30.11.2023

Курс:
Практическая работа к уроку No Lesson_1
--
Введение в курс

Задание_1:

Одна из кодировок не разбиралась на уроке и чтобы ее раскодировать нужно немного поискать.
В этом задании вам необходимо “расшифровать” 5 “шифротекстов”:

WW91IGRpZCBpdCE=
S2VlbiBvbiBnb2luZyE=
VG9wIHNlY3JldCBpbmZvcm1hdGlvbiBpcyBuZWFyIHlvdS4u
KRUGC5BAO5QXGIDUOJUWG23ZEBXW4ZJB
Q29uZ3JhdHVsYXRpb25zISBZb3UndmUgJ2RlY3J5cHRlZCcgYWxsIG1lc3NhZ2VzIQ==

В результате вы получите 5 осмысленных фраз на английском языке.

XOR encrypt ? Text

Код Цезаря

1. #!python
2. >>> from pycipher import Caesar
3. >>> Caesar(key=1).encipher('The quick brown fox jumps over the lazy dog')
4. 'UIFRVJDLCSPXOGPYKVNQTPWFSUIFMBAZEPH'
5. >>> Caesar(key=1).decipher('UIFRVJDLCSPXOGPYKVNQTPWFSUIFMBAZEPH')
6. 'THEQUICKBROWNFOXJUMPSOVERTHELAZYDOG'

Задание 2

“Расшифруйте” следующие “шифротексты”:

4c6f72656d20497073756d2069732073696d706c792064756d6d7920746578742

6f6620746865207072696e74696e6720616e642074797065736574696e6720696e6475737472792e
436865636b206f7574207468697320746c6b20696620796f75206861766e277420646f6e6520736f2068747470733a2f2f7777772e796f75747562652e636f6d2f77617463683f763d6d4b535136446a427a3377

В результате вы получите осмысленные фразы на английском языке.

XOR hex

Задание 3
Во всех последующих заданиях вы будете работать с “сырыми” байтами. Однако у большей части байт нет соответствующего печатного символа (например, 0x00 будет в лучшем случае напечатан как пробел), поэтому данные для заданий будут даваться в кодировках base64 и hex.

Следовательно, очень полезно научиться кодировать и декодировать их на вашем любимом языке программирования.

Декодируйте строку:

49276d206b696c6c696e6720796f757220627261696e206c696b65206120706f69736f6e6f7573206d757368726f6f6d

Результат представьте в base64. На выходе должно получиться:

SSdtIGtpbGxpbmcgeW91ciBicmFpbiBsaWtlIGEgcG9pc29ub3VzIG11c2hyb29t

Задание взято из http://cryptopals.com/sets/1/challenges/1

Задание 4
Напишите функцию, которая принимает на вход две последовательности байт одинаковой длины и возвращает их побайтовый XOR.

Ваша функция работает правильно, если приняв на вход hex-декодированные

1c0111001f010100061a024b53535009181c

и

686974207468652062756c6c277320657965

вернет hex-декодированное значение

746865206b696420646f6e277420706c6179

В следующем здании мы будем искать уязвимости в нашем свежеиспеченном шифре!

Задание взято из http://cryptopals.com/sets/1/challenges/2

Задание 5
Hex-кодированная строка

1b37373331363f78151b7f2b783431333d78397828372d363c78373e783a393b3736

была зашифрована Single-byte XOR-ом.

Ваша задача найти ключ шифрования и расшифровать сообщение. Как? Придумайте метрику для оценки открытого текста (частотное распределение символов - хорошая метрика), оцените каждый открытый текст по этой метрике и выберите текст с лучшим показателем.

Примечание: вы можете сделать это задание вручную, но так делать не стоит, напишите программу и пусть она сделает это за вас.

Задание взято из http://cryptopals.com/sets/1/challenges/3

Задание 6
Одна из строк в файле 6.txt зашифрована Single-byte XOR-ом. Найдите ее (программа из задания 5 поможет вам).

Задание взято из http://cryptopals.com/sets/1/challenges/4

Задание 7
В этом задании вам необходимо реализовать Repeating-key XOR. Зашифруйте следующие строки на ключе ICE:

Burning 'em, if you ain't quick and nimble
I go crazy when I hear a cymbal

В результате у вас получится шифротекст:

0b3637272a2b2e63622c2e69692a23693a2a3c6324202d623d63343c2a26226324272765272a282b2f20
0063222663263b223f30633221262b690a652126243b632469203c24212425

Задание взято из http://cryptopals.com/sets/1/challenges/5

Задание 8 (*)
В этот раз мы (традиционно) взломаем шифр, который создали в предыдущем задании.

Вам необходимо расшифровать файл 8.txt. Он был закодирован в base64 после применения repeating-key XOR. Используйте следующий алгоритм:

Шаг 1. Определить длину ключа KEYSIZE (подсказка, значение ключа лежит в диапазоне от 2 до 40).

Напишите функцию, которая вычисляет расстояние Хэмминга между двумя строками. Расстояние Хэмминга это количество отличающихся бит в строках. Расстояние между this is a test и wokka wokka!!! равно 37. Убедитесь, что ваша функция возвращает такой же результат.
Для каждого KEYSIZE:
Возьмите первые KEYSIZE байт и вторые KEYSIZE байт.
Найдите расстояние между ними.
Нормализуйте результат, поделив его на KEYSIZE.
У вас будет несколько наименьших значений KEYSIZE. У вас есть два варианта: попробовать все наименьшие значения или взять 4 (6, 8, ...) KEYSIZE блоков вместо 2-ух и усреднить полученные значения.

Шаг 2. Теперь вы знаете длину ключа. Найдите ключ.

Разбейте шифротекст на блоки длины KEYSIZE.
Транспонируйте блоки: сделайте новый блок, состоящий из 1-ого байта каждого старого блока, второй новый блок состоящий из 2-ого байта каждого старого блока и так далее.
Взломайте каждый новый блок как single-byte XOR. У вас уже есть код для этого!
Для каждого блока выберите однобайтовый ключ, у которого частотное распределение открытого текста наиболее близко к английскому тексту.
Соберите однобайтовые ключи вместе (ключ первого блока - первый байт искомого ключа, ключ второго блока - второй байт искомого ключа и так далее) и вы получите искомый ключ!

Эта задача носит скорее образовательный характер (вряд ли вы встретите repeating-key XOR в реальной жизни), но концепции, которые в ней применяются, помогут вам в понимании и поиске реальных криптографических уязвимостей.

Многие из специалистов по информационной безопасности знают как взломать repeating-key XOR, но мало кто может взломать его на практике. Вы - можете.

Задание взято из http://cryptopals.com/sets/1/challenges/

Выполнил: AndreiM