:- Assignment 2 :-

8.2

a. $X_{n+1} = a X_n \bmod 2^4$.

What is the maximum period obtainable from the following generator?

→ 16 (2^4)

{1, 3, 5, 7, 9, 11, 13} → relatively prime.

2 = 2 × 1 a. Maximum period = $2^{4-2} = 4$

4 = 2 × 2

b. a must be 5 or 11

c. The seed must be odd.

8.4

$X_{n+1} = 6X_n \mod 13$

$X_{n+1} = 7X_n \mod 13$

write out the two sequences to show that both are full period. Which one appears more random to you?

→ Start with $X_n = 1$ $= 1$

$X_{n+1} = 6 \times 1 = 6 \mod 13 = 6$

$= 6 \times 6 = 36 = 10 \mod 13 = 10$

$6 \times 10 = 60 = 8 \mod 13 = 8$

$6 \times 8 = 48 \mod 13 = 9$

$6 \times 9 = 54 \mod 13 = 2$

$6 \times 2 = 12 \mod 13 = 12$

$6 \times 12 = 72 \mod 13 = 7$

$6 \times 7 = 42 \mod 13 = 3$

$6 \times 3 = 18 \mod 13 = 5$

$6 \times 5 = 30 \mod 13 = 4$

$6 \times 4 = 24 \mod 13 = 11$

$6 \times 11 = 66 \mod 13 = 1$

Sequences are = {1, 6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11}

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

$7 \times 1 = 7 \bmod 13 = 7$

$7 \times 7 = 49 \bmod 13 = 10$

$7 \times 10 = 70 \bmod 13 = 5$

$7 \times 5 = 35 \bmod 13 = 9$

$7 \times 9 = 63 \bmod 13 = 11$

$7 \times 11 = 77 \bmod 13 = 12$

$7 \times 12 = 84 \bmod 13 = 6$

$7 \times 6 = 42 \bmod 13 = 3$

$7 \times 3 = 21 \bmod 13 = 8$

$7 \times 8 = 56 \bmod 13 = 4$

$7 \times 4 = 28 \bmod 13 = 2$

$7 \times 2 = 14 \bmod 13 = 1$

$7 \times 1 = 7 \bmod 13 = 7$

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

In these two sequences $6X_n \bmod 13$ looks more random. Because 2nd sequence next value looks half of previous so guessing is easy.

8.6 What RC4 key value will leave S unchanged during initialization? That is, after the initial permutation of S, the entries of S will be equal to the values from 0 through 255 in ascending order.

→

RC4

Initialization

for i = 0 to 255 do.

S[i] = i ;

T[i] = K[i mod keylen] ;

j = 0 ;

for i = 0 to 255 do.

j = (j + S[i] + T[i]) mod 256 ;

swap (S[i], S[j]);

→ To get S unchanged,

i = j for all the values of i.

If j = i then j + S[i] + T[i] = i ∀i.

for 0

S[0] = 0, T[0] = x, j = 0.

0 + 0 + x = 0. mod 256.

x = 256K

for i = 1

j = 0 S[1] = 1 T[1] = x. j = 1

1 = 0 + 1 + x mod 256

$x = 256k \mod 256$

if $1j = 0 + 1 + 0$

if when $i = 2$

$j = 1$

$2 = 1 + S[2] + T[2]$

$2 = 1 + 2 + x$

$-1 = x \mod 255$

$\boxed{x = 254}$

when $i = 3$

$j = 2$

$3 = 2 + S[3] + T[3]$

$3 = 2 + 3 + x$

$\boxed{-2 = x}$

$\boxed{x = 253}$

when $i = 255$

$j = 254$

$255 = 254 + 255 + x$

$x = -254 \mod 255$

$\boxed{x = 1}$

So

so the values of T are...

T[0 to 255]
= {255, 255, 254, ... 1}

8.7

Simply store i, j and s, which requires 8+8+(256*8)
= 2064 bits.

The number of states (256! * 256^2) + 1700. Therefore 1700 bits are required.

8.8

a) By considering first 80 bits of VIIC, we get initialization vector V since V, C, K are known, The message can be decrypted by:

$RC_4(VIIK) \oplus C$

b) If the adversary knows that $V_i = V_j$ for unique i, j thus he knows that the same key stream was used to encrypt m_i and m_j thus the message becomes vulnerable and can be cracked.

c) The key stream varies with selection of 80 bit V as key K in f

∴ No of bits to be encrypted using same key = 2^{40}

∴ Number of message Alice can send before same key stream used twice = 2^{40}.

d) Lifetime of key k = No of message that can be encrypted with same key k

$= 2^{40}$

8.5 Programming exercise:

$p(g(x;y)) = 1 \Rightarrow 1/6\pi^2$

8.5 Programming exercise

```
#include <stdio.h>

int gcd(int a, int b)
{
    if (a==0)
        return b;
    if (b==0)
        return a;

    if (a==b)
        return a;

8.5
    // a is greater.
    if (a>b)
        return gcd(a-b, b);
    return gcd(a, b-a);
}
```

```
int main()
{   int attempts=100, GCD, count, pi, prob
    while (attempt>0)
    {
        x=rand();
        y = rand();
        GCD=gcd(x,y);
        if (GCD==1)
        {
            count++;
        }
    }
    float prob; prob= count/attempt;
    //we know that prob = 6/π2 ;
    float pi;
    count = 6/
    pi = 6/count
    pi = sqrt(pi);

    printf("%.2f value is %f", pi);
    return 0;
}
```