

Отчет по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

Татьяна Александровна Буллер

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Определение кодировки шифротекста	5
2.2	Программа дешифрования по известному открытому тексту	6
3	Выводы	9

Список иллюстраций

2.1	Кодировка файла	5
2.2	Код программы	7
2.3	Расшифровка полученного с новым ключом сообщения	8

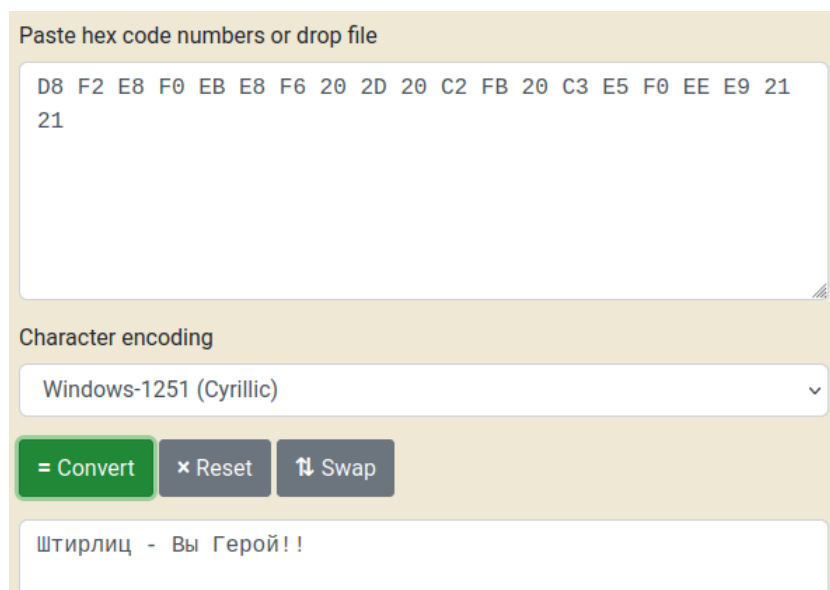
1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Выполнение лабораторной работы

2.1 Определение кодировки шифротекста

Сообщение “D8 F2 E8 F0 EB E8 F6 20 2D 20 C2 FB 20 C3 E5 F0 EE E9 21 21” написано на русском языке, однако при переводе его из hex в текст стандартных кодировок ASCII/UTF-8 результат не совпадал с тем, какой был задан условием задания. Путем перебора кодировок было выяснено, что сообщение было написано в кодировке Windows-1251.



Paste hex code numbers or drop file

D8 F2 E8 F0 EB E8 F6 20 2D 20 C2 FB 20 C3 E5 F0 EE E9 21 21

Character encoding

Windows-1251 (Cyrillic)

= Convert × Reset ↕ Swap

Штирлиц - Вы Герой!!

Рис. 2.1: Кодировка файла

2.2 Программа дешифрования по известному открытому тексту

Далее необходимо было написать код, с помощью которого можно было бы дешифровать сообщение. Метод шифрования XOR крайне уязвим к атакам по известному открытому тексту, поэтому, зная сообщение на выходе, мы без труда можем получить ключ, которым нужно было зашифровать строку, чтобы его получить. Для этого нужно обернуть операцию. Переведем известные нам строки шифра и открытого текста ("С Новым Годом, друзья!" в кодировке Windows-1251) в hex. Для того, чтобы далее провести с ними операции XOR, нужно перевести эти строки далее в бинарный формат. После сравним строки посимвольно и запишем результат: если символ в позиции n строки А совпадает с символом в той же позиции в строке Б, то в результат дописывается 0, иначе - 1. Необходимо отметить, что для корректной работы кода и получения полного ключа строки должны совпадать по длине. В случае, если какая-то из них короче другой, она повторяется до того, как длины совпадут.

```

1 # само за себя говорит
2 def hex_to_bin(ints):
3     scale = 16
4     res = bin(int(ints, scale)).zfill(8)
5     return res
6
7 # xor для двух строк (должны быть одинаковой длины, чтобы получить полный
  ключ/шифротекст. в случае, если длины строк не совпадают, ту, что короче,
  необходимо повторить до соответствия длине второй строки)
8 def xor(a, b, n):
9     ans = ""
10    for i in range(n):
11        if (a[i] == b[i]):
12            ans += "0"
13        else:
14            ans += "1"
15    return ans
16
17 # main code
18 if __name__ == "__main__":
19     a = hex_to_bin('d8f2e8f0ebe8f6202d20c2fb20c3e5f0eee92121d8f2')
20     b = hex_to_bin('d120cdeee2fbec20c3eee4eeec2c20e4f0f3e7fcff21')
21
22     n = len(a)
23     c = xor(a, b, n)
24     print(c)
25

```

Рис. 2.2: Код программы

Программа выдает ключ в формате двоичного числа, который при необходимости далее можно перевести в шестнадцатичный формат. Используя полученный ключ вместо шифротекста, мы получим другой двоичный вывод. Переведя его в текст и расшифровав в кодировке Windows-1251 получим сообщение, ключ к которому и хотели найти, что говорит о том, что код сработал корректно.

Paste hex code numbers or drop file

```
D1 20 CD EE E2 FB EC 20 C3 EE E4 EE EC 2C 20 E4 F0 F3 E7  
FC FF 21
```

Character encoding

Windows-1251 (Cyrillic) ▾

= Convert × Reset ↕ Swap

С Новым Годом, друзья!

Рис. 2.3: Расшифровка полученного с новым ключом сообщения

3 Выводы

Было освоено на практике применение режима однократного гаммирования, написана программа, переводящая строки из шестнадцатиричного формата в двоичный и проводящая между ними XOR-операцию посимвольно.