Математичне підґрунтя

Оскільки для шифруваняя повідомлень (MSG1, MSG2) був використаний однаковий ключ (KEY), то можна скористатися тим, що:
(MSG1 xor KEY) xor (MSG2 xor KEY) = MSG1 xor MSG2 xor KEY xor KEY = MSG1 xor MSG2, оскільки *KEY xor KEY* скорочується.

Завдяки цій властивості можна використати метод що називається “Crib dragging”. Вона дозволяє отрмати частину повідомлення завдяки використанню так званих *crib* слів. Переважно це короткі, часто вживані слова, котрі з великою ймовірністю присутні слова.

MSG1 xor MSG2 xor CRIB, якщо проводити цю операцію ітераційно зміщуючи позицію можна помітити що результат цієї функції може бути схожий на осмислений текск. Це буде символізувати про те, що наша CRIB строка є частиною одного з повідомлень. Шляхом підбору продовження для *crib* фрази можна розшифрувати обидва повідомлення навіть не знаючи ключа.

Демонстрація та опис реалізації

Розглянемо приклад роботи програми, та по ходу розберемо принцип її роботи.

Для демонстрації візьмемо перші дві лінії з даного шифертексту, кожна лінія якого зашифрована однаковим ключем:

```
ad924af7a9cdaf3a1bb0c3fe1a20a3f367d82b0f05f8e75643ba688ea2ce8ec88f4762fbe93b50bf5138c7b699
a59a0eaeb4d1fc325ab797b31425e6bc66d36e5b18efe8060cb32edeaad68180db4979ede43856a24c7d
a59a0eaeaad7fc3c56fe82fd1f6bb5a769c43a0f0cfae74f0df56fdae3db8d9d840875ecae2557bf563fcea2
a59a0eaea8ddf93c08fe81e11e2ab2bb6d962f0f1af2f44243b46cc1b6d6c291995d65a9a5234aa204
ad924af7a9cdaf3a1bb0c3f51439a5b628cf215a1fbdee4302a77a8ea2cc86c8984d65ffac6c58bf5b71dab8841136
b09b4afda3caf93c5aa78ce6096bb2a67ad86e4302f3e10602b37acbb1829680935137e8bb2919b6503fccfdca5461
a59a0eaeb5d7af3115b287b31425e6a460d3200f19f5e35406f567dde3cc8d9c9e4179eee92557f1463edc
a18c09ebb6ccaf2d12bbc3c41227aaf37fde274c05bdf5471aa62edaac82968093452da9eb0456bd5b71c6bfcb56

ad924af7a9cdaf3a1bb0c3e71a27adf37fdf3a474dfef44914b17d8ea2cc86c89d4d72f9e93556a44d71dfb8980034b3cea5c4d4
ab864af9a7d4e4790db797fb5b00afbd6fc5acaff9f3e95443b961dda6829680930874e6a42156bf1f25c6a4891c6d
ad924ae0a3d1fb311facc3f5142eb5f366d93c0f01f2f04f0db22ec8b1cb8786925b37eaa82219b94a23ddf1931b34fa
ad924aefaad4af341fb0c3f0143ea8a728c1275b05bdff4916f92eccb6d6c286994672a9bd2356f15224cab9d1
ad924af7a9cdaf3a1bb0c3f51227aaf37cde2b0f18f3e04911b267d8aacc85c89b4179fcbd29
b39d1ee6e6cbe6210ea7c3e01e28a9bd6cc5690f1af2f4520bf561c8e3c68b9b824979eaac6c4ba4517d89f1ca
bd9b1ffcb598e62a5aaa8bf65b0ea7a17cde6e4e03f9a64315b07cd7b7ca8b86910863e1a8381ea21f38c7f183006df6c2a5
a59a0e6c462cf83113bd8bb31238e6be67c42bcded09ff4916f262c2e3c087c897085ae8a76019bc4671dabe8455
```

1) Програма запитує користувача чи хоче він використати демонстраційні шифртексти, обираємо “так”. Якщо обрати “ні”, то можна ввести свої шифртексти.

```
Do you want to use sample ciphertext messages? (y/n):
Ciphertexts:
        ad924af7a9cdaf3a1bb0c3fe1a20a3f367d82b0f05f8e75643ba688ea2ce8ec88f4762fbe93b50bf5138c7b699
        a59a0eaeb4d1fc325ab797b31425e6bc66d36e5b18efe8060cb32edeaad68180db4979ede43856a24c7d
Do you want to skip not promising results? (y/n):
Do you want to check common crib words? (y/n):
```

2) Програма запитує чи пропускати результати, що не подають надію. Тобто такі які містять символи, що не властиві середньостатистичному тексту. Обираємо “yes”.
Функція що визначає те, чи подає текст надію:

```
def is_promising(text):
    promising_charset = string.ascii_letters + ' .,?! :;\'"'
    for char in text:
        if char not in promising_charset:
            return False
    return True
```

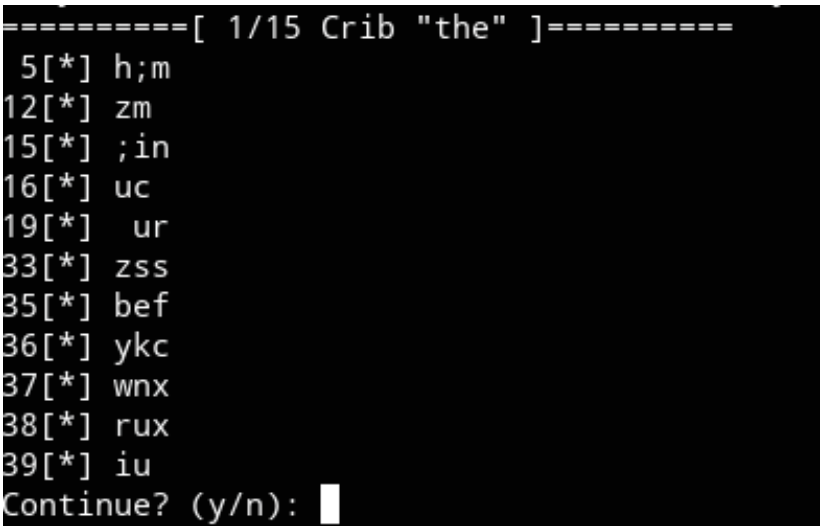
3) Програма запитує чи потрібно спочатку перебрати росповсюдженні *crib* слова. Обираємо “yes”.

Функція, що перебирає популярні *crib* слова.

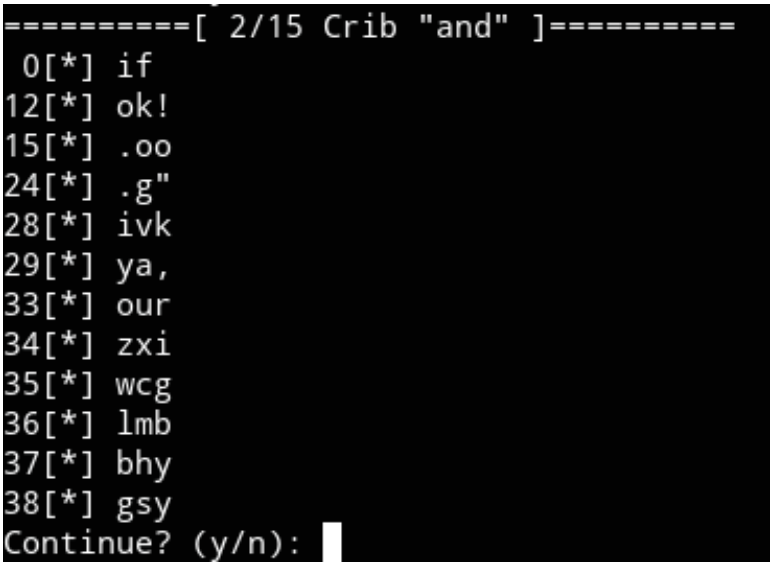
```
def print_common_cribs(xored_bts, skip_not_promising=False):
    crib_words = (
        'the', 'and', 'that', 'have',
        'for', 'not', 'with', 'you',
        'this', 'from', 'they', 'say',
        'her', 'she', 'will',
        )
    for i, crib in enumerate(crib_words):
        print('==========[ {}/{} Crib "{}" ]==========.format(
            i + 1, len(crib_words), crib))
        crib_bts = bytes(crib, encoding='ascii')
        print_crib_result(xored_bts, crib_bts, skip_not_promising)

        if not yes_no('Continue?'):
            break
```

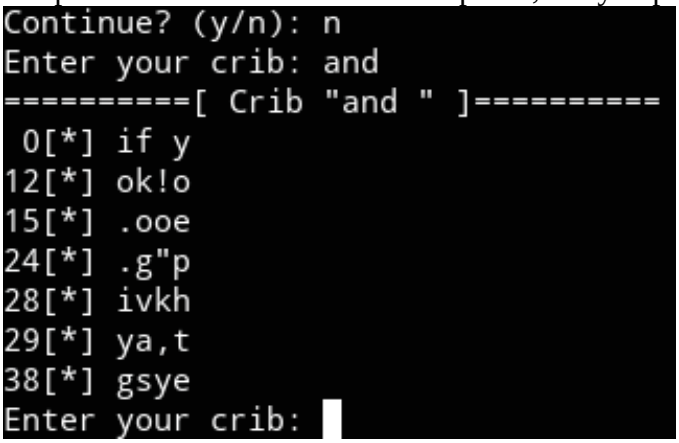
Починаємо пербір:



Не бачимо англійських слів, або частин слів. Тому продовжуємо.



Помітно, що при зміщенні 0 є слова **if**. Тому зупинимо пербір зі *crib* словом “and”.
Скоріш за все після слова **and** є пробіл, тому спробуємо таку строку.



Бачимо що при зміщенні 0 є текст що схожий на англійський. Спробуємо продовжити фразу до “if you “ і це буде нашим майбутнім *crib* словом.

```
Enter your crib: if you
==========[ Crib "if you " ]==========
 0[*] and ris
 4[*] tzsq.rt
Enter your crib:
```

Продовжуючи цей алгоритм далі можна розшифрувати задані повідомлення.
Пропустимо майбутні кроки роботи з програмою та перейдемо одразу до кінця. Весь вивід програми можна знайти у файлі ***log.txt***

```
==========[ Crib "you can make one heap of all your winn" ]==========
 3[*]  risk it on one turn of pitch-and-toss
```

Завдяки гуглу можна з легкістю знайти автора цієї фрази.