情 報 保 護 學 會 誌
第 15 卷 第 2 號, 2005. 4

최근 스트림 암호 동향 분석

류 희 수*

요 약

LFSR에 기반한 고전적인 스트림 암호는 군사적 목적으로 많이 사용되었으나 차츰 공격법의 개발과 소프트웨어적으로 성능이 우수한 블록 암호가 나타나면서 상대적으로 그 사용이 현저히 줄어들게 되었다. 표준화된 스트림 암호의 예를 찾기 힘든 것도 그 일례라 할 수 있다. 최근 USN 등의 차세대 환경이 논의되고 현실화되면서 하드웨어 구현이 용이하고 뛰어난 성능을 갖고 있는 스트림 암호가 다시 각광을 받고 있다. 1990년대 후반부터 많은 새로운 스트림 암호가 개발되고 분석되고 있으며 여러 가지 새로운 분석 방법도 제시가 되고 있다. 본 논문에서는 이러한 새로운 스트림 암호와 이에 대한 새로운 분석 방법들을 살펴보고 현재 전 세계적으로 벌어지고 있는 스트림 암호 관련 표준화 및 연구 동향에 관해 알아본 후 스트림 암호의 설계 요구조건을 제시한다.

Ⅰ. 서 론

스트림 암호 알고리즘은 1970년대 초반 유럽에서 연구되어 발전된 선형 시프트 레지스터(Linear Feedback Shift Register, LFSR)를 이용한 이진 수열 발생기를 근간으로 하여 평문을 이진 수열로 부호화하여 이진 수열 발생기에서 생성된 이진 수열과 XOR (exclusive-or)하여 이진 수열로 된 암호문을 발생하는 암호 알고리즘이다. 여기에서 발생하는 이진 수열은 키 스트림이라고 불리며 난수와 구별이 불가능해야만 안전한 암호 알고리즘이라고 할 수 있다. 또한 스트림 암호 알고리즘은 생성된 키 스트림을 송신자와 수신자가 공유하고 있어야만 암호를 해독할 수 있는 대칭키 암호 알고리즘이다. 또 다른 대칭키 암호 알고리즘인 블록 암호 알고리즘과 비교하여 보면 블록 암호가 비트들을 블록 단위로 암호화하는 반면 스트림 암호는 비트 단위로 암호화하므로 에러 전파(propagation) 현상이 없고 일반적으로 블록 암호에 비해 속도가 빠르며 하드웨어로 구현이 용이하다는 장점을 가지고 있다. 또한 블록 암호와는 다르게 근간이 되는 함수들의 수학적 분석이 가능한 경우가 많아 여러 이론적인 값을 정확히 계산할 수 있다는 것도 장점 중의 하나이다. 스트림 암호는 평문과 키 스트림과의 연관성의 유무에 따라 동기식(synchronous)과 비동기식(asynchronous)으로 구분된다. 동기식 스트림 암호는 키 스트림이 평문과 무관하게 생성되기 때문에 키 스트림이 마스터 키에만 의존하여 주기적인 동기화 과정이 필요한 알고리즘이다. 반면에 비동기식 스트림 암호는 키 스트림이 평문과 마스터 키 모두에 의존하므로 주기적인 동기화 과정을 필요하지 않다. 스트림 암호의 초기에는 하드웨어 구현에 용이한 LFSR 기반의 암호 알고리즘이 많이 소개되어 사용되었으나 LFSR 자체는 그 선형성으로 인해 높은 주기성과 좋은 통계적 성질을 갖는 여러 개의 LFSR을 이용하여 Geffe generator, Summation generator 등과 같은 비선형 결합 방식, 비선형 여과 함수를 이용하는 방식, 하나 또는 두 개의 출력이 다른 LFSR의 출력을 제어하는 방식 등을 통하여 LFSR의 선형성을 안전하게 만드는 방법을 사용하였다. 이 외에도 FCSR 방식과 cellular automata를 이용하는 스트림 암호 등 많은 종류의 스트림 암호가 제안되었고 최근에는 블록 암호와 마찬가지로 블록 단위로 키를 생성하여

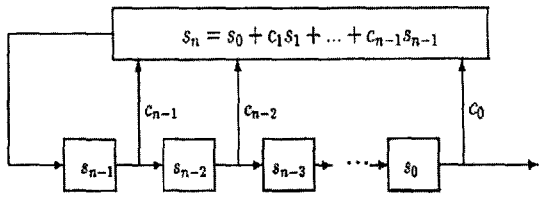
* 경인교육대학교 수학교육과 전임강사 (hsryu@ginue.ac.kr)

암호화하는 방식이 널리 제안되고 있는 실정이다. 암호기술이 전 세계적으로 일반화된 1990년대부터 소프트웨어 구현이 용이한 스트림 암호 알고리즘이 등장하기 시작하였다. 또한 기존에 사용되던 스트림 암호 알고리즘은 표준화라든가 안전성을 검증한다든가 등의 공식적 절차 없이 필요한 경우에 개발되어 사용되는 경우가 많았다. 그로 인하여 알려진 스트림 암호가 많지 않았고 스트림 암호에 대한 공격도 많이 이루어지지 않게 되었다. 그러나 1990년대 후반에 들어오면서 하드웨어의 성능에 의존하는 응용 제품이 늘어나면서 스트림 암호가 개발되고 이론에 기반을 둔 새로운 공격법이 개발되기 시작하였다. 2000년대에 들어와서 유럽의 NESSIE(New European Schemes for Signatures, Integrity, and Encryption), 일본의 CRYPTREC 등의 국제적인 암호 공모 사업의 일환으로 스트림 암호도 공모되어 여러 종류의 새로운 스트림 암호가 제안되었다. 또한 USN(Ubiquitous Sensor Network), RFID(Radio Frequency IDentification) 등의 환경을 고려하여 우리나라에서도 차세대 스트림 암호를 개발하고 있는 중이다. 본 논문에서는 이처럼 우리가 생활에서 많은 유용성을 얻을 수 있는 스트림 암호의 전반적인 부분을 알아보고 최신 동향 및 표준화 동향에 대해서도 살펴보도록 하겠다. 2장에서는 여러 스트림 암호에 대해 소개하고 3장에서는 최근에 발표된 공격법을 분석한다. 4장에서는 스트림 암호의 국내외 표준화 동향을 분석하고 마지막으로 5장에서 공격 기법 등을 고려한 스트림 암호 설계 요구사항에 대해 제시하고 마무리한다.

II. 여러 스트림 암호 개관

2.1 LFSR 기반 스트림 암호

기존의 스트림 암호는 하드웨어 구현의 용이성 등으로 LFSR 기반의 스트림 암호가 많으며 LFSR 기반의 스트림 암호는 많은 키 스트림 생성기에 사용되고 있다. 그 주요한 이유로는 LFSR이 하드웨어의 구현이 용이하다는 것이며 긴 주기를 생성할 수 있다는 것도 큰 장점이다. 또한 좋은 통계적 특성을 가지며 그 구조적 특성 때문에 대수적 기법에 의해 쉽게 분석할 수 있다. 그러나 이러한 점은 LFSR이 대수적 공격 등과 같은 공격에 쉽게 노출되는 단점을 야기하기도 한다. LFSR의 기본적인 구조의 예는 다음과 같다.



〔그림 1〕 n차 선형 시프트 레지스터

위의 그림에서 n차 선형 시프트 레지스터는 n개의 stage와 선형 feedback 함수로 구성된다. n개 단을 각각 $S_0, S_1, \cdots, S_{n-1}$로 나타내고 n개 단의 내용 $s_0, s_1, \cdots, s_{n-1}$을 하나의 state로 정의하며 $s_0 s_1 \cdots s_{n-1}$로 나타낸다. 또한 선형 feedback 함수는 다음과 같이 표시된다.

$$f(s_0, s_1, \cdots, s_{n-1}) = s_0 + c_1 s_1 + \cdots + c_{n-1} s_{n-1},$$

여기서 $c_1, c_2, \cdots, c_{n-1}$은 모두 0과 1의 값을 취하며 c_i의 값은 위의 그림에서 S_i단의 연결 상태를 나타낸다. Feedback 상수 $c_1, c_2, \cdots, c_{n-1}$을 갖는 임의의 n단 선형 시프트 레지스터의 특성 다항식을 다음과 같이 정의한다.

$$f(x) = 1 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} + x^n$$

n단 선형 시프트 레지스터에 의해 발생하는 이진 수열은 초기 상태와 feedback 상수에 의해 결정되므로 초기 상태와 특성 다항식에 의해 결정된다고 할 수 있다.

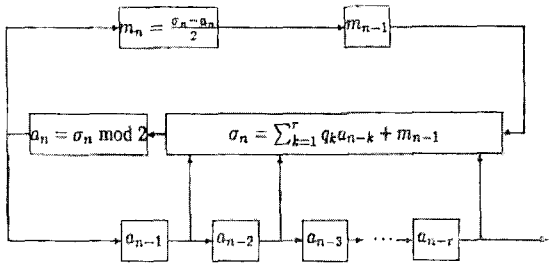
이러한 LFSR은 선형이라는 특징 때문에 여러 가지 비선형 논리를 사용하여 키 스트림 생성기를 구성한다. 이러한 비선형 논리의 예로는 결합 논리, 여과 논리, 시간 제어 논리 등이 있다.

이처럼 LFSR을 기반으로 하는 스트림 암호 알고리즘은 비선형화 과정을 거쳐 키 생성을 하게 되는데 잘 알려진 LFSR 기반 스트림 암호로는 mux generator, summation generator, shrinking generator, Geffe generator 같은 것들이 있다.

2.2 FCSR 기반 스트림 암호

이러한 LFSR 기반 스트림 암호와는 달리 1994년 Klapper와 Goresky에 의해 소개된 FCSR(Feed-

back Carry Shift Register)은 $1/p$ 생성자를 하드웨어 구현이 용이한 시프트 레지스터에 의하여 재구성한 것으로 선형 복잡도 측면에서 매우 우수한 feedback 레지스터이다. FCSR에서는 생성되는 수열을 유리수 p/q로 해석하며 이 때 q는 생성된 수열의 연결수이며 p는 FCSR의 초기 값이 된다. FCSR의 구성 및 동작은 다음과 같다.



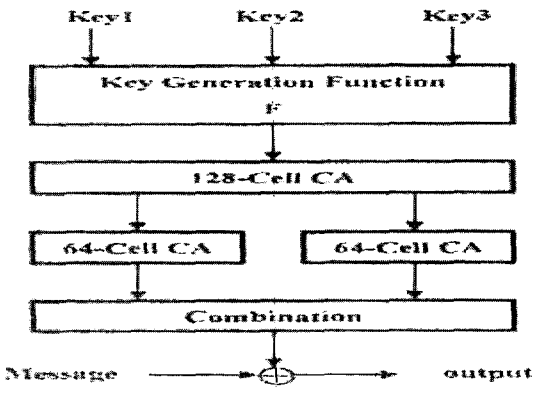
〔그림 2〕 FCSR의 구성 및 동작

FCSR의 특징은 LFSR의 특성이 다항식에 의해 결정되듯이 FCSR은 연결수에 의해 결정된다. 이 연결수의 성질에 따라 주기 및 선형 복잡도를 계산할 수 있는 경우도 있으며 그렇지 않은 경우도 있다. 동작 방법에서는 LFSR과 유사하지만 feedback 값이 다르게 계산된다. 즉 LFSR에서는 모든 위치의 값을 XOR한 결과를 feedback 하지만 FCSR은 모든 값을 정수 상에서 더한 다음 이 값을 이전 메모리 값과 더하고 이 결과를 2로 나눈 나머지 값을 feedback한다. 이와 같은 원리로 FCSR의 경우에는 carry를 저장할 수 있는 메모리를 갖고 있어야 한다. 또한 2- adic 수 위의 대수적 방법으로 분석이 가능하다는 점은 유한체 위의 성질을 가지는 LFSR과 비슷한 면이 있으며 어떤 주기적인 수열도 FCSR에 의해 생성될 수 있다.

2.3 기타 스트림 암호

위의 두 경우를 제외하고도 다른 종류의 스트림 암호가 제안되었다. 예를 들어 $GF(2)$ 상의 일차원 셀룰러 오토마타를 암호 알고리즘의 기본 구성 요소로 사용하는 방법도 제안되었는데 이의 구성은 다음과 같다.

이러한 1차원 셀룰러 오토마타를 이용함으로써 단순성과 모듈러, 규칙적인 구성, 국부적인 상호 작용 등의 효과를 얻을 수 있고 이로 인해 하드웨어 구현이 용이해짐에 따라 고속의 알고리즘 구현이 가능하게 된다.



〔그림 3〕 셀룰러 오토마타를 이용한 스트림 암호

2.4 최근의 스트림 암호

초기에 LFSR 기반의 스트림 암호가 많이 사용되었고 그 뒤에 FCSR 기반이나 셀룰러 오토마타 기반의 스트림 암호 등이 나타났지만 실제적으로 스트림 암호 기술이 전 세계적으로 일반화된 1990년대부터 하드웨어뿐만 아니라 소프트웨어 구현이 용이한 스트림 암호가 등장하기 시작하였다. RSA사에서 개발한 RC4와 IBM에서 개발한 SEAL이 바로 그것으로 RC4의 경우 개발 당시에는 알고리즘 구조가 알려지지 않았으나 1990년대 중반 알고리즘에 대한 구조가 알려진 후 이에 대한 많은 공격 기법이 제안되었다. 1993년 개발된 SEAL 알고리즘은 블록 암호와 해쉬 함수에서 사용되는 안전한 암호 기반 논리를 이용하여 설계되었으나 최초의 SEAL은 1997년 통계 분석을 통해 공격되었다. 이를 시발점으로 스트림 암호는 하드웨어의 구현뿐만 아니라 소프트웨어의 구현에도 적합하도록 많은 연구가 이루어지고 있는 실정이다. 최근의 스트림 암호는 LFSR에 기반한 설계 논리와 블록 암호의 설계 논리, 해쉬 함수의 설계 논리 등 많은 접목이 이루어져 개발되고 있는 실정이다. 그리하여 스트림 암호에 대한 공격법도 블록 암호나 해쉬 함수의 공격 기법 등이 응용되어 이루어지고 있다. 그러나 최근에 새로 개발되고 있는 스트림 암호에 대한 안전성 분석은 완전히 이루어지지 않고 있으며 유럽의 NESSIE나 일본의 CRYPTREC에서 공모하여 스트림 암호를 평가하고 선정하는 등의 노력이 지난 몇 년간 활발하게 이루어졌다.

최근의 스트림 암호를 살펴보면 블록 암호의 원리를 이용하여 스트림 암호 알고리즘을 구성하는 흐름이 있는데 블록 암호의 논리를 사용하여 개발된 스트림

암호의 경우는 SEAL이 처음이라 할 수 있으며 이러한 SEAL 계열의 스트림 암호는 블록 암호의 라운드 함수의 설계 논리를 기반으로 하여 설계된 것으로 기존의 블록 암호에서 안전성의 핵심 논리로 사용되는 S-box를 사용한다. 이러한 계열의 스트림 암호로는 TWOPRIME, SCREAM, HELIX 등이 있으며 SEAL이나 TWOPRIME의 경우는 전수 조사보다 강력한 공격법이 개발되었다. 또한 OFB(Output FeedBack), CFB(Cipher FeedBack), CTR(CounTeR) 모드 등 블록 암호의 modes of operation을 이용하여 키 스트림 수열을 생성할 수도 있다. 예를 들면 NESSIE에 제안된 BMGL의 경우가 그것이다. 이러한 블록 암호 기반의 스트림 암호들은 기본적으로 그 기반의 블록 암호의 분석에 의해 공격될 수 있다. 예를 들어 OFB 모드나 CTR 모드를 사용하는 경우 일반적인 블록 암호에 대한 구별 공격으로 분석될 수 있다.

또한, 최근에 NESSIE와 CRYPTREC 등의 표준화 공모를 통해 제안된 많은 종류의 스트림 암호가 있다. 이들은 비트 단위의 LFSR 기반 스트림 암호와 워드 단위의 LFSR 기반 스트림 암호, 기타 독창적인 암호들이 제안되고 있다. 전통적인 비트 단위의 LFSR 기반 스트림 암호는 결합 논리, 여과 논리 등을 사용하여 설계하였으나 이는 대수적 공격의 등장으로 대부분 약점이 노출되어 있는 상태이며 이를 줄이기 위해 LFSR의 크기를 크게 한다든지 시간 제어 논리를 사용하는 것은 메모리나 속도, 병렬 구현의 어려움 등으로 인해 문제점이 많이 발견되고 있다. 이에 비해 워드 단위의 LFSR 기반 스트림 암호는 비트 단위 연산에 비해 고속 동작이 가능하며 최근 많이 사용되고 있으나 워드 단위에 사용되는 유한체 연산은 열악한 환경에서는 적당한 방법이 아니며 아직은 충분한 검증이 이루어지지 않은 상태이다. NESSIE에 제안된 SOBER 계열의 스트림 암호가 이에 속하며 SOBER에서는 S-box의 논리도 사용하였다. 1998년 Daemen과 Clapp이 제안한 PANAMA나 이 계열의 스트림 암호인 CRYPTREC에 제안된 MULTI-S01, MUGI 등은 워드 단위의 LFSR 기반 스트림 암호이기는 하지만 S-box를 사용하지는 않는다. 또한 기타 반짝이는 아이디어를 사용하여 만든 스트림 암호들이 제안되고 있으나 안전성 분석 면에서 어려운 점이 있어 충분한 연구가 이루어져야 사용할 수 있을 것으로 보인다. 이런 스트림 암호로는 HC-256, VMPC 등을 들 수 있다.

Ⅲ. 스트림 암호 공격법

블록 암호 알고리즘의 분석과는 달리 LFSR 기반 스트림 암호 알고리즘은 반복적인 구조를 거의 가지지 않고 그 구조적 특징도 알고리즘마다 상이한 부분이 많아 분석 기법을 구체적으로 분류하기는 쉽지 않다. 그러나, 기존의 LFSR 기반 스트림 암호 알고리즘에 대한 대표적인 공격 기법을 바탕으로 다음과 같이 분류할 수 있다.

3.1 전통적인 스트림 암호 공격법

3.1.1 전수조사(Exhaustive key search) 공격

전수조사 공격은 어떤 종류의 스트림 암호에도 적용 가능한 가장 일반적인 공격 방법이다. 주어진 키 스트림 수열을 가지고 키를 모르는 상태에서 모든 이진 수열을 조사하여 키 수열을 찾아내는 방법이다. 이런 방법 중의 하나로 최근 스트림 암호의 공격에 많이 사용되는 Time-Memory trade-off 공격이 있다. 이 공격에서는 어떤 키 출력 관계를 미리 계산하여 메모리에 저장하고 실제 공격 상태에서는 출력 데이터를 저장된 출력 패턴과 일치하는지를 검사하여 해당되는 키를 찾아내는 방법이다. 이 경우 공격의 복잡도(complexity)는 시간과 메모리 복잡도로 나누어 생각한다.

3.1.2 주기 및 통계적 특성 공격

키 스트림 생성기의 주기가 너무 작으면 키 스트림은 반복될 수 있으며 이 경우 키를 쉽게 예측할 수 있게 된다. 그러므로 이를 방지하기 위해서는 키 스트림의 주기가 충분히 커야 한다. 일반적으로 키 스트림이 메모리가 없는 Bernoulli 분산으로부터 벗어나면 이를 분석할 수 있는 방법이 존재한다.

3.1.3 상관(Correlation) 공격

상관 공격은 1985년 Siegenthaler에 의해 처음 제안된 LFSR 기반의 스트림 암호 공격에 가장 기본적인 공격법이다. 이 공격에서는 키 스트림 생성기의 출력이 키 스트림 생성기에 사용되는 LFSR과 같은 훨씬 단순한 장치의 출력과 어떻게 연관되어 있는가를 분석하여 LFSR 버퍼의 초기 값(마스터 키)를 찾는 방법이다. 랜덤한 수열일 경우 특정한 입력 비트들과 출력 비트들 간의 상관관계가 발생할 확률은 1/2이다. 이 공격은 입력 비트와 출력 비트들 간의 상관 관계식의 확률이 1/2를 벗어날 경우 이 식을 이용하여 공격

하는 방법이다. 이는 블록 암호 알고리즘의 선형 분석과 비슷한 방법으로 큰 범위에서는 통계적 특성을 이용한 공격에 속한다고 볼 수도 있다. 1988년 Meier와 Staffelbach에 의해 제안된 빠른 상관 공격은 기반 LFSR의 feedback 함수가 많지 않을 때와 10 이하와 같이 적은 weight 다항식의 곱, 차수가 너무 크지 않을 때 등은 매우 효율적인 공격법이다. 이후로도 많은 연구가 이루어져 메모리를 갖는 경우와 그렇지 않은 경우, 시간 제어가 되는 경우 등에 대한 연구가 이루어졌다.

3.1.4 선형 복잡도

수열의 선형 복잡도는 그 수열을 생성하는 가장 짧은 길이의 LFSR의 길이를 말하는데 이것은 Berlekamp-Massey 알고리즘을 비롯한 여러 알고리즘에 의해 쉽게 계산될 수 있다. 선형 복잡도가 너무 낮으면 공격자는 LFSR의 수열을 다시 만들어낼 수 있게 된다.

3.1.5 Maximum order 복잡도

Maximum order 복잡도는 주어진 비트 수열을 생성할 수 있는 가장 짧은 길이의 비선형 시프트 레지스터의 길이를 결정할 수 있다. 이러한 방법으로 수열의 처음 1, 2, 3, ...비트들에 대해 계산할 수 있다.

3.1.6 분할 정복(Divide-and-conquer) 공격

1979년 Rubin에 의해 제안된 divide-and-conquer 공격은 키의 일부분을 추측한 후에 공격을 하는 방법으로 이 나머지 부분을 전수 조사하는 방법보다 빠르게 키의 나머지 부분을 결정할 수 있다. 알고리즘 내부의 상호 연관성이 없는 변수가 발생할 경우 이 분석법에 취약할 수 있다. 그러므로 이 공격법에 내성을 갖기 위해서는 알고리즘 설계 시 확산 및 혼돈 효과를 잘 결합하여 사용하여야 한다.

3.1.7 Rekeying 공격

스트림 암호가 수시로 rekey되는 어플리케이션 들이 많이 존재한다. 키를 찾기 위해 이러한 rekeying을 분석하는 것이 가능할 수 있다.

3.1.8 부채널 공격

부채널 공격은 사용하는 기본 특성의 움직임을 분석하는 것으로 서로 다른 키나 내부 상태에서 서로 다른 시간이나 파워를 사용한다는 원리를 이용하여 공격하는 방법이다. 이 방법은 실행 시간, 전력 소비, 전자기파 등의 유출로 인한 것뿐만 아니라 에러 메시지나 위의 방법들의 결합에 의한 방법으로도 가능한 것으로 알려져 있다. 또한 위에서 제시한 것과 같은 수동적인 공격뿐만 아니라 fault 공격, 선택 모듈러 공격 등과 같은 능동적 공격도 존재한다. 뒤에서 이야기하겠지만 특히 하드웨어 설계 시에는 이와 같은 부채널 공격에 내성을 가질 수 있는 설계가 필요하다.

3.2 최근 스트림 암호 공격 기법

최근의 스트림 암호 공격법은 기존의 스트림 암호 공격 기법을 발전시킨 것들도 물론 있지만 이를 체계적으로 발전시킨다든지 대수적 공격과 같이 새로운 기법을 제안하는 방향으로 발전해 나아가고 있다. 스트림 암호의 공격 기법을 크게 세 가지로 분류하면 distinguishing 공격, prediction 공격, key recovery 공격으로 분류할 수 있다. 다음에서 이들에 대해 간단히 살펴보고 또한 최근 가장 연구가 많이 이루어지고 있는 공격법들을 소개한다.

3.2.1 구별(Distinguishing) 공격

키 스트림과 랜덤 스트림과의 구분할 수 있는 성질을 이용하여 공격하는 방법으로 통계적인 분석 방법 및 대부분의 고전적인 분석 방법들이 distinguishing 공격에 속한다. 이러한 공격 방법으로는 통계적 특성을 이용한 공격인 χ^2 분석법 등이 있다.

3.2.2 예측(Prediction) 공격

적당한 길이의 키 스트림을 사용하여 다음에 나오는 키 스트림을 예측하는 공격으로 이전 수열과 다음 수열의 연관성을 이용한 공격법이다. 이 방법의 공격법으로는 correlation 공격, divide-and-conquer 공격, guess-and determine 공격 등이 있다.

3.2.3 키 복구(Key recovery) 공격

키 스트림을 통한 구별 및 예측 공격 등을 이용하여 마스터키를 복구하는 공격이다. 이 방법으로는 rekeying 공격 등이 있다. 위의 공격들보다 키 복구 공격이 가장 강력한 공격법이며 그 다음이 예측 공격법이다. 키 복구 공격이 가능하면 예측 공격이 가능하고 예측 공격이 가능하면 구별 공격도 가능하다.

3.2.4 추측 결정(Guess-and-determine) 공격

추측 결정 공격은 선형 반복 관계에서 각 상태들의

선형 관계와 LFSR의 상태 값과 키 스트림 값과의 관계에서 생성되는 식을 이용하여 추측되지 않은 부분의 LFSR의 상태 값을 정함으로써 다음 키 스트림의 값을 알 수 있도록 하는 공격 방법이다. NESSIE에 제안된 SNOW의 공격에 사용되었다.

3.2.5 대수적 공격

대수적 공격은 암호 내부에 알려지지 않은 미지의 변수와 알려진 입·출력과의 관계인 대수적 방정식을 만들고 이를 분석하여 대수적 방정식을 선형화하여 내부 변수를 풀어 계산해냄으로써 공격하는 방법이다. 이러한 대수적 공격은 초기에 공개키 알고리즘인 HFE(Hidden Field Equation)에 적용되었고 이후 SERPENT, AES와 같은 대수적 성질을 가지는 블록 암호의 분석에 적용되었다. 초기에 이처럼 블록 암호의 공격에 초점이 맞추어져 있었으나 점차 적용이 용이한 LILI-128, E0, Toyocrypt 등의 스트림 암호 분석에 사용되기 시작하였으며 NESSIE에 제안된 SOBER의 공격이나 HELIX의 공격에도 사용되었다. LFSR에 기반한 스트림 암호는 하드웨어에서 효율적이고 수학적 평가를 할 수 있다는 장점이 있으나 여기서 소개하는 대수적 공격에 취약하다고 알려져 있다.

기존 대칭키 암호의 기본 분석법은 대부분 확률 분포의 비균일성을 이용한 확률론적 공격인데 반해 대수적 공격은 다음과 같은 특징을 가지고 있다.

- 주어진 알고리즘에 대한 대수적 방정식들에서 방정식의 개수가 변수의 개수보다 많은 과포화(overdefined) 상태일 경우 대수적 공격은 확률 1로 내부 변수(초기 상태 또는 키)의 값을 찾을 수 있다.
- 대수적 공격의 복잡도는 오직 내부 변수의 개수에만 의존하고 변수 값을 구하기 위한 복잡도는 변수 개수의 증가에 따라 지수적으로 증가하지 않는다.

위와 같은 특성으로 대수적 공격을 효과적으로 하기 위해서는 크게 두 가지로 문제를 나누어 생각할 수 있다.

- 낮은 차수의 대수적 방정식을 어떻게 효과적으로 많이 찾을 수 있는가
- 구해진 연립방정식을 어떻게 효율적으로 계산하여 해를 찾는가

낮은 차수의 대수적 방정식을 찾는 문제는 주어진 방정식의 선형 approximation을 찾는 문제로 최근에도 계속 연구되고 있는 분야이다. 구해진 연립방정식을 푸는 문제는 XL 알고리즘이나 Grobner 기저를 이용하여 푸는 방법 등이 알려져 있으며 계속 개선하기 위해 연구되고 있다. 대부분의 LFSR 기반 스트림 암호는 이러한 대수적 공격에 취약하며 최근 제안되는 스트림 암호는 대수적 공격을 피하기 위해 비선형성을 가지도록 설계되고 있다.

3.2.6 블록 암호 분석 기법을 이용한 방법

최근의 블록 기반 스트림 암호 알고리즘은 블록 암호 알고리즘에서 사용하는 암호 논리를 스트림 암호에 적용하여 사용하고 있는 추세이다. 따라서 일반적인 스트림 암호 분석 기법만으로는 최근의 블록 기반 스트림 암호 알고리즘을 분석하는 데에는 한계가 있다고 할 수 있다.

블록 암호 알고리즘을 분석하는 가장 강력한 공격법으로는 차분 분석(Differential cryptoanalysis)과 선형 분석(Linear cryptoanalysis)이 있다. 그러나 블록 암호 알고리즘의 차분 분석을 스트림 암호 알고리즘의 분석에 그대로 사용하는 것에는 문제점이 있다. 그 이유는 블록 암호 알고리즘의 분석에서는 공격이 선택 평문 공격이기 때문에 어떠한 함수의 입력과 출력을 모두 알고 있다는 가정 하에 공격을 진행하지만 스트림 암호에서는 키 스트림의 출력은 알지만 내부 변수를 사용하는 입력을 정확히 알기에는 어려움이 있다. 또한 내부 변수가 지속적으로 변화하기 때문에 특정한 차분 특성식을 계속 이용할 수 없다. 그러므로 블록 암호의 차분 분석법은 스트림 암호의 특성에 맞게 변형하여 사용하여야 한다. 차분 분석 기법은 블록 기반 스트림 암호 알고리즘의 분석으로는 가장 용이하고 효과적인 공격법이라 할 수 있다. S-box를 사용하는 스트림 암호에 적용하는 것은 매우 강력한 공격이 될 수 있다.

이 밖에도 블록 암호의 분석법을 이용하여 공격하는 기법으로는 마스터 키의 연관성이나 취약성을 이용한 연관키 공격이나 취약키 공격 등이 있으며 워드 단위 연산 스트림 암호 분석을 위한 SQUARE 공격은 연산의 취약성을 이용하여 공격하는 기법이다. 이러한 방법은 모두 그대로 사용하기 어렵고 스트림 암호 공격에 맞게 잘 수정하여야만 효과적인 공격이 이루어질 수 있을 것이다.

3.3 최신 암호 분석 실태

앞에서 소개한 SEAL 계열의 블록 기반 스트림 암

호는 SEAL과 TWOPRIME의 경우 구조적인 취약성으로 인해 분석이 되었으나 SCREAM이나 HELIX의 경우에는 아직까지 효과적인 분석 기법은 알려져 있지 않다. 다만 MAC 기능과 스트림 암호 기능을 동시에 제공하는 HELIX의 경우 MAC에 대한 위장 공격이 존재한다. SOBER 계열의 스트림 암호에는 SSC2, SNOW, TURING 등이 있는데 이들은 모두 공격법이 소개되어 분석되었다. SOBER의 경우 제안 당시부터 해독법이 소개되어 알고리즘을 계속 수정하였으나 그 취약점이 계속 발견되었다. SSC2의 경우 이론적인 분석 뿐만 아니라 실제적인 공격을 할 수 있는 알고리즘이 개발되어 완전히 공격되었다. SNOW 알고리즘의 경우에도 추측 결정(Guess-and-determine) 공격에 의해 분석되었다. TURING 스트림 암호도 선택 초기치 공격으로 분석되었다. MULTI-S01, MUGI 등과 같은 PANAMA 계열의 스트림 암호의 경우에는 현재까지 효율적인 공격 기법은 알려져 있지 않다. 다만 해쉬 기능을 동시에 제공하는 PANAMA의 경우 해쉬 함수에 대한 충돌 공격으로 분석되었다. 기타 RC4, BEEPBEEP, RABBIT, VMPC, HC-256 등의 스트림 암호 중 RC4를 제외하고는 최근에 제안되어 많은 분석이 이루어지지 않은 실정이다.

Ⅳ. 스트림 암호의 표준화 동향

국가가 주도하던 암호 산업이 차츰 민간 주도의 공개된 암호 산업으로 바뀌면서 개개인의 프라이버시를 위한 민간 주도의 암호 공모 사업이 전 세계적으로 이루어졌다. 미국은 그동안 전 세계적으로 가장 널리 사용하던 DES(Data Encryption Standard)를 대체하기 위하여 블록 암호에 대한 공모 사업을 시행했고 이의 결과로 2000년 AES를 차세대 블록 암호로 선정하였다. 이에 유럽에서는 제 5차 IST(Information Society Technologies) 프로그램의 일환으로 2000년 1월 NESSIE 프로젝트를 기획하여 수행하게 되었다. 이러한 NESSIE 프로젝트는 3년여의 공모 기간과 평가 기간을 거쳐 2003년 최종 알고리즘을 선정하였고 여기에는 블록 암호뿐만 아니라 스트림 암호, 공개키 암호, 서명 알고리즘, 해쉬 함수 등 암호 분야의 다양한 알고리즘이 선정되었다. 비슷한 시기에 일본에서는 전자정부 구축을 위한 암호 원천 기술 연구를 위해 IPA를 중심으로 암호 기술 평가 위원회를 구성하여 이용 가능한 암호 기술에 대한 목록을 작성하고 이에 대한 안전성, 효율성 등의 특징을 기술하여 역시 2003년에 최종 알고리즘을 선정하였다. 우리나라에서도 이런 세계의 추세에 발맞추어 차세대 암호에 대한 많은 논의와 연구가 이루어지고 있으며 USN이나 RFID 환경에서 사용할 수 있는 암호 설계를 위한 연구에 매진하고 있는 실정이다. 그동안 표준화와는 관계가 없어 보였던 스트림 암호도 이러한 맥락 속에서 꾸준히 표준화와 관련되어 연구되어 오고 있으며 앞으로는 표준화된 스트림 암호를 이용하여 통신을 할 수 있을 것으로 생각된다. 다음에서 각국의 표준화 현황과 앞으로의 움직임에 대해 살펴본다.

4.1 국내 표준화 동향

국내에서는 1998년 SEED, KCDSA, HAS-160과 같은 암호 알고리즘이 표준화되어 사용되고 있다. 그 후에 ECKCDSA가 표준화 되었으며 2004년에는 국내에서 개발된 블록 암호인 ARIA가 표준화 되었다. 우리나라에서도 암호 알고리즘에 대한 표준화와 차세대 암호 알고리즘의 개발에 대한 논의가 2000년대에 들어 활발해지면서 여러 우여곡절 끝에 2004년 차세대 암호 알고리즘 개발이 시작되었다. 이 프로젝트에서는 USN, RFID 등의 결정되지 않은 미래 환경을 고려하여 초고속, 초경량의 암호 알고리즘 개발과 이들의 체계적인 평가를 목적으로 블록 암호, 스트림 암호, 해쉬 함수, 그리고 공개키 암호 프리미티브 등이 개발되고 있고 평가안이 개발되고 있는 실정이다.

4.2 국제 표준화 동향

4.2.1 NESSIE

유럽에서 추진된 NESSIE 프로젝트는 제 5차 R&D 프로젝트 프레임워크의 일환으로 전자서명, 무결성과 암호화 등에 적합한 새로운 암호 알고리즘을 개발하기 위하여 성립된 프로그램으로 2000년 1월에 시작되었다. 이 프로젝트는 선정된 암호 프리미티브를 널리 보급하자는 것과 이를 통해 공개 토론의 장을 만들어보자는 의도를 가지고 있다. 미국의 AES(Advanced Encryption Standard)가 블록 암호 알고리즘만을 공모한데 반해 NESSIE는 블록 암호, 공개키 암호, 스트림 암호, 해쉬 함수, MAC, 전자서명 등 정보보호 전반에 걸친 핵심 알고리즘을 모두 다루고 있다. 이 프로젝트에는 유럽의 여러 유명 전문가가 참여하였으며 벨기에(COSIC, UCL), 프랑스(ENS), 영국(RH),

이태리(FUB), 독일(Siemens), 노르웨이(U. Bergen), 이스라엘(Technion) 등 7개국 8개 유명 대학이나 연구소의 세계적인 암호 설계 및 분석 전문가가 참여하였다. NESSIE는 2000년 3월 암호 프리미티브 후보 모집을 시작하여 2000년 9월 까지 공개 모집을 거쳐 제 1 단계를 거치고 2000년 11월에 제 1차 NESSIE workshop을 개최하였으며 2001년 9월에 제 2차 NESSIE 프로젝트 workshop을 개최하고 2002년 2월 1차 평가를 마치고 선정 작업을 추진할 계획이었으나 이를 앞당겨 2001년 9월 24일에 총 42개의 후보 중 24개의 알고리즘을 1차로 선정하였다. 또한 2002년 5월부터 표준화 추진을 하고 있으며 2002년 11월 제 3차, 2003년 2월에 4차 workshop을 거쳐 2003년 3월에 최종 알고리즘을 선정 발표하였다. 스트림 암호의 경우 BMGL, Leviathan, LILI-128, SNOW, SOBER-t16, SOBER-t32 등 6개의 알고리즘이 제안되었으나 최종적으로는 어느 스트림 알고리즘도 선정되지 못하였다. 이 각각의 알고리즘들은 안전성과 효율성 면에서 면밀히 검토, 분석되어 최종 선정에서 제외되었다.

4.2.2 CRYPTREC

일본에서는 2003년 전자 정부의 구현을 목표로 이에 필요한 보안 기술을 확보하기 위하여 경제 산업성으로부터 위탁을 받아 정보 처리 진흥 사업 협회(Information Promotion Association: IPA)의 보안 센터(Information Security Center: ISC)와 통신방송기구(Telecommunication Advancement Organization: TAO)가 주관이 되어 CRYPTREC을 진행하였다. 2000년 6월과 7월에 걸쳐 1차 공모를 받아 2000년 10월 까지 서류 심사를 한 결과 블록 암호의 경우에는 제안된 13개의 알고리즘 중 10개만을 상세 평가하기로 결정하였다. 일본은 당초 ISO/IEC JTC 1/SC 27 동경 회의에서 차기 회의인 오슬로 회의에 CRYPTREC의 평가 결과로 선정된 알고리즘들을 제안하려 하였으나 시간이 부족하여 이를 철회하였다. CRYPTREC은 2001년 8월부터 9월에 걸쳐 일반 공모 및 전자서명과 SC27 의뢰의 특정 평가를 실시하였다. 이를 통해 지속적인 암호 기술에 대한 평가와 조사를 실시하며 전자 정부 실현을 위한 가능 기술에 대한 안전성, 구현성 등의 특징을 도출하였다. 공모에 응한 알고리즘은 19개의 블록 암호를 비롯해 인증, 암호, 서명, 키 분배 등의 여러 분야에 모두 48개의 알고리즘이 응모하였다. 이 밖에도 SHA-1과 같이 널리 알려진 암호 알고리즘도 평가 대상에 포함시켜 최종 알고리즘을 선정하였다. CRYPTREC은 2002년 4월의 CRYPTREC report 2001을 거쳐 2003년 3월 최종 선정 알고리즘을 발표하였다. 스트림 암호의 경우 1차 평가 대상 알고리즘으로는 MULTI-S01과 TOYO-CRYPT-HS1이 있으며 2차 평가 대상 알고리즘은 TOYOCRYPT- HS1이 제외되고 MULTI-S01, C4-1, FSAngo, MUGI, RC4 등을 대상으로 하였다. 위의 여러 알고리즘 중 MUGI, MULTI-S01, RC4가 선정되었으며 RC4의 경우 128 비트 이상으로 제한하였다.

4.2.3 ISO/IEC

ISO/IEC JTC 1/SC 27에서도 많은 암호 알고리즘들이 표준화되고 있는데 스트림 암호의 경우에도 일본에서 CRYPTREC의 알고리즘을 표준화하려는 노력이 있었으며 일본의 MULTI-S01 등의 경우 표준화 되어 있다. 최근에는 MUGI, SNOW2 등의 표준화 움직임이 활발한 실정이다. 우리나라에서도 꾸준한 노력으로 공개키 분야와 프로토콜, 블록 암호 등에는 표준 암호를 제안하였으나 스트림 암호의 경우 노력이 필요한 실정이다.

V. 스트림 암호 설계 요구사항

스트림 암호 설계 시 요구되는 기본적인 사항으로는 안전성과 효율성을 우선적으로 생각할 수 있다. NESSIE나 CRYPTREC 등에서도 논의가 되었고 일반적으로 스트림 암호의 설계에 고려되어야 할 것들에 대해 다음 1, 2 절에서 살펴보도록 하자. 그러나 알고리즘의 설계에는 학문적인 안전성이나 실제적인 효율성 이외에도 고려되어야 할 문제가 많이 있다.

알고리즘의 설계는 단순하고 명확한 원칙을 가지고 행해져야 하며 그 알고리즘이 가지는 특허 및 지적재산권에 대해서도 자세히 언급이 되어져야만 한다. 새로운 알고리즘의 경우 새로운 프리미티브와 좀 더 일반적인 공격법에 대해 연구가 되어야 하고 안전성에 영향을 주는 키의 길이도 두 가지 이상이 가능하도록 하거나 하여 유연성을 주는 것도 고려할 사항이다.

5.1 안전성 측면

암호 알고리즘의 경우 가장 중요하게 다루어져야 하는 부분이 안전성을 고려하는 부분이다. 안전성 분

석을 크게 둘로 나누면 수학적 안전성과 통계적 안전성으로 분석한다. 수학적 안전성을 위해 스트림 암호를 구성하는 기반함수의 특성 및 이론적 근거를 제시하고 이를 검증받아야 한다. 특히, 키 스트림 생성기의 경우 통계적 안전성을 위해 기존에 알려진 여러 가지 통계적 모델을 고려해 키 스트림 생성기를 분석하여야만 한다. 스트림 암호 분석의 경우 통상적으로 known plaintext 공격을 사용한다. 이는 이미 많은 양의 키 스트림이 공개되었다는 것을 가정한다. 앞에서도 설명한 바와 같이 안전성 분석은 크게 다음과 같은 공격으로 나누어볼 수 있다.

- 구별 공격
- 예측 공격
- 키 recovery

위의 세 공격 중 키 recovery 공격이 가장 강력한 공격법이며 예측 공격이 가능하면 구별 공격이 가능해진다. 위에서 설명한 많은 스트림 암호의 공격법은 모두 크게 이 세 범주에 속하며 안전성을 고려할 때에는 모든 알려진 공격법에 내성을 가지도록 설계되어야 한다.

스트림 암호의 안전성을 결정하는 중요 요소로 키의 길이를 들 수 있는데 이는 스트림 암호가 사용되는 응용에서 요구하는 기준을 만족하는 수준이어야 한다. 최근에는 일반적인 사용에서 128 비트 이상을 요구하고 있는 실정이다.

통계적 안전성을 위해서는 미국의 NIST와 NESSIE에서 제시한 통계적 테스트를 이용하여 난수성을 검증하여야 한다. 또한 수열의 전파 특성이나 쇄도 효과 등도 분석하여 검증하여야 한다.

5.2 효율성 측면

암호 알고리즘의 경우 안전성과 함께 고려되어야 할 중요한 요소가 바로 효율성이다. 최근의 암호는 그 암호 알고리즘을 사용하는 응용에 따라 안전성에 중점을 두는 지 아니면 속도가 중요한 지, 아니면 구현 능력이 중요한 지 등이 달라진다. 그러므로 효율성을 구체적으로 고려하기 위해서는 각 알고리즘의 사용에 대해 확실히 알아볼 필요가 있다. 다음에서는 효율성을 고려하는 일반적인 기준에 대해 설명한다.

효율성은 크게 소프트웨어와 하드웨어에서의 효율성으로 나눌 수 있다. 소프트웨어에서는 처리 속도의 효율성이 중요하며 구현이 용이한 코드 사이즈를 가지고 있는지도 중요한 사항이다. 또한 메모리의 사용에 문제가 없는지도 고려해야 한다. 하드웨어에서는 전력 소모량과 칩 사이즈, 그리고 처리 속도 등의 기준을 부합하여야 한다. 좀 더 구체적인 환경이 주어지면 요구 사항도 이에 따라 구체적으로 정해지고 평가되어야 한다.

Ⅵ. 결 론

스트림 암호의 경우 1960년대 까지 군사 업무 등에 많이 사용되었으나 다음과 같은 여러 가지 이유로 많은 부분 블록 암호로 대체되었다. 블록 암호는 스트림 암호에 비해 보다 일반적이며 안전성이 일반적으로 강한 것으로 알려져 있다. 또한 메모리가 싸고 현대 컴퓨터에서 연산이나 저장이 블록 단위로 이루어지므로 효율성의 측면에서 뒤떨어지지 않는다.

스트림 암호는 유럽의 2세대 이동 통신 시스템인 GSM에 암호 알고리즘으로 A5 알고리즘이 사용되었으며 3세대 IMT-2000 시스템에서도 KASUMI 블록 암호 알고리즘을 이용한 f9 스트림 암호 함수가 권고되었다. 또한 wireless 네트워킹에서도 RC4가 사용되고 bluetooth에 E0가 사용되는 등 꾸준히 그 사용이 있어 왔다.

현재 국내에서 개발되고 있는 스트림 암호의 경우 차세대를 겨냥한 알고리즘의 개발이므로 미래의 환경을 고려해 보는 것도 좋은 전망이 될 수 있다. 최근 국내를 비롯해 세계적으로 RFID를 이용한 여러 응용들이 시험되고 있으며 USN 등의 미래 환경이 활발히 논의되고 있다. 이런 시점에서 그 동안 쇠퇴 일로에 있던 스트림 암호가 인기를 얻던 블록 암호와 차별화되어 중요성이 인식되는 이유는 다음과 같다.

- Low-end 하드웨어 구현의 용이성
- 빠른 암호화 속도
- 적은 입·출력 delay
- 다양한 입력을 소화할 수 있는 더 단순한 프로토콜

그러나 이러한 이유들은 하드웨어가 점점 커지고 싸지며 모든 응용을 소프트웨어 적으로 처리할 수 있으며 보통은 암호화가 시스템 전체 속도에 큰 문제가 아닌 점 등을 고려하면 좀 더 면밀한 검토와 노력이 필요하다.

참 고 문 헌

[1] N. T. Courtois, W. Meier, "Algebraic attacks on stream ciphers with linear fe-

edback", Eurocrypt 2003, LNCS 2656, pp 345-359, Sringer, 2003
[2] NESSIE security report, http://www.cosic.es-at.kuleuven.ac.be/nessie/,2003
[3] CRYPTREC report 2001, http://www.ipa.goj- p/security/,2002
[4] 강주성 외, "현대 암호학", 국가보안기술연구소, 2000
[5] A. Menezes, P. Oorschot, S. Vanstone, "Hand-book of applied cryptography", CRC Press Inc., 1997
[6] S. Goldwasser, M. Bellare, "Lecture notes on cryptography", http://www.cse.ucsd.edu/users/ mihir

〈著 者 紹 介〉

류 희 수 (Heuisu Ryu)
정회원

1990년 2월: 고려대학교 수학과 이학사
1992년 2월 : 고려대학교 수학과 이학석사
1999년 5월 : Johns Hopkins Univ. 수학과 이학박사
2000년 7월~2003년 8월: 한국전자통신연구원 정보보호기반연구팀 팀장
2003년 9월~현재: 경인교육대학교 수학교육과 전임강사