МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Кафедра информационных технологий**

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ НА ОСНОВЕ ДВОИЧНОГО РЮКЗАКА**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А.Финджоян

(подпись, дата) (инициалы, фамилия)

Факультет компьютерных технологий и прикладной математики 3 курс

Направление 01.03.02 – «Прикладная математика и информатика»

Научный руководитель, проф.

д-р. физ.-мат. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.О.Осипян

(подпись, дата) (инициалы, фамилия)

Нормоконтролер, ст. преп. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В.Харченко

(подпись, дата) (инициалы, фамилия)

Краснодар 2017

РЕФЕРАТ

Курсовая работа 46 с., 11 ч., 18 рис., 1 табл., 15 источников.

КРИПТОГРАФИЯ, АСИММЕТРИЧНОЕ ШИФРОВАНИЕ, СИММЕТРИЧНОЕ ШИФРОВАНИЕ, КРИПТОГРАФИЧЕСКАЯ УСТОЙЧИВОСТЬ, ЗАДАЧА О РЮКЗАКЕ, АЛГОРИТМ МЕРКЛА – ХЕЛЛМАНА, ШИФРОВАНИЕ

Объектом исследования является криптографическая система защиты информации на основе двоичного рюкзака.

Цель работы – ознакомление с основами криптографии, разработка системы защиты информации на основе двоичного рюкзака.

В результате исследования были рассмотрены основные понятия в криптографии, а также разработана система защиты информации, основанная на алгоритме Меркла – Хеллмана.

Разработанная система защиты информации позволяет шифровать/расшифровывать файлы данных, а также генерировать открытые и секретные ключи. Время работы программы определяется объемом входных данных. Программа имеет простой интерфейс, а для работы использует «алфавит», который состоит из заглавных букв английского алфавита, пробела, цифр 0 – 9, запятой, точки, восклицательного знака, вопросительного знака, а также служебного символа «\n». Каждый символ можно представить 6 – ю битами информации. При использовании правильно выбранной пары ключей шифрование/расшифровывание данных производится однозначно (это означает, что ключ, которым можно зашифровать информацию, должен быть получен сильным модульным умножением относительно секретного ключа; для расшифровки необходимо использовать соответствующий секретный ключ).

СОДЕРЖАНИЕ

[Введение 3](#_Toc500169897)

1. [История криптографии 4](#_Toc500169898)

[2. Современная криптография 6](#_Toc500169899)

[3. Методы защиты информации 8](#_Toc500169900)

[4. Аутентикация 9](#_Toc500169901)

[5. Целостность информации 10](#_Toc500169902)

[6. Виды шифрования 12](#_Toc500169903)

[6.1. Симметричное шифрование 12](#_Toc500169904)

[6.2. Асимметричное шифрование 16](#_Toc500169905)

[7. Ключи 21](#_Toc500169906)

[8. Криптографическая стойкость алгоритма 23](#_Toc500169907)

[9. Криптографическая система защиты информации 25](#_Toc500169908)

[10. Задача о рюкзаке и СЗИ на основе этой задачи 26](#_Toc500169909)

[10.1. История создания системы защиты информации 30](#_Toc500169910)

[10.2. Описание алгоритма 31](#_Toc500169911)

[10.3. Достоинства и недостатки СЗИ на основе двоичного рюкзака 34](#_Toc500169912)

[11. Реализация СЗИ на основе двоичного рюкзака 35](#_Toc500169913)

[Заключение 45](#_Toc500169914)

[Список использованных источников 46](#_Toc500169915)

# ВВЕДЕНИЕ

Целью данной курсовой работы является знакомство с основными понятиями криптографии, изучением методов сокрытия информации, а также создание системы защиты информации на основе двоичного рюкзака.

Будут рассмотрены такие понятия, как:

* Целостность информации;
* Аутентикация и способы аутентикации;
* Открытый ключ;
* Секретный ключ;
* Асимметричное шифрование;
* Симметричное шифрование;
* Криптографическая устойчивость алгоритма;
* Криптографическая система защиты информации.

Дополнительно будет описано несколько симметричных алгоритмов, применение задачи о двоичном рюкзаке для создания системы защиты информации, а также подробно рассмотрена одна из реализаций такой системы. Для нее будут обозначены ее достоинства и недостатки, а так же рассмотрена реализация данного метода шифрования/расшифровывания информации на языке программирования Python.

# История криптографии

Криптография как инструмент сокрытия информации существует уже очень давно[1]. Важно понимать, что главным критерием, по которому можно будет различать периоды развития науки о шифровании, является набор характеристик метода шифрования информации. Историю криптографии можно разделить на несколько этапов.

Для первого этапа (до начала XVI века) было характерно использование примитивных способов шифрования. Большинство из используемых шифров сводились к перестановкам, моноалфавитным подстановкам. Одним из первых зафиксированных примеров является шифр Цезаря: каждую букву исходного текста заменяли на другую, отстоящую от оригинальной в алфавите на несколько позиций.

Второй этап (конец XV века – начало XX века) – появление формализованных и стойких к ручному криптоанализу шифров. Обеспечение более высокой криптостойкости было ознаменовано появлением роторных криптосистем, позволявших автоматизировать процесс шифрования. Однако практическое распространение такие машины получили лишь в начале XX века. Одной из известнейших машин такого типа стала немецкая Enigma.

Третий этап (30-е – 60-е годы XX века) – этап, на котором стали появляться криптосистемы, чья криптостойкость была математически обоснована. Математика стала использоваться как новый фундамент для шифрования. В работах Клода Шеннона были определены понятия количества информации, энтропии, передачи данных, функций шифрования сообщения и т.д [2, с.10]. До 1975 года наука о сокрытии информации оставалась наукой, использовавшей в качестве основного инструмента секретный ключ.

Современная криптография (с 70-х годов XX века) появилась в результате технического прогресса. Главным инструментом в руках криптографов являлись ЭВМ, которые при большой скорости шифрования обеспечивали хорошую криптостойкость. На этом этапе широко стали применяться блочные шифры, применение которых стало возможным с появлением мощных вычислительных средств. Также этот этап отличается тем, что был разработан новый подход к способу кодирования и передачи сообщения – криптография с открытым ключом. Развитию в данном направлении поспособствовал скачок в развитии техники, а так же широкое распространение шифрования частными лицами. Наука о кодировке сообщений образовала отдельное ответвление в месте, где соединяется математика и информатика. Практическое применение криптографии весьма и весьма широко: сокрытие содержания передаваемых сообщений, защита банковских транзакций, защита баз данных предприятий и т.д.

# Современная криптография

С развитием технологий развивается и криптография, причем весьма и весьма быстро. Появилась возможность использования сложных вычислений для организации защиты информации (например, вычисления в поле Галуа в шифре AES[3]). Принципы выбора алгоритмов (или как их еще называют, криптографических примитивов) постепенно усложняются. Предъявляются новые требования к данным алгоритмам, зачастую совсем не связанные никаким образом с математикой, такие как устойчивость к атакам по сторонним каналам(side-channel attack – класс атак на криптографические системы, который направлен на эксплуатацию уязвимостей в практической реализации данного метода шифрования информации). Для решения задач предлагаются новые механизмы, как организационные, так и законодательные.

Отличительной особенностью этого периода является развитие принципиально новых направлений в криптографии. Речь идет о таких направлениях, как криптография с открытым ключом, квантовая криптография вместе с квантовыми вычислениями [4]. Квантовые компьютеры – вопрос времени, поэтому даже уже сейчас предложены алгоритмы для взлома уже существующих, считающихся надежными, систем (например, алгоритм Шора [5, с.86] – квантовый алгоритм факторизации(разложения числа на простые множители), позволяющий разложить число M за O(. При помощи алгоритма становится возможным взломать криптографическую систему с открытым ключом, к примеру, RSA: данная система использует в качестве открытого ключа произведение двух больших простых чисел. Один из способов его взломать – найти эти самые числа, но проблема в том, что числа эти довольно большие, и классические алгоритмы для поиска этих значений здесь не подойдут. С другой стороны, используя квантовую механику, можно построить и принципиально новые, не похожие на уже существующие способы надежной передачи информации. Активные исследования в этом направления начались в 80-х годах прошлого столетия. Также стало возможным использование вычислительных мощностей для криптоанализа. Это позволяет специалистам в области шифрования изучать алгоритмы, с помощью которых шифруется информация, выявляя недостатки определенных систем. Одной из новых задач в криптографии является разработка скоростных методов шифрования с высоким уровнем секретности. Эта задача обусловлена появлением новых каналов связи, таких как: беспроводные сети, сотовая связь, Интернет. По этим каналам передается все больше и больше информации, а вместе с этим растет и востребованность специалистов, способных предоставить действенный алгоритм защиты информации. Как бы ни росли со временем вычислительные способности у криптографов, достичь оптимальной стойкости криптографических систем при ограниченной скорости шифрования можно будет, лишь применяя при их разработке серьезные математические методы.

2. Методы защиты информации

Основная цель использования криптографии как инструмента – защита информации. Существуют различные способы защиты информации от посторонних глаз. Можно, к примеру хранить данные в каком-нибудь секретном месте (т.е. ограничить доступ физически), однако при передаче информации при таком способе хранения возникают некоторые трудности.

Также существуют другие способы защиты:

* Скрытие самого канала передачи сообщения;
* Использование нестандартного метода передачи информации;
* Маскировка сообщения: это может быть как размещение информации в ничем не примечательном для злоумышленника «ящике», так и обмен сообщениями по открытому каналу связи, смысл которых известен как получателю, так и отправителю ;
* Затруднение получения доступа к информации и ее перехвату: использование помехов, «мусора» внутри канала таким образом, чтобы злоумышленнику было трудно определить, что именно ему необходимо перехватить.

Криптография же, в свою очередь, не ставит своей целью «спрятать» передаваемое сообщение, но преобразовать его по какому – то определенному правилу так, чтобы у того, кто перехватывает сообщение во время передачи, возникли определенные трудности при расшифровке. Предполагается, что перехватчик не должен понять смысл сообщения [6,стр. 8], ровно, как и отправить сообщение от имени одного из участников «беседы» так, чтобы был незаметен факт подмены/изменения содержания. Это так называемая проблема аутентикации.

1. Аутентикация

Аутентикация – процедура проверки подлинности. В современных системах она реализована в самых различных формах, например:

* Сравнение введенного и уже существующего пароля;
* Проверка цифровой подписи (некоторый алгоритм криптографического преобразования информации с помощью секретного ключа, позволяющий проверить отсутствие искажения сообщения с момента формирования подписи);
* Проверка контрольной суммы передаваемого файла на соответствие сумме, заявленной отправителем.

Аутентикация, ровно, как и сама криптография, была одной из основных задач, которую необходимо было решать при передаче информации какого - либо рода. К примеру, замок и ключ были придуманы уже очень давно, а в современном мире и конструкция замков несколько иная, да и сами «замки» уже не такие, какими их представляли люди, жившие в Средневековье. Поэтому, помимо перечисленных ранее методов аутентикации, также существуют следующие виды проверки подлинности:

* Аутентикация по одноразовым паролям;
* Аутентикация с помощью SMS;
* Биометрическая аутентикация;
* Аутентикация через данные о местоположении;
* Многофакторная аутентикация.

Грамотное использование метода проверки подлинности очень сильно затруднит работу перехватчика сообщения.

Также весьма и весьма важной задачей является обеспечение целостности информации.

1. Целостность информации

Целостность информации – термин, означающий, что данные не были изменены при выполнении какой либо операции над ними, будь то передача или хранение. В некотором роде, аутентикация отчасти обеспечивает целостность информации, но лишь в том случае, если для информации был использован подходящий метод. Т.е. к примеру, если владеющий информацией человек поместил ее в сейф – в таком случае, аутентикация не спасет от изменения данных, но лишь от получения несанкционированного доступа к ним. А если злоумышленник получит шифр – он сможет спокойно изменять содержимое без нарушения аутентикации. Соответственно, для данного случая необходим несколько иной подход.

Процедура шифрования в криптографии не гарантирует целостность данных. Ведь шифрованием отправитель пытается добиться того, чтобы передаваемая информация не была понятна постороннему лицу. Но это не защищает ее от обработки посторонним лицом, от внесения в шифр «мусора» вплоть до расшифровки. Поэтому для проверки целостности используются дополнительные методы.

Под нарушением целостности информации понимают:

* Инверсию битов;
* Добавление новых битов (т.е. совершенно новых данных);
* Удаление битов;
* Изменение порядка следования битов.

Задача состоит в том, чтобы обнаружить не столько случайные изменения информации(спонтанные искажения), но целенаправленную обработку сообщения криптоаналитиком. Для решения этой задачи была введена мера имитостойкости шифра(вероятность того, что данные были изменены). Например, для определенного сообщения в его конец добавляется некоторая комбинация байт. Она вычисляется согласно определенным алгоритмам и позволяет проверить, были ли изменения внутри сообщения. Добавление такой проверочной информации называется имитовставкой.

Имитовставкой называется число, вычисляемое на основе содержимого сообщения, т. е. имитовставка является функцией сообщения.

Обозначим

M = f (x),

где M – имитовставка; f – функция, вычисляющая имитовставку; x – сообщение.

Имитовставку можно использовать как для проверки целостности сообщения, так и для проверки подлинности.

Таким образом, существуют четыре цели криптографии:

* Конфиденциальность – защита информации от лиц, не имеющих права доступа к ней;
* Целостность – гарантирование того, что информация дошла до пункта назначения нетронутой;
* Аутентикация – методика проверки подлинности отправителя и получателя, а так же самой информации;
* Невозможность отказа от авторства(или невозможность приписывания авторства) – предотвращение попыток получателя или отправителя отказа от некоторых действий, которые они совершили по отношению к пересылаемой информации.

1. Виды шифрования

В этом разделе будут рассмотрены два вида шифрования:

* Симметричное шифрование;
* Асимметричное шифрование.

Функциональные различия между ними велики, ровно, как и разница во времени между появлением подхода с использованием симметричного шифрования и асимметричного шифрования соответственно. Рассмотрим симметричную криптосистему.

* 1. Симметричное шифрование

Симметричным шифрованием называют такой способ шифрования, при котором для шифрования и расшифровывания применяется один и тот же ключ. Этот метод шифрования появился гораздо раньше, чем асимметричное шифрование, потому и был единственным. Так как ключ использовался для шифровки и расшифровки, то его утеря означала полную неспособность криптосистемы выполнять поставленную задачи:

* Хранение информации в секрете: раз ключ известен, то злоумышленник может без особых сложностей узнать, что было зашифровано;
* Обеспечение целостности: раз злоумышленник может вскрыть шифр, то ему не составит труда внести некоторые изменения в сообщение. Другое дело если у принимающей стороны будет некий алгоритм, по которому он сможет определить, было ли сообщение подделано или нет – в таком случае получатель будет знать, что информация была изменена;
* Обеспечение аутентикации: при наличии ключа постороннее лицо может получить доступ к информации, к которой у него доступа изначально не было;
* Невозможности отказа от авторства/приписывания авторства: К примеру, злоумышленник может подделать сообщение так, что получатель может подумать, что сообщение было получено от другого отправителя, не говоря уже о возможности изменить содержание.

Рассмотрим несколько примеров симметричных алгоритмов:

* Простая перестановка:

Один из простейших методов шифрования, суть которого состоит в том, что данные (сообщение) записываются, к примеру, в таблицу по столбцам. Этот открытый текст далее считывается построчно, образуя шифротекст. Получатель и отправитель должны договориться о размерах матрицы – ключа. Метод довольно гибкий в выборе метода, по которому заполняется таблица. Можно записывать в таблицу сообщение по строкам, а считывать по столбцам снизу вверх, начиная с самого правого столбца – это не имеет значения. Важно знать размеры матрицы и способ расшифровки;

* Одиночная перестановка по ключу [6,стр.46]:

Метод, очень похожий на предыдущий, за исключением того, что здесь будет происходить перестановка столбцов по некоторому ключевому слову, фразе или набору чисел длиною в строку в таблице.

После перестановки матрица считывается построчно. Если для шифрования использовалось ключевое слово, то для расшифровки также необходимо знать ключевое слово – оно определит количество столбцов в матрице, но помимо этого необходимо знать, в каком порядке переставить эти столбцы для того, чтобы расшифровать сообщение.

При использовании фразы или ключевого слова каждый символ помечается некоторым номером таким образом, чтобы при перестановке символы были отсортированы в алфавитном порядке, а соответствующие им номера – по возрастанию.

Рассмотрим работу алгоритма на примере.

Пример:



Рисунок 1 – Одиночная перестановка по ключу

Для примера, изображенного выше, ключ K = ВАЛКАР, каждый элемент которого сверху помечен цифрой. В матрицу вписано сообщение построчно. Для шифрования необходимо переставить буквы в ключе так, чтобы соответствующие им цифры были отсортированы по возрастанию, а вместе с ними и столбцы матрицы, тогда шифротекстом, прочитанным из матрицы построчно будет:

Е = ОКТЬЛО Д\_\_ЯЛС УНТЕДТ ВПОФ\_М КГ\_БУУ.

* Двойная перестановка:

Метод основан на простой перестановке. Его суть заключается в том, что шифровать можно и то сообщение, которое было зашифровано. Для этого подбирается вторая матрица, но так, чтобы длины строк и столбцов были взаимно простыми (желательно) с длинами строк и столбцов в первой матрице. В первой таблице допускалась перестановка столбцов матрицы, а во второй – строк. Также вариативности добавляло то, что можно было заполнять таблицу каким – то определенным способом (т.е. определить правило распределения значений по матрице);

* «Магические квадраты»:

Квадратные таблицы с натуральными числами с 1, которые в сумме по строке, столбцу и диагонали дают одно и то же число. Текст записывался по нумерации внутри таблицы, а затем выписывалось содержимое таблицы по строкам – получалась перестановка букв. Ранее система считалась довольно хорошей, так как квадраты больших размеров не позволяли совершить перебор ключа.

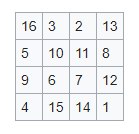


Рисунок 2 – «Магический квадрат» размерами 4 х 4



Рисунок 3 – «Магический квадрат» с сообщением

«ПриезжаюСегодня.»

Главным требованием для симметричных шифров является сильное изменение шифроблока при однобитном изменении входных данных – так называемый «лавинный эффект». Суть его проста: изменение нескольких битов входных данных вызывает большие изменения в выходных битах шифротекста. Иначе говоря, от каждого входного бита зависят все выходные биты зашифрованного сообщения.

В настоящее время симметричные шифры разделяют на 2 категории:

* Блочные шифры – обрабатывают информацию блоками определенной длины(измеряемой в битах), применяя к каждому блоку ключ. Операция повторяется некоторое количество раз;
* Поточные шифры – шифруют каждый бит(байт) открытого текста, используя гаммирование («наложение» последовательности случайных чисел на открытый текст).

Среди достоинств симметричных шифров можно отметить их скорость работы, простоту реализации, меньшую длину ключа для сопоставимой с асимметричными системами стойкость, а также изученность(за счет довольно долгой истории существования).

Недостатками же является сложность управления и обмена ключами: необходимо решить проблему передачи ключа для расшифровки по надежному каналу.

* 1. Асимметричное шифрование

Асимметричное шифрование (криптография с открытым ключём) – методика шифрования, в которой участвует пара ключей: открытый и закрытый ключи. У каждого из них свое предназначение. Открытый ключ передается по открытому каналу и используется для зашифровки сообщения, а также для проверки электронных подписей. Для генерации этих самых электронных подписей и расшифровки сообщения как раз и используется закрытый ключ. Само понятие появилось совсем недавно, в отличии от симметричного шифрования, которое существует уже не одно тысячелетие.

Идея асимметричного шифрования состоит в следующем: за основу была взята идея использования односторонних функций, которая предполагает, что существует функция f(x), причем по известному x можно найти значение функции, однако обратное действие практически невозможно.

В криптографии это означает, что есть некоторый алгоритм (односторонняя функция [6,стр.14]), который способен зашифровать сообщение, однако определить, что именно было зашифровано не представляется возможным. Криптография с открытым ключом предполагает существование некоторого секрета, который помог бы расшифровать сообщение. Другими словами, существует такой y, что зная f(x) и y, можно вычислить x.

Разберем, как можно реализовать идею использования односторонней функции на примере хранения паролей. Предположим, есть некоторый компьютер, который хранит данные о пользователях. У каждого пользователя есть учетная запись, которая хранит различные данные о нём. Для доступа к учетной записи необходимо указать имя и пароль. Пароль хранится в явном виде где то на жестком диске. Естественно, такой подход не является хорошим с точки зрения безопасности. Поэтому предлагается рассмотреть несколько иной подход к хранению паролей – использование односторонней функции.

Пусть есть некоторая односторонняя функция, которая формирует некоторый результат, основываясь на введенном имени и пароле. Этот результат сохраняется в памяти компьютера. В дальнейшем при попытке получить доступ к аккаунту будет произведена проверка на совпадение результата вычисления односторонней функции от данных аргументов(имени и пароля) и уже сохраненных результатов вычисления той же самой функции. Если совпадение есть, то это означает что имя и введенный пароль присутствуют в базе имен и паролей, а значит необходимо предоставить доступ к соответствующей учетной записи.

Такой подход позволяет не сохранять сами пароли, а лишь результаты вычисления односторонней функции от данных аргументов. По этому результату невозможно будет восстановить точное значение входных параметров. Однако функция должна быть однозначной(т.е. какому то одному аргументу она ставит в соответствие некоторый вывод), а также должна формировать абсолютно другой результат, если различия между двумя разными входными параметрами минимальны. Речь идет о «лавинном эффекте».

Данная организация не предполагает наличие «лазейки», поэтому расшифровка практически невозможна. Рассмотрим схему шифрования с открытым ключом в общем понимании.

Пусть существуют:

* Ключевое пространство K;
* Ключи шифрования и расшифровывания e и d соответственно;
* Пространство зашифрованных сообщений C;
* Пространство сообщений M;
* Функции шифрования и расшифровывания E и D соответственно.

Тогда обозначим

где – функция шифрования сообщения ключём e, результатом вычисления которой является шифротекст c.

Аналогично, обозначим

где – функция расшифовки шифротекста ключём d, результатом вычисления которой является исходное сообщение m.

В таком случае, зная и c, невозможно решить уравнение, т.е. найти такое m, которое бы удовлетворяло уравнению.

Графически можно изобразить следующим образом:

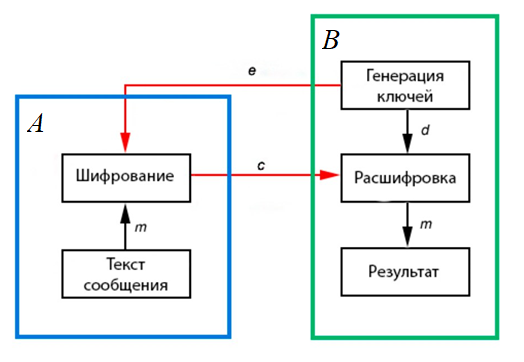


Рисунок 4 – Схема работы асимметричного шифрования

Стрелками e и c обозначается передача по открытому каналу, в рамки обведены участники диалога. Пусть, в диалоге участвуют два собеседника: собеседник А и собеседник В. Каждый из них способен совершать определенные действия.

В общем случае, диалог происходит по следующей схеме:

1. А запрашивает у B открытый ключ по незащищенному каналу связи;
2. А шифрует сообщение: ;
3. B получает сообщение и расшифровывает его: .

Рассмотрим преимущества и недостатки асимметричного шифрования:

Преимущества:

1. Отпадает нужда в поиске хорошо защищенного от атаки канала связи, т.к. мы передаём лишь открытый ключ, который не поможет в расшифровке;
2. Только лишь одной стороне известен ключ зашифровки, который держится в секрете. Для сравнения, в симметричной криптографии такой ключ должен быть известен обеим сторонам диалога.

Недостатки:

1. Меньшая скорость, по сравнению с симметричным шифрованием;
2. В дополнение к предыдущему пункту, асимметричное шифрование требовательно к вычислительным мощностям.

1. Ключи

Одним из фундаментальных понятий в криптографии является ключ. Ключ – некоторая информация, критически важная для работы криптографического алгоритма. С помощью ключа производится шифровка/расшифровка алгоритма, устанавливается и проверяется цифровая подпись, необходимая для проверки подлинности сообщения. Ключ как один из самых важных компонентов следует хранить в секрете, т.к. для современных систем защиты информации утрата ключа приводит к невозможности расшифровки сообщения, а перехват ключа ведет рассекречиванию зашифрованной информации.

Существует также принцип Керкгоффса – это правило разработки криптографического алгоритма, при котором алгоритм должен оставаться безопасным, если злоумышленнику известно о нем все, за исключением ключей шифрования. Иначе говоря: нельзя считать, что злоумышленник не сможет узнать всё о работе вашего алгоритма. Впервые принцип сформулировал голландский криптограф Огюст Керкгоффс.

Если рассмотреть принцип поподробнее, то его суть заключается в том, что количество секретов алгоритма обратно пропорционально безопасности алгоритма. Объясняется это тем, что если у алгоритма есть какой то секрет, то его утрата может привести к полному разрушению системы шифрования, т.к. злоумышленник будет обладать информацией, которой он обладать не должен, ввиду того, что эта информация – критически важная для корректной работы алгоритма, а значит, вероятно содержит в себе подсказку/прямой способ обхода системы и взлома.

Принцип Керкгоффса [7] – один из шести требований Керкгоффса к криптосистемам. Сейчас эти требования и определяют криптографически стойкие системы.

Шесть требований Керкгоффса [7]:

1. Система должна быть физически невскрываемой;
2. Необходимо, чтобы сама система не хранилась в тайне; попадание системы в руки противника не должно причинять неудобств;
3. Хранение и передача ключей должна осуществляться без помощи записей; должна существовать возможность менять ключ по своему усмотрению;
4. Система должна быть пригодной для сообщения через телеграф;
5. Система должна быть легко переносимой, работа с ней не должна требовать участия нескольких лиц одновременно;
6. Требуется, чтобы система была проста в применении, не затрачивала много умственных ресурсов и не заставляла соблюдать большое количество правил.

Второе требование и стало Принципом Керкгоффса.

Возвращаясь к ключам, стоит отметить, что довольно долго использовались криптосистемы с закрытым ключом, и лишь в XX – веке стали появляться криптосистемы, в которых использовался открытый ключ.

Чуть ранее были рассмотрены симметричные и асимметричные криптографические системы, а вместе с ними и типы ключей, которые данные системы используют. Помимо них, существуют так называемые сеансовые ключи: они создаются парой между двумя пользователями, рассчитаны на один сеанс связи и предназначены для защиты канала передачи информации.

Еще один вид ключей – эфемерные ключи. Эфемерные ключи являются практически полной противоположностью статическим ключам (таким, которые планируется использовать долгое время без изменения самого ключа). Создаются каждый раз при создании нового соединения. Такие ключи можно использовать несколько раз в одной сессии, например для каждого отдельного сообщения в потоке.

2. Криптографическая стойкость алгоритма

Криптографической стойкостью алгоритма называется способность алгоритма противостоять криптоанализу. Чем больше ресурсов затрачивает криптоаналитик на попытки взлома, тем более надежным считается алгоритм. Иначе говоря, стойким алгоритмом является тот, атака на который потребует недостижимое количество ресурсов, либо настолько значительных затрат по времени, что к моменту расшифровки информация уже окажется ненужной. Чаще всего доказать стойкость с точки зрения математики не удается; в случае асимметричных систем можно доказать его уязвимость или свести задачу взлома к задаче, причем сам взлом не будет легче решения данной задачи.

Рассмотрим несколько категорий криптостойких систем:

* Абсолютно стойкая система: подразумевается, что алгоритм не может быть вскрыт даже при наличии у взломщика бесконечно больших ресурсов, выделенных на поиск решения. Существует ряд требований к таким системам, среди которых: генерация ключа для каждого сообщения, статистическая надежность ключа(шансы появления каждого из возможных символов равны между собой, а сами символы – случайны), длина ключа больше либо равна длине сообщения, а исходный текст обладает некоторой избыточностью(для оценки качества расшифровки). Очень хорошим примером является Шифр Вернама(или одноразовый блокнот). В 1945 году Клод Шеннон доказал, что Шифр Вернама является абсолютно стойким [8,стр. 57-58]. Однако за такую стойкость приходится платить удобством использования, постоянным соблюдением правил работы алгоритма и т.д.;
* Достаточно стойкая система: потенциальная возможность вскрытия шифра существует, но лишь при определенных обстоятельствах, таких как выбор определенного ключа, например. На практике решение данной задачи не является возможным, т.к. технологии на данный момент не способны предоставить необходимые вычислительные мощности для вычисления шифра за разумное время, к тому же не исключается возможность того, что информация потеряет актуальность к моменту взлома.

1. Криптографическая система защиты информации

Криптографическая система защиты информации (сокращенно СЗИ) – общий комплекс мер по обеспечению конфиденциальности информации, а также сохранения ее целостности. Также систему защиты информации можно определить как множество отображений пространства сообщений в пространство всевозможных криптограмм.

Математическую модель[9] системы защиты информации можно представить следующим образом:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Разберем каждый элемент этой модели более подробно. Пусть – множество всех сообщений (открытых текстов) над буквенным/числовым алфавитом M (непустое упорядоченное множество символов). Здесь – множество элементарных сообщений. Ими могут быть как отдельные буквы, так и конкатенация букв из алфавита M. - множество всех криптограмм системы. – алгоритм прямого преобразования сообщения m в шифротекст s; – алгоритм обратного преобразования шифротекста s в сообщение m. Алгоритмы связаны между собой при помощи . Данная связь подразумевает, что любое сообщение однозначно преобразовывается в шифротекст при помощи , а любой шифротекст однозначно преобразовывается в оригинальное сообщение при помощи .

1. Задача о рюкзаке и СЗИ на основе этой задачи

Задача о рюкзаке – это задача, на основе которой криптографы Ральф Меркл и Мартин Хеллман реализовали алгоритм шифрования с открытым ключом[10, гл.19.2]. Он носит название «Криптосистема Меркла-Хеллмана». Для шифрования сообщения использовалась задача о рюкзаке. Считалось, что она может обеспечивать криптостойкость, однако была взломана; к тому же, на данный момент существует ряд более криптоустойчивых алгоритмов.

Постановка задачи о рюкзаке[11, c.160]:

Задан набор из n различных положительных целых чисел – веса вещей. Также задано некоторое число k – вместимость рюкзака. Каждая вещь имеет ценность . Требуется найти такое подмножество А, чтобы в сумме элементы этого подмножества давали ровно k, а суммарная ценность была наибольшей.

В асимметричных системах защиты информации используется пара ключей: открытый E и закрытый D. Получатель сам выбирает пару ключей, в отличие от симметричных систем. Оба ключа математически тесно связаны между собой, однако вычисление ключа D по ключу E без какой либо дополнительной информации – весьма трудная в вычислительном плане задача [6, стр.56].

Пусть E – открытый ключ, D – закрытый ключ, тогда

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

аналогично

|  |  |  |
| --- | --- | --- |
|  | , | (3) |

где f – функция получения ключа E по ключу D;

– функция получения получения ключа D по ключу E.

(2) – легковыполнимая задача (т.е. получить ключ шифрования из ключа расшифровки легко)

(3) – Задача трудновыполнимая(т.е. зная ключ шифровки, получить ключ расшифровывания практически невозможно)

Таким образом, речь идет об односторонней функции f.

Принцип асимметричного шифрования лег в основу ряда алгоритмов, например:

* RSA (Rivest – Shamir – Adleman) – алгоритм, основанный на факторизации большого числа при неизвестных простых множителях; выделяют несколько реализаций данного алгоритма (стандартный алгоритм DSA, многопользовательский вариант DSA, полиномиальный вариант DSA);
* Алгоритм Диффи – Хеллмана – алгоритм, позволяющий двум и более сторонам получить секретный ключ, который в дальнейшем используется для шифрования/расшифровки;
* DES (англ. Data Encryption Standard) - стандарт шифрования данных в США[10, гл. 12]. Шифрует данные блоками, размер которых – 64 бита. Результатом шифрования также является блок шифротекста длины 64 бита.

Разберем алгоритм шифрования с помощью задачи о рюкзаке.

Будем называть

упорядоченным набором из n предметов. Для того, чтобы зашифровать некоторый текст, необходимо представить его в двоичной форме и разбить на блоки длины n (для примера, блок 01100 соответствует рюкзаку, в котором находится 5 предметов). Будем считать, что значение 0 указывает на то, что предмет с порядковым номером, соответствующим порядковому номеру данного 0, отсутствует в рюкзаке. Аналогично и для значения 1: предмет, с порядковым номером, соответствующим порядковому номеру данной 1, находится в рюкзаке.

Для примера рассмотрим следующий открытый текст m = 01100 и рюкзачный вектор A = (3,6,8,16,20).

Процедура щифровки осуществляется двумя способами:

Шаг 1: Шифротекст получается перемножением элементов рюкзачного вектора, возведенных в степень соответствующих им элементов из открытого текста:

Такой способ называется мультипликативным.

Шаг 2: Шифротекст получается суммированием элементов рюкзачного вектора, умноженных на соответствующие элементы открытого текста:

Такой способ называется аддитивным.

Как правило, результат работы алгоритмов отличен друг от друга, хотя бывают случаи, когда они совпадают. Рассмотрим пример использования аддитивного алгоритма получения шифротекста для разных вариантов открытого текста. Рассмотрим на примере применение аддитивного способа шифрования для одного и того же рюкзачного вектора:

Таблица 2 – аддитивный способ шифрования

|  |  |  |  |
| --- | --- | --- | --- |
| Открытый текст | 01100 | 01001 | 10011 |
| Рюкзачный вектор | 3,6,8,16,20 | 3,6,8,16,20 | 3,6,8,16,20 |
| Шифротекст | 14 | 26 | 39 |

Сложность состоит в том, что здесь есть две проблемы: «легкая» и «трудная». «Лёгкой» называется проблема, такая, что известен закрытый ключ: тогда расшифровка сообщения не составит особого труда. «Трудной» называется проблема, такая, что известен открытый ключ: в таком случае возможна зашифровка сообщения, но не расшифровка. При отсутствии закрытого ключа необходимо решать «трудную» проблему.

Рассмотрим методы решения задачи для «легкой» и «трудной» проблем:

а) Задача о рюкзаке – легко решаема, если рюкзачный вектор –сверхрастущий.

Рюкзачный вектор называется сверхрастущим, если:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Иначе говоря, вектор – сверхрастущий, если каждый новый элемент больше суммы всех предыдущих.

В таком случае решение задачи следующее:

Пусть P – общая вместимость рюкзака. Необходимо взять наибольший элемент из рюкзачного вектора, и если он по значению меньше чем P, то этот элемент берется в учет, а от общей вместимости необходимо отнять значение выбранного элемента – получим «свободное место». Далее необходимо взять следующий наибольший элемент из рюкзачного вектора и повторить операцию со свободным местом в рюкзаке. Когда свободного места в рюкзаке не останется(т.е. P = 0), то это и будет означать, что решение найдено;

б) Задача о рюкзаке – трудно решаема, если рюкзачный вектор не является сверхвозрастающим. Единственный способ решения данной задачи – перебор всех возможных вариантов. Самый быстрый алгоритм имеет экспоненциальную зависимость от числа предметов внутри рюкзачного вектора.

* 1. История создания системы защиты информации

Задача о рюкзаке лежит в основе первого алгоритма асимметричного шифрования (или иначе шифрования с открытым ключом). Идея криптографии с открытыми ключами была выдвинута американскими криптографами Уитфилдом Диффи, Мартином Хеллманом и независимо от них Ральфом Мерклом. Впервые она была представлена Диффи и Хеллманом на Национальной компьютерной конференции в 1976 году, в том же году была опубликована их совместная работа на эту тему — «New Directions in Cryptography»[12]. Новизна по отношению к распространённым на тот момент симметричным криптосистемам заключалась в использовании парных ключей — закрытого и открытого, создаваемых пользователем. Закрытый ключ, используемый для расшифровки информации, пользователь должен скрывать, тогда как открытый ключ, необходимый лишь для шифрования, может быть общедоступным.

Первый алгоритм для шифрования на основе задачи о рюкзаке был разработан Мерклом и Хеллманом в 1978 году и получил название «Алгоритм Меркла-Хеллмана»[13]. После опубликования схемы Меркл предложил вознаграждение в 100 долларов тому, кто удачно осуществит взлом. В 1982 году Шамир осуществил успешную атаку и получил обещанное вознаграждение.

В дальнейшем было придумано множество модификаций рюкзачной системы, например рюкзак Шора – Ривеста(основан на алгебре в полях Галуа), алгоритм Гудмана – Маколи(имеет схожесть с алгоритмом RSA, поскольку основан на сложности целочисленной факторизации), и т.д.

* 1. Описание алгоритма

Принцип, по которому работает алгоритм Меркла – Хеллмана, весьма и весьма схож с классической задачей о рюкзаке. Но что делает реализацию особенной, так это подход к формированию открытого ключа. Основан он на использовании модульной арифметики. К тому же, важным было то, что эта система была пионером среди асимметричных систем шифрования, благодаря использованию этой самой модульной арифметики. Ведь главной задачей до создания криптосистемы Меркла – Хеллмана была задача по безопасной передаче ключа шифрования, которая была успешно решена, тем самым был продемонстрирован один из вариантов реализации такого рода систем.

Перейдем к описанию самого алгоритма:

Пусть – закрытый ключ, – открытый ключ.

Введем обозначение:

Тогда запись означает наименьший неотрицательный остаток от деления x на m(применимо для x и m 2), а запись – целая часть от x [14, c. 103].

Дополнительно, выберем два числа m и t, такие, что m подчиняется условию

|  |  |  |
| --- | --- | --- |
|  |  | (5) |
|  |  |  |

а для числа t, в свою очередь, выполняется условие

|  |  |
| --- | --- |
|  | (6) |

Причем наибольший общий делитель чисел t и m равен 1. Обозначим это условие следующим образом: НОД(t, m) = 1.

Если – такой вектор, что:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

то говорят, что вектор B получен из А сильным модульным умножением относительно множителя t и модуля m [14, с. 103]. Условие НОД(t,m) = 1 должно выполняться, т.к. для расшифровки потребуется число s, мультипликативно обратное множителю t по модулю m, такое, что:

Этап шифрования:

Дано n – битное сообщение , . Для его зашифровки необходимо проделать следующие действия:

Шаг 1: Вычислим число C, которое является шифротекстом:

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

Шаг 2: Данное число затем передаётся получателю.

Стоит заметить, что подразумевается следующая ситуация: отправитель сообщения знает только открытый ключ, который получает по секретному каналу от получателя. Получатель же, в свою очередь, знает закрытый ключ и пару значений t и m. Получая шифротекст, он начинает расшифровку.

Этап расшифровки:

Шаг 1: Вычислить:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Шаг 2: Для имеем:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

А это - простая задача: пусть – наибольший элемент в А, тогда:

* Если > : ;
* Если : ;
* Вычитаем из произведение , повторяя шаги до тех пор, пока не вычислим W.

Замечание. Задача расшифровки является простой в том случае, если криптоаналитику известна пара значений t и m. Так как вычислено суммированием компонент W и B, а компоненты B, в свою очередь, образованы получением остатка от деления произведения t и , т.е.:

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

Тогда получим:

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

что доказывает корректность расшифровки.

* 1. Достоинства и недостатки СЗИ на основе двоичного рюкзака

Достоинства:

* Первая система, основанная на асимметричном шифровании;
* Система проста для понимания;
* Попадание самой системы (алгоритма) в руки криптоаналитика не повредит надежности шифра;
* На основе данной системы был создан ряд других криптосистем, некоторые из них очень трудно взломать даже сейчас (напр. ECDSA).

Недостатки:

* Уязвимость к LLL – алгоритму;
* Лазейка внутри открытого ключа: в докладе израильского криптоаналитика Ади Шамира было доказано [15], что из открытого ключа можно восстановить значения t и m, необязательно совпадающие с оригинальными, с помощью которых можно получить сверхвозрастающую последовательность, также необязательно совпадающую с оригинальной, которую можно было использовать для расшифровки сообщения;
* Медленная скорость работы в сравнении с симметричными шифрами.

1. Реализация СЗИ на основе двоичного рюкзака

Поставленная задача: разработать систему защиты информации на основе двоичного рюкзака. Данный алгоритм реализован на языке Python, с использованием PyQt – набора привязок графического фреймворка Qt для языка программирования Python. Также были использованы библиотеки:

* NumPy – библиотека, необходимая для произведения научных вычислений, работы с n – мерными массивами и преобразованиями в линейной алгебре;
* Random – встроенная в Python библиотека, позволяющая работать со случайными значениями.

Целью было создание небольшой программы, способной:

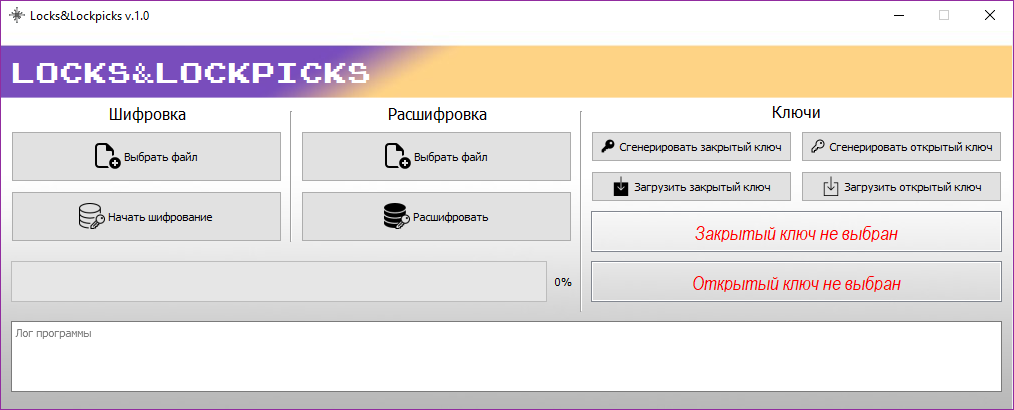
* Работать с файловой системой;
* Вести журнал событий;
* Предоставлять инструмент генерации закрытых ключей;
* Создавать открытый ключ на основе закрытого ключа;
* Производить основные операции: шифрование и расшифровывание файлов в рамках алгоритма;
* Иметь понятный интерфейс.

Рисунок 5 – Основной интерфейс программы

Работа всей программы основана на алгоритме Меркла – Хеллмана, но с некоторыми обобщениями. Рассмотрим её функционал поподробнее:

* Шифрование файла происходит следующим образом: вначале необходимо выбрать открытый ключ, затем выбрать сам файл, который следует зашифровать, затем выбрать место сохранения результирующего файла с шифром:

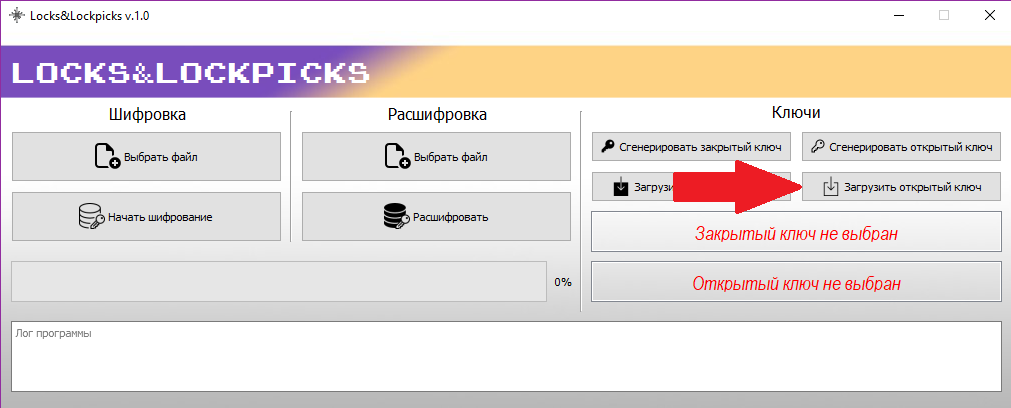


Рисунок 6 – Выбор открытого ключа

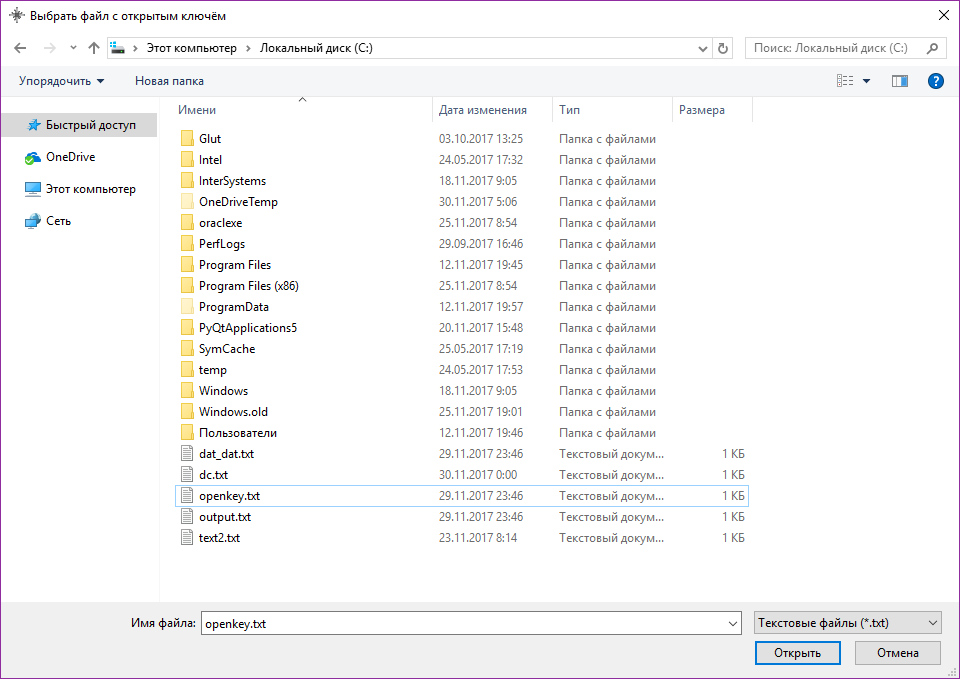
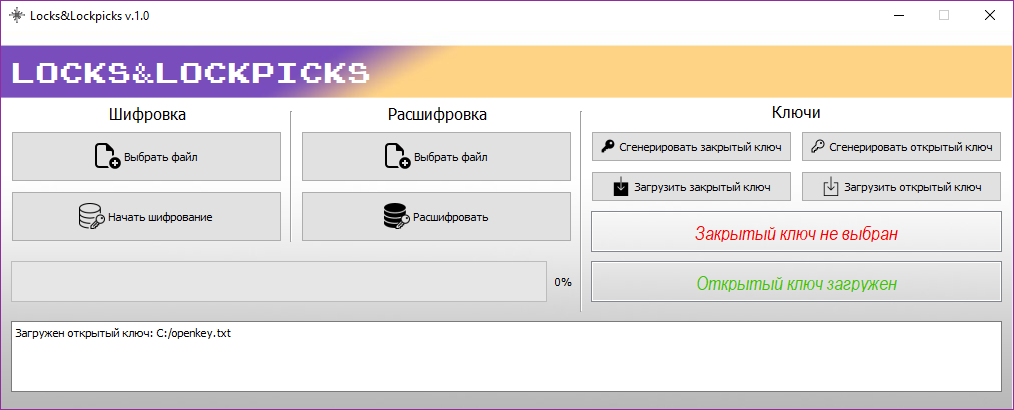


Рисунок 7 – Диалоговое окно выбора ключа

Рисунок 8 – Окно программы после выбора открытого ключа

Аналогично, после нажатия на кнопку «Выбрать файл» и выбора файла с данными:

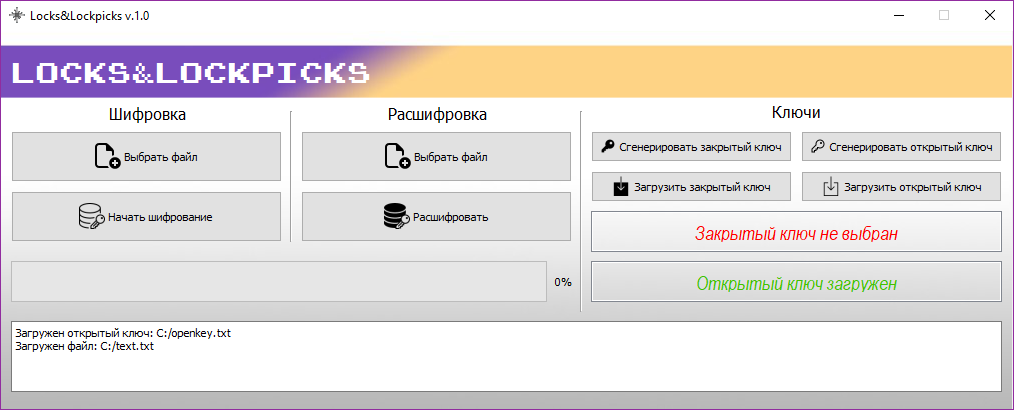


Рисунок 9 – Окно программы после выбора файла с данными

После данного шага следует нажать кнопку «Начать шифрование». Если все требования соблюдены, а именно был выбран файл с данными и загружен открытый ключ (о чем свидетельствует зеленая надпись в правой части окна программы), то при нажатии на «Начать шифрование» будет открыто диалоговое окно, в котором нужно будет выбрать место сохранения результирующего файла.

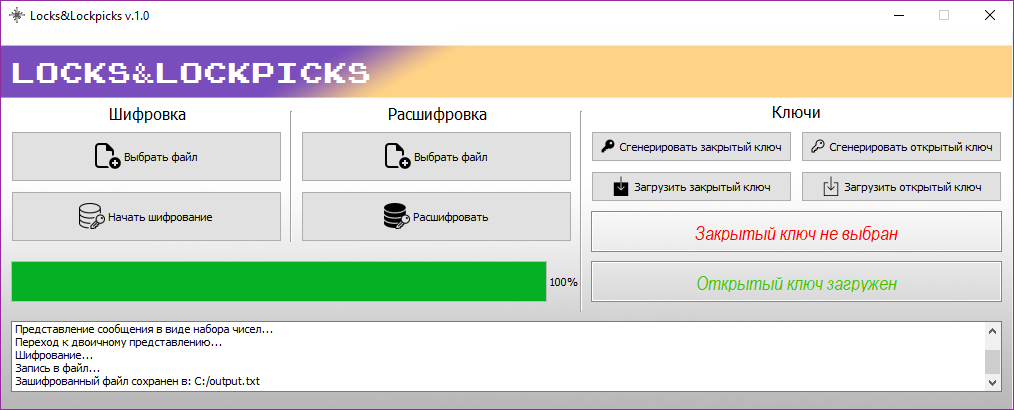


Рисунок 10 – Окно программы после процедуры шифрования

Замечание. Время работы программы определяется объемом входных данных. Данная реализация алгоритма поддерживает файлы с данными, «алфавит» которых состоит из заглавных букв английского алфавита, пробела, цифр 0 – 9, запятой, точки, восклицательного знака, вопросительного знака, а также служебного символа «\n», необходимого для переноса строки. Само шифрование проходит в несколько этапов. При выборе ключа программа определяет, по сколько символов она должна шифровать за один проход. Будем называть их группами. Длина открытого ключа кратна 6 – минимально необходимое количество бит для записи любого символа из «алфавита». Соответственно, если, к примеру, открытый ключ состоит из 12 элементов, то шифровать программа будет по 2 символа за один проход(т.е. группа в данном случае – пара символов, таких пар всего 6 штук: 1-2, 3-4, 5-6, 7-8, 9-10,11-12).

После этого каждому символу из группы ставится в соответствие число, которое представляет данный символ внутри «алфавита»: цифры 1 – 27 представляют буквы A – Z, 0 – пробел, 27 – «,», 28 – «.», 29 – «!», 30 – «?», 31 – «\n», 32 – 41 – «цифры 0 – 9».

Каждое такое число из группы затем переводится в двоичную форму, а затем на ее основе вычисляется шифротекст – некоторая сумма(алгоритм шифрования n-битного сообщения был разобран в пункте 5.2, стр. 29)

Но т.к. мы зашифровываем не все сообщение сразу, то вычисляется сумма для каждой группы символов.

Пример.

Дан шифротекст: BASIC TEXT FOR ENCRYPTION5, открытый ключ: 21650,120,240,4834,16079,12805,10491,20982,1081,8573,17146,8408.  
Пусть, ключ может шифровать по 2 символа за раз (данный параметр регулируется при создании нового ключа), тогда мы получаем следующее представление для сообщения в форме чисел:

2, 1, 19, 9, 3, 0, …

Далее – перевод в двоичную форму:

'000010', '000001', '010011', '001001', '000011', '000000', …

Т.к. открытый ключ может шифровать по 2 символа за раз, а один символ шифруется 6 битами, то длина ключа = 12 элементов.

Объединяя битовое представление в группы по 12 символов, получаем:

'000010000001', 010011001001', '000011000000', …

Каждая такая группа – сообщение W, которое шифруется по формуле:

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

На выходе получаем шифротекст, но т.к. групп таких 12-битных последовательностей несколько, то и сумм будет несколько.  
Тогда шифротекстом будет являться последовательность сумм, для данного примера:

24487, 38493, 28884, …

Данная последовательность сумм и будет записана в файл.

* Расшифровка: необходимо выбрать закрытый ключ, затем выбрать файл, который является шифротекстом, а после этого указать, куда следует сохранить расшифровку:

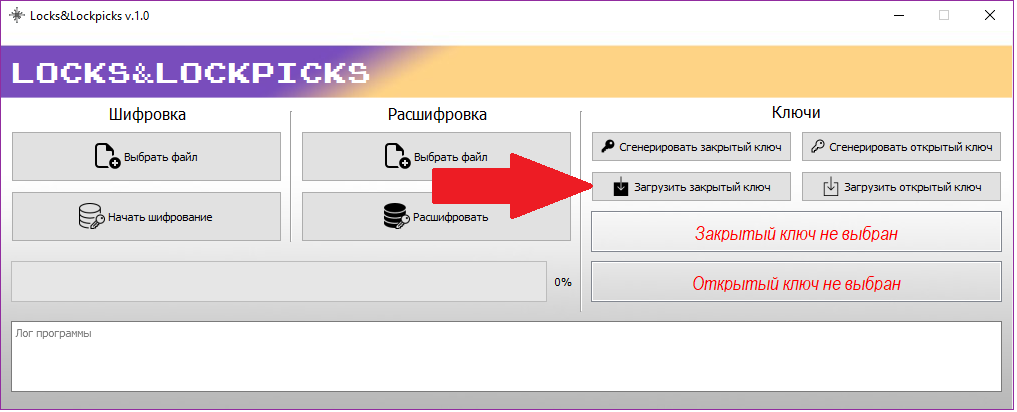


Рисунок 11 – Загрузка закрытого ключа

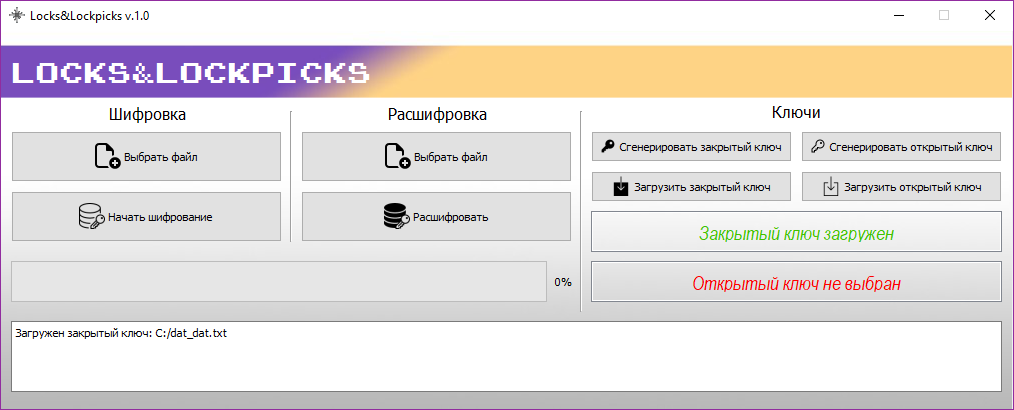


Рисунок 12 – Окно программы после загрузки закрытого ключа

Далее, после этого действия следует выбрать файл, содержащий шифротекст.

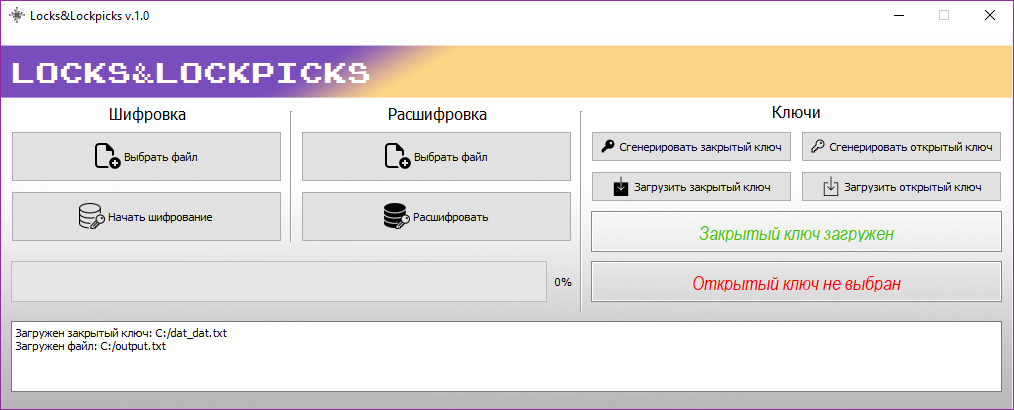


Рисунок 13 – Окно программы после загрузки зашифрованного файла

При нажатии на кнопку «Расшифровать» программа проверит, загружен ли файл закрытого ключа и файл с зашифрованным текстом. Если они загружены, то будет выведено диалоговое окно, в котором будет предложено выбрать место, в которое следует сохранить расшифрованное сообщение.

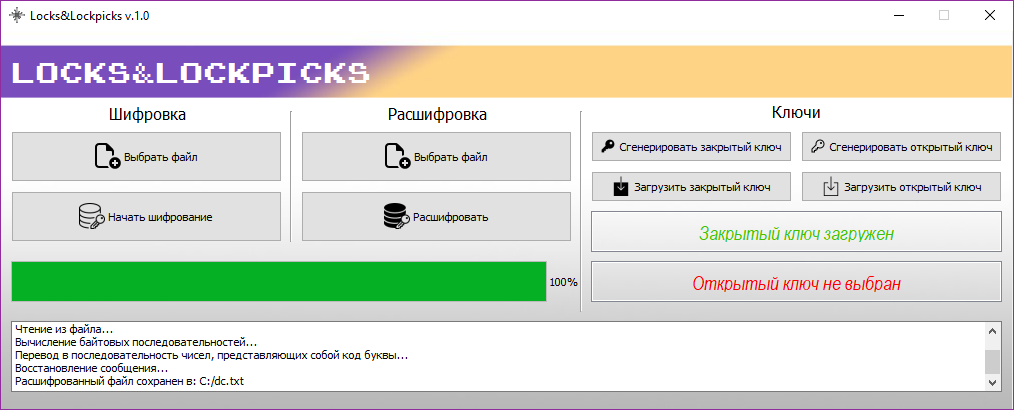


Рисунок 14 – Окно программы после расшифровки

Шифротекстом является последовательность сумм. Программа будет брать каждую сумму, а затем по готовому алгоритму (п. 10.2., стр. 30-31) восстановит изначальное сообщение.

Замечание. Количество итераций на данном этапе совпадает с количеством сумм в шифротексте. Время работы программы определяется объемом входных данных.

* Генерация ключей: программа умеет генерировать закрытый ключ шифрования, а также открытый ключ на основе закрытого. Процедура генерации открытого ключа состоит в создании сверхвозрастающей последовательности, в которой каждый элемент больше чем сумма всех предыдущих. Реализован алгоритм с помощью модуля random следующим образом: выбирается случайное натуральное число от 1 до 10, далее запускается цикл, который работает, пока не будет создан ключ определенной длины (длина ключа = количество шифруемых за раз символов × количество бит, необходимых для зашифровки одного символа(в программе этот параметр равен 6, как было показано при рассмотрении алгоритма шифрования(стр. 35)).  
  С каждой итерацией в последовательность добавляется элемент, больший, чем сумма всех предыдущих элементов (выбирается случайное значение, чуть большее, чем сумма элементов последовательности). Далее берется пара случайных чисел q и r, причем q подбирается таким образом, чтобы обеспечить выполнение условия:

Число r, в свою очередь, выбирается из отрезка:

Если была подобрана такая пара q и r, что наибольший общий делитель этих двух чисел = 1(т.е. числа являются взаимно простыми), то алгоритм создания ключа завершает работу, формируя закрытый ключ из последовательности, а также q и r.

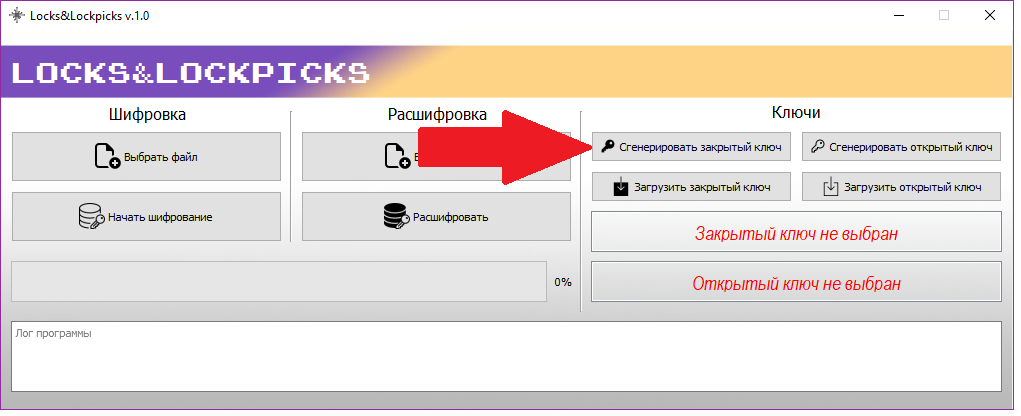


Рисунок 15 – Генерация закрытого ключа

После нажатия на кнопку «Сгенерировать закрытый ключ» откроется новое окно, в котором будет предложено выбрать количество символов, шифруемых за раз. Кнопка «Сохранить...» вызовет диалоговое окно, в котором будет предложено выбрать место, куда следует сохранить файл ключа.

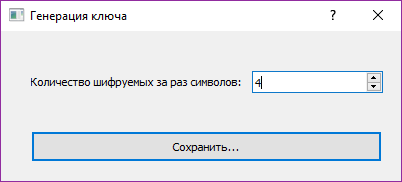


Рисунок 16 – Выбор длины закрытого ключа

Генерация открытого ключа возможна только после загрузки закрытого в программу.

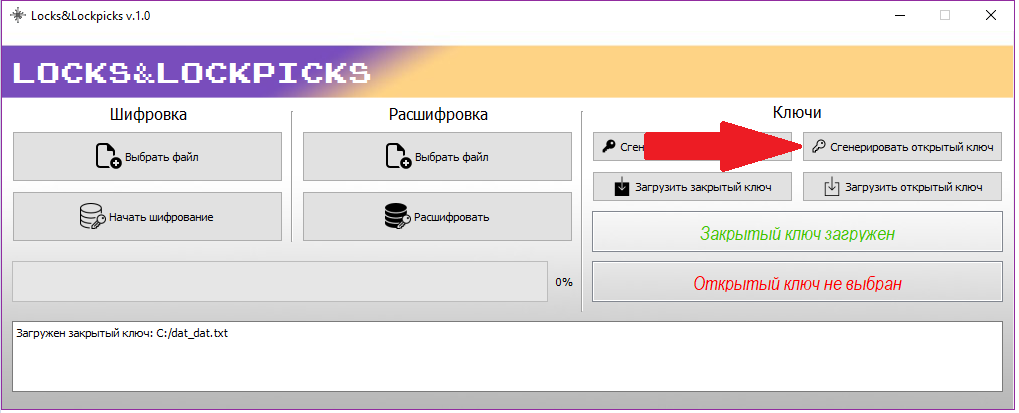


Рисунок 17 – Создание открытого ключа на основе закрытого

После нажатия на «Сгенерировать открытый ключ» откроется диалоговое окно, в котором будет предложено выбрать место, куда следует сохранить файл ключа.

Программа имеет ряд проверок файлов ключа и данных на корректность содержимого: если, к примеру, пользователь пытается загрузить файл ключа, в котором могут встречаться символы букв(заглавных, строчных), запятые, точки и т.д., то операция загрузки будет отменена, а в логе программы будет выведено сообщение о том, что файл содержит недопустимые символы.

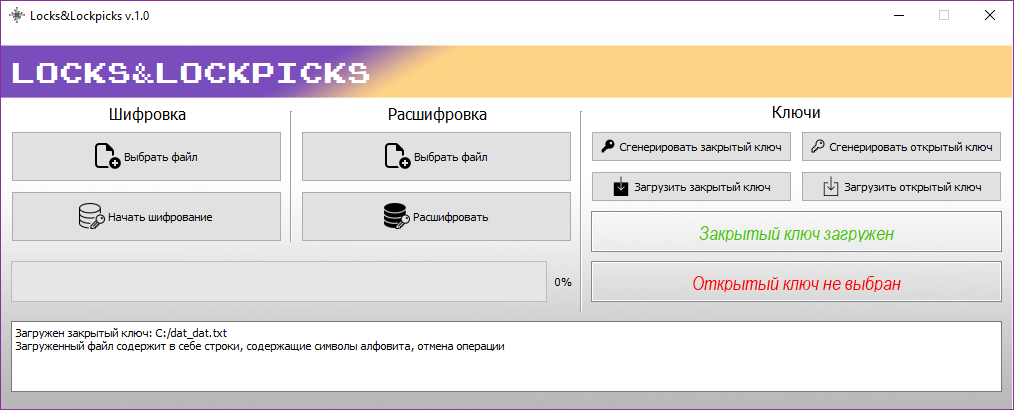


Рисунок 18 – Отмена загрузки файла ключа по причине наличия в нем недопустимых символов

# ЗАКЛЮЧЕНИЕ

Нами была разработана система защиты информации на основе двоичного рюкзака, способная шифровать и расшифровывать файлы данных, используя алгоритм Меркла – Хеллмана. Реализация алгоритма очень проста, однако скорость его работы ниже, чем скорость работы алгоритмов, основанных на симметричном шифровании. В перспективе будет рассмотрена реализация системы защиты информации на основе «обобщённого рюкзака», которая является модификацией двоичного рюкзака. Криптоустойчивость такой системы на порядок выше СЗИ на основе двоичного рюкзака.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Баричев С. Г., Гончаров В. В., Серов Р. Е. История криптографии // Основы современной криптографии — М.: Горячая линия — Телеком, 2002. — 175 с.
2. Shannon, C. A Mathematical Theory of Communication / C. Shannon // The Bell System Technical Journal — 1948.  
   – URL: http://affect-reason-utility.com/1301/4/shannon1948.pdf. (Дата обращения: 16.11.2017).
3. Баричев С. Г., Гончаров В. В., Серов Р. Е. 2.4.2. Стандарт AES. Алгоритм Rijdael // Основы современной криптографии — 3-е изд. — М.: Диалог-МИФИ, 2011. — С. 30–35. — 176 с.
4. Килин С. Я. Квантовая криптография: идеи и практика[Текст] / С.Я. Килин, Д.Б. Хорошко, А.П. Низовцев. — Беларусская навука. — 2007. — 391 с.
5. Валиев К. А., Кокин А. А. Квантовые компьютеры: надежды и реальность. — Ижевск: РХД, 2004. — 320 с.
6. Осипян, В.О. Криптография в упражнениях и задачах [Текст] / В.О. Осипян, К.В. Осипян — М.: Гелиос АРВ, 2004. — 144 с.
7. Kahn, David (1996), The Codebreakers: the story of secret writing (Abriged version) [Электронный ресурс]/ — 1996. —URL: http://mindguruindia.com/wp-content/uploads/2014/06/MP069\_The-CodeBreakers.pdf. (дата обращения: 12.11.2017)
8. Shannon, C. A Mathematical Theory of Cryptography [Электронный ресурс] / C. Shannon — 1945. — 1 September. –URL: https://www.iacr.org/museum/shannon/shannon45.pdf. (Дата обращения: 16.11.2017).
9. Осипян, В.О. Математическая модель системы защиты информации на основе диофантова множества / В.О. Осипян, А.В. Мирзаян, Ю.А. Карпенко, А.С. Жук, А.Х. Арутюнян // Чебышевский сборник. — 2014. —Т. 15. —№ 1. — С. 146-154.
10. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си [Текст] / Б. Шнайер — М.: Триумф, 2002. — 816 с.
11. Левитин А. В. Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — 576 с.
12. Diffie W., Hellman M. E. New Directions in Cryptography // IEEE Trans. Inf. Theory / F. Kschischang — IEEE, 1976. — Vol. 22, Iss. 6. — P. 644–654. — ISSN 0018-9448 — URL: doi:10.1109/TIT.1976.1055638
13. Ralph Merkle and Martin Hellman, Hiding Information and Signatures in Trapdoor Knapsacks // Information Theory. — 1978. —T. 24. —№5.   
    — C. 525–530.
14. Саломаа, А. Криптография с открытым ключом [Текст] / А. Саломаа — М.: Мир, 1995. — 318 с.
15. Adi Shamir, A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem. [Электронный ресурс]. –1982 . –C. 279–288. –URL: https://web.archive.org/web/20050424093849/http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/C82/279.PDF. (Дата обращения: 18.11.2017)