[4-3]

단일 최대장 LFSR 비선형 스트림 암호기내에서 오류제어와 선형복잡도에 관한 연구

○차 명 식, 이 차 연, 김 용 진, 이 만 영

한양대학교 전자통신공학과

Error Control and Linear Complexity in Nonlinear Stream Cipher using a Maximal Length LFSR

Myoung Sik Cha, Cha Youn Lee, Yong Jin Kim, Man Young Rhee

Dept. of Electronic Communication engineering, Hanyang Univ.

ABSTRACT

To develop the techniques for transforming a maximal length LFSR in the stream cipher system into the non linear stream generator, the corresponding linear equivalent generator and linear complexity are intensively studied in this paper. When the ciphertext encripted through the stream cipher is transmitted via the noise channel, error propagation will occur due to channel errors. Therefore, error correcting coding is needed for preventing the error propagation from the ciphertext error. Depending on where the coder device is attached to the cipher system, cryptographic analysis will produce different results for three feedback modes.

1. 서론

전송로 상의 정보는 항상 제 3자가 취득할 수 있다고 가정할 때, 정보를 보호하기 위해서는 암호시스템을 구성하는 것이 필요하다. 스트림암호 시스템에는 키이스트림을 생성하는 방법에 따라 키이자동키이법과 평문 귀환법과 암호문 귀환법이 있다.

암호시스템에서는 불가피하게 생기는 자연발생 오류를 제어하기 위하여 부호이론을 도입한다. 오류제어기를 암호기에 연결할때 그 연결순서에 따라 그 결과는 달라진다. 본 논문에서는 스트림 암호기의 무작위성(random)을 높이기 위해서 비선형함수를 사용한 키이스트림 생성기를 분석하고자 한다. 2장에서는 스트림 암호기의 일반적인 성질및 오류전파에 대해 설명한다. 그리고 최대장 LFSR를 이용한 비선형 키이스트림 생성기를 분석하고 선형복잡도를 나타낸다. 3장에서는 BCH부호를 간략하게 나타내며, 부호기를 암호기에 연결하였을때 연결위치에 따라 결과가 달리짐을 보인다.

2. 스트림 암호기

스트림 암호기에서는 평문이 키이스트림에 의해서 암호화되어 암호문을 이루고, 키이스트림은 키이와 스트림생성기의 내부상태에 의해서 생성된다.

2.1. 최대장 LFSR를 이용한 비선형 키이스트림생성기

키이스트림생성기에 의해서 생성된 출력계열의 무작위성, 불확실성, 등가성은 선형복잡도에 의해서 제공된다. 따라서 선형복잡도를 나타내어 출력계열의 비선형 결합을 분석한다.

다음은 비선형 스트림생성기의 분석에 필요한 개념들이다.

1) 대수표준형(ANF, algebraic normal form): 비선형함수를 일반적으로 나타내기 위해 레지스터의 각 출력들을 곱의 합으로 나타내는 기본적인 형태이다.

2) 선형복잡도(linear complexity): 키이스트림을 생성할 수 있는 가장 짧은 LFSR의 길이를 나타낸다.

본 논문에서는 레지스터의 내용을 결정하는 결합다항식(connection polynomial)으로 원시기약다항식(primitive irreducible polynomial)을 사용한 5단-최대장 LFSR을 사용한 비선형생성기를 분석한다. 그림 1에서, 결합다항식 $C(D) = 1 + D^2 + D^5$이고 LFSR의 초기 상태는 (01111)라 한다. LFSR 각 단의 출력계열 Z는 최대 주기를 가지므로 31차원 벡터들이고 ANF로 나타낼 수 있다.[7]

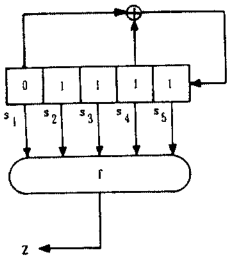

그림 1. 비선형함수 f를 이용한 $\langle 5, 1+D^2+D^5 \rangle$ 최대장 LFSR 생성기

$$
\begin{aligned}
Z = {} & a_1 s_1 + \cdots + a_5 s_5 + a_{12} s_1 s_2 + a_{13} s_1 s_3 \\
& + \cdots + a_{45} s_4 s_5 + a_{123} s_1 s_2 s_3 + a_{124} s_1 s_2 s_4 \\
& + \cdots + a_{345} s_3 s_4 s_5 + a_{1234} s_1 s_2 s_3 s_4 + \cdots \\
& + a_{2345} s_2 s_3 s_4 s_5 + a_{12345} s_1 s_2 s_3 s_4 s_5
\end{aligned} \tag{1}
$$

또 행렬(matrix) 표현은 다음과 같다.

$$Z = P^t \cdot a \tag{2}$$

여기서 P는 표 1에서처럼 모든 가능한 s의 곱벡터(product vector)를 열벡터(row vector)로 하는 행렬이고, a는 계수벡터이다. 주기 31인 비선형생성기를 등가선형 LFSR생성기로 나타내기 위해서 순수순환치환레지스터(pure cyclic shift register)를 분석하는 것이 필요하다. 순수순환치환레지스터의 결합다항식은 다음과 같다.[5]

표 1. f의 ANF single product항에 의해 생성된 기저벡터.

ANF(f) 함수에서 Single product	Single product 항에 의해서 생성된 초기 주기 벡터
1	0111110011010010000101011101100
2	1111100110100100001010111011000
3	1111001101001000010101110110001
4	1110011010010000101011101100011
5	1100110100100001010111011000111
12	0111100010000000000000011001000
13	0111000001000000000101010100000
14	0110010010010000000001001100000
15	0100110000000000000101011000100
23	1111000100000000000000110010000
24	1110000010000000001010101000000
25	1100100100100000000010011000000
34	1110001000000000000001100100001
35	1100000100000000010101010000001
45	1100010000000000000011001000011
123	0111000000000000000000010000000
124	0110000010000000000000001000000
125	0100100000000000000000011000000
134	0110000000000000000001000100000
135	0100000000000000000101010000000
145	0100010000000000000001001000000
234	1110000000000000000000100000000
235	1100000100000000000000010000000
245	1100000000000000000010001000000
345	1100000000000000000000100000001
1234	0110000000000000000000000000000
1235	0100000000000000000000010000000
1245	0100000000000000000000001000000
1345	0100000000000000000001000000000
2345	1100000000000000000000000000000
12345	0100000000000000000000000000000

$$
\begin{aligned}
1 + D^{31} = {} & (1 + D)(1 + D^2 + D^5)(1 + D^3 + D^5) \\
& (1 + D + D^2 + D^3 + D^5)(1 + D + D^2 + D^4 + D^5) \\
& (1 + D + D^3 + D^4 + D^5)(1 + D^2 + D^3 + D^4 + D^5)
\end{aligned}
$$

따라서 위 7개의 기약결합다항식을 가진 7개의 LFSR의 결합으로 선형등가 LFSR 생성기를 구성하여 일반적인 선형분해등가도를 그릴 수 있다. 이 등가도에서 레지스터 각 단의 위치에 순차적으로 1을 하나씩 넣으므로써, 31개의 31차 선형독립벡터를 얻을 수 있다. 이 벡터들을 기저(basis)로 하여 출력벡터를 표현한다.

$$Z = r_1 d_1 + r_2 d_2 + \cdots + r_{31} d_{31} \tag{3}$$

또 행렬표현은 다음과 같다.

$$Z = D^t \cdot r \tag{4}$$

여기서, D는 행렬의 열이 기저벡터 $d_1^t, d_2^t, \cdots, d_{31}^t$이다. r은 일반적인 선형등가도의 31 상태 초기값이다(표 2).

표 2. 일반적인 분해등가도의 기저벡터 d_i^t.

상태 비트 r_i	$r_i = 1$ 이고 $r_j = 0$, $j \neq i$ 일때, 생성된 초기 주기 벡터 d_i^t.
1	1111111111111111111111111111111
2	1000010101110110001111100110100
3	0100001010111011000111110011010
4	0010000101011101100011111001101
5	0001010111011000111110011010010
6	0000101011101100011111001101001
7	1000010010110011111000110111010
8	0100001001011001111100011011101
9	0010010110011111000110111010100
10	0001001011001111100011011101010
11	0000100101100111110001101110101
12	1000011001001111101110001010110
13	0100001100100111110111000101011
14	0010011111011100010101101000011
15	0001010110100001100100111110111
16	0000110010011111011100010101101
17	1000011010100100010111110110011
18	0100010111110110011100001101010
19	0010001011111011001110000110101
20	0001011111011001110000110101001
21	0000110101001000101111101100111
22	1000011100110111110100010010101
23	0100010010101100000111001101111
24	0010010101100001110011011111010
25	0001001010110000111001101111101
26	0000111001101111101000100101011
27	1000010110101000111011111001001
28	0100011101111100100110000101101
29	0010011000010110101000111011111
30	0001011010100011101111100100110
31	0000101101010001110111110010011

표 2는 계열 Z의 선형복잡도를 결정하고, 특히 이용된 다항식중의 어느 것이 Z의 생성에 기여한지를 결정한다.

예를 들어서, 주기가 31인 첫주기의 출력계열이

$$Z \quad 00100111001011111001110100010101 \qquad (5)$$

일 때, 선형등가도의 레지스터 초기상태를 결정할 수 있다. 식(4)을 이용하면 다음과 같다.

$$r = (D^t)^{-1} \cdot Z \qquad (6)$$

$$r = (011001111101000000011011101110110)$$

위 결과를 가지고 완전한 선형분해등가도를 그릴 수 있고, 식(5) 계열의 선형복잡도는 25이다.

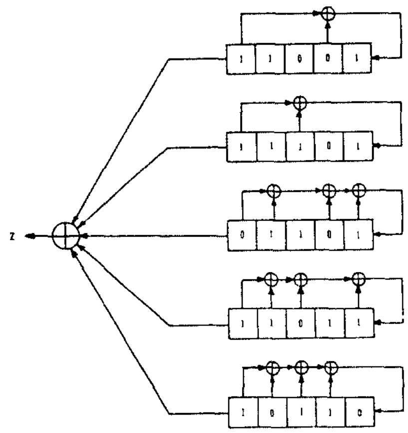

그림 2. 출력계열 Z에 대한 선형분해 등가도

그림 2를 단일 LFSR로 나타낸 것이 그림 3이다. 이 생성기의 결합다항식은 그림 2의 결합다항식들의 곱에 해당된다.

$$C'(D) = (1 + D^2 + D^5)(1 + D^3 + D^5)(1 + D + D^2 + D^4 + D^5)$$
$$(1 + D + D^3 + D^4 + D^5)(1 + D^2 + D^3 + D^4 + D^5)$$
$$= 1 + D^4 + D^5 + D^6 + D^8 + D^{10} + D^{13} + D^{15} + D^{16}$$
$$+ D^{17} + D^{18} + D^{21} + D^{22} + D^{24} + D^{25}$$

그리고 LFSR의 초기치는 출력계열 Z의 처음 25비트이다.

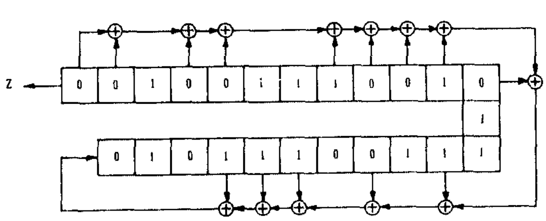

그림 3. 출력계열 Z에 대한 선형결합 등가도

이제 식(2)로 부터 비선형함수 f의 계수 a를 구하자.

$$a = (P^t)^{-1} \cdot Z \qquad (7)$$

$$a = (1110100110100101111100010010010)$$

$$f(s_1, s_2, s_3, s_4, s_5) = s_1 + s_2 + s_3 + s_5 + s_1s_4 + s_1s_5$$
$$+ s_2s_3 + s_2s_5 + s_1s_5 + s_1s_2s_3 + s_1s_2s_4 + s_1s_3s_4$$
$$+ s_2s_3s_5 + s_1s_4s_5 + s_1s_2s_3s_5 + s_1s_3s_4s_5$$

따라서 완전한 비선형 스트림생성기가 얻어진다(그림 4).

결과적으로 주기가 $2^L - 1$인 출력계열의 비선형생성기와 선형등가도를 유도하는 것이 가능한을 보였다.

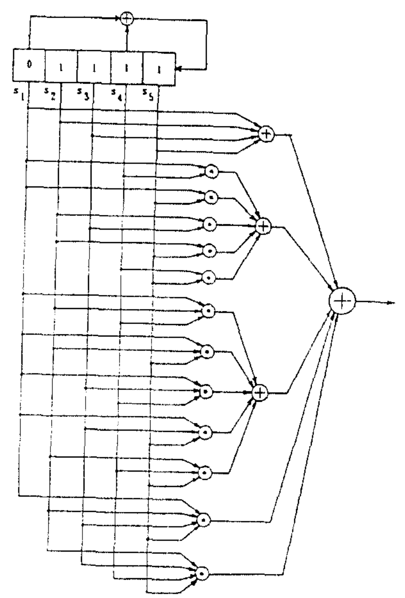

그림 4. 출력계열 Z에 대한 비선형 생성기

2.2. 스트림 암호기의 오류전파

2.2.1. 키이자동키이법

키이자동키이법은 키이스트림 자체의 귀환으로 키이스트림 비트가 얻어진다. 키이스트림 비트는 이전의 키이스트림 비트의 영향을 받으나, 그 이전의 평문 비트와 암호문 비트에는 무관하다. 채널전송중 암호문 한 비트에 오류가 생기면 deciphering 변환시 대응되는 평문 한 비트에 잘못 deciphering 될 뿐 다른 비트에는 영향이 없다. 그러나 암호문의 지연, 삽입등으로 동기화를 잃으면 그 후 다시 동기화 시키지 않는 한 수신되는 모든 메시지는 쓸모없게 된다.

2-2-2. 평문 귀환법

평문 귀환법은 평문 비트가 키이스트림 비트와 EX OR 하여 암호문 비트$_i$를 만듦과 동시에 레지스터에 입력되어 이후에 키이스트림 비트에 영향을 주는 암호법이다. 암호문 비트는 대응 평문비트뿐만 아니라 일정량의 이전 평문비트에 의존하고 deciphering시 평문 비트는 대응 암호문 비트뿐만 아니라 이전의 deciphering된 평문 비트들에 의존한다. 따라서 채널전송시 하나의 오류가 발생할 때, deciphering시 오류를 포함한 암호문 비트 가 잘못 deciphering되는 것은 물론 그 이후로 오류를 포함하지 않는 암호문이 입력되어도 모든 평문이 오류를 포함한채 deciphering 된다. 이것을 오류전파(error propagation)라 한다.

2-2-3. 암호문 귀환법

암호문 귀환법은 레지스터의 입력을 암호문비트로 하는 암호법이다. 암호문비트는 대응 평문뿐만 아니라 일정량의 이전 암호문 비트에 의존하고, deciphering시 평문비트는 (n+1)개의 암호문 비트에 의존한다. 따라서 전송채널을 통해 하나의 오류가 발생했을때 deciphering된 평문에는 최대 (n+1)비트의 블럭내에서 오류가 발생한다.

3. 오류제어기

cipher와 coder의 연결에는 두가지 방법이 있다. 하나는 외부오류제어(external error control)이고 다른 하나는 내부오류제어(internal error control)로 어느 방법을 선택하느냐에 따라 결과가 달라진다.

3-1. RS 부호

GF(2^m)상에서 오류정정능력이 t인 (n,k) RS부호는 다음과 같은 변수들을 갖는다.

부호길이 : $n = 2^m - 1$

정보길이 : $k = n - 2t$

최소거리 : $d_{min} = 2t + 1$

t중 오류정정 (n,k) RS 부호의 생성다항식은 다음과 같다.

$$g(x) = (x + \alpha)(x + \alpha^2)\cdots(x + \alpha^{2t}) \qquad (8)$$

정보다항식을 $d(x) = d_0 + d_1 x + \cdots + d_{k-1}x^{k-1}$라 하고, 검사다항식 $p(x) = p_0 + p_1 x + \cdots + p_{n-k-1}x^{n-k-1}$ 라 하면 RS 부호의 조직형 부호다항식은 다음과 같다.

$$c(x) = p(x) + x^{n-k}d(x) \qquad (9)$$

이 때 p(x)는 d(x)에 x^{n-k}를 곱하여 g(x)로 나누었을 때의 나머지 값이다.

c(x)를 전송하였을 때 채널상에서 오류가 발생하게 된다. decoder에서 e(x)를 구하는 절차는 다음과 같다.

1. 오증(syndrome) 구한다.

$$s_i = r(\alpha^i), \qquad 1 \leq i \leq 2t \qquad (10)$$

2. Peterson-Gorenstein-Zierler 알고리즘을 이용하여 오류위치다항식(error locator polynomial)을 구한다.

$$\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_\nu x^\nu \qquad (11)$$

3. Chien의 탐지법을 사용하여 오류위치다항식의 근을 구하고 이 근의 역수를 오류위치로 한다.

4. 오류추정다항식(error evaluator polynimial)을 구한다.

$$\Omega(x) = 1 + (s_1 + \sigma_1)x + \cdots + (s_{\nu-1} + \sigma_1 s_{\nu-2} + \sigma_{\nu-1})x^{\nu-1}$$

5. 오류위치가 x^{j_m} 일 때 오류치 e_{j_m} 을 구한다.

$$e_{j_m} = \frac{\Omega(\alpha^{-j_m})}{\prod_{\substack{l=1 \\ l \neq m}}^{\nu}(1 + \alpha^{j_l}\alpha^{-j_m})}, \quad 0 < m \leq \nu \qquad (12)$$

3-2 외부오류제어

외부오류제어는 채널 상에서 볼 때 오류제어장치가 암호장치 외부에 있는 경우이다. 정보 M이 encipher를 통해 암호문 X가 되고 암호문 X가 encoder를 통해서 부호어 Y가 되므로, 채널상에서 발생한 오류는 decoder에 의해서 오류를 정정하고 정정된 암호문 X를 deciphering함으로써 올바른 정보 M이 얻어진다. 이 경우 cipher로 키이자동키이법, 평문귀환법, 암호문귀환법중 어느 것을 사용해도 올바른 정보를 얻을 수 있다.

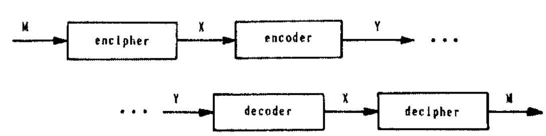

그림 5. 외부오류제어

3-3 내부오류제어

내부오류제어는 채널상에서 볼 때 오류제어장치가 암호장치 내부에 있는 경우이다. 평문과 암호문사이의 관계가

일대일 독립관계인 키이자동키이법에서는 내부오류제어를 사용할 때 오류에 의한 암호문내의 오류전파가 없으므로 오류정정능력 범위내의 모든 오류는 정정된다. 따라서 외부오류제어와 결과가 동일하다. 평문귀환법에서는 오류발생 후 모든 정보를 제대로 전송받지 못하고, 암호문귀환법에서는 오류발생 후 일정한 범위내에서 정보를 제대로 전송받지 못한다. 따라서 평문 귀환법이나 암호문 귀환법에서는 내부오류제어를 사용할 경우, 수신측에서는 오류발생 부분에대해 송신측에 재전송을 요구해야 할 것이다. 하지만 채널상에 동일 정보가 단 시간내에 반복되는 것은 바람직하지 못하다.

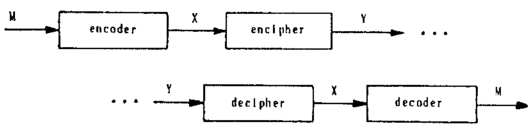

그림 6. 내부오류제어

3-4 예제

키이스트림 생성기는 2장에서 설명한 5단 최대장 LFSR을 이용한 비선형 키이스트림 생성기를 사용하고 부호는 $GF(2^5)$ 상의 2중 오류정정 (31,27) RS 부호를 사용한다. 이 때 생성 다항식은 $g(x) = x^4 + \alpha^{23}x^3 + \alpha^{17}x^2 + \alpha^{26}x + \alpha^6$ 이다. 전송할 정보 메시지는 다음과 같다.

There are two types of stream ciphers. One for which
the key - bit stream is independent of the plaintext
the other for which the key-bit stream is a function
of the plaintext or the ciphertext. For transmitting
the plaintext, it is needed to convert each message.

이 정보를 ASCII 코드로 나타내어 통신로를 통하여 송신측에서 수신측으로 정보가 전달될 때, 오류가 발생한 위치를 사각형으로 표시한다. 첫 줄에는 1 비트 오류를, 2번째 줄에는 1 바이트 오류를, 3 번째 줄에는 2 비트의 오류를, 4 번째와 5 번째 줄에는 2 바이트 오류를 포함시킨다. 그리고 그결과를 그림 7과 그림 8에 나타내었다.

There are two types of stream ciphers. One for which
the key - bit stream is independent of the plaintext
the other for which the key bit stream is a function
of the plaintext or the ciphertext. For transmitting
the plaintext, it is needed to convert each message.

그림 7. 외부오류제어

There are two types of stream ciphers. One for which
the key - bit stream is independent of the plaintext
the other for which the key bit stream is a function
of the plaintext or the ciphertext. For transmitting
the plaintext, it is needed to convert each message.

(a) 키이자동키이법

There are two types of stream ciphXV fU^EFvUCxyiZTD1
jZXDQ-L1FC4.X/*b5In4.[NDS&Qt{F8r0Δ C9!L{1{RMH[![e/B<
AyNC:SDj+>TRVZiZTGQ>FUAZ$<g_ N>AIV_)X1BPv&Li,pPXfGp
]|DN P1{07.B{F.3LD+>FUAZ+5S>"[_F.3B/Wq@]PH)[bFC<AxED
"/I/)rST N-^3@/0jRTWZ&PWzWYD-qR^KT8Pc_C3&O`9s"_|8yWS

(b) 평문 귀환법

There are two types of stream ciphers. One for which
the key - bit stream is independent of the plaintext
the other for whick the l&y-bit stream is a function
of the plaintext or the cR@hertext. For transmitting
the plaintei4, it is neY$ed to convert each message.

(c) 암호문 귀환법

그림 8. 내부오류제어

1. 결 론

본 논문에서는 암호화기법중 스트림 암호기를 키이스트림 생성기의 구조에따라, 키이자동키이법, 평문 귀환법, 암호문 귀환법으로 나누어서 각각의 오류전파를 나타내었다. 키이스트림의 무작위성을 높이기위해 비선형 함수를 사용한 단일 최대주기 LFSR 비선형 스트림 암호기를 사용하였고, 이것을 분석하기 위해서 선형복잡도의 개념을 도입하여 생성기의 동기성과 무작위성을 보였다.

암호기에 부호이론을 도입하였을 때 외부오류제어를 사용하면 채널상의 자연발생오류는 간단히 정정된다. 그러나 내부오류제어를 사용하면 키이자동키이법에서는 오류전파가 일어나지않기 때문에 사연발생오류는 간단히 정정되나 평문 귀환법에서는 오류전파에 의해서 첫 번째 오류이후 전부 decoding이 제대로 되지않고 암호문 귀환법에서는 오류발생 후 일정한 블럭내에서 decoding이 제대로 되지않는다.

참고 문헌

1. Rhee, M. Y., Error Correcting Coding Theory, McGraw-Hill, New York 1989.
2. 이 만 영, BCH부호와 Reed Solomon부호, 민음사, 1990.
3. E.L. Key, "An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators," IEEE Trans. on Inf. Vol. pp 732-736, Nov. 1976.
4. H.Beker, F. Piper, Cipher Systems, John Wiley and Sons, Inc., New York, 1982.
5. E.J. Groth, "Generation of Binery Sequences With Controllable Complexity", IEEE Trans. on Inf., Vol. pp 188-296, May 1971.
6. D.E.R. Denning, Cryptography and Data Security, Addison-Wesley Publishing Company.
7. R.A. Rueppel, Analysis and Design of stream Ciphers, Springer-Verlag, New York, 1986.