$CCC-NSG$: 순환 클럭 조절된 비선형 알고리즘을 이용한 블루투스 E_0 암호화시스템의 안전성 개선

정회원 김 형 락*, 이 훈 재**, 종신회원 문 상 재***

$CCC-NSG$: A Security Enhancement of the Bluetooth E_0 Cipher using a Circular-Clock-Controlled Nonlinear Algorithm

Hyeong-rag Kim*, Hoon-jae Lee** ***Regular Members***, Sang-jae Moon*** ***Lifelong Member***

요 약

합산수열 발생기는 간단한 하드웨어 또는 소프트웨어로 구현될 수 있고, 주기와 선형복잡도가 높은 특징이 있어 유비쿼터스 시대의 이동환경 보안장치에 적합하다. 하지만 Dawson의 각개공격과 Golic의 상관성공격 및 Meier의 고속 상관성공격에 의해 취약성이 노출되었다. 본 논문에서는 $CCC-NSG$를 제안한다. $CCC-NSG$에서는 합산수열 발생기 형태의 E_0 알고리즘을 개선하여 선형 LFSR 중 일부를 비선형 NFSR로 교체하였고, 클럭을 랜덤화해서 순환 클럭 조절함으로서 출력되는 키 수열의 안전성(2^{128} 보안 레벨)을 높였다. 또한, 제안 알고리즘에 대한 안전성 분석 및 성능을 분석하였다.

Key Words : *NSG, CCC-NSG,* Summation Generator, Stream Cipher, E_0

ABSTRACT

Summation generator with high period and high linear complexity can be easily implemented by a simple hardware or software and it is proper to apply in mobile security system for ubiquitous environments. However the generator has been some weaknesses from Dawson's divided-and-conquer attack, Golic's correlation attack and Meier's fast correlation attack. In this paper, we propose an improved version(2^{128} security level) of E_0 algorithm, $CCC-NSG$(Circular-Clock-Controlled − Nonlinear Summation Generator), which partially replaces LFSRs with nonlinear FSRs and controls the irregular clock to reinforce it's own weaknesses. Finally, we analyze our proposed design in terms of security and performance.

Ⅰ. 서 론

블루투스 기술은 1998년 스웨덴의 에릭슨이 주축이 되어 본격화된 기술로 사용자 정보 암호화를 위해서 E_0 알고리즘을 사용한다[1]. 이때 키수열 발생을 위해 4개의 선형귀환 이동레지스터(Linear Feedback Shift Register, LFSR)를 갖는 합산수열발생기를 사용하고 있다.

합산 수열 발생기는 스트림 암호를 위한 키 수열 발생기로 1985년 Rueppel [2]에 의해 최초 제안되었다. 합산 수열 발생기는 일정한 클럭을 갖는 r개의 이진 LFSRs(입력) 및 $\lceil \log_2^r \rceil$ ($=ceiling(\log_2^r)$; $\lceil x \rceil = \infty\{n \in Z | x \le n\}$, Z:정수군)비트의 메모리(입력)를 이용하며, 출력은 입력의 정수 합으로부

* 포항대학 컴퓨터응용과(hrkim@pohang.ac.kr), ** 동서대학교 컴퓨터정보공학부(hjlee@dongseo.ac.kr)
*** 경북대학교 전자전기컴퓨터학부(sjmoon@ee.knu.ac.kr)
논문번호 : KICS2009-05-199, 접수일자 : 2009년 5월 13일, 최종논문접수일자 : 2009년 7월 13일

터 얻는다. 합의 LSB(Least Significant Bit)비트는 키 수열을 생성하고, 나머지 비트들은 캐리(carry)비트들이며 메모리에 저장된다. 캐리 수열은 다음 비트 생성을 위해 결합함수(combining function)의 입력으로 사용되어진다.

LFSR은 하드웨어와 소프트웨어 구현에 적합하며, 빠른 암호속도 및 복호속도가 지원되어 스트림 암호에 많이 사용된다. 또한 원시다항식을 갖는 LFSR에 의해 발생된 수열은 큰 주기 및 좋은 통계적 특성을 갖는다. 그러나 LFSR은 그들의 선형성 때문에 출력 수열로부터 쉽게 예측(암호해독)이 가능하며, 길이 L인 LFSR에 대하여 키 수열의 전체 주기는 귀환 다항식이 알려져 있다면 수열의 연속 L항으로부터 구해지고, 알려져 있지 않다면 $2L$항으로부터 알 수 있다[3].

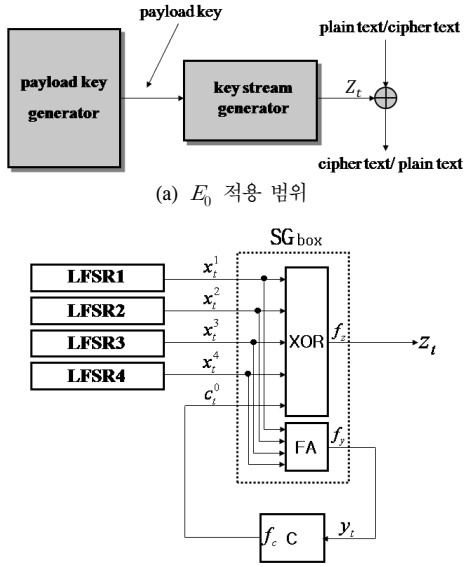
합산수열 발생기는 간단한 하드웨어 또는 소프트웨어로 구현될 수 있고, 주기와 선형복잡도가 높은 특징이 있어 유비쿼터스 시대의 이동환경 보안장치에 적합하다. 하지만 Dawson의 각개공격(divided-and-conquer attack)[3]과 Golic의 상관성공격[4] 및 Meier의 고속 상관성공격[5]에 의해 취약성이 노출되었다. 본 논문에서 제안된 $CCC-NSG$(Circular-Clock-Controlled−Nonlinear Summation Generator)에서는 합산 수열 발생기 형태의 E_0 알고리즘[1]을 개선하여 선형 LFSR 중 일부를 비선형 NFSR(Nonlinear Feedback Shift Register) 로 교체하였고, 클럭을 랜덤화해서 순환 클럭 조절함으로서 출력되는 키 수열의 안전성을 높이고자 한다. 그리고 제안 알고리즘에 대한 안전성 분석 및 성능을 분석한다.

Ⅱ. $CCC-NSG$ 제안

2.1 E_0 암호 알고리즘

짧은 거리에서 개체 간 통신을 제공하는 블루투스 기술에서 사용자의 정보는 패킷 페이로드를 암호화함으로써 보호된다. 이때 암호화는 E_0 스트림 암호화기에 의해 수행된다. 그림 1(a)는 E_0의 적용 범위를 보여주고, 그림 1(b)는 4개의 LFSR에 기초를 둔 합산수열 발생기를 사용한 E_0 암호 알고리즘을 보여 준다[1].

그림 1(a)에서 스트림 암호 시스템 E_0는 세 가지 부분으로 구성된다. 첫 번째 부분은 초기화를 수행하고, 두 번째 부분은 키 스트림 비트를 생성하고, 그리고 세 번째 부분은 암호화와 복호화를 수행한다. 페이로드 키 발생기(payload key generator)는 적절한 순서로 입력 비트들을 조합한 후 키 수열 발생기에서 사용되는 4개의 LFSR에 이동한다. 암호 시스템에서 주요한 부분은 두 번째이다. 키 수열 발생기는 Rueppel[2]이 제안한 합산 수열 암호화 발생기(summation stream cipher generator)를 사용한다.



(a) E_0 적용 범위

(b) E_0 알고리즘

그림 1. E_0 스트림 암호화시스템

2.2 NSG[11]

기존의 E_0 알고리즘에서 사용하는 합산수열발생기는 4개의 LFSR에 기초를 둔 합산수열 발생기를 사용하지만, NSG에서는 그림 2에서 보는 것처럼 1개의 LFSR과 3개의 NFSR을 사용한 합산수열 발생기를 사용 한다[11]. 이때 LFSR은 출력 키 수열의

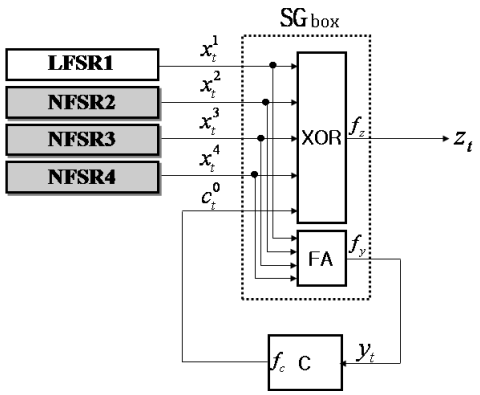


그림 2. NSG

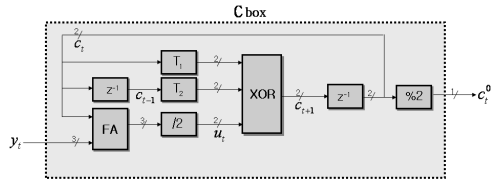


그림 3. c box

최소 주기를 보장해주고, 0-1 균형성을 제공해준다. 그리고 비선형 출력함수를 가지는 NFSR은 암호화 알고리즘에 비선형성을 높여준다[8].

그림 2는 *NSG*의 블럭도를 보여주고[11], 그림 3은 캐리 c_t^0를 출력하는 c box를 보여 준다[11].

2.3 *CCC−NSG* 제안

본 절에서는 기존의 블루투스 암호 알고리즘에서 사용하고 있는 합산수열 발생기의 비선형성을 증가시키기 위하여 de Bruijn 수열 발생기로 구현된 NFSR 및 순환 클럭 조절형 구조를 사용한 *CCC−NSG*를 제안한다.

2.3.1 de Bruijn 수열[6][7]

0과 1로 이루어진 주기 2^n인 수열 $\{a_i\}$에 대하여 연속으로 나타나는 2^n개의 벡터 $V_i = (a_i, a_{i+1}, a_{i+2}, \cdots, a_{i+n-1})$가 모두 다를 때 이 수열 $\{a_i\}$을 de Bruijn 수열이라고 한다. de Bruijn 수열은 예측이 불가능하면서 최대의 주기를 얻을 수 있고, 무작위성과 큰 선형 복잡도를 가진다. 또한 비선형성이 높으며 LFSR로부터 쉽게 생성할 수 있는 특성을 갖고 있다[6][7].

Theorem 1(Chang-Park, Chang-Song). 최대주기를 갖는 n단 LFSR의 귀환함수 f에 대하여 $h = f + x_1 x_2 \cdots x_{n-1} + 1$ 라 할 때 h를 귀환함수로 하는 이동레지스터에 의해 발생되는 수열을 de Bruijn 수열이라 하고, 이때 주기는 2^n이다[6][7].

2.3.2 *CCC−NSG*

2.3.2.1 *CCC−NSG* 키 수열 발생기

*CCC−NSG*는 순환 클럭 조절 구조가 추가된 합산 수열 발생기 이며, 그림 4와 같다. 그림에서 키 수열 발생기의 입력은 1개의 LFSR과 3개의 NFSR 그리고 1개의 캐리 비트로 구성된다.

그림 4에서 LFSR과 NFSR 각각은 불규칙한 클럭이 공급되며, 하나의 귀환 이동 레지스터에 공급되는 불규칙 클럭 수는 다른 귀환 이동 레지스터에

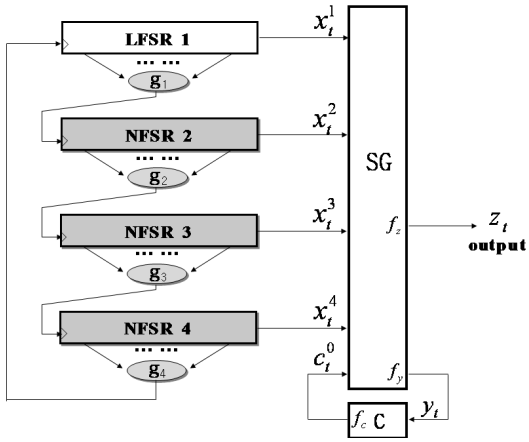


그림 4. 제안된 *CCC−NSG*

서 생성된 비선형 필터함수(g_1 , g_2 , g_3 또는 g_4)로부터 얻어진다. 이때 비선형 필터함수 g_1 , g_2 , g_3 및 g_4는 4개의 귀환 이동 레지스터의 현 상태로부터 불규칙한 클럭 값을 각각 생성한다. 생성된 각각의 불규칙 클럭 값은 다음 귀환 이동 레지스터에 클럭을 랜덤하게 조절하여 캐리 및 키 수열 출력을 생성한다.

*CCC−NSG*의 출력은 아래 식들과 같이 주어진다.

$$z_t = f_z(x_t^1, x_t^2, x_t^3, x_t^4, c_t^0) = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus c_t^0 \in \{0,1\} \quad (1)$$

$$y_t = f_y(x_t^1, x_t^2, x_t^3, x_t^4) = \sum_{i=1}^{4} x_t^i \in \{0,1,2,3,4\} \quad (2)$$

$$c_{t+1}^0 = f_c(x_t^1, x_t^2, x_t^3, x_t^4, c_t^0) = \{u_{t+1} \oplus T_1[c_t] \oplus T_2[c_{t-1}]\} \,\%\, 2 \in \{0,1\} \quad (3)$$

여기에서 u_{t+1}은 그림 3의 c box 내부 함수로서 $u_{t+1} = \lfloor \frac{y_t + c_t}{2} \rfloor$ 로 주어지고, $T_1[\cdot]$, $T_2[\cdot]$는 $GF(2^2)$상에서 두 개의 다른 선형 전단사(bijection)이다. $GF(2^2)$가 기약다항식 $x^2 + x + 1$에 의해 생성된다고 가정하고, α를 $GF(2^2)$에서 이 다항식의 근으로 두었을 때, 사상 T_1과 T_2는 아래 식 (4) 와 (5)로 정의된다.

$$T_1 : GF(2^2) \to GF(2^{2)} \quad (4)$$
$$\mathrm{x} \mapsto \mathrm{x}$$

표 1. T_1 과 T_2 의 변환

x	$T_1[x]$	$T_2[x]$
00	00	00
01	01	11
10	10	01
11	11	10

$$T_2 : GF(2^2) \rightarrow GF(2^2) \quad x \mapsto (\alpha+1)x \tag{5}$$

표 1에서 요약된 것처럼 $GF(2^2)$의 요소들은 이진 벡터로서 쓸 수 있다.

사상이 선형이기 때문에, *XOR*게이트를 사용해서 식 (6)과 (7)로 구현할 수 있다. 즉,

$$T_1 : \quad (x_1, x_0) \mapsto (x_1, x_0) \tag{6}$$

$$T_2 : \quad (x_1, x_0) \mapsto (x_0 , x_1 \oplus x_0) \tag{7}$$

그림 4에서 1개의 LFSR의 길이는 $L_1 = 127$이고, 3개의 NFSR 각각은 $N_2 = 23$, $N_3 = 35$, $N_4 = 71$이다. 모든 메모리 비트들은 *CCC−NSG*에게 256비트의 내부 상태 비트를 제공하며, 256비트 비밀키(key)와 256비트 초기화 벡터(iv)를 *XOR* 한 결과 값 256비트를 내부 상태에 채운다. *CCC−NSG*의 출력 키 수열은 귀환 이동 레지스터의 출력수열과 캐리 수열이 합쳐져서 생성된다.

$LFSR_1$, $NFSR_2$, $NFSR_3$ 및 $NFSR_4$의 귀환 다항식은 각각 다음과 같은 원시다항식 $p_1(x)$, $p_2(x)$, $p_3(x)$ 및 $p_4(x)$로부터 선택되며, $LFSR_1$의 모든 비트가 0 상태(all zero state)로 초기화 되는 것을 허용하지 않는다.

$$p_1(x) = x^{127} \oplus x^{87} \oplus x^8 \oplus x^7 \oplus x^6 \oplus x^5 \oplus x^4 \oplus x^3 \oplus x^2 \oplus x^1 \oplus x^0 \tag{8}$$

$$p_2(x) = x^{22}x^{21}x^{20}x^{19}x^{18}x^{17}x^{16}x^{15}x^{14}x^{13}x^{12}x^{11}x^{10}x^9x^8x^7x^6x^5x^4x^3x^2x^1 \oplus x^{23} \oplus x^{19} \oplus x^8 \oplus x^5 \oplus x^4 \oplus x^3 \oplus x^2 \oplus x^1 \oplus x^0 \oplus 1 \tag{9}$$

$$p_3(x) = x^{34}x^{33}x^{32}x^{31}x^{30}x^{29}x^{28}x^{27}x^{26}x^{25}x^{24}x^{23}x^{22}x^{21}x^{20}x^{19}x^{18}x^{17}x^{16}x^{15}x^{14}x^{13}x^{12}x^{11}x^{10}x^9x^8x^7x^6x^5x^4x^3x^2x^1 \oplus x^{35} \oplus x^{23} \oplus x^7 \oplus x^5 \oplus x^4 \oplus x^3 \oplus x^2 \oplus x^1 \oplus x^0 \oplus 1 \tag{10}$$

$$p_4(x) = x^{70}x^{69}x^{68}x^{67}x^{66}x^{65}x^{64}x^{63}x^{62}x^{61}x^{60}x^{59}x^{58}x^{57}x^{56}x^{55}x^{54}x^{53}x^{52}x^{51}x^{50}x^{49}x^{48}x^{47}x^{46}x^{45}x^{44}x^{43}x^{42}x^{41}x^{40}x^{39}x^{38}x^{37}x^{36}x^{35}x^{34}x^{33}x^{32}x^{31}x^{30}x^{29}x^{28}x^{27}x^{26}x^{25}x^{24}x^{23}x^{22}x^{21}x^{20}x^{19}x^{18}x^{17}x^{16}x^{15}x^{14}x^{13}x^{12}x^{11}x^{10}x^9x^8x^7x^6x^5x^4x^3x^2x^1 \oplus {}^{67} \oplus x^{43} \oplus x^{10} \oplus x^7 \oplus x^6 \oplus x^5 \oplus x^4 \oplus x^3 \oplus x^2 \oplus x \tag{11}$$

2.3.2.2 클럭 제어

*CCC−NSG*에서 네 개의 귀환 이동 레지스터는 각각 다음 귀환 이동 레지스터의 클럭을 랜덤하게 제어하여 각 레지스터에 불규칙한 클럭을 발생시킨다. $LFSR_1$ 탭의 값($g_1(x)$ 함수 값)으로부터 랜덤 값을 얻은 후 $NFSR_2$의 클럭을 제어하고, $NFSR_2$ 탭의 값($g_2(x)$ 함수 값)으로부터 $NFSR_3$을 제어하고, $NFSR_3$ 탭의 값($g_3(x)$ 함수 값)으로부터 $NFSR_4$를 제어 한다. 또한 $NFSR_4$ 탭의 값($g_4(x)$ 함수 값)으로부터 $LFSR_1$를 제어한다. 표 2에서는 클럭 제어 함수를 보여 준다.

표 2에서 예를 들어 $g_3(x) = 2x^{t_{3,1}} + x^{t_{3,2}} + 1$ 이 의미하는 것은 네 번째 이동레지스터 $NFSR_4$에 입력되는 클럭 조절함수를 의미하고, 그 계산은 세 번째 이동레지스터 $NFSR_3$의 길이가 N_3이라 할 때, $x^{t_{3,1}}$은 첫 번째 추출되는 값으로 $(1/3) \cdot N_3$를 의미하고, $x^{t_{3,2}}$는 두 번째 추출되는 값으로 $(2/3) \cdot N_3$을 의미한다. 모든 $x^{t_{i,j}}$ 값은 1 또는 0 이다.

본 설계에서 $LFSR_1$과 $NFSR_4$의 길이는 최소의 주기와 최소의 LC값을 보장하기 위해서 큰 단수를 요구한다. 클럭 조절이 없는 경우(*NSG* 발생기)와 비교하여 더 큰 주기와 더 큰 LC값을 얻기 위해서는 가운데 배치된 2개의 *NFSR*의 클럭을 조절할

표 2. *CCC−NSG* 클럭 제어 함수

클럭 조절 함수	클럭 조절 목표 레지스터	클럭 조절 범위
$g_1(x) = 1$: $LFSR_1$의 출력 클럭 함수	$NFSR_2$	1
$g_2(x) = x^{t_{2,1}} + 1$: $NFSR_2$의 출력 클럭 함수	$NFSR_3$	1~2 (랜덤)
$g_3(x) = 2x^{t_{3,1}} + x^{t_{3,2}} + 1$: $NFSR_3$의 출력 클럭 함수	$NFSR_4$	1~4 (랜덤)
$g_4(x) = x^{t_{4,1}} + 1$: $NFSR_4$의 출력 클럭 함수	$LFSR_1$	1~2 (랜덤)

[Note] $t_{2,1} = N_2/2$, $t_{3,1} = (1/3) \cdot N_3$, $t_{3,2} = (2/3) \cdot N_3$, $t_{4,1} = N_4/2$

필요가 있다. 특히 큰 단수를 갖는 첫 번째와 네 번째 레지스터에 대한 클럭 조절은 변동율을 크게 하고(안전성을 높이고), 작은 단수로 설계된 두 번째와 세 번째 레지스터에 대한 클럭 조절은 간단하게 설계함으로써 알고리즘의 성능과 효율을 높인다.

본 설계를 일반화시키면, 키 수열 발생기는 i번째 귀환 이동 레지스터 출력이 $i+1$번째 귀환 이동 레지스터의 클럭 조절에 사용된다. 그리고 마지막 레지스터 출력은 첫 번째 레지스터를 클럭 조절 할 수 있도록 순환시킨다

Ⅲ. 분 석

본 절에서는 실험적인 결과에 근거하여 $CCC-NSG$의 키 수열특성을 분석하고, 또한 알려진 공격에 대하여 안전함을 보여준다.

3.1 키 수열의 특성

PN 이진 수열들을 위한 세 가지 기본 요구사항은 긴 주기, 높은 선형복잡도, 좋은 통계 특성이며, 긴 주기는 암호화된 긴 메시지를 사용할 때 동일한 키 수열의 재사용을 방지하고, 높은 선형 복잡도는 Berlekamp-Massey 알고리즘[9]을 이용한 공격에 견딜 수 있도록 한다. 그리고 좋은 통계적인 특성은 키 수열이 "0"과 "1" 중 어느 한 방향으로 치우친 취약점을 이용한 공격에 견딜 수 있게 한다.

Theorem 2. de Bruijn 수열을 포함한 제안된 알고리즘(키수열 발생기)에서 $\{x_t^1\}$, $\{x_t^2\}$, $\{x_t^3\}$ 및 $\{x_t^4\}$를 원시다항식의 차수가 L_1, N_2, N_3 및 N_4 $(N_2 < N_3 < N_4)$로 주어진 4개의 이진 m-수열이라 한다. 이때 $CCC-NSG$ 대한 주기의 최소값 $T_{\min}$는 다음과 같이 정의된다.

$$T_{\min} = (2^{L_1}-1)\cdot 2^{N_4} \tag{12}$$

증명) 원시다항식으로 정의되는 선형 이동 레지스터 LFSR의 주기는 $(2^{L_1}-1)$이다. 그리고 de Bruijn 수열로 정의되는 비선형 이동 레지스터 $NFSR_2$, $NFSR_3$ 및 $NFSR_4$ 각각의 주기는 2^{N_2}, 2^{N_3} 및 2^{N_4}이다. 이때 비선형 이동 레지스터에 대한 주기는 $\gcd(2^{N_2},2^{N_3},2^{N_4}) = 2^{N_4}$ 이다. 따라서 전체 주기는 $\gcd((2^{L_1}-1),2^{N_4}) = 1$을 만족하므로 $T_{\min} = (2^{L_1}-1)\cdot 2^{N_4}$ 이다.

표 3. NSG와 $CCC-NSG$에서 주기 및 LC 시뮬레이션 결과(짧은 단수)

Taps of shift register	M	이론 추정치	시뮬레이션 값			
			NSG		$CCC-NSG$	
		$T_{\min}$	T_{NSG}	LC_{NSG}	T_{CCC-N}	LC_{CCC-NS}
(7,2,3,5)	17	4,064	4,064	4,065	32,512	32,508
(7,3,4,5)	19	4,064	4,064	4,065	65,024	65,025
(7,2,3,7)	19	16,256	16,256	16,257	130,048	130,048
(7,3,4,7)	21	16,256	16,256	16,257	520,192	520,186
(7,3,5,7)	23	16,256	16,256	16,255	1,040,384	1,040384
(7,2,3,11)	23	260,096	260,096	260,090	2,080,768	2,080,769
…	…	…	…	…	…	…
(127,23, 35,71)	256	(a)	(b)			

[Note] (a) : $(2^{127}-1)\cdot 2^{71}$, (b) : $((2^{127}-1)\cdot 2^{71})$

표 3에서 M열은 실수 상에서 더해지는 m-수열들의 합($M = L_1+N_2+N_3+N_4$)이다. $T_{\min}$열은 식(12)에 따른 합산 수열의 계산된 주기 값이다. T_{NSG}열은 NSG에서 시뮬레이션 주기 값을 나타내고, $T_{CCC-NSG}$는 $CCC-NSG$에서 시뮬레이션 주기 값을 나타낸다. LC_{NSG}열은 NSG에서 그리고 $LC_{CCC-NSG}$는 $CCC-NSG$에서 각각 언급된 차수의 다른 원시 다항식의 가능한 모든 조합으로 얻어진 최소 선형복잡도 LC값을 보여준다.

짧은 단수에 대한 표 3의 시뮬레이션 결과에서 de Bruijn 수열의 주기 $T_{\min}$은 시뮬레이션 주기 값 T_{NSG}와 정확하게 일치함을 확인하였고, $T_{CCC-NSG}$는 $T_{\min}$보다 크게 나타남을 확인할 수 있었다. 또한 선형복잡도 LC_{NSG}는 주기 T_{NSG}에 근사함을 보여 주었고, $LC_{CCC-NSG}$는 $T_{CCC-NSG}$에 근사함을 보여 주었다. 따라서 N값을 256으로 확장하더라도 LC_{NSG}와 $LC_{CCC-NSG}$는 각각 시뮬레이션 주기 T_{NSG}와 $T_{CCC-NSG}$에 근사하게 될 것으로 예측된다. 표 3의 시뮬레이션 값으로부터 아래의 식(13)~(16)의 추정 하한 값을 예측할 수 있다.

$$T_{NSG} = T_{\min} \tag{13}$$

$$T_{CCC-NSG} > T_{\min} \tag{14}$$

$$LC_{NSG} \approx T_{NSG} \tag{15}$$

$$LC_{CCC-NSG} \approx T_{CCC-NSG} \tag{16}$$

NSG와 $CCC-NSG$의 설계 보안 강도(보안 레

벨)는 2^{128}이며, 키 수열특성은 비선형성을 유지하면서 선형복잡도 및 주기의 하한 값을 예측할 수 있다. $CCC-NSG$의 경우에는 NSG보다 주기와 LC 값이 크게 되므로 안전성을 더 개선할 수 있다.

표 4에서는 $CCC-NSG$에서 시뮬레이션 주기 $T_{CCC-NSG}$의 특성을 분석하기 위한 세부적인 데이터를 보여주고 있다. de Bruijn 수열의 주기는 식(12)와 같이 기본적으로 $LFSR_1$ 과 $NFSR_4$에 의해 결정된다. 그러나 순환 클럭 조절형 구조로 개선되면 $NFSR_2$ 와 $NFSR_3$의 영향이 de Bruijn 수열의 주기에 반영되는 것을 표 4의 결과에서 볼 수 있다.

표 4의 결과 값을 통해 예측되는 시뮬레이션 주기 $T_{CCC-NSG}$는 식(17) 과 같다.

$$T_{CCC-NSG} = 2^{(N_2+N_3-2)} \cdot T_{\min} \quad (17)$$

이때 L_1과 N_4는 소수, $N_2 \geq 2$, $N_3 \geq 3$, $N_2 < N_3$ 그리고 $N_2+N_3 \ll N_4$ 조건이 되도록 L_1, N_2, N_3 및 N_4의 값을 선택하였다.

표 4. $CCC-NSG$ 에서 시뮬레이션 된 주기 $T_{CCC-NSG}$

(L_1, N_2, N_3, N_4)	N_2+N_3	$T_{\min}$	$T_{CCC-NSG}$	
			시뮬레이션 결과	간략화
(7, 2, 3, 7)	5	16,256	130,048	$2^3 \cdot T_{\min}$
(7, 2, 3,11)		260,096	2,080,768	
(11, 2, 3, 7)		262,016	2,096,128	
(11, 2, 3,11)		4,192,256	33,538,048	
(13, 2, 3, 7)		1,048,448	8,387,584	
(13, 2, 3,11)		16,775,168	134,201,344	
(15, 2, 3, 7)		4,194,176	33,553,408	
(7, 2, 4, 7)	6	16,256	260,096	$2^4 \cdot T_{\min}$
(7, 2, 4,11)		260,096	4,161,536	
(11, 2, 4,11)		4,192,256	67,076,096	
(7, 3, 4, 7)	7	16,256	520,192	$2^5 \cdot T_{\min}$
(7, 3, 4,11)		260,096	8,323,072	
(11, 3, 4,11)		4,192,256	134,152,192	
(7, 2, 5, 7)		16,256	520,192	
(7, 2, 5,11)		260,096	8,323,072	
(11, 2, 5,11)		4,192,256	134,152,192	
(7, 3, 5, 7)	8	16,256	1,040,384	$2^6 \cdot T_{\min}$
(7, 3, 5,11)		260,096	16,646,144	
(11, 3, 5,11)		4,192,256	268,304,384	
(7, 2, 7,11)	9	260,096	33,292,288	$2^7 \cdot T_{\min}$
(7, 3, 6,11)		260,096	33,292,288	
(11, 2, 7,11)		4,192,256	536,608,768	
(7, 2, 8,11)	10	260,096	66,584,576	$2^8 \cdot T_{\min}$
(7, 3, 7,11)		260,096	66,584,576	
(7, 2, 9,11)	11	260,096	133,169,152	$2^9 \cdot T_{\min}$
(7, 3, 8,13)		1,040,384	532,676,608	
(7, 2,10,13)	12	1,040,384	1,065,353,216	$2^{10} \cdot T_{\min}$
(7, 3, 9,13)		1,040,384	1,065,353,216	
(7, 2,11,17)	13	16,646,144	34,091,302,912	$2^{11} \cdot T_{\min}$
(7, 5, 8,17)		16,646,144	34,091,302,912	
(7, 2,12,17)	14	16,646,144	68,182,605,824	$2^{12} \cdot T_{\min}$
(7, 5, 9,17)		16,646,144	68,182,605,824	
…	…	…	…	
(127,23,35,71)		(a)	(b)	$2^{56} \cdot T_{\min}$

[Note] (a) : $(2^{127}-1) \cdot 2^{71}$, (b) : $2^{56} \cdot (2^{127}-1) \cdot 2^{71}$

3.2 공격 분석

본 절에서는 $CCC-NSG$의 각개공격(divide-and-conquer attack), 고속 상관성 공격, 시간/메모리/데이터 거래(Time/Memory/Data Tradeoff) 공격에 대하여 설명한다.

3.2.1 각개공격

알려진 각개공격 알고리즘[3]에 의한 공격은 정확한 클럭을 알고 있어야만 공격이 가능하며, $CCC-NSG$에 적용된 순환 클럭 조절형 구조는 랜덤한 클럭 정보를 예측할 수 없기 때문에 안전하다.

3.2.2 고속 상관공격

표 5는 $CCC-NSG$에서 메모리 비트 c_{t+1}^0와 키 출력 수열 비트 z_t간의 상관관계를 보여주는데, 이때 c_{t+1}^0와 z_t가 같아질 확률 p($c_{t+1}^0 = z_t$)는 0.5 로 나타난다. $z_t = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus c_t^0$ 이고 $z_{t+1} = x_{t+1}^1 \oplus x_{t+1}^2 \oplus x_{t+1}^3 \oplus x_{t+1}^4 \oplus c_{t+1}^0$으로 표현할 때, $c_{t+1}^0 = z_t \oplus 1$이 확률 0.5 로 유지되기 때문에 $z_{t+1} = x_{t+1}^1 \oplus x_{t+1}^2 \oplus x_{t+1}^3 \oplus x_{t+1}^4 \oplus z_t \oplus 1$이 확률 0.5를 유지한다.

z_t'을 키 수열의 이진 시차 비트라 두면, 임의 시점에서 $z_t' = z_{t+1} \oplus z_t$가 되고, 이때 $z_t' = x_{t+1}^1 \oplus x_{t+1}^2 \oplus x_{t+1}^3 \oplus x_{t+1}^4 \oplus 1$이 된다. $CCC-NSG$의 이진 시차 수열 z_t'은 그림 5와 같이 1개의 LFSR과 3개의 NFSR의 출력 합 그리고 이진잡음 e_t를 더한 형태로 나타낼 수 있으며, 이 모델에서 잡음 확률 e_t는 0.0 이다. 따라서 이 모델에서는 상관성 공격[4]이 어렵다고 할 수 있다.

$CCC-NSG$에 대한 고속 상관성 공격 알고리즘은 이동레지스터 수열에 기반한 잡음 모델로 생각할 수 있다. 패리티 검사에 기초한 반복 확률 알고리즘이 관측된 수열에 대한 이진 시차 수열로부터 이동레지스터 수열을 재구성할 목적으로 에러 정정 과정을 수행할 수 있도록 한다.

표 5. $CCC-NSG$의 c_{t+1}^0와 z_t의 상관특성

x_t^1	x_t^2	x_t^3	x_t^4	c_t^0	c_t	c_{t+1}^0	c_{t+1}	Z_t
0	0	0	0	0	00	0	00	0
					10	0	10	
0	0	0	0	1	01	0	10	1
					11	0	00	
0	0	0	1	0	00	0	00	1
					10	0	10	
0	0	0	1	1	01	1	11	0
					11	1	11	
0	0	1	0	0	00	0	00	1
					10	0	10	
0	0	1	0	1	01	1	11	0
					11	1	11	
0	0	1	1	0	00	1	01	0
					10	1	01	
0	0	1	1	1	01	1	11	1
					11	1	11	
0	1	0	0	0	00	0	00	1
					10	0	10	
0	1	0	0	1	01	1	11	0
					11	1	11	
0	1	0	1	0	00	1	01	0
					10	1	01	
0	1	0	1	1	01	1	11	1
					11	1	11	
0	1	1	0	0	00	1	01	0
					10	1	01	
0	1	1	0	1	01	1	11	1
					11	1	11	
0	1	1	1	0	00	1	01	1
					10	1	01	
0	1	1	1	1	01	0	00	0
					11	0	10	
1	0	0	0	0	00	0	00	1
					10	0	10	
1	0	0	0	1	01	1	11	0
					11	1	11	
1	0	0	1	0	00	1	01	0
					10	1	01	
1	0	0	1	1	01	1	11	1
					11	1	11	
1	0	1	0	0	00	1	01	0
					10	1	01	
1	0	1	0	1	01	1	11	1
					11	1	11	
1	0	1	1	0	00	1	01	1
					10	1	01	
1	0	1	1	1	01	0	00	0
					11	0	10	
1	1	0	0	0	00	1	01	0
					10	1	01	
1	1	0	0	1	01	1	11	1
					11	1	11	
1	1	0	1	0	00	1	01	1
					10	1	01	
1	1	0	1	1	01	0	00	0
					11	0	10	
1	1	1	0	0	00	1	01	1
					10	1	01	
1	1	1	0	1	01	0	00	0
					11	0	10	
1	1	1	1	0	00	0	10	0
					10	0	00	
1	1	1	1	1	01	0	00	1
					11	0	10	

[Note] $c_t = (c_t^1, c_t^0)$, $c_{t+1} = (c_{t+1}^1, c_{t+1}^0)$

고속 상관성 공격 알고리즘[4]은 다음과 같이 나타내어진다.

① 관찰된 키 수열로부터 이진 시차수열을 계산한다.

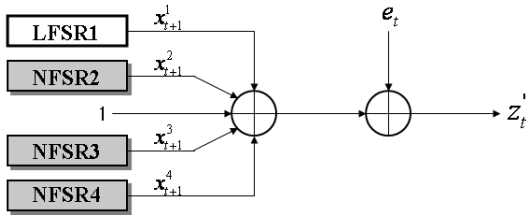


그림 5. $CCC-NSG$의 고속 상관성 공격 모델

② 각각의 이진시차수열 z_t, $t=1,2,3,...,k$에 대하여 패리티 검사 값을 계산한다.

③ 각 z_t에 대한 패리티 검사 값들을 이용하여 오차의 확률 p_t를 계산한다.

④ 만일 $p_t > 0.5$이면, $t=1,2,3,...,k$에서의 $z_t' = z_t' \oplus 1$과 $p_j = 1-p_j$를 설정한다.

⑤ 모든 패리티 검사들이 만족할 때까지 되풀이 한다.

x를 개별 이동레지스터 피드백 다항식에 대한 전체 차수라고 할 때, 고속상관성 공격에 대한 복잡도 및 키 수열 요구량은 $O(2^{x/4})$ 이다[4].

3.2.3 시간/메모리/데이터 거래 공격

시간/메모리/데이터 거래 공격[10]의 목적은 주어진 시간 내에 내부 상태를 찾아내는데 있으며, 공격은 두 단계로 처리된다. 선 처리 단계 동안에 암호 해독기는 가능한 내부 상태를 출력 키 수열과 연관된 접두어에 검사테이블(look-up table)을 작성한다. 실제 공격 단계에서는, 검사테이블 검색을 통하여 알려진 키 수열 일부 비트를 가지고 유사한 내부 상태를 발견하려 한다.

S,M,T,P 그리고 D는 각각 내부 상태의 공간 크기, ($\log_2^S$와 같은 이진 워크 크기에서의)메모리 용량, (검사 테이블에 대한)계산시간, (검사 테이블에 대한) 사전 계산 시간, 그리고 (키 갱신이 없는) 데이터 길이(즉, 알려진 데이터의 길이)를 표시한다. 시간/메모리/데이터 거래 공격[10]은 $T.M=s$, $P=M$ 그리고, $D=T$를 만족한다. 256비트의 내부 상태를 갖는 $CCC-NSG$에 대하여, T 나 M 이 2^{128}보다 더 크게 나타나며, 이는 키 전수공격보다 더 어렵다.

3.3 성능비교

제안된 $CCC-NSG$알고리즘은 E_0에서 적용한 합산수열발생기의 취약점을 보강하였다. 즉, 취약점이 알려진 기존의 합산수열발생기에서 선형 입력 $LFSR$을 대체하고자 비선형 함수인 de Bruijn 수

표 6. 성능 요약 표

구분		E_0 합산수열 발생기	NSG	제안된 $CCC-NSG$
구성 요소	LFSR 개수	4	1	1
	NFSR 개수	0	3 (de Bruijn)	3 (de Bruijn)
	클럭조절형 구조	없음	없음	있음
보안 요소	시뮬레이션 주기	T_{E_0}	$T_{\min}$	$2^{(N_2+N_3-2)} \cdot T_{\min}$
	LC	$\approx T_{E_0}$	$\approx T_{\min}$	$\approx 2^{(N_2+N_3-2)} \cdot T_{\min}$
security analysis	각개공격	weak	secure	secure
	고속상관성 공격	weak	secure	secure
	TMTO 공격	weak	secure	secure

[Note] $T_{E_0} = (2^{L_1}-1)(2^{L_2}-1)(2^{L_3}-1)(2^{L_4}-1)$, 여기서 L_1, L_2, L_3 및 L_4 는 4개 LFSR 각각의 길이

열을 적용하였다. 그리고 각개공격 등에 대한 취약점을 보강하고, 정확한 클럭 수를 예측할 수 없도록 랜덤한 클럭을 발생시키기 위한 순환 클럭 조절형 구조를 적용하였다.

표 6에서와 같이 $CCC-NSG$는 비선형 요소인 de Bruijn 발생기를 적용함으로서 기존의 공격방법에 안전함을 확인하였다. 또한 주기 $T_{CCC-NSG}$및 선형복잡도 $LC_{CCC-NSG}$가 NSG에 비하여 더 큰 값을 갖게 되며, 안전성이 개선됨을 확인하였다.

Ⅳ. 결 론

기존의 E_0 블루투스 암호 알고리즘은 LFSR을 입력으로 하는 합산수열 발생기를 사용하였다. 본 논문에서는 더 높은 비 선형성을 얻기 위해서 기존의 합산수열 발생기에서 선형 LFSR의 일부를 비선형 NFSR로 교체하였고, 또한 클럭 예측을 어렵게 하는 순환 클럭 조절형 $CCC-NSG$를 제안하였다. 제안된 $CCC-NSG$의 설계 보안강도(보안 레벨)는 2^{128}수준이며, 여러 가지 안전성 분석에 안전함을 보였다. 따라서 $CCC-NSG$는 근거리 무선통신 기술의 표준으로 자리 잡고 있는 블루투스 기술의 보안성을 크게 향상시킬 수 있다.

참 고 문 헌

〔1〕 "*Specification on the Bluetooth System*", version 1.1, February 22 2001.

〔2〕 R.Rueppel, "Correlation Immunity and the Summation Generator," *Advances in Cryptology-CRYPTO '85, Lecture Notes in Computer Science*, Vol.218, pp.260-272, Springer-Verlag, 1985.

〔3〕 E.Dawson, "Cryptanalysis of Summation Generator," *Advances in Cryptology -ASIACRYPT '92, Lecture Notes in Computer Science*, Vol.718, pp.209-215, Springer-Verlag, 1993.

〔4〕 J.Golic, M.Salmasizadeh, and E.Dawson, "Fast Correlation Attacks on the Summation Generator," *Journal of cryptology*, Vol.13, No.2, pp.245-262, 2000.

〔5〕 W.Meier and O.Staffelbach, "Correlation Properties of Combiners with Memory in Stream Ciphers," *Advances in Cryptology-EUROCRYPT' 90, Lecture Notes in Computer Science*, Vol.473, pp.204-213, Springer-Verlag, 1990.

〔6〕 T.Chang, B.Park, Y.H.Kim, "An Efficient Implementation of the D-Homomorphism for Generation of de Bruijn Sequences," *IEEE Transactions on Information Theory*, Vol.45, No.4, pp.1280-1283, May 1999.

〔7〕 T.Chang, I.Song, "Cross-Joins in de Bruijn Sequences and Maximum Length Linear Sequences", *IEICE Transactions Fundamentals*, Vol.E76-A, No.9, pp.1494-1501, September 1993.

〔8〕 M.Hell, T.Johansson, W.Meier, "Grain-A Stream Cipher for Constrained Environments," *International Journal of Wireless and Mobile Computing,* Vol.2, No.1 pp.86-93, 2007.

〔9〕 J.Massey, "Shift-Register Synthesis and BCH Decoding," *IEEE Transactions on Information Theory*, IT-15, No.1, pp.122-127, January 1969.

〔10〕 S.Babbage, "Improved Exhaustive Search Attacks on Stream Cipher", *European Convention on Security and Detection, IEEE Conference Publication*, Vol. 408, pp. 161-166, 1995.

〔11〕 김형락, 이훈재, 문상재, "*NSG* : 비선형 알고리즘을 이용한 블루투스 E_0 암호화시스템의 성능 개선," *정보처리학회논문지*, 제 16-C권 제3호, June 2009.

김 형 락 (Hyeong-rag Kim) 정회원

1992년 2월 경북대학교 전자공학과 졸업(공학사)
1994년 2월 경북대학교 대학원 전자공학과(공학석사)
1999년 2월 경북대학교 대학원 전자공학과(박사수료)
1994년 1월~1995년10월 LG 전자기술원 영상미디어연구소 연구원
1995년 11월~1996년 2월 (주)문화방송 기술연구소 연구원
1996년 3월~현재 포항대학 컴퓨터응용과 부교수
<관심분야> 암호이론, 정보통신, 이동네트워크, u-네트워크 보안 등

이 훈 재 (Hoon-jae Lee) 정회원

1985년 2월 경북대학교 전자공학과 졸업(공학사)
1987년 2월 경북대학교 대학원 전자공학과(공학석사)
1998년 2월 경북대학교 대학원 전자공학과(공학박사)
1987년 2월~1998년 1월 국방과학연구소 선임연구원(개발팀장)
1998년 3월~2002년 2월 경운대학교 컴퓨터공학과 조교수
2002년 3월~현재 동서대학교 컴퓨터정보공학부 부교수
2007년 6월~현재 동서대학교 유비쿼터스 IT전문인력양성사업단장(NURI)
2008년 9월~현재 동서대학교 유비쿼터스 헬스케어 사업단장(BK21)
<관심분야> 암호이론, 정보통신/네트워크, u-네트워크 보안, 부채널 공격 등

문 상 재 (Sang-jae Moon) 종신회원

1972년 2월 서울대학교 공업교육(전자)과 졸업(공학사)
1974년 2월 서울대학교 대학원 전자공학과(공학석사)
1984년 6월 미국 UCLA 전자공학과(공학박사)
1984년 7월~1985년 6월 UCLA Postdoctoral 근무
1984년 7월~1985년 6월 미국 OMNET 컨설턴트
1974년 12월~현재 경북대학교 전자전기컴퓨터학부 교수
2000년 8월~2008년 2월 경북대학교 이동네트워크 정보보호기술 연구센터 소장
2002년 2월~현재 한국정보보호학회 명예회장
<관심분야> 정보보호, 디지털 통신, 이동 네트워크 등