Preamble

This sheet is split into two parts. Problems 1-6 may be attempted after the lesson of Tuesday 29/10/2019 and problems 7-8 should be attempted after the lesson of Thursday 31/10/2019.

Questions

Problem 1

Show that the CBC and OFB modes of encryption do not yield CCA-secure encryption schemes.

Problem 2

DES possesses a key-size of 56-bits and produces an encryption of size 64-bits. As it is now computationally feasible to break DES, someone suggests using double-DES (encrypting the message twice with separate keys) to encrypt messages with 112-bits of security. This appears to be feasible until someone from `https://www.zerodium.com/` contacts you claiming to be able to break the scheme with a *Meet In The Middle* attack. The attack works on *any* double-symmetric encryption scheme and will take 2^{k+1} operations and require 2^k encryptions which must be stored, where k is the size of the key. Assume that you have an oracle which can generate the required encryptions. Attempt to cost the attack based upon storage requirements alone. What about AES? (You may assume a cost of 0.01 per Gigabyte).

Problem 3

What is the output of an r-round Feistel network when the input is (L_0, R_0) in each of the following two cases:

(a) Each round function outputs all 0s, regardless of the input.

(b) Each round function is the identity function.

Problem 4

Having considered **Problem 2**, you decide that you require triple-Encryption. Let F be a block cipher with n-bit block length and key length and set $F'_{k_1,k_2}(x) := F_{f_1}(F^{-1}_{k_2}(F_{k_1}(x)))$.

(a) Assume that given a pair (m_1, m_2) it is possible to find in *constant* time all keys k_2 such that $m_2 = F^{-1}_{k_2}(m_1)$. Show how to recover the entire key for F' (with high probability) in time roughly 2^n using three known input/output pairs.

(b) In general, it will *not* be possible to find k_2 as above in constant time. However, show that by using a pre-processing step taking 2^n time it is possible, given m_2, to find in (essentially) constant time all keys k_2 such that $m_2 = F^{-1}_{k_2}(0^n)$.

(c) Assume k_1 is known and that the pre-processing step above has already been run. Show how to use a single pair (x, y) for a chosen input value x to determine k_2 in constant time.

(d) Put the above components together to devise an attack that recovers the entire key by running in roughly 2^n time and requesting the encryption of roughly 2^n chosen inputs.

Problem 5

Show that DES has the property that for every key k and input x it holds that $DES_k(x) = \overline{DES_{\bar{k}}(\bar{x})}$ (where $\bar{z}$ denotes the bitwise complement of z, or equivalently $\bar{z} = z \oplus 1^{|z|}$). This property is called the *complementary property* of DES.

Problem 6

How can the complimentary property (see previous question) of DES be used to find the secret key k in DES using a strict maximum of 2^{55} local computations of DES if we are allowed two queries to the oracle $\text{DES}_k(\cdot)$?

Problem 7

Say $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ is a secure MAC, and for $k \in \{0, 1^n\}$ the tag-generation algorithm Mac_k always outputs tags of length $t(n)$. Prove that if $t(n) = O(\log n)$ then Π cannot be a secure MAC.

Problem 8

Consider the following fixed-length MAC for messages of length $l(n) = 2n - 2$ using a pseudorandom function F:

On input message $m_0||m_1$ with $|m_1| = |m_2| = n - 1$ and key $k \in \{0, 1\}^n$, an algorithm Mac_k outputs the tag $t = F_k(0||m_0)||F_k(1||m_1)$. The algorithm Vrfy is performed in the usual manner of computing $t' = \mathsf{Mac}_k(m)$ and comparing it to the provided tag t.

Is $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vrfy})$ existentially unforgeable under a chosen-message attack? Prove your answer.