Problem 1 Encryption and MACs

The following three methods for combining IND-CPA private-key encryption and secure message authentications codes have appeared in internet standards.

1. Encrypt-and-MAC$_{k_1, k_2} (m)$ : Output $(\text{Enc}_{k_1} (m), \text{Tag}_{k_2} (m))$

2. MAC-then-Encrypt$_{k_1, k_2} (m)$: Output $\text{Enc}_{k_1} (m | \text{Tag}_{k_2} (m))$

3. Encrypt-then-MAC$_{k_1, k_2} (m)$: Compute $c \leftarrow \text{Enc}_{k_1} (m), t \leftarrow \text{Tag}_{k_2} (c)$, output $(c, t)$

Here $(\text{Gen}, \text{Enc}, \text{Dec})$ is an IND-CPA private-key encryption scheme and $(\text{G}, \text{Tag}, \text{Ver})$ is a secure message authentication code and $k_1 \leftarrow \text{Gen}(1^n)$ and $k_2 \leftarrow \text{G}(1^n)$. The decryption functions are the natural ones and have been omitted.

Assuming only properties guaranteed by the definition of IND-CPA security and secure MACs, explain whether each proposal is always a CCA2-secure encryption scheme. If not, present a particular instantiation of the encryption or MAC scheme which satisfies the stated definitions, but allows an adversary to violate the CCA2 security of the proposal.

Problem 2 Oblivious Transfer

In the honest-but-curious oblivious transfer functionality discussed in class, the sender has two bits $m_0$ and $m_1$, and the receiver has a bit $b$. At the end of the protocol, the receiver is able to compute $m_b$ (but learn nothing else about $m_{1-b}$). Meanwhile, the sender learns absolutely nothing about the receiver’s choice $b$.

Imagine that there is a physical device that implements 1-out-of-2 oblivious transfer for malicious adversaries. Show how to use this physical mechanism to achieve 1-out-of-3 oblivious transfer against malicious adversaries. Hint: you might need to use the 1-out-of-2 device more than once.