Презентация

Ощепков Дмитрий Владимирович НКАбд-02-22

Дисциплина: Основы информационной безопасности

Лабораторная работа №8

Цель работы: освоить на практике применение режима однократного гаммирования

Порядок выполнения работы:

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

Приложение должно:

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 [19]:
```
def xor_bytes(data, key):
    return bytes(a ^ b for a, b in zip(data, key))

# Представление ключа в hex формате
K_hex = "05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54"
K_bytes = bytes.fromhex(K_hex)

# Исходные данные
P1 = "НаВашисходящийот1204".encode('utf-8')
P2 = "ВСеверныйфилиалБанка".encode('utf-8')

# Проверка длины ключа и данных и подгонка длины ключа
max_len = max(len(P1), len(P2))
K_bytes = (K_bytes * ((max_len + len(K_bytes) - 1) // len(K_bytes)))[:max_len]

# Шифрование
C1 = xor_bytes(P1, K_bytes)
C2 = xor_bytes(P2, K_bytes)

print("P1 (Hex):", P1.hex())
print("P2 (Hex):", P2.hex())
print("K_bytes (Hex):", K_bytes.hex())
print("C1 (Hex):", C1.hex())
print("C2 (Hex):", C2.hex())

# Дешифрование для проверки
P1_decrypted = xor_bytes(C1, K_bytes)
P2_decrypted = xor_bytes(C2, K_bytes)
print("P1_decrypted:", P1_decrypted.decode('utf-8'))
print("P2_decrypted:", P2_decrypted.decode('utf-8'))
```

```
P1 (Hex): d09dd0b0d092d0b0d188d0b8d181d185d0bed0b4d18fd189d0b8d0b9d0bed18231323034
P2 (Hex): d092d0a1d0b5d0b2d0b5d180d0bdd18bd0b9d184d0b8d0bbd0b8d0b0d0bbd091d0b0d0bdd0bad0b0
K_bytes (Hex): 050c177f0e4e37d29410092e2257ffc80bb27054050c177f0e4e37d29410092e2257ffc80bb27054
C1 (Hex): d591c7cfdedce7624598d996f3d62e4ddb0ca0e0d483c6f6def6e76b44aed8ac1365cffc
C2 (Hex): d59ec7dedefbe76044a5d8aef2ea2e43db0ba1d0d5b4c7c4def6e76244abd9bff2e72f75db08a0e4
P1_decrypted: НаВашисходящийот1204
```

```
P2_decrypted: ВСеверныйфилиалБанка
```

In [22]:
```python
P1_xor_P2 = xor_bytes(C1, C2)

# Предположим, что мы знаем, что один из оригинальных текстов начинается с "НаВашисходящийот1204"
# Это предположение может быть сделано на основе контекста или другой информации
known_text = "НаВашисходящийот1204".encode('utf-8')

# Применяем операцию XOR к P1_xor_P2 и известному тексту, чтобы получить второй оригинальный текст
P2 = xor_bytes(P1_xor_P2, known_text)

# Декодируем первый оригинальный текст (для проверки)
P1 = xor_bytes(C1, P2)

print("P1 (Decoded):", P1.decode('latin-1'))
print("P2 (Decoded):", P2.decode('utf-8'))
```

```
P1 (Decoded): ▯▯▯n▯i7Ð▯▯#kÿÆ▯µqd▯;▯M▯N7Û▯=ÃÕ▯A
P2 (Decoded): ВСеверныйфилиалБан
```

In [ ]:

Вывод: освоил на практике применение режима однократного гаммирования.