

Шифр гаммирования

Дмитревская Софья Алексеевна НФИбд-01-19

29 октября, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

Цель лабораторной работы

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

Выполнение лабораторной работы

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования:

$$C_1 = P_1 \oplus K$$

$$C_2 = P_2 \oplus K$$

Открытый текст можно найти, зная шифротекст двух телеграмм, зашифрованных одним ключом. Для это оба равенства складываются по модулю 2. Тогда с учётом свойства операции XOR получаем:

$$C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$

Предположим, что одна из телеграмм является шаблоном — т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар $C_1 \oplus C_2$ (известен вид обеих шифровок). Тогда зная P_1 имеем:

$$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2$$

Схема работы алгоритма



Figure 1: Работа алгоритма гаммирования

Пример работы программы

```
In [17]: a = ord("a")
alphabet = [ chr(i) for i in range (a, a+32)]
a = ord("0")
for i in range(a, a+10):
    alphabet.append(chr(i))
a = ord("A")
for i in range(1040, 1072):
    alphabet.append(chr(i))

P1="НаВашисходящийот1204"
P2="ВСеверныйфилиалБанка"
```

```
In [18]: def vzlom(P1, P2):
    code = []
    for i in range(20):
        code.append(alphabet[(alphabet.index(P1[i]) + alphabet.index(P2[i])) % len(alphabet) ])
    print(code)
    p3 = "".join(code)
    print(p3)
```

```
In [19]: vzlom(P1, P2)
```

```
['щ', 'С', 'З', 'в', 'а', 'ш', 'ю', 'Ж', 'ч', 'ш', '7', '4', 'р', 'й', 'щ', 'У', '1', 'Е', 'А', '4']
щСЗвщЖчш74рйщУ1ЕА4
```

Figure 2: Работа алгоритма взлома ключа

Пример работы программы

```
In [22]: def shifr(P1):
dicts = {"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
    }

    dict2 = {v: k for k, v in dicts.items()}
    text = P1
    gamma = input("Введите гамму")
    digits_gamma = list()
    digits_text = list()

    for i in text:
        digits_text.append(dicts[i])
    print("Исходный текст: ", digits_text)

    for i in gamma:
        digits_gamma.append(dicts[i])
    print("Исходная гамма: ", digits_gamma)

    digits_result = list()
    ch = 0
    for i in text:
        try:
            a = dicts[i] + digits_gamma[ch]
        except:
            ch = 0
            a = dicts[i] + digits_gamma[ch]
        if a > 75:
            a = a % 75
            print(a)
            ch = ch + 1
        digits_result.append(a)

    print("Исходный шифр: ", digits_result)

    text_crypted = ""
    for i in digits_result:
        text_crypted = text_crypted + dict2[i]
    print("Исходное сообщение: ", text_crypted)

    digits = list()
    for i in text_crypted:
        digits.append(dicts[i])
    ch = 0
    digits1 = list()
    for i in digits:
        try:
            a = i - digits_gamma[ch]
        except:
            ch = 0
            a = i - digits_gamma[ch]
        if a < 1:
            a = 75 + a
            digits1.append(a)
            ch = ch + 1
        else:
            digits1.append(a)
            ch = ch + 1

    text_dec = ""
    for i in digits1:
        text_dec = text_dec + dict2[i]
    print("Расшифрованное сообщение: ", text_dec)
```

Контрольный пример

```
In [23]: shifr(P1)

Введите гамму:СЭвэюЖч74рйцY1ЕА4
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]
1
29
21
57
30
33
63
Числа шифровки [74, 52, 1, 4, 57, 36, 51, 63, 41, 31, 29, 21, 28, 22, 43, 73, 57, 30, 33, 63]
Шифровка 9ТагЧГСЭзэуьфЙ8ЧьАЭ
Расшифровка НаВэишисходящийот1284
```

Figure 4: Результат работы алгоритма шифрования и дешифровки

Выводы

В ходе выполнения лабораторной работы было разработано приложение, позволяющее шифровать тексты в режиме однократного гаммирования.