# Лабораторная работа №8. Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Евдокимов Иван Андреевич. НФИбд-01-20

20 октября, 2023, Москва, Россия

Российский Университет Дружбы Народов

# Цель лабораторной работы

# Цель лабораторной работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

# Процесс выполнения лабораторной работы

# Процесс выполнения лабораторной работы

0. Код программы

```python
# Импортируем модули для генерации случайными символов и для работы с системами счисления.
import random
import string
# Определяем класс для работы с текстовым кодированием
class TextEncoding:
    @staticmethod  # Статический метод, который не требует экземпляр класса для вызова
    def determine_alphabet(text):
        # Определяем, используется ли латиница в шифре. Если используется, возвращаем латинский алфавит, в противном случае - кириллицу
        if text[0] in string.ascii_lowercase:
            return string.ascii_lowercase + string.digits
        else:
            return "абвгдеёжзийклмнопрстуфхцчшщъыьэюя" + string.digits
    @staticmethod
    def generate_key(size, alphabet):
        # Генерируем случайный ключ того же размера, что и вводимый текст
        return "".join(random.choice(alphabet) for _ in range(size))
    @staticmethod
    def to_hex(coding):
        # Конвертируем каждый символ в шестнадцатеричное представление и объединяем их все
        return " ".join(hex(ord(character))[2:] for character in coding)
    @staticmethod
    def encode_string(text, key):
        # Возврат xor каждого символа в тексте с соответствующим символом в ключе
        return "".join(chr(ord(char) ^ ord(key_char)) for char, key_char in zip(text, key))
    @staticmethod
    def xor_texts(ciphertext1, ciphertext2):
        # Возврат xor каждого символа в двух текстах
        return "".join(chr(ord(char1) ^ ord(char2)) for char1, char2 in zip(ciphertext1, ciphertext2))
# Получаем вводимые тексты
plaintext1 = input("Введите первый открытый текст: ")
plaintext2 = input("Введите второй открытый текст: ")

# Определяем, какой алфавит использовать для генерации ключа
alphabet = TextEncoding.determine_alphabet(plaintext1)

# Генерируем ключ
key = TextEncoding.generate_key(len(plaintext1), alphabet)

# Выводим ключ и его шестнадцатеричное представление
print(f"Ключ: {key}", f"Ключ в 16 бит: {TextEncoding.to_hex(key)}", sep='\n')
# Шифруем оба текста и выводим их и их шестнадцатеричные представления
ciphertext1 = TextEncoding.encode_string(plaintext1, key)
ciphertext2 = TextEncoding.encode_string(plaintext2, key)
print(f"Первый зашифрованный текст: {ciphertext1}", f"Первый зашифрованный текст в 16 бит: {TextEncoding.to_hex(ciphertext1)}", sep='\n')
print(f"Второй зашифрованный текст: {ciphertext2}", f"Второй зашифрованный текст в 16 бит: {TextEncoding.to_hex(ciphertext2)}", sep='\n')
# Расшифровываем оба текста и выводим их
decrypted_text1 = TextEncoding.encode_string(ciphertext1, key)
decrypted_text2 = TextEncoding.encode_string(ciphertext2, key)
print("Первый расшифрованный текст:", decrypted_text1)
print("Второй расшифрованный текст:", decrypted_text2)

# Выводим результат XOR между двумя зашифрованными текстами
xor_result = TextEncoding.xor_texts(ciphertext1, ciphertext2)
print("Результат XOR двух зашифрованных текстов:", xor_result)
```

**Рис. 1:** код программы

1. Вывод запуска программы 1

```
Z:\учёба\ИНФОбез\lab8_code\venv\Scripts\python.exe Z:\учёба\ИНФОбез\lab8_code\main.py
Введите первый открытый текст: Отвратный солнечный день!
Введите второй открытый текст: Шикарная дождливая ночь.
Ключ: нщмнщы8ыйок8об6у4мфржьц2ыв
Ключ в 16 бит: 43d 449 43c 43d 449 44b 38 44b 439 43e 43a 38 43e 36 443 34 43c 444 440 436 44c 446 32 44b 432
Первый зашифрованный текст: #ᵛᵛ}y   SᵛᵛᵛO{IᵛᵛhvᵛᵛᵛyЖxsUᵛᵛГ
Первый зашифрованный текст в 16 бит: 23 b e 7d 79 9 405 0 0 41e 7b 406 5 40b 76 473 1 f 79 416 78 73 40f 7 413
    vJᵛᵛЙ
ᵛᵛЎ
Йᵛ{Iᵛᵛᵛᵛᵛᵛᵛᵛ
Второй зашифрованный текст в 16 бит: 15 71 6 d 9 76 408 4 419 a 4 40e a 40d 7b 406 c b 460 b 72 1 47e 465
Первый расшифрованный текст: Отвратный солнечный день!
Второй расшифрованный текст: Шикарная дождливая ночь.
ᵛᵛЙН
гqѣ
```

**Рис. 2:** шифровка и дишифровка текста

2. Вывод запуска программы 3 на английском

```
Z:\учёба\ИНФОбез\lab8_code\venv\Scripts\python.exe Z:\учёба\ИНФОбез\lab8_code\main.py
Введите первый открытый текст: Disgusting sunny day!
Введите второй открытый текст: A gorgeous rainy night.
Ключ: мбебюттфдоъвктОт7щэг
Ключ в 16 бит: 43c 431 435 431 44e 44e 442 442 444 434 43e 44a 432 43a 442 30 442 37 449 44d 433
Первый зашифрованный текст: оjцілнжЫЪґОйчєьІѣSШдВ
Первый зашифрованный текст в 16 бит: 478 458 446 456 43b 43d 436 42b 42a 453 41e 439 447 454 42c 49 462 53 428 434 412
Второй зашифрованный текст: ѓБЂўМЩЧЗбчОиґґЬІѣYРЂћ
Второй зашифрованный текст в 16 бит: 47d 411 452 45e 43c 429 427 42d 431 447 41e 438 453 453 42c 49 462 59 420 42a 45b
Первый расшифрованный текст: Disgusting sunny day!
Второй расшифрованный текст: A gorgeous rainy nigh
Результат XOR двух зашифрованных текстов: ⁿ€I
ⁿI

Process finished with exit code 0
```

**Рис. 3:** шифровка и дишифровка текста

# Контрольные вопросы

# Контрольные вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа? Ответ: Это возможно сделать только в том случае если текст P1 и P2 одной длины и имеют общий ключ.

2. Что будет при повторном использовании ключа при шифровании текста? Ответ: Из-за одинаковости способа кодирование и декодирование после повторного использование слова и ключа даст нам шифротекст.

3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов? Ответ: Фактически следуя схеме 8.1 и принципу “шифра XOR” мы просто имеет два параллельных кодирование и декодирование с использованием одного ключа.

# Контрольные вопросы

4. Перечислите недостатки шифрования одним ключом двух открытых текстов. Ответ: Если вспомнить тредования для абсолютной стойкости шифра расмотренных в предедущей лабораторной то можно сразу понять по первому пункту что если ключ не будет случайным и каждый раз новым для каждой строки то найдя пересечения или аналоги в шифротекстах можно определить одинаковые символы что может пошатнуть защиту текста даже если у вас нет ни одного исходного кода, а если и есть то определить другие слова легко.

# Контрольные вопросы

5. Перечислите преимущества шифрования одним ключом двух открытых текстов. Ответ: На самом деле они есть, но они сомнительны: требуется передать один ключ что сделать проще и быстрее, при передаче большого количества шифротекста нет шанса запутаться в их порядке сочетания с ключами.

# Выводы:

# Выводы:

Освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.