Assignment 0x03

Frank Kaiser – 1742945, Jan Martin – 1796943

December 10, 2017

Contents

1 A Ciphertext Only Attack on the Vigenre Cipher **2**

2 Loss of Confidentiality in Counter Mode with Repeated Nonce **3**

3 Loss of Confidentiality in CBC Mode Using a Padding Oracle **5**

1 A Ciphertext Only Attack on the Vigenre Cipher

First we used the Kaisiki-Test do determine the Key length:

letters	distance	prime factors
KC:	41	41
CM:	17	17
SY:	66	2 * 3 * 11
XOC:	140	2 * 2 * 5 * 7
OC:	20	2 * 2 * 5
GK:	12	2 * 2 * 3
JO:	12	2 * 2 * 3
JO:	42	2 * 3 * 7
JO:	16	2 * 2 * 2 * 2
JO:	74	2 * 37
JO:	136	2*2*2*17
LZKMP	520	2*2*2*5*13
LZKMP	178	2 * 89
LZKMP	366	2*3*61
LKZMP	1448	2*2*2*181

Because of the overwhelming amount of twos we decided to try 4 as key length. Then we did a frequency analysis on the partitions created by `https://cryptotools.psi.h4q.it/vigenere.html`. The most frequent letters for each partition were:

1) P, E, L, Z, S
2) S, H, B, C, O
3) G, V, Q, C, J, K
4) O, D, K, Y, C

After trying a little bit around, we discovered that the word L O C K is readable when taking one character from every partition. We tried it out and had success. The key for the text is LOCK and Alice finds a golden key!

2 Loss of Confidentiality in Counter Mode with Repeated Nonce

When xoring two or more plaintexts with the same key it is possible two reproduce the plaintext by xor-ing two messages and trying common words. For English language words like "the" or "you" are very common and a good starting point.

Here we demonstrate the start of the process for Appendix B 1 and 2:

```
XOR 1/2:
03 01 11 52 3a 32 37 49 03 08 44 11 49 58 1d 15 00 03 01 11 48 4e 0c 01 16 0b 54 0c 01 00 1f 0a 14 00 41 1d 16
44 42 4f 06 05 04 4c 0d 0d 54 0b 16 18 11 52 46 0a 0f 0b 0c 4f 1b 02 15 45 00 16 45 59 06 13 00 10 18 4f 4e 02
11 53 07 00 0c 00 53 41 48 07 04 00 53 1c 48 0b 4f 1d 02 08 45 46 1f 00 45 03 1f 1d 5c


"the"
** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** |
 1 2  3   4  5   6  7   8  9 10   11 12 13 14 15   16 17 18 19 20   21 22 23 24 25   26 27 28 29 30   31 32
23 21 31 72 1a | 12 17 69 23 28 | 64 31 69 78 3d | 35 20 77 69 74 | 68 6e 2c 21 36 | 2b 74 2c 21 74 | 77 6f
                                                    W I   T    H                            t    w  o

"with"

** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** |
 1 2  3   4  5   6  7   8  9 10   11 12 13 14 15   16 17 18 19 20   21 22 23 24 25   26 27 28 29 30   31 32
74 68 65 3a 1a | 12 17 69 23 28 | 64 31 69 78 3d | 35 20 74 68 65 | 20 6e 2c 21 36 | 2b 74 2c 21 74 | 77 6f
 T  H  E                                           T  H   E   WS

"nonce"


** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** |
 1 2  3   4  5   6  7   8  9 10   11 12 13 14 15   16 17 18 19 20   21 22 23 24 25   26 27 28 29 30   31 32
23 21 31 72 1a | 12 17 69 23 28 | 64 31 69 78 3d | 35 20 23 21 31 | 68 20 63 6f 75 | 6e 74 2c 21 74 | 77 6f
                                                       WS C  O  U    N   T

double whitespace is unusual, that's why WITH WS COUNT is probably in one text

"count"

** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** | ** ** ** ** ** |
 1 2  3   4  5   6  7   8  9 10   11 12 13 14 15   16 17 18 19 20   21 22 23 24 25   26 27 28 29 30   31 32
74 68 65 3a 1a | 12 17 69 23 28 | 64 31 69 78 3d | 35 20 74 68 65 | 20 6e 6f 6e 63 | 65 20 2c 21 74 | 77 6f
                                                    T  H   E   WS N  O  N  C |  E  WS
```

When we found all words, we got the sentence:
"Encrypting texts with counter mode is normally just fine, unless you do not take a fre" for chiphre 2.

By xoring that with the original chiphre text we got the key:

c9 3e 1e ba b7 3f c3 3c 05 a6 75 35 41 bd 34 5a e4 57 d1 a3 74 16 43 4e 02 3a 0b 0e d2 2b 9e 2c 32 c5 2a bb 75 5c 0b bf 54 29 2f 8e 63 53 b7 f3 45 b3 a5 c4 ee dd 4c 9a 03 da 4e 2b 2e 15 6b 80 7b ca 28 4c 9b 16 24 21 9b 1e 94 8f 51 59 d6 fb 5f 31 cb 06 f8 cc

Using this key, we can encrypt all messages by xoring the message with the key:

The PSI working Group congratulates you for solving this exercise. It was easy, right?
Encrypting texts with counter mode is normally just fine, unless you do not

take a fre
For CBC mode, if the nonce is repeated, the attacker can only see if two messages have
No one should evEr use unauthenticated encryption, e.g. CTR or CBC without MAC, unless
If you authenticAte your ciphertext with a MAC, use the Encrypt then MAC construction
Given reasonable assumptions, the Encrypt then MAC construction was proven to be secur
The assumptions For the proof of security of Encrypt then MAC are a strongly unforgabl
There are cipher modes, which authenticate and encrypt in one step, without having to
One of the cipheR modes, which authenticate while encrypting is Galois Counter Mode, w
Galois Counter MOde (GCM) is very vulnerable to repeated nonces. You can even recover
Newer stream cipHers, e.g. NORX, use a duplex sponge construction to encrypt and authe

Suppose you were allowed to ask an oracle to encrypt any plaintext you like with the same nonce and key as used for the ciphertexts. For which plaintext would you like to obtain the ciphertext to get the keystream immediately? We would ask the oracle to encrypt one of the ciphertexts. This way we would receive the unencrypted plaintext and by xoring that plaintext with the corresponding ciphretext we would receive the key.

3 Loss of Confidentiality in CBC Mode Using a Padding Oracle

The ciphretext on the link `https://paddingoracle.psi.h4q.it/`is:
bd4b7c0a62018807bec676f83c0685cc-92b2e23fbc24d12a8f726becc58ee7fb-307de7ccb0127d7d772e7b4f38edd348-14eeb43229dd60c1a4168862ba42a652-56565e8b92fdc36c962ca4ba1d977d8e

Unfortunately time exceeded here..