

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550386)

[1 Характеристика объекта и предмета исследования 5](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550387)

[1.1 Описание терминологии, связанной с предметной областью рассматриваемой задачи. 5](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550388)

[1.2 Принцип работы 6](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550389)

[1.3 Виды асимметричных алгоритмов 6](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550390)

[1.4 Алгоритм RSA 7](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550391)

[1.4.1. Описание криптосистемы 7](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550392)

[1.4.2. Стандарты RSA 8](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550393)

[1.5 Описание информационных технологий, используемых для решения рассматриваемой задачи 9](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550394)

[1.5.1. ViaCrypt PGP Business Edition 4.0C 9](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550395)

[1.5.2. RSA Secure 1.1.1 10](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550396)

[2 Реализация программы «Шифрование данных с помощью асимметричного ключа» 11](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550397)

[2.1 Постановка задачи 11](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550398)

[2.1.1 Сущность задачи 11](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550399)

[2.1.2 Описание исходной (входной) информации 11](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550400)

[2.1.3 Описание результатной (выходной) информации 11](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550401)

[2.1.4 Описание алгоритма решения задачи 11](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550402)

[2.2 Информационно-технологическая схема 12](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550403)

[2.3 Интерфейс программы 15](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550404)

[2.4 Программный код с комментариями 17](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550405)

[2.4.1 Код для MainForm 17](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550406)

[2.4.2 Код для GenForm 21](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550407)

[2.4.3 Код для пользовательского открытия формы 26](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550408)

[2.5 Представление пользовательского интерфейса 27](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550409)

[ЗАКЛЮЧЕНИЕ 34](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550410)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 35](file:///Q:\Desktop\Трегуб_Д.А_курсовая_Итог.docx#_Toc104550411)

# ВВЕДЕНИЕ

Применение криптографии позволяет эффективно решить проблему обмена информацией через открытые сети. Чаще всего встречающимися криптографическими средствами, обеспечивающими безопасность, являются шифрование, аутентификация паролем и цифровые подписи.

Современная криптография – широкая область знаний, которая сложилась в результате усиленных исследований на протяжение последних четырех десятилетий. Он включает в себя поиск решений основных задач информационной безопасности – целостности и контроля участников, конфиденциальности и аутентификации.

Актуальность данной темы заключается в необходимости работы с приватной информацией и её защитой от несанкционированного доступа. В современном мире такой информации и предпосылок с ней работать очень много: Это могут быть денежные транзакции, обмен секретными документами по сети Internet, конфиденциальными сообщениями в мессенджерах, а также данными в сетях повышенной опасности и стратегической важности, таких как сети электростанций, предприятий или военных объектов. Также данная программа будет важна и в научных целях, для теоретических исследований проблем безопасности и надёжности систем. Например, зная алгоритм шифрования, научный работник может попытаться найти в ней уязвимости, чтобы устранить их и не дать воспользоваться ими злоумышленникам.

Объектом исследования курсовой работы является сфера информационной безопасности.

Предметом исследования являются процессы и методы криптографии, связанные с асимметричным шифрованием.

Целью курсовой работы является разработка программы шифрования данных с использованием асимметричного ключа.

К задачам курсовой работы относятся:

* Поиск и изучение актуальной информации и литературы в соответствии с выбранной темой;
* Исследование алгоритмов асимметричного шифрования;
* Отбор и анализ наиболее подходящего алгоритма шифрования;
* Разработка программы по шифрованию данных с предоставления пользователю возможности создания ключа шифрования;
* Формулировка выводов;
* Написание программы на языке VBA;
* Создание интуитивно понятного пользовательского интерфейса;
* Отладка и тестирование программы;

Основанием выбранной темы для курсовой работы является её соответствие по варианту в списке.

В качестве источников информации для разработки программы шифрования данных с использованием асимметричного ключа будут использоваться различные учебники и веб-сайты с теоретической информацией по данной теме.

# Характеристика объекта и предмета исследования

* 1. Описание терминологии, связанной с предметной областью рассматриваемой задачи.

Асимметричное шифрование для кодирования и расшифровки использует два ключа. Открытый известен всем пользователям системы, а закрытый доступен только владельцу. Они связаны между собой математическим образом и составляют пару. Информация шифруется открытым ключом, расшифровать ее может только владелец закрытого.

В асимметричных системах другой стороне передается открытый ключ, который позволяет шифровать, но не расшифровывать информацию. Если подробнее, то шифрование с закрытым ключом (разновидность асимметричного шифрования, асимметричного шифра) — система шифрования и/или электронной подписи (ЭП), при которой открытый ключ передаётся по открытому (то есть незащищённому, доступному для наблюдения) каналу и используется для проверки ЭП и для шифрования сообщения. Для генерации ЭП и для расшифровки сообщения используется закрытый ключ.

* 1. Принцип работы

Асимметричное шифрование основано на парах чисел. Одно из этих чисел — открытый ключ, который доступен всем. С помощью этого числа кто угодно может зашифровать сообщение. Но расшифровать его с помощью этого же числа не получится.

Для расшифровки берут второе число — закрытый ключ. Он должен быть секретным.

Это не могут быть два случайных ключа. Открытый и закрытый ключ всегда связаны между собой алгоритмом, который их выдаёт. Смысл в том, что внутри этого алгоритма есть третье, тоже секретное, число, которое связано с обоими ключами.

Самый простой способ установить такую связь — взять два больших простых числа и перемножить их. Мы получим ещё большее число, которое и будет лежать в основе нашего алгоритма. А внутри этого алгоритма будет такая математика, которая зависит от разложения чисел на множители. Если мы не знаем ни одно из первоначальных простых чисел, то разложить на множители такое огромное число будет очень сложной задачей.

* 1. Виды асимметричных алгоритмов

Наиболее распространенные алгоритмы асимметричного шифрования:

* [RSA](https://encyclopedia.kaspersky.ru/glossary/rsa/) (аббревиатура от Rivest, Shamir и Adelman, фамилий создателей алгоритма) — алгоритм, в основе которого лежит вычислительная сложность факторизации (разложения на множители) больших чисел. Применяется в защищенных протоколах SSL и TLS, стандартах шифрования, например в PGP и S/MIME, и так далее. Используется и для шифрования данных, и для создания цифровых подписей.
* DSA (Digital Signature Algorithm, «алгоритм цифровой подписи») — алгоритм, основанный на сложности вычисления дискретных логарифмов. Используется для генерации цифровых подписей. Является частью стандарта DSS (Digital Signature Standard, «стандарт цифровой подписи»).
* Схема Эль-Гамаля — алгоритм, основанный на сложности вычисления дискретных логарифмов. Лежит в основе DSA и устаревшего российского стандарта ГОСТ 34.10–94. Применяется как для шифрования, так и для создания цифровых подписей.
* ECDSA (Elliptic Curve Digital Signature Algorithm) — алгоритм, основанный на сложности вычисления дискретного логарифма в группе точек эллиптической кривой. Применяется для генерации цифровых подписей, в частности для подтверждения транзакций в криптовалюте Ripple.
  1. Алгоритм RSA
     1. Описание алгоритма

RSA многие годы противостоит интенсивному криптоанализу. Криптостойкость основана на трудоемкости разложения на множители (факторизации) больших чисел. Открытый и секретный ключи являются функциями двух больших (100 ~ 200 двоичных разрядов или даже больше) простых чисел. Предполагается, что задача восстановления открытого текста по шифротексту и открытому ключу эквивалентна задаче факторизации.

Опишем процесс шифрования. Исходный текст должен быть переведен в числовую форму, этот метод считается известным. В результате этого текст представляется в виде одного большого числа. Затем полученное число разбивается на части (блоки) так, чтобы каждая из них была числом в промежутке (о выборе N — см. ниже). Процесс шифрования одинаков для каждого блока. Поэтому мы можем считать, что блок исходного текста представлен числом , .

Каждый абонент вырабатывает свою пару ключей. Для этого он генерирует два больших простых числа , вычисляет произведение . Затем он вырабатывает случайное число , взаимно простое со значением функции Эйлера от числа , и находит число из условия. Так как , то такое число существует и оно единственно. Пару он объявляет открытым ключом и помещает в открытый доступ. Пара является секретным ключом.

Для расшифровывания достаточно знать секретный ключ. Числа в дальнейшем не нужны, поэтому их можно уничтожить.

Пользователь , отправляющий сообщение абоненту , выбирает из открытого каталога пару абонента и вычисляет шифрованное сообщение . Чтобы получить исходный текст, абонент вычисляет . Так как , т. е. , где – целое, то применяя теорему Эйлера, получим: следующее соотношение ≡.

Как шифрование, так и расшифровывание в RSA предполагают использование операции возведения целого числа в целую степень по модулю . Если возведение в степень выполнять непосредственно с целыми числами и только потом проводить сравнение по модулю , то промежуточные значения окажутся огромными. Здесь можно воспользоваться свойствами арифметики в классах вычетов. Таким образом, можно рассмотреть промежуточные результаты по модулю . Это делает вычисления практически выполнимыми.

* + 1. Стандарты RSA

Криптосистема RSA – часть многих стандартов. Стандарт ISO 9796 описывает RSA как совместимый криптографический алгоритм, соответствующий стандарту безопасности ITU-T X.509. Кроме этого криптосистема RSA является частью стандартов SWIFT, ANSI X9.31 rDSA и проекта стандарта X9.44 для американских банков. Австралийский стандарт управления ключами AS2805.6.5.3 также включает систему RSA.

Алгоритм RSA используется в Internet, в частности он входит в такие протоколы как S/MIME, IPSEC (Internet Protocol Security) и TLS (которым предполагается заменить SSL), а также в стандарт PKCS, применяемый в важных приложениях.

Множество других разрабатываемых в настоящее время стандартов включают в себя либо сам алгоритм RSA или его поддержку либо рекомендуют криптосистему RSA для обеспечения секретности и/или установления подлинности (аутентификации). Например, включают в себя систему RSA рекомендации IEEE P1363 и WAP WTLS.

* 1. Описание информационных технологий, используемых для решения рассматриваемой задачи
     1. ViaCrypt PGP Business Edition 4.0C

В программе используется шифрование с секретным ключом по алгоритму IDEA, а секретные ключи зашифровываются с помощью системы RSA с открытым ключом. Можно выбрать любую длину ключа в диапазоне от 384 до 2048 бит.

Программа поддерживает ключ аварийного доступа к файлам и заставляет пользователей применять открытые ключи, созданные администратором системы. Эти довольно простые в использовании функции очень украшают программу, но мы были сильно разочарованы, обнаружив, что доступ к аварийным ключам нельзя предоставить одновременно нескольким доверенным лицам.

Можно создавать ключи для шифрования, для цифровых подписей или для того и другого вместе. Программа позволяет легко аннулировать и восстанавливать ключи, а также указывать срок их действия.

* + 1. RSA Secure 1.1.1

Пользователям Windows 3.х пакет RSA Secure обеспечивает простое шифрование файлов.

Программа основана на алгоритме шифрования RC4, разработанном компанией RSA и базирующемся на 128-битовом ключе. Длину ключа изменить нельзя, не поддаются настройке и многие другие параметры программы.

Очевидно, RSA предназначала свою программу для защиты файлов на жестком диске отдельного пользователя и не особо заботилась об организации файлообмена в компании. Бросается в глаза отсутствие поддержки шифрования с открытым ключом, за исключением чрезвычайной ситуации, когда определенное количество доверенных лиц может расшифровать файл.

# Реализация программы «Шифрование данных с помощью асимметричного ключа»

* 1. **Постановка задачи**
     1. Сущность задачи

Программа «Шифрования данных с использованием асимметричного ключа» позволяет шифровать и дешифровывать данные, введённые пользователем. Сами данные заносятся через специальную форму в самом приложении. Данная программа подойдёт IT-работникам для локального шифрования информации.

* + 1. Описание входной информации

Пользователь может зашифровать или дешифровать введённые в специальное окно данные, использую введённый ключ. Предусмотрена функция генерирования, открытого и закрытого ключ (если у него их не было). Так же, есть возможность ввода данных из файла.

При работе с окном генерации ключей, пользователю необходимо ввести два простых числа, произведением меньше 256.

* + 1. Описание выходной информации

Программа выводит расшифрованные или дешифрованные данные.

* + 1. Описание алгоритма решения задачи

При запуске программа предлагает пользователю ввести данные вручную или из файла. Далее пользователю даётся возможность сгенерировать ключи (если их не было раньше), необходимые для шифрования и дешифрования. Затем, в случае выбора генерации ключей, пользователем вводятся два простых числа и происходит их проверка:

* Пользователь ввёл два простых числа, произведение которых меньше 256;
* Пользователь ввёл два равных простых числа;
* Пользователь ввёл не числовые значения;
* Пользователь ввёл слишком большие простые числа;
* Пользователь ввёл непростые числа.

После чего программа выводит данные, из которых формируются ключи. Далее, предоставляется возможность выхода в главное окно, полученные ключи выводятся в два отдельных файла («Открытый ключ» и «Закрытый ключ»)

В случае, когда пользователь выбирает в главном окне операцию над данными («Расшифровать» или «Расшифровать»), происходит проверка:

* Пользователь не выбрал операцию над данными.

Далее вводится ключ и происходит проверка:

* Пользователь не ввёл ключ.

Вывод данных происходит при помощи кнопки «Выполнить». Зашифрованные или расшифрованные данные отображаются в специальном поле, так же выводятся в отдельный файл.

* 1. Информационно-технологическая схема

Информационно-технологическая схема работы с формой MainForm представлена на Рисунке 1. Информационно-технологическая схема работы с формой GenForm представлена на Рисунке 2.

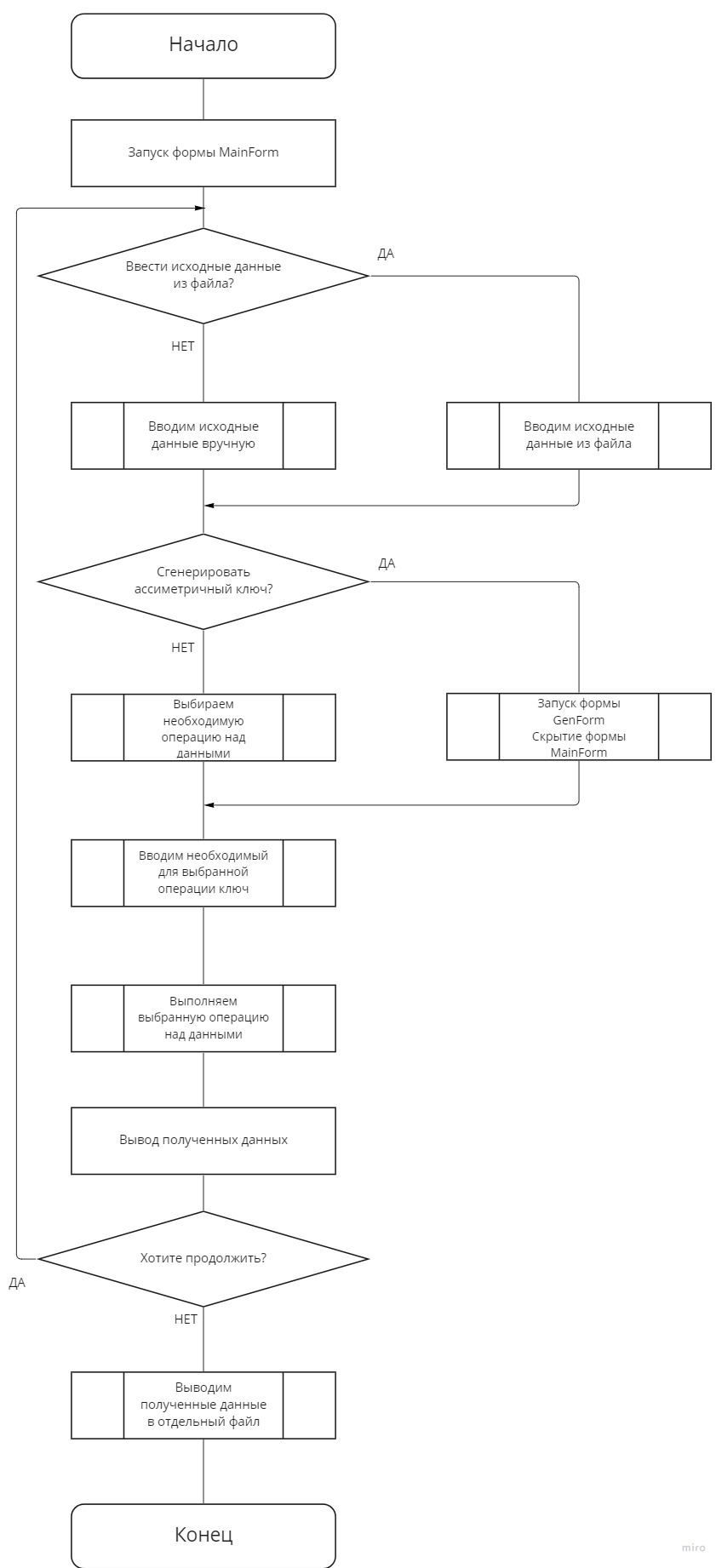


Рисунок 1 - Информационно-технологическая схема работы с формой MainForm



*Рисунок 2 - Информационно-технологическая схема работы с формой GenForm*

* 1. Интерфейс программы

Интерфейс программы состоит из 3 форм. Основная форма MainForm, которая будет появляться при запуске, содержит следующие объекты UserForm:

* Label (4 шт.);
* CommandButton (4 шт.);
* Frame (1 шт.);
* TextBox (3 шт.);
* OptionButton (2 шт.).

Форма GenForm, которая будет появляться при добавлении новой задачи или изменении уже существующей, содержит следующие объекты UserForm:

* Label (7 шт.);
* CommandButton (2 шт.);
* TextBox (6 шт.);
* Frame (1 шт.).

Форма MainForm:

Для ввода данных, которые необходимо шифровать или дешифровать, служит объект TextBox1, а объект Label1 (“Исходные данные”) содержит в себе пояснение для пользователя.

Для ввода ключа, который необходим для шифрования или дешифрования с помощью асимметричного ключа, служит объект TextBox2, а объект Label2 (“Ключ”) содержит в себе пояснение для пользователя.

Для вывода данных, которые пользователь шифровал или дешифровывал, служит объект TextBox3, а объект Label4 (“Полученные данные”) содержит в себе пояснение для пользователя.

Два объекта OptionButton, объединённых одним Frame, дают возможность выбрать один из типов операций над данными

CommandButton2 “Сгенерировать ключ” позволяет сгенерировать асимметричный ключ в отдельной форме GenForm, а объект Label1 (“Нажмите кнопку ниже, если необходимо сгенерировать асимметричный ключ для шифрования и дальнейшей расшифровки”) содержит в себе пояснение для пользователя.

CommandButton1 “Выполнить” позволяет выполнить выбранную операцию над данными, с дальнейшим их выводом в объект TextBox3.

CommandButton3 “Вывести данные в отдельный файл” позволяет вывести полученные данные в отдельный файл.

CommandButton4 “Загрузить данные из файла” позволяет ввести данные из файла в объект TextBox1.

Интерфейс формы представлен на Рисунке 3.

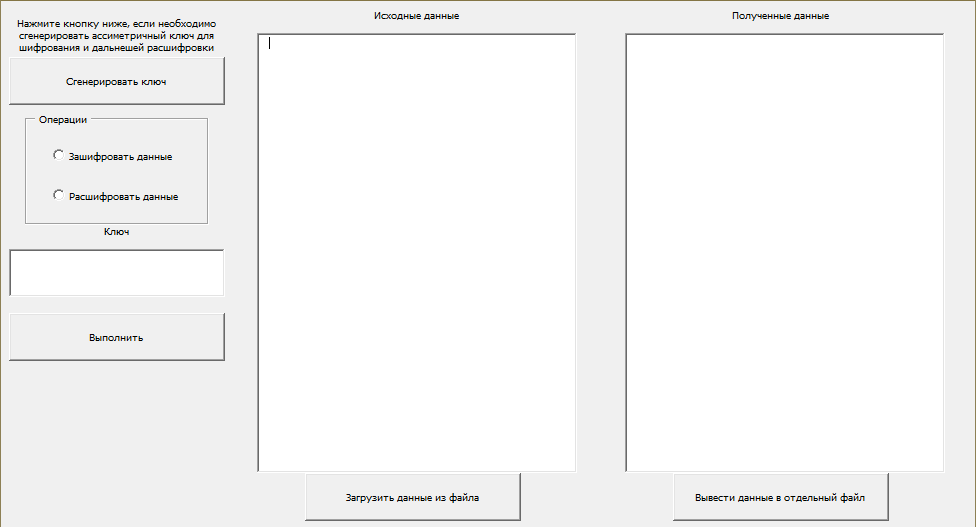


Рисунок 3 - Интерфейс формы MainForm

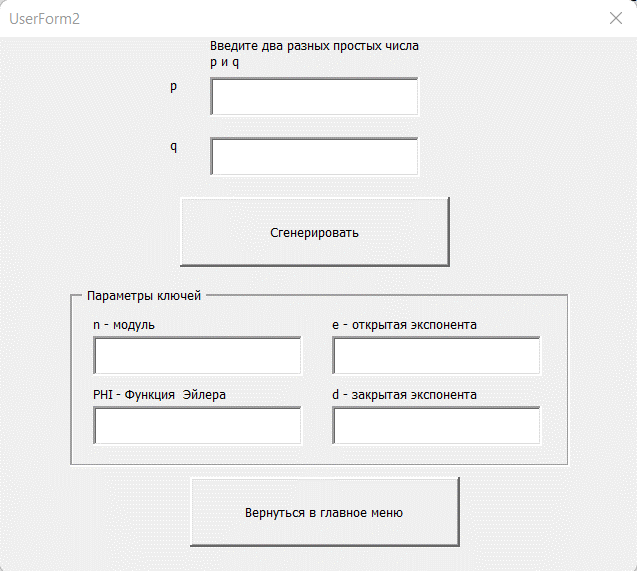
Форма GenForm:

Для ввода двух разных простых чисел, служит объекты TextBox1 и TextBox2, а объекты Label5 (“Введите два разных простых числа p и q”), Label6 (“p”), Label7 (“q”) содержат в себе пояснение для пользователя.

CommandButton1 “Сгенерировать” позволяет выполнить генерацию асимметричного ключа, с дальнейшим выводом параметров ключа в объекты TextBox3, TextBox4, TextBox5, TextBox6, объединённые в один Frame1, а объекты Label1 (“n – модуль”), Label2 (“PHI – Функция Эйлера”), Label3 (“e – открытая экспонента”), Label4 (“d – закрытая экспонента”) содержат в себе пояснение для пользователя.

CommandButton2 “Вернуться в главное меню” позволяет вернуться в форму MainForm, а сгенерированные ключи вывести в отдельные файлы.

Интерфейс формы представлен на Рисунке 4.



*Рисунок 4 - Интерфейс формы GenForm*

* 1. Программный код с комментариями
     1. Код для MainForm

Private Sub CommandButton1\_Click()

'Выполняем выбранную операцию над данными

Dim Str As String

Dim Str1 As String

Dim Str2 As String

Dim myChar As String

Dim m As LongLong

Dim S As LongLong

Dim n As LongLong

Dim c As LongLong

Dim e As LongLong

Dim k As Integer

Dim i As Integer

Dim x As Integer

'Проверка, выбрана какая-либо операция

If Not OptionButton1.Value And Not OptionButton2.Value Then

MsgBox "Выберите операцию над данными"

Exit Sub

End If

Str = TextBox1.Text

Str2 = TextBox2.Text

'Проверка, содержит ли ключ не числовые значения

If Not IsNumeric(TextBox2.Text) Then

MsgBox ("Введите числовое значение ключа")

Exit Sub

End If

'Считываем модуль ключа

n = Right(Str2, Len(Str2) - InStr(1, Str2, " "))

x = 6

Str1 = ""

'Выбрана операция - шифрование данных.

If OptionButton1.Value Then

'Считываем экспоненту

e = Left(Str2, InStr(1, Str2, " "))

Do While (e - 1) Mod x <> 0

x = x - 1

Loop

Do While Len(Str) <> 0

myChar = Left(Str, 1)

Str = Right(Str, Len(Str) - 1)

'Возвращаем код символа для считанного символа из ихсодных данных

m = Asc(myChar)

c = (m ^ x) Mod n

For i = 2 To ((e - 1) / x)

c = (c \* ((m ^ x) Mod n)) Mod n

Next i

'Кодируем и записываем в новую строку, содержащую закодированные данные

c = (c \* m) Mod n

Str1 = Str1 & c & " "

Loop

TextBox3.Text = Str1

Else

'Выбрана операция - дешифрование данных.

'Считываем закрытую экспоненту

d = Left(Str2, InStr(1, Str2, " "))

If InStrRev(" ", Str) = 0 Then

Str = Str & " "

End If

Do While Len(Str) <> 0

k = InStr(1, Str, " ")

If k <> 1 Then

myChar = Left(Str, k - 1)

Str = Right(Str, Len(Str) - k)

c = CInt(myChar)

'Декодируем исходные данные

m = (c ^ d) Mod n

'Записываем данные в новую строку, содержащую декодированные данные

Str1 = Str1 & Chr(CInt(m))

Else

Str = Right(Str, Len(Str) - 1)

End If

Loop

TextBox3.Text = Str1

End If

End Sub

Private Sub CommandButton2\_Click()

'Вызываем форму генерации ассиметричного ключа

Me.Hide

UserForm2.Show

End Sub

'Сохраняем полученные данные в новый файл

Private Sub CommandButton3\_Click()

Dim FSys As New FileSystemObject

Dim Tfile As TextStream

Set Tfile = FSys.CreateTextFile(Application.GetSaveAsFilename("Полученные данные", fileFilter:="Текстовые файлы (\*.txt), \*.txt"))

Tfile.Write TextBox3.Text

Tfile.Close

Set Tfile = Nothing

Set FSys = Nothing

End Sub

'Считываем исходные данные из файла

Private Sub CommandButton4\_Click()

Dim FSys As New FileSystemObject

Dim sFname As Variant

Dim Tfile As TextStream

Dim TextOfFile As String, CryptoText As String

Set FSys = CreateObject("Scripting.FileSystemObject")

sFname = Application.GetOpenFilename("Txt file (\*.txt),\*.txt")

If sFname = False Then

MsgBox "Файл не выбран"

Exit Sub

End If

Set Tfile = FSys.OpenTextFile(sFname, ForReading, False, TristateFalse)

TextOfFile = Tfile.ReadAll

Tfile.Close

TextOfFile = LCase(TextOfFile)

TextBox1.Text = TextOfFile

End Sub

* + 1. Код для GenForm

Option Explicit

Private Sub CommandButton1\_Click()

'Генерируем параметры ассиметричного ключа

'Проверка, является ли p числом

If Not IsNumeric(TextBox1.Text) Then

MsgBox ("введите число p")

Exit Sub

End If

'Проверка, является ли q числом

If Not IsNumeric(TextBox2.Text) Then

MsgBox ("введите число q")

Exit Sub

End If

Dim p As Long, q As Long, n As Long, e As Long, PHI As Long, d As Long

p = TextBox1.Text

q = TextBox2.Text

'Проверка, равны ли p и q между собой

If p = q Then

MsgBox ("Простые числа не должны быть равны между собой")

End If

'Проверка, произведение p и q больше 256 или нет

If p \* q < 256 Then

MsgBox ("Произведение простых чисел p и q должно быть больше 256")

Exit Sub

End If

'Проверка, является ли p простым числом

If (check\_prime(p) = False) Then

MsgBox ("p слишком велико или не является простым,обратитесь к таблице простых чисел")

Else

'Проверка, является ли q простым числом

If (check\_prime(q) = False) Then

MsgBox ("q слишком велико или не является простым,обратитесь к таблице простых чисел")

Else

'Вычисляем модуль, функцию Эйлера, открытую экспоненту, закрытую экспоненту

n = p \* q

TextBox3.Text = n

PHI = (p - 1) \* (q - 1)

e = getE((PHI))

d = getD((e), (PHI))

TextBox4.Text = PHI

TextBox5.Text = d

TextBox6.Text = e

End If

End If

End Sub

Private Function check\_prime(ByVal val As Long) As Boolean

'Проверка, является ли число простым

Dim primes, i As Integer, prime As Boolean

primes = Array(1, 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, \_

59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, \_

139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, \_

229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, \_

317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397)

check\_prime = False

For i = 0 To 78

If (val = primes(i)) Then

prime = True

End If

Next i

check\_prime = prime

End Function

Private Function getD(ByVal e As Long, ByVal PHI As Long) As Long

'Вычисляем закрытую экспоненту

Dim u(3) As Long

Dim v(3) As Long

Dim q As Long

Dim temp1 As Long

Dim temp2 As Long

Dim temp3 As Long

u(0) = 1: u(1) = 0: u(2) = PHI: v(0) = 0: v(1) = 1: v(2) = e

While (v(2) <> 0)

q = Int(u(2) / v(2))

temp1 = u(0) - q \* v(0)

temp2 = u(1) - q \* v(1)

temp3 = u(2) - q \* v(2)

u(0) = v(0)

u(1) = v(1)

u(2) = v(2)

v(0) = temp1

v(1) = temp2

v(2) = temp3

Wend

If (u(1) < 0) Then

getD = (u(1) + PHI)

Else

getD = u(1)

End If

End Function

Private Function getE(ByVal PHI As Long) As Long

'Вычисляем открытую экпоненту

Dim great As Long

Dim e As Long

great = 0

e = 2

While (great <> 1)

e = e + 1

great = get\_common\_denom(e, PHI)

Wend

getE = e

End Function

Private Function get\_common\_denom(ByVal e As Long, ByVal PHI As Long) As Long

'Вычисляем НОД

Dim great As Long, temp As Long, a As Long

If (e > PHI) Then

While (e Mod PHI <> 0)

temp = e Mod PHI

e = PHI

PHI = temp

Wend

great = PHI

Else

While (PHI Mod e <> 0)

a = PHI Mod e

PHI = e

e = a

Wend

great = e

End If

get\_common\_denom = great

End Function

Private Sub CommandButton2\_Click()

'Сохраняем параметры ассиметричного ключа в два отдельных файла

Dim FSys\_1 As New FileSystemObject

Dim Tfile\_1 As TextStream

Set Tfile\_1 = FSys\_1.CreateTextFile(Application.GetSaveAsFilename("Открытый ключ", fileFilter:="Текстовые файлы (\*.txt), \*.txt"))

Tfile\_1.Write TextBox5.Text + " " + TextBox3.Text

Tfile\_1.Close

Set Tfile\_1 = Nothing

Set FSys\_1 = Nothing

Dim FSys\_2 As New FileSystemObject

Dim Tfile\_2 As TextStream

Set Tfile\_2 = FSys\_2.CreateTextFile(Application.GetSaveAsFilename("Закрытый ключ", fileFilter:="Текстовые файлы (\*.txt), \*.txt"))

Tfile\_2.Write TextBox6.Text + " " + TextBox3.Text

Tfile\_2.Close

Set Tfile\_2 = Nothing

Set FSys\_2 = Nothing

Unload Me

'при закрытии формы очищаем текстовые поля

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ""

TextBox5.Text = ""

TextBox6.Text = ""

'Открываем главную форму

UserForm2.Hide

UserForm1.Show

End Sub

* + 1. Код для пользовательского открытия формы

Sub Data\_Visualization()

'Сочетание клавиш: Ctrl+й

UserForm1.Show

End Sub

* 1. Представление пользовательского интерфейса

После запуска программы открывается UserForm MainForm и пользователь предлагается программа для шифрования данных с помощью асимметричного ключа.

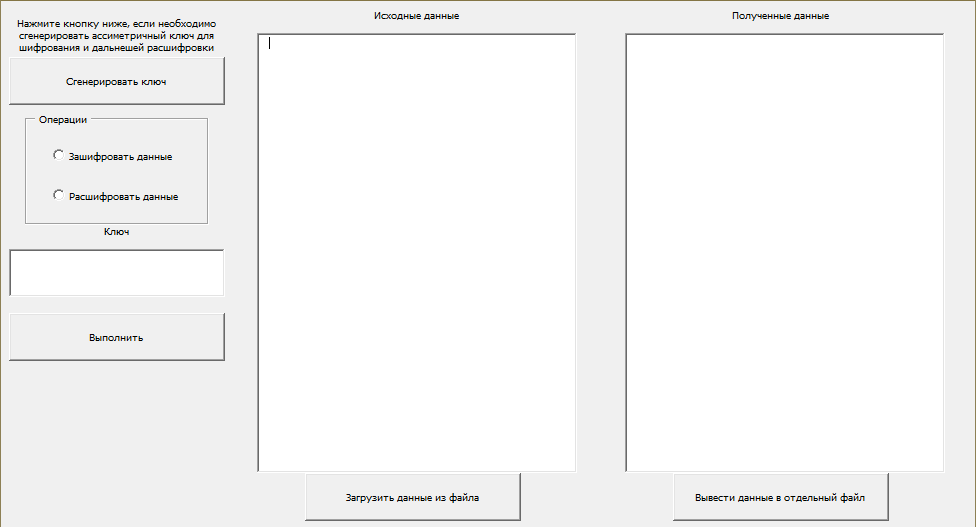


Рисунок 5 - Стартовое окно приложения – MainForm

В начале следует ввести исходные данные для шифрования или дешифрования, это можно сделать через соответствующее поле «Исходные данные» (TextBox1) или ввести данные из файла, нажав на кнопку “Загрузить данные из файла” (CommandButton4). После ввода данных, необходимо сгенерировать асимметричный ключ (если его не было раньше), для этого необходимо нажать на кнопку “Сгенерировать ключ” (CommandButton2). При нажатии соответствующей кнопки открывается новое окно UserForm GenForm, в котором предусмотрена возможность генерирования ассиметричного ключа.

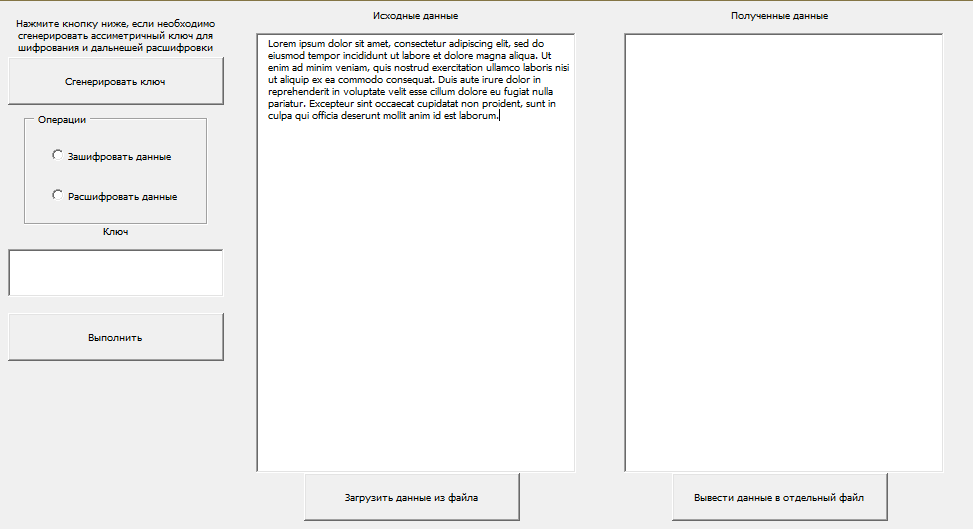


Рисунок 6 - Окно с введёнными исходными данными - MainForm

На данном этапе пользователю предлагается заполнить форму необходимыми данными для генерирования асимметричного ключа. Для этого необходимо указать в обязательные поля требуемую информацию (два разных простых числа) следующую информацию как это показано на Рисунке 7.

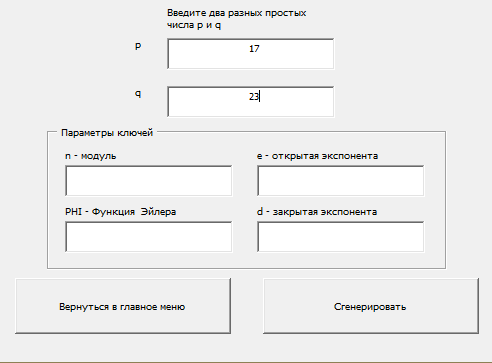


Рисунок 7 - Пример заполнения формы GenForm

После успешного заполнения формы следует нажать кнопку «Сгенерировать» (CommandButton1). В случае некорректного заполнения программа выдаст пользователю подробное описание ошибки и предложит ее исправить, после чего можно сгенерировать ключ еще раз.

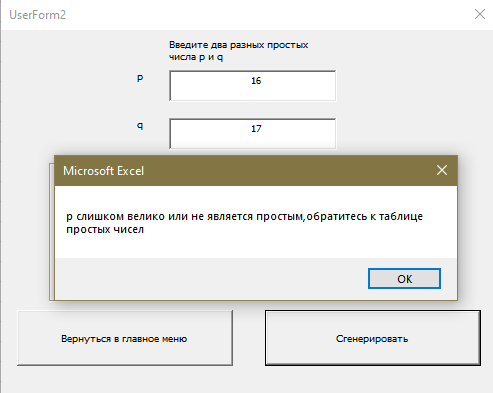


Рисунок 8 - Пример неверно заполненной формы GenForm

Если форма была заполнена правильно, то в соответствующие поля (TextBox3, TextBox4, TextBox5, TextBox6) выведутся параметры асимметричного ключа. Это продемонстрированно на Рисунке 9.

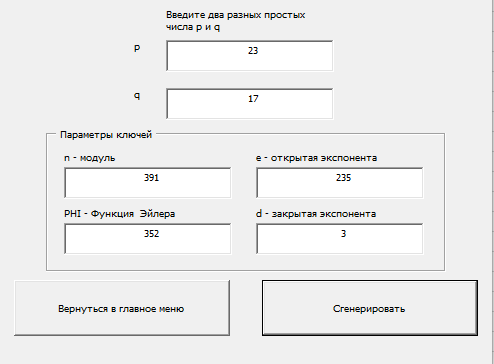


Рисунок 9 - Пример правильно заполненной формы GenForm и вывода сгенерированных параметров ассиметричного ключа в форме GenForm

Если пользователя устраивают сгенерированные ключи, то ему предлагается вернуться в главное меню, с выводом сгенерированных ключей в отдельные файлы (“Открытый ключ”, ”Закрытый ключ” или под другими названиями, если пользователя не устраивают эти), нажав на кнопку “Вернуться в главное меню” (CommandButton2). После нажатия откроется окно UserForm MainForm.

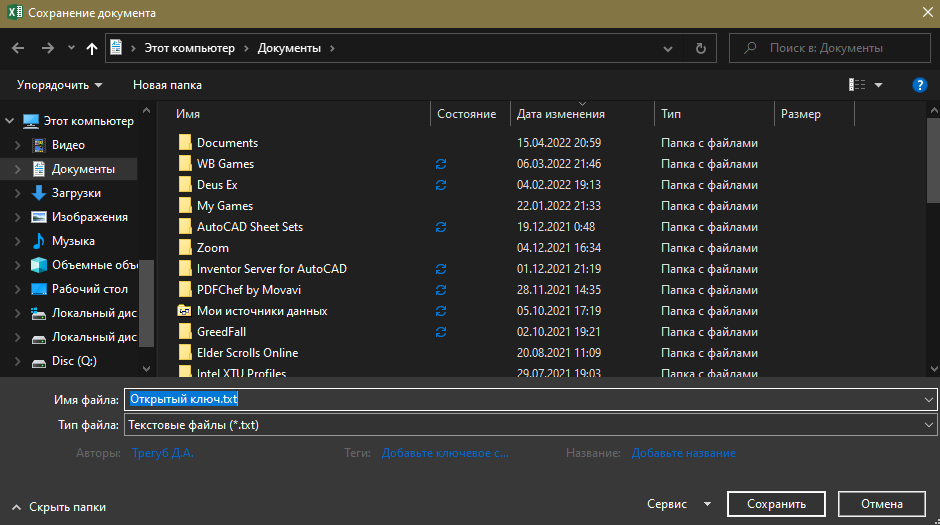


Рисунок 10 - Окно сохранения параметров ассиметричного ключа - GenForm

Теперь, в открытом окне, можно выбрать соответствующую операцию над данными, нажав на соответствующую кнопку “Зашифровать данные” (OptionButton1) или “Расшифровать данные” (OptionButton2). Если пользователь не выберет операцию, то выведется ошибка.

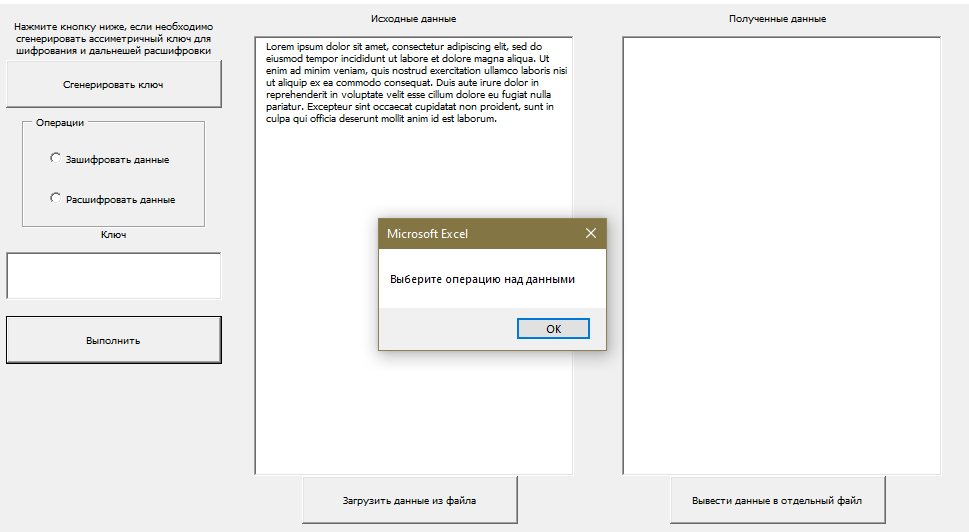


Рисунок 11 - Пример неверного выбора операции над данными в форме MainForm

Затем, следует ввести сгенерированные ключи (с учётом выбранной операции: для шифрования данных – ввести параметры открытого ключа, для расшифровывания данных – ввести параметры закрытого ключа). При вводе не числовых значений выведется ошибка, потребуется ввести ключ заново.

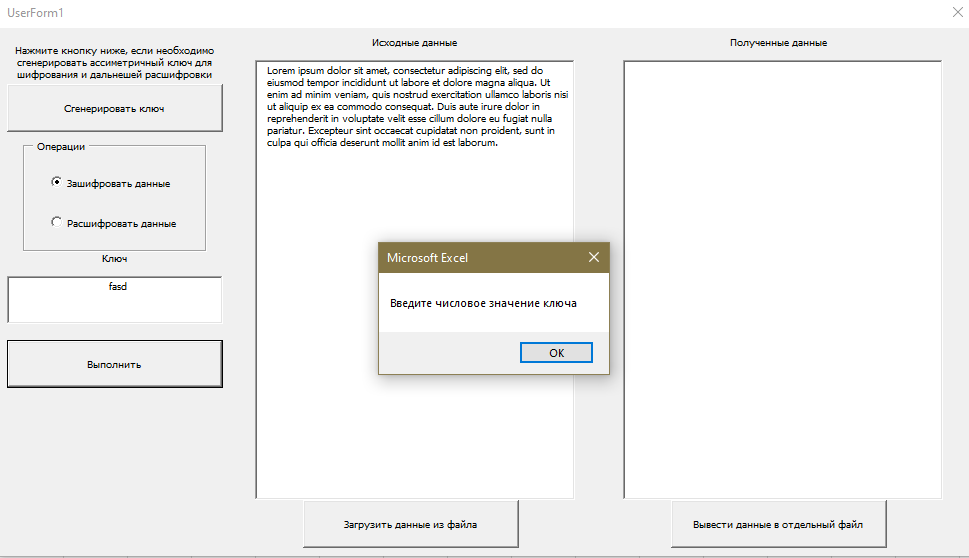


Рисунок 12 - Пример неправильного заполнения поля ассиметричного ключа в форме MainForm

После успешного заполнения формы следует нажать кнопку «Выполнить» (CommandButton1). В случае некорректного заполнения программа выдаст пользователю подробное описание ошибки и предложит ее исправить, после чего можно попробовать внести изменения еще раз.

Если форма была заполнена правильно, то выбранная операция выполнится над данными и полученные значения выведутся в специальное поле (TextBox3). Из данного поля можно вывести данные отдельный файл, нажав на кнопку “Вывести данные в отдельный файл” (CommandButton3). Это продемонстрированно на Рисунке 13 и Рисунке 14.

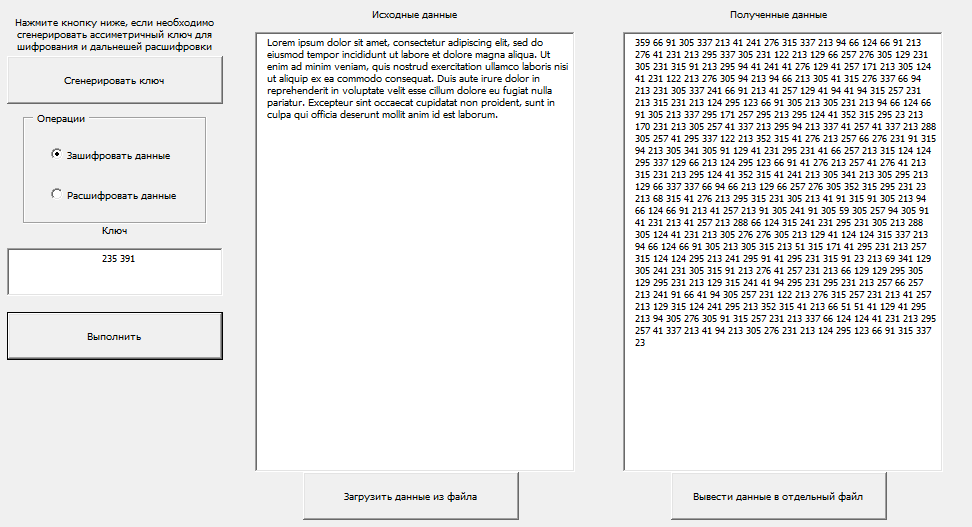


Рисунок 13 - Пример шифрования исходных данных в форме MainForm

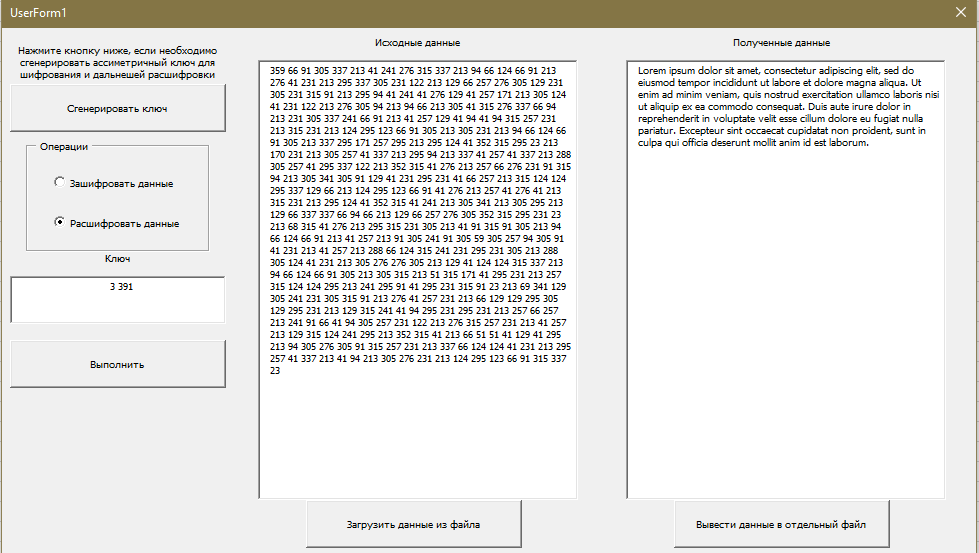


Рисунок 14 - Пример дешифрования исходных данных в форме MainForm

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были выполнены поставленные задачи. Среди них:

* совершенствование навыков программирования на современных структурных языках программирования (Visual Basic for Applications);
* реализация приложения;
* поиск и изучение актуальной информации и литературы, соответствующей выбранной теме;
* тестирование результатов работы программного продукта;
* изучение существующих решений;
* разработка программы по обработке и хранению данных с предусмотренным предоставлением пользователю возможности выбора параметров обработки;
* спроектировать интерфейс программы и порядок ее взаимодействия с пользователем.

Наконец была достигнута цель, заключавшаяся в написании программы менеджера задач на языке программирования VBA и в разработке пользовательского интерфейса.

Программа обладает высокой эффективностью. Были учтены ошибки, которые могли бы сломать работу программы, а также ошибки, которые нарушают логику событий.

Использование «Программы шифрования данных с помощью асимметричного ключа» позволяет предотвратить несанкционированный доступ к конфиденциальным данным.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алгоритм RSA : метод. указания к выполнению лабораторных работ для студентов /О. Н. Жданов, И. А. Лубкин ; Сиб. гос. аэрокосмич. ун-т. – Красноярск, 2007.
2. Ростовцев А.Г. Теоретическая криптография / А.Г. Ростовцев, Е.Б. Маховенко - НПО «Профессионал», Санкт-Петербург, 2004.
3. Фергюсон Н. Практическая криптография / Н. Фергюсон, Б. Шнайер. - М.: Диалектика, 2004.
4. Э. Мэйволд. Безопасность сетей. / 2006.
5. Мельников В.П., Клейменов С.А., Петраков А.М. Информационная безопасность: учебное пособие для студентов сред. проф. образования / М.: Издательский дом «Академия», 2009
6. Сдвижков О.А. Excel-VBA. Словарь-справочник пользователя / О.А. Сдвижков – М.: Эксмо, 2008.
7. Гарнаев А.Ю. MS Excel 2002: Разработка приложений / А.Ю. Гарнаев – СПб. БХВ-Петербург, 2003.
8. «Справочник по VBA для Office. Объект Chart» – URL: <https://docs.microsoft.com/ru-ru/office/vba/api/excel.chart(object)> (дата обращения: 28.04.2022)
9. «Асимметричное шифрование» -– URL: <https://encyclopedia.kaspersky.ru/>(дата обращения: 27.04.2022)
10. «Асимметричное шифрование» -– URL: <https://thecode.media/asymmetric/>(дата обращения: 27.04.2022)