NAME - RAM CHANDRA JANGIR
ROLL NO. - CS21M517
Subject- CS6530 - Assignment-2

Que 1 (Page-280, Problem-8.2)

(a) what is the maximum period obtainable from the following generator ?

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

here $m = 2^4$ and $c = 0$

Max possible period $P = \frac{m}{4} = \frac{2^4}{4} = 2^{4-2} = 2^2 = 4$

(b) Value of a should be following.

$a = 3 + 8K$ or

$a = 5 + 8K$ for $K = 0, 1, 2 \cdots$

so a must be 3 or 5.

(c) what restrictions are required on the seed?

Since seed X_0 is odd

Que 2 (Page-280, Problem-8.4)

With the Linear congruential algorithm, a choice of parameters that provides a full period does not necessarily provide a good randomization. For example cosider the following two generators

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

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

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

Let's assume that initial seed $X_0 = 1$

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

$X_0 = 1$

$X_1 = (6 X_0) \bmod 13 = 6$

$X_2 = (6.6) \bmod 13 = 10$

$X_3 = (6.10) \bmod 13 = 8$

$X_4 = (6.8) \bmod 13 = 9$

$X_5 = (6.9) \bmod 13 = 2$

$X_6 = (6.2) \bmod 13 = 12$

$X_7 = (6.12) \bmod 13 = 7$

$X_8 = (6.7) \bmod 13 = 3$

$X_9 = (6.3) \bmod 13 = 5$

$X_{10} = (6.5) \bmod 13 = 4$

$X_{11} = (6.4) \bmod 13 = 11$

$X_{12} = (6.11) \bmod 13 = 1$

$X_{13} = (6.1) \bmod 13 = 6$

So Now sequence started repeating

So the sequence is $\{1, 6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11, 1, 6 \ldots\}$

(ii)

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

$X_0 = 1$

$X_1 = (7.1) \bmod 13 = 7$

$X_2 = (7.7) \bmod 13 = 10$

$X_3 = (7.10) \bmod 13 = 5$

$X_4 = (7.5) \bmod 13 = 9$

$X_5 = (7.9) \bmod 13 = 11$

$X_6 = (7.11) \bmod 13 = 12$

$X_7 = (7.12) \bmod 13 = 6$

$X_8 = (7.6) \bmod 13 = 3$

$X_9 = (7.3) \bmod 13 = 8$

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

$X_{11} = (7.4) \bmod 13 = 2$

$X_{12} = (7.2) \bmod 13 = 1$

So Now sequence started repeating

So the sequence is = $\{1, 7, 10, 5, 9, 11, 12, 6, 3, 8, 4, 2, 1, \ldots\}$

Which one is more random to you?

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

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

Because of patterns evident in the second half of the second sequence we consider it to be less random than the first sequence.

Question-3 (Page-281, Problem-8.5)

rjangir@rjangir-linux:/local/mnt/workspace/rjangir/crypto_assign2$ cat test_knut98.c

```
#include <stdio.h>
#include <stdlib.h>

#define SEED 35791246
#define MAX_SIZE 100

/* Calculate gcd(a,b) */
Int  gcd (int a, int b)
{
 int m = a, n = b, r;
 while (n > 0)
  {
    int r = m % n;
    m = n;
    n = r;
  }
 return m;
}

/* Calculate the square root of value */
double calculate_sqrt (double n)
{
 double start, end, mid;
 start = 0, end = n;

 while ((end - start) >= 0.000001)
  {

    mid = (start + end) / 2;

    if (mid * mid < n)
      start = mid;
    if (mid * mid >= n)
      end = mid;

  }

 return mid;
}

double get_probability (int sum, int total)
{
```

```
 double Pi, p;
 p = 6 * total / sum;
 /* Calculate the square root of value */
 Pi = calculate_sqrt (p);
 return Pi;
}

Int main ()
{
 /* initialize random numbers */
 srand (SEED);
 int i, niter, a[MAX_SIZE];

 printf ("Enter the number of iterations(<= %d) used to estimate pi: ",
       MAX_SIZE);
 scanf ("%d", &niter);

 if (niter > MAX_SIZE)
  {
   printf ("the number of iterations are greater than %d, Exiting\n",
         MAX_SIZE);
   exit - 1;
  }

 for (i = 0; i < niter; i++)
  a[i] = rand () % niter + 1;
 printf ("The generated random numbers with seed (%d) are: \n", SEED);

 int m, j;
 int sum = 0;

 for (j = 0; j < niter; j += 2)
  {
   m = gcd (a[j], a[j + 1]);
   if (m == 1)
    sum = sum + 1;
   printf ("sum:%d, GCD(%d,%d) = %d \n", sum, a[j], a[j + 1], m);
  }

 double Pi = get_probability (sum, niter / 2);
 printf ("\nThe estimate value of Pi is:%f\n ", Pi);

}
```

```
rjangir@rjangir-linux:/local/mnt/workspace/rjangir/crypto_assign2$ gcc  -o test test_knut98.c

rjangir@rjangir-linux:/local/mnt/workspace/rjangir/crypto_assign2$ ./test
Enter the number of iterations (<= 100) used to estimate pi: 100
The generated random numbers with seed (35791246) are:
sum:0, GCD (54,87) = 3
sum:1, GCD (25,52) = 1
sum:2, GCD (79,37) = 1
sum:3, GCD (58,53) = 1
sum:3, GCD (6,60) = 6
sum:3, GCD (36,74) = 2
sum:4, GCD (68,55) = 1
sum:5, GCD (5,68) = 1
sum:6, GCD (17,9) = 1
sum:7, GCD (95,12) = 1
sum:7, GCD (38,38) = 38
sum:8, GCD (29,43) = 1
sum:8, GCD (40,30) = 10
sum:9, GCD (4,67) = 1
sum:10, GCD (73,71) = 1
sum:10, GCD (65,26) = 13
sum:11, GCD (9,41) = 1
sum:11, GCD (29,87) = 29
sum:12, GCD (29,38) = 1
sum:12, GCD (91,35) = 7
sum:13, GCD (50,79) = 1
sum:14, GCD (8,69) = 1
sum:15, GCD (85,12) = 1
sum:15, GCD (36,2) = 2
sum:15, GCD (21,30) = 3
sum:15, GCD (65,10) = 5
sum:15, GCD (20,46) = 2
sum:16, GCD (52,59) = 1
sum:17, GCD (75,56) = 1
sum:17, GCD (77,99) = 11
sum:17, GCD (78,93) = 3
sum:17, GCD (76,86) = 2
sum:17, GCD (34,56) = 2
sum:17, GCD (24,62) = 2
sum:17, GCD (94,66) = 2
sum:18, GCD (48,95) = 1
sum:18, GCD (44,8) = 4
sum:19, GCD (63,29) = 1
```

sum:20, GCD(19,50) = 1
sum:21, GCD (82,91) = 1
sum:22, GCD (31,46) = 1
sum:23, GCD (53,50) = 1
sum:24, GCD (91,4) = 1
sum:25, GCD (61,17) = 1
sum:26, GCD (11,37) = 1
sum:27, GCD (67,40) = 1
sum:27, GCD (30,42) = 6
sum:28, GCD (77,15) = 1
sum:28, GCD (98,100) = 2
sum:29, GCD (28,43) = 1

The estimate value of Pi is:3.162277

We are getting probability as 3.162277 which is very close to 3.14. So we are good.

Que-4 (Page-281, Problem-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.

Ans:

Use a key of length 255 bytes. The first two bytes are zeros, that is $K[0] = K[1] = 0$.

Thereafter, we have:

$K[2] = 255$

$K[3] = 254$

$K[4] = 253$

$\vdots \quad \vdots$

$K[255] = 2$.

Que :5 (Page-281, Problem-8.7)

RC4 has a secret internal state which is a permutation of all possible values of the vector S and the two indices i and j.

(a) Using a straightforward scheme to store the internal state, how many bits are used?

~~Ans:~~

(b) Suppose we think of it from the point of view of how much information is represented by the state. In that case we need to determine how many different states there are, then take log to the base 2 to find out how many bits of information this represents. Using this approach, how many bits would be needed to represent the state?

Answer:

(a)

Simply store x, y and s requires $8+8+(256\times 8)$

$= 8+8+($

$= 2064$ bits

(b)

The number of states is $[256! \times 256^2]$

$= 2^{1700}$

Therefore 1700 bits are required.

Que-6 (Page-281, Problem-8.8)

Alice and Bob agree to communicate privately via email using a scheme based on RC4, but want to avoid using a new secret key for each transmission. Alice and Bob privately agree on a 128-bit key K. To encrypt a message m, consisting of a string of bits, the following procedure is used

1. choose a random 80-bit value V.
2. Generate the ciphertext $c = RC4(V||K) \oplus m$
3. Send the bit string $(V||C)$

(a). Suppose Alice uses this procedure to send a message m to Bob. Describe how Bob can recover the message m from $(V||C)$ using K.

(b) If an adversary observes several values $(V_1||C_1), (V_2||C_2) \ldots$ transmitted between Alice and Bob, how can he/she determine when the same key stream has been used to encrypt two messages?

(c) Approximately how many messages can Alice expect to send before the same key stream will be used twice? Use the result from the birthday paradox described in Appendix U.

(d) What does this imply about the lifetime of the key K (i.e. the number of messages that can be encrypted using K)?

Answer:

(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 (i.e. decrypted) by computing: $RC4(V||K) \oplus C$.

(b) If the adversary observes that $V_i = V_j$ for distinct i, j the he/she knows that the same key stream 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 80bit V, which is selected randomly. Thus after approximately $\sqrt{\frac{\pi}{2} \cdot 2^{80}} \approx 2^{40}$ 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 sometime before 2^{40} messages are sent.