

Правительство Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(НИУ ВШЭ)

Московский институт электроники и математики им. А.Н.Тихонова

ОТЧЕТ
О ПРАКТИЧЕСКОЙ РАБОТЕ № 1
по дисциплине «Криптографические методы защиты информации»
ТЕМА РАБОТЫ
АССИМЕТРИЧНАЯ СИСТЕМА RSA

Студент гр. МКБ231

«__» _____ 2024 г.

Руководитель
Заведующий кафедрой информационной
безопасности киберфизических систем
канд. техн. наук, доцент

«__» _____ 2024 г.

Москва 2024

СОДЕРЖАНИЕ

1. Задание на практическую работу	3
2. Описание алгоритма криптосистемы RSA.....	4
2.1 Описание процедуры генерации ключей.....	4
2.2 Описание процедуры зашифрования	4
2.3 Описание процедуры расшифрования.....	4
3. Реализация алгоритма RSA	5
3.1 Общие сведения	5
3.2 Описание работы программы.....	5
4. Результаты работы программы	6
4.1 Входные данные	6
4.2 Выходные данные.....	6
4.3 Использование программы	6
4.4 Результат работы программы(консольный вывод)	6
5 Выводы о проделанной работе.....	7
6 Список использованных источников.....	8
ПРИЛОЖЕНИЕ А. Листинг программного кода	8

1. Задание на практическую работу

Целью работы является создание программной реализации ассиметричной системы RSA с использованием больших чисел.

В рамках практической работы необходимо выполнить следующее:

1. Написать программную реализацию криптосистемы RSA. Программа должна:
 - принимать на вход файл, содержащий открытый текст (для зашифрования) или шифр текст (для расшифрования)
 - принимать на вход ключевую пару (открытый и закрытые ключи)
 - предоставлять возможность генерации ключевой пары
 - осуществлять зашифрование или расшифрование
2. Составить описание работы алгоритма криптосистемы RSA
3. Реализовать не менее одной атаки на криптосистему RSA
4. Подготовить отчет о проделанной работе.

2. Описание алгоритма криптосистемы RSA

Криптосистема RSA является криптосистемой с открытым ключом. Она основывается на сложности проблемы факторизации целых чисел (разложение числа на простые множители)

2.1 Описание процедуры генерации ключей

Генерация ключей происходит следующим образом:

1. Пользователь А генерирует два больших простых числа p и q , отличных друг от друга, при этом $|p - q|$ большое число, хотя p и q числа примерно одинакового размера.
2. Держа p и q в секрете, А вычисляет их произведение $n = p \cdot q$ – это будет модуль алгоритма.
3. Пользователь А вычисляет значение функции Эйлера для n по формуле $\varphi(n) = (p - 1)(q - 1)$.
4. Пользователь А выбирает целое число e , взаимно простое со значением функции $\varphi(n)$. Это число называется экспонентой зашифрования.
5. Пользователь А применяет расширенный алгоритм Евклида к паре чисел e и $\varphi(n)$ и вычисляет значение d , удовлетворяющее соотношению $ed \equiv 1 \pmod{\varphi(n)}$. Это значение называется экспонентой расшифрования.
6. Пара (e, n) публикуется в качестве открытого ключа пользователя А, d является закрытым ключом и держится в секрете.

2.2 Описание процедуры зашифрования

1. Пользователь получает копию открытого ключа пользователя А – пару чисел (n, e)
2. Пользователь В представляет сообщение в виде числа m , меньшего модуля алгоритма n . Если сообщение большое, тогда оно разбивается на блоки, каждый из которых представляется своим числом.
3. Пользователь В вычисляет $c = m^e \pmod{n}$
4. Зашифрованное сообщение c отправляется А.

2.3 Описание процедуры расшифрования

1. Пользователь А получает шифр текст c от В.
2. Пользователь А вычисляет $m = c^d \pmod{n}$.

3. Реализация алгоритма RSA

3.1 Общие сведения

Программа за(рас)шифрования реализована на языке программирования C# в виде консольного приложения с использованием IDE Visual Studio 2022 и включает в себя три файла: Program.cs, GenExpEnc.cs, GenPrime.cs, ComRSA.cs.

3.2 Описание работы программы

1. Файл сообщения или шифр текста загружается в большое число, которое разбивается на блоки в зависимости от значения модуля алгоритма.
2. Файлы ключей (открытых и закрытых) загружаются в большие числа, которые затем используются в математических преобразованиях.
3. Производится процедура за(рас)шифрования входящего сообщения (шифр текста), в результате которого получается шифр текст (исходящее сообщение) в виде большого числа, которое затем преобразуется в массив байт и записывается в файл.
4. Отдельно реализована возможность генерации ключевых значений p и q – в зависимости от задаваемой длины ключа генерируются большие простые числа. Затем осуществляется генерация и проверка числа e , а уже на основе p , q , e осуществляется вычисление числа d .
5. Список методов, реализованных в программе:
 - 5.1 **bool IsPrime(BigInteger n, int k)** – метод, осуществляющий проверку целого числа на простоту по алгоритму Миллера-Рабина с вероятностью $4^{(-k)}$
 - 5.2 **BigInteger GenBigInteger(int length)** – метод, который создает большое случайное целое число с заданной длиной бит
 - 5.3 **GenPrime(int length, int k)** – метод генерирует простое число, стартуя с большого целого числа заданной длины $length$ с вероятностью $4^{(-k)}$
 - 5.4 **int GetComMaxDiv(BigInteger a, int b)** – метод возвращает наибольший общий делитель для двух больших целых чисел a и b (алгоритм Евклида)
 - 5.5 **GenExpEnc(BigInteger input)** – метод «подбирает» значение числа e , чтобы оно было взаимно простым с большим целым числом $input$
 - 5.6 **byte[] ReverseBytes(byte[] bytes)** – метод осуществляет обратную перестановку массива байт (чтобы избежать путаницы при отладке)
 - 5.7 **BigInteger GetExpDec(BigInteger phi, int e)** – метод реализует расширенный алгоритм Евклида для поиска закрытого ключа d по значениям чисел e и $\varphi(n)$
 - 5.8 **byte[] Encrypt(byte[] input, BigInteger n, int e)** – метод осуществляет зашифрование массива байт с исходным сообщением используя алгоритм RSA
 - 5.9 **byte[] Decrypt(byte[] input, BigInteger n, BigInteger d)** – метод осуществляет расшифрование массива байт с шифр текстом используя алгоритм RSA

4. Результаты работы программы

4.1 Входные данные

1. Файл исходного сообщения – message.txt
2. Файлы ключей – n-key.key, e-key.key, d-key.key
3. Файл зашифрованного сообщения – cipher.dat

4.2 Выходные данные

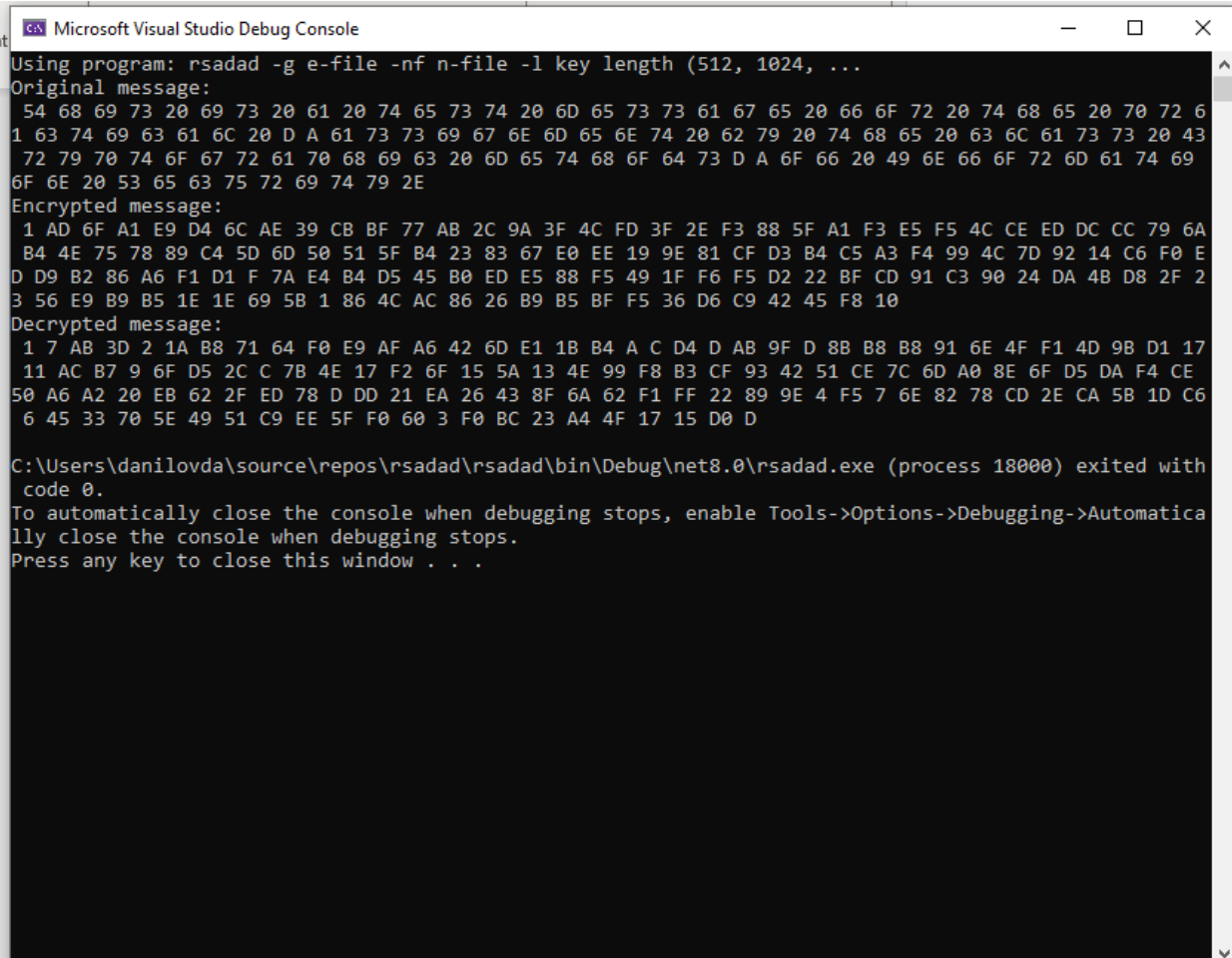
1. Файлы зашифрованного сообщения – cipher.dat,
2. Файл расшифрованного шифр текста – uncipher.txt

4.3 Использование программы

1. Генерация ключей – **rsadad.exe -g e-key.key -nf n-key.key -l 512**
2. Зашифрование – **rsadad.exe -e message.txt -ef e-key.key -nf n-key.key**
3. Расшифрование - **rsadad.exe -d cipher.dat -df d-key.key -nf n-key.key**

4.4 Результат работы программы(консольный вывод)

1. Работа с файлами:

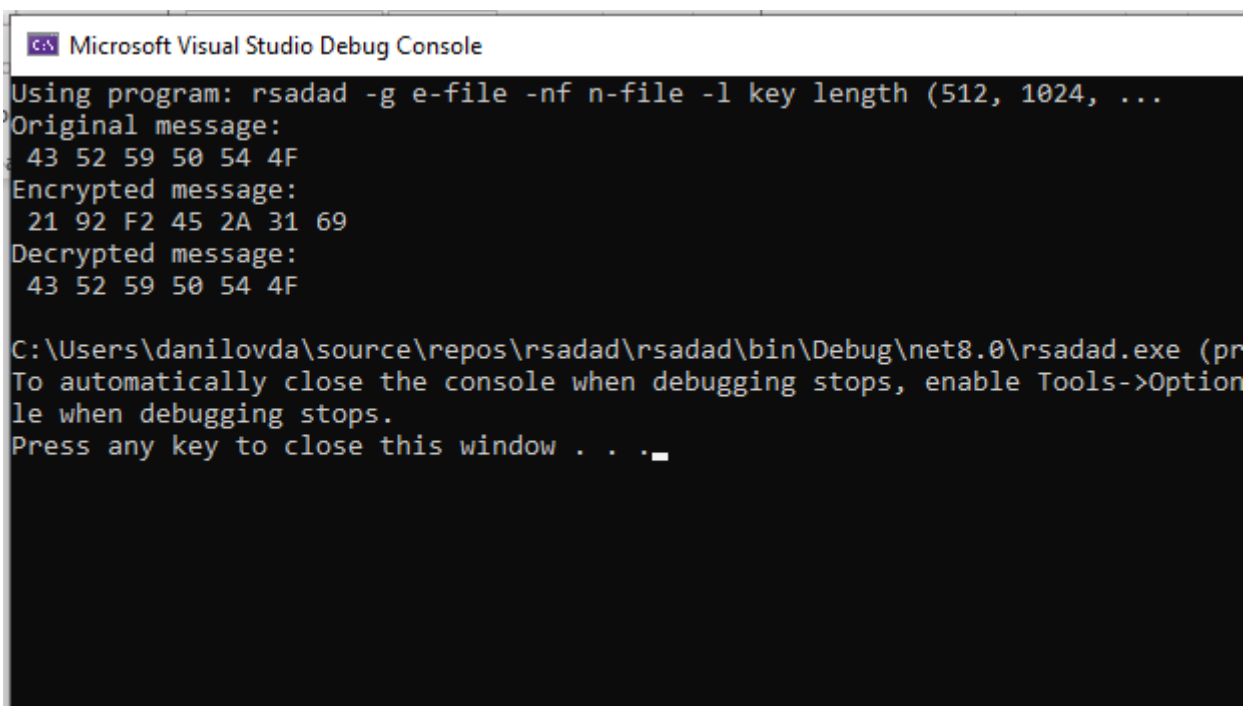


```
Microsoft Visual Studio Debug Console

Using program: rsadad -g e-file -nf n-file -l key length (512, 1024, ...
Original message:
 54 68 69 73 20 69 73 20 61 20 74 65 73 74 20 6D 65 73 73 61 67 65 20 66 6F 72 20 74 68 65 20 70 72 6
1 63 74 69 63 61 6C 20 D A 61 73 73 69 67 6E 6D 65 6E 74 20 62 79 20 74 68 65 20 63 6C 61 73 73 20 43
72 79 70 74 6F 67 72 61 70 68 69 63 20 6D 65 74 68 6F 64 73 D A 6F 66 20 49 6E 66 6F 72 6D 61 74 69
6F 6E 20 53 65 63 75 72 69 74 79 2E
Encrypted message:
 1 AD 6F A1 E9 D4 6C AE 39 CB BF 77 AB 2C 9A 3F 4C FD 3F 2E F3 88 5F A1 F3 E5 F5 4C CE ED DC CC 79 6A
B4 4E 75 78 89 C4 5D 6D 50 51 5F B4 23 83 67 E0 EE 19 9E 81 CF D3 B4 C5 A3 F4 99 4C 7D 92 14 C6 F0 E
D D9 B2 86 A6 F1 D1 F 7A E4 B4 D5 45 B0 ED E5 88 F5 49 1F F6 F5 D2 22 BF CD 91 C3 90 24 DA 4B D8 2F 2
3 56 E9 B9 B5 1E 1E 69 5B 1 86 4C AC 86 26 B9 B5 BF F5 36 D6 C9 42 45 F8 10
Decrypted message:
 1 7 AB 3D 2 1A B8 71 64 F0 E9 AF A6 42 6D E1 1B B4 A C D4 D AB 9F D 8B B8 B8 91 6E 4F F1 4D 9B D1 17
11 AC B7 9 6F D5 2C C 7B 4E 17 F2 6F 15 5A 13 4E 99 F8 B3 CF 93 42 51 CE 7C 6D A0 8E 6F D5 DA F4 CE
50 A6 A2 20 EB 62 2F ED 78 D DD 21 EA 26 43 8F 6A 62 F1 FF 22 89 9E 4 F5 7 6E 82 78 CD 2E CA 5B 1D C6
6 45 33 70 5E 49 51 C9 EE 5F F0 60 3 F0 BC 23 A4 4F 17 15 D0 D

C:\Users\danilovda\source\repos\rsadad\rsadad\bin\Debug\net8.0\rsadad.exe (process 18000) exited with
code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatica
lly close the console when debugging stops.
Press any key to close this window . . .
```

2. Работа с малыми числами:



```
Microsoft Visual Studio Debug Console
Using program: rsadad -g e-file -nf n-file -l key length (512, 1024, ...
Original message:
43 52 59 50 54 4F
Encrypted message:
21 92 F2 45 2A 31 69
Decrypted message:
43 52 59 50 54 4F

C:\Users\danilovda\source\repos\rsadad\rsadad\bin\Debug\net8.0\rsadad.exe (pr
To automatically close the console when debugging stops, enable Tools->Option
le when debugging stops.
Press any key to close this window . . .
```

5 Выводы о проделанной работе

Был изучен и реализован алгоритм криптосистемы с открытым ключом RSA в виде консольного приложения, запускаемого с параметрами. Программа осуществляет генерацию ключей шифрования произвольной длины. Программа производит зашифрование и расшифрование файлов с использованием сгенерированных файлов ключей или с заданными параметрами. Наибольшую сложность представляет собой работа с блоками неизвестной длины (длина вычисляется в процессе работы алгоритма исходя из вводных данных). Если размер блока «зафиксировать», тогда можно значительно упростить и ускорить алгоритм. Сгенерированный ключ остается неизменным, никаких дополнительных преобразований алгоритм не осуществляет, поэтому передача больших сообщений недопустима, алгоритм не обеспечивает защиту от частотного анализа данных. Основное использование алгоритма – сеансовая передачи ключей для последующего обмена сообщения с использованием синхронного шифрования.

6 Список использованных источников

1. Википедия <https://ru.wikipedia.org/wiki/RSA>