情報保護學會論文誌
第13券 第1號, 2003. 2

과포화(Overdefined) 연립방정식을 이용한 LILI-128 스트림 암호에 대한 분석*

문 덕 재**, 홍 석 희**, 이 상 진**, 임 종 인**, 은 희 천***

Cryptanalysis of LILI-128 with Overdefined Systems of Equations

Dukjae Moon**, Seokhie Hong**, Sangjin Lee**, Jongin Lim**, Hichun Eun***

요 약

본 논문은 과포화 다변수 방정식을 이용하여 LILI-128 스트림 암호를 분석한다. LILI-128 암호[8]는 128비트 키를 가진 선형귀환 쉬프트 레지스터 기반의 스트림 암호로 구조를 살펴보면 크게 "CLOCK CONTROL" 부분과 "DATA GENERATION" 부분으로 나뉘어진다. 분석 방법은 "DATA GENERATION" 부분에 사용되는 함수 f_d의 대수적 차수가 높지 못하다는 성질을 이용한다. 간략히 설명하면 차수(K)가 6차인 다변수 방정식을 많이 얻을 수 있고, 이를 7차(D)의 다변수 방정식으로 확장하여 주어진 변수보다 많은 연립방정식을 얻어 그 해를 구하는 XL 알고리즘을 통해 전수조사보다 빠르게 키정보를 찾을 수 있다. 결과 중 가장 좋은 것은 출력 키수열 $2^{26.3}$비트를 가지고 $2^{110.7}$ CPU 시간을 통해 128비트 키정보를 얻는 것이다.

ABSTRACT

In this paper we demonstrate a cryptanalysis of the stream cipher LILI-128. Our approach to analysis on LILI-128 is to solve an overdefined system of multivariate equations. The LILI-128 keystream generator[8] is a LFSR-based synchronous stream cipher with 128 bit key. This cipher consists of two parts, "CLOCK CONTROL", part and "DATA GENERATION", part. We focus on the "DATA GENERATION" part. This part uses the function f_d that satisfies the third order of correlation immunity, high nonlinearity and balancedness. But, this function does not have highly nonlinear order(i.e. high degree in its algebraic normal form). We use this property of the function f_d. We reduced the problem of recovering the secret key of LILI-128 to the problem of solving a largely overdefined system of multivariate equations of degree K=6. In our best version of the XL-based cryptanalysis we have the parameter D=7. Our fastest cryptanalysis of LILI-128 requires $2^{110.7}$ CPU clocks. This complexity can be achieved using only $2^{26.3}$ keystream bits.

Keyword : *Stream Cipher LILI-128, Overdefined System of Multivariate Equations, XL-Algorithm*

Ⅰ. 서 론

암호 알고리즘은 키를 미지수로 보고 다변수 연립방정식으로 전개했을 때 풀기 어려워야 한다. 그런데 특별히 기지 평문의 개수가 많아서 연립방정식에서 변수의 개수보다 방정식의 개수가 많은 경우, 이를 과포화(Overdefined)연립방정식이라 부르고 이때 해를 구하는 문제는 상당히 쉽다. 이런 성질은 비밀키 암

* 이 논문은 고려대학교 특별연구비에 의하여 수행되었음.
** 고려대학교 정보보호기술연구센터(CIST)({djmoon, hsh, sangjin, jilim}@cist.korea.ac.kr)
*** 고려대학교 자연과학대학 자연과학부(hceun@tiger.korea.ac.kr)

호체계상에서 그리 어렵지 않게 얻을 수 있다. 왜냐하면 주어진 비밀키를 통해 일정 기간동안의 암·복호화 이루어지기 때문이다. 특히, 스트림 암호에서 더욱 그러하다. 왜냐하면 획득할 수 있는 키수열의 량만큼의 방정식을 얻을 수 있기 때문이다.

본 논문에서는 많은 스트림 암호들의 구조가 낮은 차수의 과포화 연립방정식으로 쉽게 전환될 수 있음을 보인다. 이렇게 얻어진 과포화 연립방정식은 생각보다 쉽게 풀리는데 이 사실은 2000년 EUROCRYPT에서 N. Courtois외 3명에 의해 연구된 바 있다[2]. 이 결과는 1999년 CRYPTO에서 A. Shamir와 A. Kipnis에 의해 제시된 방정식들의 선형화 기술[1]을 발전시킨 것이다. 최근 이 과포화 연립방정식을 이용하여 비밀키 암호를 분석할 수 있다는 가능성은 N. Courtois와 J. Pieprzyk에 의해 발표되었고[6], 특히 이 방법을 스트림 암호에 적용한 논문[7]이 N. Courtois에 의해 발표되었다. 이런 종류의 공격이 가능함은 암호 알고리즘에 대응하는 공격 매개 변수가 커짐에 따라 많은 상수들이 생기지만 그 계산 복잡도는 느리게 증가함에 기인한다.

본 논문에서는 이 공격 방법을 LILI-128 스트림 암호에 적용한다. 이는 이 암호에 사용되는 비선형함수의 대수적 차수가 작다는 사실을 이용한 것이다. 주어진 비선형함수를 통해 다변수 방정식을 얻어 LILI-128 스트림 암호를 분석한 결과는 키 전수조사보다 빠르며, 분석 결과는 [표 1]과 같다. 특별히 이 분석 결과는 기존의 분석 결과와 달리 비연속 키수열을 가지고도 분석이 가능하다. 이 결과를 살펴보면 기존의 공격[12,13]보다 적은 키수열이 사용되지만 이들 공격보다 많은 메모리와 계산량이 필요하다. 이 표에서 "D&C"는 Divied-and-Conqure 공격[8]을 나타내고, "FCA"는 Fast correlation 공격[12]을 나타내며, "TMA"은 Time Memory trade-off 공격[13]을 나타낸다. 또한, XL Alg (1,2,3)은 D. Coppersmith의 알고리즘[3]을 사용한 결과이고, XL Alg (4)는 Strassen의 알고리즘[4]을 사용한 결과이다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 2차보다 높은 차수에 대한 다변수 연립방정식을 푸는 확장된 XL 알고리즘을 살펴본다. 3장에서는 확장된 XL 알고리즘을 스트림 암호에 적용하는 방법을 설명하고, 이 방법을 적용할 LILI-128 스트림 암호를 4장에서 설명한다. 이후 5장에서 실제적으로 XL 알고리즘을 LILI-128 암호에 적용하는 과정 및 계산 과정을 살펴본다. 마지막으로 본 논문의 결론 및 공격 결과를 정리한다.

II. XL 알고리즘

본 장에서는 2000년 N. Courtois외 3명에 의해 EUROCRYPT에서 발표된 확장된 XL 알고리즘에 대하여 살펴본다. 주어진 체 $GF(q)$위에서 정의된 n개의 변수로 이루어진 m개의 방정식을 생각하자. 이때 방정식의 차수 K는 2이상이다. 변수 D를 XL 알고리즘의 매개 변수라 하며, 기본 방정식으로부터 얻을 수 있는 새로운 방정식들의 최고 차수를 의미한다. 그리고 이 방정식들의 특별한 출력값 $b=(b_0,\cdots,b_{m-1})$을 얻었을 때, 이에 대응하는 식의 모양이 $f_i(x_0,\cdots,x_{n-1})$이라면 우리는 기본 방정식을 $l_i(x_0,\cdots,x_{n-1})=f_i(x_0,\cdots,x_{n-1})-b_i$ 와 같이 정의한다. 그리고 얻어진 출력값들을 이용하여 m개의 방정식을 다음과 같이 얻는다.

$$A:\begin{cases} l_0(x_0,\cdots,x_{n-1})=0 \\ \vdots \\ l_{m-1}(x_0,\cdots,x_{n-1})=0 \end{cases}$$

이 m개의 기본 연립방정식들과 XL 알고리즘을 이용하여 해를 구할 수 있다.

〔표 1〕 LILI-128에 대한 공격 결과들

공격 유형	공격 결과			비고
	키수열량	메모리량	공격 복잡도	
D&C	2^{11}	-	2^{112}	설계자들의 가설[8]
FCA	2^{30}	2^{79}	2^{71}	IPL 2001[12]
TMA	2^{46}	2^{45}	2^{48}(DES)	FSE 2002[13]
XL Alg (1)	$2^{26.3}$	$2^{65.4}$	$2^{110.7}$	본 논문 결과 (K=6, D=7)
XL Alg (2)	$2^{24.1}$	$2^{72.1}$	$2^{118.6}$	본 논문 결과 (K=6, D=8)
XL Alg (3)	$2^{22.4}$	$2^{78.4}$	$2^{126.2}$	본 논문 결과 (K=6, D=9)
XL Alg (4)	$2^{26.3}$	$2^{65.4}$	$2^{127.6}$	본 논문 결과 (K=6, D=7)

[정의 1] (XL 알고리즘)

XL 알고리즘은 다음의 단계를 거쳐 수행된다.

1. **변수들의 곱 :** 매개 변수 D에 대하여 $\prod_{j=1}^{k} x_{i_j} \cdot l_i$를 계산한다. 여기서 k는 $D-K$보다 작거나 같은 값이다. 이렇게 얻어진 방정식의 차수는 D보다 작거나 같다.
2. **선형화 작업 :** D보다 작은 차수의 x_i의 모든 항들(Monomials)을 새로운 변수로 생각하고, 1단계에서 얻어진 방정식에 대한 가우스 소거법을 시행한다.
3. **방정식 풀기 :** 2단계에서 가우스 소거법을 통해 변수 x_1의 지수승들 만으로 이루어진 방정식을 얻는다면 이 방정식을 주어진 체위에서 Berlekamp의 알고리즘을 통해 푼다.
4. **위과정 반복 :** 위의 과정을 반복하여 모든 변수에 대한 값을 찾는다.

위 알고리즘의 동작에 대한 이해를 돕기 위해 간단한 예를 살펴보다.

예 1. (XL 알고리즘 동작의 간단한 예)

$\mu \neq 0$으로 놓고 다음의 2차 연립방정식을 생각한다.

$$\begin{cases} x_1^2 + \mu x_1 x_2 = \alpha & (1) \\ x_2^2 + \nu x_1 x_2 = \beta & (2) \end{cases}$$

XL 알고리즘의 매개 변수 $D=4$로 놓고 1단계 과정을 통해 4차의 여러 독립인 방정식을 더 얻는다. 이때 위의 식 (1), (2)를 기본 방정식이라 부르며 이를 l이라 놓는다. 이 방정식에 2차의 단항을 곱하여 얻어진 방정식들은 $l \bigcup x^2 \cdot l$ 집합의 원소들이다. 여기서 x^2는 x_1^2, x_2^2, x_1x_2를 의미한다. 이런 과정을 통해 추가적으로 얻어지는 방정식의 개수는 6개 (2×3)이고 그 모양은 아래와 같다.

$$\begin{cases} x_1^4 + \mu x_1^3 x_2 = \alpha x_1^2 & (3) \\ x_1^2 x_2^2 + \nu x_1^3 x_2 = \beta x_1^2 & (4) \\ x_1^2 x_2^2 + \mu x_1 x_2^3 = \alpha x_2^2 & (5) \\ x_2^4 + \nu x_1 x_2^3 = \beta x_2^2 & (6) \\ x_1^3 x_2 + \mu x_1^2 x_2^2 = \alpha x_1 x_2 & (7) \\ x_1 x_2^3 + \nu x_1^2 x_2^2 = \beta x_1 x_2 & (8) \end{cases}$$

이렇게 얻어진 총 8개의 방정식을 가지고 2단계 과정을 통해 x_1변수에 대한 4차 방정식으로 정리할 수 있는데 그 과정은 다음과 같다

- (1)번식으로부터 $x_1x_2 = \frac{\alpha}{\mu} - \frac{x_1^2}{\mu}$ 의 관계식을 얻는다.
- (2)번식으로부터 $x_2^2 = \left(\beta - \frac{\alpha \cdot \nu}{\mu}\right) + \frac{\nu}{\mu} \cdot x_1^2$의 관계식을 얻는다.
- (3)번식으로부터 $x_1^3x_2 = \frac{\alpha}{\mu} x_1^2 - \frac{x_1^4}{\mu}$ 의 관계식을 얻는다.
- (4)번식으로부터 $x_1^2x_2^2 = \left(\beta - \frac{\alpha \cdot \nu}{\mu}\right) \cdot x_1^2 + \frac{\nu}{\mu} \cdot x_1^4$의 관계식을 얻는다.
- (8)번식으로부터 $x_1x_2^3 = \frac{\alpha \cdot \beta}{\mu} + \left(\frac{\alpha \cdot \nu^2}{\mu} - \beta \cdot \nu - \frac{\beta}{\mu}\right)x_1^2 - \frac{\nu^2}{\mu} x_1^4$ 의 관계식을 얻는다.
- (6)번식으로부터 $x_2^4 = \left(\beta^2 - \frac{2\alpha \cdot \beta \cdot \nu}{\mu}\right) + \left(\frac{2\nu \cdot \beta}{\mu} + \beta \cdot \nu^2 - \frac{\alpha \cdot \nu^2}{\mu}\right) \cdot x_1^2 + \frac{\nu^3}{\mu} \cdot x_1^4$ 의 관계식을 얻는다.

이 6개의 관계식을 5번식에 대입하면 변수 x_1 한 개로 이루어진 하나의 방정식을 얻게 되고 그 모양은 다음과 같다.

$$\alpha^2 + x_1^2(\alpha \cdot \mu \cdot \nu - \beta \cdot \mu^2 - 2\alpha) + x_1^4(1 - \mu \cdot \nu) = 0$$

이 방정식을 Berlekamp의 알고리즘을 통해 풀면 x_1의 값을 알 수 있고, 이 값을 원래 방정식에 대입한 후 x_2의 값을 찾아내면 된다.

이 알고리즘에 가장 큰 문제는 변수들의 곱을 통해 얻어진 여러 방정식들 모두가 독립이 아니라는 것이다. 따라서 위의 알고리즘으로 반드시 하나의 해를 찾을 수 있다는 것을 보장하기 위해서는 얻어진 여러 방정식들이 최소한 변수의 개수 이상으로 독립임을 보여야한다. 이를 위해 N. Courtois는 컴퓨터 작업을 통해 다음의 결과[7]를 찾아내어 공격에 사용하였다. 이 가설에 사용되는 변수들은 다음과 같다. R은 XL 알고리즘에 의해 생성되는 방정식의 수를 나타내며, 그 값은 $m \times \sum_{i=0}^{D-K} \binom{n}{i}$이다. 그리고 T는 D차까지 가질 수 있는 단항들의 수를 의미하며, 그 값을 계산하면 $\sum_{i=0}^{D} \binom{n}{i}$와 같다.

가설 1. (D < 3K상에서의 XL 알고리즘의 동작)

1. $D=K,\cdots,2K-1$**인 경우 :** $R \geq T$일 때 XL 알고리즘을 통해 얻어진 방정식들은 모두 독립이고, 독립인 방정식의 수 $Free$는 $Min(T,R)-\varepsilon$과 같다. 여기서 ε은 0, 1, 2, 또는 3이다.
2. $D=2K,\cdots,3K-1$**인 경우 :** 이 경우 XL 알고리즘을 통해 얻어진 방정식들은 모두 독립이 아니다. 이에 대한 컴퓨터 작업 결과를 일반화하면, 독립인 방정식의 수 $Free$는 $Min\left(T, R-\left(\sum_{i=0}^{D-2K}\binom{n}{i}\right)\left(\binom{m}{2}+m\right)\right)-\varepsilon$과 같다. 여기서 ε은 0, 1, 2, 또는 3이다. $\left(\binom{m}{2}+m\right)$의 값은 XL 알고리즘을 통해 얻어진 방정식들 중 $l_i \cdot l_j = l_j \cdot l_i$와 $l_i \cdot l_i = l_i$의 사실에 의해 만들어진 값이다. 더욱이 D가 $2K$보다 클 경우, $D-2K$까지 차수의 x_i항을 곱하는 방정식들은 모두 독립이 아니다. 따라서 $\sum_{i=0}^{D-2K}\binom{n}{i}$의 값을 곱하게 된다.

이 가설을 이용하면 XL 알고리즘을 통한 분석이 가능하게 된다. 왜냐하면 $R \geq T$을 만족하는 매개 변수 D에 대하여 만일 $Free \geq T-D$의 조건을 만족한다면, 이 XL 알고리즘은 가우스 소거법에 의해 하나의 해를 구할 수 있음이 알려져 있다. 이때의 $Free$ 값은 위의 N. Courtois의 가설에 의해 구하여 조건을 만족하는지를 확인해 보면 된다.

이 XL 알고리즘의 복잡도는 가우스 소거법(Gaussian Elimination)의 복잡도로 T^3임이 알려져 있다. 하지만 1969년에 V. Strassen과 1990년 D. Coppersmith에 의해 더 작은 값으로 복잡도를 낮추었다. 다음은 이 두 가지 방법을 통한 복잡도 계산 방법이고, 이후 본 논문의 복잡도 계산 결과들은 이 방법들에 의해 계산되어진다.

1. **[D. Coppersmith의 알고리즘] :** 복잡도는 약 $T^{2.376}$이고, 이는 $\frac{1}{64} \cdot T^{2.376}$ CPU clocks을 갖는다.
2. **[V. Strassen의 알고리즘] :** 복잡도는 약 $7 \cdot T^{\log_2^7}$이고, 이는 $\frac{7}{64} \cdot T^{\log_2^7}$CPU clocks을 갖는다.

Ⅲ. XL 알고리즘의 스트림 암호에 적용

본 장에서는 일반적인 스트림 암호에 대한 XL 알고리즘을 적용하는 방법에 대하여 설명한다.

우선, 각 레지스터의 상태(State)가 평문과 상관없이 이전 상태에 의해 생성되는 동기식 스트림 암호에 대하여 적용한다. 이때 스트림 암호는 정규적인 클럭을 갖는다고 가정한다. 즉, 이진 스트림 암호에 대하여 한번에 한 비트의 키수열을 생성한다고 생각하자. L을 이 암호에 사용되는 공개된 연결 함수(connection function)라고 놓는다. 이 함수는 비밀 키 정보와는 상관없이 이전 상태의 정보를 통해 다음 상태의 정보를 생성해 내는데 이 과정은 선형적으로 이루어진다. 이렇게 생성된 상태들은 비선형 과정(nonlinear filtering)을 통하여 최종의 키수열을 생성하게 된다. 이런 스트림 암호의 분석에 필요한 방정식은 다음과 같이 만들어진다. $(k_0,\cdots,k_{n-1})$을 비밀의 초기 상태정보라고 놓으면, 이 스트림 암호의 출력, 즉 키수열은 다음과 같이 주어진다.

$$\begin{cases} f(k_0,\cdots,k_{n-1}) \\ f(L(k_0,\cdots,k_{n-1})) \\ f(L^2(k_0,\cdots,k_{n-1})) \\ \vdots \end{cases}$$

비정규적인 클럭을 이용하는 스트림 암호의 경우 클럭되는 값을 알 수 있다면 이 방법을 적용할 수 있다. 위의 설명과 비슷하게 알고 있는 선형 연결 함수를 통한 상태 정보를 가지고 미리 얻어진 클럭값 t를 통해 얻어진 m개의 키수열과 초기 값들과의 관계는 다음과 같이 m개의 방정식으로 생각할 수 있다.

$$\begin{cases} b_{t_1} = f(L^{t_1}(k_0,\cdots,k_{n-1})) \\ b_{t_2} = f(L^{t_2}(k_0,\cdots,k_{n-1})) \\ \vdots \\ b_{t_m} = f(L^{t_m}(k_0,\cdots,k_{n-1})) \end{cases}$$

이 방정식에서 사용되는 선형 함수 L과 비선형 함수 f가 공개되어 있고 단지 초기 상태 정보 k_i만 비밀이다. 이 연립방정식을 얻기 위해서는 m개의 키수열을 얻어야 한다. 하지만 이 비트 정보는 연속된 정보일 필요는 없다.

지금까지 설명한 방법은 스트림 암호에 사용되는 비선형 과정인 f함수가 특정한 성질을 만족할 때 가능하다. 이 성질을 살펴보기 전에 우선 지금까지 알려진 비선형 함수의 설계 원리를 살펴본다.

1. **Balancedness** : 통계적 테스트를 통한 공격을 피하기 위해 필요한 성질.
2. **Non-linearity** : Fast correlation 공격을 피하기 위해 필요한 성질.
3. **High - order** : 대수적 관점의 공격을 피하기 위해 필요한 성질.
4. **Higher-order correlation immunity** : Correlation 공격을 피하기 위해 필요한 성질.

이들 중 다음의 두가지 성질들이 만족될 때 공격이 가능하게 된다.

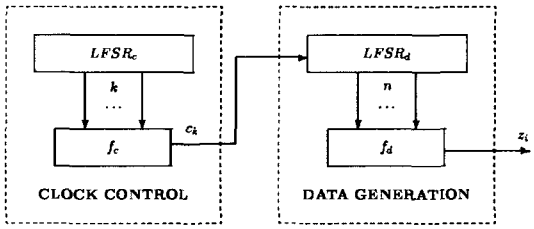
1. 비선형 과정의 f 함수의 대수적 차수 K가 작을 경우.
2. 비선형 과정의 f 함수의 대수적 차수 K는 크지만 차수가 작은 다른 함수 g에 대하여 높은 확률로 근사 될 수 있는 경우.

참고 논문 [7]에서는 N. Courtois가 두 번째 성질을 이용하여 CRYPTREC project에 제안된 Toyocrypt 스트림 암호를 분석하였다. 본 논문에서는 이와는 달리 첫 번째 성질을 이용하여 LILI-128 스트림 암호를 분석한다. 이는 LILI-128 암호가 낮은 대수적 차수를 갖는 비선형 함수를 사용하기에 가능하다. 하지만 이 암호는 비정규적 클럭 과정을 사용하기에 앞에서 설명한 바에 의하면 XL 알고리즘을 적용하는데 어려움이 있다. 따라서 공격을 위해서는 클럭 값을 생성해 내는 "CLOCK CONTROL"부분에 대한 모든 가능한 경우를 다 고려하여 공격한다. 즉, "CLOCK CONTROL" 부분에 사용되는 $LFSR$의 초기 상태 정보를 전수조사하여 그 값에 따른 공격을 한다.

Ⅳ. LILI-128 암호 소개

이번 장에서는 공격에 사용될 스트림 암호 LILI-128의 구조를 살펴본다. LILI-128은 클럭에 의해 움직이는 비선형 과정을 이용하는 키수열 생성기이다. 이 암호의 구조는 두 부분으로 나뉘어 있는데 그 한 부분이 "CLOCK CONTROL"부분이고 다른 한 부분은 "DATA GENERATION"부분이다. 각 부분은 각자의 $LFSR$을 가지고 있는데 이를 각각 $LFSR_c$와 $LFSR_d$라 놓는다. 이 두 $LFSR$들은 총 128비트 초기 상태 정보를 사용한다. 이 구조는 [그림 1]과 같다.

각 부분을 자세히 살펴보면 다음과 같다.



〔그림 1〕 LILI-128 스트림 암호의 구조

우선, "CLOCK CONTROL"부분은 39비트 초기 상태 정보가 입력되어 2비트의 출력을 내는 부분이다. 이 과정에 사용되는 $LFSR_c$는 다항식 $g_c(x)=x^{39}+x^{35}+x^{33}+x^{31}+x^{17}+x^{15}+x^{14}+x^2+1$을 연결 다항식으로 가지는 $LFSR$이다. $g_c(x)$는 원시(Primitive) 다항식이므로 $LFSR_c$는 최대 길이의 수열을 생성한다. 이때 $LFSR_c$의 12번째와 20번째 상태 정보의 값이 f_c함수의 입력으로 들어간다. 이 f_c함수는 클럭값인 정수 t_i를 생성하며 그 정의는 $t_i=f_c(y_1,y_2)=2y_1+y_2+1$, $i\geq 1$와 같다. 이 함수의 출력 수열 $t=t_1, t_2, \cdots$는 "DATA GENERATION"부분의 $LFSR$인 $LFSR_d$의 클럭 값이 된다. 즉, t_i의 값만큼의 동작을 통하여 출력값 z_i를 출력한다. 이때 f_c함수의 정의에 의하여 최소 한번에서 최대 네 번의 클럭이 이루어짐을 알 수 있다.

"DATA GENERATION"부분의 $LFSR$인 $LFSR_d$은 원시 다항식 $g_d(x)=x^{89}+x^{83}+x^{80}+x^{35}+x^{53}+x^{42}+x^{39}+x+1$을연결다항식으로 가지는 $LFSR$이다. 이때 $LFSR_d$의 10개의 상태 정보 (0, 1, 3, 7, 12, 20, 30, 44, 65, 80)가 부울 함수 f_d의 입력값이 된다. 이 부울 함수 f_d는 비선형 과정으로 균일(Balanced)하고 높은 비선형성을 갖으며 3차의 상관 면역도(correlation immunty)를 갖는 좋은 함수이다. 이 함수의 출력값이 바로 이 스트림 암호의 키수열로 $z_1, z_2, \cdots$와 같다. 이 비선형 과정의 부울 함수의 모양은 다음과 같다.

$$
\begin{aligned}
f_d = {} & x_3x_5x_6x_7x_8x_9 \oplus x_4x_5x_6x_7x_8x_9 \oplus x_2x_6x_7x_8x_9 \\
& \oplus x_3x_5x_6x_7x_8 \oplus x_3x_5x_6x_8x_9 \oplus x_3x_6x_7x_8x_9 \\
& \oplus x_4x_5x_6x_7x_8 \oplus x_4x_5x_6x_8x_9 \oplus x_0x_7x_8x_9 \oplus \\
& x_1x_6x_7x_8 \oplus x_1x_6x_8x_9 \oplus x_2x_6x_7x_9 \oplus x_2x_7x_8x_9 \\
& \oplus x_3x_5x_6x_8 \oplus x_3x_6x_7x_8 \oplus x_3x_6x_8x_9 \oplus x_3x_7x_8x_9 \\
& \oplus x_4x_5x_6x_8 \oplus x_4x_6x_7x_9 \oplus x_5x_6x_8x_9 \oplus \\
& x_5x_7x_8x_9 \oplus x_1x_8x_9 \oplus x_2x_6x_8 \oplus x_2x_7x_8 \oplus \\
& x_2x_7x_9 \oplus x_2x_8x_9 \oplus x_3x_6x_8 \oplus x_3x_6x_9 \oplus x_3x_7x_9 \\
& \oplus x_3x_8x_9 \oplus x_4x_6x_9 \oplus x_4x_8x_9 \oplus x_5x_6x_8 \oplus
\end{aligned}
$$

$$x_5x_6x_9 \oplus x_5x_7x_8 \oplus x_0x_7 \oplus x_0x_8 \oplus x_1x_7 \oplus x_2$$
$$x_8 \oplus x_3x_9 \oplus x_5x_6 \oplus x_5x_9 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4$$

마지막으로 출력 키수열 $z_1, z_2, \cdots$과 평문들을 비트별 XOR하여 암호문을 생성한다.

V. XL 알고리즘의 LILI-128 에 적용

지금부터 LILI-128 스트림 암호에 XL 알고리즘을 적용한다. 이는 $LFSR_d$이 알려져 있기에 비밀 89비트 초기 상태 정보 $k=(k_0, \cdots, k_{88})$에 대한 t_i 클럭 후의 결과 $s_0, \cdots, s_{88}$를 $s_i = LFSR_d^{t_i}(k)$와 같이 나타낼 수 있다. 또한 알려진 f_d함수를 통해 그 출력값을 계산할 수 있고, 그 값은 $f_d(LFSR_d^{t_i}(k))$이다. 전수조사를 통해 얻어진 클럭값 t_i에 대하여 다음과 같이 여러 방정식을 얻을 수 있다.

$$\begin{cases} z_{t_1} = f_d(LFSR_d^{t_1}(k_0, \cdots, k_{88})) \\ z_{t_2} = f_d(LFSR_d^{t_2}(k_0, \cdots, k_{88})) \\ z_{t_3} = f_d(LFSR_d^{t_3}(k_0, \cdots, k_{88})) \\ \vdots \end{cases}$$

이 비선형 함수 f_d의 대수적 차수(K)가 6이므로 XL 알고리즘을 이용하여 분석이 가능하고, 이 분석 과정은 다음과 같다.

목표 : 128비트 키 찾기.

1. 주어진 D ($>K$)에 대하여, R, T의 값을 계산한다.
2. $R/T \geq 1$을 만족하게 하는 값 m을 선택한다.
3. 만일 $m < 2^{89}$라면, D. Coppersmith의 알고리즘을 통한 복잡도 C_1을 계산한다.
   1) 만일 $C_1 < 2^{89}$이면, XL 알고리즘을 통하여 128비트 키를 찾아내고, 조건을 만족하지 않는다면 3 과정을 종료 한다.
   2) 이때의 공격에 필요한 키수열의 양은 m이고, 메모리의 양은 $M = T \times T$이며 총 공격 복잡도는 $2^{39} \cdot C_1$CPU clocks이다.
4. V. Strassen의 알고리즘을 통한 복잡도 C_2을 계산한다.
   1) 만일 $C_2 < 2^{89}$이면, XL 알고리즘을 통하여 128비트 키를 찾아내고, 만일 조건을 만족하지 않는다면 전체 과정을 종료한다.
   2) 이때의 공격에 필요한 키수열의 양은 m이고, 메모리의 양은 $M = T \times T$이며 총 공격 복잡도는 $2^{39} \cdot C_2$ CPU clocks이다.

이 공격 과정을 통하여 매개 변수 D가 7인 경우의 결과를 살펴보면 다음과 같다.

$$R = m \cdot \sum_{i=0}^{7-6} \binom{89}{i} \approx m \cdot \binom{89}{1} = 89m,$$
$$T = \sum_{i=0}^{7} \binom{89}{i} \approx \binom{89}{7} = 689026857 2,$$
$$m = 2^{26.3} \qquad (\because R/T \geq 1),$$
$$M = T^2 = 2^{65.4},$$
$$C_1 = \frac{1}{64} T^{2.376} = 2^{71.7}, \quad Total\ Complexity = 2^{110.7},$$
$$C_2 = \frac{7}{64} T^{\log \frac{7}{2}} = 2^{88.6}, \quad Total\ Complexity = 2^{127.6}.$$

이와 같은 방법으로 모든 경우에 대한 복잡도를 계산할 수 있다. 가능한 공격에 대한 결과는 [표 1]에 정리하였다. 이중 가장 좋은 결과는 D가 7일 경우 필요한 키수열의 양은 $2^{26.3}$이고, 메모리량은 $2^{65.4}$이며 총 계산 복잡도는 $2^{110.7}$ CPU clocks이다.

VI. 결 론

본 논문에서는 과포화 연립방정식을 이용하여 LILI-128 스트림 암호에 대한 분석하였는데, 연립방정식을 풀기 위해서 확장된 XL 알고리즘을 사용하였다.

본 논문은 스트림 암호에 대한 새로운 분석 방법을 제시하였다는데 그 의미가 크며 특히 스트림 암호 설계시 요구되는 설계 원리의 준수가 얼마나 중요한지를 잘 보여준다. 공격 결과는 [표 1]에 정리되었으며, 그중 가장 좋은 결과는 D가 7일 경우 필요한 키수열의 양은 $2^{26.3}$이고, 메로리량은 $2^{65.4}$이며, 총 계산 복잡도는 $2^{110.7}$ CPU clocks이다.

과포화 연립방정식을 통한 분석 방법은 정규적인 클럭을 사용하는 동기식 스트림 암호나, 비정규적이지만 그 정보를 얻을 수 있는 스트림 암호면 어느 것이든지 적용 가능한 방법이다. 특히, 최근 제안되고 있는 비트 단위의 스트림 암호가 아닌 워드 단위의 스트림 암호에도 적용이 가능한 공격 방법으로 비밀키 암호 분석에 폭넓게 적용될 매우 중요한 공격법이라고 생각되며, 지속적인 연구가 이루지리라 예상된다.

참 고 문 헌

[1] A. Shamir and A. Kipnis, "Cryptanalysis of the HFE Public Key Cryptosystem", Advances in Cryptology - CRYPTO'99, LNCS 1666, Springer-Verlag, pp. 19~30, 1999.

[2] A. Shamir, J. Patarin, N. Courtois and A. Klimov, "Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations", Advances in Cryptology - EUROCRYT' 2000, LNCS 1807, Springer-Verlag, pp. 392~407, 2000.

[3] D. coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions",]. Symbolic Computation, Vol. 9, pp. 251~280, 1990.

[4] V. Strassen, "Gaussian Elimination is Not Optimal", Numerische Mathematik, Vol. 13, pp. 354~356. 1969.

[5] W. Meier, N. Courtois, L. Goubin and J. D. Tacier, "Solving Underdefined Systems of Multivariate Quadratic Equations", Public Key Cryptography - PKC2002, LNCS 2274, Springer-Verlag, pp. 211~227, 2002.

[6] N. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations", Advances in Cryptology - ASIACRYPT 2002, Springer-Verlag, 2002.

[7] N. Courtois, "Higher Order Correlation Attacks, XL algorithm and Cryptanalysis of Toyocrypt", Information Security and Cryptology - ICISC 2002, Springer-Verlag, 2002.

[8] E. Dawson, J. Golic, W. Millan and L. Simpson, "The LILI-128 Keystream Generator", Selected Areas in Cryptography - SAC 2000, LNCS 2012, Springer-Verlag, pp. 248~261, 2000.

[9] S. Babbage, "Cryptanalysis of LILI-128", NESSIE Public Report, https://www.cosic.esat.kuleuven.ac.be/nessie/ reports, 2001.

[10] E. Dawson, J. Golic, W. Millan and L. Simpson, "Response to initial Report on LILI-128", Submitted to Second NESSIE Workshop, 2001.

[11] J. White, "Initial Report on the LILI-128 Stream Cipher", NESSIE Public Report, https://www. cosic. esat.kuleuven.ac.be/nessie/reports, 2001.

[12] F. Jonsson and T. Johansson, "A Fast Correlation Attack on LILI-128", Information Processing Letters. Vol 81, No. 3, pp. 127~132, 2001.

[13] M-J. O. Saarinen, "A Time-Memory Tradeoff Attack Against LILI-128}", Fast Software Encryption 2002, LNCS 2365, Springer-Verlag, pp. 231~236, 2002.

[14] W. Meier and O. Staffelbach, "Nonlinearity Criteria for Cryptographic Functions", Advances in Cryptology - EUROCRYT'89, LNCS 434, Springer - Verlag, pp. 549~562, 1989.

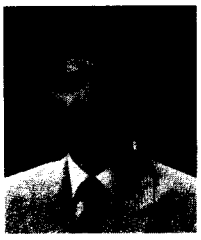
[15] P. Camion, C. Carlet, P. Charpin and N. Sendrier, "On Correlation-immune Functions", Advances in Cryptology - CRYTO'91, LNCS 576, Springer-Verlag, pp. 86~100, 1991.

[16] J. Golic, "Fast low order approximation of Cryptographic Functions", Advances in Cryptology - EUROCRYT'96, LNCS 1070, Springer-Verlag, pp. 268~282, 1996.

〈著者紹介〉

문 덕 재 (Duk-Jae Moon)
2000년 2월 : 서울시립대학교 수학과 학사
2001년 3월~현재 : 고려대학교 정보보호대학원 석사 과정
<관심분야> 블록 암호 및 스트림 암호 분석

홍 석 희 (Seok-hie Hong)
1995년 2월 : 고려대학교 수학과 학사
1997년 2월 : 고려대학교 수학과 석사
2001년 2월 : 고려대학교 수학과 박사
2001년 3월~현재 : 고려대학교 정보보호기술연구센터 선임 연구원
<관심분야> 블록 암호 및 스트림 암호의 분석과 설계



이 상 진 (Sang-Jin Lee)
1987년 2월 : 고려대학교 수학과 학사
1989년 2월 : 고려대학교 수학과 석사
1994년 2월 : 고려대학교 수학과 박사
1989년 2월~1999년 2월 : 한국전자통신연구소 선임 연구원
1999년 3월~현재 : 고려대학교 자연과학대학 부교수, 고려대학교 정보보호대학원 겸임 교수, 고려대학교 정보보호기술연구센터 연구실장
<관심분야> 블록 암호 및 스트림 암호의 분석과 설계, 암호 프로토콜, 공개키 암호 알고리즘 분석

임 종 인 (Jong-In Lim)
1980년 2월 : 고려대학교 수학과 학사
1982년 2월 : 고려대학교 수학과 석사
1986년 2월 : 고려대학교 수학과 박사
1986년 2월~현재 : 고려대학교 수학과 정교수
2000년 10월~현재 : 고려대학교 정보보호대학원 원장
<관심분야> 블록 암호 및 스트림 암호의 분석과 설계, 암호 프로토콜, 공개키 암호 알고리즘 분석

은 희 천 (Hi-Chun Eun)
1969년 2월 : 고려대학교 이공대학 학사
1974년 2월 : 고려대학교 수학과 석사
1982년 2월 : 고려대학교 수학과 박사
1982년 3월~현재 : 고려대학교 자연과학대학 수학과 교수
<관심분야> 블록 암호 및 스트림 암호의 분석과 설계, 암호 프로토콜, 공개키 암호 알고리즘 분석