

# Шифр гаммирования

---

Балтаева Галина НПИбд-01-19

27 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 [10]: 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 [11]: 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 [12]: vzlom(P1, P2)

['ш', 'С', 'З', 'в', 'э', 'ш', 'ю', 'ж', 'ч', 'ш', '7', '4', 'р', 'й', 'щ', 'у', '1', 'Е', 'А', '4']
щСЗэвэШюжчШ74рщУ1ЕА4
```

Figure 2: Работа алгоритма взлома ключа

```
print("Расшифровка", text_decrypt)
```

```
In [18]: shifr(P1)

Введите гамму:щСЗэвэШюжчШ74рщУ1ЕА4
Числа текста [47, 1, 35, 1, 26, 42, 19, 23, 16, 5, 32, 27, 10, 11, 16, 20, 66, 67, 75, 69]
Числа гаммы [27, 51, 41, 3, 31, 58, 32, 40, 25, 58, 72, 69, 18, 11, 27, 53, 66, 38, 33, 69]
1
25
29
21
57
30
33
63
Числа шифровки [74, 52, 1, 4, 57, 25, 51, 63, 41, 63, 29, 21, 28, 22, 43, 73, 57, 30, 33, 63]
Шифровка 9ТагЧсЭЗЗэуьфй8ЧьАэ
Расшифровка НаВашИсходящийот1204
```

## **Выводы**

---

## Результаты выполнения лабораторной работы

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