Build a provably secure PRG

In case of discrete logarithms : G(s) = (g^s mod p, msb(s))

Here, msb(x) is a hardcore predicate of Discrete Logarithm Problem and it is the hardest bit of information about x to obtain from f(x).
A function hc : {0,1}* -> {0,1} is a hard-core predicate of a function f if

1. hc can be computed in polynomial time, and
2. For every probabilistic polynomial-time algorithm A there exists a negligible function negl such that

   Pr (x <- $\{0,1\}^n$) [A(f(x)) = hc(x)] <= ½ + negl(n),

   Where the probability is taken over the uniform choice of x in $\{0,1\}^n$ and the random coin tosses of A.

The function PRG uses the hardcore predicate of the discrete log function to generate a binary string of the specified length.

The argument seed is associated with the random string. It returns an integer whose binary representation was generated.

The code has been explained in the comments present in the script wherever necessary.