**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)**

**Пояснительная записка**

Инв. № дубл.

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729. 501430-01 81 01-1**

Взам. инв. №

Исполнитель:

студент группы БПИ 165

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

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

Подп. и дата

Инв. № подл

RU.17701729.

501430-01

81 01-1

**2017**

УТВЕРЖДЕН  
RU.17701729. 501430-01 81 01-1-ЛУ

**КРИПТОСИСТЕМА РАБИНА  
(RABIN CRYPTOSYSTEM)**

**Пояснительная записка**

**RU.17701729. 501430-01 81 01-1**

Подп. и дата

Инв. № дубл.

Взам. инв. №

**Листов 35**

Подп. и дата

Инв. № подл

RU.17701729.

501430-01

81 01-1

**2017АННОТАЦИЯ**

В данном программном продукте приведена пояснительная записка к программе «Криптосистема Рабина (Rabin Cryptosystem)», предназначенная для кодирования и декодирования файлов с использования криптосистемы Рабина.

В разделе «Введение» указано наименование программы и документы, на основе которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

– постановка задачи на разработку программы;

– описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;

– описание и обоснование выбора метода организации входных и выходных данных;

– описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Настоящий документ разработан в соответствии с требованиями:

* ГОСТ 19.101-77 Виды программ и программных документов [1];
* ГОСТ 19.102-77 Стадии разработки [2];
* ГОСТ 19.103-77 Обозначения программ и программных документов [3];
* ГОСТ 19.104-78 Основные надписи [4];
* ГОСТ 19.105-78 Общие требования к программным документам [5];
* ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
* ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 5](#_Toc483387482)

[1.1. Наименование программы 5](#_Toc483387483)

[1.2. Основание для разработки 5](#_Toc483387484)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 6](#_Toc483387485)

[2.1. Назначение программы 6](#_Toc483387486)

[2.1.1. Функциональное назначение программы 6](#_Toc483387487)

[2.1.2. Эксплуатационное назначение 6](#_Toc483387488)

[2.2. Область применения 6](#_Toc483387489)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 7](#_Toc483387490)

[3.1. Постановка задачи на разработку программы 7](#_Toc483387491)

[3.2. Описание применяемых математических методов 7](#_Toc483387492)

[3.3. Общее описание алгоритма криптосистемы Рабина 7](#_Toc483387493)

[3.3.1. Генерация ключа 7](#_Toc483387494)

[3.3.2. Кодирование 7](#_Toc483387495)

[3.3.3. Декодирование 7](#_Toc483387496)

[3.4. Описание ограничений и допущений, связанных с выбранным математическим материалом 8](#_Toc483387497)

[3.5. Описание функционирования программы 8](#_Toc483387498)

[3.5.1. Генерация ключей 8](#_Toc483387499)

[3.5.2. Шифрование файла 9](#_Toc483387500)

[3.5.3. Дешифрование файла 9](#_Toc483387501)

[3.6. Обоснование алгоритма решения задачи 10](#_Toc483387502)

[3.7. Описание и обоснование выбора метода организации входных и выходных данных 10](#_Toc483387503)

[3.8. Реализация программы 11](#_Toc483387504)

[3.8.1. Диаграмма классов программы Rabin 11](#_Toc483387505)

[3.8.2. Диаграмма классов библиотеки RabinLib 12](#_Toc483387506)

[3.9. Описание и обоснование выбора состава технических и программных средств 13](#_Toc483387507)

[3.9.1. Состав технических и программных средств 13](#_Toc483387508)

[3.9.1.1. Состав технических средств 13](#_Toc483387509)

[3.9.1.2. Состав программных средств 13](#_Toc483387510)

[3.9.2. Обоснование выбора технических и программных средств 13](#_Toc483387511)

[3.9.2.1. Обоснование выбора программных средств 13](#_Toc483387512)

[3.9.2.2. Обоснование выбора технических средств 14](#_Toc483387513)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 15](#_Toc483387514)

[4.1. Предполагаемая потребность 15](#_Toc483387515)

[4.2. Экономические преимущества программного обеспечения в сравнении с аналогами 15](#_Toc483387516)

[5. СПИСОК ЛИТЕРАТУРЫ 16](#_Toc483387517)

[ПРИЛОЖЕНИЕ 1 17](#_Toc483387518)

[СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ 17](#_Toc483387519)

[ПРИЛОЖЕНИЕ 2 18](#_Toc483387520)

[ОПИСАНИЕ ФОРМАТА .publicKey 18](#_Toc483387521)

[ОПИСАНИЕ ФОРМАТА .privateKey 18](#_Toc483387522)

[ОПИСАНИЕ ФОРМАТА .encrypted 18](#_Toc483387523)

[ПРИЛОЖЕНИЕ 3 19](#_Toc483387524)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ 19](#_Toc483387525)

[ПРИЛОЖЕНИЕ 4 20](#_Toc483387526)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КОНСТРУКТОРОВ, МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ 20](#_Toc483387527)

# 1. ВВЕДЕНИЕ

## **1.1. Наименование программы**

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

Краткое наименование программы – «Rabin».

Наименование программы на английском языке – «Rabin Cryptosystem».

## **1.2. Основание для разработки**

Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук № 2.3-02/0812-01 от 08.12.2016.

# 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## **2.1. Назначение программы**

### **2.1.1. Функциональное назначение программы**

Данная программа предназначена для шифровки и дешифровки любых файлов, представленных в цифровом виде, реализуя алгоритм криптосистемы Рабина.  
 Программа генерирует ключи для шифрования/дешифрования и сохраняет их в указанный пользователем файл. Используя ключи шифрует/дешифрует указанный файл и сохраняет результат в указанное пользователем место.

### **2.1.2. Эксплуатационное назначение**

Скрытая передача важной информации является востребованной задачей на сегодняшний момент. Программное обеспечение «Криптосистема Рабина (Rabin Cryptosystem)» может быть использована для передачи и получения секретной информации в следующих сферах деятельности:

* В банковской сфере
* На закрытых предприятиях
* Во время военных действий
* Любым человеком, заботящимся о безопасности своих личных данных

## **2.2. 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).

# 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## **3.1. Постановка задачи на разработку программы**

Цель разработки: Предоставить библиотеку, способную генерировать открытый и закрытый ключи. И с использованием соответствующего ключа закодировать или раскодировать файл. Если кодировка или декодировка невозможна, выводить соответствующее сообщение. Также предоставить программный интерфейс, демонстрирующий работу данной библиотеки.

## **3.2. Описание применяемых математических методов**

Криптосистема Рабина [11] – криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших чисел. В силу того, что для больших чисел задача факторизации является вычислительно сложной, криптосистема является доказуемо стойкой к атаке.

## **3.3. Общее описание алгоритма криптосистемы Рабина**

Алгоритм работает следующим образом: Сначала получателем A генерируются открытый [Приложение 1] и закрытый [Приложение 1] ключи. После этого открытый ключ публикуется в общем доступе (например, на сайте получателя A) или просто передается по открытому каналу отправителю B. Далее отправитель B шифрует открытым ключом необходимые файлы и посылает зашифрованные файлы получателю A. Получатель A, имея в своем распоряжении закрытый ключ, расшифровывает полученные файлы. В итоге, сообщение было отправлено от B к A по открытому каналу, но знают содержимое сообщения только A и B.

### **3.3.1. Генерация ключа**

Закрытым ключом является пара простых чисел, дающих остаток 3 при делении на 4. Обозначения: p и q.

Открытым ключом является число, равное произволению пары простых чисел. Обозначение: N ().

### **3.3.2. Кодирование**

Входное значение: сообщение M.

Закодированным сообщением будет являться число C, такое что:

### **3.3.3. Декодирование**

Входное значение: сообщение C (см. п. 3.2.3.)

Алгоритм декодирования следующий:

1. Вычисляем такие R и S, что: , ;
2. С помощью расширенного алгоритма Евклида находим такие A и B, что:

1. Находим четыре числа, одно из которых является посланным сообщением:

## **3.4. Описание ограничений и допущений, связанных с выбранным математическим материалом**

Поскольку в криптосистеме Рабина при декодировке требуется выбрать один корень из четырех ( п. 3.2.4), то было принято решение, при кодировании файлов добавлять 13 фиксированных байтов, с помощью которых при декодировке можно было определить верное сообщение. Из-за этого вероятность ошибки при декодировке с верным ключом составляет:

Также, из-за этого, минимальная возможная бинарная длина открытого ключа составляет: 128 бит.

## **3.5. Описание функционирования программы**

### **3.5.1. Генерация ключей**

Поскольку закрытым ключом является пара простых чисел, дающих остаток 3 по модулю 4, в дальнейшем будем называть их P и Q. А открытым ключом является число, равное перемножению P и Q, будем называть его N. То алгоритм генерации ключей выглядит следующим образом:

Входные данные: Бинарная длина открытого ключа N

1. Вычисление необходимой бинарной длины числа P (Защита от алгоритма Ферма (длина P ≈ длина N \* (0.5 – Random (0.05, 0.1)))
2. Генерация числа P

3.1. Генерация простого числа [12] с длинной полученной в п. 2

3.2. Если число дает остаток 3 по модулю 4, то переход к п. 4. В противном случае к п. 3.1.

1. Аналоготивном случае к п. открытого ключа ующим образом:ичным образом генерируем число Q (длина Q = длина N – длина P)
2. Находим N = P \* Q

Входные данные: Открытый и закрытый ключи сохраненные в файлах с расширениями .publicKey [Приложение 2] и .privateKey [Приложение 2].

Все процессы генераций проводятся случайным образом, иначе злоумышленник, зная алгоритм, может найти способ разгадать секретный ключ.

### **3.5.2. Шифрование файла**

Входные данные: Открытый ключ (N) и шифруемый файл

1. Используя N, вычисляется максимальная длина числа, которую можно зашифровать, с учетом фиксированных символов, которые в дальнейшем помогут выбрать один из четырех вариантов ответов
2. Если файл не пуст, то из него бинарно считывается кусок, с учетом посчитанной длины в п. В противном случае шифрование завершается.
3. Добавляя фиксированные символы, кусок переводится в число, с учетом того, что передние нули не отбрасываются.
4. Полученное число в п. 4 возводится в квадрат по модулю N, и записывается в файл. Переход к п. 3.

Входные данные: Зашифрованный файл записанный на диск с расширением .encrypted [Приложение 2].

### **3.5.3. Дешифрование файла**

Входные данные: Закрытый ключ (P и Q) и дешифруемый файл.

1. Если файл не пуст, то считывается кусок и переводится в число. В противном случае завершение дешифрования.
2. С использованием числа из п. 2, вычисляем:

R = C ^((P + 1) / 4) % P

S = C^((Q + 1) / 4) % Q

1. Находим A и B, такие что: A \* P + B \* Q = 1 (Расширенный алгоритм Евклида)
2. Находим предполагаемые сообщения:

m1 = (A \* P \* S + B \* Q \* R) % N

m2 = N – m1

m3 = (A \* P \* S - B \* Q \* R) % N

m4 = N – m3

1. С помощью фиксированных символов, используемых в шифровании, выбираем верное сообщение из m1 – m4 и записываем его в файл. Переход к п. 2.

Выходное значение: расшифрованный файл

## **3.6. Обоснование алгоритма решения задачи**

Криптосистема Рабина схожа со всеми известной криптосистемой RSA, также основанная на сложности факторизации больших чисел, и используемой людьми по всему миру. И поскольку проблема факторизации чисел остается нерешенной, алгоритм обеспечивает доказуемую стойкость к атаке. И, в добавок, данная криптосистема не на столько известна миру, что усложняет взлом системы.  
 С принятием во внимание этих факторов, было принять решение создать шифрующую программу на основе криптосистемы Рабина.

## **3.7. Описание и обоснование выбора метода организации входных и выходных данных**

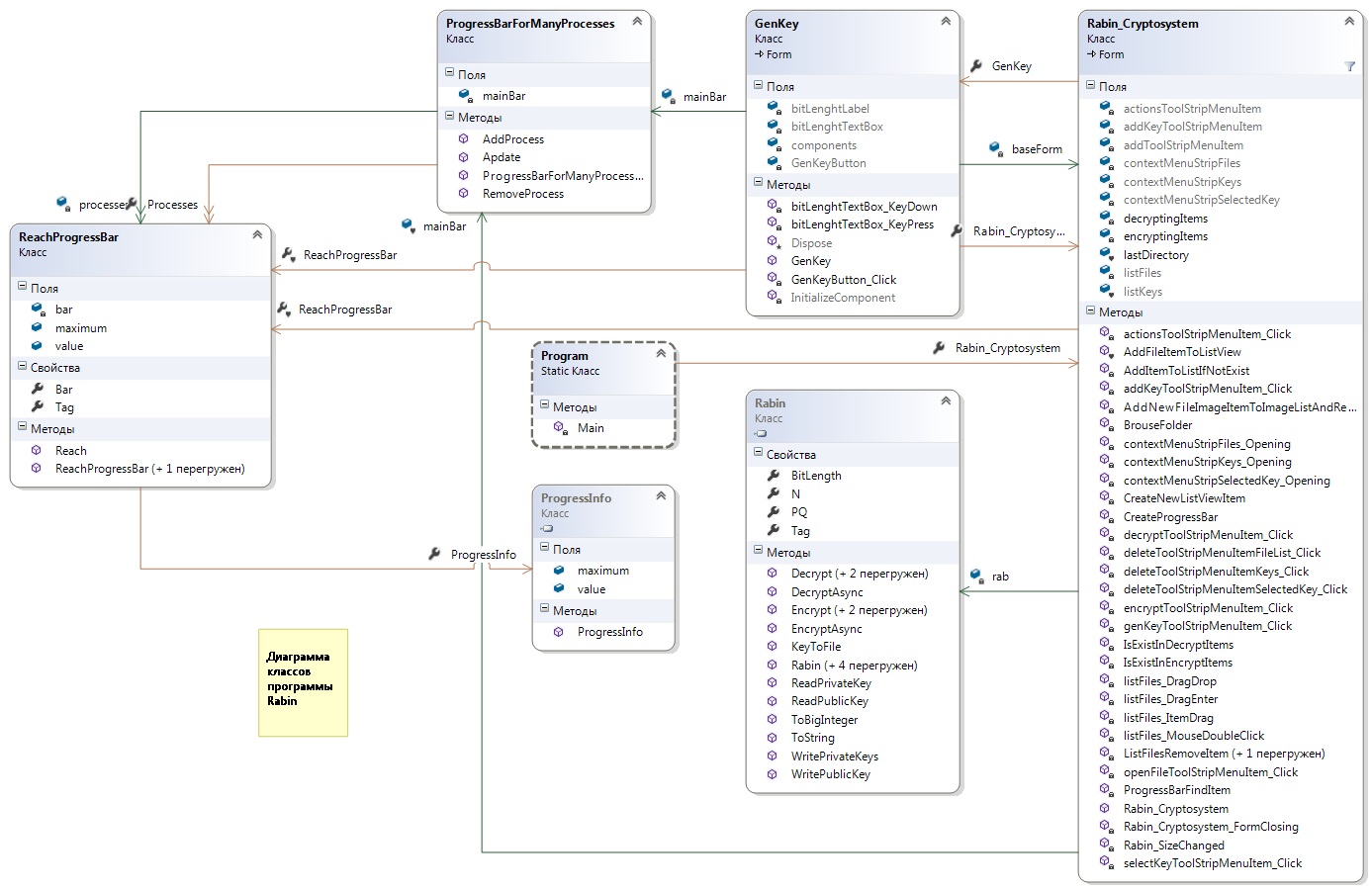
Входными и выходными данными программы являются файлы следующего формата:

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

Поскольку файлы с расширениями .publicKey, .privateKey и .encrypted будут содержать только цифры от 0 до 9 и символ разделителя, то было принято решение записывать информацию в новой кодировке, RabinEncoding, что позволило уменьшить вдвое размер данных файлов и понизить их читабельность для злоумышленника не знающего кодировки, что позволяет также увеличить криптостойкость.

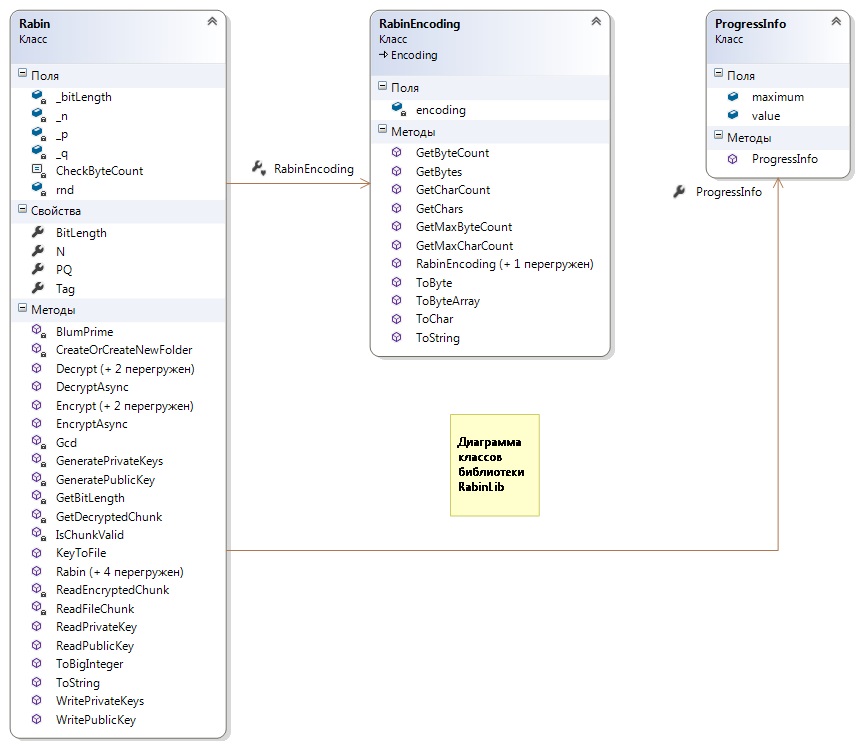
**3.8. Реализация программы**

### **3.8.1. Диаграмма классов программы Rabin**



Описание и функциональное назначение классов смотри в Приложении 3. Описание и функциональное назначение конструкторов, методов, полей и свойств данных классов смотри в Приложении 4

### **3.8.2. Диаграмма классов библиотеки RabinLib**

****

Описание и функциональное назначение классов смотри в Приложении 3. Описание и функциональное назначение конструкторов, методов, полей и свойств данных классов смотри в Приложении 4.

**3.9. Описание и обоснование выбора состава технических и программных средств**

**3.9.1. Состав технических и программных средств**

**3.9.1.1. Состав технических средств**

Для корректной работы программы необходим 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

**3.9.1.2. Состав программных средств**

Для корректной работы программы необходим следующий состав программных средств:

* Операционная система Microsoft Windows 7 или более поздняя версия.
* Microsoft .NET Framework 4.5.2.

**3.9.2. Обоснование выбора технических и программных средств**

**3.9.2.1. Обоснование выбора программных средств**

Разработка данного программного обеспечения велась на платформе Microsoft .NET Framework 4.5.2. Данная среда разработки была принята из-за нескольких причин:

* В предоставленном программном обеспечении было использовано множество конструкций асинхронного кода (например таких как async await). А поддержка асинхронного кода была добавлена в .NET Framework 4.5 [15]
* Также в программе было использовано MenuStrip, а в .NET Framework 4.5.2 были добавлены улучшения для ToolStripMenuItem. [10, 16]
* Ну и поскольку из-за важности возможностей описанных в п. 1 поддержка Windows XP стала невозможна, было принять решение вести разработку на более современной версии, .NET Framework 4.5.2, обеспечивающей большую производительность и эффективность программного обеспечения.

Операционная система, стабильно поддерживающая работу данной платформы, является Windows 7 [14] и выше.

**3.9.2.2. Обоснование выбора технических средств**

Разработка данного программного обеспечения велась на платформе Microsoft .NET Framework 4.5.2. Данная платформа, в свою очередь, требует следующий состав технических средств [13]:

* 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.

Для беспроблемного управления программой необходимы:

* Мышь или совместимое указывающее устройство.
* Клавиатура.

Для установки программы с компакт диска [Приложение 1] или флеш-драйва [Приложение 1] в свою очередь требуется:

* CD-ROM Drive
* USB-порт

**4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

**4.1. Предполагаемая потребность**

«Криптосистема Рабина (Rabin Cryptosystem)» может быть использована в следующих сферах:

* В банковской сфере
* На закрытых предприятиях
* Во время военных действий
* Любым человеком, заботящимся о безопасности своих личных данных

**4.2. Экономические преимущества программного обеспечения в сравнении с аналогами**

«Криптосистема Рабина (Rabin Cryptosystem)» обладает следующими преимуществами:

* В данной программе используется алгоритм криптосистемы Рабина с открытым ключом, о котором не известно многим злоумышленникам.
* В отличие от многих других программ, основанных на криптосистеме Рабина, сгенерированные ключи защищены от «метода Ферма» (этот метод позволяет довольно быстро по открытому ключу найти закрытый, и расшифровать сообщение).
* Позволяет зашифровать/расшифровать любую бинарную информацию (любой файл).
* Позволяет зашифровать/расшифровать информацию неограниченного объема.
* Данное программное обеспечение распространяется бесплатно.
* Не требует вложения денежных средств во время использования.
* Имеет неограниченный срок службы.
* Интуитивно просто в использовании.

**5. СПИСОК ЛИТЕРАТУРЫ**

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 19.404-79 Пояснительная записка. 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. Microsoft Developer Network [Электронный ресурс]. URL.: <https://msdn.microsoft.com> (1.09.2016 - ?).
11. 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).
12. Mono Documentation [Электронный ресурс]. URL.: <http://docs.go-mono.com/?link=N%3aMono.Math> (дата обращения 9.01.2017).
13. Требования к системе для .NET Framework [Электронный ресурс]. URL.: <https://msdn.microsoft.com/ru-ru/library/8z6watww(v=vs.110).aspx> (дата обращения: 10.04.2017).
14. Системные требования для ОС Windows 7 [Электронный ресурс]. URL.: <https://support.microsoft.com/ru-ru/help/10737/windows-7-system-requirements>(дата обращения: 10.04.2017).
15. Microsoft .NET Framework 4.5 [Электронный ресурс]. URL.: [https://netframework.ru/8-net-framework-45.html](https://netframework.ru/8-net-framework-45.html%20) (дата обращения: 10.04.2017).
16. Новые возможности Microsoft .NET Framework [Электронный ресурс]. URL.: [https://msdn.microsoft.com/ru-ru/library/ms171868(v=vs.110).aspx#Новые возможности .NET Framework 4.5](https://msdn.microsoft.com/ru-ru/library/ms171868(v=vs.110).aspx%23Новые%20возможности%20.NET%20Framework%204.5) (дата обращения: 10.04.2017).

# ПРИЛОЖЕНИЕ 1

## **СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ**

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.

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».

**ПРИЛОЖЕНИЕ 3**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

|  |  |
| --- | --- |
| **Класс** | **Назначение** |
| Rabin.cs | Основной класс библиотеки RabinLib.dll. В этом классе реализована генерация открытых и закрытых ключей, кодирование и декодирование всех видов информации. |
| RabinEncoding.cs | Класс, наследуемый от System.Text.Encoding.cs и реализующий кодировку, с помощью которой ведется запись информации в зашифрованные файлы и файлы с ключами. |
| ProgressInfo.cs | Класс, реализуемый для System.IProgress<T>, представляющий формат отчета, описывающего ход процесса. |
| Rabin\_Cryptosystem.cs | Класс, описывающий логику действий интерфейса основного окна программы. |
| GenKey.cs | Класс, описывающий логику действий интерфейса диалогового окна, общающегося с пользователем при генерации ключей. |
| ReachProgressBar.cs | Класс, описывающий логику отображения прогресса, для отдельных процессов. |
| ProgressBarForManyProcesses.cs | Класс, описывающий логику отображения общего процесса. |

**ПРИЛОЖЕНИЕ 4**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КОНСТРУКТОРОВ, МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ**

Rabin.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Аргументы* | | *Назначение* |
| Generate  PrivateKeys | private | void | int bitLength, IProgress  <ProgressInfo> progress = null | | Генерирует закрытые ключи и отсылает информацию о процессе, если progress инициализирован пользователем. И инициализирует поля \_p и \_q. |
| Generate  PublicKey | private | void | - | | Подсчитывает открытый ключ и определяет его бинарную длину. И инициализирует поля \_n и \_bitLength. |
| GetBitLength | private | void | - | | Считает бинарную длину открытого ключа. И инициализирует поле \_bitLength. |
| BlumPrime | private static | BigInteger | int bitLength | | Генерирует простое число, дающее остаток 3 по модулю 4. |
| Decrypt | public | BigInteger[] | BigInteger c | | Используя поля \_p и \_q, дешифрует сообщение c и возвращает массив с четырьмя шифротекстами, один из которых верный. |
| Gcd | private static | BigInteger | BigInteger a, BigInteger b, out BigInteger x, out BigInteger y | | Реализует расширенный алгоритм Евклида. |
| Encrypt | public | BigInteger | BigInteger m | | Используя \_m, шифрует сообщение. |
| ToBigInteger | public | BigInteger | string str | | Переводит строку в число, используя кодировку Unicode. |
| ToString | public | string | BigInteger m | | Переводит число в строку, используя кодировку Unicode. |
| KeyToFile | public | void | String keysPath | | Сохраняет открытый и закрытый ключи в указанное место на диске. |
| Write  PublicKey | public | void | string publicKeyPath | | Записывает открытый ключ в указанный файл в кодировке RabinEncoding. |
| Write  PrivateKeys | public | void | string privateKeyPath | | Записывает закрытый ключ в указанный файл в кодировке RabinEncoding. |
| CreateOr  CreateNewFolder | private | void | string folderPath | | Создает новую папку в указанном месте с указанным названием |
| ReadPublicKey | public | static BigInteger | string publicKeyPath | | Считывает из указанного файла открытый ключ и возвращает его численное значение. |
| ReadPrivateKey | public | static BigInteger[] | string privateKeyPath | | Считывает из указанного файла закрытые ключи и  возвращает массив с их численными значениями. |
| Encrypt | public | void | string filename, string encryptPath, IProgress  <ProgressInfo> progress = null | | Используя поле \_n, шифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе, если progress инициализирован пользователем. |
| EncryptAsync | public | async Task | string filename, string encryptPath, IProgress  <ProgressInfo> progress | | Используя поле \_n, асинхронно шифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе. |
| Encrypt | public | void | Stream read, Stream write, IProgress  <ProgressInfo> progress | | Используя поле \_n, кодирует информацию из первого объекта Stream и записывает ее во второй объект Stream. И отсылает информацию о процессе. |
| ReadFileChunk | private | byte[] | BinaryReader br, int byteLength | | Бинарно считывает из файла кусок указанной длины. |
| Decrypt | public | void | string filename, string decryptPath, IProgress  <ProgressInfo> progress = null | | Используя поля \_p и \_q, дешифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе, если progress инициализирован пользователем. |
| DecryptAsync | public | async Task | string filename, string decryptPath, IProgress  <ProgressInfo> progress | | Используя поля \_p и \_q, асинхронно дешифрует файл с именем filename и сохраняет зашифрованный файл в указанном месте. Отсылает информацию о процессе, если progress инициализирован пользователем. |
| Decrypt | public | void | Stream read, Stream write, IProgress  <ProgressInfo> progress | | Используя поля \_p и \_q, декодирует сообщение из первого объекта Stream и записывает ее во второй объект Stream. И отсылает информацию о процессе. |
| Read  EncryptedChunk | private | string | StreamReader sr | | Считывает закодированный кусок из указанного места. |
| Get  DecryptedChunk | private | byte[] | BigInteger[] numbers | | Выбирает верный раскодированный кусок и возвращает его. |
| IsChunkValid | private | bool | byte[] chunk | | Проверяет, является ли кусок верным. |
| **Конструкторы** | | | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | | *Назначение* | |
| Rabin | public | int bitLength | | Инициализирует поля \_n, \_p, \_q, \_bitLength, генерируя открытые ключи и считая закрытый. | |
| Rabin | public | int bitLength, IProgress  <ProgressInfo> progress | | Инициализирует поля \_n, \_p, \_q, \_bitLength, генерируя открытые ключи и считая закрытый. И отсылает информацию о прогрессе. | |
| Rabin | public | BigInteger n | | Инициализирует поля \_n, \_bitLength. | |
| Rabin | public | BigInteger p, BigInteger q | | Инициализирует поля \_n, \_p, \_q, \_bitLength. | |
| Rabin | public | string keyPath | | Считывает из файла открытый или закрытые ключ, в зависимости от самого файла. | |
| **Поля** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | | *Назначение* | |
| \_n | private | BigInteger | | Численное значение открытого ключа. | |
| \_q | private | BigInteger | | Одно из двух численных значений закрытого ключа. | |
| \_p | private | BigInteger | | Одно из двух численных значений закрытого ключа. | |
| \_bitLength | private | int | | Бинарная длина открытого ключа. | |
| CheckByteCount | private | const int | | Количество фиксированных битов для определения одного верного из четырех сообщений. | |
| Rnd | private | static Random | | Объект класса Random. | |
| **Свойства** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | | *Назначение* | |
| Tag | public | object | | Возвращает или задает объект, содержащий данные об объекте. | |
| PQ | public | BigInteger[] | | Возвращает массив BigInteger, который содержит 2 закрытых ключа. | |
| N | public | BigInteger | | Возвращает открытый ключ. | |
| BitLength | public | int | | Возвращает бинарную длину открытого ключа. | |

RabinEncoding.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Аргументы* | *Назначение* |
| ToByte | public | byte | char simbol | Переводит символ в его числовое представление в кодировке RabinEncoding. |
| ToChar | public | char | byte number | Переводит число в символ, используя кодировку RabinEncoding. |
| ToByteArray | public | byte[] | string input | Переводит строку в массив байтов, используя кодировку RabinEncoding. |
| ToString | public | string | byte[] input | Переводит массив байтов в строку, используя кодировку RabinEncoding. |
| GetBytes | public | override int | char[] chars, int charIndex, int charCount, byte[] bytes, int byteIndex | Кодирует набор символов из указанного массива символов в указанный массив байтов, используя кодировку RabinEncoding. |
| GetCharCount | public | override int | byte[] bytes, int index, int count | Вычисляет количество символов, полученных при декодировании последовательности байтов из заданного массива байтов, используя кодировку RabinEncoding. |
| GetByteCount | public | override int | char[] chars, int index, int count | Вычисляет количество байтов, полученных при кодировании набора символов из указанного массива символов, используя кодировку RabinEncoding. |
| GetChars | public | override int | byte[] bytes, int byteIndex, int byteCount, char[] chars, int charIndex | Декодирует последовательность байтов из указанного массива байтов в указанный массив символов, используя кодировку RabinEncoding. |
| GetMaxByteCount | public | override int | int charCount | Вычисляет максимальное количество байтов, полученных при кодировании заданного количества символов, используя кодировку RabinEncoding. |
| GetMaxCharCount | public | override int | int byteCount | Вычисляет максимальное количество символов, полученных при декодировании заданного количества байтов, используя кодировку RabinEncoding. |
|  | | | | |
| **Конструкторы** | | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | *Назначение* | |
| RabinEncoding | public | - | Инициализирует поле byte[] encoding базовыми значениями. | |
| RabinEncoding | public | byte[] newEncoding | Инициализирует поле byte[] encoding с помощью массива newEncoding. | |
| **Поля** | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Назначение* | |
| Encoding | private | byte[] | Задает кодировку для символов цифр и символа разделителя. | |

ProgressInfo.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Конструкторы** | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | *Назначение* |
| ProgressInfo | public | long newValue, long newMaximum | Инициализирует объект ProgressInfo, задавая новое положение и максимальное значение. |
| **Поля** | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Назначение* |
| Value | public | readonly long | Положение в данный момент. |
| maximum | public | readonly long | Максимальное значение. |

ReachProgressBar.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Аргументы* | *Назначение* |
| Reach | public | void | object sender, ProgressInfo e | Описывает действия, совершаемые после события изменения прогресса. |
| **Конструкторы** | | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | | *Назначение* |
| ReachProgressBar | public | ProgressBar myBar | | Создает объект ReachProgressBar, используя данный ProgressBar |
| ReachProgressBar | public | int value, int maximum | | Создает объект ReachProgressBar, используя поступившее настоящее значение и максимальное значенте. |
| **Поля** | | | | |
| *Имя* | *Мод. доступа* | *Тип* | | *Назначение* |
| Bar | private | ProgressBar | | ProgressBar, связанный с данным объектом |
| Value | public | int | | Значение в данный момент. |
| Maximum | public | int | | Максимальное значение. |
| **Свойства** | | | | |
| *Имя* | *Мод. доступа* | *Тип* | | *Назначение* |
| Bar | public | ProgressBar | | Возвращает ProgressBar, связанный с данным объектом |
| Tag | public | object | | Возвращает или задает объект, содержащий данные об объекте. |

ProgressBarForManyProcesses.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Аргументы* | | *Назначение* |
| AddProcess | public | void | ReachProgressBar process | | Добавляет указанный процесс в коллекцию. |
| RemoveProcess | public | void | ReachProgressBar process | | Удаляет указанный процесс из коллекции. |
| Apdate | public | void | object sender, ProgressInfo e | | Описывает действия, совершаемые после события изменения прогресса. |
| **Конструкторы** | | | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | | *Назначение* | |
| ProgressBarFor  ManyProcesses | public | ProgressBar bar | | Создает объект ProgressBarFor  ManyProcesses инициализируя поле mainBar. | |
| ProgressBarFor  ManyProcesses | public | ProgressBar bar, List<ReachProgressBar> listOfNewProcesses | | Создает объект ProgressBarFor  ManyProcesses инициализируя поля mainBar и processes. | |
| **Поля** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | | *Назначение* | |
| Processes | private | List<ReachProgressBar> | | Представляет коллекцию процессов | |
| mainBar | Private | ProgressBar | | Является ссылкой на главный ProgressBar | |

Rabin\_Cryptosystem.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Аргументы* | *Назначение* |
| openFileToolStrip  MenuItem\_Click | private | void | object sender, EventArgs e | Позволяет пользователю выбрать файлы и добавить их в список файлов |
| addKeyToolStrip  MenuItem\_Click | private | void | object sender, EventArgs e | Позволяет пользователю выбрать ключи и добавить их в коллекцию ключей |
| AddFileItem  ToListView | internal | void | string fileName, ListView listToAdd, ImageList imageList | Добавляет файл с указанным путем в указанный ListView, используя указанный ImageList |
| AddNewFile  ImageItemTo  ImageListAnd  ReturnsPictureIndex | private | int | string fileName, ImageList imageList | Добавляет иконку файла в указанный ImageList и возвращает индекс этой иконки |
| CreateNew  ListViewItem | private | ListViewItem | string fileName, int pictureIndex | Создает и возвращает новый ListViewItem с указанным путем к файлу и индексом картинки |
| AddItemToList  IfNotExist | private | void | ListViewItem newItem, ListView listToAdd | Добавляет указанный ListViewItem в ListView, если такой еще не добавлен |
| CreateProgressBar | private | ProgressBar | ListViewItem item, ListView listToAdd | Создает новый ProgressBar, привязывая его к указанному ListViewItem в указанном ListView |
| contextMenuStrip  Files\_Opening | private | void | object sender, CancelEventArgs e | Задает параметры contextMenuStripFiles |
| genKeyToolStrip  MenuItem\_Click | private | void | object sender, EventArgs e | Открывает форму GenKey |
| contextMenuStrip  Keys\_Opening | public | void | object sender, CancelEventArgs e | Задает параметры contextMenuStripKeys |
| selectKeyToolStrip  MenuItem\_Click | private | void | object sender, EventArgs e | Выполняет выбор ключа |
| contextMenuStrip  SelectedKey\_Opening | private | void | object sender, CancelEventArgs e | Задает параметры contextMenuStripselected  Keys |
| deleteToolStrip  MenuItemSelected  Key\_Click | private | void | object sender, EventArgs e | Удаляет выбранный ключ из listSelected  Key |
| encryptToolStrip  MenuItem\_Click | private | async void | object sender, EventArgs e | С использованием выбранного ключа, асинхронно кодирует выбранные файлы |
| IsExistIn  EncryptItems | private | bool | ListViewItem item | Определяет находится ли элемент в очереди на кодирование |
| listFiles\_Mouse  DoubleClick | private | void | object sender, MouseEventArgs e | Открывает файл используя программу по умолчанию. |
| decryptToolStrip  MenuItem\_Click | private | async void | object sender, EventArgs e | С использованием выбранного ключа, асинхронно декодирует выбранные файлы |
| IsExistInDecrypt  Items | private | bool | ListViewItem item | Определяет находится ли элемент в очереди на декодирование |
| deleteToolStrip  MenuItemFileList  \_Click | private | void | object sender, EventArgs | Удаляет файл из коллекции файлов |
| deleteToolStrip  MenuItemKeys\_Click | private | void | object sender, EventArgs e | Удаляет ключ из коллекции ключей |
| ListFilesRemoveItem | private | void | ListViewItem item | Удаляет указанный файл из коллекции файлов и правит расположение элементов |
| ListFilesRemoveItem | private | void | int i | Удаляет файл по указанному индексу из коллекции файлов и правит расположение элементов |
| ProgressBarFindItem | private | void | ListView list | Правит расположение элементов в listFiles |
| listFiles\_ItemDrag | private | void | object sender, ItemDragEventArgs e | Визуализирует перемещение элементов в listFiles |
| listFiles\_DragEnter | private | void | object sender, DragEventArgs e | Визуализирует перемещение элементов в listFiles |
| listFiles\_DragDrop | private | void | object sender, DragEventArgs e | Визуализирует перемещение элементов в listFiles |
| Rabin\_Cryptosystem  \_FormClosing | private | void | object sender, FormClosingEventArgs e | Предупреждает пользователя о незавершенных процессах, если такие есть |
| actionsToolStrip  MenuItem\_Click | private | void | object sender, EventArgs e | Визуализирует меню actionsToolStripMenu |
| Cryptosystem\_Rabin  \_SizeChanged | private | void | object sender, EventArgs e | Является обработчиком события изменения размеры главной формы |
| Dispose | protected | override void | bool disposing | Очищает все использованные ресурсы |
| InitializeComponent | private | void | - | Описывает интерфейс основного окна программы |
| **Конструкторы** | | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | *Назначение* | |
| Rabin\_Cryptosystem | public | string[] args | Загружает основное окно программы, и открывает файлы из args[], если это возможно | |
| **Поля** | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Назначение* | |
| Rab | private | Rabin | Объект класса для кодирования и декодирования | |
| lastDirectory | private | string | Путь к последней открытой дирректории | |
| encryptingItems | private | List<ListViewItem> | Очередь кодируемых файлов | |
| decryptingItems | private | List<ListViewItem> | Очередь декодируемых файлов | |
| mainBar | internal | ProgressBarFor  ManyProcesses | ProgressBar для всех процессов | |
| menuStrip1 | private | MenuStrip | Главное меню | |
| addToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Добавить» в главном меню | |
| openFileToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Открыть файл» в главном меню | |
| KeyToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Ключ» в главном меню | |
| genKeyToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Сгенерировать ключ» в главном меню | |
| addKeyToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Открыть ключ» в главном меню | |
| progressBar1 | private | ProgressBar | Главный ProgressBar | |
| listFiles | private | ListView | Список файлов | |
| imageList1 | private | ImageList | Коллекция картинок | |
| listKeys | private | ListView | Список ключей | |
| contextMenu  StripFiles | private | ContextMenuStrip | Контекстное меню для listFiles | |
| encryptToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Кодировать и сохранить» в контекстном меню для listFiles | |
| decryptToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Декодировать и сохранить» в контекстном меню для listFiles | |
| deleteToolStrip  MenuItemFileList | private | ToolStripMenuItem | Кнопка «Удалить» в контекстном меню для listFiles | |
| listSelectedKey | private | ListView | Список с выбранным ключем | |
| label1 | private | Label | Надпись «Файлы» | |
| label2 | private | Label | Надпись «Выбранный ключ» | |
| label3 | private | Label | Надпись «Ключи» | |
| contextMenu  StripKeys | private | ContextMenuStrip | Контекстное меню для listKeys | |
| выбратьКлюч  ToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Выбрать» в контекстном меню для listKeys | |
| deleteTool  StripMenu  ItemKeys | private | ToolStripMenuItem | Кнопка «Удалить» в контекстном меню для listKeys | |
| contextMenuStrip  SelectedKey | private | ToolStripMenuItem | Контекстное меню для listSelectedKeys | |
| deleteToolStrip  MenuItem  SelectedKey | private | ToolStripMenuItem | Кнопка «Удалить» в контекстном меню для listSelectedKeys | |
| actionsToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Действия» в главном меню | |
| encrypt2ToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Закодировать и сохранить» в главном меню | |
| decrypt2ToolStrip  MenuItem | private | ToolStripMenuItem | Кнопка «Раскодировать и сохранить» в главном меню | |
| components | private | IContainer | Переменная конструктора | |

GenKey.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | *Аргументы* | | *Назначение* |
| GenKeyButton\_Click | private | async void | object sender, EventArgs e | | Асинхронно генерирует ключи |
| bitLenghtText  Box\_KeyDown | private | Void | object sender, KeyEventArgs e | | При нажатии клавиши Enter обращается к методу  GenKeyButton\_Click |
| bitLenghtText  Box\_KeyPress | private | Void | object sender, KeyPressEventArgs e | | Позволяет записывать в textbox только цифры |
| Dispose | protected | override void | bool disposing | | Очищает все использованные ресурсы |
| InitializeComponent | private | void | - | | Описывает интерфейс окна генерации ключа |
| **Конструкторы** | | | | | |
| *Имя* | *Мод. доступа* | *Аргументы* | | *Назначение* | |
| GenKey | public | Rabin\_Cryptosystem form | | Загружает окно генерации ключа получая ссылку на базовую форму | |
| **Поля** | | | | | |
| *Имя* | *Мод. доступа* | *Тип* | | *Назначение* | |
| Rabin\_Cryptosystem | private | baseForm | | Ссылка на базовую форму | |
| ProgressBarFor  ManyProcesses | private | mainBar | | Ссылка на основной процесс | |
| bitLenghtLabel | private | Label | | Надпись описывающая назначение TextBox | |
| bitLenghtTextBox | private | TextBox | | Окно, для записи длины открытого ключа | |
| GenKeyButton | private | Button | | Кнопка для старта генерирования ключа | |
| components | private | IContainer | | Переменная конструктора | |

**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 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |