8.2 (a) Maximum period obtainable from the generator is 4

(b) a must be 5 or 11

(c) Seed must be odd.

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 following sequence:

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

∴ There is a pattern in the 2nd half of the second sequence, we can say that it is less random than the 1st sequence.

8.6. Use a key of length 255 bytes. The first 2 bytes are zero;

i.e $K[0] = K[1] = 0$

Post that, we have

$K[2] = 255$

$K[3] = 254$

⋮

$K[255] = 2$

8.7 (a) We can simply store i, j and s which required $8 + 8 + (256 \times 8) = 2064$ bits

(b) The no. of states is $[256! \times 256^2] = 2^{1700}$

∴ 1700 bits are required

8.8 (a) By taking the first 80 bits of v||k, we obtain the initialization vector, v. Since, v, c, k are known, the message can be recovered (i.e decrypted) by computing $RC4(v||k) \oplus c$.

(b) If the adversary observes that $v_i = v_j$ for distinct i, j then he/she knows that the same key was used to encrypt both m_i and m_j.

(c) Since, the key is fixed, the key stream varies with the choice of the 64-bit v, that is selected randomly. Thus, after approx 2^{32} messages are sent, we expect the same 'v' and hence the same key stream to be used more than once.

(d) Key 'k' should be changed sometime before 240 messages are sent.