PRF Documentation

__init__(self, t):

Input variables: t

- t decides which secure mode the cpa is to be used in, by initializing self.mode

binstring(self, g, p):

Input variables: x

- Takes a number x and outputs it’s corresponding binary representation in the form of a
- string
- It’s a helper function

genKey(self,x):

Input variables: n

- Generates a random binary string of length n

getint(self,s):

Input variables: s

- Returns the integer value of a string containing the binary representation

setStrLen(self, s, n):

Input variables: s, n

- Sets a binary string to given length n and returns it

getxor(self, s1, s2):

Input variables: s1, s2

- Performs xor between to binary strings and returns the outcome in binary string format

prf_basic(self, prg, k, x):

Input variables:prg, k , x

- Takes in a prg object, key and a message x
- Implements a fixed-length PRF using the prg object
-

cpa(self, prg, m, k=None):

Input variables: prg, m, k

- Implements secure cpa encryption scheme for fixed-length message m

cpa_dec(self, prg, c, k):

Input variables: prg, iv_init, k, c

- Implements decryption step for secure cpa scheme