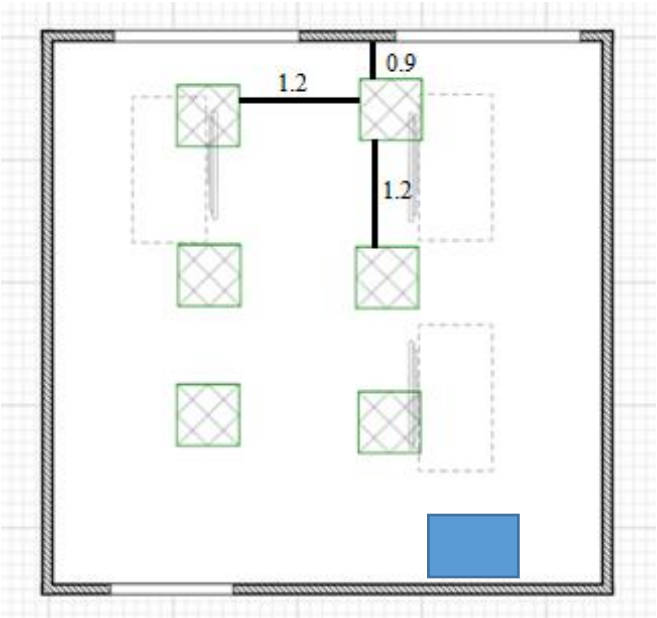


Рисунок 4.2 – Планировка рабочего помещения

4.1 Расчет тепловых нагрузок в помещении

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).

4.1.1 Наружные тепловые нагрузки.

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными. Теплопоступления и теплопотери в результате разности температур определяются по формуле:

$Q_{огр} = V_{пом} * X_о * (t_{Нрасч} – t_{Врасч})$, Вт (0.1), где

$V_{пом}$ – объем помещения, $м^3$;

$V_{пом} = 4.5*4.5*3=60.75$ $м^3$;

$X_о$ – удельная тепловая характеристика, $Вт/м^3*^0С$;

$X_о= 0,42$ $Вт/м^3*^0С$.

$t_{Нрасч}$ – наружная температура (параметр А). Для холодного периода – средняя температура самого холодного месяца в 14 часов, для теплого периода – средней температуре самого жаркого месяца в 14 часов.

$t_{Врасч}$ – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года:

$t_{Нрасч} = 29{,}4\ ^0C$

$t_{Врасч} = 26\ ^0C$

$Q_{огр}= 60{,}75*0{,}42\cdot 3{,}4=86{,}75$ Вт

Для холодного времени года

$t_{Нрасч} = -9\ ^0C$

$t_{Врасч} = 19\ ^0C$

$Q_{огр} = 60{,}75\cdot 0{,}42\cdot 28 = 714{,}42$ Вт

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле 0.2:

$Q_р = (q^I F^I_o + q^{II} F^{II}_o) * \beta с.з$ (0.2), где

q^I, q^{II} – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м^2;

F^I_0, F^{II}_0 – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м^2;

βс.з. – коэффициент теплопропускания. Для штор-жалюзи с металлическими пластинами:

βс.з. = 0,15

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F^I_0 = F^{II}_0 = F_0 = 0$:

$Q_р = q^{II} F_0 * \beta с.з = q_{вр} * K^T_1 * K_2 * \beta с.з * n * S_0$, где

$q_{вп}$; $q_{вр}$ – тепловые потоки от рассеянной радиации, Вт/м^2. Для широты в 44^0 СШ после полудня в 14-15 ч. при расположении ЮВ:

$q_{вр} = 63$ Вт/м^2;

$F_o = nS_o = 2*2 = 4$ м^2 – площадь светового проема (n – число окон; S_o – площадь 1 окна);

K_1 – коэффициент затемнения остекления переплетами (K^T_1 – для проемов в тени).

$K^T_1 = 1{,}28$;

K_2 – коэффициент загрязнения остекления:

$К_2$ = 0,95.

Тогда:

$Q_р$ = 63*1,28*0,95*0,15*4=45,96 Вт

Для широты в 44^0СШ после полудня в 14-15 ч. при расположении ЮЗ:

$q_{вр}$ = 101 Вт/м^2;

$F_о = nS_о$ = 2*2 = 4 м^2

Тогда:

$Q_р$ = 101*1,28*0,95*0,15*4=73,69 Вт

Тогда общее теплопоступление солнечного излучения с обеих сторон равно:

$Q_р$ = 45,96+73,69 = 119,65 Вт

4.1.2 Внутренние тепловые нагрузки

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла:

– выделяемого людьми;

– выделяемого лампами и осветительными, электробытовыми приборами;

– выделяемого компьютерами, печатающими устройствами.

Летом при 24 ^{0}С один мужчина выделяет явного тепла 67 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$Q_л^я$ = 67*2 + 67*1*0,85 = 190,95 Вт

А выделение общего тепла:

$Q_л^о$ = 102*2 + 102*1*0,85 = 290,7 Вт

Зимой при 18 ^{0}С один мужчина выделяет явного тепла 89 Вт, а общего – 104 Вт. Тогда выделение явного тепла в помещении составит:

$Q_з^я$ = 89*2 + 89*1*0,85= 253,65 Вт

А выделение общего тепла:

$Q_з^о$ = 104*2 + 104*1*0,85= 296,4 Вт

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле:

$$Q_{осв} = \eta \cdot N_{осв} \cdot F_{пол}, \quad Вт$$

где η – коэффициент перехода электрической энергии в тепловую (для лампы накаливания η=0,92-0,97);

$N_{осв}$ – установленная мощность ламп (N=36 Вт);

$F_{пол}$ – площадь пола:

$F_{пол}$ = 4,5*4,5=20,25 м^2

Тогда:

$Q_{осв}$ = 0,92*36*20,25 = 670,68

Тепло, выделяемое производственным оборудованием, определяется по формуле:

$$Q_{об} = N_{уст} \cdot K$$

$Q_{об} = 0,3 * 3 * 0,75 * 10^3 = 0,67$ кВт.

Теплопритоки, возникающие за счёт находящейся оргтехники – это 30% мощности оборудования:

$Q_{орг} = 3 * 0,3 * 0,3 * 10^3 = 0,27$ кВт

4.1.3 Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс теплопоступлений в помещении:

$$Q_{изб} = Q_р + Q^я + Q_{осв} + Q_{об} + Q_{орг} + Q_{огр}$$

Лето: $Q_{изб}^{Л} = 119,65 + 190,95 + 670,68 + 670 + 270 + 86,75 = 2,1$ кВт

Зима: $Q_{изб}^{З} = 119,65 + 253,65 + 670,68 + 670 + 270 + 714,42 = 2,7$ кВт

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле:

$$Q_н = \frac{Q_{изб.лето} \times 860}{V_{пом}}$$

$$Q_н = \frac{2,1 \cdot 860}{60.75} = 29,73 \text{ ккал/м3}$$

При $Q_н > 20$ккал/м^3, $\Delta t = 8$ °С,

при $Q_н < 20$ккал/м^3, $\Delta t = 6$ °С.

Определение количества воздуха, необходимое для поступления в помещение:

$$L = \frac{Q_{изб} \times 860}{C \times \Delta t \times \gamma}$$

$$L = \frac{0,27 \cdot 860}{0,21 \cdot 8 \cdot 1,206} = 114,60 \text{ м3/час}$$

где С=0,21 ккал/(кг · °С) – теплоемкость воздуха,

γ=1,206 кг/м^3 – удельная масса приточного воздуха.

4.2 Выбор кондиционера и схема расположения

Исходя из полученных результатов, для удаления лишнего тепла и очистки воздуха нужно использовать вентиляционную систему, которая способна обеспечить требуемую подачу воздуха L=114,60 (м3/ч). В данном случае подойдет Кондиционер MIDEA MDSA-09HRFN1 INVERTER. Данный кондиционер способен обеспечить подачу воздуха до 1200 м3 /ч.

Технические характеристики:

- мощность (охлаждение): 2.93 кВт ;
- мощность (обогрев): 2.93 кВт ;
- потребляемая мощность при охлаждении: 2200 Вт;
- потребляемая мощность при обогреве: 2240 Вт ;
- обслуживаемая площадь: 28 м2 ;
- уровень шума внутреннего блока: 37-41 дБ;
- уровень шума внешнего блока: 48 дБ;
- цвет: серый;
- характеристики подключения;
- вентиляция: 1200 м3/час;
- класс энергоэффективности при охлаждение/обогреве: А++/А+;
- электропитание, В/Гц/Ф:220 Вт;
- энергопотребление в режиме ожидания не более 1 Вт.

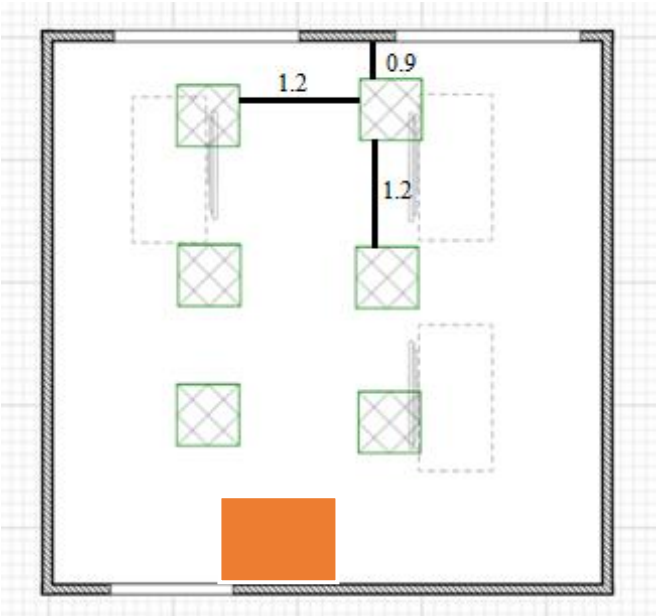


Рисунок 4.3 – Расположение кондиционера в помещений

В этом разделе моего дипломного проекта я рассмотрел и рассчитал анализ я рассмотрел и рассчитал воздушные показатели для благоприятных условий труда, а именно, тепловые нагрузки в помещении, наружные и внутренние. По расчетам, для создания хороших условий труда необходим один кондиционер с подачей воздуха не менее 114,60 м3 /ч, в моем случае используется кондиционер MIDEA AURORA 1 MSAB-24HRN1-WG с подачей воздуха до 1200 м3 /ч.

Заключение

За последние годы криптография и криптографические методы все шире входят в нашу жизнь и даже быт. Вот несколько примеров. Отправляя Email, мы в некоторых случаях отвечаем на вопрос меню: «Нужен ли режим шифрования?» Владелец интеллектуальной банковской карточки, обращаясь через терминал к банку, вначале выполняет криптографический протокол аутентификации карточки. Пользователи сети Интернет наверняка знакомы с дискуссиями вокруг возможного принятия стандарта электронной подписи для тех страниц, которые содержат «критическую» информацию (юридическую, прайс-листы и др.). С недавних пор пользователи сетей стали указывать после своей фамилии наряду с уже привычным «Email . . . » и менее привычное —«Отпечаток открытого ключа . . . ».С каждым днем таких примеров становится все больше.

Именно новые практические приложения криптографии и являются одним из источников ее развития.

В ходе дипломного проекта была реализована программа блочного симметричного шифрования. В основе алгоритма шифрования легла сеть Фейстеля. Достоинства сети заключатся в простоте реализации и масштабируемости. Алгоритм легко реализовывается как программно так и аппаратно.

Список литературы

1 Секунов Н Программирование на C++ в Linux. – СПб.: БХВ-Петербург, 2004.
2 Прата С. Язык программирования C++. Лекции и упражнения. – М.: Вильямс, 2012.
3 Земсков В.Ю. Программирование на C++ с использованием библиотеки Qt 4. – БХВ-Петербург, 2007.
4 Шлее М. Qt 4.5. Профессиональное программирование на C++.- БХВ-Петербург, 2010.
5 Иванов М.А. Криптография. Криптографические методы защиты информации в компьютерных системах и сетях.- М.: «КУДИЦ-Образ», 2010.
6 Щербаков А, Домашев А, Прикладная криптография. Использование и синтез криптографических интерфейсов.- М.: «Русская Редакция», 2003.
7 Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си.-М.: «Триумф», 2002.
8 Нечаев В. И. Элементы криптографии. Основы теории защиты информации. –М.:«Высшая школа», 1999.
9 Венбо М. Современная криптография. Теория и практика- М.: «Вильямс»,2005.
10 Вельшенбах М. Криптография на Си и C++ в действии. – М.:«Триумф»,2004.
11 Шилдт Г. C++: базовый курс. – М.: «Вильямс»,2008.
12 Романов Е. Л. Практикум по программированию на C++. - БХВ-Петербург, 2004.
13 Астахова И. Ф, Власов С. В , Фертиков В. В, Ларин А. В. Язык C++. Учебное пособие
14 Андерсон Дж. Дискретная математика и комбинаторика. - БХВ-Петербург, 2004.
15 Игошин В.И. Математическая логика и теория алгоритмов - М.: «Вильямс»,2009.
16 Аршинов М.Н ,.Садовский Л.Е. Коды и математика – М.: «Наука», 1983.
17 Wikipedia шифр Цезаря / URL:https://ru.wikipedia.org/wiki/Шифр_Цезаря (время посещения 5.05.2019)
18 Studrev.com шифр «Сцитала» / URL:https://studref.com/441633/informatika/shifr_stsitala (время посещения 10.06.19)
19 Wikipedia.org сеть Фейстеля / URL:https://ru.wikipedia.org/wiki/Сеть_Фейстеля (время посещения 11.04.19)