Pseudo Random Generator

Seed_size = 16

generator = 223

modulus = 36389

function L:

Input: An integer K

Output: value of a polynomial of k

Here the polynomial L(k) = 2*k

Function H:

Input(x,y): Let input be random bit sequence of length n.

X and Y are of same length.

Output: Bit sequence of length n+1.

Return logic: f(X) || Y || Hard_Core_Bit

$f(X) = g^x \bmod p$

Hard_Core_Bit is the Xor operation of X&Y. Here & is a Bitwise- AND operation.

|| refers to the conacatenation of strings.

function G:

Input: Random bit sequence seed of arbitrary length.

Output: Random bit sequence of length l(K).

Algorithm:

For i in range of l(seed_size):

Let x be the first half of the seed

Let y be the second half of the seed

Compute H(x,y) and take out hard_core_bit and append to the result

Update the seed

End-for

Return result