Chapter 20 Symmetric Encryption and Message Confidentiality

20.1 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 missiles 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.

20.2 **a.** Let c = DESX(K, m). We first XOR both sides with k_2, which gives $c \oplus k_2$= DES(k_1, (m $\oplus$ k_2)). Next, we apply DES decryption, to get DES^{-1}(k_1, (c $\oplus$ k_2)) = DES^{-1}(k_1, [DES(k_1, (m $\oplus$ k_2)]) = m $\oplus$ k_2. Finally, we again XOR both sides with $k_2$to get the final result

$$m = \mathrm{DES}^{-1}(k_1, (c \oplus k_2)) \oplus k_2$$

b. DESX' can be attacked using a meet-in-the-middle attack. We assume that the adversary has a few plaintext/ciphertext pairs (m, c). He can then do a brute force attack on the DES part, i.e. compute x = DES(k_1, m) for all possible keys k_1, and store the resulting pairs (x, k_1) in a dictionary. Then he goes through all possible $k_2$values, computes $c \oplus k_2$ and looks it up in the dictionary. When found, he has a potential key pair (k_1, k_2). The complexity of this attack is 2^{64}, which shows that the cipher does not provide 120 bits of security.

Alternatively, with two plaintext/ciphertext pairs (m_1, c_1) and (m_2, c_2), one notes that $c_1 \oplus c_2$ = DES(k_1, m_1) $\oplus$ DES(k_1, m_2), which makes it possible to do a brute force attack with only twice the cost of an attack against DES.

20.3 The state in AES is a 4 × 4 matrix with entries in the field of 256 elements. The shift row layer shifts each row to the right a certain 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.

20.4 DES takes a 8-octet (64-bit) plaintext block and yields a 8-octet cipher block. CBC requires a 8-octet initialization vector (IV) to be sent along with the cipher blocks. So X now sends 64 octets of cipher blocks plus 8 octets of IV, for a total of 72 octets.