EPC Gen2標準仕様RFIDタグ向けの擬似乱数生成器の安全性解析と改良案の提案

野間口　広[1]　宮地　充子[1,2,3]　蘇　春華[3]

概要：ユビキタス社会やIotの実現に向け，スマートデバイスのセキュリティとプライバシーの保護の重要性が増してきている．なかでもセキュリティのコアプリミティブである擬似乱数生成器は重要である．しかし，既存の擬似乱数生成器では，省リソースで動作することが求められるスマートデバイスにそのまま適応できない．このため，スマートデバイスに搭載可能な擬似乱数生成器が必要である．本研究では，NLFSRとDLFSRを組み合わせ，十分な安全性を持ち，統計的にも優れた擬似乱数生成器を提案し評価する．

キーワード：NLFSR, 擬似乱数生成器, RFID, EPC Gen2

Evaluation and Improvement of Pseudo-Random Number Generator for EPC Gen2 RFID Tags

Hiroshi Nomaguchi[1]　Atsuko Miyaji[1,2,3]　Chunhua Su[3]

Abstract: RFID enable applications are ubiquitous in our society, especially become more and more important as IoT management rises. Meanwhile, the concern of security and privacy of RFID is also increasing. The pseudorandom number generator is one of the core primitives to implement RFID security. Therefore, it is necessary to design and implement a secure and robust pseudo-random number generator (PRNG) for current RFID tag. In this paper, we study the security of light-weight PRNGs for EPC Gen2 RFID tag which is an ISO international standard. Based on our analysis, we propose an new scheme which outperform the existing PRNGs for EPC Gen2 RFID tag. We build our PRNG with a combination of NLFSR and DLFSR, and achieve more efficiency and security. We also show that our proposed PRNG has good randomness and passed the NIST randomness test.

Keywords: NLFSR, Pseudo-random number generator, RFID, EPC Gen2

1. はじめに

RFIDをはじめとするスマートデバイスはIoTやユビキタス社会の主な構成要素の一つである「通信機能」を有する．特にRFIDは様々な活用や応用が考えられ，多様なニーズにこたえられるスマートデバイスの一つと期待されている．このような背景から，RFIDのセキュリティとプライバシーの保護の重要性が増してきている．

なかでも将来的に暗号通信におけるの鍵の生成などに利用され，セキュリティのコアプリミティブとなる可能性のある擬似乱数生成器は重要である．しかし，既存の擬似乱数生成器では，省リソース，省電力で動作することが求められるスマートデバイスにそのまま利用することができない．このため，スマートデバイスに搭載可能な省リソース（2000GE以下），省電力 [1][2][3] で動作し，プリミティブ

[1] 北陸先端科学技術大学院大学 情報科学研究科
〒 923-1292 石川県能美市旭台 1-1
Japan Advanced Institute of Science and Technology
[2] 独立行政法人科学技術振興機構
〒 332-0012 埼玉県川口市本町 4-1-8 川口センタービル
CREST
[3] 大阪大学大学院工学研究科 電気電子情報工学専攻
〒 565-0871 大阪府吹田市山田丘 2-1
Osaka University

として利用できる十分な安全性を有する擬似乱数生成器が必要であるが，Warbler は（32，2，5，6）[4] と（62，3，5，6）[5] と発表されているが鍵長が短く，J3Gen[6] は出力から鍵の一部が復元されることが示されている [7].

本研究では，既存のスマートデバイス向けの擬似乱数生成器をベースに非線形部フィードバックシフトレジスタ（以後，NLFSR）と動的フィードバックシフトレジスタ（以後，DLFSR）を組み合わせ，セキュリティのコアプリミティブとしての利用に必要な鍵長を有し，十分な安全性を持ち，統計的にも優れた擬似乱数生成器を提案し評価する．

2. 既存研究

EPC Gen 2 は省リソース，省電力で動作することが求められる．既存研究においては，べき乗算と Trace を利用した非線形フィードバックシフトレジスタ（以後，NLFSR）をベースとした Warbler と，複数の多項式を乱数 r により切り替える動的フィードバックシフトレジスタ（以後，DLSFR）をベースとした J3Gen がある．これらについて説明する．

2.1 記号の定義

本稿では以下のように記号を定義する．

田	: 算術加算
$+$	: 排他的論理和
f	: 原始多項式
f_j	: i 番目の原始多項式
Select	: フィードバックで使用する原始多項式の格納先
Select(j)	: Select より j 番目の原始多項式を選ぶ
ζ, λ, μ	: NLFSR のレジスタ．添え字 i はレジスタ番号を示す
a	: NLFSR6 のレジスタ．各レジスタは 5 ビット
lf	: DLFSR のレジスタ番号
l	: DLFSR の原始多項式，乱数 r, rm の切り替えの周期
t	: 5 ビットのメモリ．添え字 i は i ビット目を示す

2.2 Warbler

Warbler は，1 ビット 1 レジスタの NLFSR を三つと拡大体を利用する 5 ビット 1 レジスタの NLFSR 一つから構成される．提案方式で利用する Warbler(62,3,5,6) のアルゴリズムを図 1 内に示す．Warbler(62,3,5,6) の内部状態は 92 ビットで，鍵長は 60 ビットである．三つの NLFSR は，それぞれ最大周期 ($2^n - 1 : n$ は LFSR の段数) をとるように設計され，各 NLFSR の周期は互いに素となっているため全体としての周期は約 2^{62} となっている．5 ビットの NLFSR は，6 段の NLFSR となっており，複数の NLFSR からの出力を拡大体における乗算に利用することにより，さらなる非線形化を行う．Warbler を構成する WG，NLFSR19，21，22，NLFSR6，メモリの順序で説明する．まず，WG

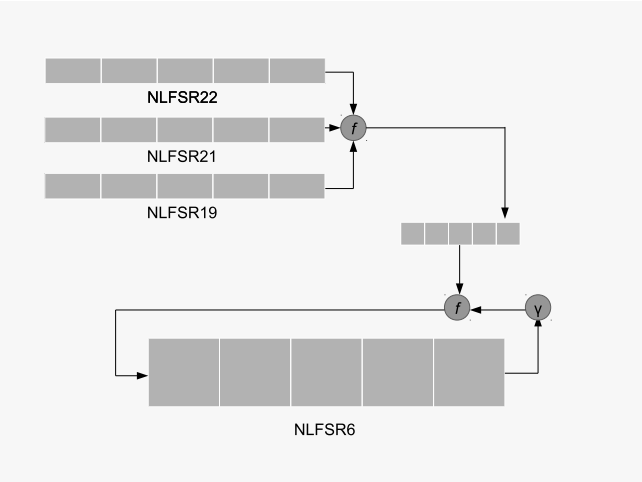

図 1 Warbler(62,3,5,6)

について説明する．WG は，ガロア体におけるべき乗算を行う WGP と Trace から構成される．ガロア体の法はそれぞれ

NLFSR22 : $f_1 = x^5 + x^3 + x + 1$,
NLFSR21: $f_2 = x^5 + x^4 + x^2 + x + 1$,
NLFSR19: $f_3 = x^5 + x^4 + x^3 + x^2 + 1$
である．まず，WGP から説明する．入力を $x(x \in F_2^5)$ とする．法 f_1，f_2，f_3 のもと WGP は次式で示される．

$$\mathrm{WGP}(x) = x + (x+1)^5 + (x+1)^{13} + (x+1)^{19} + (x+1)^{21}.$$

次に，Trace について説明する．Trace は 5 ビットの入力を 1 ビットにする．Trace に対する入力を $x(x \in F_2)$ とする．Trace は次式で示される．

$$\mathrm{Trace}(x) = x + x^2 + x^{2^2} + x^{2^3} + x^{2^4} \ (F_2^5 \to F_2)$$

WGP と Trace により，WG は次のように示される．

$$\mathrm{WG}(x)=\mathrm{Trace}(\mathrm{WGP}(x^d)) \ (x \in F_2^5)$$

次に，NLFSR19,21,22 について説明する．NLFSR19，21，22 はそれぞれ 1 ビットのレジスタが 19，21，22 段から構成される線形フィードバックシフトレジスタ（以後，LFSR）である．周期は，n 段の LFSR と同じく，$2^n - 1$ である．各レジスタを μ，λ，ζ とすると状態遷移は次式で表される．

$$\zeta_{k+22} = 1 + \zeta_k + \mathrm{WG}(y^7),$$
$$y = (\zeta_{k+3}, \zeta_{k+4}, \zeta_{k+8}, \zeta_{k+12}, \zeta_{k+20}),$$
$$\lambda_{k+21} = 1 + \lambda_k + \mathrm{WG}(y^{11}),$$
$$y = (\lambda_{k+4}, \lambda_{k+10}, \lambda_{k+12}, \lambda_{k+15}, \lambda_{k+20}),$$
$$\mu_{k+19} = 1 + \mu_k + \mathrm{WG}(z^7),$$
$$z = (\mu_{k+3}, \mu_{k+6}, \mu_{k+14}, \mu_{k+16}, \mu_{k+18})$$

次にメモリについて説明する．メモリは 5 ビットであり，NLFSR19,21,22 の出力を f を通しそれを保存する役割を持つ．メモリの動作を下記に示す．

$$f(x_0, x_1, x_2) = x_0x_1 + x_1x_2 + x_0x_2 + x_0 + x_1 (x_{0,1,2} \in F_2),$$
$$s_i = f(\zeta_i, \lambda_i + 1, \mu_i + 1), s_j = 0, j = 0, 1, 2, 3 \ ,$$
$$t_{i+4} = (s_i, s_{i+1}, s_{i+2}, s_{i+3}, s_{i+4}) \ ,$$

次に NLFSR6 について説明する．NLFSR6 は，5 ビット 6 段の NLFSR(a) である．NLFSR6 は，NLFSR19，21，22 の出力を保存したメモリも利用し，状態遷移を行うとともに拡大体における乗算を行う．動作を下記に示す．

$a_{k+6} = \gamma a_k + a_{k+1} + w_k + t_k,\ w_k = (0,0,0,0,WG(a_k^{11}))$

Warbler の出力 O は下記の式で表される．

$O_{k+1} = WG(a_{k+5}^3)$

2.3 J3Gen

J3Gen は，外部乱数を利用し動的にフィードバックを変える DLFSR ベースの擬似乱数生成器である．アルゴリズムを図 2 に示す．LFSR とフィードバックを決める Polynomial Selector により構成される．本提案方式においては，LFSR は 16 段，使用する原始多項式の数は 8 を利用するので，その場合の構成を Polynomial Selector，LFSR の順で説明する．Polunomian Selector は LFSR が使用する原始多項式を切り替える機能を持つ．ラウンド (l) ごとに外部乱数 $r(r \in \{0,1\})$ を取り込み，LFSR が利用する原始多項式 fb_i を選ぶ．その切り替えを下式に示す．

$fb_i = Select(j)$

$j = j + r \pmod 8$

$Select = f_1, f_2, \cdots, f_8$

$f_j : j = 1, 2, \cdots, 8$

LFSR(lf_i) は，1 ビット 1 6 段のものを使用する．その状態遷移は下式のとおり．

$lf_{i+1} = fb_i(lf_i)$

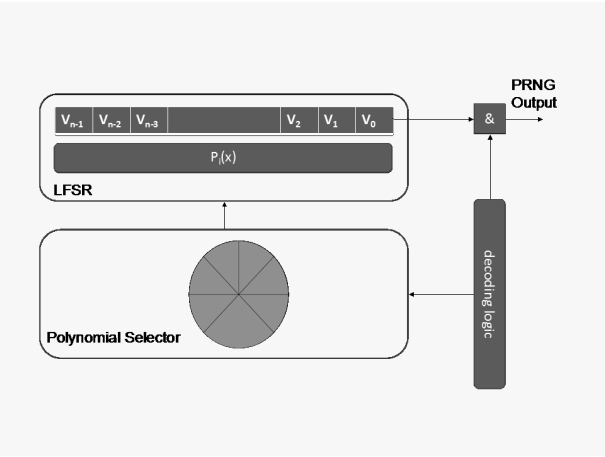

図 2 J3Gen

3. 既存研究の安全性解析

Warbler，J3Gen の安全性解析について述べる．

3.1 Warbler の安全性解析

Gangqiang Yang 氏らは代数攻撃，Resistance Against Algebraic Attacks およびその発展攻撃，Weak Internal States and Fault Injection Attacks などにも耐性を有するとしている．現在のところ攻撃は報告されていない．Warbler は，NLFSR6 の出力を次式のとおり WG に通す形で出力している．

$O_{k+1} = WG(a_{k+5}^3)$

このため，時刻 $k+1$ における出力 O_{k+1} が 0,1 のどちらかわかれば，その時の NLFSR6 の内部状態 a_{k+5} は 2^4 で特定できる．これは，NLFSR6 の内部状態を圧縮する形で出力しているためである．また，NLFSR6 と，メモリの $t_{1,2,3,4}$ 推測した状態であれば，次の時刻の出力から，メモリに新たに加わる 1 ビットが定まる．このため，NLFSR6 の状態遷移に関わる変数等は，計算量的安全性がより高いほうが望ましい．次に NLFSR19，21，22 に着目する．NLFSR19，21，22 の出力を次式のとおり f に入力しすることにより出力を得る．

$f(x_0, x_1, x_2) = x_0x_1 + x_1x_2 + x_0x_2 + x_0 + x_1 (x_{0,1,2} \in F_2)$

この入力と出力関係の真理値表は表 1 の通り．表 1 より

		x_2	
		0	1
(x_0, x_1)	(0,0)	0	0
	(0,1)	1	0
	(1,0)	1	0
	(1,1)	1	1

表 1 f の真理値表

f が 0 を出力する場合 $x_2 = 1$ である確率が 3/4，$x_2 = 1$ である確率が 1/4，1 を出力している場合は $x_2 = 0$ の確率が 3/4 であり，$x_2 = 1$ である確率が 1/4 である．同様に f の出力から，x_0，x_1 にも偏りがある．f は 1/3 で x_0，x_1，x_2 のいずれかの出力を無視することによりその時の入力の一部を隠す効果もあると考えられるが，このような偏りは攻撃に利用される可能性があるため，偏りなく計算量的安全性を保つ別の関数に置き換える方が望ましいと考える．次に回路規模について考察する．回路規模 937GE である Warbler(32,2,5,6) に比べ，内部状態が 27 増えた Warbler(62, 3, 5, 6) の回路規模は 1238GE である．鍵長増加のために単純に内部状態を増やすのでは，EPC Gen 2 の制限である 2000GE を超える可能性がある．

3.2 J3Gen の安全性解析

J3Gen の安全性解析について述べる．J3Gen は出力から使用している原始多項式を復元する攻撃 [7] が報告されている．この攻撃により，176bit の出力から使用している原始多項式を 8 個中 7 個復元した．Joan Meli'a-Segu´ らは，使用する原始多項式は秘密にすることにより，J3Gen の鍵として利用できると考えていたので，出力から鍵の一部を復元することができることになる．ただし，J3Gen への攻撃は，多数の出力を必要とする．逆に言えば，多数の

出力が無ければ使用している原始多項式の復元はできない．次に，回路規模について考察する．J3Gen の 16 段シフトレジスタ，8 個の原始多項式利用時の回路規模は外部乱数のためのパーツである Decoding Logic module，TRNG を加えても 439.1GE であり省リソースで実装可能である．

4. 提案方式

本研究では，スマートデバイスなど省リソースの環境に実装でき，計算量的安全性を有する擬似乱数生成器を構成するため，図 3 のような構造をとる．このように設計したのは，Warbler を大きな鍵長に対応させるためにそのまま内部状態を大きくするより，J3Gen とうまく組み合わせる方が少ないリソースの増加で可能であると考えたためである．この際，J3Gen は Warbler 側の出力を得ることができれば使用している原始多項式を復元されてしまうので，Warbler 側の最低保証周期が鍵長に近いことが望ましいと考えた．本提案方式は，NLFSR 群により確実な周期を保持し，外部乱数を NLFSR6 と DLFSR に取り込むことにより，環境により周期が変動するように構成する．また，NLFSR 群と NLFSR6，DLFSR を組み合わせることにより，線形複雑度を向上させ鍵長と同等の計算量的安全性の向上を図る．NLFSR 群，NLFSR6，DLFSR の順に説明する

NLFSR 群は，Warbler のものより内部状態を拡大させた 3 つの NLFSR で構成され，周期は約 2^{80} である．また，その出力は，WG により非線形化される．各パーツごとに示す．WG は，Warbler のものを参照する．WG は，ガロア体におけるべき乗算を行う WGP と Trace から構成し．ガロア体の法はそれぞれ

NLFSR28:$f_1 = x^5 + x^3 + x + 1$,

NLFSR27:$f_2 = x^5 + x^4 + x^2 + x + 1$,

NLFSR25:$f_3 = x^5 + x^4 + x^3 + x^2 + 1$

である．続いて，WG で使用する WGP を説明する．入力を $x(x \in F_2^5)$ とする．法 f_1，f_2，f_3 のもと WGP は次式で示される．

$\mathrm{WGP}(x) = x+(x+1)^5+(x+1)^{13}+(x+1)^{19}+(x+1)^{21}$.

次に，Trace について説明する．Trace は次式で示される．

$\mathrm{Trace}(x) = x + x^2 + x^{2^2} + x^{2^3} + x^{2^4}$ $(F_2^5 \to F_2)$

WGP と Trace により，WG は次のように示される．

$\mathrm{WG}(x)$=Trace(WGP(x^d)) $(x \in F_2^5)$

次に，WG を利用する NLFSR について説明する．NLFSR25, 27, 28 はそれぞれ 1 ビットのレジスタが 25(μ), 27(λ), 28(ζ) 段から構成される非線形フィードバックシフトレジスタである．周期は，n 段の LFSR と同じく $2^n - 1$ であり．各 NLFSR の周期は互いに素になるように設計してある．各状態遷移は次式で表される．

$\zeta_{k+28} = 1 + \zeta_k + \mathrm{WG}(x^7)$,

$x = (\zeta_{k+3}, \zeta_{k+4}, \zeta_{k+8}, \zeta_{k+12}, \zeta_{k+20})$,

$\lambda_{k+27} = 1 + \lambda_k + \mathrm{WG}(y^{11})$,

$y = (\lambda_{k+4}, \lambda_{k+10}, \lambda_{k+12}, \lambda_{k+15}, \lambda_{k+20})$,

$\mu_{k+25} = 1 + \mu_k + \mathrm{WG}(z^7)$,

$z = (\mu_{k+3}, \mu_{k+6}, \mu_{k+14}, \mu_{k+16}, \mu_{k+18})$ メモリは，NLFSR らの出力を保存しておく 5 ビットのメモリ (t) である．Warbler においては NLFSR19,21,22 の出力を f を通しそれを保存していたが，本提案においては WG を利用し次のように動作する．表記における d_i は DLDSR の 16 レジスタ (lf_{k+16})，rm_i は時刻 i における外部乱数を示す．

$s_i = WG(\zeta_i, \lambda_i + 1, \mu_i + 1, rm_i, d_i), s_j = 0, j = 0, 1, 2, 3$,

$t_{i+4} = (s_i, s_{i+1}, s_{i+2}, s_{i+3}, s_{i+4})$

本提案において，NLFSR は次のように動作する．NLFSR6 は，5 ビット 6 段の NLFSR(a) である．NLFSR は，拡大体における乗算を行う．動作を下記に示す．

$a_{k+6} = \gamma a_k + a_{k+1} + w_k + t_k$, $w_k = (0, 0, 0, 0, WG(a_k^{11}))$

DLFSR 部は，外部乱数を利用し動的にフィードバックを変える J3Gen をベースとしている．J3Gen と同じく LFSR とフィードバックを決める Polynomial Selector により構成される．Polynomial Selector は LFSR が使用する原始多項式を切り替える機能を持つ．ラウンド (l) ごとに外部乱数 $r(r \in \{0, 1\}$ を取り込み，LFSR が利用する原始多項式 fb_i を選ぶ．その切り替えを下式に示す．

$fb_i = Select(j)$

$j = j \boxplus r \pmod 8$

$Select = f_1, f_2, \cdots, f_8$

$f_j : j = 1, 2, \cdots, 8$ LFSR(l_i) は，1 ビット 1 6 段のものを使用する．その状態遷移は下式のとおり．

$lf_{i+1} = fb_i(lf_i)$

NLFSR6 と DLFSR より，本提案方式における出力 (O_k) は下式で表される．$O_k = WG(a_{k+5}^3) + l_{k+15}$

5. 処理ステップ

処理ステップについて，初期化，疑似乱数生成ステップに分け説明する．

本提案方式においては，96bit の鍵と 35bit の IV を利用し初期化する．なお，DLFSR で使用する原始多項式は，回路ごとに焼きこむため 16 次の原始多項式 2048 個から 8 個選んですでに焼きこんでいるものとする．鍵を $Key = K_0, K_1, K_2 \cdots, K_{95}$，初期ベクトル IV を $IV = IV_0, IV_1, IV_2 \cdots IV_{34}$ で表す．鍵，IV の配置は Algorithm1，初期化は Algorithm2 のとおり．

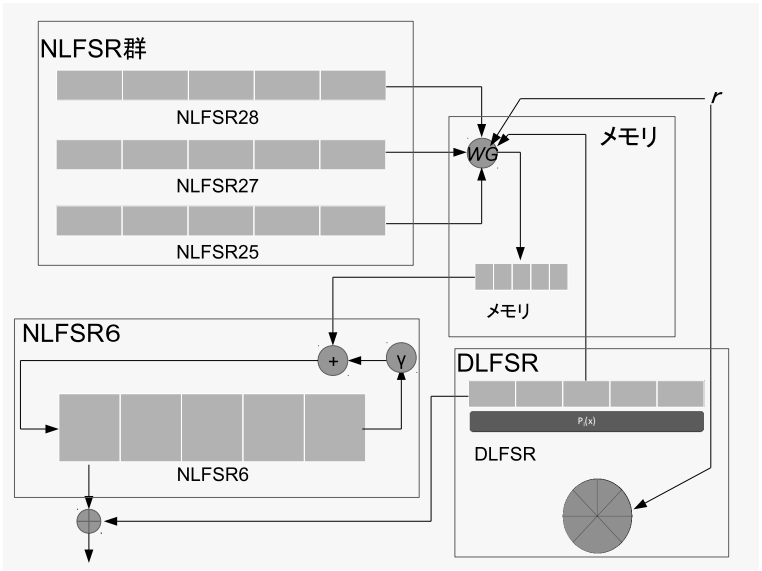

図 3 提案アルゴリズム

Algorithm 1 鍵，IV 配置

for $i = 0$ to 27 **do**
  $\zeta_i = K_i$
end for
for $i = 0$ to 26 **do**
  $\lambda_i = K_{i+28}$
end for
for $i = 0$ to 24 **do**
  $\mu_i = K_{i+55}$
end for
for $i = 0$ to 15 **do**
  $lf_i = Key_{i+80}$
end for
for $i = 0$ to 4 **do**
  $t_i = IV_i$
end for
for $i = 0$ to 5 **do**
  $a_i = \{IV_{i+5}, IV_{i+6}, IV_{i+7}, IV_{i+8}, IV_{i+9}\}$
end for

Algorithm 2 初期化

  $j = 0$
for $i = 0$ to 55 **do**
  $\zeta_{i+28} = 1 + \zeta_i + \mathrm{WG}(y^7) + WG(a^3_{i+5})$
  $\lambda_{i+27} = 1 + \lambda_i + \mathrm{WG}(y^{11}) + WG(a^3_{i+5})$
  $\mu_{i+25} = 1 + \mu_i + \mathrm{WG}(z^7) + WG(a^3_{i+5})$
  $s_i = WG(\zeta_i, \lambda_i + 1, \mu_i + 1, rm_i, d_i), s_j = 0, j = 0, 1, 2, 3$
  $t_{i+4} = (s_i, s_{i+1}, s_{i+2}, s_{i+3}, s_{i+4})$
  $w_i = (0, 0, 0, 0, WG(a_i^{11}))$
  $a_{i+6} = \gamma a_i + a_{i+1} + w_i + t_i$
  if $i/l = 0$ **then**
    $j = j \boxplus r \pmod 8$
    $fb_k = Select(j)$
  end if
  $lf_{i+1} = fb_i(lf_i)$
end for

擬似乱数生成ステップは，初期化処理終了後より，Algorithm3 のとおり擬似乱数を出力する．

Algorithm 3 生成ステップ

  $j = 0$
for $i = 0$ to ∞ **do**
  $\zeta_{i+28} = 1 + \zeta_i + \mathrm{WG}(y^7)$
  $\lambda_{i+27} = 1 + \lambda_i + \mathrm{WG}(y^{11})$
  $\mu_{i+25} = 1 + \mu_i + \mathrm{WG}(z^7)$
  $s_i = WG(\zeta_i, \lambda_i + 1, \mu_i + 1, rm_i, d_i), s_j = 0, j = 0, 1, 2, 3$
  $t_{i+4} = (s_i, s_{i+1}, s_{i+2}, s_{i+3}, s_{i+4})$
  $w_i = (0, 0, 0, 0, WG(a_i^{11}))$
  $a_{i+6} = \gamma a_i + a_{i+1} + w_i + t_i$
  if $i/l = 0$ **then**
    $j = j \boxplus r \pmod 8$
    $fb_k = Select(j)$
  end if
  $lf_{i+1} = fb_i(lf_i)$
  $O_i = WG(a^3_{k+5}) + lf_{i+15}$
end for

6. 評価

本章においては，安全性評価および乱数検定の結果を評価する．

6.1 安全性評価

DLFSR のベースとなる J3Gen は，176bit の出力から使用している原始多項式を 8 個中 7 個復元された．J3Gen においては，使用している原始多項式さえ復元できれば外部乱数を予想するだけで，次の出力がわかるため問題であった．

本提案方式においては，Warbler ベースの NLFSR6，NLFSR 群，メモリと組み合わせることにより，DLFSR の出力がそのまま得られない．このため，Warbler ベースの

NLFSR 等の内部状態を推測せずに DLFSR で使用している原始多項式を得るには，Warbler 側の出力を 100bit 以上の出力を推測する必要がある．これは，鍵長以上に推測する必要があるため，現実的ではない．

ベースとした Warbler は，特に攻撃方法などは報告されていない. そこで，DLFSR と組み合わせることにより，LFSR を利用している擬似乱数生成器やストリーム暗号などに有効な推測決定攻撃 [8]，[9] など出力を利用する攻撃に耐性を持つようにした．

DLFSR と組み合わせることにより，まず，DLFSR の出力を推測してその上で Warbler ベースの NLFSR 群，NLFSR6，メモリを攻撃しなければならない．例を挙げると，Warbler ベースのみであれば，出力から NLFSR6 のその時の s_{k+5} の内部状態が 2^4 の試行でわかるが，DLSFR と組み合わせることにより，$2^4 \times 2^1 = 2^5$ の計算量が必要となる．このため，推測決定攻撃など出力を利用する攻撃は難しいと考える． また，周期については，外部乱数を取り込むことにより，NLFSR6 に特定の周期が存在しなくなるため，周期を利用した攻撃も困難であると考える．

一方，Warbler では NLFSR 群に相当する部位で f を利用していたが，本提案方式では外部乱数 r と DLSFR を加え WG を利用するので，Warbler 側の内部状態総当たりのみでは，DLFSR で使用している原始多項式を特定している最中にも Warbler ベースの NLFSR6 の内部状態に変化が加わるため，推測すべきビットが増加する．

以上から，本提案方式に対する最も有効な攻撃手段は，使用している使用している原始多項式と鍵の総当たりであると考える．

6.2 乱数特性

乱数特性の評価には，NIST SP800-22[10] を利用した．実験では，各パラメータの初期値を下記のように設定した．

$\zeta = 1, \zeta_i = 0, (i = 1, 2, \cdots, 27)$
$\lambda = 1, \lambda_i = 0, (i = 1, 2, \cdots, 26)$
$\mu = 1, \mu_i = 0, (i = 1, 2, \cdots, 24)$
$a_k = 1, a_{k+i}, (i = 1, 2, \cdots, 5)$
$lf = 1, lf_i = 0, (i = 1, 2, \cdots, 15)$
$l = 15$(乱数の取り込みのタイミングは 15 ラウンド毎)

DLFSR で使用した 8 個の原始多項式は，下記の 16 次のものである．記述通りの順番で Select に格納した．

$x^{16} + x^{11} + x^7 + x^6 + x^5 + x + 1,$
$x^{16} + x^{11} + x^7 + x^6 + x^5 + x^4 + 1,$
$x^{16} + x^{11} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1,$
$x^{16} + x^{11} + x^{10} + x^6 + x^5 + x^3 + 1,$
$x^{16} + x^{11} + x^6 + x^5 + 1,$
$x^{16} + x^{13} + x^{11} + x^{10} + x^6 + x^5 + 1,$
$x^{16} + x^{11} + x^{10} + x^6 + x^5 + x + 1,$
$x^{16} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^3 + x + 1$

また，乱数 r, rm の生成は python2.7 の「random.randint」を利用し，100,000bit，100 標本を用意し，検定を行った．結果を表 2 に示す．なお，検定の中に同一項目で複数あるものについては出力されたものの最初のものを記述した．

検定名	P-VALUE	PROPORTION
Frequency	0.987896	98/100
BlockFrequency	0.719747	99/100
CumulativeSums	0.574903	99/100
Runs	0.924076	99/100
LongestRun	0.275709	100/100
Rank	0.779188	100/100
FFT	0.115387	100/100
NonOverlappingTemplate	0.816537	98/100
OverlappingTemplate	0.798139	100/100
ApproximateEntropy	0.897763	99/100
RandomExcursions	0.739918	10/10
RandomExcursionsVariant	0.911413	10/10
Serial	0.181557	100/100
LinearComplexity	0.739918	99/100

表 2 NIST SP800-22 の検定結果

乱数検定の結果，表 2 に記述したもの以外を含めて 148 個ある NonOverlappingTemplate 検定のうち 100 標本の乱数系列のうち 96 しか通らなかったものが 3 個，97 しか通らなかったものが 15 個以外は，すべて 98 標本を以上通過している．これから乱数特性については特に問題ないと考える．

7. まとめ

本研究では，スマートデバイスのセキュリティのコアプリミティブとなりうるように既存の EPC Gen2 向けの擬似乱数生成器をベースに鍵長が 96 ビット，DLFSR で使用する原始多項式が第 3 者に知られることなくすでに焼きこまれているものとしてモデルを提案した．

本提案方式は，NLFSR と DLFSR を組み合わせ外部乱数を利用するものであり，その安全性を評価した．また，NIST の SP-800-22 を利用し，統計的にも優れた擬似乱数生成器であることを示した．今後は，さらなる安全性の評価と，実際に回路に焼きこむことにより，回路規模，消費電力等を評価していく．

8. 謝辞

本研究の一部は科学研究費基盤 C (15K00183) と (15K00189) 及び科学技術振興機構（JST）の国際科学技術協力基盤整備事業の助成を受けています．

参考文献

[1] GS1 EPCglobal, EPC TM Radio-Frequency Identity Protocols Generation-2 UHF RFID,

http://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_2_0_0_standard_20131101.pdf.

[2] EPCglobal. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960 MHz.

[3] EPC Global. UHF Air Interface Protocol Standard Generation2/Version2, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2.

[4] Gangqiang Yang, Mark D. Aagaard, Guang Gong *Efficient Hardware Implementations of the Warbler Pseudorandom Number Generator*, NIST Lightweight Cryptography Workshop 2015.

[5] KALIKINKAR MANDAL, XINXIN FAN, GUANG GONG, *Design and Implementation of Warbler Family of Lightweight Pseudorandom Number Generators for Smart Devices*, ACM Transactions on Embedded Computing Systems, Vol. 15, No. 1, Article 1, Publication date: February 2016.

[6] Joan Meli'a-Segu, Joaquin Garcia-Alfaro, Jordi Herrera-Joancomart ´ , *J3Gen: A PRNG for Low-Cost Passive RFID*, Sensors 2013.

[7] Peinado, A.; Munilla, J.; F?ster-Sabater, A. EPCGen2 Pseudorandom Number Generators: AnalysisofJ3Gen. Sensors 2014, 14, 6500?6515.

[8] Experimental Analysis of Guess-and-Determine Attacks on Clock-Controlled Stream Ciphers(Cryptography and Information Security, ¡Special Section¿Information Theory and Its Applications),ShinsakuKIYOMOTO, Toshiaki TANAKA, and Kouichi SAKURAI, Members, IEICETRANS.FUNDAMENTALS,VOLE88-A,NO.100CTOBER2005.

[9] Guess and Determine Attack on SNOW,Philip Hawkes, Gregory G. Rose,Volume 2595 of the series Lecture Notes in Computer Science pp 37-46

[10] NIST:A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, Special Publication 800-22 Revision 1a(2014.04) .