1. In class, we showed how to build a stream cipher from a pseudo-random generator (PRG): the key for the encryption scheme is a random seed for the PRG, and encryption works by stretching the seed using the PRG, and taking the exclusive-or of this with the message. We proved that this encryption scheme is semantically secure if the PRG is secure. Prove the converse: the PRG is secure if this encryption scheme is semantically secure.

2. In class, we discussed Internet Roulette. Suppose now that instead of paying just 0 or 1 dollars, the house pays out 0, 1, . . . , n dollars. Here, n is bounded, and the probability of paying out i dollars, for i = 0, . . . , n, is p_i. Just as in class, the random value r is encrypted using a semantically secure encryption scheme, and the player sees the corresponding ciphertext before placing his bet. Show that the expected value of the amount of money the house pays out is negligibly close to the amount of money it would pay out in a version of the game in which the player does not see the ciphertext.

3. In class, we showed that if one could effectively distinguish a random bit string from a pseudo-random bit string, then one could succeed in predicting the next bit of a pseudo-random bit string with probability significantly greater than 1/2 (where the position of the “next bit” was chosen at random). Generalize this from bit strings to strings over the alphabet \{0, . . . , n−1\}, for any n ≥ 2, assuming that n is bounded. Hint: first generalize the distinguisher/predictor lemma discussed in class.

4. This exercise develops an alternative characterization of semantic security. Let \(E = (E, D)\) be a cipher defined over \((K, M, C)\). Also, assume that one can efficiently generate random messages. We define an attack game between an adversary \(A\) and a challenger as follows. The adversary selects a message \(m \in M\) and sends \(m\) to the challenger. The challenger then computes:

\[b \leftarrow_R \{0, 1\}, \ k \leftarrow_R K, \ m_0 \leftarrow m, \ m_1 \leftarrow_R M, \ c \leftarrow_R E(k, m_b),\]

and sends the ciphertext \(c\) to \(A\), who then computes and outputs a bit \(\hat{b}\). That is, the challenger encrypts either \(m\) or a random message, depending on \(b\). We define \(A\)'s advantage to be \(|\Pr[\hat{b} = b] - 1/2|\), and we say the \(E\) is real/random secure if this advantage is negligible for all efficient adversaries.

Show that \(E\) is semantically secure if and only if it is real/random secure.

5. In the text, at the end of §4.1.1, the following algorithm is presented for generating a random element \(y \in \mathcal{X} \setminus \{y_1, \ldots, y_{i-1}\}\):

\[\text{repeat } y \leftarrow_R \mathcal{X} \text{ until } y \notin \{y_1, \ldots, y_{i-1}\}\]

Assuming this algorithm is used to implement the challenger in Experiment 1 of the Block Cipher Attack Game (i.e., Attack Game 4.1 in the text), estimate the total amount of time that is needed for the challenger to process \(q\) queries. Your estimate should be expressed as a function of \(q\) and \(|\mathcal{X}|\).
6. Because DES has such a short key, one might attempt to strengthen it as follows. Let $\mathcal{E} = (E, D)$ denote the DES encryption scheme, defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$, where $\mathcal{K} = \{0, 1\}^{56}$, and $\mathcal{M} = \mathcal{C} = \{0, 1\}^{64}$. “Double DES” is the cipher $\mathcal{E}' = (E', D')$, defined over $(\mathcal{K} \times \mathcal{K}, \mathcal{M}, \mathcal{C})$, where $E'((k_1, k_2), m) := E(k_2, E(k_1, m))$ and $D'((k_1, k_2), c) := D(k_1, D(k_2, c))$.

Now suppose we model $\mathcal{E}$ as an ideal cipher. Further suppose that you are given a few plaintext/ciphertext pairs, encrypted using $\mathcal{E}'$ using a key $(k_1, k_2)$. Show how to recover the key $(k_1, k_2)$ in time $\approx 2^{56}$. Your algorithm may use space proportional to $2^{56}$.

Now show how to modify your algorithm so that it uses less space, at the expense of using more time: it should recover the key in time $\approx 2^{56+t}$, but use space $\approx 2^{56-t}$, for any given value of the parameter $t$.

This exercise shows that although “Double DES” doubles the key length, it does not really double the security, as one might expect.