Security of the Wolfram-PRG

Varul Srivastava

Suppose we have access to the previous K numbers produced by the PRG.
The PRG code contains a private key p_1 a public key p_2 and a third key a_1(no secrecy value).

We know that for any cryptographic system, if we can know the key, we can break the PRG and thereby we can break the system itself. So first part will contain, proof that obtaining the key after knowing any K inputs we cannot predict the output.

PART 1 :: Obtaining the key from K numbers.
Case 1: K < 64

In this case, we cannot reverse engineer the value $k_1 = (k_0 * p_1 + p_2) \bmod M$.
Since, available information < 64 bits
required information = 64 bits
available information < required information
Therefore, it is impossible to generate such system that predicts 64 bits from less tham 64 bits.

Case 2: K $\geqslant$ 64

In this case, we can concatenate the first bits of the 64 numbers, and generate the numbers
k_1, k_2, k_3..... and so on. Now, to compute k_{n+1} from k_n numbers,
$k_{n+1} = (k_n * p_1 + p_2) \bmod M$.
$P(i^{th}$ bit of $k_{n+1} \mid k_{1,2,...n}) = 0.5$ (approx.) [Equivalent of a random guess]

Predicting from theory of cellular automata :

Let k^i_n be the i^{th} bit of the n^{th} number.
Now, our method is
$k^i_n = W(k_{n-1}{}^{i-1}, k_n{}^{i-1}, k_{n+1}{}^{i-1})$.

But we know, k_{n-1} but not k_{n+1} and k_n.
Let a,b,c be 3 bits and F(a,b,c) = a ^ b ^ c (^ → bitwise xor).
Then, if we know a,

b	c	F(a,b,c)
0	0	a
0	1	~a
1	0	~a
1	1	a

Thus, if we predict a or ~a, the chance of our correctedness is 1/2 which is same as random guess.
Thus Cellular Automata is not acting as a security breach.

The main reason of using Cellular Automata here is to expand a 64 bit random number into 64 different random numbers. Wolfram's method is not itself a PRG here, but it acts as an expansion for k-bit random numbers into z (even z⩾k !!) k-bit random numbers.

NOTE := This is not a mathematical proof but my intiution behind the PRG. Thus, if there is any absurdity, ambiguity or errors feel free to communicate and/or raise a PR.

E Mail : varul.srivastava@research.iiit.ac.in
Github: https://github.com/vs666/CellAutomaton/tree/master/Applications/PRG
