PRG Concepts

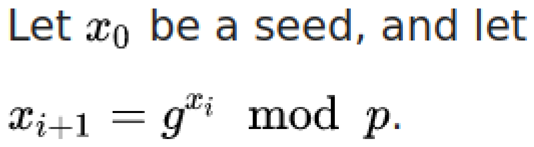
Overview:

A pseudorandom generator G is an efficient, deterministic algorithm for transforming a short, uniform string (called a seed) into a longer, “uniform-looking” (or “pseudorandom”) output string. Stated differently, a pseudorandom generator uses a small amount of true randomness in order to generate a large amount of pseudorandomness. This is useful whenever a large number of random(-looking) bits are needed, since generating true random bits is often difficult and slow.

Code Construction:

Pseudo-Random Generator that is implemented in the code is commonly known as the **Blums Micali Pseudo-Random Generator** based on the **Discrete Log Problem**.

Here let p be an odd prime number and g be the generator of the cyclic group G generated by taking modulo from the given odd prime number.



Let x_0 be a seed, and let

$$x_{i+1} = g^{x_i} \mod p.$$

When x_i <= (p-1)/2 the algorithm outputs 1 otherwise its output is 0

Since the Discrete log problem is a hard problem this is a cryptographically secure pseudorandom generator.

In reality, for it to be secure the prime number p needs to be a large number, and any method that predicts the numbers that are generated will be able to solve the Discrete Log problem which is impossible to be solved in polynomial time.