

Отчет по лабораторной работе №7

Дисциплина: Информационная безопасность

Дорофеева Алёна Тимофеевна

Содержание

1	Цель работы	5
2	Задачи	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	11
	Список литературы	12

Список иллюстраций

4.1 Название рисунка 10

Список таблиц

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Задачи

1. Изучить принцип одноразового гаммирования.
2. Разработать приложение, позволяющее шифровать и дешифровать данные в режиме одноразового гаммирования.

3 Теоретическое введение

Шифрование гаммированием – это метод шифрования, который основан на использовании гаммы [key-1?].

Гамма шифра – это псевдослучайная последовательность, выработанная по определенному алгоритму для шифрования открытых данных и дешифрования зашифрованных данных. Она играет роль ключа в одноразовой система шифрования. Строго говоря, она не удовлетворяет ни требованию случайности, так как используется детерминированный алгоритм для ее выработки, ни требованию бесконечной длины, так как все псевдослучайные последовательности имеют конечный период. Тем не менее, при правильно выбранном алгоритме генерации гаммы шифра можно получить метод шифрования с хорошей практической стойкостью, достаточной для решения реальных задач защиты информации [key-2?].

Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста[key-3?].

4 Выполнение лабораторной работы

1. Разработаем приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования (код выполнен на языке программирования Python):

```
def main():  
    # Создаем алфавит  
    alphabet = {  
        "а": 1, "б": 2, "в": 3, "г": 4, "д": 5, "е": 6, "ё": 7, "ж": 8, "з": 9, "  
        "й": 11, "к": 12, "л": 13, "м": 14, "н": 15, "о": 16, "п": 17, "р": 18, "  
        "у": 21, "ф": 22, "х": 23, "ц": 24, "ч": 25, "ш": 26, "щ": 27, "ъ": 28, "  
        "э": 31, "ю": 32, "я": 33  
    }  
  
    # Меняем местами ключи и значения для дешифрования  
    alphabet_reversed = {v: k for k, v in alphabet.items()}  
  
    gamma = input("Введите гамму (на русском языке без пробелов): ").lower()  
    text = input("Введите текст для шифрования: ").lower()  
  
    # Функция для шифрования текста  
    def encrypt(text, gamma):  
        result = ""  
        gamma_index = 0
```



```

for char in text:
    if char in alphabet:
        char_index = alphabet[char]
        gamma_char = gamma[gamma_index]
        gamma_index = (gamma_index + 1) % len(gamma)
        gamma_index = gamma_index if gamma_char in alphabet else 0
        gamma_index = 0 if gamma_char not in alphabet else gamma_index
        char_index = (char_index + alphabet[gamma_char]) % 33
        result += alphabet_reversed[char_index]
    else:
        result += char # Если символ не из алфавита, оставляем без измен

return result

```

Функция для дешифрования текста

```

def decrypt(text, gamma):
    result = ""
    gamma_index = 0

    for char in text:
        if char in alphabet:
            char_index = alphabet[char]
            gamma_char = gamma[gamma_index]
            gamma_index = (gamma_index + 1) % len(gamma)
            gamma_index = gamma_index if gamma_char in alphabet else 0
            gamma_index = 0 if gamma_char not in alphabet else gamma_index
            char_index = (char_index - alphabet[gamma_char]) % 33
            result += alphabet_reversed[char_index]

```

```

        else:
            result += char # Если символ не из алфавита, оставляем без измен

    return result

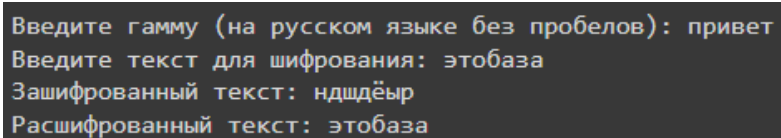
encrypted_text = encrypt(text, gamma)
decrypted_text = decrypt(encrypted_text, gamma)

print("Зашифрованный текст:", encrypted_text)
print("Расшифрованный текст:", decrypted_text)

if __name__ == "__main__":
    main()

```

В качестве примера зашифруем текст “этобаза” (рис. 4.1). Затем полученный результат пробуем расшифровать тем же ключом (рис. 4.1). Видим, что все успешно расшифровалось (рис. 4.1). Затем определяем ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста (рис. 4.1). Для этого используем функцию дешифровки, передавая ей исходный текст и то, что получилось в зашифрованном варианте (рис. 4.1).



```

Введите гамму (на русском языке без пробелов): привет
Введите текст для шифрования: этобаза
Зашифрованный текст: ндшдёыр
Расшифрованный текст: этобаза

```

Рис. 4.1: Название рисунка

5 Выводы

Освоила на практике применение режима однократного гаммирования.

Список литературы