Introduction to Cryptology (GBIN8U16)
TP — Birthday attack on CBC

2019-04

Grading

This TP is graded as the *contrôle continu* of this course. You must send a written report (in a portable format) **detailing** your answers to the questions, and the corresponding source code, *including all tests*, **with compilation and execution instructions** by the end of the month, (2019-04-30T23:59+0200) to:

pierre.karpman@univ-grenoble-alpes.fr.

Working in teams of two is allowed and encouraged (but not mandatory), in which case only one report needs to be sent, with the name of both students clearly mentioned.

Part one: preparatory work

Download the tarball at https://www-ljk.imag.fr/membres/Pierre.Karpman/cry_intro2018_tp.tar.bz2. This tarball contains the files `tczero.c` and `tczero.h` that define TCZero, which is a toy block cipher with variable block size.

Question 1

Implement a function

```
size_t cbc_enc(uint64_t key[2], uint8_t *pt, uint8_t *ct, size_t ptlen)
```

that encrypts a `ptlen`-long plaintext with TCZero with the currently-defined block size. You may assume that the plaintext length is always a multiple of the block size, and you are not required to implement any padding.

Be careful in how you generate the IVs.

Write a small test function that verifies that the encryption is non-deterministic.

Question 2

Implement a function

```
size_t cbc_dec(uint64_t key[2], uint8_t *ct, uint8_t *pt, size_t ctlen)
```

Write a small test function that verifies that an encrypted plaintext is properly decrypted to itself.

Part two: the attack

The goal is now to implement the birthday attack on CBC mode.

Question 1

Recall the attack and its data complexity.

Question 2

Implement the attack when setting the block size of TCZero to 32 bits by recovering the XOR of two plaintext blocks from the ciphertexts only, that is implement a function

```
uint64_t attack(uint8_t *ct, size_t ctlen)
```

What kind of data structure can you use to find colliding ciphertext blocks efficiently? How much data do you need on average for the attack to be successful in practice? Is this consistant with the theoretical predictions?

Question 3

Same as the previous question, but with 50-bit blocks.

Question 4 (bonus)

Sames as the previous question, but with 64-bit blocks.

Question 5

Would it be feasible to implement this attack on a decent desktop computer for 80-bit blocks? For 128-bit blocks?