■1 群（信号・システム）- 3 編（暗号理論）

11 章 擬似乱数

（執筆者：小柴健史）［2008 年 12 月 受領］

■概要■

擬似乱数は様々な応用分野で利用されているが，通常は暗号分野で利用される擬似乱数は他分野で利用される擬似乱数よりも厳しい条件が求められる．擬似乱数の暗号用途として例をあげるならば，ストリーム暗号，暗号系・署名系の鍵生成，暗号文・署名生成などがある．暗号において擬似乱数は一般に「真性乱数」の代替手段として利用される．そのため，擬似乱数の性能が十分でないと，用いられる暗号システムの安全性に悪影響を与える可能性がある．本章では，擬似乱数の定義と理論的側面に言及した後，実際的にいくつかの擬似乱数生成法について触れる．

【本章の構成】

擬似乱数の定義（11-1 節），擬似乱数の基礎理論（11-2 節），擬似乱数の生成法（11-3 節），擬似乱数性の検定（11-4 節）について述べる．

■1 群 - 3 編 - 11 章

11-1 擬似乱数の定義

（執筆者：小柴健史）[2008 年 12 月 受領]

擬似乱数の性質の前に，真の乱数の性質について言及する．あるランダムな対象を連続的に生成していく過程を考えたとき，ある時点までに得られている対象列が既知であるとき，次の対象が何であるかを有意に予測することはできない．それが一様ランダムな系列の定義である．この定義を考えると，コンピュータで一様ランダムな系列が発生できないのはもちろん，コンピュータでできることの対局に位置していることが分かる．「対象列が既知である」ということはコンピュータの計算状況の列が既知であることに対応する．コンピュータの計算においては計算状況の列が決まれば次の計算状況は一意である．一方で，コンピュータがランダムな対象を出力できるとすれば，ランダムな対象の候補数だけの次の計算状況が存在しないといけない．これは上で述べた一意性と相反する要求である．つまり，コンピュータでは乱数列を出力することは原理上不可能で，代替として「擬似」乱数列をコンピュータで出力する方法が擬似乱数生成法である．コンピュータの内部状態を完全に開示する状況では出力系列は一意に決定できるので「ランダムに見える」系列を出力するには何らかの情報を秘密に保つ必要がある．擬似乱数生成法とは何らかの秘密情報を保ちつつ「ランダムに見える」出力を行う計算過程を定める方法である．擬似乱数列を出力する過程においてコンピュータ内部には秘密情報の系列が存在することになり，その秘密情報のことを擬似乱数列の**内部状態**と呼び，内部状態系列の初期内部状態を擬似乱数の**種**と呼ぶ．種が決まれば出力系列が一意に決定するので，種は秘密でなければならず，かつ，ランダムである必要がある．ランダムな種はコンピュータでは生成不可能なので，ランダムであると思われる何らかの物理的な手段で種を決定するしかない．種が十分に長ければ擬似乱数列を出力する意味がないので，種の長さは通常は短いことを想定する．

「ランダム」の定義は上述したが，より形式的な定義を与えよう．これは「ランダムに見える」ことを形式的に定義するためでもある．乱数を集合上の一様分布と定義する．コンピュータで扱えるように n ビットからなるビット列全体の一様分布を U_n とする．乱数を分布として定義することにより，その性質を統計的に議論することができる．例えば，U_n にしたがって生成される n ビットのビット列に現れる 1 の個数の期待値は $n/2$ である．この 1 の個数が平均的に $n/2$ よりも大きく偏っていた場合は一様な乱数でないと考えることができる．このような乱数の統計的な性質を利用して乱数性を判定するという考え方が一般的な乱数性判定法であり，詳細については後述する．

形式的な定義として，すべての統計的な性質を判定した結果，すべて合格となったものを擬似乱数とすることも可能である．ただ，すべての統計的テストが効率的に実行できるとは限らない．そこで，ヤオ（Yao）は多項式時間で実行可能なすべての統計的テストに合格する分布を擬似ランダムな分布として定義した [19]．また，彼は次ビット予測テストと呼ばれる概念を導入し，次ビット予測テストに合格すれば，そのほかの多項式時間実行可能は統計的テストも合格すること，つまり，次ビット予測テストの万能性を証明した．現実的な観点からは，次ビット予測テストは過去のビット系列をどのように利用するかの自由度が多過ぎて実際のテストとしては利用できないという問題点がある．一方で，理論的計算機科学の観点からは非常に価値のある結果である．このようにして定義される擬似乱数を**暗号学的擬似乱数**

（cryptographically secure pseudorandomness）と呼び，形式的には以下のように定義される．まず，関数 g を考え，関数 g は入力サイズごとに以下を満たす写像になっているとする．

$$g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$$

ただし，$\ell(n)$ は n の多項式で，g を計算する決定性多項式時間アルゴリズムが存在するとする．このような関数 g が**暗号学的擬似乱数生成器**（cryptopgrahically secure pseudorandom generator）であるとは，任意の確率的多項式時間アルゴリズム $\mathcal{D}$ と任意の正多項式 p に対して十分大きな任意の n で，

$$\left|\Pr[\mathcal{D}(g(U_n)) = 1] - \Pr[\mathcal{D}(U_{\ell(n)}) = 1]\right| < \frac{1}{p(n)}$$

を満たすときをいう．上の式で，U_n は等確率で $\{0,1\}^n$ の値をとる確率変数で，$g(U_n)$ や $\mathcal{D}(g(U_n))$ も自然に確率変数となる．この定義が意味しているのは多項式時間で実行可能なすべての統計テストに対して擬似乱数を入力したときと真性乱数を入力とした場合の確率の差が無視できるほど小さいということである．

上の暗号学的擬似乱数生成器の定義と冒頭で述べた「内部状態を更新していく」方法との関係について言及する．上の定義では g は n ビットの入力 x_0 に対して出力長は少なくとも $n+1$ ビットである．$g(x_0)$ の前半 n ビットを x_1，後半を b_1 とする．この x_1 に対して g を適用し $g(x_1)$ の前半 n ビットを x_2 とし後半を b_2 とする，のように再帰的に繰り返すことができる．ここで $b_1, b_2, \ldots$ が擬似乱数ビットとなることが証明されている．また，x_0 が擬似乱数の種に対応し，$x_1, x_2, \ldots$ が内部状態の変遷列に対応する．これは同時に，入力長 n ビットで出力長が $n+1$ ビットの暗号学的擬似乱数生成器があれば，任意の多項式長の出力をもつ暗号学的擬似乱数生成器が構成可能であることも示している．

■1 群 - 3 編 - 11 章

11-2 擬似乱数の基礎理論

（執筆者：小柴健史）[2008 年 12 月 受領]

電子署名や公開鍵暗号などの暗号プロトコルは一方向性関数の存在性を前提としており，暗号学的擬似乱数についても同様である．この節では暗号学的擬似乱数生成器の存在性について見ていくことにする．

関数 f が**一方向性関数**（one-way function）であるとは，

- n ビット入力 x に対して $f(x)$ を計算する多項式時間アルゴリズムが存在し，
- 出力分布 $f(U_n)$ に対して，任意の確率的多項式時間アルゴリズム $\mathcal{A}$ と任意の正多項式 p に対して，十分大きな任意の n で，

$$\Pr[f(\mathcal{A}(f(U_n))) = f(U_n)] < \frac{1}{p(n)}$$

となるときをいう．つまり，一方向性関数は，順方向計算は容易だが，その逆計算はできたとしても無視できる割合程度しか成功しない．一方向性関数 f の入力を n ビットに限定した関数を f_n とする．f_n の任意の関数値に対して逆像の個数が（n に依存して）一定であるとき，f を**正則**一方向性関数という．f_n の出力長が一定のとき，f を**長さ正則**な一方向性関数という．特に，入力長と出力長が一致するとき f を**長さ保存**な一方向性関数という．f が長さ保存で一対一関数のとき，f を一方向性**置換**（one-way permutation）という．一方向性置換はその定義から正則一方向性関数の特殊な場合でもある．

任意の一方向性関数について，ハードコア述語とよばれる予測が困難なビット関数が存在することが証明されており，ゴールドライヒ-レビン（Goldreich-Levin）定理[9]と呼ばれる．形式的にはハードコア述語は以下のように定義される．$b_n : \{0,1\}^n \to \{0,1\}$ が**ハードコア述語**（hard-core predicate）であるとは

- n ビット入力 x に対して $b_n(x)$ を計算する多項式時間アルゴリズムが存在し，
- 任意の確率的多項式時間アルゴリズム $\mathcal{A}$ と任意の正多項式 p に対して，十分大きな任意の n で，

$$\Pr[\mathcal{A}(f_n(U_n)) = b_n(U_n)] < \frac{1}{2} + \frac{1}{p(n)}$$

を満たすときをいう．つまり，ハードコア述語は関数値から有意に予測が困難なビットである．

今，$f_n(x)$ を一方向性置換とし，$b_n(x)$ をそのハードコア述語とすると，ヤオ（Yao）の次ビット予測テストの万能性[19]の議論により，$g_n(x) = f_n(x)\|b_n(x)$ は暗号学的擬似乱数生成器となる．ただし，$u\|v$ はビット列 u と v の連接を表す．この手法で暗号学的擬似乱数生成器を構成することはブラム-ミカリ-ヤオ（Blum-Micali-Yao（BMY））方式[2, 19]と呼ばれ，多くの具体的な構成方法がとっている方法である．

正則一方向性関数から暗号学的擬似乱数生成器を構成する手法にゴールドライヒ-クラウチック-ルビー（Goldreich-Krawczyk-Luby）方式[8]と呼ばれる方法があり，k-対独立ハッシュ関数族の性質を巧みに利用して構成法を得ている．ハイトナー-ハーニック-レインゴールド

（Haitner-Harnik-Reingold）[10]により，対独立ハッシュ関数族でも十分であることが確認されている．一般の一方向性関数から暗号学擬似乱数生成器を構成する手法にはハスタッド-インパグリアッツォ-レビン-ルビー（Håstad-Impagliazzo-Levin-Luby（HILL））方式[11]がある．この結果により，一方向性関数と暗号学的擬似乱数生成器の存在の等価性が証明されたことになる．HILL 方式の中で暗に利用されている擬似エントロピー対と呼ばれる概念はハードコア述語を一般化した概念であるが，ホレンシュタイン（Holenstein）[12]やハイトナー-ハーニック-レインゴールド（Haitner-Harnik-Reingold）[10]により，効率の良い擬似エントロピーの構成が効率の良い暗号学的擬似乱数生成器につながることが示されている．

■1 群 - 3 編 - 11 章

11-3 擬似乱数の生成法

（執筆者：小柴健史）[2008 年 12 月 受領]

本節では具体的にいくつかの擬似乱数生成法を紹介する．まず，計算困難と予想される問題への帰着が存在するような方式について言及する．続いて，そのような帰着が証明されているわけではないが現実的に多用されている方式について言及する．

11-3-1 計算困難問題への帰着をもつ擬似乱数生成法

この範ちゅうに属する擬似乱数生成法の多くは構成の容易さから BMY 方式に基づくものが多い．BMY 方式をとる代表的な擬似乱数生成法として ブラム-ミカリ（Blum-Micali）法[2], RSA 法, ブラム-ブラム-シュブ（Blum-Blum-Shub（BBS））法[3] を紹介する．また，BMY 方式に基づかない構成法としてジェナロ（Gennaro）法[7]を紹介する．そのほかにもインパグリアッツォ-ナオア（Impagliazzo-Naor）法[13], フィッシャー-スターン（Fischer-Stern）法[6] など興味深い様々な擬似乱数生成法が考案されているがここでは割愛する．

【ブラム-ミカリ（Blum-Micali）法】
p を素数とする．g を Z_p^* のもとで大きな位数をもつものとする．x_0 を Z_p^* からランダムに選び，種とする．以下のような数列を考える．

$$x_j = g^{x_{j-1}} \bmod p, \qquad y_j = \begin{cases} 1 & \text{if } x_j \geq (p-1)/2 \\ 0 & \text{if } x_j < (p-1)/2 \end{cases}$$

内部状態の変遷が x_j の系列で，出力系列が y_j で表されている．内部状態の一回の更新につき，1 ビットの出力を行う方式である．ブラム-ミカリ（Blum-Micali）法の安全性は離散対数問題への帰着可能性である．擬似乱数列の生成効率を決める要素は，内部状態の更新の計算コストと内部状態から抽出できる最大ビット数である．冪乗計算は平均で $n/2$ 回の乗算が必要なため，ブラム-ミカリ（Blum-Micali）法は現在では効率が悪い方式に位置づけられてしまっているが，何らかの証明可能性をもつ最初の方法であり，それ以降の基礎を与えている意味で重要である．

【RSA 法】
p, q を素数とし，$N = pq$, $\phi = (p-1)(q-1)$ とする．n は N のビット数．また，e を $1 < e < \phi$ の範囲で，$\gcd(e, \phi) = 1$ を満たすものを選ぶ．x_0 を Z_N^* からランダムに選び，種とする．以下のような数列を考える．

$$x_j = (x_{j-1})^e \bmod N, \qquad y_j = \mathrm{lsb}_1(x_j)$$

ただし，$\mathrm{lsb}_1(x)$ は x の最下位ビットを表す．RSA 法の安全性として RSA 関数の逆計算問題への帰着可能性である．RSA 法は RSA 暗号の考案者であるリベスト（Rivest），シャミア（Shamir），エイドルマン（Adleman）によるものではなくアレクシ-コール-ゴールドライヒ-シュノア（Alexi-Chor-Goldreich-Schnorr）[1]による RSA 関数の最下位ビットのハードコア性の研究と BMY 方式の帰結として導かれる．ブラム-ミカリ（Blum-Micali）法と比較して

効率の面で有利な点がある．RSA 暗号の暗号化関数の効率化手法の一つとして，e を小さめにとるという方法があるが，同様なことが可能であり，一回の状態更新に必要な乗算回数を $n/2$ よりも減少させることが可能である．

【ブラム-ブラム-シュブ（Blum-Blum-Shub）法】
p と q を mod 4 で 3 と合同な素数とし，$N = pq$ とする．Z_N^* からランダムに s を選び，$s^2 \bmod N$ を種 x_0 とする．以下のような数列を考える．

$$x_j = (x_{j-1})^2 \bmod N, \qquad y_j = \mathrm{lsb}_1(x_j)$$

BBS 法の安全性は N の素因数分解問題への帰着可能性である．状態更新関数が乗算 1 回で実現できるためにブラム-ミカリ（Blum-Micali）法や RSA 法よりも効率的である．

【ジェナロ（Gennaro）法】
p を mod 4 で 3 と合同な素数とする．c は $\omega(\log n)$ を満たす数とする．s と g を Z_p^* からランダムに選ぶ．$\hat{g} = g^{2^{n-c}} \bmod p$ を計算し，$x_0 = \hat{g}^{\lfloor s/2^{n-c} \rfloor} g^{\mathrm{lsb}_1(s)} \bmod p$ を種とする．以下のような数列を考える．

$$x_j = \hat{g}^{\lfloor x_{j-1}/2^{n-c} \rfloor} g^{\mathrm{lsb}_1(x_{j-1})} \bmod p, \qquad y_j = \mathrm{msb}_{n-c}(\mathrm{lsb}_{n-c+1}(x_j))$$

ジェナロ（Gennaro）法の安全性は短い冪の離散対数問題への帰着可能性である．特筆すべきは，ジェナロ（Gennaro）法は BMY 方式に基づかない新奇方式による構成法であることである．

11-3-2 発見的手法に基づく擬似乱数生成法

単に擬似乱数生成法といった場合，発見的手法に基づくアプローチで構成法が与えられていることになり，古くから存在する多くの方法がこの範ちゅうに属する．実際に用いるためには安全面の考慮が必要で何らかの評価がなされていることが望ましい．暗号技術評価事業（CRYPTREC）[14]によって推奨されている方法として以下がリストに記載されている．

1. ANSI X9.42-2001 Annex C.1 記載の擬似乱数生成法
2. FIPS PUB 186-2 (+ change notice 1) Appendix 3.1 記載の一般用の擬似乱数生成法
3. FIPS PUB 186-2 (+ change notice 1) revised Appendix 3.1 記載の擬似乱数生成法

FIPS PUB 186-2 は電子署名標準（DSS）に関する文書であり，その付録は，DSS のために利用されることを前提としている．DSS はアルゴリズムの内部で，署名のための鍵生成や署名作成に乱数を利用するが，これらの乱数入力を生成するための方法として，付録に擬似乱数生成法が記載されている．以下では，一般用の擬似乱数生成法として利用できる 2 番目の方法について説明する．

【FIPS PUB 186-2 (+ change notice 1) Appendix 3.1 記載の一般用の擬似乱数生成法】
入力: k, s_i $(1 \le i \le m)$, 出力: x_i $(1 \le i \le m)$
for $i \leftarrow 1$ **to** m **do**

$$x_i \leftarrow G(IV, (k + s_i) \bmod 2^b); \quad k \leftarrow (1 + k + x_i) \bmod 2^b$$

end for

ただし, 関数 $G(t, c)$ は一方向性関数であることが期待される関数で DES ベースのものと SHA-1 ベースのものが記載されている．SHA-1 ベースの $G(t, c)$ は $\{0,1\}^{160} \times \{0,1\}^b \rightarrow \{0,1\}^{160}$ の関数で SHA-1 の初期ベクトルを t に，メッセージパディング方式を単に右 0 パディングに変更したものである．

■1 群 - 3 編 - 11 章

11-4 擬似乱数性の検定

（執筆者：小柴健史）［2008 年 12 月 受領］

擬似乱数は実行可能な任意の統計的テストに合格するのが望ましいが，これは現実的ではないので，基本的であると思われるテスト（検定法）セットを用意して判定するという方法が一般的である．代表的なものとして FIPS PUB 140-2 [4], NIST SP800-22 [18], DIEHARD [17] やクヌース（Knuth）[16] の方法がある．テストセットごとに選択した検定法の選択基準が不明確であったり，同じ検定法でもテストセットによってパラメータが異なっていたりする．特に，FIPS PUB 140-2 や NIST SP800-22 には不具合が報告されている[14]．それを受けて CRYPTREC によって最小テストセットが策定されている[15]．既に前述したが，統計的検定という方法では，安全性能が著しく劣るものを弾くことしか期待できず，検定に合格したからといって安全性の保証が与えられるものではないことに留意して利用すべきである．

■参考文献

1) W. Alexi, B. Chor, O. Goldreich and C. P. Schnorr, “RSA and Rabin functions: Certain parts are as hard as the whole,” SIAM J. Comput., vol.17, no.2, pp.194-209, 1988.
2) M. Blum and S. Micali, “How to generate cryptographically strong sequences of pseudo-random bits,” SIAM J. Comput., vol.13, no.4, pp.850-864, 1984.
3) L. Blum, M. Blum and M. Shub, “A simple unpredictable pseudo-random number generator,” SIAM J. Comput., vol.15, no.2, pp.364-383, 1986.
4) FIPS PUB 140-2, “Security requirements for cryptographic modules,” Federal Information Processing Standards Publication 140-2, U.S. Department of Commerce/N.I.S.T., National Technical Information Service.
5) FIPS PUB 186-2, “Digital signature standard,” Federal Information Processing Standards Publication 186-2, U.S. Department of Commerce/N.I.S.T., National Technical Information Service.
6) J.-B. Fischer and J. Stern, “An efficient pseudo-random generator provably as secure as syndrome decoding,” In Proc. EUROCRYPT ’96, Lecture Notes in Computer Science 1070, pp.245-255 1996.
7) R. Gennaro, “An improvemed pseudo-random generator based on discrete logarithm problem,” J. Cryptol., vol.18, no.2, pp.91-110, 2005.
8) O. Goldreich, H. Krawczyk and M. Luby, “On the existence of pseudorandom generators,” SIAM J. Comput., vol.22, no.6, pp.1163-1175, 1993.
9) O. Goldreich and L.A. Levin, “A hard-core predicate for all one-way functions,” In Proc. the 21st ACM Symposium on Theory of Computing, pp.25-32, 1989.
10) I. Haitner, D. Harnik and O. Reingold, “On the power of the randomized iterate,” In Proc. CRYPTO ’06, Lecture Notes in Computer Science 4117, pp.22-40, 2006.
11) J. Håstad, R. Impagliazzo, L.A. Levin and M. Luby, “A pseudorandom generator from any one-way function,” SIAM J. Comput., vol.28, no.4, pp.1364-1396, 1999.
12) T. Holenstein, “Pseudorandom generators from one-way functions: A simple construction for any hardness,” In Proc. 3rd Theory of Cryptography Conference, Lecture Notes in Computer Science 3876, pp.443-461, 2006.
13) R. Impagliazzo and M. Naor, “Efficient cryptographic schemes provably as secure as subset sum,” J. Cryptol., vol.9, no.4, pp.199-216, 1996.
14) 情報処理振興事業協会, 通信・放送機構, “暗号技術評価報告書（2002 年度版）,” CRYPTREC Report 2002.

15) 情報通信研究機構, 情報処理推進機構, “暗号技術監視委員会報告書（2004 年度版）,” CRYPTREC Report 2004.
16) D.E. Knuth, “The Art of Computer Programming,” Addison Wesley, Vol.2: Seminumerical Algorithms（Third Edition）, 1998.
17) G. Marsaglia: DIEHARD, http://stat.fsu.edu/pub/diehard.html
18) A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” NIST Special Publication 800-22, U.S. Department of Commerce/N.I.S.T., National Institute of Standards and Technology 2000.
19) A.C. Yao, “Theory and applications of trapdoor functions,” In Proc. the 23rd IEEE Symposium on Foundations of Computer Science, pp.80-91, 1982.