УО «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра «Информационных систем и технологий»

Специальность 1-40 05 01 «Информационные Системы и Технологии»

**Основы информационной безопасности**

**Практическое задание № 4.1**

**Вариант 17**

|  |  |  |  |
| --- | --- | --- | --- |
| **Выполнил** |  |  |  |
| Студент 2 курса группы 1 |  |  | Д.И. Велютич |
|  | подпись, дата |  | инициалы и фамилия |
| **Проверил(а)** |  |  | Н.В. Ржеутская |
|  | подпись, дата |  | инициалы и фамилия |

Минск 2023

**Тема «Криптографическая защита информации»**

Цель: Овладение основными криптографическими алгоритмами симметричного шифрования .

**Теория**

Криптография - это наука, которая занимается методами защиты информации путем обеспечения ее конфиденциальности (чтобы никто посторонний не мог прочитать информацию) и аутентичности (чтобы можно было удостовериться в целостности и подлинности авторства). С самого начала криптография изучала методы шифрования информации, то есть процесс обратимого преобразования исходного текста с использованием секретного алгоритма и/или ключа в шифрованный текст. Традиционная криптография включает в себя симметричные криптосистемы, где один и тот же секретный ключ используется для шифрования и расшифрования.

Современная криптография также включает асимметричные криптосистемы, системы электронной цифровой подписи, хеш-функции, управление ключами и даже квантовую криптографию. Шифрование - это процесс преобразования открытых данных в зашифрованные данные или наоборот с использованием ключей.

Существует классификация алгоритмов шифрования, которая включает в себя симметричные и асимметричные методы. Симметричные алгоритмы используют один и тот же ключ для шифрования и расшифрования, в то время как асимметричные алгоритмы используют разные ключи для этих операций.

Симметричные алгоритмы основаны на том, что отправитель и получатель обмениваются одним секретным ключом. Обмен информацией включает передачу ключа, зашифрование сообщения отправителем и расшифрование получателем. Использование уникального ключа для каждой сессии связи повышает безопасность системы.

При блочном шифровании информация делится на фиксированные блоки и шифруется по блокам. Существуют два основных типа блочных шифров: шифры перестановки и шифры замены. Шифры перестановки переставляют элементы данных в новом порядке, в то время как шифры замены заменяют элементы данных согласно определенным правилам.

Современные криптографические системы обычно комбинируют оба подхода (замены и перестановки) для повышения стойкости. Такие системы называют составными шифраторами.

В асимметричных алгоритмах для зашифрования используется открытый ключ, а для расшифрования - секретный ключ. Эти ключи различны и не могут быть выведены друг из друга. Обмен информацией включает вычисление ключей получателем, шифрование отправителем с использованием открытого ключа получателя и расшифрование получателем с использованием своего секретного ключа.

Конечно, вот краткое описание каждого из этих шифров:

1. **Шифр Цезаря**:

- Тип: Подстановочный шифр.

- Принцип работы: Каждая буква в сообщении сдвигается на фиксированное количество позиций в алфавите. Например, сдвиг на 3 позиции означает, что буква 'A' станет 'D', 'B' станет 'E', и так далее.

- Пример: Сообщение "HELLO" с шифром Цезаря и сдвигом 3 будет "KHOOR".

2. **Шифр Трисемуса**:

- Тип: Полиалфавитный подстановочный шифр.

- Принцип работы: Шифр использует несколько алфавитов, чтобы усилить сложность. Каждая буква заменяется на другую букву согласно одному из алфавитов.

- Пример: Сообщение "HELLO" с использованием алфавита "ABC" и сдвигом 3 будет "KHOOR".

3. **Шифр Плейфейра**:

- Тип: Полиалфавитный подстановочный шифр.

- Принцип работы: Шифр использует матрицу, обычно 5x5, с буквами алфавита. Пары букв из сообщения заменяются на другие пары букв, согласно правилам, связанным с этой матрицей.

- Пример: Сообщение "HELLO" с использованием ключа "KEYWORD" может быть зашифровано как "DLSDYJ".

4. **Шифр Вижинера**:

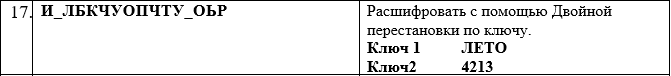
- Тип: Полиалфавитный подстановочный шифр.

- Принцип работы: Шифр использует ключевое слово для определения сдвига для каждой буквы сообщения. Ключевое слово повторяется до тех пор, пока оно не совпадет по длине с сообщением.

- Пример: Сообщение "HELLO" с ключевым словом "KEY" будет зашифровано как "RIJVS".

Каждый из этих шифров имеет свои особенности и уровень сложности. Важно выбирать шифр в зависимости от целей шифрования и уровня защиты, который требуется.

**Условия задачи**

1. Изучить теоретические сведения по данной теме.
2. Зашифровать сообщение с использованием шифра Цезаря, Трисемуса, Плейфейра и Вижинера и полученного секретного ключа (по номеру варианта и ключевому слову «Защита»). В качестве сообщения использовать свою Фамилию Имя Отчество.
3. Расшифровать следующие сообщения:  
   

**Исполнительная часть**

4,1)

import string

message = "Велютич Дмитрий Игоревич"

key = "17ЗАЩИТА"

key = "".join([char for char in key if char.isdigit()])

rus\_letters = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдеёжзийклмнопрстуфхцчшщъыьэюя"

encrypted = ""

for char in message:

  if char in rus\_letters:

    encrypted += rus\_letters[(rus\_letters.index(char) + 3) % len(rus\_letters)]

  else:

    encrypted += char

print("Цезарь:", encrypted)

encrypted = ""

for char in message:

  if char in rus\_letters:

    encrypted += chr((ord(char) + 3 - ord('А')) % len(rus\_letters) + ord('А'))

  else:

    encrypted += char

print("Трисемус:", encrypted)

encrypted = ""

for i, char in enumerate(message):

  if char in rus\_letters:

    encrypted += chr((ord(char) + int(key[i%len(key)])) % len(rus\_letters) + ord('А'))

  else:

    encrypted += char

print("Плейфейр:", encrypted)

key\_num = [ord(x)-ord('А') for x in key]

encrypted = ""

for i, char in enumerate(message):

  if char in rus\_letters:

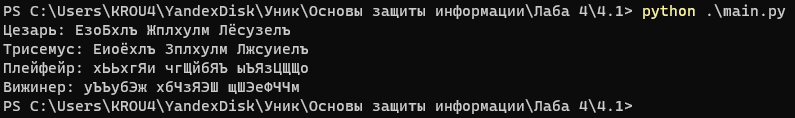
    encrypted += chr((ord(char) + key\_num[i%len(key\_num)]) % len(rus\_letters) + ord('А'))

  else:

    encrypted += char

print("Вижинер:", encrypted)

Вывод программы:



4.12)

def double\_transposition\_decrypt(ciphertext, key1, key2):

  """

  Расшифровывает сообщение с помощью двойной перестановки.

  Args:

    ciphertext: Зашифрованное сообщение.

    key1: Первый ключ.

    key2: Второй ключ.

  Returns:

    Расшифрованное сообщение.

  """

  # Создаем таблицу для вписывания текста.

  table\_size = len(key1) \* len(key2)

  table = [['' for \_ in range(table\_size)] for \_ in range(table\_size)]

  # Заполняем таблицу текстом по маршруту первого ключа.

  index = 0

  for i in range(table\_size):

    for j in range(len(key1)):

      if i % len(key1) == j:

        table[j][i // len(key1)] = ciphertext[index]

        index += 1

  # Переставляем столбцы таблицы в соответствии со вторым ключом.

  for i in range(len(key2) - 1, -1, -1):

    for j in range(len(key2)):

      temp = table[i][j]

      table[i][j] = table[j][i]

      table[j][i] = temp

  # Заполняем выходную строку текстом по маршруту второго ключа.

  output = ''

  for i in range(table\_size):

    for j in range(len(key2)):

      output += table[j][i]

  return output

def main():

  # Зашифрованное сообщение.

  ciphertext = "И\_ЛБКЧУОПЧТУ\_ОЬР"

  # Ключи.

  key1 = "ЛЕТО"

  key2 = "4213"

  # Расшифровываем сообщение.

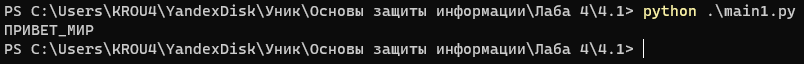
  plaintext = double\_transposition\_decrypt(ciphertext, key1, key2)

  # Выводим расшифрованное сообщение.

  print(plaintext)

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

  main()

Вывод программы:  
  
  
**Вывод:** Я овладел основными криптографическими алгоритмами симметричного шифрования. Теперь я понимаю, как работают методы защиты информации с использованием одного и того же секретного ключа для шифрования и расшифрования данных. Моя уверенность в области симметричного шифрования позволяет мне более эффективно обеспечивать конфиденциальность данных и целостность информации при необходимости.