**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

СОГЛАСОВАНО

Профессор департамента

программной инженерии факультета компьютерных наук, к.т.н.

\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Авдошин

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

УТВЕРЖДАЮ

|  |
| --- |
| Академический руководитель образовательной программы «Программная инженерия», к.т.н. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Подп. и дата

**КРИПТОСИСТЕМА РАБИНА  
(RABIN CRYPTOSYSTEM)**

**Программа и методика испытаний**

Инв. № дубл.

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729. 501430-01 51 01-1**

Взам. инв. №

Исполнитель:

студент группы БПИ 165

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. С. Чуев

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г.

Подп. и дата

Инв. № подл

RU.17701729.

501430-01

ТЗ 01-1

**2017**

УТВЕРЖДЕН  
RU.17701729. 501430-01 51 01-1-ЛУ

**КРИПТОСИСТЕМА РАБИНА  
(RABIN CRYPTOSYSTEM)**

**Программа и методика испытаний**

**RU.17701729. 501430-01 51 01-1**

Подп. и дата

Инв. № дубл.

Взам. инв. №

**Листов 42**

Подп. и дата

Инв. № подл

RU.17701729.

501430-01

51 01-1

**2017**

**АННОТАЦИЯ**

Программа и методика испытаний – это документ, в котором содержится информация о программном продукте, а также полное описание приемочных испытаний для данного программного продукта.

Настоящая Программа и методика испытаний для «Криптосистемы Рабина (Rabin Cryptosystem)» содержит следующие разделы: «Объект испытаний», «Цель испытаний», «Требование к программе», «Требования к программным документам», «Средства и порядок испытаний», «Приложения» [7].

В разделе «Объект испытаний» указано наименование и область применения «Криптосистемы Рабина (Rabin Cryptosystem)».

В разделе «Цель испытаний» указана цель проведения испытаний.

Раздел «Требования к программе» содержит требования к программе, которые подлежат проверке во время испытания.

Раздел «Требования к программным документам» содержит состав программной документации, которая представляется на испытания, а также специальные требования к ней.

Раздел «Средства и порядок испытаний» содержит информацию о технических и программных средствах, которые следует использовать во время испытаний, а также порядок этих испытаний.

Раздел «Методы испытаний» содержат информацию об используемых методах испытаний.

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению [7].

Изменения к данному документу оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

СОДЕРЖАНИЕ

[1. ОБЪЕКТ ИСПЫТАНИЙ 5](#_Toc483415050)

[1.1. Наименование 5](#_Toc483415051)

[1.2. Область применения 5](#_Toc483415052)

[1.3. Обозначение испытуемой программы 5](#_Toc483415053)

[2. ЦЕЛЬ ИСПЫТАНИЙ 6](#_Toc483415054)

[3. ТРЕБОВАНИЕ К ПРОГРАММЕ 7](#_Toc483415055)

[3.1. Требование к функциональным характеристикам 7](#_Toc483415056)

[3.1.1. Требование к составу выполняемых функций 7](#_Toc483415057)

[3.1.2. Требование к организации входных данных 8](#_Toc483415058)

[3.1.3. Требование к организации выходных данных 8](#_Toc483415059)

[3.2. Требования к надежности 8](#_Toc483415060)

[3.2.1. Отказы из-за некорректных действий оператора 8](#_Toc483415061)

[3.2.2. Время восстановления после отказа 8](#_Toc483415062)

[3.3. Требования к интерфейсу 9](#_Toc483415063)

[3.3.1. Требования к GUI интерфейсу 9](#_Toc483415064)

[3.3.2. Требования к API интерфейсу 9](#_Toc483415065)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 10](#_Toc483415066)

[4.1. Состав программной документации 10](#_Toc483415067)

[4.2. Специальные требования к программной документации 10](#_Toc483415068)

[5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ 11](#_Toc483415069)

[5.1. Технические средства, используемые во время испытаний 11](#_Toc483415070)

[5.2. Программные средства, используемые во время испытаний 11](#_Toc483415071)

[5.3. Порядок проведения испытаний 11](#_Toc483415072)

[5.4. Условия проведения испытаний 12](#_Toc483415073)

[5.4.1. Климатические условия 12](#_Toc483415074)

[5.4.2. Требования к численности и квалификации персонала 12](#_Toc483415075)

[6. МЕТОДЫ ИСПЫТАНИЙ 13](#_Toc483415076)

[6.1. Испытание выполнения требований к программной документации 16](#_Toc483415077)

[6.2. Испытание выполнения требований к интерфейсу 17](#_Toc483415078)

[6.3 Испытания выполнения требований к функциональным характеристикам 17](#_Toc483415079)

[6.3.1. Испытание выполнения требований к функциональным характеристикам для получения ключа 17](#_Toc483415080)

[6.3.1.1. Генерация открытого ключа и вычисление закрытого ключа 17](#_Toc483415081)

[6.3.1.2. Удаление ключей из программы 20](#_Toc483415082)

[6.3.1.3. Добавление в программу ключей 22](#_Toc483415083)

[6.3.1.4. Выбор ключа для шифрования или дешифрования 24](#_Toc483415084)

[6.3.2. Испытание выполнения требований к функциональным характеристикам для работы с файлом. 26](#_Toc483415085)

[6.3.2.1. Добавление файла в программу 26](#_Toc483415086)

[6.3.2.2. Открытие файла из самой программы 28](#_Toc483415087)

[6.3.2.3. Удаление файла 28](#_Toc483415088)

[6.3.3. Испытание выполнения требований к функциональным характеристикам для шифрования файла 29](#_Toc483415089)

[6.3.4. Испытание выполнения требований к функциональным характеристикам для дешифрования файла 31](#_Toc483415090)

[6.4. Испытание выполнения требований к надежности 34](#_Toc483415091)

[6.4.1. Защита от добавления двух одинаковых файлов 34](#_Toc483415092)

[6.4.2. Защита от удаления файлов 34](#_Toc483415093)

[6.4.3. Защита от кодирования/декодирования без ключа 34](#_Toc483415094)

[6.4.4. Защита от выбора не соответствующего ключа 35](#_Toc483415095)

[6.4.5. Зашита от кодирования/декодирования одного файла дважды одновременно 35](#_Toc483415096)

[6.4.6. Защита при кодировании/декодировании 36](#_Toc483415097)

[6.4.7. Зашита при корректной записи числа, при генерации ключей 36](#_Toc483415098)

[7. ЛИТЕРАТУРА 38](#_Toc483415099)

[ПРИЛОЖЕНИЕ 1 40](#_Toc483415100)

[СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ 40](#_Toc483415101)

[ПРИЛОЖЕНИЕ 2 41](#_Toc483415102)

[ОПИСАНИЕ ФОРМАТА .publicKey 41](#_Toc483415103)

[ОПИСАНИЕ ФОРМАТА .privateKey 41](#_Toc483415104)

[ОПИСАНИЕ ФОРМАТА .encrypted 41](#_Toc483415105)

# **1. ОБЪЕКТ ИСПЫТАНИЙ**

## **1.1. Наименование**

Наименование программы – «Криптосистемы Рабина (Rabin Cryptosystem)».

## **1.2. Область применения**

Программа предназначена для безопасной передачи или получения секретной информации, представленной в цифровом формате (в данном случае имеется ввиду файл, содержащийся на жестком диске компьютера). Данный программный продукт будет полезен в следующих сферах деятельности:

* В банковской сфере
* На закрытых предприятиях
* На военных действиях
* Любым человеком, заботящимся о безопасной передаче информации

## **1.3. Обозначение испытуемой программы**

Наименование темы разработки – «Криптосистемы Рабина (Rabin Cryptosystem)».

**2. ЦЕЛЬ ИСПЫТАНИЙ**

Цель проведения испытаний - проверка соответствия характеристик разработанной программы функциональным требованиям и отдельным требованиям к надежности, изложенных в документе «Техническое задание» к данной программе.

# **3. ТРЕБОВАНИЕ К ПРОГРАММЕ**

## **3.1. Требование к функциональным характеристикам**

### **3.1.1. Требование к составу выполняемых функций**

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

* Получение ключа
* Генерация закрытого ключа [Приложение 1] по указанной пользователем бинарной длине (пара простых чисел [18], дающих остаток 3 по модулю 4);
* Вычисление открытого ключа [Приложение 1] (произведение двух простых чисел [18], дающих остаток 3 по модулю 4);
* Запись открытого ключа в файл с расширением .publicKey [Приложение 2] и закрытого ключа в файл с расширением .privateKey [Приложение 2] на жесткий диск компьютера;
* Добавление в программу ключей, по указанному пользователем пути;
* Удаление ключа из списка ключей в программе.
* Кодирование файла
* Выбор открытого ключа для шифрования;
* Кодирование файла с использованием открытого ключа [17];
* Кодирование группы файла с использованием открытого ключа;
* Запись закодированного файла с расширением .encrypted [Приложение 2] на жесткий диск компьютера.
* Декодирование файла
* Выбор закрытого ключа для дешифрования;
* Используя закрытый ключ, декодирование [17] зашифрованного файла с расширением .encrypted [Приложение 2];
* Используя закрытый ключ, декодирование группы зашифрованных файлов с расширением .encrypted [Приложение 2];
* Запись раскодированного файла на жесткий диск компьютера.
* Работа с файлом
* Добавление файла в программу, по указанному пользователем пути;
* Открытие файла по умолчанию из самой программы;
* Удаление файла из списка файлов в программе.

### **3.1.2. Требование к организации входных данных**

Программа должна иметь возможность получать на вход файлы следующего формата:

* Файлы с любым расширением (шифруемый файл)
* Файлы с расширением .publicKey [Приложение 2] и .privateKey [Приложение 2] (открытый и закрытый ключи соответственно)
* Файл с расширением .encrypted [Приложение 2] (зашифрованный файл)

### **3.1.3. Требование к организации выходных данных**

Программа должна иметь возможность выдавать на выходе файлы следующего формата:

* Файлы с любым расширением (расшифрованный файл)
* Файлы с расширением .publicKey [Приложение 2] и .privateKey [Приложение 2] (открытый и закрытый ключи соответственно)

Файл с расширением .encrypted [Приложение 2] (зашифрованный файл)

**3.2. Требования к надежности**

### **3.2.1. Отказы из-за некорректных действий оператора**

Программа не должна завершаться аварийно, при некорректных действиях пользователя, программа должна выводить сообщение с соответствующей ошибкой и дальнейшими указаниями.

### **3.2.2. Время восстановления после отказа**

В случае возникновения сбоя, вызванного внешними факторами (непредвиденное выключение источника питания) время восстановления программы не должно превышать суммарно затраченного времени на решение проблем с техническими средствами (перезагрузка операционной системы компьютера).

**3.3. Требования к интерфейсу**

### **3.3.1. Требования к GUI интерфейсу**

Данный программный продукт должен иметь Windows-интерфейс. На интерфейсе должны располагаться компоненты, позволяющие пользователю выполнять все операции, описанные в пункте 3.1.1.

### **3.3.2. Требования к API интерфейсу**

Программный интерфейс приложения должен обеспечивать генерацию открытого и закрытого ключей при полученной бинарной длине открытого ключа.

Должен обеспечивать свободное кодирование и декодирование информации представленной в формате String [Приложение 1], BigInteger [Приложение 1] и любого наследника класса Stream [Приложение 1].

# **4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

## **4.1. Состав программной документации**

На испытания должна быть представлена документация к программе в следующем составе:

* «Криптосистема Рабина (Rabin Cryptosystem)» Техническое задание [11].
* «Криптосистема Рабина (Rabin Cryptosystem)». Программа и методика испытаний [7].
* «Криптосистема Рабина (Rabin Cryptosystem)». Текст программы [12].
* ««Криптосистема Рабина (Rabin Cryptosystem)». Пояснительная записка [13].
* «Криптосистема Рабина (Rabin Cryptosystem)». Руководство оператора [14].
* «Криптосистема Рабина (Rabin Cryptosystem)». Руководство программиста [15]

## **4.2. Специальные требования к программной документации**

* Все документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 (Требования к программным документам, выполненным печатным способом) [6] и ГОСТ к этому виду документа (см. п. 4.1.).
* Вся документация сдается в печатном виде, при этом она должна быть обязательно подписана руководителем организации, утвердившей документ на разработку, руководителем разработки и исполнителем перед сдачей курсовой работы в информационно-образовательную среду НИУ ВШЭ LMS (Learning management system) и ее защитой.
* Вся документация также сдается в электронном виде в формате .pdf или .docx. в архиве формата .rar. Все документы перед защитой курсовой работы должны быть загружены в информационно-образовательную среду НИУ ВШЭ LMS (Learning management system) в личном кабинете во вкладке «Проекты» - «Курсовая работа».

**5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ**

**5.1. Технические средства, используемые во время испытаний**

Необходимые требования и минимальные характеристики для надежной и бесперебойной работы программы требуется следующий состав технических средств:

* 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 ГГц [Приложение 1] или выше.
* 1 ГБ [Приложение 1] (для 32-разрядного процессора) или 2 ГБ (для 64-разрядного процессора) ОЗУ [Приложение 1].
* 16 ГБ (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) свободного места на жестком диске.
* Графическое устройство DirectX 9 с драйвером WDDM [Приложение 1] 1.0 или более поздней версии.
* Мышь или совместимое указывающее устройство.
* Клавиатура.
* CD-ROM Drive
* USB-порт

**5.2. Программные средства, используемые во время испытаний**

Для работы программы необходим следующий состав программных средств:

* Операционная система Microsoft Windows 7 или более поздняя версия.
* Microsoft .NET Framework 4.5.2.
* Диск в состав которого входит программная документация, приложение (исполняемые файлы, два примера задачи и прочие необходимые для работы программы файлы) и презентация проекта.

**5.3. Порядок проведения испытаний**

Испытания должны проводиться в следующем порядке:

* Проверка требований к программной документации;
* Проверка требований к интерфейсу;
* Проверка требований к надежности;
* Проверка требований к функциональным характеристикам.

**5.4. Условия проведения испытаний**

### **5.4.1. Климатические условия**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к персональным компьютерам и компакт-дискам (CD) в части условий их эксплуатации.

Персональный компьютер предназначен для работы в закрытом отапливаемом помещении со стабильными климатическими условиями категории 4.1 согласно [10].

Для предотвращения повреждений компакт-диска и сохранения на нем записанной информации необходимо поддерживать следующие климатические условия:

* влажность от 20% до 70%;
* температура от 5°C до 30°C;
* атмосферное давление — от 84 до 106,7 кПа (от 630 до 800 мм рт. ст).

### **5.4.2. Требования к численности и квалификации персонала**

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 1 штатной единицы:

конечный пользователь – оператор ЭВМ.

Оператор ЭВМ должен:

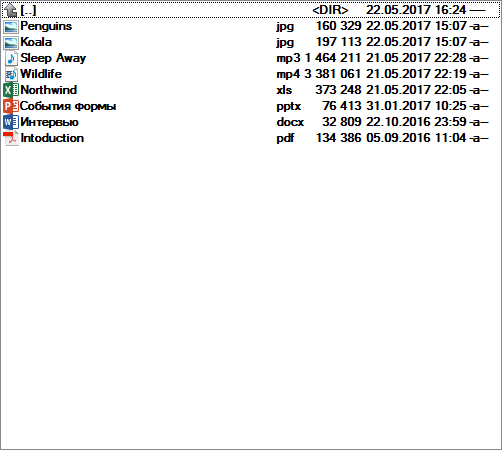
* иметь образование не ниже среднего (полного) общего;
* обладать практическими навыками работы с пользовательским интерфейсом операционной системы.
* иметь поверхностные знания о криптосистемах с открытым ключом.

# **6. МЕТОДЫ ИСПЫТАНИЙ**

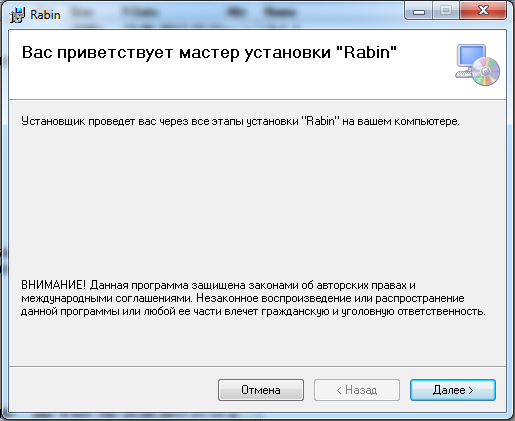
Испытания представляют собой процесс установления соответствия программы и программной документации заданным требованиям.

Перед проведением испытаний необходимо установить программу с компакт-диска.

Программное обеспечение «Rabin» поставляется на CD-диске. В комплект поставки программы входит CD-диск, который содержит файлы примеров к программе (см. рис. ниже), техническую документацию и презентацию проекта.

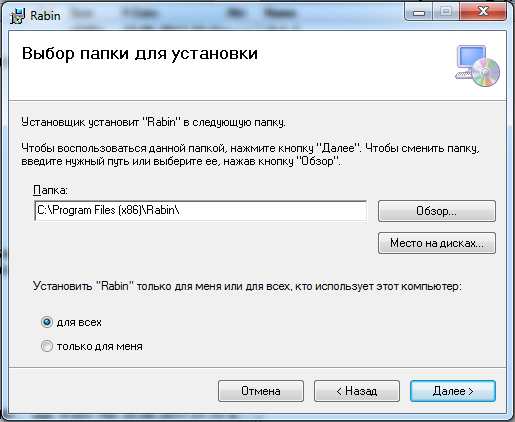


При запуске диска необходимо открыть папку «Установка» и запустить пакет установщика Windows Rabin\_Setup.msi. После запуска появится следующее диалоговое окно (рис. 1):



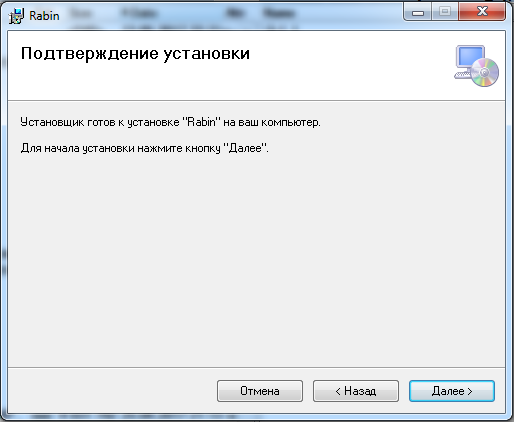
*Рисунок 1*

В этом окне (рис. 1) необходимо после прочтения информации нажать кнопку «Далее». После этого окна последует окно (рис. 2):



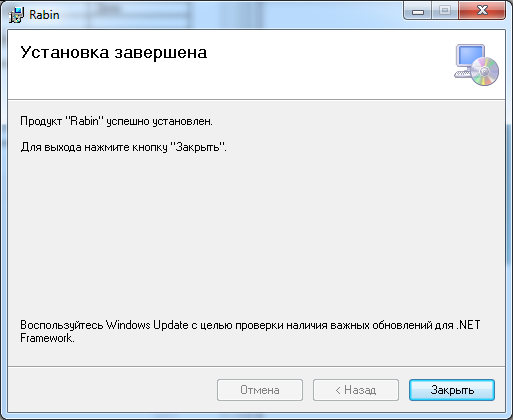
*Рисунок 2*

В поле «Папка» необходимо указать папку, в которую будет установлена программа. Также необходимо выбрать какие пользователи смогут пользоваться этой программой. Нажать «Далее». Появится окно (рис. 3):



*Рисунок 3*

Снова нажать «Далее», появится окно установки, а потом последует окно, сообщающее о завершении установки программы (рис. 4):



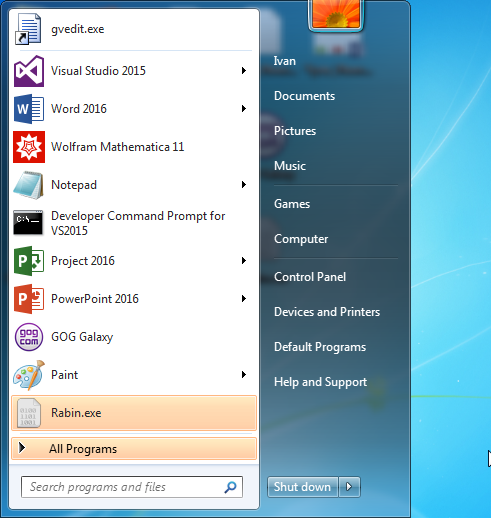
*Рисунок 4*

Нажать кнопку «Закрыть». После успешной установки программы, на рабочем столе компьютера появился ярлык для запуска программы «Rabin» (рис. 5):

C:\Users\Ivan\Documents\ShareX\Screenshots\2017-05\explorer_2017-05-13_22-58-26.png

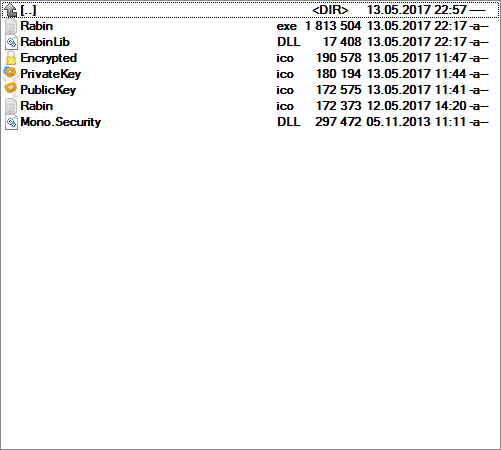
*Рисунок 5*

Также ярлык для программы «Rabin» можно найти в меню «Пуск» (рис. 6):



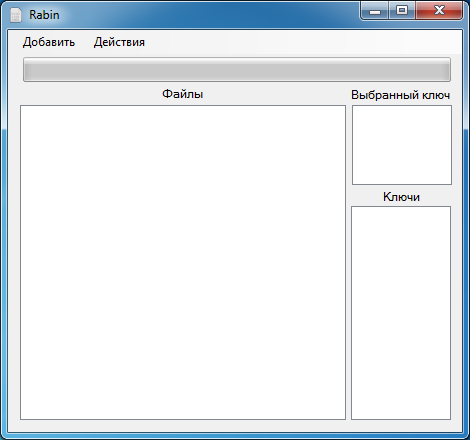
*Рисунок 6*

В папке, куда была установлена программа, должен появиться следующий состав файлов (рис. 7):



*Рисунок 7*

После установки программы выполняется ее открытие, нужно два раза кликнуть на иконку, которая появилась на рабочем столе, или на иконку в меню «Пуск» (Программу можно открыть также и из папки, в которую была выполнена установка. Необходимо два раза кликнуть по приложению Rabin). В случае успешного запуска появится следующее окно (рис. 8):



*Рисунок 8*

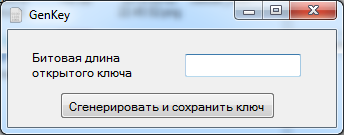
**6.1. Испытание выполнения требований к программной документации**

Состав программной документации проверяется визуально, проверяется наличие всех подписей и наличие программной документации в системе LMS. Также визуально проверяется соответствие документации требованиям ГОСТ. Все документы удовлетворяют представленным требованиям.

**6.2. Испытание выполнения требований к интерфейсу**

Интерфейс программы состоит из двух окон:

1. Основное окно (рис. 8)
2. Окно для генерации ключей (рис. 9)



*Рисунок 9*

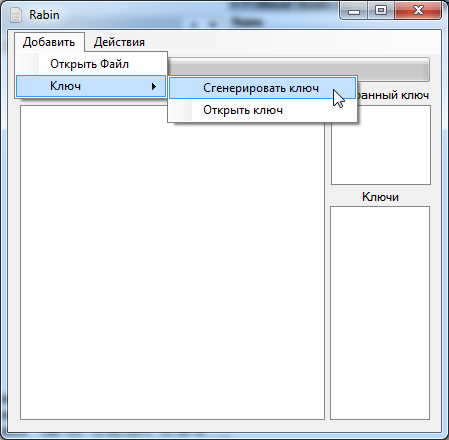
Данное программное обеспечение имеет Windows интерфейс. На главном окне программы управление реализовано с помощью главного и контекстного меню, выделены поля для файлов, ключей и выбранного ключа. Также вверху располагается ProgressBar, который описывает прогресс. Управление окном генерации ключей осуществляется с помощью системы кнопочного типа. Также выделено окно для записи бинарной длины открытого ключа.

## **6.3 Испытания выполнения требований к функциональным характеристикам**

### **6.3.1. Испытание выполнения требований к функциональным характеристикам для получения ключа**

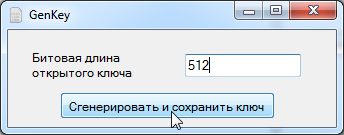
**6.3.1.1. Генерация открытого ключа и вычисление закрытого ключа**

В главном окне программы пройдем по следующему пути «Добавить\Ключ\Сгенерировать ключ» (рис. 10)



*Рисунок 10*

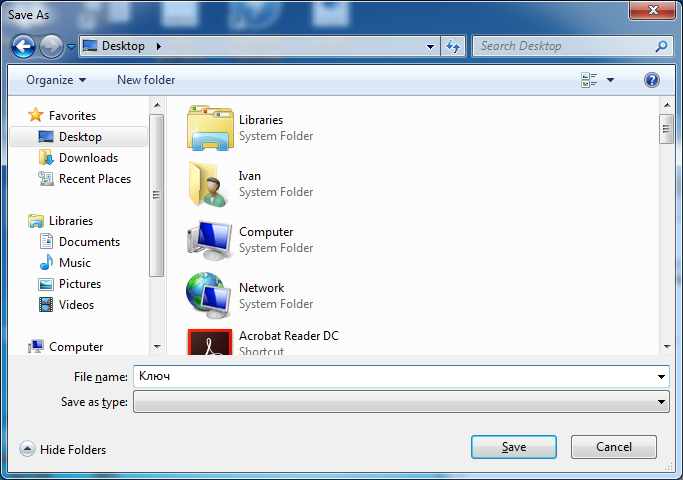
После данных действий откроется окно генерации ключа (рис. 9). В поле для введения введем бинарную длину открытого ключа, пусть это будет число 512, и нажмем кнопку «Сгенерировать и сохранить ключ». (рис. 11)



*Рисунок 11*

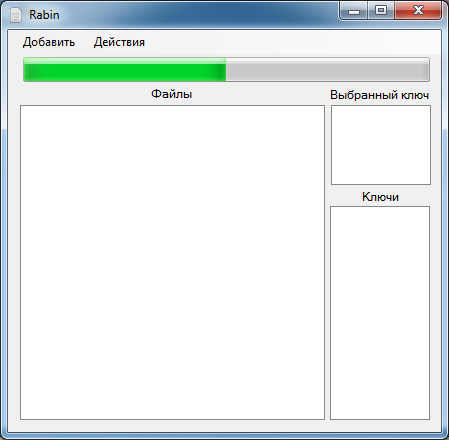
**6.3.1.2. Запись открытого и закрытого ключей в файлы с расширением .publicKey и .privateKey соответственно**

После действий, выполненных в п. 6.3.1.1., откроется диалог для выбора названия места сохранения ключей. Выберем место сохранения: «Desktop» и название для ключей: «Ключ» (Рис 12)



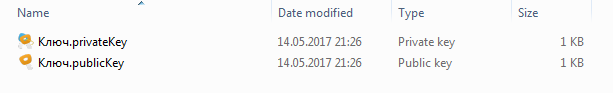
*Рисунок 12*

После этого запустится генерация ключей (рис. 13)

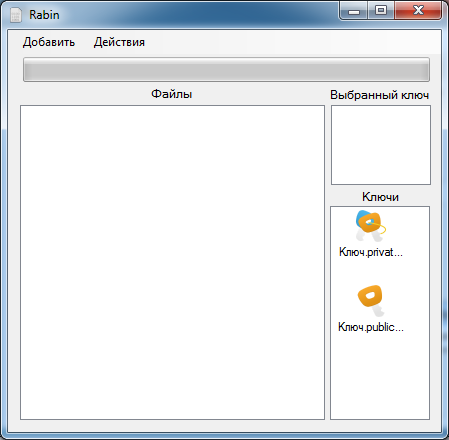


*Рисунок 13*

После того, как открытый и закрытый ключи успешно сгенерируются, они будут сохранены в указанном месте в одной папке. (рис. 14) И также будут добавлены в коллекцию ключей в самой программе. (рис. 15)



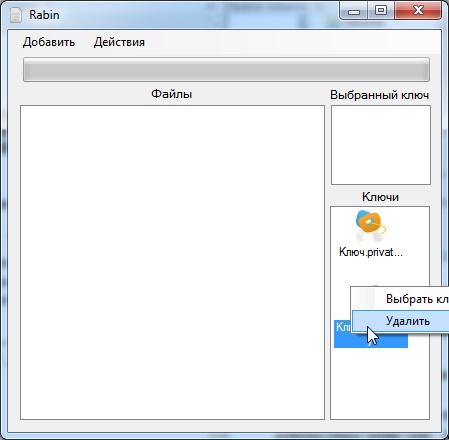
*Рисунок 14*



*Рисунок 15*

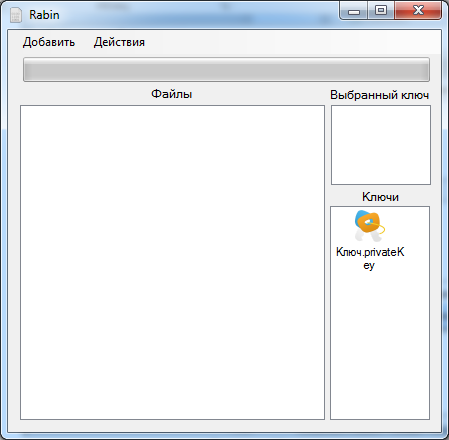
#### **6.3.1.2. Удаление ключей из программы**

Удалим ключи из программы. Для этого выделим удаляемые ключи и кликнем на них правой кнопкой мыши. Появится контекстное меню, в котором нажмем кнопку «Удалить». (Рис 16)



*Рисунок 16*

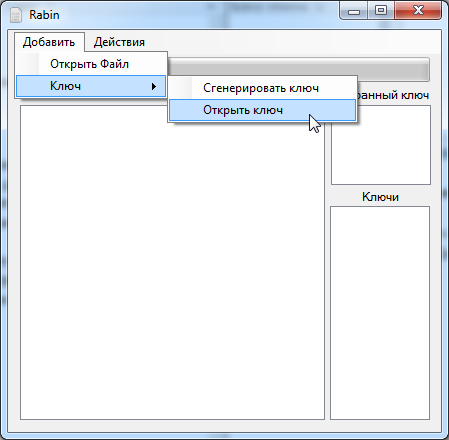
После выполнения данных действий ключ успешно удалится из программы. (Рис 17)



*Рисунок 17*

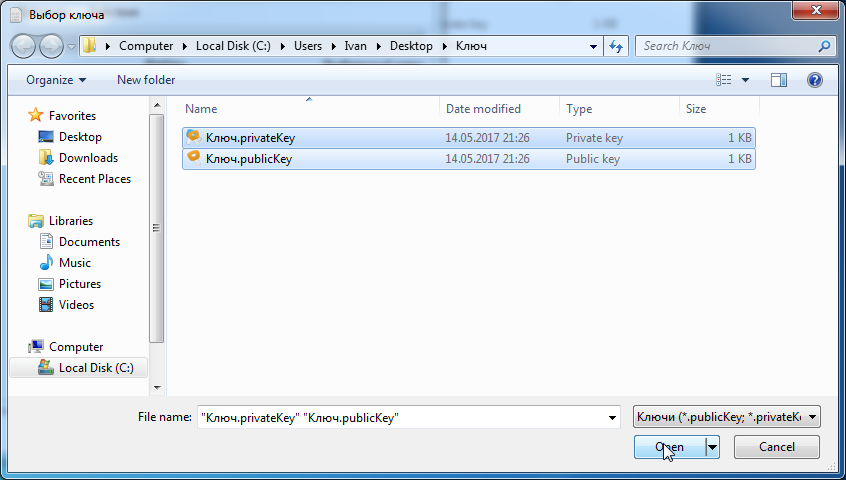
**6.3.1.3. Добавление в программу ключей**

Добавим существующие ключи в программу. Для этого пройдем по пути: «Добавить\Ключ\Открыть ключ» (рис. 18)



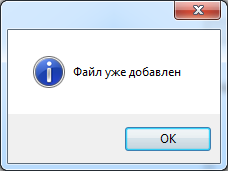
*Рисунок 18*

После этого откроется меню для выбора ключа. Пройдем в папку, где сохранены наши сгенерированные ключи, выберем и нажмем кнопку «Open». (рис. 19)



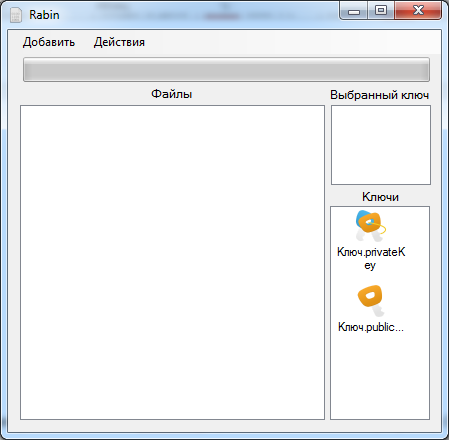
*Рисунок 19*

Но поскольку закрытый ключ «Ключ.privateKey» был уже добавлен в программу, то появится информирующее сообщение (Рис 20)



*Рисунок 20*

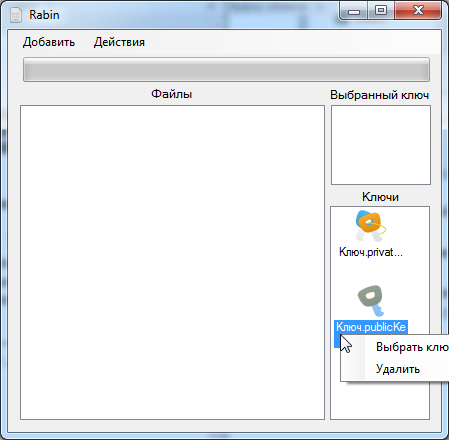
А второй ключ «Ключ.publicKey» успешно добавится в программу (Рис 21)



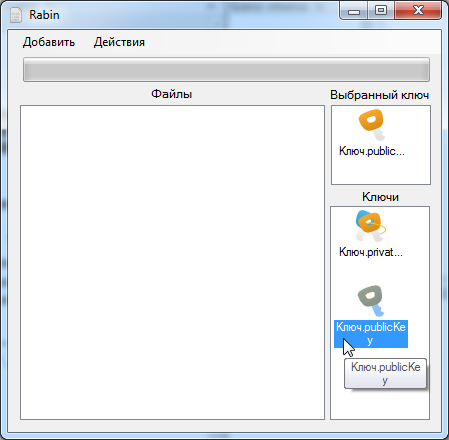
*Рисунок 21*

#### **6.3.1.4. Выбор ключа для шифрования или дешифрования**

Выберем ключ для шифрования, для этого дважды кликнем на ключ, или воспользовавшись контекстным меню (рис. 22), нажмем кнопку «Выбрать ключ». После этого произойдет выбор ключа. (рис. 23).



*Рисунок 22*

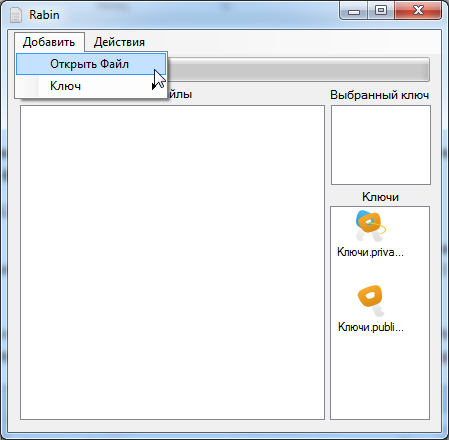


*Рисунок 23*

### **6.3.2. Испытание выполнения требований к функциональным характеристикам для работы с файлом.**

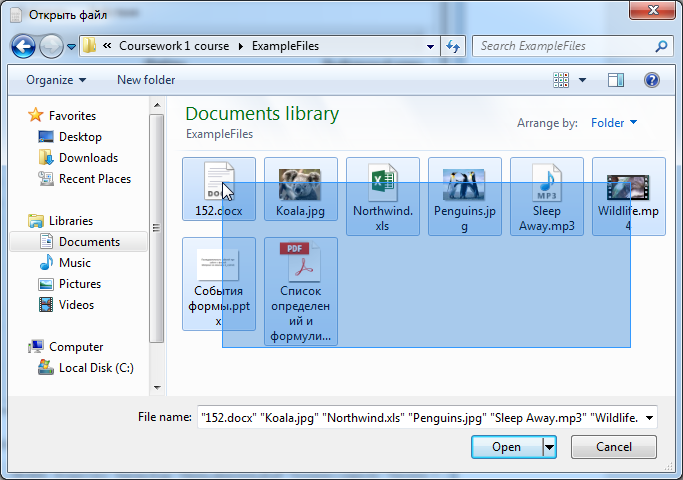
#### **6.3.2.1. Добавление файла в программу**

Добавим файлы в программу, для этого воспользуемся главным меню и пройдем по следующему пути: «Добавить\Открыть Файл». (рис. 25)



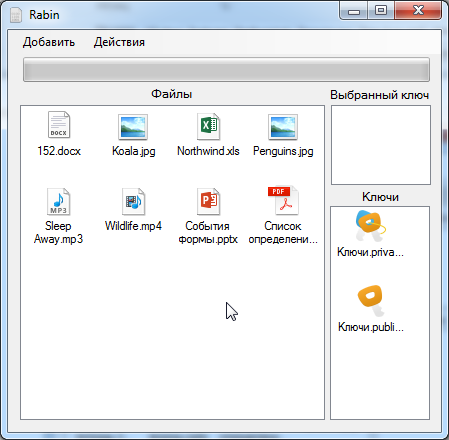
*Рисунок 25*

Далее откроется меню выбора файлов, в котором выберем нужные файлы примеров, предоставленных на диске, расположенных в директории «Coursework 1 course\ExampleFiles». (рис. 22)



*Рисунок 22*

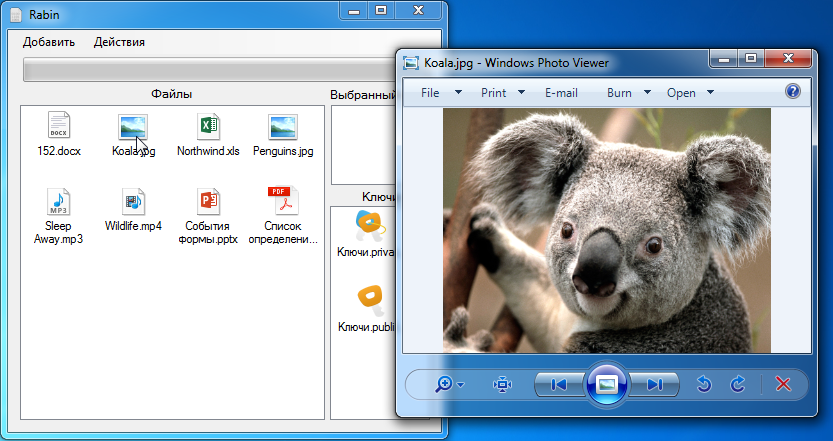
После нажатия кнопки «Open» файлы появятся в приложении (рис. 23)



*Рисунок 23*

#### **6.3.2.2. Открытие файла из самой программы**

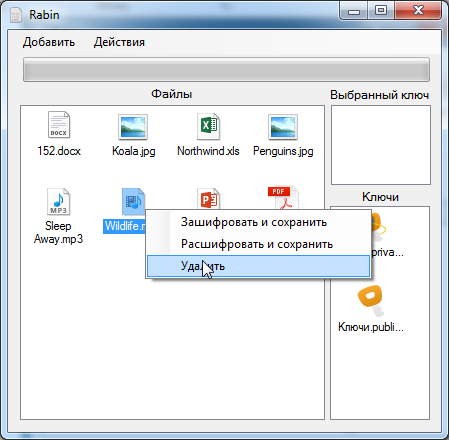
Теперь, чтобы проверить следующее соответствие функциональным требованиям, откроем файл программой по умолчанию из самой программы Rabin. Для этого, нажмем двойным кликом на его иконку «Koalla.jpg». После этого произойдет открытия файла. (рис. 24)



*Рисунок 24*

#### **6.3.2.3. Удаление файла**

Теперь удалим файл «Wildlife.mp4» из программы. Для этого нажмем на него правой кнопкой мыши и во всплывающем контекстном меню нажмем кнопку «Удалить». (рис. 25)



*Рисунок 25*

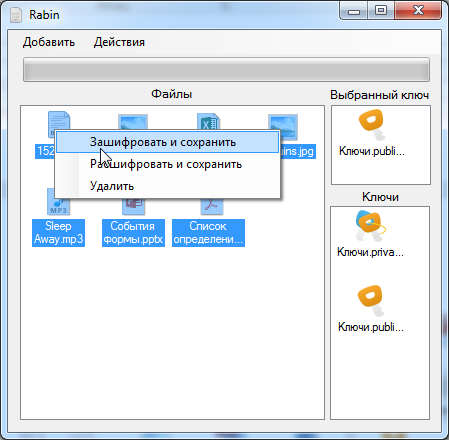
### **6.3.3. Испытание выполнения требований к функциональным характеристикам для шифрования файла**

Для того, чтобы начать шифрование, следует выбрать открытый ключ (рис. 26). 

*Рисунок 26*

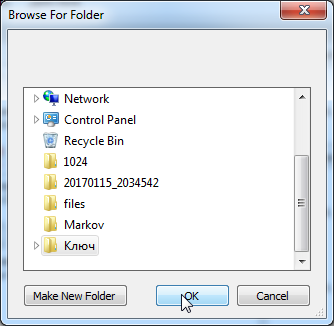
Сделаем это аналогично действиям в п. 6.3.1.4.

После того, как выбран открытый ключ, можно переходить к шифрованию. Теперь зашифруем все добавленные файлы, для этого выделм нужные файлы и кликнем на один из правой кнопкой мыши, далее воспользовавшись всплывающим контекстным меню, нажмем кнопку «Зашифровать и сохранить». (рис. 27)



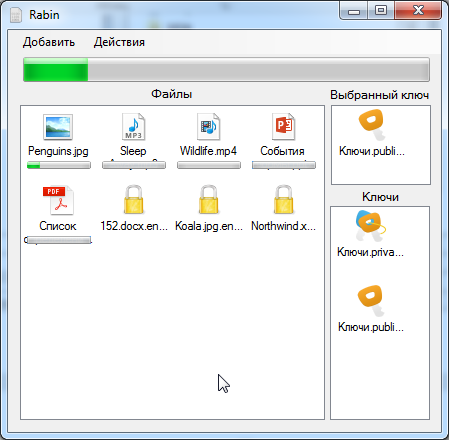
*Рисунок 27*

После этого появится диалог выбора папки, в которую следует сохранить зашифрованные файлы. Выберем папку с ранее сгенерированными ключами, находящуюся на рабочем столе. (рис. 28)



*Рисунок 28*

После выбора папки начнется кодирование файлов. (рис. 29)



*Рисунок 29*

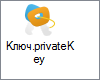
После этого, в окне с файлами вместо шифруемых файлов появятся зашифрованные файлы. (рис. 30)



*Рисунок 30*

### **6.3.4. Испытание выполнения требований к функциональным характеристикам для дешифрования файла**

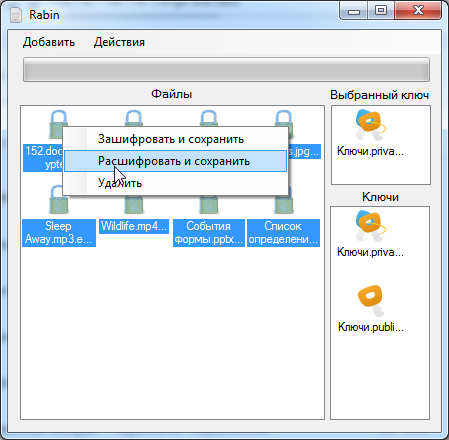
До начала дешифрования, выберем (рис. 31).



*Рисунок 31*

Пример выбора ключа описан в п. 6.3.1.4.

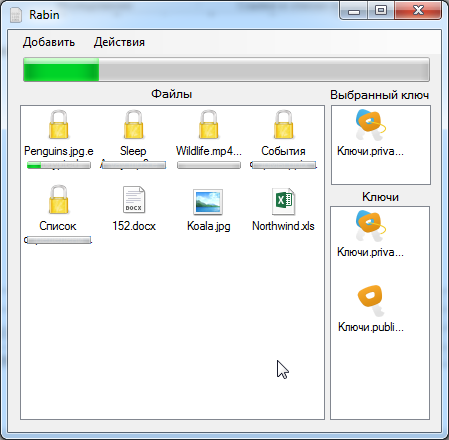
После того, как выбран закрытый ключ, можно переходить к дешифрованию. Расшифруем наши файлы, для этого выделим их и нажмем на один из них правой кнопкой мыши, воспользовавшись всплывающим контекстным меню, нажмем кнопку «Расшифровать и сохранить». (рис. 32)



*Рисунок 32*

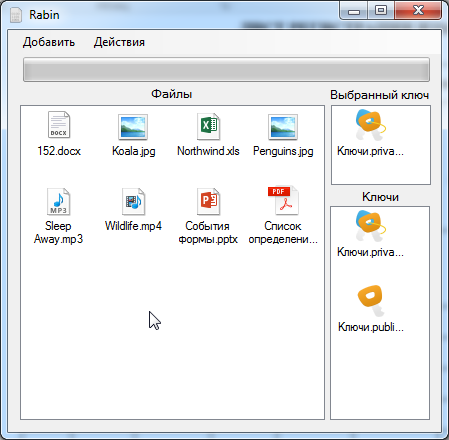
После этого появится диалог выбора папки, в которую сохранятся расшифрованные файлы. (рис. 28) Выберем папку «Desktop \Ключ» и нажмем кнопку «ОК»

После выбора папки начнется декодирование файлов. (рис. 33)



*Рисунок 33*

После этого, в окне с файлами вместо дешифруемых файлов появятся расшифрованные файлы. (рис. 34) Удостоверимся, что расшифрованные файлы являются в точности такими же, как и перед шифрованием.

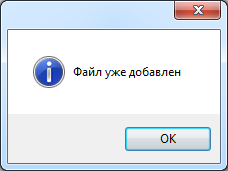


*Рисунок 34*

## **6.4. Испытание выполнения требований к надежности**

### **6.4.1. Защита от добавления двух одинаковых файлов**

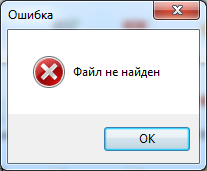
При попытке добавить существующий файл появится предупреждающее сообщение (рис. 35)



*Рисунок 35*

### **6.4.2. Защита от удаления файлов**

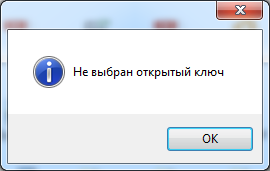
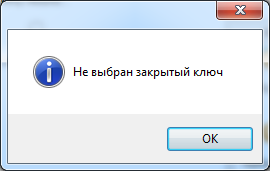
Если добавленный файл по какой-то причине был удален, а пользователь захочет выполнять с ним действия, то появится следующее сообщение (рис. 36), т файл удалится из списка.



*Рисунок 36*

### **6.4.3. Защита от кодирования/декодирования без ключа**

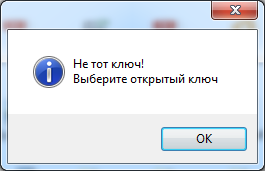
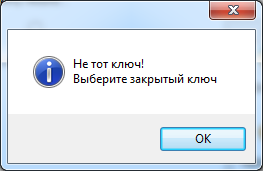
Если пользователь захочет кодировать/декодировать файл без ключа, то появятся следующие сообщения (рис. 37 и рис. 38)

*Рисунок 37 Рисунок 38*

### **6.4.4. Защита от выбора не соответствующего ключа**

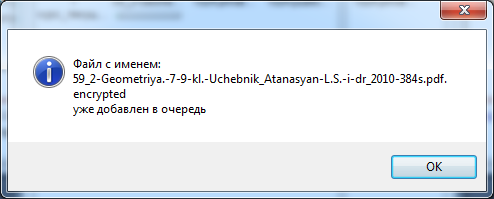
Если пользователь захочет расшифровать открытым ключом или зашифровать закрытым ключом, то выведется следующее сообщение об ошибке (рис. 39 и рис. 40)

ис.ся следующее сообщение об ошибке ()в=ытым ключем е сообщения ()очет  

*Рисунок 39 Рисунок 40*

### **6.4.5. Зашита от кодирования/декодирования одного файла дважды одновременно**

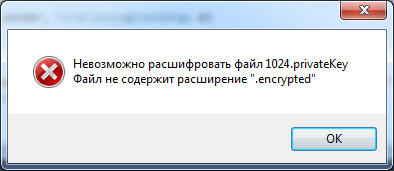
Если пользователь попытается зашифровать/расшифровать файл, который уже находится в очереди, то появится следующее сообщение (рис. 41)



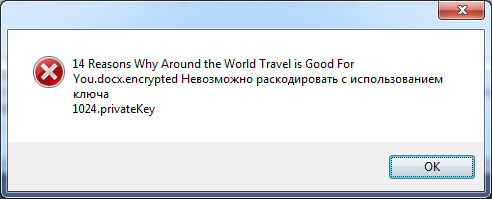
*Рисунок 41*

### **6.4.6. Защита при кодировании/декодировании**

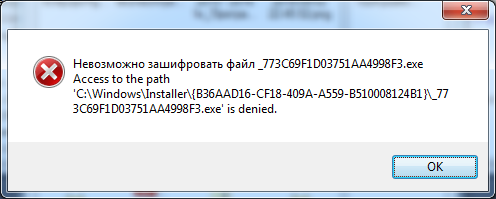
Если файл невозможно зашифровать/расшифровать, то выведутся следующие сообщения. (рис. 42, рис. 43, рис. 44)



*Рисунок 42*



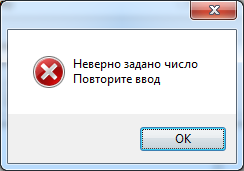
*Рисунок 43*



*Рисунок 44*

### **6.4.7. Зашита при корректной записи числа, при генерации ключей**

Если пользователь запишет число в неверном формате, то выведется следующее сообщение об ошибке (рис. 45)



*Рисунок 45*

**7. ЛИТЕРАТУРА**

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом.
10. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
11. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
14. ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
15. ГОСТ 19.504-79 Руководство программиста //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
16. The Laws of Cryptography: Rabin’s Version of RSA [Электронный ресурс]. URL.: <http://www.cs.utsa.edu/~wagner/laws/Rabin.html> (даты обращения 5.01.2017 - 10.01.2017).
17. Mono Documentation [Электронный ресурс]. URL.: <http://docs.go-mono.com/?link=N%3aMono.Math> (дата обращения 9.01.2017).

# **ПРИЛОЖЕНИЕ 1**

## **СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ**

API интерфейс – Программный интерфейс приложения, интерфейс программирования приложений, интерфейс прикладного программирования.

GUI интерфейс – Графический интерфейс пользователя.

String – Вид информации представленный в текстовом виде.

BigInteger – Вид информации представленный в виде произвольно большого числа со знаком.

Stream – Универсальное представление последовательности байтов.

ГБ – Согласно ГОСТ 8.417—2002 термин гигабайт с обозначением «ГБ» равен 109 = 1 000 000 000 байт.

Гц – Единица частоты периодических процессов ( 1 Гц = 1 с-1 ).

ОЗУ – Оперативно запоминающее устройство

WDDM – это архитектура графических драйверов для видеокарты под управлением Microsoft Windows.

Компакт диск – Лазерный носитель информации в виде диска.

Флеш-драйв – запоминающее устройство, использующее в качестве носителя флеш-память и подключаемое считывающему устройству по USB.

Закрытый ключ – пара простых чисел P и Q, дающих остаток 3 при делении на 4. С помощью закрытого ключа производится дешифровка сообщения.

Открытый ключ – число N, численно равное произведению пары простых чисел P и Q. С помощью открытого ключа производится шифрования сообщения.

# **ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ ФОРМАТА .publicKey**

Файл формата .publicKey содержит открытый ключ записанный в кодировке RabinEncoding.

Преимуществами данной кодировки является уменьшение в 2 раза используемой памяти компьютера по сравнению со стандартной, UTF8Encoding. Кодировка также уменьшает читабельность информации другими программами, что повышает степень защиты от злоумышленников.

Корректно открыть файл .publicKey можно программой «Rabin».

**ОПИСАНИЕ ФОРМАТА .privateKey**

Файл формата .privateKey содержит закрытый ключ записанный в кодировке RabinEncoding.

Преимуществами данной кодировки является уменьшение в 2 раза используемой памяти компьютера по сравнению со стандартной, UTF8Encoding. Кодировка также уменьшает читабельность информации другими программами, что повышает степень защиты от злоумышленников.

Корректно открыть файл .privateKey можно программой «Rabin».

**ОПИСАНИЕ ФОРМАТА .encrypted**

Файл формата .encrypted содержит зашифрованный файл записанный в кодировке RabinEncoding.

Преимуществами данной кодировки является уменьшение в 2 раза используемой памяти компьютера по сравнению со стандартной, UTF8Encoding. Кодировка также уменьшает читабельность информации другими программами, что повышает степень защиты от злоумышленников.

Корректно открыть файл .encrypted можно программой «Rabin».

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |