Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

ОТЧЁТ

по лабораторной работе №2

по дисциплине

Средства и методы защиты информации в интеллектуальных системах

|  |  |
| --- | --- |
|  | Выполнил:  Гопка А.А. гр. 221703 |
|  | Проверил:  Крищенович В.А. Н.В |

Минск 2024

Вариант 4: Шифр перестановки, использующий простые (прямоугольные) таблицы.

Задание:

Реализовать в виде программы шифр (зашифрование и расшифрование) в соответствии с вариантом. Язык исходного текста русский.

import numpy as np

# Функция для шифрования текста с использованием перестановки

def encrypt\_permutation\_cypher(plain\_text, key):

# Преобразуем текст в матрицу согласно ключу

num\_rows = len(plain\_text) // key + (len(plain\_text) % key != 0)

matrix = np.full((num\_rows, key), ' ')

index = 0

for i in range(num\_rows):

for j in range(key):

if index < len(plain\_text):

matrix[i][j] = plain\_text[index]

index += 1

# Шифруем текст по столбцам

cypher\_text = ''

for j in range(key):

for i in range(num\_rows):

cypher\_text += matrix[i][j]

return cypher\_text

# Функция для дешифрования текста с использованием перестановки

def decrypt\_permutation\_cypher(cypher\_text, key):

num\_rows = len(cypher\_text) // key

matrix = np.full((num\_rows, key), ' ')

index = 0

for j in range(key):

for i in range(num\_rows):

if index < len(cypher\_text):

matrix[i][j] = cypher\_text[index]

index += 1

# Восстанавливаем исходный текст по строкам

plain\_text = ''

for i in range(num\_rows):

for j in range(key):

plain\_text += matrix[i][j]

return plain\_text.strip()

# Функция атаки полным перебором

def brute\_force\_attack(cypher\_text, original\_text):

for key in range(2, len(cypher\_text)):

decrypted\_text = decrypt\_permutation\_cypher(cypher\_text, key)

if decrypted\_text == original\_text:

print(f'Найден ключ: {key}')

return decrypted\_text

return None

# Пример использования программы

plain\_text = "Какой хороший день!"

key = 4

cypher\_text = encrypt\_permutation\_cypher(plain\_text, key)

print(f'Зашифрованный текст: {cypher\_text}')

decrypted\_text = decrypt\_permutation\_cypher(cypher\_text, key)

print(f'Дешифрованный текст: {decrypted\_text}')

# Атака полным перебором

print('Атака полным перебором:')

brute\_force\_attack(cypher\_text, plain\_text)

Зашифрованный текст: Кйрйна о ькхшд!ооие

Дешифрованный текст: Какой хороший день!

Атака полным перебором:

Найден ключ: 4

'Какой хороший день!'

Вариант усложнения шифра: Множественная перестановка по нескольким ключам.

Вместо одного ключа используем два или более ключей, которые будут использоваться для перестановки строк и столбцов.

def encrypt\_double\_permutation(plain\_text, key1, key2):

# Преобразуем текст в матрицу согласно ключу1 (по количеству столбцов)

num\_rows = len(plain\_text) // key1 + (len(plain\_text) % key1 != 0)

matrix = np.full((num\_rows, key1), ' ')

index = 0

for i in range(num\_rows):

for j in range(key1):

if index < len(plain\_text):

matrix[i][j] = plain\_text[index]

index += 1

# Перестановка строк по ключу2

matrix = matrix[key2]

# Зашифруем текст по столбцам

cipher\_text = ''

for j in range(key1):

for i in range(num\_rows):

cipher\_text += matrix[i][j]

return cipher\_text

# Функция для дешифрования с использованием множественной перестановки

def decrypt\_double\_permutation(cipher\_text, key1, key2):

num\_rows = len(cipher\_text) // key1

matrix = np.full((num\_rows, key1), ' ')

index = 0

for j in range(key1):

for i in range(num\_rows):

if index < len(cipher\_text):

matrix[i][j] = cipher\_text[index]

index += 1

# Восстановление исходного порядка строк по ключу2

reversed\_key2 = np.argsort(key2)

matrix = matrix[reversed\_key2]

# Восстанавливаем исходный текст по строкам

plain\_text = ''

for i in range(num\_rows):

for j in range(key1):

plain\_text += matrix[i][j]

return plain\_text.strip()

# Пример использования программы

plain\_text = "Какой хороший день!"

key1 = 5 # Количество столбцов

key2 = np.array([1, 0, 2, 3]) # Ключ перестановки строк

cipher\_text = encrypt\_double\_permutation(plain\_text, key1, key2)

print(f'Зашифрованный текст: {cipher\_text}')

decrypted\_text = decrypt\_double\_permutation(cipher\_text, key1, key2)

print(f'Дешифрованный текст: {decrypted\_text}')

Зашифрованный текст: Кшехаинокйьро !ойд

Дешифрованный текст: Какой хороший день!

Предложить вариант усложнения шифра:

* поменять шифр на более надежный;
* комбинирование шифров.
* Использовать усовершенствованный шифр перестановки - множественная перестановка по нескольким ключам.