

Лабораторная работа №8

Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

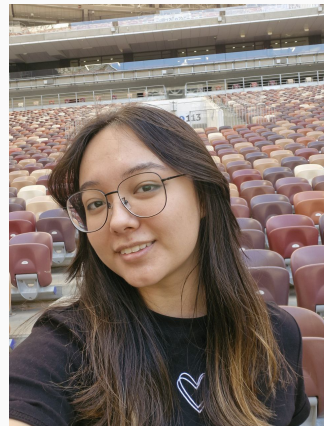
Латыпова Диана

24 октября 2024

Российский университет дружбы народов, Москва, Россия

Информация

- Латыпова Диана
- студент группы НФИбд-02-21
- Российский университет дружбы народов
- 1032215005@rudn.ru
- <https://github.com/dlatypova>



Вводная часть

- Освоение на практике применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

Теоретическое введение

- метод шифрования, при котором каждый бит открытого текста комбинируется с битом ключа с помощью операции *исключающего ИЛИ* (XOR).

$$[C = P \oplus K]$$

где: - (C) — шифртекст, - (P) — открытый текст, - (K) — ключ.

- **Идеальная стойкость**
- **Простота реализации**
- **Гибкость**

Недостатки:

- Проблемы с управлением ключами
- Уязвимость к атаке при повторном использовании ключа
- Необходимость в хранении ключей

Задание

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитав оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты $P1$ и $P2$ в режиме однократного гаммирования. Приложение должно определить вид шифротекстов $C1$ и $C2$ обоих текстов $P1$ и $P2$ при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Выполнение лабораторной работы

```
[5] start_char = ord("a")
    alphabet = [chr(i) for i in range(start_char, start_char + 32)]
    start_char = ord("0")
    for i in range(start_char, start_char + 10):
        alphabet.append(chr(i))

    start_char = ord("A")
    for i in range(1040, 1072):
        alphabet.append(chr(i))
    print(alphabet)
```

Рис. 1: Алфавит из русских букв и цифр для гаммирования

```
P1 = "НаВашисходящийот1204"  
P2 = "ВСеверныйфилиалБанка"  
K = "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54"  
  
def hack(P1, P2):  
    xor_code = []  
    for i in range(20):  
        xor_code.append(alphabeth[(alphabeth.index(P1[i]) + alphabeth.index(P2[i])) % len(alphabeth)])  
    print(xor_code)  
    print(xor_code[16], " и ", xor_code[19])  
    combined_text = "".join(xor_code)  
    print(combined_text)  
  
hack(P1, P2)
```


 ['ш', 'С', 'з', 'в', 'э', 'ш', 'ю', 'ж', 'ч', 'ш', '7', '4', 'р', 'я', 'ш', 'у', '1', 'Е', 'А', '4']
1 и 4
щСэвэжжчш74рщу1ЕА4

Рис. 2: Вызов функции hack

```
def encrypt(P1):
    char_to_num = {"a": 1, "b": 2, "c": 3, "d": 4, "e": 5, "f": 6, "g": 7, "h": 8, "i": 9, "j": 10, "k": 11, "l": 12, "m": 13,
                  "n": 14, "o": 15, "p": 16, "q": 17, "r": 18, "s": 19, "t": 20, "u": 21, "v": 22, "w": 23, "x": 24, "y": 25,
                  "z": 26, "A": 27, "B": 28, "C": 29, "D": 30, "E": 31, "F": 32, "G": 33, "H": 34, "I": 35, "J": 36,
                  "K": 37, "L": 38, "M": 39, "N": 40, "O": 41, "P": 42, "Q": 43, "R": 44, "S": 45, "T": 46, "U": 47, "V": 48,
                  "W": 49, "X": 50, "Y": 51, "Z": 52, "a": 53, "b": 54, "c": 55, "d": 56, "e": 57, "f": 58, "g": 59, "h": 60,
                  "i": 61, "j": 62, "k": 63, "l": 64, "m": 65, "n": 66, "o": 67, "p": 68, "q": 69, "r": 70, "s": 71, "t": 72,
                  "u": 73, "v": 74, "w": 75}

    num_to_char = {v: k for k, v in char_to_num.items()}

    input_text = P1
    gamma = input("Введите гамму (только символы из строки): ")

    text_nums = []
    gamma_nums = []

    for char in input_text:
        text_nums.append(char_to_num[char])
        print("числа текста", text_nums)

    for char in gamma:
        gamma_nums.append(char_to_num[char])
        print("числа гаммы", gamma_nums)

    encrypted_nums = []
    idx = 0

    for char in input_text:
        try:
            new_num = char_to_num[char] + gamma_nums[idx]
        except:
            idx = 0
            new_num = char_to_num[char] + gamma_nums[idx]

        if new_num > 75:
            new_num = new_num % 75
            idx += 1
        encrypted_nums.append(new_num)

    print("числа зашифрованного текста", encrypted_nums)
```

Рис. 3: Первая часть основного кода

Код (4)

```
encrypted_text = ""
for num in encrypted_nums:
    encrypted_text += num_to_char[num]

print("Зашифрованный текст: ", encrypted_text)

decrypted_nums = []
for char in encrypted_text:
    decrypted_nums.append(char_to_num[char])

idx = 0
decrypted_result = []

for num in decrypted_nums:
    try:
        new_num = num - gamma_nums[idx]
    except:
        idx = 0
        new_num = num - gamma_nums[idx]

    if new_num < 1:
        new_num = 75 + new_num
    decrypted_result.append(new_num)
    idx += 1

decrypted_text = ""
for num in decrypted_result:
    decrypted_text += num_to_char[num]

print("Расшифрованный текст", decrypted_text)
```

encrypt(P1)


```
encrypt(P1)
```

↕

Введите гамму (только символы из словаря): цСЗавзюжч74рйщ1ЕА4
Числа текста [47, 1, 35, 1, 26, 10, 19, 23, 16, 5, 32, 27, 10, 11, 16, 20, 66, 67, 75, 69]
Числа гаммы [27, 51, 41, 3, 31, 26, 32, 40, 25, 26, 72, 69, 18, 11, 27, 53, 66, 38, 33, 69]
Числа зашифрованного текста [74, 52, 1, 4, 57, 36, 51, 63, 41, 31, 29, 21, 28, 22, 43, 73, 57, 30, 33, 63]
Зашифрованный текст: 9ТагЧГСЭззюф08ЧыАЭ
Расшифрованный текст: НаВашисходящийот1204

Рис. 5: Результат

12/13

Выводы

- Освоено на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом