8.4

Let us start with an initial seed of 1. The first generator yields the sequence:

1, 6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11, 1, . . .

The second generator yields the sequence,

1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2, 1, . . .

Because of the pattern evident in the second half of the latter sequence, most people would consider it to be less random than the first sequence.

8.6:

Use a Key of length 255 bytes. The first two bytes are Zero; that is K[0] = K[1] = 0. Thereafter, we have; K[2] = 255; K[3] = 254; . . . . K[255] = 2.

8.7

a.) Simple store i, j and S, which requires 8+8+(256 $\boxed{?}$ 8] = 2064

b) The number of states is [256! $\boxed{?}$ 2562] $\boxed{?}$ 21700.

Therefore, 1700 bits are required.

8.8:

a.) By taking the first 80 bits of V || C; we obtain the initialization vector V. Since V, C, K are known, the message can be recovered by computing RC4 (V || K) ⊕ C.

b.) If the adversary observes that $v_i = v_j$ for distinct i, j then he/she knows that the same key stream was used to encrypt both m_i and m_j. In this case the message m_i and m_j may be vulnerable to the type of cryptanalysis carried out in part (a).

c.) Since the key is fixed, the key stream varies with the choice of 80 bit V, which is selected randomly. Thus after approximately messages are sent, we expect the same V, and hence the same key stream, to be used more than Once.

d.) The Key K should be changed some times before 2^{40} messages are sent.