Problem 11. Given a “good” blockcipher $E: \{0,1\}^{56} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$, define from it a blockcipher $E: \{0,1\}^{112} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ by way of $E(K_1K_2, X) = E(K_2, E(K_1, X))$ where $|K_1| = |K_2| = 56$.

Suppose an adversary $A$ knows a couple plaintext/ciphertext pairs: $(M, C)$ and $(M', C')$ where $C = E(K, M)$ and $C' = E(K, M')$. Describe and analyze an algorithm that finds a key $K$ consistent with these plaintext/ciphertext pairs. How long does your algorithm take to run? It should run in much less than $2^{112}$ time (where you should count one unit of time to compute $E$ or its inverse, and ignore anything else).

Problem 12.

Part A. Alice designs a blockcipher $E$ that has one little defect: the first bit of $E_K(X)$ doesn’t depend on the last bit of $X$. Is it possible that $E$ is, nonetheless, secure as a PRP? Briefly explain.

Part B. Alice designs a blockcipher $E$ that has one little defect: the first bit of $E_K(X)$ doesn’t depend on the last bit of $K$. Is it possible that $E$ is, nonetheless, secure as a PRP? Briefly explain.

Problem 13. Given a blockcipher $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ (e.g., $n = 128$) construct from it a blockcipher $E: \{0,1\}^{k\ell} \times \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ that is plausibly PRP-secure as long as $E$ is. You can make $\ell \geq 1$ whatever small number you need. (You do not have to prove that $E$ is PRP-secure when $E$ is, but if the TA can see that this isn’t, or if your construction is not clear or is extremely inefficient, then that’s a problem.)

Problem 14. Fix a blockcipher $E$ with an 8-byte (64-bit) blocksize. Consider the following generalization of CBC to allow the encryption of arbitrary byte strings. Given a byte string $M$, let $\text{pad}(M)$ be $M$ followed by enough bytes to take you to the next multiple eight bytes, where the extra bytes are either: 01, or 02 02, or 03 03 03, and so on, up to 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 08 (all of these constants written in hexadecimal). Let CBCX be the variant of CBC8 encryption that encrypts $M$ by applying CBC, over $E$, with a random $IV$, to $\text{pad}(M)$.

The CBCX method is specified in Internet Standard RFC 2040. Note that a CBCX ciphertext for $M$ will have the form $C = IV || C'$ where $|IV| = 64$ and $|C'|$ is the least multiple of 64 exceeding $|M|$.

Part A. Do you believe that CBCX achieves “good” (at least birthday-bound) ind$^*$-security when $E$ is a good PRP? Why or why not?

Part B. Write a careful fragment of pseudocode for an algorithm $D$ to decrypt a byte string $C$ under CBCX. Have $D(K, C)$ return the distinguished symbol $\bot$ if it is provided an invalid ciphertext; otherwise, it returns a byte string $M$.

Part C. Suppose an adversary is given an oracle, Valid, that, given a ciphertext $C$, returns the bit “1” if $C$ is valid, meaning $D(K, C) \in \{0,1\}^*$, and returns the bit “0” if it is not, meaning $D(K, C) = \bot$. Show how to use the oracle to decipher a block $Y = E_K(X)$ for an arbitrary eight-byte $X$. (Hint: all your queries to the Valid oracle will be 16 bytes, and I don’t mind if you make hundreds or thousands of them.)

Part D. Show how to decrypt any ciphertext $C = \text{CBCX}(K, M)$ given a Valid oracle.

Part E. What advice would you give to security practitioners who were considering the use of CBCX in their networking protocol?