ALGORITHMICS, CRYPTOGRAPHY & COMPUTER SECURITY, graduate programs 2023
CRYPTOGRAPHY, 2022, assignments list # 6, 18.4.2023

1. Recall Vaudenay's padding attack ($\approx$ page 30 of the lecture notes). **Question:** Design a padding scheme, where the attack does not work anymore. Is it possible with any deterministic padding algorithm?

2. You have a textfile T. In order to store it in the cloud, you have to encrypt it. In order to save space, you apply a lossless compression scheme[1].

   option 1: store $\texttt{Compress}(\mathsf{Enc}_K(T))$. **Question 1:** Why is it a bad idea?

   option 2: store $\mathsf{Enc}_K(\texttt{Compress}(T))$, where by Enc_K we mean CBC with a standard block cipher such as AES. Assume that an eavesdropper does not know anything about T.
   Question 2: Is this situation more advantageous for the eavesdropper than in the case where an uncompressed file is transmitted: $\mathsf{Enc}_K(T)$.
   In order to answer this question, check the headers of compressed files.

3. Let us assume that option 2 of Exercise 2 is used to encrypt file T.

   Assume that the attacker can influence T by appending to it a padding of his choice (compare your solution for exercise 1!). The attack idea is to guess a suffix of T and repeat it in the padding. If the guess is correct, then the size of the compressed file does not grow as in the case when the choice is wrong.

   Question 1: construct an attack based on this idea. Consider encryption with a stream cipher (the plaintext is XOR-ed with a pseudorandom string generated from the key) and compression method with sliding window (**read a description of LZ77**).

   Question 2: Reconsider the situation for block encryption in CBC mode. Does it make the attack easier or harder?

   (*Comment*: after solving Exercises 2 and 3 you should see why data compression is frequently forbidden in newer browsers.)

4. Recall LRW encryption mode.

   question 1: What happens if the plaintext block I is equal to $-F \otimes I$?

   question 2: Given a ciphertext, is it possible to detect that all plaintext blocks are equal?

5. How to convert block encryption in blocks of length k into block encryption for blocks of length m, where $m > k$? (Example: we are given block encryption for block size of 256 bits, but the customer wants encryption for 500 bit blocks.)

6. Is it possible to design a secure Format-Preserving Encryption for 10-bit blocks?

7. We desgin a Format-Preserving Encryption on 64-bit blocks. For an input T:

   - run $\texttt{PRNG}(K)$ for an encryption key K and split its output into 6-bit blocks a_1, a_2, ...
   - perform k steps: in step i flip the bit a_i of T, that is, $T := \texttt{Flip}(a_i, T)$.

   Question: is $k = 32$ sufficient?

/-/ Mirosław Kutyłowski

[1] by uncompressing, we get exactly the original file.