Module `crypto-lib.prg.prg`

Classes

`class DiscreteLog`

Class for performing dicrete log computation and finding the hardcore predicate.

Initialize prime and generator

Methods

`def evaluate(self, val)`

Performs the discrete log computation for the assigned prime and generator

`def hardcore_pred(self, val)`

Returns the hardcore predicate given the output of dicrete log (The MSB)

`class OneWayFunc (type=1)`

General function adaptible to any one way function or permutation to be added in the future

Initialize type of one way function used

Methods

`def evaluate(self, val)`

Compute the one way function

`def hardcore_pred(self, val)`

Return the hardcore predicate given the output

`class PRG (type=1)`

Class for generating a n-bit pseudo random number

Initialize initial value and the type of one-way function

Methods

`def add_bit(self, bit)`

Internal function for genrating an extra bit of the prg

```
def gen_n_bit(self, n)
```

Generate a random n-bit (pseudo)random number using the one way function and return it.

```
def init_val(self, val)
```

Initialize the value of the PRG using some seed