CS6530 - Cryptography Basics

Nileena PC
CS21M519

Assignment-2

Q8.3.

Generator ⇒ $X_{n+1} = (aX_n) \bmod 2^4$

(a) We need to find maximum m for which

$$a^m \bmod 2^4 = 1 \quad , c = 0, m = 2^4$$

$$\therefore \text{maximum period} = \frac{2^4}{4} = 2^{4-2} = 2^2 = \underline{\underline{4}}$$

(b) $a^n \bmod 16 = 1$

$a \cdot a^{n-1} \bmod 16 = 1$

$a^{n-1} \bmod 16 = a^{-1}$

For a^{-1} to exist, a must be coprime with 16.

Therefore a must be an odd number.

When $a = 3$ we get $\{5, 9, 13, 1\}$

When $a = 5$ we get $\{11, 9, 3, 1\}$

$\therefore$ a must be 3 or 5

(c) As explained above a must be odd.

Q8.4.

Consider the initial seed $X_0 = 1$

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

$X_1 = 6 \bmod 13 = 6$

$X_2 = 6 \cdot 6 \bmod 13 = 10$

$X_3 = 6 \cdot 10 \bmod 13 = 8$

$X_4 = 6 \cdot 8 \bmod 13 = 9$

$X_5 = 6 \cdot 9 \bmod 13 = 2$

$X_6 = 6 \cdot 2 \bmod 13 = 12$

$X_7 = 6 \cdot 12 \bmod 13 = 7$

$X_8 = 6 \cdot 7 \bmod 13 = 3$

$X_9 = 6 \cdot 3 \bmod 13 = 5$

$X_{10} = 6 \cdot 5 \bmod 13 = 4$

$X_{11} = 6 \cdot 4 \bmod 13 = 11$

$X_{12} = 6 \cdot 11 \bmod 13 = 1$

Period = 13

Sequence - $\{1, 6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11\}$

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

$X_0 = 1$

$X_1 = 7 \bmod 13 = 7$

$X_2 = 7.7 \bmod 13 = 10$

$X_3 = 10.7 \bmod 13 = 5$

$X_4 = 7.5 \bmod 13 = 9$

$X_5 = \cancel{7}.9 \bmod 13 = 11$

$X_6 = 7.11 \bmod 13 = 12$

$X_7 = 7.12 \bmod 13 = 6$

$X_8 = 7.6 \bmod 13 = 3$

$X_9 = 7.3 \bmod 13 = 8$

$X_{10} = 7.8 \bmod 13 = 4$

$X_{11} = 4.7 \bmod 13 = 2$

$X_{12} = 2.7 \bmod 13 = 1$

Period = 13

Sequence = $\{1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2\}$

Considering the 2 sequences we can see a pattern in the second half of the second sequence (a number is half the previous value in some cases).

Hence the sequence generated by $X_{n+1} = (6X_n) \bmod 13$ is more random as we cannot establish any pattern between the consecutive generated numbers.

Q.8.6

The initialisation logic is given by:

```
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
```

- here keylength is the length of the key

Initial permutation logic is given by

```
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) mod 256
    Swap (S[i], S[j])
```

- The value of S[] will remain unchanged if the value of j calculated is same as i.

Initially $j=0 \Rightarrow$ if $\underline{\underline{T[0]=0}}$ then $j = S[i] = S[0] = 0$

next iteration, $j = 0$, $S[i] = 1$ $\therefore$ $T[1] + 1 + 0 = i = 1$ $\Rightarrow$ $\underline{T[1] = 0}$

now j must be 2 $\Rightarrow$ $1 + S[2] + T[2] = 2$ $\Rightarrow$ $T[2] + 3 = 2 \pmod{256}$

$\Rightarrow$ $\underline{\underline{T[2] = 255}}$

Similarly $T[3] + S[3] + 2 = 3 \pmod{256}$

$$T[3] + 5 = 3 \pmod{256}$$

$$\Rightarrow \underline{T[3] = 254}$$

$\therefore \quad T[4] = -j \pmod{256} = -3 \pmod{256} = \underline{\underline{253}}$

$T[5] = 252$

$\vdots$

$T[255] = 2$

We can see $j = i-1 \Rightarrow T[i] = -j \pmod{256} = -(i-1) \pmod{256} = 256-(i)+1 = \underline{\underline{257-i}}$

We can write $T[j] = \begin{cases} 0 & \text{if } j = 0 \text{ or } 1 \\ 257-j, & \text{if } j = 2 \text{ to } 255. \end{cases}$

Q8.7

(a) Total number of bits required to store in the system is given by

no. of bits (i) + no. of bits (j) + no. of bits (s)

$= 8$ bits $+ 8$ bits $+ (256 \times 8)$ bits

$= \underline{\underline{2064 \text{ bits}}}$

(b) The number of states $= 256 \times 256 \times 256!$

$$= \underline{\underline{2^{1700}}}$$

Therefore no. of bits required $= \underline{\underline{1700 \text{ bits}}}$

Q8.8

(a) To get v, we take the first 80 bits of $v \| c$.

Given $C = RC4(v \| k) \oplus m$

$\Rightarrow m = RC4(v \| k) \oplus c$

Since we know v, c, k, the message is decrypted using XOR on $v \| k$ and c.

(b) Since the adversary can get hold of v_i by taking the 1st 80 bits, on comparing the ciphertexts, if he observes $v_i = v_j$ for distinct i, j, then the key stream generated must also be the same $\because$

$k_i = k_j$ as it is constant

$RC4(v_i \| k_i) = RC4(v_j \| k_j) \Rightarrow v_i = v_j$

(c) Given that the key ~~stream~~ is fixed. Therefore the key stream varies with 80 bit value which is randomly selected (v). Therefore 2^{80} possible values for v

According to birthday paradox for the key stream to have same value, atleast $\sqrt{\frac{\pi}{2} \cdot 2^{80}} \simeq 2^{40}$ messages needs to be sent. We can expect key stream to be same only after atleast 2^{40} messages sent.

(d) The above statement with the help of birthday paradox implies that if the key is not changed after 2^{40} messages are sent atleast one of the key stream is going to repeat and adversary can find v and k. Therefore key must be changed before 2^{40} messages are sent.