TOYOCRYPT-HR1

評価報告書

記号

本稿では、以下の記号・略称・用語を説明無しに用いる。

LFSR	：線形フィードバックシフトレジスタ
LC	：線形複雑度
固定鍵	：LFSR の状態遷移関数を定義するパラメータ
ストリーム鍵	：LFSR の初期内部状態を定義するパラメータ
$\oplus$	：排他的論理和
$\otimes$	：論理積
・	：内積
F_{2^n}	：2^n 個の元からなる有限体

1. はじめに

本稿は、政府調達暗号公募に対して応募された擬似乱数生成器 TOYOCRYPT-HR1 の安全性について評価を行ったものである。検討項目は、周期、線形複雑度、およびその他の統計的な性質である。検討の結果、ビット出力の等頻度性、周期について提案者の行っている解析は妥当であることが検証された。また、統計的な性質について、いくつかの実験的な検証評価を行ったが、特に問題は見つからなかった。

2. アルゴリズム概略

TOYOCRYPT-HR1 の内部状態を

$X = x_{127} \,||\, x_{126} \,||\, \ldots \,||\, x_0$

$\quad = x_{127} \,||\, x_{126} \,||\, X_h$

$X_L = x_{125} \,||\, x_{124} \,||\, \ldots \,||\, x_{63}$

$X_R = x_{62} \,||\, x_{61} \,||\, \ldots \,||\, x_0$

とし、特に時刻 t を明記する必要がある場合には、$x_i(t)$と記述する。

TOYOCRYPT-HR1 の乱数生成部は、レジスタ長 128 のフィルター型生成器である。出力フィルターfは、

$$f(X) = x_{127} \oplus h(X_h)$$
$$= x_{127} \oplus (X_L \bullet \pi(X_R) \oplus g(X_R))$$

で与えられる。ここで、πは 63 ビットの置換、gは 64 次 3 項のブール多項式である。

固定鍵は状態遷移関数が M 系列を出力するように定め、ストリーム鍵は 0 以外の値を取る。また、初期化として、ストリーム鍵をレジスタにロードした後、256 回の状態遷移を行う。

3. 評価結果

本章では、TOYOCRYPT-HR1 の安全性について理論的、数値的な検証を行った結果について述べる。

擬似乱数生成器の出力する列は、

- (1) 鍵長に見合う長い周期を持つこと
- (2) いくつかの統計的性質(0/1 等頻度性、自己相関性)で偏りが小さいこと
- (3) 大きな線形複雑度を持つこと

を満たすことが求められる。以下、節ごとにそれぞれの項目について論じる。

3.1. 周期

TOYOCRYPT-HR1 では、LFSR が M 系列を生成するように固定鍵を選ぶので、LFSR 部の周期は $2^{128}-1$ である。LFSR 部の周期の中で 1 の出現する回数は

$$\sum_{x\in_2^{128}} f(X) = \sum_{x\in F_2^{128}} h(X_h)\oplus x_{127}$$
$$= 2\sum_{X_h\in F_2^{126}}((h(X_h)\oplus 0)+(h(X_h)\oplus 1))$$
$$= 2\cdot 2^{126} = 2^{127}$$

となる。固定鍵 *C*、ストリーム鍵 *S* に対する TOYOCRYPT-HR1 の出力列の周期を Per(*C, S*)とおけば、

$$k\cdot Per(C,S) = 2^{128}-1$$

を満たす自然数 *k* が存在する。Per(*C, S*)の間に出力される 0、1 の出現数をそれぞれ N_0、N_1 とすれば、

$$k\cdot N_0 = 2^{127}-1$$
$$k\cdot N_1 = 2^{127}$$

が成り立つので、*k*=1 となる。すなわち、Per(*C, S*)=2^{128}-1 である。

3.2. 統計的性質

3.2.1. 1/0 等頻度性

3.1.で述べたように、TOYOCRYPT-HR1 の周期中に 1 が出力される回数は 2^{127} 回である。すなわち、TOYOCRYPT-HR1 の出力列は 0/1 を理想的な頻度で出力する。

3.2.2. 数値的な検証

線形フィードバックシフトレジスタ自体は理想的な乱数を出力するが、非線形フィルターによる影響を理論的に完全に検証することは困難である。ここでは、いくつかの乱数性指標について、実験的に検証した結果について述べる。

FIPS 140-1[1]では、乱数列が最低限満たすべき性質として、1 ビット、4 ビットの頻度検定、および連の検定に合格することが挙げられている。今回は、FIPS 140-1 にもとづき、これらの検定を TOYOCRYPT-HR1 の出力列(4K～16M バイト)に対して実施した。検定には χ^2 検定を用いた。

また、固定鍵、ストリーム鍵は C 言語標準ライブラリの rand 関数を用いてランダムに与えた。

1 ビットの頻度検定(mono-bit test)

0/1 の出現頻度は、自由度 1 の χ^2 分布を用いて検定することができる。表 3.1に自由度 1 の χ^2 値の代表的な値を挙げる。

表3.1 自由度 1 の χ^2 分布の値

棄却率	χ^2 値
0.05	3.8
0.01	6.6
0.001	10.8

検定は、4K, 64K, 1M, 16M バイトの乱数列それぞれ 256 個に対して行った。表 3.2の値は、これらの列に対し、0/1 頻度の χ^2 統計量を計算し、棄却率 0.05, 0.01, 0.001 を超えたものの数である。

表 3.2 1 ビットの頻度検定

列の長さ	0.05	0.01	0.001
4K バイト	8/256	2/256	0/256
64K バイト	10/256	6/256	0/256
1M バイト	5/256	1/256	0/256
16M バイト	9/256	2/256	0/256

実験の結果、いずれの長さの乱数列についても、χ^2 値が極端に大きくなることはなかった。また、出力列が「真の乱数ではない」となる頻度についても許容できる範囲である。

4 ビットの頻度検定(poker test)

0/1 の出現頻度は、自由度 1 の χ^2 分布を用いて検定することができる。表 3.3に自由度 15 の χ^2 値の代表的な値を挙げる。

表 3.3 自由度 15 の χ^2 値

棄却率	χ^2 値
0.05	25.0
0.01	30.6
0.001	37.7

検定は、4K, 64K, 1M, 16M バイトの乱数列それぞれ 256 個に対して行った。表 3.3は、これらの列に対し、4 ビット単位の出現頻度の χ^2 統計量を計算し、棄却率 0.05, 0.01, 0.001 を超えたものの数である。

表 3.4 4 ビットの頻度検定

列の長さ	0.05	0.01	0.001
4K バイト	10/256	2/256	0/256
64K バイト	7/256	1/256	0/256
1M バイト	15/256	0/256	0/256
16M バイト	17/256	2/256	2/256

実験の結果、いずれの長さの乱数列についても、χ^2 値が極端に大きくなることはなかった。また、出力列が「真の乱数ではない」となる頻度についても許容できる範囲である。

連検定(run test)

連の検定では、有効な数値を取るため、出現頻度の低い長さの連を検定に用いない。ここでは、乱数列の長さを m ビットとしたとき、$\log_2(m)-2$ を検定を行う連の長さの下限とした。長い連は、出現する期待値が 1/256 以下のものとした。

表 3.5 連検定

列の長さ	自由度	0.05	0.01	長い連(最長値)
4K バイト	24	9/256	3/256	2(24)
64K バイト	32	9/256	4/256	3(26)
1M バイト	40	7/256	4/256	3(31)
16M バイト	48	13/256	4/256	3(37)

実験の結果、いずれの長さの乱数列についても、χ^2 値が極端に大きくなることはなかった。また、出力列が「真の乱数ではない」となる頻度についても許容できる範囲である。長い連が出現する確率は期待値に比べて多い傾向にあるが、問題となるほどではない(FIPS による長すぎる連の定義は出現する期待値が 1/1000000 であり、このような連は観測されていない)。

3.2.3. 自己相関性

周期 N のビット列 s に対する自己相関関数を、
で定義する。

$$A(s,t) := \sum_{i=0}^{N-1} (-1)^{s_i + s_{i+t}},\ 0 \le t \le N-1$$

$t \neq 0$ ならば、$A(s, t)$の値は-1 であることが望ましい。しかし、非常に長い周期を持つ乱数列についてこれを検証することは困難である。ここでは、近似的に、出力列を t ビットシフトした列とを排他的論理和した列の 0/1 頻度検定を行った結果について述べる。表 3.6は、TOYOCRYPT-HS1 の出力列の自己相関性を χ^2 検定した結果で、各項目の数値は、256 個の固定鍵、ストリーム鍵のペアをランダムに与え、棄却率 0.01 を超えたものの数である。

表 3.6 シフト数 1〜10 の自己相関性

乱数列の長さ	シフト数 t									
	1	2	3	4	5	6	7	8	9	10
4K バイト	4/256	4/256	3/256	3/256	4/256	2/256	2/256	1/256	1/256	3/256
64K バイト	2/256	1/256	2/256	0/256	4/256	4/256	5/256	2/256	2/256	5/256

実験の結果、出力列が「真の乱数ではない」となる頻度は許容できる範囲であると言える。

3.3. avalanche 性

TOYOCRYPT-HR1 のフィルター関数 f は、

$$f(X) = h(X_L, X_R) \oplus x_{127}$$

で定義されるので、次の性質を持つ。

- $f(x_0, x_1, \ldots, x_{126}, \overline{x_{127}}) = \overline{f(x_0, x_1, \ldots, x_{126}, x_{127})}$
- $f(x_0, x_1, \ldots, x_{126}, \overline{x_{127}}) = f(x_0, x_1, \ldots, x_{126}, x_{127})$

一方、論理積において $\overline{a}b = \overline{ab}$, $\overline{a}\overline{b} = \overline{ab}$ となる確率はいずれも 1/2 であるから、レジスタ内部のビットのうち、x_{126}, x_{127} 以外のビットが反転している場合には出力ビットが反転するかどうかを知ることはできない。
ストリーム鍵 S のある 1 ビットを反転したものを S' とし、それぞれの出力列を $s(t)$, $s'(t)$ と置く。TOYOCRYPT-HR1 の状態遷移関数は線形写像であるから、差分の列 $s(t) \oplus s'(t)$ もまた M 系列となる。

以上の考察から、ストリーム鍵の任意の 1 ビットを反転した場合、出力列のあるビットが反転するか否かを推定することは困難である。

3.4. 入力空間のサイズ

LFSR の状態遷移は拡大体の指数演算+トレースで表現できる。このため、スクリーニング評価の際に付されたコメントにもあるように、固定鍵 C を固定した場合、2^{64} 個所の内部状態 X(0), X(2^{64}), ..., X((2^{64}-1)2^{64})を初期値とした一定長の出力を事前に計算しておき、このデータと実際の出力系列との衝突を測定すれば、2^{64} ビット程度の出力列で衝突が起こることになる(これは、すべてのフィルター型生成器に言えることである)。この事実がただちに攻撃に結びつくわけではないが、攻撃者に固定鍵の情報が漏れれば、2^{64} 程度の計算量でストリーム鍵を決定することができるため、TOYOCRYPT-HR1 を乱数生成に用いる場合には、2^{64} ビット程度を出力するごとに固定鍵を取りかえることが望ましい。

3.5. 推測可能性

LFSR の場合、入力情報の一部が漏れたとしても、256 ラウンド後のレジスタの内部状態を推定することは困難であり、入手情報以上の情報量を手に入れることはできない。

一方、TOYOCRYPT-HR1 はシードの選択以外は確定的なアルゴリズムであるため、シードが漏れた場合には、出力列を完全に確定することができる。このため、シードは予測不可能な値を取る必要がある。したがって、付記情報にあるような、キータイピング情報やネットワークのトラフィック情報など、攻撃者が取得可能と思われる情報をシードに使うことは望ましくない。

4. まとめ

評価内容からは実用上問題となるような点は見つからなかった。しかし、近年の暗号技術評価では、提案されているアルゴリズムが「現実的に解読可能か？」ではなく、「鍵の全数探索より効率的な攻撃法はあるか？」ということを検討する傾向にある。このような観点から見ると、TOYOCRYPT-HR1 は H/W での実装を軽量にするために、セキュリティマージンをほとんどとっていない設計である。また、レジスタ幅を鍵長ぎりぎりのサイズにとっているため、3.4に述べたような攻撃法が存在し得る。

5. 参考文献

[1] FIPS 140-1, *Security Requirements for Cryptographic Modules*, Federal Information Processing Statndard(FIPS), Publication 140-1, National Institute of Standards adn Technology, US Department of Commerce, Washington D.C., January 1994, http://www.itl.nist.gov/fipspubs/index.htm.

[2] Menezes, Alfred J., van Oorschot, Paul C., Vanstone, Scott A., *HANDBOOK of APPLIED CRYPTOGRAPHY*, CRC Press, 1997.

[3] R. A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag, 1986.