Pseudo Random Generator

To make a PRG, we use a one-way function which prevents finding the input given the output.
For our purposes, we can use the Discrete Log Problem which is a one way permutation (one-way function).

Discrete Log Problem -

Prime p, Generator g

$y = f(x) = g^x \bmod p$.

Given y, g, p finding x is a hard problem. This allows usage of some bit from y as a random bit as there is no way to find x from the random bit. Thus, the process of generation of this random number is not repeatable by an adversary.

Hardcore Predicate -

The bit of x which is hardest to figure out given the output y is called the hardcore predicate. It can be used as the random bit in PRG. For the Discrete Log Problem, the MSB is the harcore predicate.

PRG -

A generator is pseudo random if no polynomial time adversary can distinguish a random or non-random output.
To generate an n-bit random number we use the following algorithm -

```
for i : 1 to n:
    x = f(x)
    output += harcore_pred(x)
```

This algorithm hence generates a stream of n random bits stored in the output variable.