Problems

7.1 Electronic mail systems differ in the manner in which multiple recipients are handled. In some systems, the originating mail handler makes all the necessary copies, and these are sent out independently. An alternative approach is to determine the route for each destination first. Then a single message is sent out on a common portion of the route, and copies are made only when the routes diverge; this process is referred to as mail hugging.
   a. Leaving aside considerations of security, discuss the relative advantages and disadvantages of the two methods.
   b. Discuss the security requirements and implications of the two methods.

7.2 Section 7.2 describes the use of message length as a means of constructing a covert channel. Describe three additional schemes for using traffic patterns to construct a covert channel.

7.3 One local area network vendor provides a key distribution facility, as illustrated in Figure 7.15.
   a. Describe the scheme.
   b. Compare this scheme to that of Figure 7.9. What are the pros and cons?
We are under great pressure, Holmes. Detective Lestrade looked nervous. "We have learned that copies of sensitive government documents are stored in computers of one foreign embassy here in London. Normally these documents exist in electronic form only on a selected few government computers that satisfy the most stringent security requirements. However, sometimes they must be sent through the network connecting all government computers. But all messages in this network are encrypted using a top secret encryption algorithm certified by our best crypto experts. Even the NSA and the KGB are unable to break it. And now these documents have appeared in hands of diplomats of a small, otherwise insignificant, country. And we have no idea how it could happen."

"But you do have some suspicion who did it, do you?" asked Holmes.

"Yes, we did some routine investigation. There is a man who has legal access to one of the government computers and has frequent contacts with diplomats from the embassy. But the computer he has access to is not one of the trusted ones where these documents are normally stored. He is the suspect, but we have no idea how he could obtain copies of the documents. Even if he could obtain a copy of an encrypted document, he couldn't decrypt it."

"Hmm, please describe the communication protocol used on the network."

Holmes opened his eyes, thus proving that he had followed Lestrade's talk with an attention that contrasted with his sleepy look.

"Well, the protocol is as follows. Each node \( N \) of the network has been assigned a unique secret key \( K_v \). This key is used to secure communication between the node and a trusted server. That is, all the keys are stored also on the server. User A, wishing to send a secret message \( M \) to user B, initiates the following protocol:

1. A generates a random number \( R \) and sends to the server his name A, destination B, and \( E(K_v, R) \).
2. Server responds by sending \( E(K_v, R) \) to A.
3. A sends \( E(R, M) \) together with \( E(K_v, R) \) to B.
4. B knows \( K_v \), thus decrypts \( E(K_v, R) \) to get \( R \) and will subsequently use \( R \) to decrypt \( E(R, M) \) to get \( M \).

You see that a random key is generated every time a message has to be sent. I admit the man could intercept messages sent between the top secret trusted nodes, but I see no way he could decrypt them."
"Well, I think you have your man, Lestrade. The protocol isn’t secure because the server doesn’t authenticate users who send him a request. Apparently designers of the protocol have believed that sending $E(K_s, R)$ implicitly authenticates user X as the sender, as only X (and the server) knows $K_s$. But you know that $E(K_s, R)$ can be intercepted and later replayed. Once you understand where the hole is, you will be able to obtain enough evidence by monitoring the man’s use of the computer he has access to. Most likely he works as follows. After intercepting $E(K_s, R)$ and $E(R, M)$ (see steps 1 and 3 of the protocol), the man, let’s denote him as Z, will continue by pretending to be A and...

Finish the sentence for Holmes.

7.5 If we take the linear congruential algorithm with an additive component of 0:

$$X_{n+1} = (aX_n) \mod m$$

then it can be shown that if $m$ is prime, and if a given value of $a$ produces the maximum period of $m-1$, then $a^k$ will also produce the maximum period, provided that $k$ is less than $m$ and that $m-1$ is not divisible by $k$. Demonstrate this by using $X_0 = 1$ and $m = 31$ and producing the sequences for $a = 3, 3^2, 3^3,$ and $3^4$.

7.6 a. What is the maximum period obtainable from the following generator?

$$X_{n+1} = (aX_n) \mod 2^b$$

b. What should be the value of $a$?

c. What restrictions are required on the seed?

7.7 You may wonder why the modulus $m = 2^{31} - 1$ was chosen for the linear congruential method instead of simply $2^{31}$, because this latter number can be represented with no additional bits and the mod operation should be easier to perform. In general, the modulus $2^k - 1$ is preferable to $2^k$. Why is this so?

7.8 With the linear congruential algorithm, a choice of parameters that provides a full period does not necessarily provide a good randomization. For example, consider the following two generators:

$$X_{n+1} = (6X_n) \mod 13$$

$$X_{n+1} = (7X_n) \mod 13$$

Write out the two sequences to show that both are full period. Which one appears more random to you?

7.9 In any use of pseudorandom numbers, whether for encryption, simulation, or statistical design, it is dangerous to trust blindly the random number generator that happens to be available in your computer’s system library. [PARK88] found that many contemporary textbooks and programming packages make use of flawed algorithms for pseudorandom number generation. This exercise will enable you to test your system.

The test is based on a theorem attributed to Ernesto Cesaro (see [KNUT98] for a proof), which states the following: Given two randomly chosen integers, $x$ and $y$, the probability that $\gcd(x, y) = 1$ is $6/\pi^2$. Use this theorem in a program to determine statistically the value of $\pi$. The main program should call three subprograms: the random number generator from the system library to generate the random integers; a subprogram to calculate the greatest common divisor of two integers using Euclid’s Algorithm; and a subprogram that calculates square roots. If these latter two programs are not available, you will have to write them as well. The main program should loop through a large number of random numbers to give an estimate of the aforementioned probability. From this, it is a simple matter to solve for your estimate of $\pi$.

If the result is close to 3.14, congratulations! If not, then the result is probably low, usually a value of around 2.7. Why would such an inferior result be obtained?

7.10 Suppose you have a true random bit generator where each bit in the generated stream has the same probability of being a 0 or 1 as any other bit in the stream and that the bits are not correlated; that is the bits are generated from identical independent distribution. However, the bit stream is biased. The probability of $a$ is $0.5 + \delta$ and the probability of
a 0 is $0.5 - \alpha$, where $0 < \alpha < 0.5$. A simple deskewing algorithm is as follows: Examine the bit stream as a sequence of non-overlapping pairs. Discard all 00 and 11 pairs. Replace each 01 pair with 0 and each 10 pair with 1.

a. What is the probability of occurrence of each pair in the original sequence?

b. What is the probability of occurrence of 0 and 1 in the modified sequence?

c. What is the expected number of input bits to produce $x$ output bits?

d. Suppose that the algorithm uses overlapping successive bit pairs instead of nonoverlapping successive bit pairs. That is, the first output bit is based on input bits 1 and 2, the second output bit is based on input bits 2 and 3, and so on. What can you say about the output bit stream?

7.11 Another approach to deskewing is to consider the bit stream as a sequence of non-overlapping groups of $n$ bits each and the output the parity of each group. That is, if a group contains an odd number of ones, the output is 1; otherwise the output is 0.

a. Express this operation in terms of a basic Boolean function.

b. Assume, as in the preceding problem, that the probability of a 1 is $0.5 + \alpha$. If each group consists of 2 bits, what is the probability of an output of 1?

c. If each group consists of 4 bits, what is the probability of an output of 1?

d. Generalize the result to find the probability of an output of 1 for input groups of $n$ bits.

7.12 Suppose that someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key, and send the result over the channel. Your partner XORs the incoming block with the key (which should be the same as your key) and sends it back. You check, and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme?