情報保護學會論文誌
第14券 第1號, 2004. 2

Summation Generator에 대한 대수적 공격

이 동 훈,†‡ 김 재 현, 한 재 우, 홍 진, 문 덕 재
국가보안기술연구소

Algebraic Attacks on Summation Generators

Dong Hoon Lee,†‡ Jaeheon Kim, Jae Woo Han, Jin Hong, Dukjae Moon
National Security Research Institute

요 약

n개의 LFSR과 l 비트의 메모리를 이용하는 combiner에 대하여 $\lceil \frac{n(l+1)}{2} \rceil$ 차 이하의 대수적 관계식이 존재하는 것이 이론적으로 밝혀졌다. 본 논문에서는 k 비트의 메모리를 사용하는 2^k개의 LFSR로 이루어진 summation Generator는 연속된 k+1개의 출력값을 이용하여 초기치에 관한 2^k차 이하의 대수적 관계식을 만들 수 있음을 보인다. 일반적으로 n개의 LFSR로 이루어진 summation Generator는 연속된 $\lceil \log_2 n \rceil$+1개의 출력값을 이용하여 초기치에 관한 $2^{\lceil \log_2 n \rceil}$ 차 이하의 대수적 관계식을 만들 수 있다.

ABSTRACT

It was proved that there is an algebraic relation of degree $\lceil \frac{n(l+1)}{2} \rceil$ for an (n, l)-combiner which consists of n LFSRs and l memory bits. For the summation generator with 2^k LFSRs which uses k memory bits, we show that there is a non-trivial relation of degree at most 2^k using k+1 consecutive outputs. In general, for the summation generator with n LFSRs, we can construct a non-trivial algebraic relation of degree at most $2^{\lceil \log_2 n \rceil}$ using $\lceil \log_2 n \rceil$+1 consecutive outputs.

Keywords: *stream cipher, algebraic attack, Summation generator*

Ⅰ. 서 론

LFSR을 기반으로하는 스트림 암호는 LFSR의 초기치를 비밀키로 하여 키 수열을 생성하고, 생성된 키 수열을 평문과 비트별 논리합(XOR)을 하여 암호문을 만든다. 암호문을 복호화하는 것도 같은 초기치를 사용하여 키 수열을 생성한 후, 암호문과 비트별 논리합을 취해주면 구할 수 있다. 이러한 스트림 암호의 공격은 일반적으로 이미 알고 있는 평문과 암호문의 쌍으로부터 비밀키(LFSR의 초기치)를 알아내려는 시도이다. 즉, 키 수열을 알고 있을때 초기치를 구하려는 시도라고 말할 수 있다.

최근의 LFSR 기반의 스트림 암호의 가장 위협적인 공격은 키 수열과 초기치 사이의 대수적 관계식을 얻어내어 이를 대수적으로 초기치를 계산하는 방식으로 이를 대수적 공격이라고 부른다. 초기의 대수적 공격은 AES와 같은 블록 암호나 HFE와 같은 공개키 암호의 분석에 사용되었다.[1,2] 스트림 암호로는 2002년 Toyocrypt에 성공적으로 적용되었고, LILI-128에 확장 적용됨에 따라서 주목을 받기 시작하였다.[3,~5]

LFSR에 메모리를 사용하는 방식은 대수적 관계식을 쉽게 찾을 수 없어 대수적 공격에 저항성이 있을 것으로 생각했으나, 이 경우에는 항상 대수적 관

접수일: 2003년 10월 9일; 채택일: 2003년 12월 22일
† 주저자, dlee@etri.re.kr
‡ 교신저자, dlee@etri.re.kr

계식이 존재하는 것이 증명되었다.[6,7]

[정리 1][6,7] n개의 LFSR과 l 비트의 메모리를 이용하는 combiner는 $\lceil \frac{n(l+1)}{2} \rceil$ 차 이하의 대수적 관계식이 존재한다.

메모리를 사용하는 대표적인 방식으로는 Rueppel에 의하여 제안된 summation generator를 들 수 있다.[8] Bluetooth의 보안에 사용된 스트림 암호인 E_0는 4개의 LFSR로 이루어진 summation generator를 메모리를 4개 가지도록 변형한 것인데, [정리 1]에 의하여 존재성이 증명된 10차의 관계식보다 실제로 4차의 대수적 관계식을 실제로 찾아냄으로써 더욱 공격량을 낮출 수 있었다.[6]

본 논문에서는 Rueppel에 의하여 최초 제안된 summation generator의 경우에도 [정리 1]의 관계식보다 더 낮은 차수의 대수적 관계식이 있음을 보이고, 일반적으로 2^k개의 LFSR을 사용할때, $k+1$개의 출력을 이용하여 2^k차 이하의 관계식을 얻을 수 있음을 보인다.

II. Summation Generator의 대수적 관계식

Summation generator는 주기가 매우 크고, 선형복잡도가 다른 generator에 비해 매우 우수하여 스트림 암호 설계에 많이 이용되었다. Summation generator는 각 출력비트를 더하고 발생된 올림수(carry)를 메모리에 저장하여 다음 출력에 영향을 미친다. n개의 LFSR의 t번째 시각에서 출력을 각각 x_j^t라고 할때, Summation generator의 출력 z^t는 다음과 같다.

$$z^t = \sum_{j=1}^{n} x_j^t + c^t \mod 2$$

$$c^{t+1} = \lfloor (\sum_{j=1}^{n} x_j^t + c^t)/2 \rfloor$$

이것을 그림 1으로 나타내면 다음과 같다.

여기서 c^t는 올림수이며, $\lceil \log_2 n \rceil$ 비트로 표현된다. 그러면 $c^t = (\alpha_k^t, \alpha_{k-1}^t, \cdots, \alpha_2^t, \alpha_1^t)$를 c^t의 이진 전개로 표기한다. 따라서 [정리 1]에 따르면 $\lceil \frac{n(\log n+1)}{2} \rceil$ 차의 대수적 관계식이 존재한다. 그러나 [정리 1]은 대수적 관계식의 존재성만을 말해줄 뿐이며 실제 관계식이 어떤 것인지는 알 수 없다.

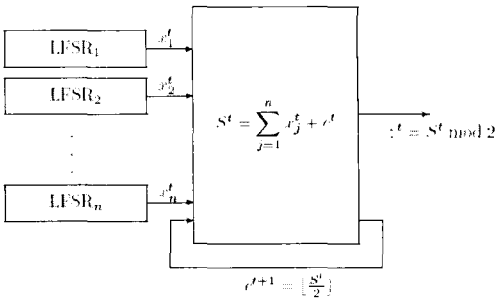

그림 1. Summation generator의 다이어그램

본 논문에서는 summation generator의 실제 대수적 관계식을 직접 찾아보고자 한다.

2.1 대칭 다항식과 행렬

다음과 같은 기호를 사용한다. S_i^t를 $x_1^t, \cdots, x_n^t$를 변수로 가지는 부울 함수로서, i차 기본 대칭 다항식이라고 하자. 편의상 $S_0^t = 1$로 정의한다.

$$S_1^t = \oplus_{j=1}^{n} x_j^t$$
$$S_2^t = \oplus_{1 \le j_1 < j_2 \le n} x_{j_1}^t x_{j_2}^t$$
$$\vdots$$
$$S_n^t = \prod_{j=1}^{n} x_j^t$$

그러면 $0 \le a, b \le n$에 대해서 $m_{a,b}$를 $\Sigma x_j^t = b$일때, S_a^t의 값이라고 정의한다. $m_{a,b}$를 다시 표현하면 n개의 x_j들의 변수 중에서 b개만 1일때, S_a^t를 이루는 단항식 중에서 선택된 b개를 제외한 항이 포함되어 있는 것은 모두 0이 되므로, 선택된 b개의 변수 중에서 a개를 선택하는 개수를 세어서 이를 (mod 2)로 보는 것이라고 볼 수 있다. 따라서 다음이 성립한다.

$$m_{a,b} = {}_bC_a \mod 2. \quad (1)$$

그러면 다음과 같은 $(n+1) \times (n+1)$ 행렬 M을 생각하자.

$$M = (m_{a,b})$$

$n=4$일때의 행렬 M을 나타내보면 다음과 같다.

a \ b	0	1	2	3	4
S_0^t	1	1	1	1	1
S_1^t	0	1	0	1	0
S_2^t	0	0	1	1	0
S_3^t	0	0	0	1	0
S_4^t	0	0	0	0	1

M'을 M에서 $(n+1)$ 번째 행과 $(n+1)$ 번째 열을 제외한 $n\times n$ 행렬이라고 하자. 그리고 n에 대한 M, M'을 표기할때는 각각 $M(n)$, $M'(n)$으로 표기한다.

〔보조 정리 2〕

$$M'(2^{k+1}) = \begin{pmatrix} M'(2^k) & M'(2^k) \\ 0 & M'(2^k) \end{pmatrix}$$

〔증명〕 $k=1$일때는 앞의 예로 구한 것에 의해 보였고, 이후 귀납법을 이용하여 증명한다. $M'(2^{k+1})$을 $\begin{pmatrix} I & II \\ III & IV \end{pmatrix}$로 나누어서 생각하자.

식(1)에 의하여 I번이 $M'(2^k)$이고 M'의 하 삼각 부분인 III번 역시 0임은 자명하다. 그러면 I, II, IV번이 모두 같음을 보이기 위해서 $0\le a, b<2^k$인 모든 a, b에 대해서 다음의 식을 증명하면 충분하다.

$${}_bC_a = {}_{2^k+b}C_a = {}_{2^k+b}C_{2^k+a} \bmod 2 \qquad (2)$$

이를 위해서 우선 다음의 관계식을 생각한다.

$$\begin{aligned}(1+x)^{2^k+b} &= (1+x)^{2^k}(1+x)^b \\ &= (1+x^{2^k})(1+x)^b \mod 2 \\ &= (1+x)^b + x^{2^k}(1+x)^b\end{aligned}$$

위의 왼쪽 관계식에서 x^a의 계수는 식 (2)의 가운데 항이다. 그런데 위의 오른쪽 관계식에서 x^a는 $a<2^k$이므로 첫 번째 항에서만 존재하고 계수는 식 (2)의 첫 번째 항과 같다. 같은 방법으로 x^{2^k+a}의 계수를 비교하면 식(2)에서 첫 번째 항과 마지막 항이 서로 같음을 보일 수 있다. 따라서 〔보조 정리 2〕가 성립한다.

2.2 n=4일 때의 관계식

$z^t = S_1^t \oplus \alpha_1^t$이 성립하는 것은 당연하다. 또한 행렬 M을 이용하여 α_i를 대칭 다항식을 이용하여 표현할 수 있다. 즉, 이전의 올림수 c^t의 경우에 따라서 α_i^{t+1}이 1이 되도록 하는 Σx_j의 값을 구하고, 이 값에서만 1이 되도록 하는 대칭 다항식을 행렬 M을 이용하여 구한다. 이전의 올림수 c^t는 0, 1, 2, 3의 경우가 있으며 각각은 α_1^t와 α_2^t로 나타낼 수 있다. 아래에서 ~는 왼쪽의 Σx_j의 값에서만 오른쪽의 대칭 다항식이 1이 되는 것을 의미한다.

1. $\alpha_1^t=0$, $\alpha_2^t=0$일때:

$$\begin{cases}\alpha_1^{t+1}=1 \leftrightarrow \Sigma x_j^t = 2,3 & \sim[S_2^t] \\ \alpha_2^{t+1}=1 \leftrightarrow \Sigma x_j^t = 4 & \sim[S_4^t]\end{cases}$$

2. $\alpha_1^t=1$, $\alpha_2^t=0$일때:

$$\begin{cases}\alpha_1^{t+1}=1 \leftrightarrow \Sigma x_j^t = 1,2 & \sim[S_1^t+S_2^t] \\ \alpha_2^{t+1}=1 \leftrightarrow \Sigma x_j^t = 3,4 & \sim[S_3^t+S_4^t]\end{cases}$$

3. $\alpha_1^t=0$, $\alpha_2^t=1$일때:

$$\begin{cases}\alpha_1^{t+1}=1 \leftrightarrow \Sigma x_j^t = 0,1,4 & \sim[1+S_2^t] \\ \alpha_2^{t+1}=1 \leftrightarrow \Sigma x_j^t = 2,3,4 & \sim[S_2^t+S_4^t]\end{cases}$$

4. $\alpha_1^t=1$, $\alpha_2^t=1$일때:

$$\begin{cases}\alpha_1^{t+1}=1 \leftrightarrow \Sigma x_j^t = 0,3,4 \\ \qquad\qquad \sim[1+S_1^t+S_2^t] \\ \alpha_2^{t+1}=1 \leftrightarrow \Sigma x_j^t = 1,2,3,4 \\ \qquad\qquad \sim[S_1^t+S_2^t+S_3^t+S_4^t]\end{cases}$$

위의 사실을 식으로 표현하면 다음과 같다.

$$\begin{aligned}\alpha_1^{t+1} &= (1\oplus\alpha_1^t)(1\oplus\alpha_2^t)S_2^t \\ &\oplus \alpha_1^t(1\oplus\alpha_2^t)(S_1^t\oplus S_2^t) \\ &\oplus (1\oplus\alpha_1^t)\alpha_2^t(1\oplus S_2^t) \\ &\oplus \alpha_1^t\alpha_2^t(1\oplus S_1^t\oplus S_2^t) \\ &= S_2^t\oplus\alpha_1^t S_1^t\oplus\alpha_2^t\end{aligned}$$

$$\begin{aligned}\alpha_2^{t+1} &= (1\oplus\alpha_1^t)(1\oplus\alpha_2^t)S_4^t \\ &\oplus \alpha_1^t(1\oplus\alpha_2^t)(S_3^t\oplus S_4^t) \\ &\oplus (1\oplus\alpha_1^t)\alpha_2^t(S_2^t\oplus S_4^t) \\ &\oplus \alpha_1^t\alpha_2^t(S_1^t\oplus S_2^t\oplus S_3^t\oplus S_4^t) \\ &= S_4^t\oplus\alpha_1^t S_3^t\oplus\alpha_2^t S_2^t\oplus\alpha_1^t\alpha_2^t S_1^t\end{aligned}$$

위에서 α_1은 x_j에 대한 1차식으로 표현되고, α_2는 α_1에 대한 2차식으로 표현된다. 그러므로 α_2는 x_j에 대한 2차식이고, 마지막 관계식에서 α_i를 소거하면 x_j에 대한 4차 이하의 대수적 관계식이 존재함을 알 수 있다.

2.3 $n=2^{k+1}$일 때의 관계식

$\alpha_i(2^k)$와 $\alpha_i(2^{k+1})$은 각각 $n=2^k$일 때와 $n=2^{k+1}$일 때의 α_i를 나타낸다. 그러면 다음의 보조 정리를 증명한다.

[보조 정리 3]

1. $i<k$에 대해서 $\alpha_i^{t+1}(2^{k+1})=\alpha_i^{t+1}(2^k)$.
2. $\alpha_k^{t+1}(2^{k+1})=\alpha_k^{t+1}(2^k)\oplus\alpha_{k+1}^t$.
3. $\alpha_k^{t+1}(2^k)$를 이루는 S_i들의 아래첨자를 2^k만큼 증가시킨 것을 $\beta_k^{t+1}(2^k)$라고 하자.
   $\alpha_{k+1}^{t+1}(2^{k+1})=\beta_k^{t+1}(2^k)\oplus\alpha_k^{t+1}(2^k)\alpha_{k+1}^t$.

[주의사항] $n=2^k$일 때와 $n=2^{k+1}$일 때의 S_i는 변수의 개수가 다르므로 실제로는 서로 다른 값이지만, 표기의 편의상 같은 것으로 사용한다.

[증명] c^{t+1}의 정의에 따라서 다음이 성립한다.

$$\alpha_i^{t+1}=1 \leftrightarrow (\Sigma x_j^t+c^t)\equiv 2^i,\cdots,2^{i+1}-1 \bmod 2^{i+1}$$

(1) $i<k$이므로 $\Sigma x_j^t=2^{k+1}$인 경우는 α_i에 영향을 주지 않는다. $0\le x<2^k$에 대해서, $\Sigma x_j^t=x$와 $\Sigma x_j^t=2^k+x$는 α_i에 같은 영향을 준다. 따라서 [보조 정리 2]에 의하여 $c^t<2^k$일 때 (즉, $\alpha_{k+1}^t=0$일 때), $\alpha_i^{t+1}(2^{k+1})$은 $\alpha_i^{t+1}(2^k)$와 같다.
한편 $c^t\ge 2^k$일 때 (즉, $\alpha_{k+1}^t=1$일 때), $c^t=2^k+c$라고 하자. 이 경우는 $c^t=c$인 경우와 α_i에 같은 영향을 주므로 $\alpha_i^{t+1}(2^{k+1})$은 $\alpha_i^{t+1}(2^k)$와 같다. 따라서 다음이 성립한다.

$$\alpha_i^{t+1}(2^{k+1})=(1\oplus\alpha_{k+1}^t)\alpha_i^{t+1}(2^k)\oplus\alpha_{k+1}^t\alpha_i^{t+1}(2^k)=\alpha_i^{t+1}(2^k)$$

(2) $0\le c^t<2^k$에 대해서 α_k^{t+1}에 영향을 미치기 위한 Σx_j^t의 값의 집합은 다음과 같다.

$$2^k-c^t,2^k+1-c^t,\cdots,2^{k+1}-1-c^t.$$

한편 $n=2^k$에서 $\alpha_k^{t+1}(2^k)$에 영향을 미치기 위한 Σx_j^t의 값의 집합은 다음과 같다.

$$2^k-c^t,2^k+1-c^t,\cdots,2^k.$$

이 값들에서 $\alpha_k^{t+1}(2^k)$가 1이 되도록 하는 S들의 합을 적당한 첨자 집합 $J\subset 0,1,\cdots,2^k$에 대하여 $\oplus_{j\in J}S_j^t$라고 하자. 그러면 $2^k-c^t>0$이므로 $0\notin J$이고, 2^k가 Σx_j^t의 집합에 포함되므로 $2^k\in J$이다.

이제 $n=2^k$에서 구한 $\oplus_{j\in J}S_j^t$를 $n=2^{k+1}$로 확장해보자. J'을 J에서 2^k을 뺀 첨자 집합이라고 하자. [보조 정리 2]에 의하여 $\oplus_{j\in J'}S_j^t$가 1이 되는 Σx_j^t의 값의 집합은 다음과 같다.

$$\begin{matrix}2^k-c^t,2^k+1-c^t,\cdots,2^k-1\\ \bigcup 2^{k+1}-c^t,2^{k+1}+1-c^t,\cdots,2^{k+1}-1\end{matrix}$$

$S_{2^t}^t$가 1이 되는 집합은 $2^k,2^k+1,\cdots,2^{k+1}-1$이므로 $\oplus_{j\in J}S_j^t$와 $\alpha_k^{t+1}(2^{k+1})$는 1이 되는 Σx_j^t의 집합이 같다. 따라서 $\oplus_{j\in J}S_j^t$와 $\alpha_k^{t+1}(2^{k+1})$는 서로 같다.

$2^k\le c^t<2^{k+1}$에 대해서 α_k^{t+1}에 영향을 미치기 위한 Σx_j^t의 값의 집합은 다음과 같다.

$0,1,\cdots,2^{k+1}-1-c^t\bigcup 2^{k+1}-(c^t-2^k),\cdots,2^{k+1}$

$c=c^t-2^k$라고 하면, 위의 집합은 다음 집합의 여집합이다.

$$2^k-c,2^k+1-c,\cdots,2^{k+1}-1-c$$

즉, 같은 Σx_j^t에 대해서 $\oplus_{j\in J}S_j^t$와 $\alpha_k^{t+1}(2^{k+1})$는 서로 다른 값을 가진다. 따라서 다음이 성립한다.

$$\alpha_k^{t+1}(2^{k+1})=(1\oplus\alpha_{k+1}^t)\alpha_k^{t+1}(2^k)\oplus\alpha_{k+1}^t(1\oplus\alpha_i^{t+1}(2^k))=\alpha_k^{t+1}(2^k)\oplus\alpha_{k+1}^t$$

(3) $i=k+1$이므로 α_k^{t+1}에 영향을 미치기 위한 Σx_j^t의 값의 집합은 다음과 같다.

$$2^{k+1}-c^t, 2^{k+1}+1-c^t, \cdots, 2^{k+1}$$

특히 $0 \le c^t < 2^k$일 때는 위의 집합을 다음과 같이 쓸 수 있다.

$$2^k+2^k-c^t, 2^k+1-c^t, \cdots, 2^k$$

그러므로 $\alpha_k^{t+1}(2^{k+1})$를 이루는 S_i들은 $\alpha_k^{t+1}(2^k)$를 이루는 S_i들의 첨자를 2^k만큼 증가시킨 것과 같다. $\alpha_k^{t+1}(2^k)$를 이루는 S_i들의 아래첨자를 2^k만큼 증가시킨 것을 $\beta_k^{t+1}(2^k)$이라고 하자.

$2^k \le c^t < 2^{k+1}$일 때는 $c = c^t - 2^k$라고 하면 다음과 같이 두개의 합집합으로 나누어 쓸 수 있다.

$$\begin{aligned} &2^{k+1}-c^t, 2^{k+1}+1-c^t, \cdots, 2^{k+1} \\ &= 2^k-c, \cdots, 2^{k+1}-1-c \cup 2^{k+1}-c, \cdots, 2^{k+1} \end{aligned}$$

첫번째 집합은 (2)번의 집합과 동일하고, 두번째 집합은 앞서 구한 $c^t < 2^k$인 경우와 동일하다. 따라서 다음이 성립한다.

$$\begin{aligned} &\alpha_{k+1}^{t+1}(2^{k+1}) = (1 \oplus \alpha_{k+1}^t)\beta_k^{t+1}(2^k) \\ &\oplus \alpha_{k+1}^t(\alpha_k^{t+1}(2^k) \oplus \beta_k^{t+1}(2^k)) \\ &= \beta_k^{t+1}(2^k) \oplus \alpha_{k+1}^t \alpha_k^{t+1}(2^k) \end{aligned}$$

〔정리 4〕 $n=2^{k+1}$일 때, α_i는 x_j에 대한 2^{i-1}차의 식으로 표현되고, 이것을 각각 α_{k+1}^{t+1}에 대입하여 α_i를 모두 소거하면 x_j에 대한 2^{k+1}차 이하의 대수적 관계식을 얻을 수 있다.

〔증명〕 모든 $n=2^k$에 대해서 $z^t = S_1^t \oplus \alpha_1^t$이 성립하는 것은 당연하므로 α_1은 x_j의 1차식이다. 그리고 $n=2^2$일때, 2.2절의 예에서 다음을 보였다.

$$\begin{aligned} \alpha_1^{t+1} &= S_2^t \oplus \alpha_1^t S_1^t \oplus \alpha_2^t \\ \alpha_2^{t+1} &= S_4^t \oplus \alpha_1^t S_3^t \oplus \alpha_2^t S_2^t \oplus \alpha_1^t \alpha_2^t S_1^t \end{aligned}$$

즉, α_2는 x_j의 2차식이며, 마지막 관계식에서 α_i를 모두 소거하면 x_j에 대한 4차식의 관계식을 얻는다.

$n=2^3$이면, 〔보조 정리 3〕의 (1)에 의해 α_1^{t+1}은 $n=2^2$의 경우와 동일하므로 α_2는 그대로 2차식이다. 그리고 〔보조 정리 3〕의 (2)와 (3)에 의해 다음이 성립한다.

$$\begin{aligned} \alpha_2^{t+1} &= S_4^t \oplus \alpha_1^t S_3^t \oplus \alpha_2^t S_2^t \oplus \alpha_1^t \alpha_2^t S_1^t \oplus \alpha_3^t \\ \alpha_3^{t+1} &= S_8^t \oplus \alpha_1^t S_7^t \oplus \alpha_2^t S_6^t \oplus \alpha_1^t \alpha_2^t S_5^t \\ &\oplus \alpha_3^t (S_4^t \oplus \alpha_1^t S_3^t \oplus \alpha_2^t S_2^t \oplus \alpha_1^t \alpha_2^t S_1^t) \end{aligned}$$

마지막 관계식을 보면, $n=2^2$인 경우에 비해 차수가 2^2만큼 증가하므로 8차식의 관계식을 얻을 수 있으므로 정리가 성립한다.

같은 방법으로 일반적인 $n=2^{k+1}$에 대한 경우에도 α_{k+1}이 x_j에 대한 2^k차식임을 보일 수 있고, 마지막 관계식에서 $n=2^k$의 경우보다 차수가 2^k만큼 증가하므로 2^{k+1}차의 관계식을 얻을 수 있다.

〔따름 정리 5〕 n개의 LFSR로 이루어진 Summation generator는 연속된 $\lceil \log_2 n \rceil + 1$개의 출력값을 이용하여 초기치에 관한 $2^{\lceil \log_2 n \rceil}$차 이하의 대수적 관계식을 만들 수 있다.

〔증명〕 $2^{\lceil \log_2 n \rceil}$개로 이루어진 Summation generator에서 n개를 제외한 나머지 LFSR의 출력을 모두 0으로 가정하면, n개의 LFSR로 이루어진 Summation generator와 동일하고, 〔정리 4〕에 의하여 $2^{\lceil \log_2 n \rceil}$차 이하의 관계식을 만들 수 있다.

〔참고〕 본 논문의 결과를 확장하여 FSE 2004에 발표하였다.[9] $n=2^k$인 경우 〔정리 4〕에서 얻은 관계식이 〔10〕에서 정의한 double-decker 방정식을 이룬다는 사실을 증명하여 공격복잡도를 더욱 낮출 수 있음을 보였다.

Ⅲ. 결 론

본 장에서는 Summation generator에 대하여 이론적인 차수보다 더 낮은 차수를 가지는 실제 대수적 관계식을 만들 수 있었다. 표 1은 기존의 결과와 위의 〔정리 4〕, 〔따름 정리 5〕의 결과와 실제 계산 결과를 나타낸 표이다.

아래의 표에서 볼 때, 연속된 $\lceil \log_2 n \rceil + 1$개의 출력값을 이용하여 만들 수 있는 관계식의 차수는 본 논문에서 제시는 결과가 매우 상한과 근접함을 알 수 있으며, 특히 $n=2^k$인 경우 상한과 동일함을 알 수 있다.

표 1을 이용하면 기존의 방법보다 summation ge-

표 1. Summation generator의 대수적 차수

n	2	3	4	5	6	7	8
[6,7]	2	5	6	10	12	14	16
[정리 4,5]	2	4	4	8	8	8	8
직접 계산	2	3	4	6	6	7	8

표 2. Summation generator의 공격 복잡도

	Divide & Conquer	Courtois[7]	[정리 4]
128 비트	2^{96}	$2^{90.8}$	$2^{65.5}$
256 비트	2^{192}	$2^{107.9}$	$2^{76.9}$

nerator를 공격하는 복잡도가 훨씬 더 낮아진다. 예를 들면 128 비트 또는 256 비트의 초기치를 사용하면서 4개의 입력 LFSR을 사용한다고 가정하면, 공격 복잡도가 표 2와 같다.

참 고 문 헌

[1] N. Courtois and J. Pieprzyk, "Cryptanalysis of block ciphers with overdefined systems of equations," *Asiacrypt 2002*, LNCS 2501, Springer-Verlag, pp. 267-287, 2002.

[2] N. Courtois, "The security of Hidden Field Equations (HFE)," *CT-RSA 2001*, LNCS 2020, Springer-Verlag, pp. 266-281, 2001.

[3] N. Courtois, "Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt," *ICISC 2002*, LNCS 2587, Springer-Verlag, pp. 182-199, 2002.

[4] N. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback," *Advances in Cryptology - Eurocrypt 2003*, LNCS 2656, Springer-Verlag, pp. 345-359, 2003.

[5] 문덕재, 홍석희, 이상진, 임종인, 은희천, "과포화(Overdefined) 연립방정식을 이용한 LILI-128 스트림 암호에 대한 분석," 정보보호학회 논문지, 13(1), pp. 139-146, 2003.

[6] F. Armknecht and M. Krause, "Algebraic attacks on combiners with memory," *Advances in Cryptology - Crypto 2003*, LNCS 2729, Springer-Verlag, pp. 162-175, 2003.

[7] N. Courtois, "Algebraic attacks on combiners with memory and several outputs," *E-print archive*, 2003/125.

[8] R. A. Rueppel, "Correlation immunity and the summation generator," *Advances in Cryptology-Crypto'85*, LNCS 219, Springer-Verlag, pp. 260-272, 1985.

[9] D. H. Lee, J. Kim, J. Hong, J.W. Han, and D. Moon, "Algebraic attacks on summation generators," *Preproceeding of Fast Software Encryption*, pp.15-29, 2004.

[10] N Courtois, "Fast algebraic attack on stream ciphers with linear feedback," *Advances in Cryptology - Crypto 2003*, LNCS 2729, Springer-Verlag, pp. 176-194, 2003.

〈著者紹介〉

이 동 훈 (Dong Hoon Lee) 정회원
1994년 2월: 서울대학교 수학교육과 학사
1996년 2월: 한국과학기술원 수학과 석사
2000년 2월: 한국과학기술원 수학과 박사
2000년 2월~ 2002년 3월 : (주)퓨쳐시스템 선임 연구원
2002년 4월 ~ 현재 : 국가보안기술연구소 선임연구원
〈관심분야〉 응용 정수론, 암호론, 인터넷 보안

김 재 헌 (Jaeheon Kim) 정회원
1991년 2월: 서울대학교 수학과 학사
1993년 2월: 서울대학교 수학과 석사
2000년 8월: 서울대학교 수학과 박사
2000년 4월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 응용 정수론, 암호론

한 재 우 (Jaewoo Han) 정회원
1991년 2월: 서강대학교 수학과 졸업
1993년 2월: 한국과학기술원 수학과 석사
1999년 8월: 한국과학기술원 수학과 박사
1999년 7월~2000년 1월: 한국전자통신연구원 선임연구원
2000년 1월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 암호 프로토콜, 스트림 암호, 매듭이론

홍 진 (Jin Hong) 정회원
1994년 2월: 서울대학교 수학과 학사
1996년 2월: 서울대학교 수학과 석사
2000년 8월: 서울대학교 수학과 박사
2000년 9월~2002년 9월: 고등 과학원 연구원
2002년 9월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 암호론

문 덕 재 (Dukjae Moon) 정회원
2000년 2월: 서울시립대학교 수학과 학사
2003년 2월: 고려대학교 정보보호대학원 석사
2003년 3월~현재: 국가보안기술연구소 연구원
〈관심분야〉 암호학, 알고리즘 분석