Journal of Korea Multimedia Society Vol. 11, No. 8, August 2008(pp. 1111-1120)

셀룰러 오토마타 기반 블록 암호에 대한 안전성 분석

류한성[†], 이제상[††], 이창훈[†††], 홍석희[††††]

요 약

셀룰러 오토마타(CA: cellular automata)의 특징 중에서 확산과 국소적인 상호 작용(Local Interaction)은 암호시스템을 설계하는데 적합하여 암호 알고리즘, 의사난수 생성기를 비롯한 암호시스템의 설계 논리로 활용되고 있다. 국내에서는 2002년 CA 기법을 이용한 128 비트 블록 암호(CAB1)가 처음으로 소개되었고, CEC'04에서는 가역 CA를 이용한 64 비트 블록 암호(CAB2)가 제안되었다. 본 논문에서는 두 알고리즘이 각각 차분 공격과 통계 분석에 취약함을 보인다. 먼저, $2^{31.41}$의 선택 평문을 이용하여 $2^{13.41}$의 공격 복잡도를 갖는 CAB1에 대한 차분 공격을 소개한다. 그리고 CAB2는 제안 논문에서 224 비트의 안전성을 갖는다고 제안되었지만, CAB2의 키가 균일 성질을 만족해야만 하는 취약점을 이용하여 184 비트의 안전성만을 가짐을 보인다. 본 논문에서 제안하는 공격 결과는 이 CA 기반 블록 암호들에 대한 첫 번째 분석 결과이다.

Cryptanalysis of Two Block Ciphers based on Cellular Automata

Han Seong Ryu[†], Je Sang Lee[††], Chang Hoon Lee[†††], Seok Hie Hong[††††]

ABSTRACT

Cellular automata(CA) is often applied to design cryptosystems because it has good diffusion and local interaction effects. Recently, a 128-bit CA-based block cipher, called CAB1, and a 64-bit reversible CA-based block cipher, called CAB2, were proposed in KMMS'02 and CEC'04, respectively. In this paper, we introduce cryptanalytic results on CAB1 and CAB2. Firstly, we propose a differential attack on CAB1, which requires $2^{31.41}$ chosen plaintexts with about $2^{13.41}$ encryptions. Secondly, we show that CAB2 has a security of 184 bits using the statistical weakness. Note that the designers of CAB2 insist that it has a security of 224 bits. These are the first known cryptanalytic results on them.

Key words: Block Cipher(블록 암호), Cellular Automata(셀룰러 오토마타), Differential Attack(차분 공격)

1. 서 론

CA는 스스로 조직화하고 재생산할 수 있는 모델로서, 국소적 상호작용을 통하여 모든 상태가 동시에 갱신되는 유한상태머신이다. 이것은 Von Neumann에 의해 처음 소개되었으며[1], Wolfram에 의해 처음으로 암호학에 응용되었다[2]. CA의 특징 중에서 확산과 국소적인 상호 작용은 암호시스템을 설계하는데 적합하여 LFSR의 대안으로 제시되었으며, 부울 방정식의 해법, 의사난수 생성기, 암호 알고리즘 설계 등과 같은 다양한 응용분야에서 사용되고 있다. CA는 AND, OR, NOT, XOR과 같은 단순한 연산을

※ 교신저자(Corresponding Author) : 홍석희, 주소 : 서울 성북구 안암동 5가(136-713), 전화 : 02)3290-4894, FAX : 02)928-9109, E-mail : hsh@cist.korea.ac.kr
접수일 : 2008년 5월 15일, 완료일 : 2008년 6월 10일
[†] 준회원, 고려대학교 정보경영공학전문대학원 석사과정 (E-mail : ybr251@cist.korea.ac.kr)
[††] 준회원, 고려대학교 정보경영공학전문대학원 박사과정 (E-mail : jslee@cist.korea.ac.kr)
[†††] 정회원, 고려대학교 정보보호연구원 연구교수 (E-mail : crypto77@cist.korea.ac.kr)
[††††] 정회원, 고려대학교 정보경영공학전문대학원 조교수
※ 본 연구는 "지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-(C1090-0801-0025))

이용하여 상태를 갱신하며, 특히 비선형 성질을 만족하는 법칙을 이용한 암호 알고리즘이 국내에서 다수 발표되었다[3-5]. 그러나 확산 효과가 좋지 않은 구조적 취약점 때문에 대부분의 기 제안된 암호 알고리즘이 분석되었다[6,7].

2002년 한국 멀티미디어 학회 논문지와 CEC'04에서 CA 기반 블록 암호와 가역 CA 기반 블록 암호가 각각 제안되었다[3,8]. 본 논문에서는 논문 전개상의 편의성을 위하여 [3]에서 제안된 블록 암호를 CAB1라고 표기하고, [8]에서 제안된 블록 암호를 CAB2라고 표기한다. 본 논문에서는 CAB1과 CAB2에 대한 안전성 분석 결과를 소개한다.

첫 번째, CAB1의 라운드 함수는 일반적인 블록 암호의 라운드 함수가 S-box로 구성되는 것과는 다르게 CA의 비선형 법칙을 이용한다. 특히, 이 블록 암호의 가장 큰 특징은 라운드 함수의 연산이 라운드 키의 비트값에 종속적으로 결정된다는 것이다. 공격자는 라운드 키를 알지 못하기 때문에, 라운드 함수의 결과값을 예측하기 어렵다. [3]에서는 CAB1에 대한 안전성 분석 결과로서 CAB1의 축소 버전에 대한 차분 분석 결과를 제시하였다. 즉, F함수의 모든 입·출력 차분에 대한 차분 분포를 계산하는 것이 불가능하기 때문에 입력과 출력이 4 비트인 F함수로 축소하여 분석하였다. 비록 축소된 F함수의 분석이 전체 F함수에 대한 안전성을 보장하지는 못하지만, 축소된 F함수의 차분 분포가 다소 고른 분포를 나타내기 때문에 차분 공격에 대하여 안전하다고 주장하였다. 하지만 본 논문에서는 CAB1이 차분 공격에 취약함을 보인다[9]. 이 공격은 $2^{31.41}$의 선택 평문을 이용하여 $2^{13.41}$의 공격 복잡도를 갖는다.

두 번째, CAB2는 반지름이 2인 3개의 가역 CA와 반지름이 3인 1개의 가역 CA로 구성된 64 비트 블록 암호로서 키 길이는 224 비트이다. CAB2는 전수조사보다 더 좋은 공격이 없다고 제안되었지만, CAB2의 출력값은 0과 1의 개수가 동일한 균일 성질을 만족해야 하고, 그러기 위하여 키가 균일 성질을 만족해야 하기 때문에 $\log_2\left[\binom{32}{16}\times\binom{32}{16}\times\binom{32}{16}\times\binom{128}{64}\right]=211.7$ 비트의 안전성을 갖게 된다. 게다가, 단순히 키보드를 이용하여 키가 입력될 경우에는 184 비트의 안전성을 갖게 된다.

본 논문의 구성은 다음과 같다. 2 장에서는 CA에 대한 기본적인 내용을 소개하고, 3 장에서는 CAB1에 사용되는 표기법과 CAB1을 간략히 설명하고 4 장에서는 CAB2에 대해 간략히 설명한다. 5 장에서는 F함수의 차분 성질을 이용하여 CAB1에 대한 차분 공격을 제시하고, 6 장에서는 CAB2에 대한 취약점 분석을 한다. 마지막으로 7 장은 본 논문의 결론이다.

2. 셀룰러 오토마타

2.1 1차원 CA

1차원 CA는 1차원 배열에서 1차원 배열로 갱신되는 CA를 말한다. 국소적 상호 작용이 세 개의 셀, 즉 자기 자신과 인접한 두 셀에 의해 갱신되는 반지름이 1인 CA이다. CA를 설명하기 위하여 다음 기호들이 정의된다.

$$s_i^{t+1}=f(s_{i+r}^t,\cdots,s_{i+1}^t,s_i^t,s_{i-1}^t,\cdots,s_{i-r}^t)$$

- t : 시간단계
- r : 반지름
- s_i^t : 시간 t에서 i번째 셀의 위치
- s_i^{t+1} : 시간 $t+1$에서 i번째 셀의 위치

반지름이 1인 CA에 대한 상태전이 함수(state - transition function)는 그림 1과 같다.

GF(2)상에서 반지름이 1인 CA의 상태전이 함수는 i번째 셀 s_i^t을 이웃한 2개의 셀(s_{i-1}^t, s_{i+1}^t)과의 상호작용을 통하여 s_i^{t+1}로 갱신한다. 상호작용에 이용되는 상태전이 함수를 법칙(Rule)이라 명하며, 법칙은 부울 함수($Z_{2^3}\rightarrow Z_2$)로 정의할 수 있다. GF(2)상의 서로 이웃한 3개의 셀이 가지는 상태의 경우의 수는 2^3이며, 2^{2^3}의 상태전이 함수가 존재한다. 갱신 법칙의 두 가지 예로서 법칙 90과 150은 표 1과 같이 정의되며, 부울 함수로 표현하면 표 2와 같다. 여기서

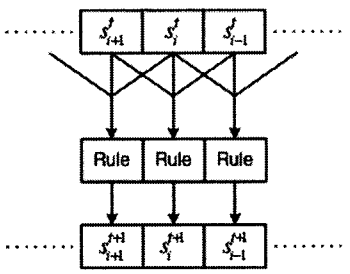

그림 1. 반지름이 1인 CA에 대한 상태전이 함수

표 1. 갱신 법칙

	111	110	101	100	011	010	001	000
법칙 78	0	1	0	0	1	1	1	0
법칙 90	0	1	0	1	1	0	1	0
법칙 92	0	1	0	1	1	1	0	0
법칙 150	1	0	0	1	0	1	1	0

표 2. 상태전이 함수

법칙	논리 함수
78	$s_i^{t+1} = (\overline{s_{i-1}^t} \bullet s_{i+1}^t) \oplus (s_i^t \bullet \overline{s_{i+1}^t})$
90	$s_i^{t+1} = s_{i-1}^t \oplus s_{i+1}^t$
92	$s_i^{t+1} = (\overline{s_{i-1}^t} \bullet s_i^t) \oplus (s_{i-1}^t \bullet \overline{s_{i+1}^t})$
150	$s_i^{t+1} = s_{i-1}^t \oplus s_i^t \oplus s_{i+1}^t$

s_i는 i번째 셀의 상태값, $\bullet$는 AND 연산, $\oplus$는 XOR 연산을 각각 의미한다. 법칙의 이름은 모든 입력값에 대한 출력값을 순차적으로 나열하여 2진수로 표현한 다음 2진수를 10진수로 변환하여 표기한다. 따라서 법칙 90과 150은 모든 입력값에 대한 출력값을 비트열로 표현하면 각각 "01011010"과 "10010110"이며 10진수로 표현할 경우 90과 150이다.

반지름이 2인 CA의 상태전이 함수는 i번째 셀 s_i^t을 이웃한 4개의 셀(s_{i-2}^t, s_{i-1}^t, s_{i+1}^t, s_{i+2}^t)과의 상호작용을 통하여 s_i^{t+1}로 갱신하며, 법칙은 부울 함수 ($Z_{2^5} \rightarrow Z_2$)로 정의할 수 있다. 서로 이웃한 5개의 셀이 가지는 상태의 경우의 수는 2^5이며, 2^{2^5}의 상태전이 함수가 존재한다.

CA는 셀들에 적용되는 법칙에 따라서 다음과 같이 분류한다. 모든 셀들의 법칙이 XOR 논리로만 이루어진 CA를 linear CA라고 하고, 셀들의 법칙이 XOR/XNOR의 조합으로만 이루어진 CA를 additive CA라고 하고, 셀들의 법칙이 AND-OR 논리로 이루어진 CA를 nonadditive CA라고 한다. 모든 셀이 하나의 같은 논리로만 이루어진 CA를 uniform CA라고 하고, 2개 이상의 논리로 이루어진 CA를 Hybrid CA라고 한다. CA를 구성하는 양 끝셀의 경계조건은 다음과 같이 분류한다. 경계조건이 "0"으로 가정하는 NBCA(Null Boundary CA)와 경계조건이 양 끝셀을 서로 연결되는 PBCA(Periodic Boundary CA) 등으로 분류한다. 그림 2는 경계조건이 "0"인 Hybrid NBCA의 한 예이다.

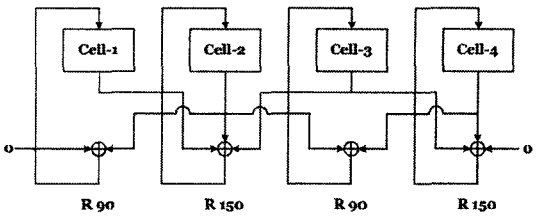

그림 2. Hybrid NBCA

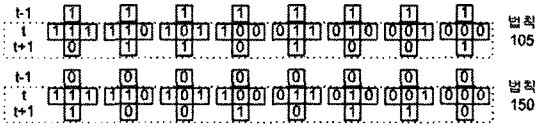

그림 3. 가역 법칙 105/150

2.2 가역 CA

셀룰러 오토마타(CA)가 모든 현재 상태에 대해서 이전 상태가 유일하게 존재할 때, 가역이라고 말한다. CA에서 갱신되는 상태간의 관계를 함수라고 했을 때, 가역은 1대1 함수를 의미하고, 대부분의 CA는 비가역이다. 간단한 CA에 대하여, 적은 개수의 법칙만이 가역의 성질을 갖고 있다는 것이 알려져 있다. 예를 들면, 반지름이 1인 256개의 CA중에서 6개만이 가역 CA이다.

CAB2는 Wolfram 등에 의하여 제안된 가역 CA 법칙을 이용하고, CAB2에서 이용하는 CA 법칙은 아래의 식과 같이 현재 상태와 이전 상태에 의존하여 갱신한다.

$$s_i^{t+1} = f(s_{i+r}^t, \cdots, s_{i+1}^t, s_i^t, s_i^{t-1}, s_{i-1}^t, \cdots, s_{i-r}^t)$$

예를 들면, 반지름이 1인 CA 법칙은 그림 3과 같이 2개의 보수 법칙으로 구성한다. 여기에서 시간 $t-1$에서 셀의 값이 1이면 법칙 105를 이용하고, 0이면 법칙 150을 이용한다.

3. CAB1 블록 암호

본 장에서는 2002년 한국 멀티미디어 학회 논문지에 이준석 등이 제안한 CA 기법을 이용한 CAB1에 대하여 간략히 소개한다. 본 논문에서는 CAB1에 대한 차분 공격을 이용할 때 키는 독립이라고 가정하기 때문에, 키 생성과정은 소개하지 않는다. CAB1을 소개하기에 앞서 본 논문 전반에 걸쳐 사용될 표기법을

소개한다.

3.1 표기법

본 논문에서 사용될 표기법은 다음과 같다.

- $P(=P_0 \| P_1 \| P_2 \| P_3)$: 128 비트 평문
- $C(=C_0 \| C_1 \| C_2 \| C_3)$: 128 비트 암호문
- B_i : i 번째 32 비트 입력 워드 $(0 \le i \le 3)$
- B_i^j : j번째 라운드에서 i 번째 32 비트 입력 워드 $(1 \le j \le 16)$
- $B_{i,k}^j$: j번째 라운드에서 i 번째 32 비트 입력 워드의 k번째 비트 $(0 \le k \le 31)$
- K_0, K_1, K_2, K_3 : 초기 화이트닝 32 비트 키
- K_{4j+m} : j번째 라운드에서 m번째 사용되는 32 비트 키 $(1 \le j \le 16,\ 0 \le m \le 3)$
- $K_{68}, K_{69}, K_{70}, K_{71}$: 최종 화이트닝 32 비트 키
- ΔP_i : 평문 P_i와 P_i'의 XOR 차분
- ΔC_i : 암호문 C_i와 C_i'의 XOR 차분
- $P_{i,k}$: P_i의 k번째 비트$(0 \le k \le 31)$
- $C_{i,k}$: C_i의 k번째 비트$(0 \le k \le 31)$
- $\Delta P_{i,k}$: 평문 P_i와 P_i'의 k번째 비트의 XOR 차분 $(0 \le k \le 31)$
- $\Delta C_{i,k}$: 암호문 C_i와 C_i'의 k번째 비트의 XOR 차분$(0 \le k \le 31)$
- $\Delta B_{i,k}^j$: B_i^j와 $B_i^{j'}$의 k번째 XOR 비트의 차분 $(0 \le k \le 31)$
- $LRot(B_i^j, K_{4j})$: j번째 라운드 키 K_{4j}의 최하위 4 비트에 의존한 B_i^j의 왼쪽 순환 이동$(1 \le j \le 16)$

3.2 CAB1의 전체 구조

CAB1은 일반화된 Feistel 구조를 가지고 16 라운드로 구성되며 전체 구조는 그림 4와 같다. F함수는 32 비트 입력 받아 32 비트 출력하는 함수이다. 편의상 P_i와 C_i는 각각 B_i^0와 B_i^{18}로 표기법을 변경한다.

CAB1 알고리즘은 초기 화이트닝 과정(IW), 16 라운드 암호화 과정, 최종 화이트닝 과정(FW)으로 구성된다.

[초기 화이트닝 과정]

128 비트 평문은 128 비트 초기 화이트닝 키와 32 비트 워드 단위로 XOR 연산을 수행한다.

[16 라운드 암호화과정]

16 라운드 암호화 과정은 32 비트 워드 단위로 수

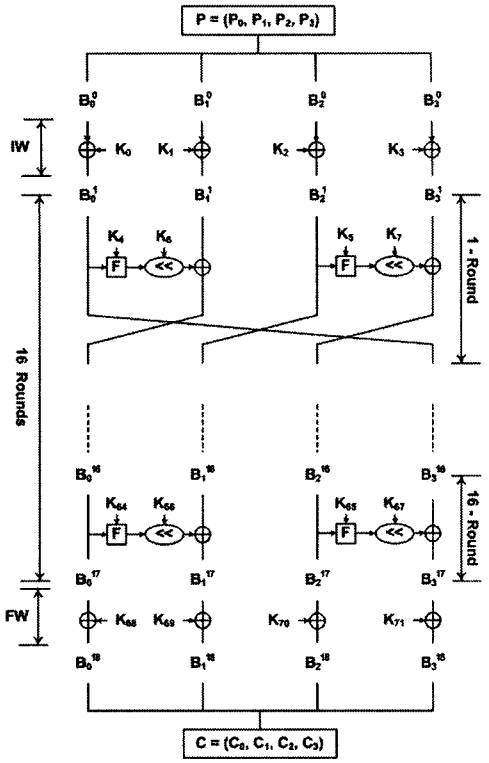

그림 4. CAB1

행되며, 라운드 키 128 비트를 이용하여 라운드 함수에 의하여 갱신된 워드는 이웃한 워드에 영향을 주도록 설계되었다. 4 개의 32 비트 워드가 모두 갱신되면 스왑과정을 거치며, 위와 동일한 과정에 의하여 16 라운드 암호화 과정을 수행한다. 단, 마지막 라운드는 스왑과정이 없다.

[최종 화이트닝 과정]

16 라운드 암호화 과정을 수행한 128 비트 중간 변수는 최종 화이트닝 키와 XOR 연산을 수행한 후에 암호문으로 출력한다.

3.3 라운드 함수

j번째 라운드 함수는 다음과 같은 과정을 통하여 암호화하고, F함수는 표 3과 같이 법칙 78과 법칙 92를 적용하는 1차원 Hybrid NBCA를 이용한다.

$$B_0^{j+1} = LRot(F(B_0^j, K_{4j}), K_{4j+2}) \oplus B_1^j$$
$$B_1^{j+1} = B_2^j$$
$$B_2^{j+1} = LRot(F(B_2^j, K_{4j+1}), K_{4j+3}) \oplus B_3^j$$
$$B_3^{j+1} = B_0^j$$

표 3. $F(B_0^j, K_{4j})$와 $F(B_2^j, K_{4j+1})$

$F(B_0^j, K_{4j})$
{
$B_{0,-1}^j = 0,\ B_{0,32}^j = 0$
for $(k=0;\ k<32;\ k+1)$
{
if $(K_{4i,k}=0)$
$B_{0,k}^* = (\overline{B_{0,k-1}^j} \cdot B_{0,k+1}^j) \oplus (B_{0,k}^j \cdot \overline{B_{0,k+1}^j})$
else
$B_{0,k}^* = (\overline{B_{0,k-1}^j} \cdot B_{0,k}^j) \oplus (B_{0,k-1}^j \cdot \overline{B_{0,k+1}^j})$
}
return B_0^*
}

$F(B_2^j, K_{4j+1})$
{
$B_{2,-1}^j = 0,\ B_{2,32}^j = 0$
for $(k=0;\ k<32;\ k+1)$
{
if $(K_{4j+1,k}=0)$
$B_{2,k}^* = (\overline{B_{2,k-1}^j} \cdot B_{2,k+1}^j) \oplus (B_{2,k}^j \cdot \overline{B_{2,k+1}^j})$
else
$B_{2,k}^* = (\overline{B_{2,k-1}^j} \cdot B_{2,k}^j) \oplus (B_{2,k-1}^j \cdot \overline{B_{2,k+1}^j})$
}
return B_2^*
}

4. CAB2 블록 암호

본 장에서는 CEC'04에서 제안된 CAB2에 대한 알고리즘과 키 생성 방법에 대하여 간략히 소개한다.

4.1 라운드 함수

CAB2는 64 비트 블록 암호로서 224 비트 키를 이용한다. 16 라운드로 구성된 CAB2의 각 라운드 함수는 가역 CA 기법을 이용한 CA_L, CA_R, CA_C, CA_S로 구성되어 있으며, 알고리즘은 그림 5와 같다.

4.2 라운드 알고리즘

각 라운드 알고리즘은 라운드 초기 데이터 과정, 라운드 연산과정, 다음 라운드 초기 데이터 과정으로 나누어진다.

[라운드 초기 데이터 과정]

각 라운드는 2개의 64 비트 초기 데이터(C_{0init}, C_{1init})로 시작한다. 첫 라운드에서 C_{0init}는 의사난수 생성기에 의해 생성된 데이터이고, C_{1init}는 평문이다. 2~16번째 라운드에서 C_{0init}와 C_{1init}는 이전 라운드에서 얻은 결과를 이용한다. C_{0init}, C_{1init}은 32 비트로 나누어 C_{0L}, C_{0R}, C_{1L}, C_{1R}로 표기한다.

[라운드 연산과정]

CA_L과 CA_R의 초기 상태는 각각 C_{0L}, C_{1L}과 C_{0R}, C_{1R}이고, C_{0L}, C_{1L}과 C_{0R}과 C_{1R}은 CA_L과 CA_R을 이용하여 각각 n번 반복한다. C_{nL}과 C_{nR}은 스왑과정을 수행한 후, CA_c의 초기 데이터 C_{0CR}과 C_{0CL}로 각각 변환된다. $C_{(n+1)L}$과 $C_{(n+1)R}$은 BitShift 변환을 이용

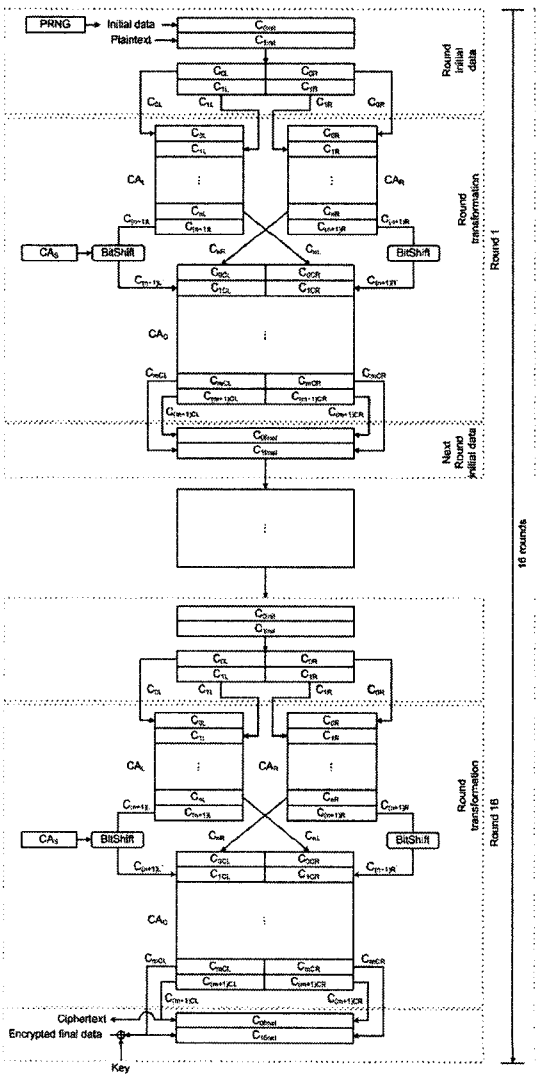

그림 5. CAB2 알고리즘

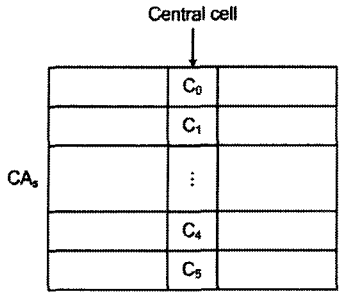

그림 6. n_s의 생성

하여 n_s만큼 왼쪽 순환 이동한 후, 각각 CA_c의 데이터 C_{1CL}과 C_{1CR}로 변환된다. 이때, n_s는 CA_S를 이용하여 6번 반복하여 그림 6과 같이 생성한다. C_{0CL}, C_{0CR}과 C_{1CL}, C_{1CR}은 CA_c를 이용하여 m번 반복한다. 여기에서 CA_L, CA_R, CA_C, CA_S는 가역 법칙을 이용한다. n과 m의 반복 횟수는 쇄도효과를 만족하기 위하여, 적어도 각각 19, 17번 이상 반복해야 한다.

[다음 라운드 초기 데이터 과정]

반복된 마지막 2개의 열 $C_{mCL} \| C_{mCR}$과 $C_{(m+1)CL} \| C_{(m+1)CR}$은 각각 C_{1final}과 C_{0final}로 변환하고, 다음 라운드의 초기 데이터가 된다. 마지막 라운드에서 C_{0final}은 암호문이 되고, C_{1final}은 다음 블록의 암호화 과정을 위한 초기 데이터가 된다.

첫 번째와 마지막 평문 블록을 암호화 할 때에는 특별한 경우가 발생한다. 첫 번째 경우에서 초기 데이터 C_{0init}는 의사난수생성기에 의해 생성된 64 비트 값이고, 두 번째 경우에서 C_{1fianl}는 키의 첫 번째 64 비트와 XOR 연산을 수행하고, 암호화된 최종 데이터 C_{1fianl}'가 된다.

4.3 키 생성

가역 법칙을 이용하는 CA_L, CA_R, CA_C, CA_S는 키에 의하여 정의된다. 즉, 키가 CA_L, CA_R, CA_C, CA_S의 갱신법칙을 정의하는 진리표가 된다. CA_L, CA_R, CA_C는 반지름이 2인 가역 법칙을 이용하고, CA_S는 반지름이 3인 가역 법칙을 이용한다. 반지름이 2인 가역 법칙에서 키는 $2^5 = 32$ 비트이고, 반지름이 3인 가역 법칙에서 키는 $2^7 = 128$ 비트이다. CA_L, CA_R, CA_C, CA_S의 셀 크기는 각각 32, 32, 64, 16 비트이고, 표 4와 같다. CA_C의 마지막 상태는 키의 0～63번째 비트를 이용하여 XOR 연산을 수행하고, CA_S의 마지막 상태는 키의 64～95번째 비트를 이용하여 암호화한다.

표 4. CAB2에 이용된 CA

	CA_L	CA_R	CA_C	CA_S
셀 크기	32	32	64	16
가역 법칙의 반지름	2	2	2	3
키 길이	32	32	32	128

5. CAB1에 대한 차분 공격

차분 공격은 평문이 특정한 형태로 변할 때 높은 확률로 암호문이 변하는 특성이 있으면 이 특성을 이용하여 주어진 암호를 해독하는 공격 방법이다. 기존의 암호가 만족해야 하는 성질로 평문 한 비트가 변할 때 암호문의 각 비트가 반 정도 변해야 한다는 요구 조건을 일반화시켜 암호 공격에 이용한 것이라고 할 수 있다.

5.1 F함수에 대한 암호학적 성질

본 절에서는 CAB1의 구성 요소들에 대한 차분 성질을 소개한다.

(성질 1) F함수를 제외한 모든 연산은 선형성을 만족하기 때문에, F함수를 제외한 연산은 XOR 차분 관점에서 입력 차분에 대한 출력 차분을 확률 1로 알 수 있다.

CAB1은 (성질 1)을 만족하므로, 비선형 함수인 F함수의 입력 차분에 대한 출력 차분의 관계를 분석함으로서, CAB1의 차분 특성을 도출할 수 있다. 1 비트 입력 차분은 상태전이 함수에 의하여 이웃한 3 비트에 영향을 주며, 상태전이 함수는 3 비트 키에 의하여 결정되기 때문에 키 값에 따라 각기 다른 차분 분포를 따른다.

(성질 2) F함수의 입력 차분 α_m에 대한 출력 차분은 그림 7과 같이 키 값에 종속적으로 결정된다. 임의의 키에 대하여 F함수의 입력 차분이 α_m $(2 \le m \le 29)$일 때, 출력 차분이 0이 될 확률은 2^{-3}이고 차분 분포표는 표 5와 같다. 여기서 "α_m"은 32 비트 워드로서 m번째 비트만 1이고, 나머지 비트는 0의 값을 갖는다.

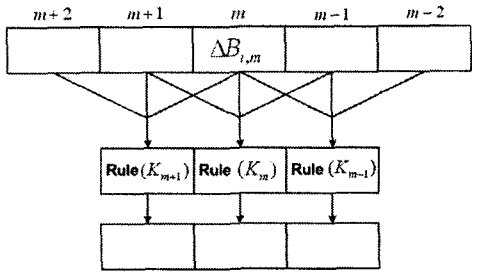

그림 7. F함수의 입력 차분 α_m에 대한 출력 차분

표 5. (성질 2)에 대한 F함수의 차분 분포표

법칙 \ 출력	000	001	010	011	100	101	110	111
000	4	4	4	4	4	4	4	4
001	4	4	4	4	4	4	4	4
010	4	4	4	4	4	4	4	4
011	4	4	4	4	4	4	4	4
100	4	4	4	4	4	4	4	4
101	4	4	4	4	4	4	4	4
110	4	4	4	4	4	4	4	4
111	4	4	4	4	4	4	4	4

표 6. 16 라운드 반복 차분 특성 A

j \ input	ΔB_0^j	ΔB_1^j	ΔB_2^j	ΔB_3^j	확률
0(IW)	0	α_m	0	0	1
1	0	α_m	0	0	1
2	α_m	0	0	0	2^{-3}
3	0	0	0	α_m	1
4	0	0	α_m	0	2^{-3}
5	0	α_m	0	0	1
6	α_m	0	0	0	2^{-3}
7	0	0	0	α_m	1
8	0	0	α_m	0	2^{-3}
9	0	α_m	0	0	1
10	α_m	0	0	0	2^{-3}
11	0	0	0	α_m	1
12	0	0	α_m	0	2^{-3}
13	0	α_m	0	0	1
14	α_m	0	0	0	2^{-3}
15	0	0	0	α_m	1
16	0	0	α_m	0	2^{-3}
17(FW)	0	0	α_m	β	1
18(C)	0	0	α_m	β	1
Total	·	·	·	·	2^{-24}

표 7. 16 라운드 반복 차분 특성 B

j \ input	ΔB_0^j	ΔB_1^j	ΔB_2^j	ΔB_3^j	확률
0(IW)	0	0	0	α_m	1
1	0	0	0	α_m	1
2	0	0	α_m	0	2^{-3}
3	0	α_m	0	0	1
4	α_m	0	0	0	2^{-3}
5	0	0	0	α_m	1
6	0	0	α_m	0	2^{-3}
7	0	α_m	0	0	1
8	α_m	0	0	0	2^{-3}
9	0	0	0	α_m	1
10	0	0	α_m	0	2^{-3}
11	0	α_m	0	0	1
12	α_m	0	0	0	2^{-3}
13	0	0	0	α_m	1
14	0	0	α_m	0	2^{-3}
15	0	α_m	0	0	1
16	α_m	0	0	0	2^{-3}
17(FW)	α_m	β	0	0	1
18(C)	α_m	β	0	0	1
Total	·	·	·	·	2^{-24}

5.2 전체 라운드 반복 차분 특성

본 절에서는 5.1에서 소개한 (성질 1)과 (성질 2)를 이용하여 표 6, 7과 같은 두 종류의 전체 라운드 반복 차분 특성을 구성한다. "β"는 32 비트 워드로서 연속된 3 비트만 1이고, 나머지 비트는 0의 값을 갖는다.

5.3 키 복구

본 절에서는 표 6, 7의 차분 특성을 이용하여 K_{66}, K_{67}의 최하위 4 비트 키와 $K_{64,1} \sim K_{64,30}$, $K_{65,1} \sim K_{65,30}$, K_{68}, K_{70}을 구하고, 공격 복잡도를 계산한다.

5.3.1 K_{64}, K_{65}, K_{66}, K_{67}의 부분키 복구와 K_{68}, K_{70}의 키 복구

K_{67}의 최하위 4 비트와 $K_{65,1} \sim K_{65,30}$, K_{70}는 단계 (1)~(3)을 이용하여 복구하고, 시나리오는 다음과 같이 여과과정, K_{67}의 최하위 4 비트 복구과정, $K_{65,1} \sim K_{65,30}$, K_{70} 복구과정으로 나누어진다.

(1) 여과과정

(a) $\Delta P=(0,0,\alpha_m,0)$을 만족하는 $2^{30.41}$의 선택 평문 쌍 (P, P')을 선택하고, 그에 대응하는 암호문 쌍 (C, C')을 획득한다.

(b) 차분 특성 $\Delta C=(0,0,\alpha_m,\beta)$을 만족하는 암호문 쌍 (C, C')을 필터링하고, 그 결과를 데이터 셋으로 구성한다.

(2) K_{67} 최하위 4 비트 복구과정 - 여과과정에서 구성한 데이터 셋의 원소에 대하여 다음 과정을 반복한다.

(a) ΔC_2와 ΔC_3의 순환 이동 값을 비교하여, K_{67}의 최하위 4 비트를 알아낸다.

(3) $K_{65,1}\sim K_{65,30}$, K_{70} 복구과정 - 여과과정에서 구성한 데이터 셋의 원소에 대하여 다음 과정을 반복한다.

(a) 16번째 라운드 F함수의 연접된 5 비트 입력값을 알기 위하여 $K_{70,m-2}\sim K_{70,m+2}$를 추측한다 $(2\le m\le 29)$.

(b) 3-(a)에 대응하는 16번째 라운드 F함수의 3 비트 출력값을 알기 위하여 $K_{65,m-1}\sim K_{65,m+1}$을 추측한다 $(2\le m\le 29)$.

(c) 16번째 라운드 F함수의 출력 차분은 (2)에서 복구한 K_{67}의 최하위 4 비트를 이용하여 왼쪽 순환 이동 연산을 수행한다.

(d) 3-(c)의 결과가 β와 같은 키는 카운트한다.

(4) 3-(d)에서 가장 많이 카운트 된 키 비트를 옳은 키 비트로 출력한다.

K_{66}의 최하위 4 비트 키와 $K_{64,1}\sim K_{64,30}$, K_{68}의 키는 위와 동일한 방법으로 복구한다.

5.3.2 전체 라운드 차분 공격에 대한 공격 복잡도

표 6에서 $\Delta C_3=\beta$는 16번째 라운드 F함수의 출력 차분을 K_{67}의 최하위 4 비트를 이용하여 왼쪽 순환 이동 시킨 차분이다. 표 6의 차분 특성을 만족하는 2^{24}의 선택 평문쌍에 대하여 K_{67}의 최하위 4 비트는 16번째 라운드 F함수의 출력 차분과 β의 순환 이동 된 비트 수를 이용하여 복구한다. 위와 같이 K_{66}의 최하위 4 비트는 표 7을 이용하여 복구한다.

표 6의 차분 특성을 만족하는 16번째 라운드 F함수의 연접한 3 비트 입력값을 알기 위하여 $K_{65,m-1}\sim K_{65,m+1}$와 $K_{70,m-2}\sim K_{70,m+2}$을 추측하면, F함수는 (성질 2)에 의하여 2^{-3}의 확률로 옳은 키 비트를 찾을 수 있고, 옳지 않은 키 비트를 옳은 키 비트로 찾을 확률은 $1-2^{-3}$이다. n번 시행할 경우 옳지 않은 키 비트를 옳은 키 비트로 찾을 확률은 $(1-2^{-3})^n$이다. 그러므로 n번 시행할 경우 옳게 추측한 키 비트가 옳은 키 비트가 될 확률 p는 다음 식과 같다.

$$p=1-[(1-2^{-3})^n]$$

$n=34$일 때, $K_{65,m-1}\sim K_{65,m+1}$와 $K_{70,m-2}\sim K_{70,m+2}$는 99%의 확률로 복구된다. 위의 과정을 10번 반복하면 $K_{65,1}\sim K_{65,30}$와 K_{70}을 복구할 수 있다. $K_{64,1}\sim K_{64,30}$와 K_{68}는 표 7을 이용하여 동일한 방법으로 복구한다.

CAB1에 대한 차분 공격은 $2\times[10\times34\times2^{22}]=2^{31.41}$의 선택 평문을 이용하면 성공 확률은 99%이고, 공격 복잡도는 다음과 같다.

$$2\times\left\{\frac{1}{16}[10\times(34\times2^8)]\right\}\approx 2^{13.41}$$

6. CAB2에 대한 취약점 분석

안전한 암호 알고리즘은 키를 모르는 상태에서 평문과 암호문의 어떠한 연관관계도 유출할 수 없어야 하며 키와 암호문 사이의 연관 관계 또한 찾을 수 없어야 한다. 그러나 암호문은 알고리즘에 의해 생성되기 때문에 평문과 암호문, 키와 암호문 사이에는 특별한 연관 관계가 존재한다. 따라서 키 전수 조사보다 적은 계산량으로 미지의 암호문으로부터 키나 평문의 어떠한 정보도 이끌어 낼 수 없을 때 주어진 암호 알고리즘은 안전하다고 한다. 암호 알고리즘의 출력문인 암호문에 대한 난수성 검정은 그 암호 알고리즘의 안전성을 증명할 수 있는 필요조건이다.

CAB2에서 키는 반지름이 2인 가역 법칙과 반지름이 3인 가역 법칙을 따른다. 여기에서, 0과 1의 개수가 다른 법칙을 이용하면, 0과 1중에서 많은 개수가 있는 값에 의하여 암호문은 편차(bias)가 발생하기 때문에, 이런 경우에 CAB2는 의사난수로서 취약하다. 그러므로 CAB2는 균일 성질을 만족하는 법칙만을 키로 이용하여야 한다. 이를 이용하여 CAB2는 다음과 같이 통계 분석을 한다.

[경우 1] CAB2의 키는 반지름이 2인 3개의 가역 법칙과 반지름이 3인 2개의 가역 법칙으로 구성되고,

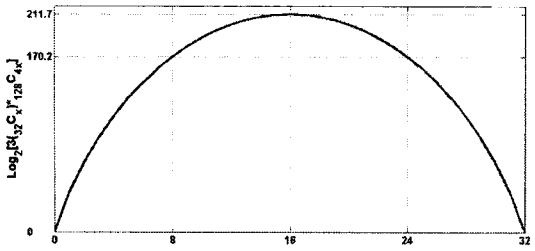

그림 8. [경우 1]에 대한 통계 분석

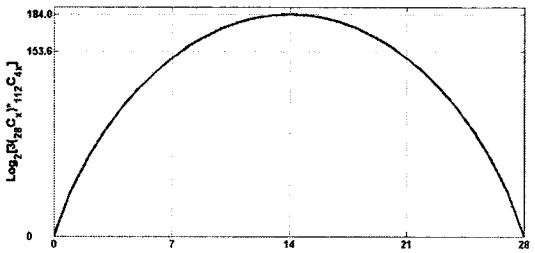

그림 9. [경우 2]에 대한 통계 분석

총 224 비트의 키 길이를 갖는다. 암호 알고리즘이 안전하기 위하여 가역 법칙을 만족하는 32, 32, 32, 128 비트의 키는 각각 균일 성질을 만족해야 하기 때문에, 각각의 가역 법칙은 0과 1의 개수가 동일해야 한다. 즉, $\log_2\left[\binom{32}{16}\times\binom{32}{16}\times\binom{32}{16}\times\binom{128}{64}\right]=211.7$ 비트의 안전성을 갖고, 그림 8과 같다.

[경우 2] 키가 단순히 키보드를 이용하여 입력될 경우를 생각해본다. 키보드를 이용하여 문자를 입력 시 ASCII CODE는 1 바이트로 하나의 문자를 정의하지만, 최상위 비트는 0으로 고정되어 있다. [경우 1]과 같이 암호 알고리즘이 안전하기 위하여 가역 법칙을 만족하는 28, 28, 28, 112 비트의 키는 각각 균일 성질을 만족해야 하기 때문에, 각각의 가역 법칙은 0과 1의 개수가 동일해야 한다. 즉, $\log_2\left[\binom{28}{14}\times\binom{28}{14}\times\binom{28}{14}\times\binom{112}{56}\right]=184$ 비트의 안전성을 갖고, 그림 9와 같다.

CAB2는 224 비트의 안전성을 가져야 하지만, 키가 균일 성질을 만족해야만 하는 취약점을 이용하여 184 비트의 안전성을 갖는다는 것을 알 수 있다.

7. 결 론

현재까지, CAB1과 CAB2는 제안 논문에서 제시한 안전성 분석 이외의 어떠한 분석 결과도 알려져 있지 않다. 본 논문에서는 $2^{31.41}$의 선택 평문을 이용하여, 공격 복잡도 $2^{13.41}$을 갖는 CAB1에 대한 차분 공격을 처음으로 소개하였고, 통계적 특성을 이용하여 CAB2가 224 비트 키 길이만큼의 충분한 안전성을 제공하지 못함을 보였다. 이 결과들은 CA가 아무리 암호학적으로 유용한 프리미티브 일지라도 유의해서 설계해야 함을 보여준다.

참 고 문 헌

[1] J. V. Neumann. The Theory of Self-Reproducing Automata, A. W. Burks (ed), Univ. of Illinois Press, Urbana and London, 1966.

[2] S. Wolfram, "Cryptography with Cellular Automata," Advances in Cryptology - CRYPTO 85, LNCS Vol.218, pp. 429-432, 1985.

[3] 이준석, 장화식, 이경현, "셀룰라 오토마타를 이용한 블록 암호 알고리즘," 한국멀티미디어학회, 제5권, 제6호, pp. 665-673, 2002.

[4] 신상욱, 윤재우, 이경현, "셀룰러 오토마타에 기반한 안전한 해쉬 함수," 한국통신정보보호학회 논문지, 제8권, 제4호, pp. 71-82, 1998. 12.

[5] 이준석, 장화식, 이경현, "셀룰라 오토마타를 이용한 스트림 암호," 한국멀티미디어학회 논문지, 제5권, 제2호, pp. 191-197, 2002. 4.

[6] 정기태, 이제상, 장동훈, 성재철, 이상진, "셀룰러 오토마타 기반 해쉬 함수 분석," 한국통신정보보호학회 논문지, 제14권, 제6호, pp. 111-123, 2004. 12.

[7] 임홍수, 홍득조, 성재철, 이상진, "셀룰러 오토마타 기반 스트림 암호 알고리즘에 대한 분석," 한국정보보호학회 동계 학술대회, 제14권, 제2호, pp. 20-24, 2004. 12.

[8] M. Seredynski and P. Bouvry, "Block Cipher based on Reversible Cellular Automata," CEC'04, pp. 2138-2143, 2004.

[9] Eli Biham and Adi Shamir, "Differential Crypt- analysis of the Data Encryption Standard," Advances in Cryptology - CRYPTO 90, LNCS Vol.537, pp. 2-21, 1990.

류 한 성

2006년 2월 고려대학교 수학과 학사
2006년 3월~현재 고려대학교 정보경영공학전문대학원 석사과정
관심분야 : 블록암호, 스트림 암호 및 해쉬 함수의 분석 및 설계

이 창 훈

2001년 2월 한양대학교 수학과 학사
2003년 2월 고려대학교 정보보호대학원 석사
2008년 2월 고려대학교 정보경영공학전문대학원 박사
2008년 3월~현재 고려대학교 정보보호연구원 연구교수
관심분야 : 대칭키 암호의 분석 및 설계

이 제 상

2003년 2월 고려대학교 수학과 학사
2006년 8월 고려대학교 정보보호대학원 석사
2006년 9월~현재 고려대학교 정보경영공학전문대학원 박사과정
관심분야 : 대칭키 암호의 분석 및 설계, 정보은닉이론, DRM

홍 석 희

1995년 2월 고려대학교 수학과 학사
1997년 2월 고려대학교 수학과 석사
2001년 2월 고려대학교 수학과 박사
1999년 8월~2004년 2월 (주) 시큐리티 테크놀로지스 선임연구원
2004년 4월~2005년 2월: K.U.Leuven 박사후연구원
2005년 3월~현재 고려대학교 정보경영공학전문대학원 조교수
관심분야 : 암호 알고리즘 설계 및 분석, 컴퓨터 포렌식