Pseudo-random number generators with proven statistical properties

Boris Ryabko and Viacheslav Zhuravlev

September 15, 2020

Summary

Pseudo-random number generators (PRNGs) are widely used in data protection systems and are intensively researched in modern cryptography. In this report, we describe the PRNG class, which,

firstly, has been **successfully tested by the most powerful modern test batteries**, and secondly,
it is proved that they generate normal sequences, that is, for any generated sequence $x_1x_2\ldots$ **and any binary word** w

$$\lim_{t\to\infty} \nu_t(w)/(t-|w|) = 2^{-|w|}$$

where $\nu_t(w)$ is several occurrences of w in the sequence $x_1\ldots x_{|w|}$, $x_2\ldots x_{|w|+1}, \ldots, x_{t-|w|+1}\ldots x_t$).

The longer sequence, the better RNG!!!

Introduction-1

Pseudo-random number generators (PRNG) are designed to generate sequences of binary digits, which are distributed as a result of throwing an "honest" coin, or, precisely, obey the Bernoulli distribution with parameters $(1/2, 1/2)$. (Quite often this process and the sequences generated from them are called "truly random").

True random sequences are very desirable in cryptography, simulation, and modeling applications. Nowadays there are many so-called generators of pseudo-random numbers, whose aim is, informally speaking, to calculate sequences that mimic the truly random.

A modern PRNG is a computer program whose input is a short word (a so-called seed), whereas its output is a long (compared to the input) word. Having taken into account that the seed is a true random word, **the PRNG can be considered as an expander of randomness which stretches a short seed into a long word.**

Introduction- normal numbers

Researchers suggest and investigate PRNGs, which meet some "probabilistic" properties of truly random sequences. One of such properties is that a PRNG generates so-called normal (or ∞-distributed) sequences. The following definition of normal sequences belongs to Borel: A sequence of digits in base 2 is k-distributed if for any k-letter word w over the alphabet $\{0, 1\}$

$$\lim_{t \to \infty} \nu_t(w)/(t - |w|) = 2^{-|w|} \tag{1}$$

where $\nu_t(w)$ is a number of occurrences of w in the sequence $x_1 \ldots x_{|w|}$, $x_2 \ldots x_{|w|+1}, \ldots, x_{t-|w|+1} \ldots x_t$). The sequence is normal (or ∞-distributed) if it is k-distributed for any $k \geq 1$. Borel showed that almost all real numbers are normal That is why, the property that PRNG generates normal sequences is very desirable.

The problem

This paper aims to suggest such PRNGs that
i) generate normal sequences.
ii) generate *statistically* **acceptable sequences of any length.**

The first property was proved mathematically, and the second one was tested experimentally using batteries of statistical tests, which are currently considered the most powerful.

Theoretical background - two-faced processes

There are so-called two-faced processes which, on the one hand, generate normal sequences, and on the other hand, their entropy (in a letter) may be close to zero.

They are a promising tool for constructing a PRNG, because by definition any PRNG should have a small entropy per letter, and the ability to generate normal sequences is very desirable. The only problem is that the property of being a normal sequence is asymptotic, but real PRNGs should generate statistically acceptable sequences for any length.

Examples

The purpose of this part is an informal explanation of the basic ideas underlying the proposed PRNG, which are connected with so-called two-faced Markov chains.
First we consider several examples of the two-faced Markov chains.
Let a matrix of transition probabilities T_1 be as follows:

$$T_1 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & \nu & 1-\nu \\ 1 & 1-\nu & \nu \end{array} \quad , \tag{2}$$

where $\nu \in (0,1)$ (i.e. $P\{x_{i+1}=0|x_i=0\}=\nu$, $P\{x_{i+1}=0|x_i=1\}=1-\nu, ...$).

For example, let $\nu = 0.9$. Then, "typical" output sequence can be as follows:

$$0000000000\ 111111111\ 0000000000\ 1111111\ 0\ldots$$

(Here gaps correspond to seldom transitions). If $\nu = 0.1$, then "typical" output sequence is

$$01010101\ 1010101010\ 0101010101010101\ 1010\ldots\ .$$

On the one hand, those sequences are not truly random. On the other hand, the frequencies of 1's and 0's go to $1/2$ due to the symmetry of the matrix (2). Hence, the output is 1-distributed.

Example of two-faced sequence of $k = 2$.

$$000000000000\ 1101101101101101101101101\ 000\ldots$$

Now, we describe a family of processes that generate normal or ∞-distributed sequences.
Suppose that $m^* = m_1, m_2, ...$ is a sequence of integers, $m_1 < m_2 < m_3....$ and $X^1 = x_1^1 x_2^1 ...$, $X^2 = x_1^2 x_2^2 ...$, $X^3 = x_1^3 x_2^3 ...$, ... are (asymptotically) two-faced processes of order $m_1, m_2, ...$, correspondingly. Define a process $W = w_1 w_2$ ... by the following equation:

$$w_i = \begin{cases} x_i^1 & i \leq m_1, \\ x_i^1 \oplus x_i^2 & m_1 < i \leq m_2, \\ x_i^1 \oplus x_i^2 \oplus x_i^3 & m_2 < i \leq m_3, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots & \end{cases} \tag{3}$$

and denote it as $\bigoplus_{i=1}^{\infty} X^i$.

Theorem

Let all X^i, $i = 1, 2, ...,$ be two-faced. Then, $\bigoplus_{i=1}^{\infty} X^i$ is normal two-faced. If X^i, $j = 1, 2, ...$ are asymptotically two-faced, then $\bigoplus_{i=1}^{\infty} X^i$ is asymptotically normal two-faced.

Note that the proof immediately follows from Theorem 2.
The proposed PRNG construction repeats the process W in (3), but before describing it, we need to describe how to transform any random process into k -distributed for any integer k. It can be done as follows:

Theorem

Let there be an integer k, a finite word $x_{-k+1}x_{-k+2}...x_0$ and infinite sequence $u_1u_2...$, which is generated by a stationary ergodic source. Then, $u_1u_2...$ can be transformed into k-distributed $x_1x_2...$ as follows:

$$x_r = u_{r+1} \oplus \bigoplus_{i=r-k+1}^{r} x_i, \quad r = 1, 2, \ldots . \tag{4}$$

Now we can describe a PRNG which is an implementation of (3).

Input: 1. The desired length of the generating pseudo-random sequence (N).

2. A sequence of integers $m_1 < m_2 < ...m_s \leq N$, which are parameters of the method (to simplify the notation, it will be convenient to assume that all N/m_i are integers).

3. a seed of the PRNG, i.e. a sequence of random bits $r_1r_2...r_R$, where $R = m_1 + \sum_{i=1}^{s} M/m_i$.

Algorithm:

We generate sequences $x_1^0 x_2^0 ... x_N^0$, $x_{m_1+1}^1 x_{m_1+2}^1 ... x_N^1$, ..., $x_{m_s+1}^s x_{m_s+2}^s ... x_N^s$ according to (4), where $k^0 = m_1$, $x_{-m_1+1}^0 = r_1, x_{-m_1+2}^0 = r_2, ..., x_0^0 = r_{m_1}$,

$$u_i^0 = \begin{cases} 0, & if \quad i/m_1 \quad is\ not\ integer \\ r_{m_1+i/m_1}, & if \quad i/m_1 \quad is\ integer \end{cases},$$

$i = 1, 2, ..., N$,

$k^1 = m_2$, $x_{-m_1+1}^1 = x_1^0, x_{-m_1+2}^1 = x_2^0, ..., x_0^1 = x_{m_1}^0$,

$$u_i^1 = \begin{cases} 0, & if \quad i/m \quad is\ not\ integer \\ r_{m_1+N/m_1+i/m_2}, & if \quad i/m_2 \quad is\ integer \end{cases},$$

$i = 1, 2, ..., N$, ...,

$k^{s+1} = m_s$, $x_{-m_s+1}^s = x_1^{s-1}, x_{-m_s+2}^s = x_2^{s-1}, ..., x_0^s = x_{m_{s-1}}^{s-1}$,

$$u_i^s = \begin{cases} 0, & if \quad i/m \quad is\ not\ integer \\ r_{m_1+N/m_1+N/m_2+N/m_2+...+i/m_s}, & if \quad i/m_s \quad is\ integer \end{cases},$$

Finally, we calculate the output sequence $x_1^{output} = \bigoplus_{i=0}^{s} x_1^i, ...,$ $x_N^{output} = \bigoplus_{i=0}^{s} x_N^i$.

Experiments.

Here we describe the results of experiments that were carried out with described above PRNG. We tested the PRNG output by test from the well-known test batteries.
In our experiments, we took the PRNG under consideration with three parameters m_1, m_2, m_3, which were investigated for different values and different lengths of the output sequence. The results of applying the tests are presented in Table 1.

Table: Results of experiments.When using batteries "Alphabit" and "Rabbit" the length of the output sequence was 1 GB. (Other batteries define their strategy for using output sequences).

m_1	m_2	NIST	Rabbit	SmallCrush	Crush	BigCrush
31	257	Passed	Rejected	Passed	Rejected	Rejected
31	509	Passed	Passed	Passed	Rejected	Rejected
31	757	Passed	Passed	Passed	Rejected	Rejected
31	907	Passed	Passed	Passed	Passed	Rejected
31	1009	Passed	Passed	Passed	Passed	Passed
31	2251	Passed	Passed	Passed	Passed	Passed
31	3571	Passed	Passed	Passed	Passed	Passed
61	257	Passed	Rejected	Passed	Rejected	Rejected
61	509	Passed	Passed	Passed	Rejected	Rejected
61	757	Passed	Passed	Passed	Passed	Passed
61	907	Passed	Passed	Passed	Passed	Passed
61	1009	Passed	Passed	Passed	Passed	Passed

Our experiments have shown that Alphabet, Rabbit and NIST batteries are very weak. Finally, we experimented with the two most powerful batteries.

Table: Results of experiments.

m_1	m_2	BigCrush	pRand
127	257	Rejected	Rejected
127	509	Rejected	Rejected
127	1021	Passed	Rejected
127	2053	Passed	Rejected
127	16381	Passed	Passed
127	32749	Passed	Passed
127	65532	Passed	Passed

Conclusion

The results of these experiments show that there are values of the parameters m_1, m_2 for which the output sequences cannot be distinguished from truly random ones using the best modern battery tests. Ratio $|seed|/|ouput|$ may be around 0.001, which is acceptable for many cryptographic applications. In particular, the PRNG with parameters $m_1 = 1021$, $m_2 = 1021001$ can be recommended for practice.

So, we propose PRNGs that generate normal sequences (as far as the asymptotic properties can be valid for a real computer program). A series of experiments made it possible to find PRNG parameters for which the output sequence does not differ from truly random ones if we use the best batteries of statistical tests. This PRNG may be interesting for practice.

Thank you for attention!