L-011

CDM 系列と既存の擬似乱数生成器との比較
The Comparison of Continuously Decimalized M-sequence with other Existing Pseudo-Random Number Generator

裘　カイ‡　武田　祐樹‡　野上　保之‡　日下　卓也‡
Kai Qiu　Yuki Taketa　Yasuyuki Nogami　Tasuya Kusaka

1. はじめに

乱数生成器は、ハードウェア乱数生成器と擬似乱数生成器を含み、いずれもハードウェアまたはソフトウエアを用いて実現される。ハードウェア乱数生成器で生成された乱数は乱数性が高いが、その安定的な生成に難しさがある。本研究では、擬似乱数生成器を考える。

擬似乱数シーケンスは情報暗号化や、コンピューターシミュレーション、プロトコル認証などの多くの分野で広く使われている。乱数は暗号設計において最も重要なパラメータであり、強固な暗号技術を測定するための基準である。それだけではなく、これらの暗号技術のランダム性は全部暗号論的擬似乱数生成器により生成された鍵に基づく。

本稿は、従来の擬似乱数発生器の生成方法と新しい乱数生成法を対象とし、生成方法、生成速度とコスト、乱数性および統計的性質のテストを行い、各乱数生成器の比較を行う。

2. 擬似乱数生成器(PRNG)

擬似乱数生成器は確定的な計算アルゴリズムにより乱数を生成する。初期値として、短いハードウェア乱数を入力し、長いハードウェア乱数系列に近い振舞う乱数系列を生成する。入力する乱数はシード(seed)と呼ばれる[2]。擬似乱数生成器の中でも、暗号論的擬似乱数生成器は高い予測困難性を持つため、暗号技術に利用される。

2.1 線形合同法(LCG)

線形合同法とは、擬似乱数列の生成式の一つで、(1)の漸化式で示される。

$$X_{n+1} = (\mathrm{A} \times X_n + \mathrm{B}) \bmod \mathrm{M} \quad (1)$$

A、B、M は定数で、$\mathrm{M} > \mathrm{A}$、$\mathrm{M} > \mathrm{B}$、$\mathrm{A} > 0$、$\mathrm{B} \geq 0$である。線形合同法は特定な条件を満足すると、最大周期 M をもつ。定数による乗算と加算のみの演算で生成されるため、低機能なプロセッサにおいても高速に実装できる。しかし暗号論的擬似乱数生成器ではなく、乱数列の一部分から次のビットを予測することが可能である。

2.2 Blum-Blum-Shub(BBS)

BBS 生成器とは$x^2 mod n$ 生成器である。指定されたシードs_0は mod n の下で平方剰余である。それに mod n の下で連続的に平方計算を行い $s_1, s_2, \cdots, s_l$ を計算し、mod 2 を行うことで最下位ビットにより出力される 2 値系列 $z_1, z_2, \cdots, z_l$ は BBS 系列である。

$$z_i = \left(s_0^{2i} mod\, n\right) mod\, 2 \qquad 1 \leq i \leq l \quad (2)$$

n は大きな素数 p と q の積である。p と q はp ≡ q ≡ 3(mod)4 を満足する k/2 ビット素数である。BBS の安全性は n の分解の困難性に基づく。

RSA 暗号と同様に、大きな数の因数分解は非常に困難である。この性質のため、BBS 生成器が暗号を生成できる[3]。

2.3 RC4

RC4 は、ストリーム暗号の一つである。ストリーム暗号とは、PRNG により生成された乱数列そのものを用い逐次的に暗号化および復号を行う共通鍵暗号方式である。RC4 は、鍵を用いて共通鍵として擬似乱数系列の一部を生成し、平文を暗号化する。原理を図 1 に示す。

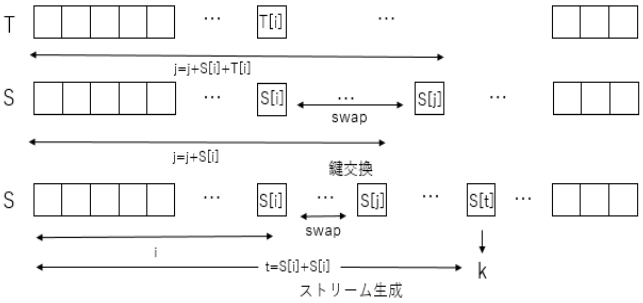

図 1　RC4

2.4 メルセンヌ・ツイスタ(MT)

まず M 系列を利用して Twisted General Feedback Shift Register(TGFSR)が提案された[4]。これは

$$x_{n+p} = x_{n+q} + x_n A \quad (p > q > 0) \quad (3)$$

を用いて乱数列を生成する。ここで A はツイスタと呼ばれる GF(2) の元を要素とする ω × ω の行列である。ただし $x_n \in GF(2)^{\omega}$である。TGFSR は M 系列と同様に nω ビットの記憶領域を使用し、$2^{p\omega} - 1$ の最大周期をもつ。

最大周期が $2^{19937} - 1$ であるとき、とくに MT19937 と呼ばれる。長い周期と高次元均等分布のため、python、C++などの言語でデフォルトの乱数生成器として利用されている。

2.5 CDM 系列

CDM 系列(Continuously Decimarized M-sequence)は M 系列とルジャンドル系列の生成手法を組み合わせた系列である[1]。M 系列の優秀なビット分布とルジャンドル系列の高い線形複雑性の特徴を併せ持つ。

M 系列のビット列を 10 進数とみなし。ルジャンドル符号を利用して、(4)の漸化式で CDM 系列を生成する。

$$S' = \{s_j'\}, s_j' = M_2((s_i, l)_{10}/p) \quad (4)$$

3. 実験環境と生成速度

本節では，実験に使用した環境とパラメータを示す。実験機器とパラメータの設定は観測される乱数列の性質に大きく影響を及ぼす。

‡ 〒700-8530 岡山大学, 岡山県岡山市北区津島中 3 丁目 1 番 1 号, Faculty of Engineering, Okayama University, 3-1-1, tsushi-manaka, Okayama, Okayama, 700-8530, Japan

3.1 実験機器

本実験のハードウェア環境は Windows 10 64 ビット、プロセッサは Intel(R)Core(TM) i5-8400。開発環境は VScode、使用言語は C++、ライブラリは NTL である。

3.2 生成速度

この実験では、同じ乱数生成器における異なるパラメータを使用した際の生成速度を比較して、表 1 に示す。その中で、*は設定されたパラメータでは周期が短いため、複数周期分生成している。

表 1　各乱数生成器の生成速度

ビット数 / 生成器	100mbit	1gbit	10gbit
LCG*	1.32s	13.3s	134.6s
LCG	1.62s	13.9s	151.5s
BBS*	0.21s	1.98s	10.2s
BBS	3.32s	33.13s	341.2s
RC4	7.35s	75.6s	747.1s
MT	0.35s	3.57s	37.0s
CDM	0.56s	5.69s	59.5s

4. 乱数性と統計的検定

暗号の分野においては暗号化に乱数を用いる場合、その乱数性の統計的性質が議論され、その性質は統計的検定を用いて評価される。各統計的検定により、各系列がハードウェア乱数系列の特徴の有無が確認できる。

4.1 NIST 検定

NIST 検定法は米国の NIST(National Institute of Standards and Technology)の研究者グループ開発した方法で、暗号分野で暗号化送信の乱数として用いるための乱数性の検定方法として、15 種類の検定法を提案している[5]。その結果は表 2 に示す[6]。

表 2　NIST 検定

NIST TEST	LCG	BBS	RC4	MT	CDM
Frequency Test	○	○	○	○	○
BlockFrequeny Test	○	○	○	○	○
CumulativeSums Test	○	○	○	○	○
Runs Test	○	○	○	○	○
Longest Run Test	○	○	○	○	○
Rank Test	○	○	○	○	○
FFT Test	○	○	○	○	○
NonOverlappintTemplate	×	○	○	×	○
OverlappingTemplate Test	○	○	○	○	○
Universal Test	○	○	○	○	○
ApproximateEntropy Test	○	○	○	○	○
RandomExcursions Test	○	○	○	○	○
RandomExcursionsVariant	○	○	○	○	○
Serial Test	○	○	○	○	○
LinearComplexity Test	○	○	○	×	○

NIST 検定の結果によると、RC4、BBS および CDM 系列は全て合格する。LCG とメルセンヌ・ツイスタはいくつかの検定に合格できない。

4.2 ビット分布

各系列のビット分布を表 3 に示す。

表 3　ビット分布

ビット	LCG	BBS	RC4*	MT*	CDM
0	32768	33249	32669	32625	32768
1	32768	33213	32867	32911	32767
00	16389	16628	16408	16315	16384
01	16379	16621	16261	16310	16384
10	16379	16621	16261	16310	16384
11	16389	16592	16606	16601	16383
000	8199	8327	8183	8174	8192
001	8190	8301	8225	8141	8192
010	8189	8275	8107	8130	8192
011	8190	8346	8154	8180	8192
100	8190	8301	8225	8141	8192
101	8189	8320	8036	8169	8192
110	8190	8346	8154	8180	8192
111	8199	8246	8452	8421	8191

ただし、RC4*と MT*系列の周期が長周期となり、測定が困難となるため、部分系列におけるビット分布を示す。

4.3 実験結果

本節では、乱数性の統計的検定である NIST 検定を用いた結果によると、LCG と BBS の周期が 1Gbit 未満のときそれぞれの乱数性は良くないといえる。ただし、BBS において使用する素数が十分大きいとき、NIST 検定に全て合格するが、その生成コストは大きくなる。RC4 は乱数性が良いといえるが、生成速度は遅く、ビットの分布性もよくない。メルセンヌ・ツイスタ系列は周期と分布が優れているが、予測困難性が低いという欠点もある。CDM 系列は、優れていることがわかった。しかし、CDM 系列の安全性について理論的な証明がなされていないため、現時点では LCG や MT と同様に暗号技術への応用に課題がある。

5. むすび

本稿では、いくつかの擬似乱数生成器の乱数性、生成速度、統計的な特徴を比較した。各乱数は、それぞれの長所と短所を持っている。そのため、状況によって、使用されている乱数生成器は異なる。この実験を通じて、現在広く使用されている擬似乱数生成器の特徴を捉えることにより、改善が期待できる。

参考文献

[1] 武田 祐樹,小寺 雄太,日下 卓也,野上 保之, “線形複雑度が可変な擬似乱数系列の生成法”, 第 4 回有限体理論とその擬似乱数生成への応用ワークショップ, pp. 14-18 (2018).
[2] Divyanjali, Ankur. V., “An Overview of Cryptographically Secure Pseudorandom Number Generators ans BBS” vol. ICACEA, pp. 19–28, (2014)
[3] B. Shen, X. F. Dong, B.J. Xu, Y. Zhou, “Random Number Generators and Their Applications in Cryptography” , pp.9-11, (1986)
[4] Makoto Matsumoto, “Twisted GFSR Generators”(1992)
[5] 藤井 光昭, “暗号と乱数: 乱数の統計的検定”, 共立出版, pp. 20-22, (2018)
[6] E. Barker, J. Kelsey, “Recommendation for Random Number Generation Using Deterministic Random Bit Generators.” NIST, SP 800-90A, https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final. (2015)