Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

**Лабораторная работа № 7**

**по дисциплине**

**«Администрирование и безопасность программно-информационных систем.»»**

**на тему: «Шифрование с помощью аналитических преобразований»**

Выполнил:

ст. гр. ПРИ-122

А.А. Ерофеев

Принял:

доц. каф. ИСПИ

С.В. Курочкин

Владимир, 2025

ЦЕЛЬ РАБОТЫ

Освоить на практике применение алгоритма шифрования с помощью аналитических преобразований.

ВЫПОЛНЕНИЕ РАБОТЫ (4 вариант)

Листинг кода программной реализации алгоритма:

import numpy as np  
from typing import List  
  
  
A = np.array([  
 [ 5, 12, -4, 8],  
 [ 3, 0, 7, 21],  
 [ 5, 2, 1, 1],  
 [ 5, 3, 2, 9],  
], dtype=np.int64)  
  
  
def \_to\_blocks(data: bytes, n: int) -> np.ndarray:  
 pad = (-len(data)) % n  
 full = data + b'\x00' \* pad  
 return np.frombuffer(full, dtype=np.uint8).reshape(-1, n).astype(np.int64)  
  
def \_from\_blocks(blocks: np.ndarray, orig\_len: int) -> bytes:  
 flat = blocks.astype(np.uint8).ravel()[:orig\_len]  
 return bytes(flat)  
  
  
def encrypt(plaintext: str, key: np.ndarray = A) -> List[int]:  
 data = plaintext.encode('utf-8')  
 blocks = \_to\_blocks(data, key.shape[0])  
 cipher = (blocks @ key.T).ravel().tolist()  
 return cipher  
  
def decrypt(cipher: List[int], orig\_len: int, key: np.ndarray = A) -> str:  
 n = key.shape[0]  
 C = np.array(cipher, dtype=np.float64).reshape(-1, n)  
 B = (C @ np.linalg.inv(key).T).round().astype(np.int64)  
 data = \_from\_blocks(B, orig\_len)  
 return data.decode('utf-8')  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 text = (  
 'Помехоустойчивое кодирование — это кодирование с возможностью '  
 'восстановления потерянных или ошибочно принятых данных.'  
 )  
  
 cipher = encrypt(text)  
 recovered = decrypt(cipher, len(text.encode('utf-8')))  
  
 print('Первые 20 чисел шифра:', cipher[:20], '...')  
 print('\nРасшифровка:\n', recovered)  
 assert recovered == text

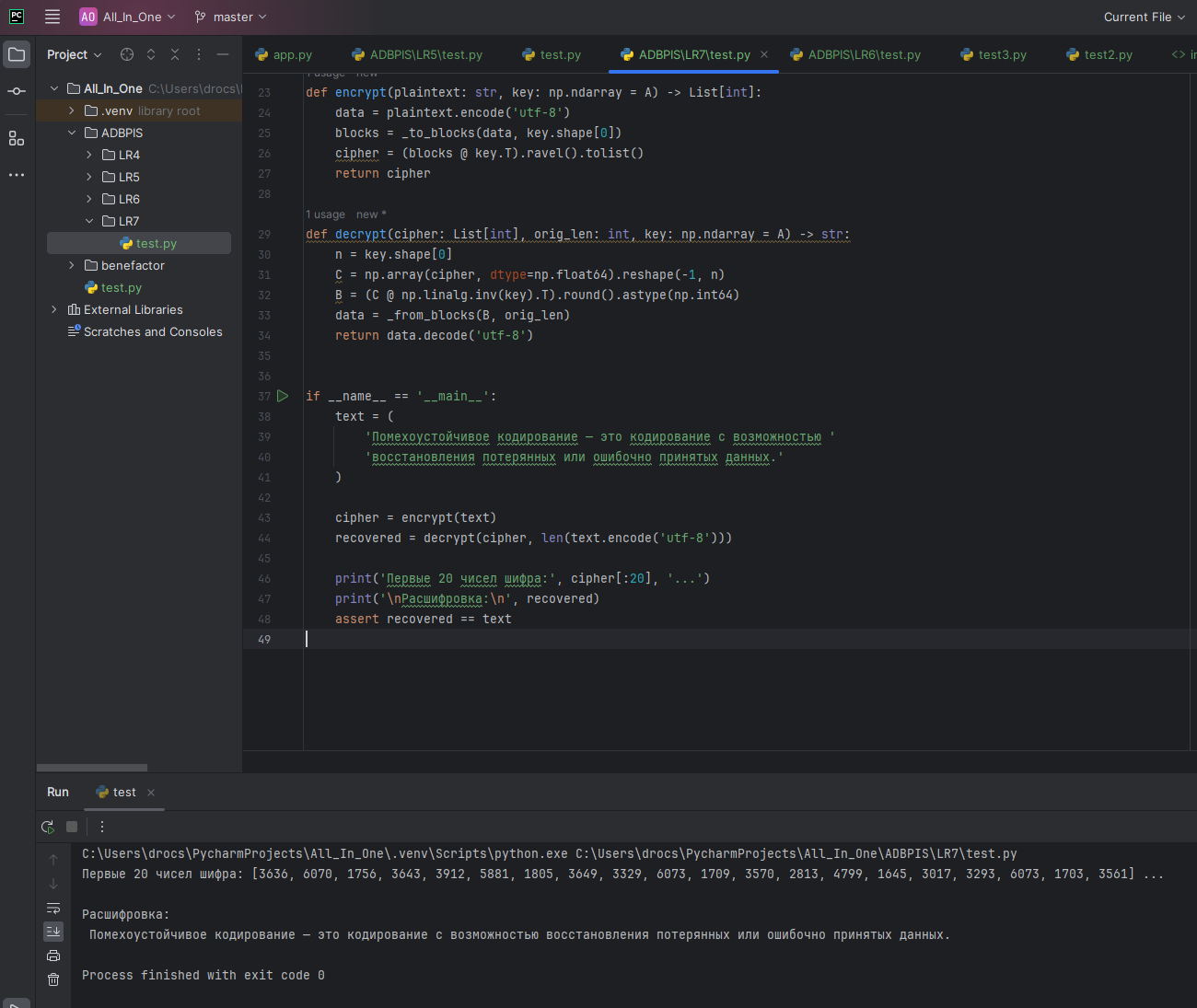


Рисунок 1. Демонстрация работы.

ВЫВОД

В ходе лабораторной работы освоил на практике применение алгоритма шифрования с помощью аналитических преобразований.

.