Homework 6 – Due Tuesday, March 20

1. Consider a cipher that has the following property: When you view the plaintext bits and ciphertext bits as binary numbers, the cipher preserves the evenness of the number. In other words, if the plaintext is an even number then the ciphertext will also be an even number, and if the plaintext is odd then the ciphertext will also be odd.

   (a) Prove that such a cipher cannot be IND-CPA secure.

   (b) Joe decides to "fix" the cipher by doing the following: he adds one additional bit to the key, and then adds this (as a number) to the ciphertext after encryption. If this key bit is chosen randomly, then half the time even numbers map to even numbers, and the other half of the time even numbers map to odd numbers. Since this is random and unpredictable if the bit isn't known, he thinks this fixes the problem. However, it is still not IND-CPA secure — prove this.

2. Give the RSA algorithm, including descriptions of how keys are generated and how encryption and decryption work. Use formulas, and describe the size of values (i.e., number of bits) used in a typical real system. Explain why decryption is the inverse if encryption (i.e., what is the mathematics that explains why $M = D(PR_a, E(PU_a, M))$ for all messages M — you can give a simplified explanation, in which you assume that M is relative prime to the modulus n).

3. Joe makes a system using RSA for public key cryptography, but in the key generation routine uses the standard C library `rand()` function for random values, which uses a 15-bit value for a seed. Why is this insecure? Be very specific in your answer, describing a real, practical attack with an explanation of the time complexity of the attack.