Problems

3.1 a. In Section 3.1, under the subsection on the motivation for the Feistel cipher structure, it was stated that, for a block of $n$ bits, the number of different reversible mappings for the ideal block cipher is $2^n$. Justify.

b. In that same discussion, it was stated that for the ideal block cipher, which allows all possible reversible mappings, the size of the key is $n \times 2^n$ bits. But, if there are $2^n$ possible mappings, it should take $\log_2 2^n!$ bits to discriminate among the different mappings, and so the key length should be $\log_2 2^n!$. However, $\log_2 2^n! < n \times 2^n$. Explain the discrepancy.

3.2 Consider a Feistel cipher composed of 16 rounds with block length 128 bits and key length 128 bits. Suppose that, for a given $k$, the key scheduling algorithm determines values for the first 8 round keys, $k_1, k_2, \ldots, k_8$, and then sets

$$k_9 = k_8, k_{10} = k_7, k_{11} = k_6, \ldots, k_{16} = k_1$$

Suppose you have a ciphertext $c$. Explain how, with access to an encryption oracle, you can decrypt $c$ and determine $m$ using just a single oracle query. This shows that such a cipher is vulnerable to a chosen plaintext attack. (An encryption oracle can be thought of as a device that, when given a plaintext, returns the corresponding ciphertext. The internal details of the device are not known to you and you cannot break open the device. You can only gain information from the oracle by making queries to it and observing its responses.)

3.3 Consider a block encryption algorithm that encrypts blocks of length $n$, and let $N = 2^n$. Say we have $i$ plaintext-ciphertext pairs $P_i, C_i = E(K, P_i)$, where we assume that the key $K$ selects one of the $N!$ possible mappings. Imagine that we wish to find $K$ by exhaustive search. We could generate key $K$ and test whether $C_i = E(K', P_i)$ for $1 \neq i \neq t$. If $K'$ encrypts each $P_i$ to its proper $C_i$, then we have evidence that $K = K'$. However, it may be the case that the mappings $E(K, \cdot)$ and $E(K', \cdot)$ exactly agree on the $r$ plaintext-ciphertext pairs $P_i, C_i$ and agree on no other pairs.

a. What is the probability that $E(K, \cdot)$ and $E(K', \cdot)$ are in fact distinct mappings?

b. What is the probability that $E(K, \cdot)$ and $E(K', \cdot)$ agree on another $r'$ plaintext-ciphertext pairs where $0 \neq r' \neq N - r$?

3.4 Let $\pi$ be a permutation of the integers $0, 1, 2, \ldots, (2^n - 1)$, such that $\pi(m)$ gives the permuted value of $m$, $0 \neq m < 2^n$. Put another way, $\pi$ maps the set of $n$-bit integers into itself and no two integers map into the same integer. DES is such a permutation for 64-bit integers. We say that $\pi$ has a fixed point at $m$ if $\pi(m) = m$. That is, if $\pi$ is an encryption mapping, then a fixed point corresponds to a message that encrypts to itself. We are interested in the probability that $\pi$ has no fixed points. Show the somewhat unexpected result that over 60% of mappings will have at least one fixed point.
3.5 Consider the substitution defined by row 1 of S-box $S_1$ in Table 3.3. Show a block diagram similar to Figure 3.1 that corresponds to this substitution.

3.6 Compute the bits number 1, 16, 33, and 48 at the output of the first round of the DES decryption, assuming that the ciphertext block is composed of all ones and the external key is composed of all ones.

3.7 Suppose the DES $F$ function mapped every 32-bit input $R$, regardless of the value of the input $K_i$, to:
   a. 32-bit string of ones,
   b. bitwise complement of $R$.

   Hint: Use the following properties of the XOR operation:
   1. What function would DES then compute?
   2. What would the decryption look like?

   $$(A \otimes B) \otimes C = A \otimes (B \otimes C)$$
   $$A \otimes A = 0$$
   $$A \otimes 0 = A$$
   $$A \otimes 1 = \text{bitwise complement of } A$$

   where
   - $A$, $B$, $C$ are $n$-bit strings of bits
   - $0$ is an $n$-bit string of zeros
   - $1$ is an $n$-bit string of one

3.8 This problem provides a numerical example of encryption using a one-round version of DES. We start with the same bit pattern for the key $K$ and the plaintext, namely:

   in hexadecimal notation: $0123456789ABCDEF$
   in binary notation: $00000001001001110100101101101111$

   a. Derive $K_1$, the first-round subkey.
   b. Derive $L_0$, $R_0$.
   c. Expand $R_0$ to get $E[R_0]$, where $E[\cdot]$ is the expansion function of Figure 3.8.
   d. Calculate $A = E[R_0] \otimes K_1$.
   e. Group the 48-bit result of (d) into sets of 6 bits and evaluate the corresponding $S$-box substitutions.
   f. Concatenate the results of (e) to get a 32-bit result, $B$.
   g. Apply the permutation to get $P(B)$.
   h. Calculate $R_1 = P(B) \otimes L_0$.
   i. Write down the ciphertext.

3.9 Show that DES decryption is, in fact, the inverse of DES encryption.

3.10 The 32-bit swap after the sixteenth iteration of the DES algorithm is needed to make the encryption process invertible by simply running the ciphertext back through the algorithm with the key order reversed. This was demonstrated in Problem 3.7. However, it is still not entirely clear why the 32-bit swap is needed. To demonstrate why, solve the following exercises. First, some notation:

   $A|B$ = the concatenation of the bit strings $A$ and $B$

   $T_i(R|L)$ = the transformation defined by the $i$th iteration of the encryption algorithm, for $1 \neq i \neq 16$

   $TD_i(R|L)$ = the transformation defined by the $i$th iteration of the decryption algorithm, for $1 \neq i \neq 16$

   $T_{16}(R|L) = L|R$. This transformation occurs after the sixteenth iteration of the encryption algorithm.
a. Show that the composition \( T_D(\text{IP}^{-1}(T_{17}(\text{IP}^{-1}(L_{15}|R_{15})))) \) is equivalent to the transformation that interchanges the 32-bit halves, \( L_{15} \) and \( R_{15} \). That is, show that
\[
T_D(\text{IP}^{-1}(T_{17}(\text{IP}^{-1}(L_{15}|R_{15})))) = R_{15}|L_{15}
\]

b. Now suppose that we did away with the final 32-bit swap in the encryption algorithm. Then we would want the following equality to hold:
\[
T_D(\text{IP}^{-1}(T_{16}(L_{15}|R_{15})))) = L_{15}|R_{15}
\]

Does it?

3.11 Compare the initial permutation table (Table 3.2a) with the permuted choice one table (Table 3.4b). Are the structures similar? If so, describe the similarities. What conclusions can you draw from this analysis?

3.12 When using the DES algorithm for decryption, the 16 keys \( (K_1, K_2, \ldots, K_{16}) \) are used in reverse order. Therefore, the right-hand side of Figure 3.5 is no longer valid. Design a key-generation scheme with the appropriate shift schedule (analogous to Table 3.4d) for the decryption process.

3.13 a. Let \( X' \) be the bitwise complement of \( X \). Prove that if the complement of the plaintext block is taken and the complement of an encryption key is taken, then the result of DES encryption with these values is the complement of the original ciphertext. That is,
\[
Y = E(K, X) \\
Y' = E(K', X')
\]

Hint: Begin by showing that for any two bit strings of equal length, \( A \) and \( B \),
\[
(A \oplus B)' = A \oplus B
\]

b. It has been said that a brute-force attack on DES requires searching a key space of \( 2^{56} \) keys. Does the result of part (a) change that?

3.14 Show that in DES the first 24 bits of each subkey come from the same subset of 28 bits of the initial key and that the second 24 bits of each subkey come from a disjoint subset of 28 bits of the initial key.

3.15 For any block cipher, the fact that it is a nonlinear function is crucial to its security. To see this, suppose that we have a linear block cipher \( EL \) that encrypts 128-bit blocks of plaintext into 128-bit blocks of ciphertext. Let \( EL(k, m) \) denote the encryption of a 128-bit message \( m \) under a key \( k \) (the actual bit length of \( k \) is irrelevant). Thus
\[
EL(k, [m_1 \oplus m_2]) = EL(k, m_1) \oplus EL(k, m_2) \text{ for all } 128\text{-bit patterns } m_1, m_2
\]

Describe how, with 128 chosen ciphertexts, an adversary can decrypt any ciphertext without knowledge of the secret key \( k \). (A "chosen ciphertext" means that an adversary has the ability to choose a ciphertext and then obtain its decryption. Here, you have 128 plaintext/ciphertext pairs to work with and you have the ability to choose the value of the ciphertexts.)

Note: The following problems refer to simplified DES, described in Appendix C.

3.16 Refer to Figure C.2, which depicts key generation for S-DES.

a. How important is the initial P10 permutation function?

b. How important are the two LS-1 shift functions?

3.17 The equations for the variables \( q \) and \( r \) for S-DES are defined in the section on S-DES analysis. Provide the equations for \( s \) and \( t \).
3.18 Using S-DES, decrypt the string (10100010) using the key (01111111101) by hand. Show intermediate results after each function (IP, F_K, SW, F_K, IP^{-1}). Then decode the first 4 bits of the plaintext string to a letter and the second 4 bits to another letter where we encode A through P in base 2 (i.e., A = 0000, B = 0001, ..., P = 1111). Hint: As a midway check, after the application of SW, the string should be (00010011).

Programming Problems

3.19 Create software that can encrypt and decrypt using a general substitution block cipher.

3.20 Create software that can encrypt and decrypt using S-DES. Test data: Use plaintext, ciphertext, and key of Problem 3.15.