情報保護學會論文誌
第16卷 第1號, 2006. 2

메모리를 가지는 Combiner 모델에 대한 새로운 대수적 방정식 구성 방법과 그 응용

김 재 헌,†‡ 한 재 우, 문 덕 재
국가보안기술연구소

Finding New Algebraic Relations on Some Combiners with Memory And Its Applications

Jaeheon Kim,†‡ Jae Woo Han, Dukjae Moon
National Security Research Institute

요 약

Summation generator와 같이 메모리를 가지는 combiner 모델에 대해 대수적 공격이 적용 가능함은 잘 알려져 있다.[1,8] 메모리를 가지는 combiner 모델에 대하여 대수적 공격을 적용하기 위해서는 대수적 방정식 수립이 필요한데, 현재까지의 모든 결과는 2비트 이상의 연속적인 출력 키수열을 필요로 하였다[1,4,8]. 본 논문에서는 summation generator에 대한 대수적 방정식을 1비트 키수열만으로 구성할 수 있음을 보인다. 또한 ISG 알고리즘 [9]에 대해서도 1비트 키수열만을 이용한 방정식 구성이 가능함을 보인다. 이를 이용하여, summation generator 및 ISG 여러 개를 하나의 부울함수로 결합한 형태의 키수열발생기에 대해서도 대수적 공격이 가능함을 보인다.

ABSTRACT

It is known that we can apply algebraic attacks on combiners with memory such as summation generators. [1,8] To apply algebraic attacks on combiners with memory, we need to construct algebraic relations between the keystream bits and the initial bits of the LFSRs. Until now, all known methods produce algebraic relations involving several consecutive bits of keystream. [1,4,8] In this paper, we show that algebraic relations involving only one keystream bit can be constructed for summation generators. We also show that there is an algebraic relation involving only one keystream bit for ISG [9] proposed by Lee and Moon. Using this fact, we analyze the keystream generators which generate the keystreams by combining summation generators.

Keywords : *Algebraic Attack, Algebraic Relation, One Keystream Bit, Summation Generator, ISG*

Ⅰ. 서 론

대수적 공격은 스트림 암호 분석에 지대한 영향을 미쳤다. 대수적 공격이란 스트림 암호의 초기값(키)과 출력 키수열 사이의 대수적 방정식을 구성하여, 이를 선형화 기법을 이용하여 그 해(키)를 구하는 방법이다.

스트림 암호에 대한 대수적 공격은 최초에 Toyocrypt에 대하여 이루어졌고, [3] 그 후에 LILI-128 등과 같이 메모리를 가지지 않는 모든 키수열발생기로 그 적용이 확대되었다. [5] 초기에는 메모리를 사

접수일 : 2005년 10월 5일 ; 채택일 : 2005년 12월 6일
† 주저자, ‡ 교신저자 : jaeheon@etri.re.kr

용하는 스트림 암호는 대수적 공격에 좀 더 강할 것으로 보였으나, 스트림 암호에 대한 대수적 공격이 발전함에 따라 summation generator나 Bluetooth에 사용되는 E2 알고리즘 등의 메모리를 가지는 combiner 모델에 대하여도 대수적 공격이 적용된다는 사실이 밝혀졌다. [1] 메모리를 가지는 combiner 모델에 대수적 공격을 적용하기 위해서는 대수적 방정식을 수립해야 하는데, 이때 필요한 키수열은 연속적이어야 한다고 알려져 왔다. [1,2,7,8] 예를 들어, k개의 LFSR을 입력으로 하고 l개의 메모리 비트를 가지는 (k,l)-combiner의 경우, 연속된 $(l+1)$개의 키수열이 있으면 $\lceil k(l+1)/2 \rceil$ 차 이하의 대수적 방정식 구성이 가능하다. [2]

그러나, 본 논문에서는 summation generator에 대한 대수적 방정식을 연속된 여러 개의 키수열을 이용하지 않고 단 1비트 키수열만을 이용하여 생성할 수 있음을 보인다. 또한, ISG(Improved Summation Generator with 2-bit Memory) [9] 알고리즘에 대해서도 1비트 키수열만을 이용한 대수적 방정식을 제시한다.

일반적으로 연속된 키수열을 이용하여 대수적 방정식을 구성할 수 있는 키수열발생기 여러 개를 부울함수 하나로 결합한 형태의 키수열발생기의 경우에는 이에 대한 대수적 방정식 구성이 어려워서 대수적 공격에 어려움이 있다. 본 논문에서는 위의 결과를 이용하여 summation generator 및 ISG 여러 개를 하나의 부울함수로 결합한 형태의 일반적인 키수열발생기에 대해서도 대수적 공격이 가능함을 보인다.

II. Summation Generator에 대한 새로운 대수적 방정식 구성 방법

2.1 Summation Generator 개요

Summation generator [11]는 주기가 매우 크고, 선형복잡도가 다른 generator에 비해 매우 우수하여 스트림 암호 설계에 많이 이용되었다. Summation generator는 각 출력 비트를 더하고 발생된 carry를 memory에 저장하여 다음 출력 계산에 이용한다. 여기서 c^t 를 carry라 하면, $k=\lceil \log_2 n \rceil$ 비트로 표현 가능하다. 이제, $c^t=(c^t_{k-1},\cdots,c^t_0)$ 를 c^t의 이진 전개로 나타낸다.

n개의 LFSR의 t번째 clock에서 출력을 각각 x^t_j 라고 할 때, summation generator의 출력 z^t 는 다음과 같다.

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

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

σ^t_i 를 $x^t_1,\cdots,x^t_n$를 변수로 가지는 부울 함수로서, i차 elementary symmetric polynomial 이라고 하자.

$$\sigma^t_0=1,\ \sigma^t_1=\oplus^n_{j=1}x^t_j,\ \ \sigma^t_2=\oplus_{1\le j_1<j_2\le n}x^t_{j_1}x^t_{j_2},$$
$$\vdots$$
$$\sigma^t_n=\Pi^n_{j=1}x^t_j,\ \sigma^t_i=0\ (i>n).$$

2.2 Summation Generator에 대한 기존의 대수적 공격

$n=2^k$ 개를 기반으로 하는 summation generator에 대하여, Lee 등은 $k+1$ 개의 출력 키수열을 알면 차수가 2^k 인 대수적 방정식을 구성할 수 있음을 보였다. [8] 대수적 방정식을 구성한 후의 대수적 공격 방법과 그 계산 복잡도는 참고문헌 [8]을 참고하기 바란다. 후에, Braeken과 Semaev는 덧셈의 ANF(Algebraic Normal Form)를 이용하여 그 증명을 간단히 하였다. [2] 참고문헌 [2]에 따르면, 다음번의 carry 비트를 현재 LFSR의 출력 비트와 carry 비트를 이용하여 다음과 같이 표현할 수 있다.

$$c_0^{t+1}=\sigma_2^t\oplus c_0^t\sigma_1^t\oplus c_1^t, \tag{3}$$
$$c_1^{t+1}=\sigma_4^t\oplus c_0^t\sigma_3^t\oplus c_1^t\sigma_2^t\oplus c_0^tc_1^t\sigma_1^t\oplus c_2^t, \tag{4}$$
$$c_2^{t+1}=\sigma_8^t\oplus c_0^t\sigma_7^t\oplus c_1^t\sigma_6^t\oplus c_0^tc_1^t\sigma_5^t\oplus c_2^t\sigma_4^t \oplus c_0^tc_2^t\sigma_3^t\oplus c_1^tc_2^t\sigma_2^t\oplus c_0^tc_1^tc_2^t\sigma_1^t\oplus c_3^t, \tag{5}$$
$$\vdots$$
$$c_k^{t+1}=\sigma_{2^k}^t\oplus c_0^t\sigma_{2^{k-1}}^t\oplus c_0^t\cdots c_{k-1}^t\sigma_1^t. \tag{6}$$

2.3 Summation Generator에 대한 새로운 대수적 방정식 구성 방법

본 소절에서는 일반적인 summation generator에 대해서 출력 키수열 1비트와 LFSR의 초

기값 사이의 대수적 방정식을 구성할 수 있음을 보인다.

〔정리 1〕 n개의 LFSR을 기반으로 하는 summation generator에 대해서, 출력 키수열 1비트와 입력 LFSR의 초기값 사이에 대수적 방정식이 존재한다.

(증명) n이 2의 멱승이 아닌 경우, n개의 LFSR을 입력으로 하는 summation generator는 $2^{\lceil \log_2 n \rceil}$개의 LFSR을 입력으로 하면서 $(2^{\lceil \log_2 n \rceil}-n)$개의 LFSR은 0으로 세팅한 summation generator로 볼 수 있다. 따라서, n이 2의 멱승, 다시 말해서 $2^k(k>0)$인 경우에 대해서만 증명하면 된다.

식 (1)에서, $(t+k)$ clock의 키수열 z^{t+k} 는 다음과 같이 표현 가능하다.

$$z^{t+k}=\sigma_1^{t+k}\oplus c_0^{t+k} \tag{7}$$

$(t+k-1)$ clock에서의 식 (3)을 (7)에 대입하면,

$$z^{t+k}=\sigma_1^{t+k}\oplus\sigma_2^{t+k-1}\oplus c_0^{t+k-1}\sigma_1^{t+k-1}\oplus c_1^{t+k-1}. \tag{8}$$

$(t+k-2)$ clock에서의 식 (4)를 (8)에 대입하면,

$$\begin{aligned} z^{t+k} = {} & \sigma_1^{t+k}\oplus\sigma_2^{t+k-1}\oplus c_0^{t+k-1}\sigma_1^{t+k-1} \\ & \oplus\sigma_4^{t+k-2}\oplus c_0^{t+k-2}\sigma_3^{t+k-2} \\ & \oplus c_1^{t+k-2}\sigma_2^{t+k-2} \\ & \oplus c_0^{t+k-2}c_1^{t+k-2}\sigma_1^{t+k-2}\oplus c_2^{t+k-2}. \end{aligned} \tag{9}$$

이러한 과정을 k번 반복하면, 마지막 단계에서 다음과 같은 식을 얻게 된다.

$$z^{t+k}=f\oplus\bigoplus c_{i_1}^{j_1}c_{i_2}^{j_2}\cdots c_{i_l}^{j_l}g_{i_1i_2\cdots i_l}^{j_1j_2\cdots j_l}. \tag{10}$$

여기에서 f와 $g_{i_1i_2\cdots i_l}^{j_1j_2\cdots j_l}$는 σ_i에 대한 함수인데, 계산해보면 상수함수가 되지 않음을 알 수 있다.

식 (10)의 양변에 모든 i_u와 j_v에 대하여 $(g_{i_1i_2\cdots i_l}^{j_1j_2\cdots j_l}\oplus 1)$를 곱하면, 다음 방정식을 얻는다.

$$z^{t+k}\prod(g_{i_1i_2\cdots i_l}^{j_1j_2\cdots j_l}\oplus 1)=f\prod(g_{i_1i_2\cdots i_l}^{j_1j_2\cdots j_l}\oplus 1). \tag{11}$$

따라서, 우리는 summation generator에 대하여 출력 키수열 1비트 z^{t+k}만을 이용한 대수적 방정식을 구할 수 있게 된다. □

식 (11)의 차수를 k가 1, 2, 3, 4인 경우에 대하여 각각 구해보면, 2, 6, 15, 30이 된다. 연속된 2, 3, 4, 5비트의 키수열을 이용한 〔8〕의 결과인 2, 4, 8, 16과 각각 비교해볼 때, $k=1$인 경우에만 차수가 같고 나머지 경우에는 차수가 좀 더 커짐을 알 수 있다. 대수적 공격 복잡도는 구성된 방정식의 차수에 비례해서 커지므로, summation generator에 대한 새로운 대수적 공격이 기존의 대수적 공격보다 효율성은 떨어진다. 하지만 연속된 키수열을 얻지 못하는 상황에서, 기존의 대수적 공격은 적용이 불가능하지만 본 논문에서 제안한 방법은 적용이 가능하다.

2.4 4개의 LFSR을 기반으로 하는 Summation Generator에 대한 추가 분석

본 소절에서는 4개의 LFSR을 기반으로 하는 summation generator에 대한 대수적 분석 중 흥미로운 몇 가지 결과에 대하여 기술한다.

〔정리 2〕 4개의 LFSR을 기반으로 하는 summation generator에 대해서, 연속된 2비트의 키수열만을 이용하여 4차 방정식 수립이 가능하다. 또한, 1비트 키수열만을 이용하여 5차 방정식을 세울 수 있다.

(증명) Clock $(t+2)$에서의 키수열 z^{t+2}는 다음과 같이 표현 가능하다.

$$z^{t+2}=\sigma_1^{t+2}\oplus c_0^{t+2}. \tag{14}$$

식 (14)의 오른 변을 식 (1)과 (2)를 이용하여 계산하면, 다음과 같다.

$$\begin{aligned} z^{t+2} &= \sigma_1^{t+2}\oplus\sigma_2^{t+1}\oplus c_0^{t+1}\sigma_1^{t+1}\oplus c_1^{t+1} \\ &= \sigma_1^{t+2}\oplus\sigma_2^{t+1}\oplus c_0^{t+1}\sigma_1^{t+1}\oplus\sigma_4^t\oplus c_0^t\sigma_3^t \\ &\quad \oplus c_1^t\sigma_2^t\oplus c_0^tc_1^t\sigma_1^t \\ &= \sigma_1^{t+2}\oplus\sigma_2^{t+1}\oplus\sigma_4^t\oplus c_0^{t+1}\sigma_1^{t+1}\oplus c_0^t\sigma_3^t \\ &\quad \oplus(c_0^{t+1}\oplus\sigma_2^t\oplus c_0^t\sigma_1^t)\sigma_2^t \\ &\quad \oplus c_0^t(c_0^{t+1}\oplus\sigma_2^t\oplus c_0^t\sigma_1^t)\sigma_1^t \\ &= \sigma_1^{t+2}\oplus\sigma_2^{t+1}\oplus\sigma_2^t\oplus\sigma_4^t \\ &\quad \oplus c_0^{t+1}(\sigma_1 \end{aligned} \tag{15}$$

식 (15)의 양 변에 $(\sigma_1^t\oplus 1)$을 곱한 후에 $\sigma_1^t\sigma_2^t=$

$\sigma_1^t\sigma_3^t=\sigma_2^t\sigma_3^t=\sigma_3^t, \sigma_1^t\sigma_4^t=\sigma_2^t\sigma_4^t=\sigma_3^t\sigma_4^t=0$을이용하여 간단히 하면 우리는 다음과 같이 연속된 2비트 z^{t+1}, z^{t+2} 만이 포함된 4차 방정식을 얻게 된다.

$$z^{t+2}(\sigma_1^t\oplus 1)=\sigma_1^t\sigma_1^{t+2}\oplus\sigma_1^t\sigma_2^{t+1}\oplus\sigma_3^t \oplus\sigma_1^{t+2}\oplus\sigma_2^{t+1}\oplus\sigma_2^t\oplus\sigma_4^t \oplus(z^{t+1}\oplus\sigma_1^{t+1})(\sigma_1^{t+1}\oplus\sigma_2^t)(\sigma_1^t\oplus 1). \quad (16)$$

식 (15)의 양 변에 $(\sigma_1^t\oplus1)(\sigma_1^{t+1}\oplus\sigma_2^t\oplus1)$을 곱하면, 또한 다음 식을 얻게 된다.

$$z^{t+2}(\sigma_1^t\oplus1)(\sigma_2^t\oplus\sigma_1^{t+1}\oplus1)= (\sigma_1^{t+2}\oplus\sigma_2^{t+1}\oplus\sigma_2^t\oplus\sigma_4^t)(\sigma_1^t\oplus1)(\sigma_2^t\oplus\sigma_1^{t+1}\oplus1). \quad (17)$$

식 (17)의 최고차 항이 $\sigma_2^{t+1}\sigma_1^t\sigma_2^t\oplus\sigma_4^t\sigma_1^{t+1}$ 이므로, (17)의 차수는 5가 된다. 따라서, 우리는 키수열 1비트만이 포함된 5차 방정식을 구성하였다. □

정리 2의 결과는 4개의 LFSR을 기반으로 하는 summation generator에 대하여, 참고문헌 [8]에서 연속된 3비트를 이용하여 4차 방정식을 구성한데 반하여, 연속된 2비트만을 가지고도 동차의 방정식 구성이 가능함을 보여준다. 또한, 앞에서 1비트 키수열만을 이용할 때는, 6차 방정식 구성이 가능하였는데, 정리 2는 이를 5차까지 떨어뜨릴 수 있음을 보여준다.

III. Improved Summation Generator (ISG)에 대한 새로운 대수적 방정식 구성 방법

3.1 ISG 개요

일반적으로 summation generator는 주기가 매우 크고 선형복잡도가 다른 논리들에 비하여 매우 우수하기 때문에 스트림 암호 설계에 많이 이용된다. 그러나, 두 개의 LFSR로 이루어진 summation generator의 경우에는 correlation 공격에 취약함이 잘 알려져 있다.[6,11] 이를 보완하기 위하여 이훈재 등은 참고문헌 [9]에서 기존의 summation generator에 메모리 1비트를 추가하여 correlation 공격에 강한 새로운 논리 ISG(Improved Summation Generator with 2-bit Memory)를 제안하였다.

j번째 clock에서 ISG의 동작 방식은 다음과 같

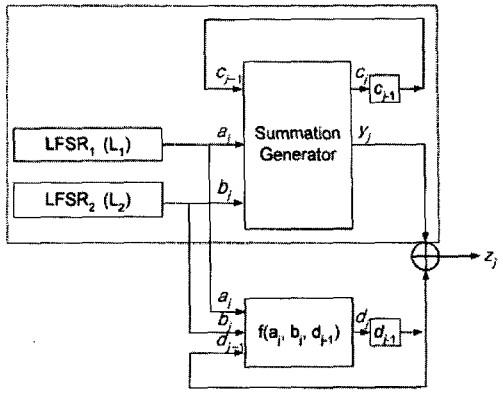

그림 1. ISG의 동작 방식

고, 이를 그림으로 표현하면 그림 1과 같다.

$$z_j=a_j\oplus b_j\oplus c_{j-1}\oplus d_{j-1} \quad (18)$$

$$c_j=a_jb_j\oplus(a_j\oplus b_j)c_{j-1} \quad (19)$$

$$d_j=b_j\oplus(a_j\oplus b_j)d_{j-1} \quad (20)$$

3.2 ISG에 대한 새로운 대수적 방정식 구성 방법

참고문헌 [7]에서는 ISG의 대수적 구조가 2개의 LFSR을 입력으로 하는 summation generator와 크게 다르지 않음을 이용하여 연속된 2비트의 키수열을 알면 ISG에 대한 2차의 방정식을 구성할 수 있음을 보였다. 본 소절에서는 ISG에 대하여 2차의 방정식을 구성할 때 연속된 2비트의 키수열이 아닌 단순한 1비트 키수열만을 알아도 충분함을 보인다.

[정리 3] ISG에 대하여 출력 키수열 1비트와 입력 LFSR의 초기값 사이의 대수적 방정식을 구성할 수 있다.

(증명) 먼저 식 (19) 와 (20) 으로부터 식을 얻고, 또한 다음과 같은 식을 얻을 수 있다.

$$c_j\oplus d_j=a_jb_j\oplus b_j\oplus(a_j\oplus b_j)(c_{j-1}\oplus d_{j-1}) \quad (21)$$

$$c_{j-1}\oplus d_{j-1}=z_j\oplus a_j\oplus b_j \quad (22)$$

식 (18)를 $(j+1)$번째 clock에 적용한 관계식에 식 (21)과 (22)를 대입하면, $z_{j+1}=a_{j+1}\oplus b_{j+1}\oplus c_j\oplus d_j=a_{j+1}\oplus b_{j+1}\oplus a_jb_j\oplus b_j\oplus(a_j\oplus b_j)(c_{j-1}\oplus d_{j-1})=a_{j+1}\oplus b_{j+1}\oplus a_jb_j\oplus b_j\oplus(a_j\oplus b_j)(z_j\oplus1)$ 임을 알 수 있다.

즉, 연속된 2비트의 키수열 z_j, z_{j+1} 을 알면 다음과 같은 2차의 관계식이 성립함을 알 수 있다[7].

$$z_{j+1} = a_{j+1} \oplus b_{j+1} \oplus a_j b_j \oplus b_j \oplus (a_j \oplus b_j)(z_j \oplus 1) \quad (23)$$

이제 식 (23)의 양변에 $(a_j \oplus b_j \oplus 1)$을 곱한 후에, 정리하면 $z_{j+1}(a_j \oplus b_j \oplus 1) = (a_{j+1} \oplus b_{j+1})(a_j \oplus b_j \oplus 1)$라는 2차 방정식을 얻을 수 있다. 이 방정식은 참고문헌 [7]의 결과와 달리 키수열 1비트 z_{j+1}만을 이용한 방정식이다. □

Ⅳ. 새로운 대수적 방정식 구성 방법의 응용

일반적으로 연속된 키수열을 이용하여 대수적 방정식을 구성할 수 있는 키수열발생기 여러 개를 부울함수 하나로 결합한 형태의 키수열발생기의 경우에는 이에 대한 대수적 방정식 구성이 어려워서 대수적 공격에 어려움이 있다. 본 절에서는 1비트 키수열만을 이용하여 대수적 방정식을 구성할 수 있는 키수열발생기 몇 개를 다시 부울함수를 이용하여 결합하는 형태의 키수열발생기에 대해서는 대수적 공격이 가능함을 보인다.

[정리 4] Summation generator 여러 개를 부울함수로 결합한 형태의 일반적인 키수열발생기에 대해서도 대수적 방정식 구성이 가능하다.

(증명) 이제 $X_i (1 \le i \le m)$를 LFSR 2개를 입력으로 하는 summation generator라 하고, 시간 $(t+1)$에서의 X_i의 출력과 최종 키수열을 각각 $z^{t+1}(X_i)$과 z^{t+1}로 나타내자. 또한, 부울함수는 f로 표현하면, 우리는 다음과 같은 식을 얻게 된다.

$$z^{t+1} = f(z^{t+1}(X_1), z^{t+1}(X_2), \cdots, z^{t+1}(X_m)). \quad (24)$$

X_i가 2개의 LFSR을 기반으로 하는 summation generator이므로 다음 식이 성립한다.

$$z^{t+1}(X_i)(\sigma_1^t(X_i) \oplus 1) = \sigma_2^t(X_i) \oplus (\sigma_1^t(X_i) \oplus 1)\sigma_1^{t+1}(X_i). \quad (25)$$

$1 \le i \le m$인 모든 i에 대하여, 식 (24)의 양변에 $(\sigma_1^t(X_i) \oplus 1)$를 곱한 후에 식 (25)을 이용하여 정리하면 키수열 1비트만이 관여하는 $(m+d)$차 대수적 방정식을 얻게 된다.

X_i가 임의의 n개의 LFSR을 기반으로 하는 summation generator의 경우에도, 위와 같은 방법을 이용하여 출력 키수열 1비트만을 이용한 대수적 방정식을 세울 수 있다. □

마찬가지로 ISG 여러 개를 하나의 부울함수로 결합한 형태의 키수열발생기에 대해서도 정리 4의 방법을 그대로 적용하면 대수적 방정식 구성이 가능함을 알 수 있다.

Ⅴ. 결 론

본 논문에서는 메모리를 이용하는 몇 가지 키수열발생기에 대해서 최초로 1비트 키수열만을 이용하여 대수적 방정식을 구성할 수 있음을 보였다. Summation generator나 ISG(Improved Summation Generator with 2-bit Memory)[9]가 그 예이다. 또한, 이를 이용하여 summation generator 및 ISG 여러 개를 하나의 부울함수로 결합한 형태의 일반적인 키수열발생기에 대해서 대수적 공격이 가능함을 보였다.

한편, 4개의 LFSR을 기반으로 하는 summation generator의 경우에 참고문헌 [8]에서는 연속된 3비트가 있어야만 4차의 방정식을 구성할 수 있었는데, 본 논문에서는 이보다 더 적은 2비트의 연속된 키수열을 이용하여 4차의 대수적 방정식을 구성할 수 있음을 보였다.

참 고 문 헌

[1] F. Armknecht, M. Krause, "Algebraic Attacks on Combiners with Memory", Crypto 2003 , pp. 162-175, 2003.

[2] A. Braeken, I. Semaev, "The ANF of the Composition of × and + mod 2^n with a Boolean Functions", FSE 2005, pp. 115-127, 2005.

[3] N. Courtois, "Higher order correlation attacks, XL algorithm and Cryptanalysis of Toyocrypt", ICISC 2002, pp. 182--19, 2002.

[4] N. Courtois, "Algebraic Attacks on Combiners with Memory and Several Outputs", ICISC 2004, pp. 3-20, 2004.

[5] N. Courtois, W. Meier, "Algebraic

attacks on stream ciphers with linear feedback", Eurocrypt 2003, pp. 345-359, 2003.

[6] E. Dawson, "Cryptanalysis of summation generator", Auscrypt 1992, pp. 209--215, 1992.

[7] Daewan Han, Moonsik Lee, "An algebraic attack on the improved summation generator with 2-bit memory", Information Proceeding Letters 2005, pp. 43-46, 2005.

[8] Donghoon Lee, Jaeheon Kim, Jin Hong, Jaewoo Han and Dukjae Moon, "Algebraic Attacks on Summation Generators", FSE 2004, pp. 34-48, 2004.

[9] Hoon Jae Lee, Sang Jae Moon, "On an improved summation generator with 2-bit memory", Signal Processing 80, pp. 211-217, Elsevier. 2000.

[10] Willi Meier, Othmar Staffelbach, "Correlation Properties of Combiners with Memory in Stream Cipher", Journal of Cryptology, vol.5, pp. 67-86, 1992.

[11] R. A. Rueppel, "Correlation immunity and the summation generator", Crypto 1985, pp. 260-272, 1985.

〈著者紹介〉

김 재 헌 (Jaeheon Kim) 정회원
1991년 2월: 한국과학기술원 수학과 학사
1993년 2월: 서울대 수학과 석사
2000년 8월: 서울대 수학과 박사
2000년 4월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 정보보호, 대수학

한 재 우 (Jaeheon Kim) 정회원
1991년 2월: 서강대학교 수학과 학사
1993년 2월: 한국과학기술원 수학과 석사
1999년 8월: 한국과학기술원 수학과 박사
1999년 7월~현재: 국가보안기술연구소 선임연구원
〈관심분야〉 정보보호, 위상수학

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