PRG-Documentation

The class PRG implements the PRG.

init :

Params:
p: prime number
k: number of bits in input
l: number of bits in output

prime_factor:

Params:
number: any positive integer

Returns:
List of prime factors of the number

FInds prime factors using a method close to the sieve of eratosthenes.

find_g:

Returns:
Primitive root of prime number p.

Checks for all numbers below p, if the modular exponent is 1. If so, return such number, else return -1/

to_binary:

Params:
x: input number
bit_length: number of bits in the output binary

Returns
Binary of the input decimal

to_decimal:

Params:
list_bin:
Binary number in the form of a list

Returns
Decimal equivalent of the binary number

get_last_bit:

Params:
Number in decimal form

Returns:
Returns the last bit

get_one_bit:

Params:
x: first half of the input number
y: second half of the input number
bit_length: length of x or y

Returns:
A decimal number with one bit more than the input number

Computes the modular exponent of the first half with respect to prime number p. Later, computes the bit wise And of x and y and computes the xor of the resultant, which becomes the extra bit. The modular exponent, y and bit b are concatenated respectively and returned.

generate:

Params:
s: seed

Returns:
Random value of l length

Computes 1 extra bit in each iteration using get_one_bit and returns l bit output.