Pseudo Random Generator(PRG)

Sarthak Mahajan, 2018111005

March 7, 2022

1 Code structure:

- This folder contains prg.py and run.py code files
- prg.py contains the code to create a PRG output
- run.py is used to take inputs and return outputs to the PRG

2 Instructions to run the code:

- Run the following command in the folder where this document is contained: python3 run.py
- There will be prompts asking you to input the relevant information serially, and will return the output.

3 Explanation of Functions in run.py:

3.1 run():

- It asks for inputs of the key, key length in binary, new key length after expansion(all in decimal format)
- Calculates p: the order of the group for DLP and g: the generator of this group.
- prints the output of the PRG.

4 Explanation of Functions in prg.py:

4.1 primecheck(N):

- N is in decimal format.
- Function return True if N is a prime, otherwise it returns False

4.2 largestprime(p):

- p is in decimal format
- returns the largest prime number lesser than p

4.3 pad(key_len, num):

- num is a binary number, key_len is a decimal number
- The function pads num with zeros on the left till its size is equal to key_len if length of num is lesser than key_len and returns padded num(in binary form), otherwise num(in binary form) is returned directly.

4.4 dlp(s, p, g)

- s,p,g are all decimal numbers
- Calculates Discrete log: $g^s \mod p$ and returns it(in decimal format)
- Note that the value of p should be prime such that its length is lesser than 2^n bits, where n is the length of the key (s), for which this function is being used
- g's value lies in the range: $(1, p-1)$, but is fixed in terms of p, for the program, but could be any random value in the given range.

4.5 hc_dlp(num, key_len):

- both num, key_len are decimal numbers
- It returns the MSB of num in binary format after it has been padded. till length key_len using the pad function.
- Uses pad function

4.6 single_PRG(seed, key_len,p,g, func="dlp")

- p,g are in decimal format, they are needed to be passed to the dlp function. p is the largest prime possible in n bits, where n= length of the key, g is a random number in the range(1,p-1); note that this is fixed for the program in terms of p(so that same seed gives same value for different iterations of the program), but it could be any random number in the given range.
- seed, key_len are decimal numbers while func is a string
- This function calculates and returns the single bit expansion PRG output in decimal format with seed as 'seed', and key_len is the length of the key in binary format. func is the to check the kind of one way function being used.
- Uses dlp, hc_dlp functions.

4.7 PRG(seed, new_l, key_len,p,g):

- p,g are in decimal format, they are needed to be passed to the dlp function. p is the largest prime possible in n bits, where n= length of the key, g is a random number in the range(1,p-1); note that this is fixed for the program in terms of p(so that same seed gives same value for different iterations of the program), but it could be any random number in the given range.
- seed, new_l, key_len are all decimal numbers and returned value is a binary number.
- It generates a pseudorandom number of length new_l, from the number: seed of length key_len(which is just the key size in binary format).
- Uses pad, single_PRG functions.