**Discrete Logarithm Function (DL)**

Taking p as a prime, g is a generator such that and x as the initial seed.

Assuming discrete logarithm is a one-way function.

A function is one way if the following two conditions hold:

1. (Easy to compute): There exists a polynomial time algorithm Mf computing f that is Mf (x) =f (x) for all x.
2. (Hard to Invert): For every probabilistic polynomial time algorithm A, there exists a negligible function such that

**Hardcore Predicate**

A function is a hard-core predicate of a function f if

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

where the probability is taken over the uniform choice of x in and the random coin tosses of A.

**Pseudo Random Generator (PRG)**

PRG is obtained by taking DL as a one-way function and the hard-core predicate of the one-way function. Let f be a one-way permutation and let hc be a hard-core predicate of f. Then,

constitutes a pseudorandom generator with expansion factor .

In case of DL, MSB(x) is a hardcore predicate. Hence, the function G is as follows

Hardcore bit of DL

Let p be an n-bit prime, . Define as

B(x) is hard to compute from f(x), where f(x) is the DL function.

The above function G, can generate an expansion from n bit to n+1 bit. Assuming that there exists a pseudorandom generator with expansion factor them for any polynomial p(.), there exists a pseudorandom generator with expansion factor.

1. Take the last bit from l+1 length string for output.
2. Consider the remaining l length output as input of G for the next iteration.
3. Apply above 1,2 steps ntimes to get output of string n

References

[1] J. K. a. Y. Lindell, Introduction to Modern Cryptography.

[2] B. Micali, "Hardcord bits," [Online]. Available: <https://crypto.stanford.edu/pbc/notes/crypto/hardcore.html>.

[3] Lecture Slides