

Лабораторная работа №3. Шифрование гаммированием.

Alexander S. Baklashov

07 October, 2023

RUDN University, Moscow, Russian Federation

Цель работы

Рассмотреть и реализовать алгоритм шифрования гаммированием конечной гаммой.

Задачи

Реализовать шифрование гаммированием конечной гаммой.

Шифрование гаммированием конечной гаммой. Функция для шифрования текста.

Напишем функцию для шифрования текста

```
In [1]: def encrypt(text, key):  
    # Создаем словарь, который соотносит буквы с их номерами в алфавите  
    alphabet = "абвгдезийклмнопрстуфхцщъыьэя"  
    letter_to_number = {letter: index for index, letter in enumerate(alphabet, start=1)}  
  
    encrypted_text = ""  
  
    for i in range(len(text)):  
        # Получаем номер буквы в тексте и ключе  
        text_letter = text[i]  
        key_letter = key[i % len(key)]  
  
        text_number = letter_to_number.get(text_letter, 0)  
        key_number = letter_to_number.get(key_letter, 0)  
  
        # Вычисляем зашифрованную букву  
        encrypted_number = (text_number + key_number) % 33  
  
        # Получаем зашифрованную букву из словаря  
        encrypted_letter = list(letter_to_number.keys())[list(letter_to_number.values()).index(encrypted_number)]  
  
        encrypted_text += encrypted_letter  
  
    return encrypted_text
```

Figure 1: Шифрование текста

Шифрование гаммированием конечной гаммой. Функция для дешифрования текста.

Напишем функцию для дешифрования текста

```
In [2]: def decrypt(encrypted_text, key):  
    # Также создаем словарь, который соотносит буквы с их номерами в алфавите  
    alphabet = "абвгдезийклмнопрстуфхцщъыьэяя"  
    letter_to_number = {letter: index for index, letter in enumerate(alphabet, start=1)}  
  
    decrypted_text = ""  
  
    for i in range(len(encrypted_text)):  
        # Получаем номер зашифрованной буквы и ключа  
        encrypted_letter = encrypted_text[i]  
        key_letter = key[i % len(key)]  
  
        encrypted_number = letter_to_number.get(encrypted_letter, 0)  
        key_number = letter_to_number.get(key_letter, 0)  
  
        # Вычисляем исходную букву  
        text_number = (encrypted_number - key_number) % 33  
  
        # Получаем исходную букву из словаря  
        text_letter = list(letter_to_number.keys())[list(letter_to_number.values()).index(text_number)]  
  
        decrypted_text += text_letter  
  
    return decrypted_text
```

Figure 2: Дешифрование текста

Шифрование гаммированием конечной гаммой. Реализация.

Напишем реализацию шифрования гаммированием конечной гаммой с помощью функций

```
In [3]: text = "приказ"
        key = "гамма"

        # Шифрование
        encrypted_text = encrypt(text, key)
        print("Зашифрованный текст:", encrypted_text)

        # Дешифрование
        decrypted_text = decrypt(encrypted_text, key)
        print("Расшифрованный текст:", decrypted_text)

Зашифрованный текст: усхчбл
Расшифрованный текст: приказ
```

Figure 3: Реализация

Вывод

В ходе данной лабораторной работы я рассмотрел и реализовал алгоритм шифрования гаммированием конечной гаммой.