A probabilistic approach to cryptographically secure pseudo-random generation

Hiroshi SUGITA
Department of Mathematics, Kyushu University
6-10-1, Hakozaki, Higashi-ku
Fukuoka, 812-8581, Japan
sugita@math.kyushu-u.ac.jp

1. Cryptographically secure pseudo-random generation: basic idea

We first introduce a basic idea of cryptographically secure pseudo-random generation. See [1,2] for details.

Let $B_m := \{0, 1, \dots, 2^m - 1\}$ be the set of all m-bit integers. For a function $f : B_m \to B_m$ and a *seed* $x_1 \in B_m$, we define a sequence $\{x_n\}_n$ by $x_{n+1} := f(x_n)$, $n \in \mathbf{N}$. Since x_n and x_{n+1} are strongly correlated by definition, this sequence does not have good pseudo-randomness. Now, we define a sequence of pseudo-random bits $\{y_n\}_n$ using a function $g : B_m \to B_1 = \{0, 1\}$ by

$$y_n := g(x_n) = g(f^{n-1}(x_1)), \qquad n = 1, 2, \dots, \tag{1}$$

where f^{n-1} stands for the $(n-1)$ times iteration of f. Let us consider the correlation between the first n outputs $\{y_1, \dots, y_n\}$ and the $(n+1)$-st output y_n. Suppose $n > m$ and that $\{y_1, \dots, y_n\}$ are given. Investigating all of (or a part of) possible paths, we can, in principle, discover (or estimate) the seed x_1 and can predict y_{n+1} (with probability $> \frac{1}{2} + \varepsilon$). But if we take a very complicated function g, it becomes practically impossible to predict y_{n+1} with probability $> \frac{1}{2} + \varepsilon$ on account of large computational complexity. Then, we say that the pseudo-random bits (**??**) are "not rejected by the next bit test".

An important fact is that pseudo-random bits which are not rejected by the next bit test are not rejected by *any feasible* test (See Theorem 4.1 of [1]). So those pseudo-random bits are said to be *cryptographically secure*.

Remark 1 In the field of cryptography, a general theory of cryptographically secure pseudo-random generation by means of one-way functions is well-known(c.f.[1,2]). But this theory is not practical. Instead, pseudo-random generation utilizing *hash functions* — complicated functions that compress long bits into short bits(e.g. SHA, MD5. See [2].) — as g in the formula (**??**) is widely implemented. But to this point, there is no established theory for the latter method.

2. Pseudo-random bit generator by means of irrational rotation

Definition 1 ([3]) For irrational $\alpha \in [0,1)$ and $m \in \mathbf{N}$, we define

$$Y_n^{(m)}(x;\alpha) := \left(\sum_{i=1}^{m} d_i\left(x+n\alpha\right)\right) \bmod 2, \qquad n=1,2,\ldots, \tag{2}$$

where $d_i(t)$ denotes the i-th bit(0 or 1) of $t \geq 0$ below the decimal point in the dyadic expansion.

In the implementation[1] of (**??**), we approximate real numbers by finite dyadic decimals. Suppose that we wish to generate the first 2^j terms of (**??**). To this end, we let $\tilde{x} := \lfloor 2^{m+j}x \rfloor \in B_{m+j}$ and $\tilde{\alpha} := \lfloor 2^{m+j}\alpha \rfloor \in B_{m+j}$. We then apply (**??**) with $f : B_{m+j} \to B_{m+j}$ and $g : B_{m+j} \to B_1 = \{0,1\}$ defined by

$$f(\tilde{x}) = f_{m+j}(\tilde{x}) := (\tilde{x}+\tilde{\alpha}) \bmod 2^{m+j}, \tag{3}$$

$$g(\tilde{x}) = g_{m,j}(\tilde{x}) := \left(\sum_{i=j+1}^{m+j} D_i(\tilde{x})\right) \bmod 2, \tag{4}$$

where $D_i(k)$ denotes the i-th bit of $k \in \mathbf{N}$, i.e., $D_i(k) := d_1(2^{-i}k)$. By this scheme, we can exactly generate the first 2^j terms of (**??**).

If we let m be larger and larger, the function $g = g_{m,j}$ becomes more and more complicated. Roughly speaking, this means that if we let m be larger and larger, it becomes more and more difficult to reject the pseudo-random bits (**??**) by the next bit test, in other words, they become more and more secure. A probabilistic approach to this phenomenon is done by Theorem **??** below[2].

Theorem 1 *([3]) For a.e. α, the stochastic process $\{Y_n^{(m)}(\,\cdot\,;\alpha)\}_{n=0}^{\infty}$ defined on the Lebesgue probability space $([0,1), P(dx) = dx)$ converges in law to the fair coin-tossing process as $m \to \infty$.*

Generally speaking, if we take a very complicated function g, it would be so difficult to compute the distribution of (**??**). In this context, to the best of our knowledge, the pseudo-random bits (**??**) is the only exception. Namely, each finite dimensional distribution

$$P\left(Y_j^{(m)}(\,\cdot\,;\alpha) = \epsilon_j,\ j=0,\ldots,k-1\right), \qquad k \in \mathbf{N}, \quad \epsilon_0,\ldots,\epsilon_{k-1} \in \{0,1\}, \tag{5}$$

can be explicitly computed by making use of Lemma **??** and Theorem **??** below. A great advantage of this fact is that we are able to estimate the quality of the pseudo-random bits (**??**) in advance without tests. See [3] for details.

---

[1] A sample code in C language is available, where we adopt the following parameters: $\alpha = (\sqrt{5}-1)/2$, $m = 90$ and $j = 60$. Make contact with *sugita@math.kyushu-u.ac.jp*.

[2] Theorem 1 has been widely extended with a farsighted proof by Yasutomi[4].

Lemma 1 *(i) Each of (??) can be derived from*

$$F^{(m)}(k_0,\dots,k_{l-1};\alpha) := P\left(\sum_{j=0}^{l-1} Y_{k_j}^{(m)}(\,\cdot\,;\alpha) = \mathrm{odd}\right), \quad 0 \le k_0 < \ldots < k_{l-1}, \quad l \in \mathbf{N}. \tag{6}$$

Indeed, for $\epsilon_n \in \{0,1\}$, we have

$$\begin{aligned} &P\left(Y_n^{(m)}(\,\cdot\,;,\alpha) = \epsilon_n, \quad n = 0,\dots,k-1\right) \\ &= 2^{-k}\left(\sum_{l=1}^{k} \sum_{0\le k_0<\ldots<k_{l-1}\le k-1} \prod_{j=0}^{l-1}(1-2\epsilon_{k_j})\,\left(1-2F^{(m)}(k_0,\dots,k_{l-1};\alpha)\right)+1\right). \end{aligned}$$

(ii) If $l \in \mathbf{N}$ is odd, $F^{(m)}(k_0,\dots,k_{l-1};\alpha) = 1/2$.
(iii) $F^{(m)}(k_0,\dots,k_{l-1};\alpha) = F^{(m)}(0,k_1-k_0\dots,k_{l-1}-k_0;\alpha)$. Hence we only need the case $k_0 = 0$.

Let $l \in \mathbf{N}$ be even. We introduce an algorithm to compute $F^{(m)}(0,k_1,\dots,k_{l-1};\alpha)$. For each irrational α, we put[3]C

$$\alpha_j := \{k_j\alpha\}, \quad j = 1,\dots,l-1,$$

and

$$\alpha_j^{(m)L} := \lfloor 2^m\alpha_j\rfloor/2^m, \quad \alpha_j^{(m)U} := \{\lfloor 2^m\alpha_j+1\rfloor/2^m\}, \quad \beta_j^{(m)} := 2^m(\alpha_j-\alpha_j^{(m)L}).$$

We next define a permutation $\sigma(m,\cdot)$ over the set $\{1,\dots,l-1\}$ by

$$1 > \beta_{\sigma(m,1)}^{(m)} \ge \beta_{\sigma(m,2)}^{(m)} \ge \ldots \ge \beta_{\sigma(m,l-1)}^{(m)} > 0.$$

For convenience, define $\beta_{\sigma(m,0)}^{(m)} := 1$, $\beta_{\sigma(m,l)}^{(m)} := 0$. Setting

$$\alpha_{\sigma(m,j)}^{(m),s} := \begin{cases} \alpha_{\sigma(m,j)}^{(m)U}, & (\,j \le s\,) \\ \alpha_{\sigma(m,j)}^{(m)L}, & (\,j > s\,) \end{cases}$$

we define

$$\boldsymbol{\alpha}^{(m),s} := (\alpha_1^{(m),s},\dots,\alpha_{l-1}^{(m),s}), \quad s = 0,1,\dots,l-1.$$

Finally we set

$$D := \bigcup_{m\in\mathbf{N}} \{i/2^m;\, i = 0,1,\dots,2^m-1\}.$$

Theorem 2 *It holds that*

$$F^{(m)}(0,k_1,\dots,k_{l-1};\alpha) = \sum_{s=0}^{l-1}\left(\beta_{\sigma(m,s)}^{(m)} - \beta_{\sigma(m,s+1)}^{(m)}\right) B(\boldsymbol{\alpha}^{(m),s}).$$

[3] $\{t\}$ denotes the fractional part of $t \ge 0$, i.e., $\{t\} := t - \lfloor t\rfloor$.

Here $B(\cdot)$ *is a real function defined on* $D^{l-1} = \overbrace{D \times \cdots \times D}^{l-1}$ *by*

$$\begin{cases} B(\boldsymbol{\alpha}^{(0),s}) &= 0, \qquad s = 0, 1, \ldots, l-1, \\ B(\boldsymbol{\alpha}^{(m),s}) &= \begin{cases} \dfrac{1}{2}B(\boldsymbol{\alpha}^{(m-1),s_2}) + \dfrac{1}{2}B(\boldsymbol{\alpha}^{(m-1),s_1+s_2}) & (\ s \text{ is even}\ ) \\ \dfrac{1}{2}\left(1 - B(\boldsymbol{\alpha}^{(m-1),s_2})\right) + \dfrac{1}{2}\left(1 - B(\boldsymbol{\alpha}^{(m-1),s_1+s_2})\right) & (\ s \text{ is odd}\ ) \end{cases} \end{cases}$$

where s_1, s_2 *are given by*

$$s_1 := \sum_{j=1}^{l-1} d_m(\alpha_j^{(m),s}), \qquad s_2 := \sum_{j=1}^{s} d_m(\alpha_{\sigma(m,j)}).$$

3. Special features

The pseudo-random bit generator (**??**) has outstanding probabilistic characteristics described in Theorem **??** and Theorem **??**. In addition, it has other special features as follows.

- Arbitrarily quick generation is possible by parallel computation. Namely, if K processors are available, let the j-th processor generate a subsequence $\{Y_n^{(m)}(x + (j-1)\alpha\,; K\alpha)\}_{n=0}^{\infty}$. The processes run completely independent of each other.
- It can instantly generate a very far (in future or past) bit y_n without computing the intermediate bits. It is because we can instantly compute $n\alpha$ or $n(1-\alpha)$ with high precision even for large n.

REFERENCES

[1] M.Luby, *Pseudorandmness and cryptographic applications*, Princeton Computer Science Notes, Princeton University Press, (1996).

[2] D.R.Stinson, *Cryptography (Theory and practice)*, CRC Press, Boca Raton/ Ann Arbor / London / Washington, D.C., (1995).

[3] H.Sugita, Pseudo-random number generator by means of irrational rotation, *Monte Carlo Methods and Applications*, VSP, **1-1** (1995), 35–57.

[4] K.Yasutomi, A limit theorem for sequences generated by Weyl transformation. Disappearance of dependence, in this volume.

RESUME

Cette nouvelle approche de "Pseudo-random number generator by means of irrational rotation" (*Monte Carlo methods and Appl.* **1-1**(1995), 35–57) présente un générateur pseudo-aléatoire cryptographiquement sûr qui admet une approche probabiliste. Ce générateur a des propriétés particulières, telle que: (1) les distributions marginales de dimension finie sont explicitement calculables, (2) une génération arbitrairement rapide est possible par calcul parallèle, et (3) un saut instantané arbitraire vers le futur ou le passé est possible.