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 №2

По теме «Симметричная криптография. СТБ 34.101.31-2011»

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

СТБ 34.101.31-2011 – государственный стандарт симметричного шифрования и контроля целостности Республики Беларусь. Криптографические алгоритмы шифрования и контроля целостности. Принят в качестве предварительного стандарта в 2007 году. Введён в действие в качестве окончательного стандарта в 2011 году.

Данный алгоритм в режиме сцепления блоков описан на рисунке 1.1.

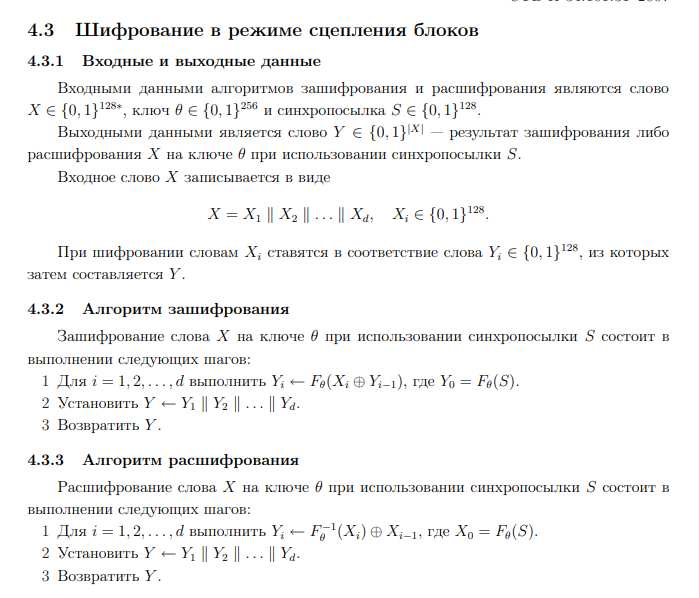


Рисунок 1.1 — Режим сцепления блоков

**2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

Необходимо реализовать программные средства шифрования и дешифрования текстовых файлов при помощи алгоритма СТБ 34.101.31-20 в режиме сцепления блоков. Программное средство реализовано на языке программирования Python. Демонстрация работы показана на рисунке 2.1

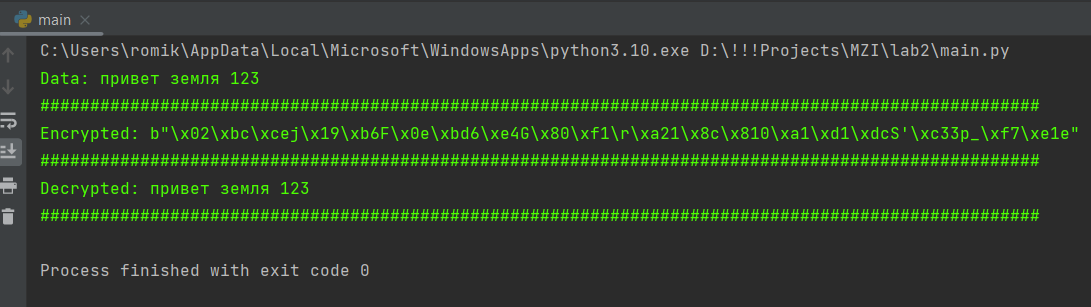


Рисунок 2.1 — Демонстрация работы

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

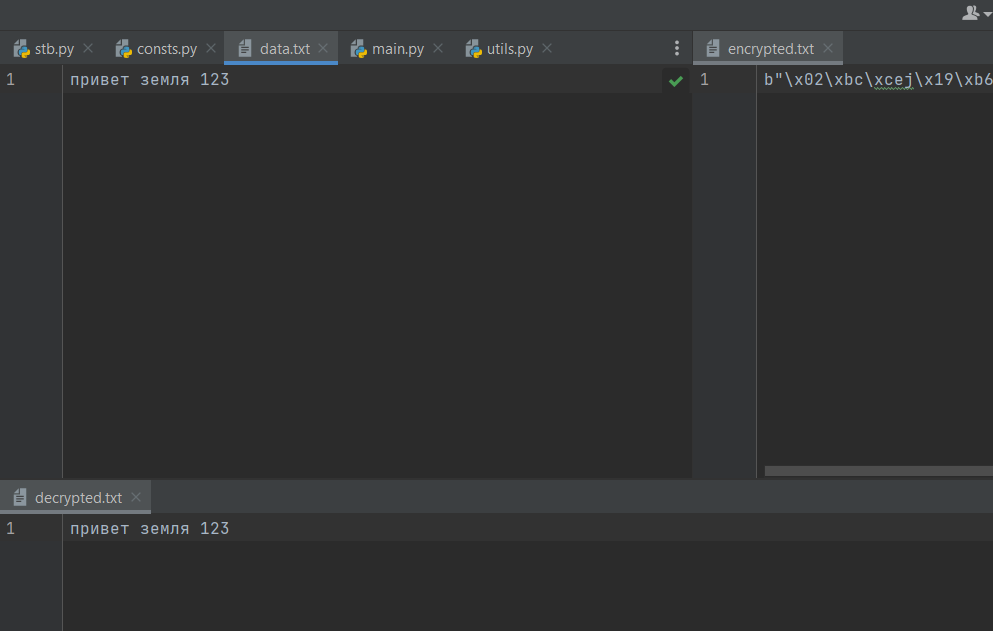


Рисунок 2.2 — Ввод/вывод файлов

# ЗАКЛЮЧЕНИЕ

Данная группа алгоритмов предназначены для обеспечения конфиденциальности. Она включает алгоритм зашифрования и алгоритм расшифрования на секретном ключе. Алгоритм зашифрования преобразует открытые данные в защищенные, а алгоритм расшифрования выполняет обратное преобразование. Стороны, располагающие общим секретным ключом, могут организовать защищенный обмен данными путем их зашифрования перед отправкой и расшифрования после получения.

# ПРИЛОЖЕНИЕ А

(**обязательное)**

**Листинг программного кода**

Файл main.py

from consts import \*

from stb import STB

def main():

data = read\_data()

print(f"Data: {data}")

print("#"\*100)

key = int.from\_bytes('erpwkwekrokpdasdasda'.encode(), 'big')

synchro = 312312412412

stb = STB(key)

# encrypt

encrypted\_text = stb.encrypt(data, synchro)

write\_encrypted\_data(encrypted\_text)

print("#"\*100)

# decrypt

decrypted\_text = stb.decrypt(encrypted\_text, synchro)

write\_decrypted\_data(decrypted\_text)

print("#"\*100)

def read\_data():

file = open("data.txt", "r", encoding="utf8")

data = file.read()

file.close()

return data

def write\_encrypted\_data(encrypted\_text: bytes):

file = open("encrypted.txt", "w+", encoding="utf8")

file.write(str(encrypted\_text))

print(f"Encrypted: {encrypted\_text}")

file.close()

def write\_decrypted\_data(decrypted\_text: str):

file = open("decrypted.txt", "w+", encoding="utf8")

file.write(decrypted\_text)

print(f"Decrypted: {decrypted\_text}")

file.close()

if \_\_name\_\_ == '\_\_main\_\_':

main()