

# Основы информационной безопасности

Элементы криптографии. Однократное гаммирование

---

Сунгурова Мариян Мухсиновна

## Информация

---

- Сунгурова Мариян Мухсиновна
- НКНбд-01-21
- Российский университет дружбы народов

## Введение

---

Освоить на практике применение режима однократного гаммирования.

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Гаммиирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст(intro\_crypto\_2017?): Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных.

## Выполнение лабораторной работы

---



## Выполнение лабораторной работы

Создадим функции: **get\_key** – отвечает за генерацию случайного ключа(составляется выбором из букв кириллицы больших и малых, символов, цифр), **encrypt** – принимает на вход текст и ключ, а затем осуществляет посимвольное сложение по модулю 2, **decrypt** – подбирает точную часть ключа для известного фрагмента сообщения, а затем оставшуюся часть выбирает случайным образом(используются ранее описанные функции): ##

Выполнение лабораторной работы

```
def get_key(text):    l = [chr(i) for i in range(0x0410, 0x042F + 1)]  
+ [chr(i) for i in range(0x0030, 0x0039 + 1)]    key =  
"".join([random.choice(l) for i in range(len(text))])    return key
```

```
def encrypt(text, key):    return  
"".join([chr(ord(key[i])^ord(text[i])) for i in range(len(key))])
```

```
def decrypt(text, key):    k = encrypt(text, key[:len(text)])  
return k + get_key(key[len(text):])
```

Затем применим эти функции к заданному сообщению:

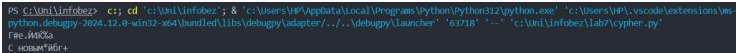
```
key = get_key(key_text) encrypted = encrypt(text_1, key)
print(encrypted) fragment = "С новым" # известный фрагмент сообщения
part_key = decrypt(fragment, encrypted) # ключ на основе фрагмента
сообщения guess = encrypt(encrypted, part_key) # предположительный
текст print(guess)
```

# Выполнение лабораторной работы

```
lab7 > cypher.py > ...
17
18
19 text_1 = "👁 Новым Годом!"
20
21 def get_key(text):
22     l = [chr(i) for i in range(0x0410, 0x042F + 1)] + [chr(i) for i in range(0x0030, 0x0039 + 1)]
23     key = "".join([random.choice(l) for i in range(len(text))])
24     return key
25
26 def encrypt(text, key):
27     return "".join([chr(ord(key[i])^ord(text[i])) for i in range(len(key))])
28
29 def decrypt(text, key):
30     k = encrypt(text, key[:len(text)])
31     return k + get_key(key[len(text):])
32
33
34 key = get_key(key_text)
35 encrypted = encrypt(text_1, key)
36
37 print(encrypted)
38
39 fragment = "👁 Новым" # известный фрагмент сообщения
40 part_key = decrypt(fragment, encrypted) # ключ на основе фрагмента сообщения
41 guess = encrypt(encrypted, part_key) # предположительный текст
42 print(guess)
```

Рис. 1: Код программы

В результате получим следующий вариант шифрования и один из вариантов прочтения текста(рис. (fig:002?))



```
PS C:\Uni\infobez> c::; cd 'c:\Uni\infobez'; & 'c:\Users\HP\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\HP\.vscode\extensions\ms-python.debugpy-2024.12.0-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher' '63718' '--' 'c:\Uni\infobez\lab7\cypher.py'
```

Рис. 2: Результаты работы программы

## Контрольные вопросы

---

1. Поясните смысл однократного гаммирования.

Гаммирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть.

2. Перечислите недостатки однократного гаммирования.

- Если один и тот же ключ используется для шифрования нескольких сообщений, это может привести к уязвимостям. Например, если злоумышленник узнает открытый текст и соответствующий шифротекст, он может использовать эту информацию для взлома ключа.
- Однократное гаммирование не обеспечивает аутентификацию или целостность данных. Это означает, что злоумышленник может изменить шифротекст без заметных изменений в открытом тексте.

3. Перечислите преимущества однократного гаммирования.

- Однократное гаммирование обеспечивает высокий уровень конфиденциальности, поскольку шифротекст не может быть легко взломан без знания ключа.
- Однократное гаммирование обеспечивает равномерное распределение вероятностей для каждого символа в шифротексте, что делает его статистически неразличимым от случайной последовательности.
- Однократное гаммирование является простым и быстрым методом шифрования.

4. Почему длина открытого текста должна совпадать с длиной ключа?

Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть.

5. Какая операция используется в режиме однократного гаммирования, назовите её особенности?

В режиме однократного гаммирования используется операция XOR (исключающее ИЛИ). Операция XOR комбинирует биты открытого текста и ключа, чтобы получить шифротекст. Особенностью операции XOR является то, что она возвращает 1 только в том случае, если один из входных битов равен 1, но не оба.



6. Как по открытому тексту и ключу получить шифротекст?

Нужно побитово сложить по модулю численное представление символов в ключе и в открытом тексте.

7. Как по открытому тексту и шифротексту получить ключ?

Нужно побитово сложить по модулю численное представление символов в шифротексте и в открытом тексте.

8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

Необходимые и достаточные условия абсолютной стойкости шифра:

- полная случайность ключа;
- равенство длин ключа и открытого текста;
- однократное использование ключа.

## Выводы

---

В результате выполнения работы были освоены практические навыки применения режима однократного гаммирования.