Math 445
Introduction to Cryptography
First Exam Solutions

1. (10) State Kerckhoff's principle. Explain briefly why a cryptosystem designed by someone who follows this principle is likely to be stronger than one designed by someone who does not.
Solution: Kerckhoff's principle says that one should always assume that the attacker knows the algorithm being used. A designer who believes this principle will circulate his or her algorithm widely and it will be tested by many talented cryptanalysts. A designer who does not believe the principle may try to keep the algorithm secret. Thus the algorithm will only be tested by a few cryptanalysts and we can be much less confident of its strength.

2. (5) What is the main drawback of the one time pad cryptosystem?
Solution: The one time pad system requires that we secretly communicate in advance a key which is at least as long as the message we will send. This is a severe practical difficulty since it requires substantial secret communication in advance of the desired secret communication.

3. (10) Assuming you can do 2^{20} encryptions per second and the key size is 40 bits, how long would a brute force attack take? Give a scenario where this would be practical and another where it wouldn't. What happens if you double the key size?
Solution: The key space has 2^{40} elements, so brute force would take 2^{20} seconds, which is about 12 days. This would be practical if the message revealed the location of enemy missles in a cold-war situation. It would be impractical if the message's useful life was very short, for example if it was a few frames in a pay-per-view sports video. Doubling the key size would make the brute force decryption time 2^{60} seconds, which is about 3.8×10^{16} years. There is no scenario in which this would be practical.

4. (15) You have intercepted a message encrytped with an affine cipher. The ciphertext starts with $BBDJ$ and you know the plaintext starts with *oops*. Find the key.
Solution: An affine cipher has the form $y \equiv ax + b$ where x is the plaintext and y is the ciphertext (both integers modulo 26). We need to find a and b. Converting to numbers, the plaintext is 14, 14, 15, 18 and the ciphertext is 1, 1, 3, 9. Thus we need to solve the equations $14a + b = 1$ and $15a + b = 3$. Substracting the equations,we find $a = 2$ and plugging this into either equation gives $b = 25$.

Unfortunately, the encryption key is not a 1-1 transformation, so if you try to find the decryption key you may get stuck. Full credit was given if you did something reasonable.

5. (10) Consider a language with three letters, a, b, and c, with frequencies .6, .3, and .1. Suppose that a long message (1000 characters) in this language is encrypted with a Vignère cipher and we plan to break it using a index of coincidence attack. About how big is the largest index of coincidence we are likely to see?
Solution: If we shift by a multiple of the key length, the probability of coincidence is $(.6)^2 + (.3)^2 + (.1)^2 = .46$. So we would expect about $1000 * .46 = 460$ coincidences. Other shifts would give lower indices of coincidence.

6. (15) Use the extended Euclidean algorithm to compute the greatest common divisor d of 654 and 123 and to find integers m and n such that $654m + 123n = d$.

Solution: We have

$$\begin{aligned} 654 &= 5 \cdot 123 + 39 \\ 123 &= 3 \cdot 39 + 6 \\ 39 &= 6 \cdot 6 + 3 \\ 6 &= 2 \cdot 3 + 0 \end{aligned}$$

and so the gcd $d = 3$. Working backwards, we have

$$\begin{aligned} 3 &= 39 - 6 \cdot 6 \\ &= 39 - 6(123 - 3 \cdot 39) = 19 \cdot 39 - 6 \cdot 123 \\ &= 19(654 - 5 \cdot 123) - 6 \cdot 123 = 19 \cdot 654 - 101 \cdot 123 \end{aligned}$$

and so $m = 19$ and $n = -101$.

7. (10) Recall that we can represent the field of 256 elements as polynomials $\mathbf{F}_2[X]$ modulo $X^8 + X^4 + X^3 + X + 1$. Find the inverse of X in this field.
Solution: We start to use the Euclidean algorithm:

$$X^8 + X^4 + X^3 + X + 1 = (X^7 + X^3 + X^2 + 1) \cdot X + 1$$

and after 1 step we are done: this equation says that $X \cdot (X^7 + X^3 + X^2 + 1) \equiv 1 \pmod{X^8 + X^4 + X^3 + X + 1}$ and so the inverse of X is $X^7 + X^3 + X^2 + 1$.

8. (10) Recall that in a Feistel system, we divide the state into left and right halves $L_i R_i$ and then define the new state by $L_{i+1} = R_i$ and $R_{i+1} = L_i \oplus f(K_i, R_i)$ where K_i is the key for the i-th round and f is a function of the key and half of the state. Prove that no matter what the function f is, the round transformation is 1-to-1, i.e., we can recover the old state from the new state and the key.
Solution: We just solve the equations for L_i and R_i:

$$\begin{aligned} R_i &= L_{i+1} \\ L_i &= R_{i+1} \oplus f(K_i, R_i) = R_{i+1} \oplus f(K_i, L_{i+1}) \end{aligned}$$

This shows that we can solve for L_i and R_i if we know L_{i+1}, R_{i+1} and K_i.

9. (15) Briefly describe the Shift Rows and Byte Substitution layers of Rijndael. Explain why we can apply them in either order with the same result.
Solution: The state in Rijndael is a 4×4 matrix with entries in the field of 256 elements. The shift row layer shifts each row to the right a certian amount, wrapping the entries around. More precisely, the first row is not shifted, the second row is shifted by one, the third row is shifted by two, and the fourth row is shifted by three.

The Byte Substitution layer can be viewed as a lookup table. Each matrix entry, represented by an 8-bit byte, is broken into two pieces which index the rows and columns of a 16×16 lookup matrix. The byte is replaced by the corresponding entry in the table, which is another 8-bit byte.

The Byte Substitution layer is applied entry by entry to the state, with all entries treated in the same way. The Row Shift layer simply moves the bytes around. Thus it doesn't matter in which order we apply these layers: shifting and substituting is the same as first substituting then shifting.