Pseudo Random Generator:

Pseudo Random Generator should produce a bit sequence indistinguishable from a real random source. It is an algorithm that executes in polynomial time and calculates a random bit sequence such that

$$G : \{0,1\}^k \rightarrow \{0,1\}^{l(k)}$$

Where l(k) is monotonically increasing function.

The probability that a probabilistic polynomial time algorithm distinguishes the random output sequence form a real random generator tends to 0 as the length of the seed increases.

$$\left| Pr\left[x \leftarrow \{0,1\}^k ; r \leftarrow G(x) : D(r) = 1\right] - Pr\left[r \leftarrow \{0,1\}^{l(k)} ; D(r) = 1\right] \right| < \frac{1}{p(k)}$$

The Generator G takes the initial seed of 'k' bits and returns the random output seed of l(k) bits. To build the generator 'G', we define a simple Pseudo Random Generator 'H' that takes the input of k bits and outputs the bit sequence of (k+1) bits. The extra bit is known as Hard Core bit.

$$H : \{0,1\}^k \rightarrow \{0,1\}^{k+1}$$

H is one way permutation if it is hard to invert i.e., given y it is difficult to calculate x such that H(x)=y.

$$H(x \| y) = f(x) \| y \| \langle x, y \rangle$$

Where

$$f(x) = g^x \mod p$$

f(x) is a one way permutation,

<x,y> is hardcore bit.