Отчёт по лабораторной работе №7

ван яо"

Содержание

# Цель работы

Освоить на практике применение режима однократного гаммирования.

# Порядок выполнения работы

XOR две строки

Определить функцию с именем xor\_strings, которая принимает два строковых параметра s1 и s2.

def xor\_strings(s1, s2):  
 return ''.join(chr(ord(a) ^ ord(b)) for a, b in zip(s1, (b for b in s2 \* (len(s1) // len(s2) + 1))))

* zip(s1, (b for b in s2 \* (len(s1) // len(s2) + 1))): здесь используется выражение-генератор (b for b in s2 \* (len(s1) // len( s2) + 1)) для повторения строки s2 так, чтобы ее длина была как минимум той же длины, что и s1.
* ord(a) ^ ord(b): операция XOR для каждого символа a и b.
* chr(...): преобразовать результат XOR обратно в символы.
* ''.join(...): Объединить все символы в строку и вернуть

Зашифровать открытый текст с помощью одноразового блокнота  
Определить функцию с именем encrypt, которая принимает два параметра: открытый текст и ключ.

def encrypt(plaintext, key):  
 return xor\_strings(plaintext, key)

Расшифровать зашифрованный текст с помощью одноразового блокнота  
Определить функцию с именем decrypt, которая принимает два параметра: зашифрованный текст и ключ.

def decrypt(ciphertext, key):  
 return xor\_strings(ciphertext, key)

Учитывая зашифрованный текст и целевой открытый текст, найдите соответствующий ключ.  
Определить функцию с именем find\_key\_for\_plaintext, которая принимает два параметра ciphertext и target\_plaintext.

def find\_key\_for\_plaintext(ciphertext, target\_plaintext):  
 return xor\_strings(ciphertext, target\_plaintext)

использование  
Убедитесь, что приведенный ниже код выполняется только тогда, когда этот файл запускается в качестве основной программы.

if \_\_name\_\_ == "\_\_main\_\_":  
 # Известная информация  
 known\_plaintext = "С Новым Годом, друзья!"  
 known\_key = "1234567456734567890898"  
 #Предположим, у нас есть известный ключ   
 # Процесс шифрования  
 plaintext\_bytes = known\_plaintext.encode('utf-8')  
 key\_bytes = known\_key.encode('utf-8')  
 ciphertext\_bytes = xor\_strings(plaintext\_bytes.decode('utf-8'), key\_bytes.decode('utf-8')).encode('utf-8')  
 ciphertext = ciphertext\_bytes.decode('utf-8')  
 print(f"ciphertext: {ciphertext}")  
   
 # Процесс расшифровки  
 decrypted\_text\_bytes = xor\_strings(ciphertext, key\_bytes.decode('utf-8')).encode('utf-8')  
 decrypted\_text = decrypted\_text\_bytes.decode('utf-8')  
 print(f"plaintext: {decrypted\_text}")  
  
 # Найдите ключ, необходимый для определенного открытого текста  
 target\_plaintext = "С Новым Годом, друзья!"  
 found\_key\_bytes = xor\_strings(ciphertext, target\_plaintext).encode('utf-8')  
 found\_key = found\_key\_bytes.decode('utf-8')  
 print(f"key: {found\_key}")

# Контрольные вопросы

### 1. Пояснение смысла однократного гаммирования

Однократное гаммирование (one-time pad) — это метод шифрования, при котором текст сообщения (открытый текст) складывается по модулю два (XOR) с случайным ключом такой же длины. Если ключ используется только один раз и хранится в секрете, то данный метод считается абсолютно надёжным.

### 2. Недостатки однократного гаммирования

* Сложность управления ключами
* Затруднённая дистрибуция ключей
* Необходимость безопасного хранения ключей

### 3. Преимущества однократного гаммирования

* Абсолютная безопасность (при правильном применении)
* Простота реализации
* Высокая скорость шифрования и дешифрования

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

Для обеспечения безопасности каждый символ открытого текста должен быть зашифрован соответствующим символом ключа. Если длина ключа не совпадает с длиной открытого текста, то шифр становится уязвимым.

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

Используется операция **XOR (исключающее ИЛИ)**.

* Особенности: простота выполнения, обратимость, непредсказуемость результата при использовании случайного ключа.

### 6. Как по открытому тексту и ключу получить шифротекст?

Шифротекст получают путём применения операции XOR между каждым битом открытого текста и соответствующим битом ключа.

### 7. Как по открытому тексту и шифротексту получить ключ?

Чтобы восстановить ключ, нужно применить операцию XOR между каждым битом открытого текста и соответствующим битом шифротекста.

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

* Ключ должен быть абсолютно случайным.
* Длина ключа должна быть не меньше длины открытого текста.
* Ключ не должен использоваться для шифрования других сообщений.
* Ключ должен оставаться секретным.

# выводы

Освоил на практике применение режима однократного гаммирования.