Министерство образования Республики Беларусь

Учреждение образования БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы защиты информации

Отчёт по лабораторной работе №3

По теме «Асимметричная криптография. Криптосистема Рабина»

Выполнили: студент гр. 053501 Кривецкий Р. А.

Проверил: ассистент кафедры информатики Лещенко Е. А.

Минск 2023

**СОДЕРЖАНИЕ**

[1 Теоретические сведения 3](#_Toc148061125)

[2 Программная реализация 4](#_Toc148061126)

[Заключение 5](#_Toc148061127)

[Приложение А(обязательное) Листинг программного кода 6](#_Toc148061128)

# 1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Одним из классических примеров асимметричной криптосистемы является криптосистема Рабина, разработанная Рональдом Л. Рабином в 1979 году. Криптосистема Рабина отличается от симметричных алгоритмов шифрования тем, что в ней используются два различных ключа: открытый и закрытый. Открытый ключ используется для шифрования информации, в то время как закрытый ключ используется для расшифровки.  
 Данный алгоритм показан на рисунке 1.1.

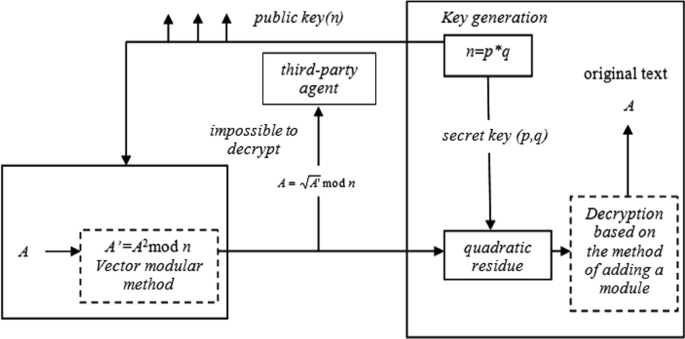


Рисунок 1.1 — Блок-схема алгоритма

**2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

Необходимо реализовать криптостойкое программное средство шифрования и дешифрования текстовых файлов при помощи Криптосистемы Рабина. Программное средство реализовано на языке программирования Python. Демонстрация работы показана на рисунке 2.1

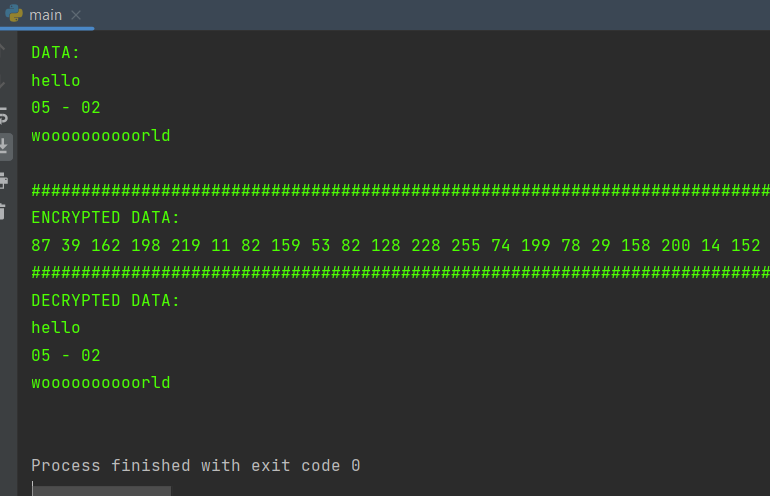


Рисунок 2.1 — Демонстрация работы

Вывод в консоль сделан для дополнительной наглядности, корректная работа с файлами показана на рисунке 2.2.

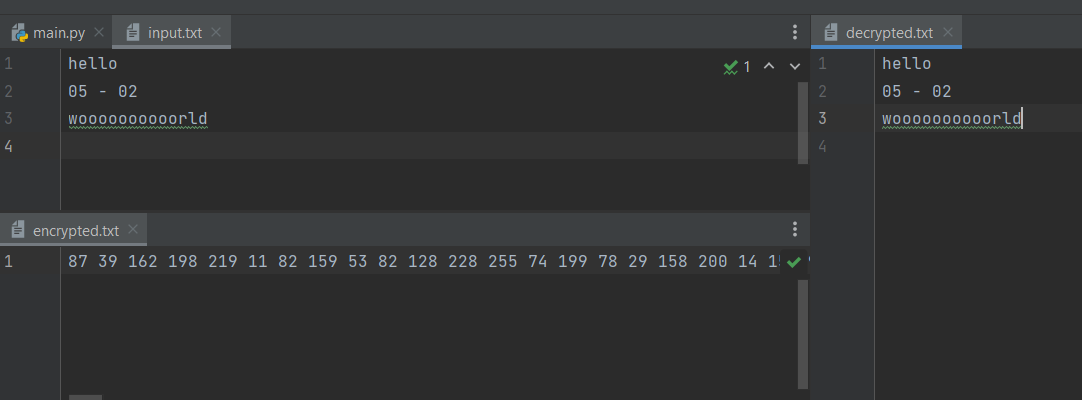


Рисунок 2.2 — Ввод/вывод файлов

# ЗАКЛЮЧЕНИЕ

Криптосистема Рабина - это криптографический алгоритм, основанный на трудности факторизации больших чисел. Он обеспечивает теоретическую безопасность, но является менее эффективным и практичным по сравнению с более современными алгоритмами, такими как RSA и AES. Рабинова криптосистема чаще используется в исследовательских целях, чем в реальных приложениях.

# ПРИЛОЖЕНИЕ А

(**обязательное)**

**Листинг программного кода**

Файл main.py

import random

def modulo(a, b):

if a >= 0:

return a % b

else:

return (b - abs(a % b)) % b

def generate\_key(bits):

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n \*\* 0.5) + 1):

if n % i == 0:

return False

return True

def generate\_prime(bits):

while True:

num = random.getrandbits(bits)

if num % 4 == 3 and is\_prime(num):

return num

p = generate\_prime(bits)

q = generate\_prime(bits)

open\_key = p \* q

close\_key = (p, q)

return open\_key, close\_key

def number\_to\_text(numbers):

result = []

for item in numbers:

for i in range(0, len(item), 4):

number = item[i:i + 4]

while number[0] == '0':

number = number[1:]

result.append(chr(int(number)))

return result

def extended\_gcd(a, b):

if a == 0:

return (0, 1)

for item in (r1, r2, r3, r4):

if item <= 1200:

return chr(item)

def main():

with open("input.txt", "r", encoding='utf-8') as f:

text = f.read()

with open("decrypted.txt", "w", encoding='utf-8') as f:

pass

with open("encrypted.txt", "a", encoding='utf-8') as f:

pass

open\_key, close\_key = generate\_key(16)

print(close\_key)

while close\_key[0] == close\_key[1]:

open\_key, close\_key = generate\_key(42)

for item in text:

encrypted\_text = encrypted(item, open\_key)

with open("encrypted.txt", "a", encoding='utf-8') as f:

f.write(str(encrypted\_text))

decrypted\_text = decrypted(encrypted\_text, open\_key, close\_key)

with open("decrypted.txt", "a", encoding='utf-8') as f:

f.write(decrypted\_text)

if \_\_name\_\_ == "\_\_main\_\_":

try:

main()

except:

print(Exception())

with open("input.txt") as f:

print(f"DATA:\n{f.read()}")

print("#" \* 200)

with open("encrypted.txt") as f:

print(f"ENCRYPTED DATA:\n{f.read()}")

print("#" \* 200)

with open("decrypted.txt") as f:

print(f"DECRYPTED DATA:\n{f.read()}")