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

Hiroshi Nomaguchi Atsuko Miyaji and Chunhua Su *Fellow*

***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 EPC Global 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. we also shows that it is resistant to identification attacks and GD attacks.**

***Index Terms*—Non-linear feedback shift register, Pseudo-random number generator, RFID, EPC Gen2**

I. INTRODUCTION

A smart device having a communication function is one of the main components of IoT. In particular RFID[1] are considered various usages and applications, it is expected that one of the smart device that is responding to various needs. Cryptographic primitives can provide secure communications between the RFID reader and tag by using elaborately generated cryptographic keys. These unpredictable and irreproducible secret keys determine the communication security, which are created by a pseudo-random number generator (PRNG). Under such background, the importance of RFID orientated PRNG is on the rise. Because the restricted resource and power, light-weight PRNG is one of the core primitives of RFID security. The regular PRNG is difficult directly to be applied to RFID tags. In this paper, we focus on the extremely light-weight pseudo-random number generator for EPC Gen2 RFID tags.

We study the problem of constructing PRNG on EPC Gen2 tags which are required to function with less resources and power comsuption. Therefore, possible saving resources mounted on smart device operates in the power saving, it is necessary to pseudo-random number generator with sufficient security can be used as a primitive. Warbler（32,2,5,6)[2]and (62,3,5,6)[3] are a short key length, J3Gen[4] is reported attack[5]. We will be the next target. The key length is 80 or more[6][7]. Circuit scale is less than or equal to 2000GE[8][9]. Two of the above, is the most severe conditions in which we were in the eye. In the present study,it have a key length of the need to use as security of the core primitive, a sufficient security, and propose a pseudo-random number generator that also has excellent statistical evaluation. It is pseudo-random number generator for the existing smart devices based on the non-linear part feedback shift register (hereafter, NLFSR) and the dynamic feedback shift register (hereafter, DLFSR) . Based on the evaluation results, our proposal method has excellent statistical random number characteristics and it is found to be resistant to existing attacks.

H.Nomaguchi and A.Miyaji were with Japan Advanced Institute of Science and Technology, Asahidai 1-1, Nomi-shi, Ishikawa, 923-1292 Japan. email(s1510041@jaist.ac.jp,miyaji@comm.eng.osaka-u.ac.jp)

A.Miyaji is with CREST, Japan Science and Technology Agency, Kawaguchi Center Building 4-1-8, Honcho, Kawaguchi-shi, Saitama, 332-0012 Japan.

A.Miyaji C.Su are with Osaka University, Yamadaoka1-1 Suita-shi, Osaka 565-0871 Japan. email(su@comm.eng.osaka-u.ac.jp)

A. Related works

As a pseudo-random number generator for the EPC Gen2, J3Gen, Warbler (32,2,5,6), (62,3,5,6), LAMED [10], AKARI[11], Grain[12] has been proposed. These are the performance of the pseudo-random number indicated by the EPC Gen2 meets. However, looking security and privacy that is beginning to be newly requested to RFID, there are many that are not security evaluation. Further, there is even the security evaluation by the author, also those attacks as J3Gen and Grain has been reported. There is a Warbler as one of those who have not been reported for about attack and have the security assessment to the author, but the key length is less than 80bit. In addition, Warbler may not provide enough security under powerful adversarial setting. For example,the internal state of the NLFSR6 can be recovered form the output and there is a bias in the input and output of the PRNG.

B. Our contributions

In this paper, we provide security analysis on Wabler pseudo-random number generator for EPC Gen2 RFID tag. Based on our analysis, we propose a new light-weight PRNG which have a large key length and are resistant to existing attacks against Warbler and J3Gen. Our contributions are summarized as follows:

- Based on Wabler construction, we improve the security by extending the key length (at least 80bit), which is more secure with larger key space.
- We consider the implementation on the real-world EPC Gen2 RFID tags. The scale of the circuit is less than or equal to 2000GE which is outperform the existing PRNG scheme for EPC Gen2 tags under the same security level.

- Based on our experimental analysis using the NIST pseudo-randomness test package, we show that our proposed PRNG pass all 16 tests and does not have bias.
- We also present the security analysis on our PRNG, we show that our PRNG has resistance to existing attacks against Wabler and J3Gen.

II. Security Analysis for Wabler and J3Gen PRNG

EPC Gen2 tag is required to be operated with restricted computing resources such as memory and power consumption. In the existing research, there are two major PRNG constructions for EPC Gen2 RFID tag. The Warbler is based on multiplied and NLFSR using Trace. The J3Gen is based on DLSFR to switch between multiple polynomial by a random number r. In describing these, defined as follows.

$\boxplus$	:Addition
$+$	: Exclusive OR
f	:Primitive polynomial
f_j	: The j-th primitive polynomial
Select	:Location of the primitive polynomial to be used in a feedback
Select(j)	: Choose from the j-th primitive polynomial Select
ζ, λ, μ	: NLFSR of register. It shows the index i the register number
a	: NLFSR6 register. Each register is 5 bits
lf	: Register number of DFSR
l	:The period of the switching of Primitive polynomial of DFSR and random number r, rm
t	: 5 bits memory. Subscript denotes the i-th bit

A. Warbler

This section describes the Warbler (62,3,5,6). Algorithm is shown in Figure1. Warbler is configured to the three NLFSR

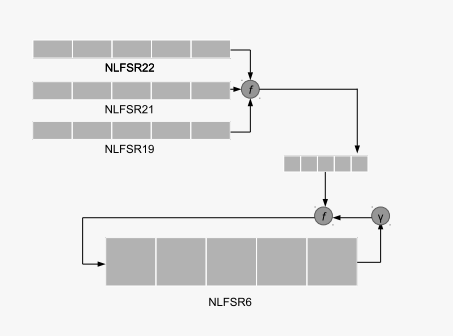

Fig. 1. Warbler(62,3,5,6)

of 1 bit 1 register, one NLFSR of five bits of 1 register and use the extension field. The internal state of the Warbler (62,3,5,6) is a 92-bit, the key length is 60 bits. The feature of Warbler is NLFSR which guarantees the maximum possible cycle and NLFSR 6 which multiplies the extension field. Since the explanation of the details is not our work, we omit it.

B. J3Gen

J3Gen is pseudo-random number generator based DLFSR to change the dynamic feedback using an external random number. The algorithm is shown in Fig2. J3Gen is constituted by the LFSR and Polynomial Selector to determine feedback. we explain the case of LFSR stages is and 8

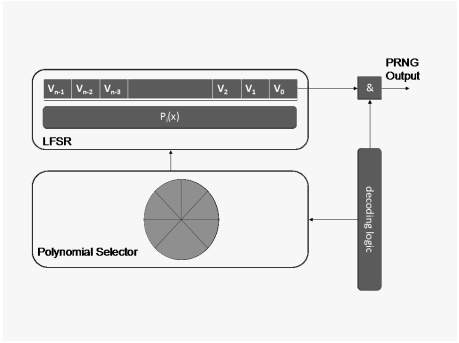

Fig. 2. J3Gen

primitive polynomials. Order is Polynomial Selector, LFSR. Polynomian Selector has a function to switch the primitive polynomial LFSR is used. This captures the external random number$r(r \in \{0, 1\})$ for each (l) round, choosing a primitive polynomial fb_i that LFSR to use. It shows the switch to the following formula.

$fb_i = Select(j)$

$j = j + r$ (mod 8)

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

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

State transition of the LFSR(lf_i) are as the following equation.

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

C. Security analysis of Warbler

Gangqiang Yang says Warbler has claimed to have a tolerance to the next attack. Algebraic attack, Resistance Against Algebraic Attacks and its development attack, Weak Internal States and Fault Injection Attacks. Currently attack has not been reported, but the following features are disturbing. The output of the Warbler is as follows.

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

If you know the outputO_{k+1}(0 or 1) at the time $k+1$ round , you can identify the internal state of the NLFSR6 by order 2^4. This is because the outputs in the form of compression of the internal state of NLFSR6. In addition, if the attacker has to guess the NLFSR6 and memory $t_{K+1,K+2,K+3,K+4}$, 1 bit is determined to be applied to new in memory t_{K+5}From the output of the next time $K + 1$,. For this reason, the value associated with the state transition of NLFSR6 is better higher computational security. Next, attention is paid to the NLFSR19,21,22. The output of the NLFSR19,21,22 obtain an output by enter the following formula to f.

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

A truth table of the input-output relationship shown in Table I. From the TableI, if f outputs 0, $x_2 = 1$ a is the probability of 3/4, is the probability $x_2 = 1is$ 1/4. If f is outputs 1 is 3/4 $x_2 = 0$ of probability, is a 1/4 probability that $x_2 = 1$. There is a bias in this way. f is considered to be the effect of hiding a portion of the input at that time by ignoring any of the x_0, x_1,x_2. However, such bias is likely to be used in the attack. Therefore, it considered that it is desirable to replace the another function to keep the

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

TABLE I
A TRUTH TABLE OF f

deviation without computational security. Next, consider the circuit scale. Compared to the Warbler (32,2,5,6) is a circuit scale 937GE, the circuit scale 1238GE of Warbler (62, 3, 5, 6) that internal state has increased 27. When the future to think, than increasing the simple internal state for key length increase, there is a possibility that more than 2000GE is a restriction of the EPC Gen2. Considering that in the future be required to recommend equivalent to the key length of the common key encryption, it is to increase the simple internal state for key length increase, there is a possibility that more than 2000GE is a restriction of the EPC Gen2.

III. Security analysis of existing research

J3Gen has been reported attack[5] to restore the using primitive polynomial from the output. This attack was restored eight-seven primitive polynomial from the output of the 176bit. Taking in J3Gen, primitive polynomial to be used is a secret key. This attack show that it is possible to recover the part of the key from the output. However, attack on J3Gen requires multiple outputs. Conversely, if a large number of output is not, can not restore the primitive polynomial Next, consider the circuit scale. J3Gen circuit scale (16-stage shift register, the 8 primitive polynomial) can be implemented resources 439.1GE be added a part for an external random number TRNG and Decoding Logic module .

IV. The proposed scheme

In this study, it can be implemented in the environment of smart devices, such as saving resources, to constitute a pseudo-random number generator with a computational security, take a structure such as Figfig:teian. we designe combine Warbler

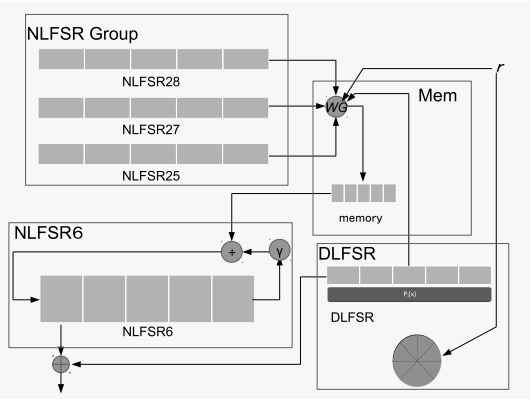

Fig. 3. The proposed scheme

and J3Gen. This is because that it is possible to cope with a large key length with increasing resource it is less than simply increasing the internal state of the Warbler. J3Gen if you could obtain an output of Warbler side, would be restored to primitive polynomial being used. For this reason, we have designed to bring the minimum guaranteed period of Warbler side in the key length. NLFSR group is composed of three NLFSR obtained by enlarging the internal state than that of Warbler, period around $2^{80}is$. In addition, the output of which is non-linear reduction by the WG. Shown in each of the parts. WG is to use the WGT of Warbler. WG is composed of WGP

for multiplying should be definitive in the Galois field and Trace. As of the following formulas, respectively law of Galois field

NLFSR28:$f_1 = x^5 + x^3 + x + 1$,
NLFSR27:$f_2 = x^5 + x^4 + x^2 + x + 1$,
NLFSR25:$f_3 = x^5 + x^4 + x^3 + x^2 + 1$
Next, a description will be given of WGP to be used in the WG. Input is referred to as $x(x \in F_2^5)$. WGP is expressed by the following equation.
$\mathrm{WGP}(x) = x + (x+1)^5 + (x+1)^{13} + (x+1)^{19} + (x+1)^{21}$.
Next, a description will be given of Trace. Trace is shown by the following equation.
$\mathrm{Trace}(x) = x + x^2 + x^{2^2} + x^{2^3} + x^{2^4}$ $(F_2^5 \to F_2)$
By WGP and Trace, WG is shown as the following equation.
WG(x)=Trace(WGP(x^d)) $(x \in F_2^5)$
Next, a description will be given NLFSR to use the WG. NLFSR25(μ), 27(λ), 28(ζ) consists of 25, 27, 28 stages each one bit of the register. Period is also $2^n - 1$ and LFSR of n stage, the entire period is about 2^{80}. This is because, since the period of each NLFSR are are designed to be relatively prime Each state transition is represented by the following formula.
$\zeta_{k+28} = 1 + \zeta_k + \mathrm{WG}(x^7)$,
$x = (\zeta_{k+3}, \zeta_{k+4}, \zeta_{k+8}, \zeta_{k+12}, \zeta_{k+20})$,
$\lambda_{k+27} = 1 + \lambda_k + \mathrm{WG}(y^{11})$,
$y = (\lambda_{k+4}, \lambda_{k+10}, \lambda_{k+12}, \lambda_{k+15}, \lambda_{k+20})$,
$\mu_{k+25} = 1 + \mu_k + \mathrm{WG}(z^7)$,
$z = (\mu_{k+3}, \mu_{k+6}, \mu_{k+14}, \mu_{k+16}, \mu_{k+18})$. Memory($t$) is a save to keep the 5-bit memory output, such as NLFSR group. In Warbler had saved it through the output of NLFSR19,21,22 f, in this proposal utilizes WG operates as follows. (Notation as follows .d_i indicates the 16 i-th register of DLFSR .rm_i external random number at the time i).
$s_i = WG(\zeta_i, \lambda_i + 1, \mu_i + 1, rm_i, d_i), s_j = 0, j = 0, 1, 2, 3$,
$t_{i+4} = (s_i, s_{i+1}, s_{i+2}, s_{i+3}, s_{i+4})$ In this proposal, NLFSR operates as follows. NLFSR6 is 5-bit 6-stage NLFSR (a). NLFSR6 do multiplication in the enlarged body. Operations are shown below.
$a_{k+6} = \gamma a_k + a_{k+1} + w_k + t_k$, $w_k = (0, 0, 0, 0, WG(a_k^{11}))$
DLFSR is based J3Gen. Similarly J3Gen, constituted by Polynomial Selector and LFSR. Polynomial Selector has a function to switch the primitive polynomial LFSR is used .It captures the external random number$r(r \in \{0, 1\}$ for each round(l), choosing a primitive polynomialfb_i. It shows the switch to the following formula.
$fb_i = Select(j)$
$j = j \boxplus r$ (mod 8)
$Select = f_1, f_2, \cdots, f_8$

$f_j : j = 1, 2, \cdots, 8$
LFSR (l_i) is to use a 1-bit 16-stage. The state transition are as the following equation.
$lf_{i+1} = fb_i(lf_i)$
By NLFSR6 and DLFSR, output (O_k) is represented by the following formula.
$O_k = WG(a^3_{k+5}) + l_{k+15}$.

V. Processing step

In this chapter, described divided into initialization step, a pseudo-random number generation step. In the proposed scheme, to initialize using the IV(35bit) and the key(96bit). It should be noted that the primitive polynomial used in DLFSR is from the 2048 16-order primitive polynomial is assumed that the elaborate baked already chosen eight in secret. It represents a key $Key = K_0, K_1, K_2 \cdots, K_{95}$, in the initial vector $IV = IV_0, IV_1, IV_2 \cdots IV_{34}$. The arrangement of the key and IV is Algorithm 1, initialization is Algorithm2. After the initialization process is completed,outputs a pseudo-random number. As the operation of the following Algorithm3. For initialization step, to correlate the output of NLFSR6 on feedback NLFSR Group and DLFSR. As a result, in addition to the nonlinearity possessed by the NLFSR group and DLFSR, give the non-linearity of the multiplication in the enlarged body with the NLFSR6 to each. The number of initialization rounds were set as key is agitated. It indicates the number of rounds necessary early in the following. The required number of rounds was considered as the number of rounds required to change from the value of the initial placement. For example, a XOR with other registers is true in the shift register .

- NLFSR28, 27, 25 : 28,27,25 round
- mem : 5 round
- NLFSR6 : 6 round
- DLFSR : 16 round

From the above, it is most necessary initialization(28 round) of the group NLFSR. Using the output of Stirred NLFSR group and DLFSR, for further stirred by the multiplication in the enlarged body of NLFSR6. This is 28 rounds to spread to all the registers. Because of this initialization round was set with 56 round.

Algorithm 1 The Arrangement of the key and IV

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 Initialization

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

VI. Evaluation

For security evaluation, we evaluate in order of random number characteristics, period, linear complexity, distinguishing attack, GD attack.

A. Random number characteristic

The evaluation of the random number characteristics, using the NIST SP800-22. We set the initial value of each parameter as described below in the experiment.
$\zeta = 1, \zeta_i = 0, (i = 1, 2, \cdots, 27)$
$\lambda = 1, \lambda_i = 0, (i = 1, 2, \cdots, 26)$
$\mu = 1, \mu_i = 0, (i = 1, 2, \cdots, 24)$
$a_k = 1, a_{k+i}, (i = 1, 2, \cdots, 5)$
$lf = 1, lf_i = 0, (i = 1, 2, \cdots, 15)$
$l = 15$
8 primitive polynomial used in DLFSR is of 16 following below. It was stored in the Select in the order of description Street.
$x^{16} + x^{11} + x^7 + x^6 + x^5 + x + 1$,
$x^{16} + x^{11} + x^7 + x^6 + x^5 + x^4 + 1$,
$x^{16} + x^{11} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$,
$x^{16} + x^{11} + x^{10} + x^6 + x^5 + x^3 + 1$,
$x^{16} + x^{11} + x^6 + x^5 + 1$,
$x^{16} + x^{13} + x^{11} + x^{10} + x^6 + x^5 + 1$,
$x^{16} + x^{11} + x^{10} + x^6 + x^5 + x + 1$,
$x^{16} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^3 + x + 1$
In addition, the generation of random numbersr, rm using the "random.randint" of python2.7. It was prepared 100,000bit × 100 samples to test. The results are shown in the tableII. It should be noted that, for some multiple of the same item in the test described the first one of those is output. A result of the random number test, 18 things that did not pass only 96, 97 out of the random number sequence of 100 samples out of the

Algorithm 3 Pseudo-random number generation step

$j = 0$
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**
$j = j \boxplus r$ (mod 8)
$fb_k = Select(j)$
end if
$lf_{i+1} = fb_i(lf_i)$
$O_i = WG(a_{k+5}^3) + lf_{i+15}$
end for

Test name	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

TABLE II
RESULT OF NIST SP800-22 TEST

148 pieces of the test. All of these except for passing throug more than 98 specimens. We think that there is no particul problem for the future random number characteristics.

B. period

The NLFSR group inside the proposed algorithm has period of 2^{80}. On the other hand, since the external rando number is incorporated in the determination of the feedba equation of the DLFSR, it does not have the DLFSR cycl For this reason, since there is no cycle of NLFSR 6 using tl output of the DLFSR and the NLFSR group to perform the state transition of the proposed scheme, it is also impossible to attack using the cycle.

C. linear complexity

Evaluation of linear complexity is often done using the Berlekamp-Massey algorithm. This is effective for a pseudo random number which is a finite period. In the proposed algorithm, there are no cycles for taking external random numbers. Moreover, since 99 of 100 samples are passed also in the linear complexity test by NIST 's random number test tool SP 800-22, we consider that the partial linear complexity of the pseudo - random number sequence is satisfactory. Therefore, it can not be a valid attack means.

D. Resistance of DLFSR

In this proposed method, it makes it difficult to get the output of DLFSR directly, thereby making it less susceptible to attack on J3Gen, so it shows it.
In order to make the evaluation easier, let the external random number r involved in switching the feedback expression be 0. In this case, when eight primitive polynomials are used for $l = 15$, consecutive 140 (= 15 (switching timing) × 8 (number of primitive polynomial)) bits are at least necessary. Actually, when switching with l = 15, the first primitive polynomial is used only up to the 30th bit, so we need to guess the remaining 2 bits but omit it.The output of this proposed scheme is configured by exclusive OR of the outputs of NLFSR 6 and DLFSR. For this reason, in order to obtain the output of the DLFSR from the output of the proposed method, it is necessary to estimate the output of the NLFSR 6 and so on. Therefore, to restore the primitive polynomial, it is required to estimate the output of NLFSR 6 of at least 140 bits. This exceeds the key length of this proposed method. Therefore, it is resistant to this attack.

E. distinguishing attack

In order to evaluate the security against distinguishing attacks, in this paper we divided the linear part and the nonlinear part like Fig.4 and examined the attack.

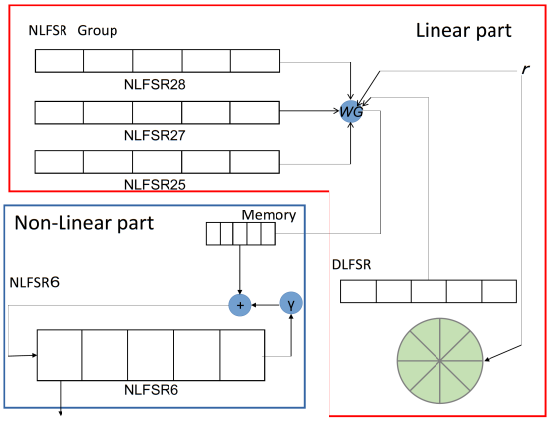

Fig. 4. Simplified model for distinguishing attack

As shown in Fig. 4, NLFSR group and DLFSR are set to linear part, memory and NLFSR 6 are made nonlinear part. The reason is that when applying the identification attack to SNOW, we looked at the fact that the LFSR part was considered as a linear part and the FSM as a nonlinear part. Since the NLFSR group uses it as a random number source, it is regarded as a linear part. Although the DLFSR is influenced by external random numbers to determine the feedback of the DLFSR, the state transition is a linear part because the influence of the internal register is large. The memory operates similar to R1, R2 of SNOW 1.0 which saves the output from

the LFSR, so it was set as a nonlinear part. When NLFSR 6 is regarded as a memory with shift function, multiplication of extension field is added when performing state transition. This is the same as the S function of SNOW, as the output is one to one correspondence to the input. For this reason, NLFSR 6 can be regarded as an S function and a shifted memory, so it is a nonlinear part.Next, a linear approximation formula of the nonlinear part is obtained. As shown in Fig.5, the input to the linear part to the nonlinear part was 1 bit added to every round memory, and the output was 1 bit after passing through WG of NLFSR 6. Do not use the exclusive OR of the output 1 bit (compressed 5 bit) of register 0 of NLFSR 6, which is the original output, and DLFSR in order to facilitate the evaluation. Since input and output are both 1 bit, $\Gamma = 1$, we did a full search. As a result, we measured 2, 3, 4 null, but neither bias was detected. If there is no valid linear approximation expression, this identification attack can not be performed, so it can be seen that it can not be applied to this proposed scheme.

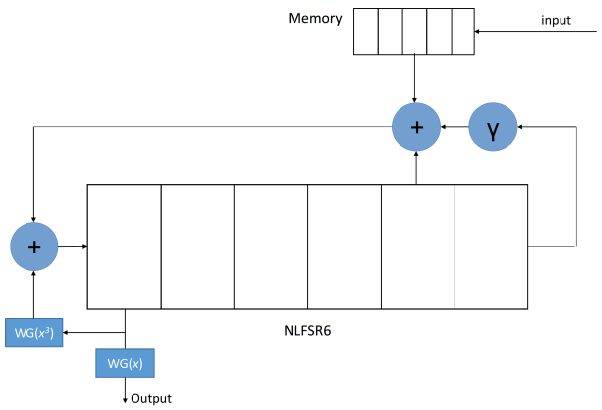

Fig. 5. Search for linear approximation

F. GD attack

In order to facilitate security evaluation for this proposed method, the output of the main body shall be NLFSR 6 only. In other words, NLFSR 6 only (output on the DLFSR side is 0) as in Fig.6, which is originally the exclusive OR of NLFSR 6 and DLFSR.

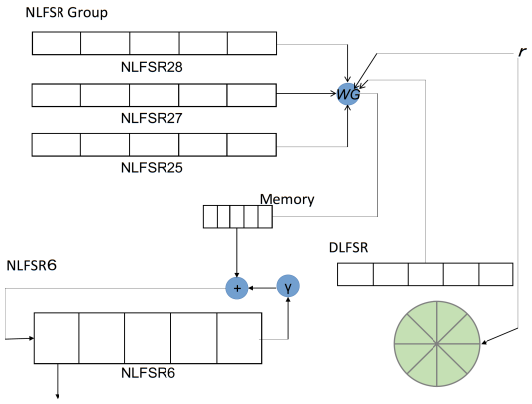

Fig. 6. Simplified model for GD attack

1) Simple GD attack: We describe the security of NLFSR6. The internal state of the 5-bit register of NLFSR 6 is compressed to 1 bit by WG and outputted. If 1 bit output of the model is obtained, if 4 bits out of 5 bits related to output are guessed, the remaining 1 bit can be determined from estimation and output. If memory is always 0, we can infer the register at time $t+6$ by estimating the $t, t+1$ th register related to the first state transition by 5 bits at a time. Then, if the $t+2, t+3, t+4$ registers of the remaining NLFSR 6 are estimated 4 bits at a time, the remaining 1 bit of these registers can be determined from the output. The amount of computation required at this time is $2^{5\times 2} \times 24 \times 4 = 226$. This is less than 2^{30}, which is the total number of registers in NLFSR 6 which is an exhaustive search, but this is the case when memory is always 0, and in fact memory must also be considered. Next, we evaluate the security of the input bit to be added to memory. It is assumed that all the internal states of the NLFSR 6 are obtained, and it is assumed that 1 bit added to the memory can be obtained from the output of the proposed scheme. That is, it is assumed that 1 bit to be added to the memory can be directly obtained. 1 bit added to the memory is compressed by WG, 3 bit output of NLFSR group, external random number rm and 1 bit of DLFSR. If you guess 4 bits out of 5 bits here, you can also determine the remaining 1 bit from the output. At this time, various guess combinations are conceivable. First. Including it on the side that guesses DLFSR like Fig.7, it shows the computational complexity. In estimating the DLFSR, if the feedback equation is not known even if the internal state is inferred, the output after the inferred internal state is unknown. For this reason, it is necessary to calculate (2^{88}) which is the sum of the internal state 16 bits and the feedback formula (2^{72}) only by estimating DLFSR. Consider the case where NLFSR 28 is determined. This is because much less can be determined by output with few guesses as compared with estimating other NLFSR and external random numbers. When determining the NLFSR 28, it is necessary to estimate the internal state of the NLFSR 27,

25 by outputting 28 bits and the random number rm 2 bits (when l = 15). The amount of calculation required to determine this NLFSR 28 is 88 + 27 + 25 + 2 = 142, exceeding the key length.

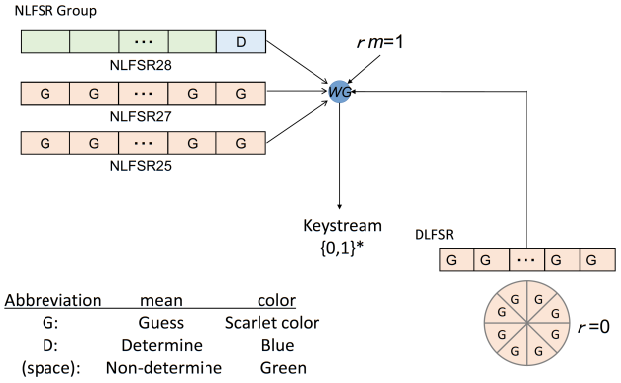

Fig. 7. GD attack application example1

Next, consider the case of estimating the NLFSR group and the external random number as shown in Fig.8, and determining the internal state and the feedback expression of the DLFSR. The estimate of the NLFSR group is $28+27+25 = 80$ bits. About the bit to be estimated of the external random number rm. Even if the external random number r involved in switching DLFSR is always 0, in order to obtain the feedback expression from the output of DLFSR it is necessary to estimate at least rm 8 bits. In reality, when obtaining the feedback formula, it is a state where it is not known whether the restore is performed except for the feedback expression that is actually used. In order to check whether the combination of the primitive polynomial obtained by restoration is correct, it is necessary to guess the external random number rm involved in the same degree of output, so 16 bits are actually needed. Therefore, even in a state where the internal state of the NLFSR 6 is known, a calculation amount of at least 96 bits or more is required. This is a calculation amount equivalent to the key length. In addition, in the actual method, it is necessary to estimate the internal state of the NLFSR 6. As a result, it is understood that the computation amount larger than the key length is necessary as a result. From the above, it was shown that GD attack can not be applied to this proposed method.

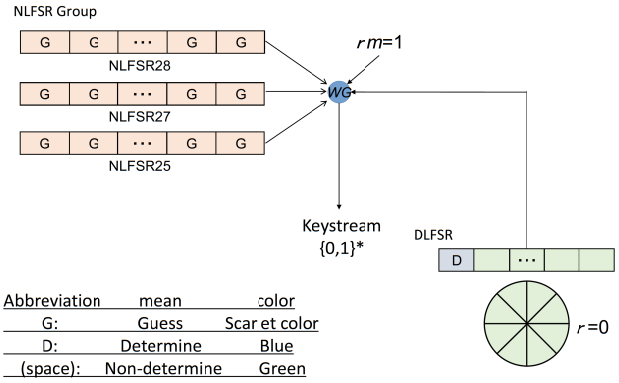

Fig. 8. GD attack application example2

2) GD attack using assumption: When introducing assumptions, it is necessary to set so that more decisions can be made. In this paper we introduce the assumption that the internal states of two different registers introduced in SNOW 1.0 are together. It is assumed that all the internal states of the NLFSR 6 are found and it is possible to obtain 1 bit to be added to the memory. First, estimate DLFSR, NLFSR 25, 27 and evaluate the effectiveness of the assumption when NLFSR 28 is determined. If both of the outputs of the NLFSR group (NLFSR 25, 27) are 0, the probability is 1/4 = 2^{-2}. The probability of occurrence when this happens continuously for 25 bits is 2^{-50}. In addition, we need to guess the remaining two registers of NLFSR 27. The same is true even if the two outputs are (0, 1), (1, 0), (1, 1). The reason for this is that the WG function has no bias for a specific input bit , Because valid assumptions can not be made. From the above, when combined with the calculation amount 2^{88} required for DLFSR guessing, it is understood that the calculation amount when this attack is applied to the proposed scheme exceeds the key length.
Next, assume the case where NLFSR group, inferring to random numbers, introducing assumptions, and determining DLFSR. At this time as well, there is no bias in WG and no valid assumption can be found. The probability of establishing that the random numbers r and NLFSR 25 are 0 between 16 bits is $2^{-2} \times 2^{-16} = 2^{-18}$, and the remaining 9 bits of NLFSR 25 and $2^{9+27+28+14} = 2^{78}$ is required for estimating the remaining random numbers until the primitive polynomials of the NLFSRs 27 and 28 and the DLFSR are specified. From these, it is found from the probability of assumption and the calculation amount of estimation that the amount of calculation required for attack is the same as the key length. In fact, when restoring the primitive polynomial of DLFSR, verification is necessary because there is a mixture of what you are using and what is wrongly restored. In addition, since it is essentially necessary to estimate the random number r which determines the feedback of the DLFSR, we consider GD attack under the key length to this proposed algorithm to be difficult.

VII. Summary

In this paper, we propose a model based on an existing pseudorandom number generator for resource - saving device and conducted its security evaluation. Security assessment was performed on linear complexity, cycle, DLFSR attack, distinguishing attack, GD attack, which are necessary items as pseudo random numbers used as security core primitives. We showed that linear complexity and period do not have finite period by capturing external random number and are not attacked using them. It shows the computational complexity necessary to apply the attack against J3Gen which is the base of DLFSR and showed improvement on security. Regarding the identification attack, we classified it into linear part and nonlinear part, and searched for bias on input and output of nonlinear part, but did not find it, it showed that we can not apply identification attack because we can not construct linear approximation formula . Regarding the GD attack, it showed its security in a simple model. In the simplified model, it is divided into ”NLFSR 6”, ”memory”, ”NLFSR group and DLFSR”, and the calculation amount of each is individually obtained. Then, we confirmed that the total calculation amount exceeds the key length. ”Computational complexity of proposed method” $>$ “Computational complexity of simplified model”, so we could show the security against GD attack. From these, it was shown that the scheme of the pseudorandom number generator proposed by us is safe. In the future, we will install it in hardware, measure the circuit scale and power consumption, and evaluate whether it can withstand practical use.

Acknowledgments

This work is partially supported by JSPS KAKENHI Grant (C)(JP15K00183) , (JP15K00189) , (JP15K16005) and Japan Science and Technology Agency, CREST(JPMJCR1404) and Infrastructure Development for Promoting International S&T Cooperation and Project for Establishing a Nationwide Practical Education Network for IT Human Resources Development, Education Network for Practical Information Technologies.

References

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

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

[3] 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.

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

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

[6] Paar, C.; Poschmann, A.; Robshaw, M. New Designs in Lightweight Symmetric Encryption. In RFID Security; Kitsos, P., Zhang, Y., Eds.; Springer: New York, NY, USA, 2009; pp. 349-371.

[7] Bogdanov, A.; Knudsen, L.; Leander, G.; Paar, C.; Poschmann, A.; Robshaw, M.; Seurin, Y.; Vikkelsoe, C. PRESENT: An ultra-lightweight block cipher. Lect. Note. Comput. Sci. 2007, 4727, 450-466.

[8] Juels, A. RFID security and privacy: A research survey. IEEE J. Sel. Areas Commun. 2006,24, 381?394.

[9] Weis, S.; Sarma, S.; Rivest, R.; Engels, D. Security and privacy aspects of low-cost radio frequency.

[10] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador, Arturo Ribagorda, LAMED-A PRNG for EPC Class-1 Generation-2 RFID specification.

[11] Honorio Martn, Enrique San Milln, Pedro Peris-Lopez, and Juan E. Tapiador, Efficient ASIC Implementation and Analysis of Two EPC-C1G2 RFID Authen tication Protocols, IEEE SENSORS JOURNAL, VOL. 13, NO. 10, OCTOBER 2013.

[12] Haina Zhang, and Xiaoyun Wang, Cryptanalysis of Stream Cipher Grain Family.

[13] KIYOMOTO Shinsaku,TANAKA Toshiaki,SAKURAI Kouichi Experimental Analysis of Guess-and-Determine Attacks on Clock-Controlled Stream Ciphers(Cryptography and Information Security, ¡Special Section¿Information Theory and Its Applications) IEICE transactions on fundamentals of electronics, communications and computer sciences 09168508 IEICE 2005-10-01 88 10 2778-2791 http://ci.nii.ac.jp/naid/110003213299/

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

[15] NIST:A Statistical Test Suite for Random and Pseudo- random Number Generators for Cryptographic Applica- tions, Special Publication 800-22 Revision 1a(2014.04) .