1. (5 + 10 = 15 points)

   (a) Consider the simple cipher called the Caesar cipher or the additive cipher. We want to break a ciphertext encoded with the Caesar cipher by doing most of the work automatically. We use brute-force. We try all possible displacements for the letters of the alphabet, but want to visually look at just the top few results of “breaking” to see which ones make some sense. We know the standard frequencies of occurrence of letters for the natural language under consideration. Let the language have $n$ letters in its alphabet. Let the standard frequency vector for the language be $F$.

   $$F = [f_1 \cdots f_n]^T$$

   Let the frequencies obtained from one run of our cryptanalysis experiment be $F_e$.

   $$F_e = [f_{e1} \cdots f_{en}]^T$$

   The $T$ superscript means transpose. How do you automatically compute how similar $F$ and $F_e$ are so that you can present only the most probable cases for visual inspection? Assume we have enough text for statistical analysis.

   (b) Consider the Vigenère cipher, a polyalphabetic cipher that uses the Caesar cipher several times, based on the length of a keyword. Assume the length of the keyword under question is $l$. Assume that the letters of the keyword are randomly chosen. Let the alphabet under consideration be that of a natural language with $N_a$ characters. Let the number of characters in the ciphertext be $N_c$.

   We try Kasiski, Friedman and brute-force techniques for cryptanalysis of a ciphertext about which we have the fortuitous knowledge that it was produced using the Vigenère cipher. We assume we have enough text for these statistical approaches to be helpful.

   What is the worst-case time required by the Kasiski and Friedman tests? Express your answers using an order notation expression.

   How many possibilities do you have to try for complete brute-force analysis? What is the maximum amount of time required by the brute-force analysis? Express your answer using order notation. Assume you present only the best few alternatives to the person trying to break the code.
2. (8 + 7 = 15 points)

(a) Let \( \overline{A} \) be the bitwise complement of \( A \). Show that if the complement of the plaintext block is taken and the complement of an encryption key is taken in DES, the result of the encryption with these values is the complement of the original ciphertext. That is, if \( C = e_{DES}(K, M) \) then \( \overline{C} = e_{DES}(\overline{K}, \overline{M}) \). Here, \( K \) is the key, \( M \) is the plaintext, and \( C \) is the ciphertext. \( \overline{K}, \overline{M} \) and \( \overline{C} \) are the bitwise complements of \( K, M \) and \( C \), respectively.

**Hint:** Start by showing that for any two bit strings \( A \) and \( B \) of equal length, \( (A \oplus B) = \overline{A} \oplus \overline{B} \).

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

(b) The following is the specification of the \( S_0 \) box used in a simplified version of the Data Encryption Standard (DES). This simplified version is for academic purposes only and is called S-DES.

\[
S_0 = \begin{bmatrix}
1 & 0 & 3 & 2 \\
3 & 2 & 1 & 0 \\
0 & 2 & 1 & 3 \\
3 & 1 & 3 & 2
\end{bmatrix}
\]

Assume the input bits to \( S_0 \) are \( b_0 \cdots b_3 \). The first and the fourth bits are used as the binary specification of a row. The second and the third bits are used as the binary specification of a column. The row and the column are used to obtain a two-bit binary output from the two matrices. Let the two outputs for \( S_0 \) be \( s_0 \) and \( s_1 \).

Let an input to the \( S_0 \) box be called \( I \) and the output be called \( O \). We say \( I \) produces \( O \) by the \( S_0 \) box and write it as \( I \rightarrow O \) if an input of \( I \) produces an output of \( O \).

Suppose we have two inputs to the \( S_0 \) box: \( I \) and \( I^* \). We perform the XOR of the two inputs: \( I' = I \oplus I^* \). \( I' \) is called the input XOR. Let the outputs of the \( S_0 \) boxes corresponding to inputs \( I \) and \( I^* \) be \( O \) and \( O^* \), respectively. Let the XOR of the two outputs, i.e., the output XOR be \( O' = O \oplus O^* \).

The inputs to the \( S_0 \) box varies from 0000 to 1111, all binary. The outputs to the \( S_0 \) box varies from 00 to 11. Note that given a specific input XOR, not all possible output XORs actually are produced. Some output XORs are produced and others are not.

If we want to specify all possible input XORs and the number of possible output XORs for \( S_0 \) box in a table or matrix, what would be the dimensions of this table? Let the inputs represent rows of the matrix and the columns represent the numbers of output XORs. Fill up the row corresponding to the input XOR value of 0000.

3. (2 + 2 + 2 + 3 + 3 + 2 + 3 + 3 = 20 points)

Answer the following questions in a few sentences.

(a) What are the layers in the TCP/IP architecture for network communication?

(b) What is a socket in abstract terms?

(c) How does the ping command work?
(d) Compare MD2, MD4 and MD5 algorithms for one-way hashing.
(e) How does the Unix password system work?
(f) What is PAM used Linux?
(g) Specify how to use passwords in client-server computing so that it is “doubly” secure.
(h) Recently a Harvard professor has claimed that he has found a way to absolute secrecy in client-server communication. What is the essence of his claim?