Лабораторная работа № 8

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

Хамдамова А. А.

17 May

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Хамдамова Айжана Абдукаримовна
- студент Факультета Физико-математических и естественных наук
- Российский университет дружбы народов
- 1032225989@pfur.ru
- https://github.com/AizhanaKhamdamova/study_2023-2024_infosec

Вводная часть

Исходные данные. Две телеграммы Центра: P1 = НаВашисходящийот1204 P2 = ВСеверныйфилиалБанка Ключ Центра длиной 20 байт: K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 OB B2 70 54 Режим шифрования однократного гаммирования одним ключом двух видов открытого текста реализуется в соответствии со схемой, приведённой на рис. 8.1. Шифротексты обеих телеграмм можно получить по формулам режима однократного гаммирования: C1 = P1 ⊕ K, C2 = P2 ⊕ K.

Тогда с учётом свойства операции XOR 1 ⊕ 1 = 0, 1 ⊕ 0 = 1 (8.2) получаем: C1 ⊕ C2 = P1 ⊕ K ⊕ P2 ⊕ K = P1 ⊕ P2. 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 C1 ⊕ C2 (известен вид обеих шифровок). Тогда зная P1 и учитывая (8.2), имеем: C1 ⊕ C2 ⊕ P1 = P1 ⊕ P2 ⊕ P1 = P2. (8.3) Таким образом, злоумышленник получает возможность определить те символы сообщения P2, которые находятся на позициях известного шаблона сообщения P1. В соответствии с логикой сообщения P2, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения P2. Затем вновь используется (8.3) с подстановкой вместо P1 полученных на предыдущем шаге новых символов сообщения P2. И так далее. Действуя подобным образом, злоумышленник даже если не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

Цели и задачи

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

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать 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

In [4]:
```python
import random
import string

def xor_text_f(text, key):
    if len(key) != len(text):
        return "Ошибка. Ключ и текст разной длины"
    xor_text = ''
    for i in range(len(key)):
        xor_text_symbol = ord(text[i]) ^ ord(key[i])
        xor_text += chr(xor_text_symbol)
    return xor_text

# Пример использования
text = 'С Новым годом, друзья!'
key = ''

# Генерация ключа той же длины, что и текст
random.seed(22)
for i in range(len(text)):
    key += random.choice(string.ascii_letters + string.digits)

# Шифрование текста
xor_text = xor_text_f(text, key)
print("Зашифрованный текст:", xor_text)

# Дешифрование текста (шифрование снова тем же ключом)
decrypted_text = xor_text_f(xor_text, key)
print("Расшифрованный текст:", decrypted_text)
```

```
Зашифрованный текст: И▯VюёSШLЏiЪ▯J[yЁЦbxvЫТ
Расшифрованный текст: С Новым годом, друзья!
```

```
Расшифрованный текст: С Новым годом, друзья!
```

In [8]:

```python
# Пример для двух текстов
P1 = 'С Новым годом, друзья!'
P2 = 'Be happyyyyyyyyyyyyy!'

# Генерация одного ключа для обоих текстов
key = ''
random.seed(22)
for i in range(len(P1)):
    key += random.choice(string.ascii_letters + string.digits)

# Шифрование обоих текстов
C1 = xor_text_f(P1, key)
C2 = xor_text_f(P2, key)

# Определение P1 XOR P2 без знания ключа
P1_XOR_P2 = xor_text_f(C1, C2)
print("P1 XOR P2:", P1_XOR_P2)
```

```
P1 XOR P2: ЬЕніґльУьчэчхUУэйкюеж
```