이차원 셀룰라 오토마타 기반 해쉬함수에 대한 충돌쌍 공격[1)]

*최준근 **류한성 ***이제상 ****홍석희

고려대학교 정보경영공학전문대학원

*joongeun@cist.korea.ac.kr

Collision Attack of a Hash Function based on 2D Cellular Automata

*Choi, Joon-Geun **Ryu, Han-Seong ***Lee, Je-Sang ****Hong, Seok-Hie

Graduate School of Information Management and Security, Korea University

요약

김재겸은 2005년 한국 멀티미디어 학회 논문지에 새로운 이차원 셀룰라 오토마타 설계 방법을 소개하고 이 설계 방법으로 구성된 이차원 셀룰라 오토마타를 이용한 해쉬함수를 제안하였다. 본 논문에서는 이 해쉬함수에 대한 첫 번째 분석 결과를 소개한다. 이 해쉬함수는 8 라운드로 구성되고 한 라운드는 두 개의 비선형 연산 부분을 포함하고 있으며, 메시지는 두 비선형 연산 부분에 모두 사용된다. 메시지 차분이 비선형 연산 부분을 거친 뒤 사라질 확률은 2^{-14}이다. 따라서 1 라운드 후 약 2^{-28}의 확률로 이 해쉬함수의 충돌쌍을 찾을 수 있다. 본 논문의 분석 결과를 통하여 이 해쉬함수는 매우 취약함을 알 수 있다.

1. 서론

셀룰라 오토마타(CA, Cellular Automata)란 공간과 시간을 이산적으로 다루고 셀룰라 공간의 기본 단위인 각 셀(Cell)의 상태를 유한하게 처리하며, 각 셀의 상태가 국소적 상호작용에 의해 동시에 갱신되는 시스템이다. 이런 셀룰라 오토마타의 확산 특성과 국소적인 상호작용은 암호시스템을 설계하는데 적합하여 LFSR의 대안으로 제시되었으며, 의사난수 생성기나 알고리즘 설계 등 다양한 암호학적 분야에서 쓰이고 있다.

2005년도에는 CA기법을 이용한 해쉬함수가 김재겸에 의해 제안되었다[1]. 본 논문에서는 전개상의 편의를 위해 이 해쉬함수를 2CAH라고 표기하겠다. 2CAH는 셀의 동시변환규칙과 비트의 위치변환규칙을 이용하여 고정된 해쉬값을 출력한다.

본 논문에서는 동시변환규칙의 취약점을 이용하여 2^{-28}의 확률로 충돌쌍을 찾을 수 있음을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 CA에 대한 기본적인 내용을 소개하고, 3장에서는 2CAH를 설명한다. 4장에서는 동시변환규칙의 차분 성질을 이용하여 전체라운드의 충돌쌍 공격을 제시한다. 마지막으로 5장은 본 논문의 결론이다.

2. 셀룰라 오토마타

CA란 동역학계(dynamic system)를 해석하는 한 방법으로 공간과 시간을 이산적으로 다루고, 이산적인 공간을 셀룰라 공간(cellular space)의 기본 단위인 각 셀의 취할 수 있는 상태를 유한하게 처리한다. CA는 배열에 따라서 1차원과 2차원으로 나뉜다. 가장 간단한 구조를 가지는 1차원 CA는 국소적인 상호작용이 인접하는 3개의 셀들에 의해서 이루어지는 1차원 3-이웃 CA이다.

CA를 설명하기 위해서는 다음의 기호들이 사용된다.

i : 1차원 배열에서 각 셀의 위치

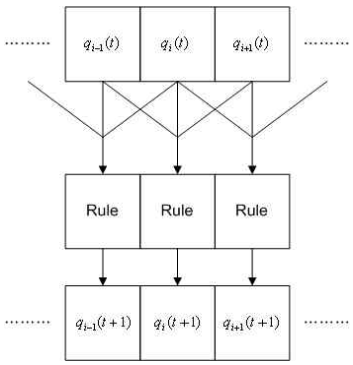
t : 시간 단계

$q_i(t)$: t번째 시간 단계에서 i번째 셀의 상태

$q_i(t+1)$: $(t+1)$번째 시간 단계에서 i번째 셀의 상태

3-이웃 CA에서의 각 셀의 상태의 동시 갱신에 사용되는 상태전이 함수는 다음과 같이 정의하며, [그림 1]과 같다.

$$q_i(t+1) = f[q_{i-1}(t), q_i(t), q_{i+1}(t)]$$



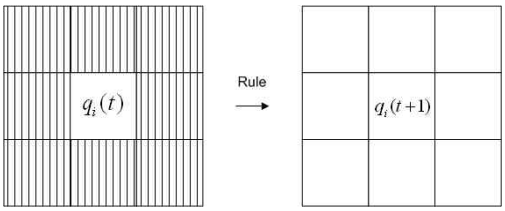
[그림 1] 3-이웃에 대한 상태전이 함수

2차원 CA는 2차원 배열의 각 셀에 대해 이웃해 있는 셀들이 다음 상태에 영향을 미치는 구조를 가지며, [그림 2]와 같다.

2CAH에서는 셀 공간을 2차원 배열로 구성하고, 1차원 3-이웃

1) 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-(C1090-0801-0025))

CA의 상태전이 함수를 사용한다.



[그림 2] 2차원 CA의 상태전이 함수의 예

3. 2CAH

본 절에서는 2005년 한국 멀티미디어 학회 논문지에 김재겸이 제안한 2CAH에 대하여 간략히 소개한다.

가. 표기

X_l : l번째 셀의 현재 상태
X_{l-1} : l번째 셀의 왼쪽 이웃의 현재 상태
X_{l+1} : l번째 셀의 오른쪽 이웃의 현재 상태
Y_l : l번째 셀의 다음 상태
Y_{l-1} : l번째 셀의 왼쪽 이웃의 현재 상태
Y_{l+1} : l번째 셀의 오른쪽 이웃의 현재 상태
$\overline{X_l}$: l번째 셀의 현재 상태의 보수
$\oplus$: bitwise XOR
OR : bitwise OR
$a \lll n$: a의 현재 상태를 정수 n만큼 왼쪽으로 로테이션
M_l : l번째 32비트 메시지 블록
A_3 : 32비트 블록 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번 비트라고 하였을 때, 24번 비트에서 31번 비트까지의 8비트
A_2 : 32비트 블록 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번 비트라고 하였을 때, 16번 비트에서 23번 비트까지의 8비트
A_1 : 32비트 블록 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번 비트라고 하였을 때, 8번 비트에서 15번 비트까지의 8비트
A_0 : 32비트 블록 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번 비트라고 하였을 때, 0번 비트에서 7번 비트까지의 8비트

나. 셀룰라 공간

2CAH의 셀룰라 공간은 1차원 CA와 달리 셀의 공간을 36개의 사각형들로 분할되어 있는 평면도형으로 설계하였다. 각 셀은 각각 C_0부터 C_{35}번까지의 셀 번호를 가지며, 각 셀은 8비트의 입력 값을 가진다.

C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}
C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}
C_{27}	C_{28}	C_{29}	C_{30}	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}

[표 1] 2CAH 셀룰라 공간

다. 상태전이 함수

2CAH에 사용되는 상태전이 함수 L과 NL은 다음과 같다.

1) 동시변환 규칙 L

동시변환 규칙 L은 선형함수로 구성되며, 다음과 같이 정의된다.

$$Y_l = (X_{l-1} \oplus X_l \oplus X_{l+1})$$

이 때 [표 1]의 C_0, C_9, C_{18}, C_{27} 셀의 왼쪽 이웃은 각각 C_8, C_{17}, C_{26}, C_{35}번 셀로 정의하고, [표 1]의 C_8, C_{17}, C_{26}, C_{35}번 셀의 오른쪽 이웃은 각각 C_0, C_9, C_{18}, C_{27} 번 셀로 정의한다.

2) 동시변환 규칙 NL

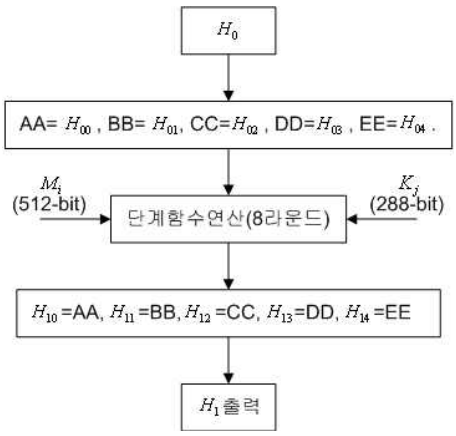
동시변환 규칙 NL은 비선형함수로 구성되며, 식은 다음과 같이 정의된다.

$$Y_l = (X_{l-1} \oplus X_l \oplus X_{l+1})\, OR\, (X_l \oplus \overline{X_{l+1}})$$

이 때도 위와 마찬가지로 [표 1]의 C_0, C_9, C_{18}, C_{27} 셀의 왼쪽 이웃은 각각 C_8, C_{17}, C_{26}, C_{35}번 셀로 정의하고, [표 1]의 C_8, C_{17}, C_{26}, C_{35}번 셀의 오른쪽 이웃은 각각 C_0, C_9, C_{18}, C_{27} 번 셀로 정의한다.

라. 2CAH

2CAH는 2005년 김재겸에 의하여 개발된 해쉬함수로서 임의의 길이메시지를 입력받아 160비트 해쉬값을 출력한다. 임의의 메시지가 주어지면 2CAH 해쉬함수는 메시지의 길이가 512비트의 배수가 되도록 덧붙이기(padding)를 수행하고, Merkle-Damgård 구성 방법을 이용하여 160비트 해쉬값을 출력한다. 전체 구조는 [그림 3]과 같다.



[그림 3] 2CAH

2CAH에서는 키를 사용하는 경우와 임의의 상수를 사용하는 경우에 모두 적용 가능하도록 구성되었다. 본 논문에서는 이와 무관하게 충돌쌍 공격이 수행되므로, 288비트의 임의의 상수 $K_j (0 \le j \le 8)$를 사용하겠다.

2CAH의 초기값, 단계함수 및 출력값은 다음과 같다.

1) 초기값

해쉬함수 연쇄변수의 초기 상수값 H_0는 16진수로 다음과 같다.

$H_{00} \leftarrow 67452301_x$ $H_{01} \leftarrow efcdab89_x$ $H_{02} \leftarrow 98badcfe_x$
$H_{03} \leftarrow 10325476_x$ $H_{04} \leftarrow c3d2e1f0_x$

이 값을 다음과 같이 초기화한다.

$AA = H_{00},\ BB = H_{01},\ CC = H_{02},\ DD = H_{03},\ EE = H_{04}$

2) 단계함수연산

단계함수연산은 다음과 같이 정의된다.

$C_0 \leftarrow AA_3, C_9 \leftarrow AA_2, C_{18} \leftarrow AA_1, C_{27} \leftarrow AA_0,$
$C_2 \leftarrow BB_3, C_{11} \leftarrow BB_2, C_{20} \leftarrow BB_1, C_{29} \leftarrow BB_0,$
$C_4 \leftarrow CC_3, C_{13} \leftarrow CC_2, C_{22} \leftarrow CC_1, C_{31} \leftarrow CC_0,$
$C_6 \leftarrow DD_3, C_{15} \leftarrow DD_2, C_{24} \leftarrow DD_1, C_{33} \leftarrow DD_0,$
$C_8 \leftarrow EE_3, C_{17} \leftarrow EE_2, C_{26} \leftarrow EE_1, C_{35} \leftarrow EE_0$
$i = 0, j = 0$
while(i<8) // 라운드함수
{
// 라운드 입력 메시지 및 키 셋팅
$C_1 \leftarrow M[2i]_3, C_{10} \leftarrow M[2i]_2, C_{19} \leftarrow M[2i]_1, C_{28} \leftarrow M[2i]_0,$
$C_3 \leftarrow K[j]_3, C_{12} \leftarrow K[j]_2, C_{21} \leftarrow K[j]_1, C_{30} \leftarrow K[j]_0,$
$C_5 \leftarrow K[j+1]_3, C_{14} \leftarrow K[j+1]_2, C_{23} \leftarrow K[j+1]_1, C_{32} \leftarrow K[j+1]_0,$
$C_7 \leftarrow M[2i+1]_3, C_{16} \leftarrow M[2i+1]_2, C_{25} \leftarrow M[2i+1]_1, C_{34} \leftarrow M[2i+1]_0$

for $k \leftarrow 0$ **to** 2 // 선형단계1
$Y_l = (X_{l-1} \oplus X_l \oplus X_{l+1})$
$(K[j]_3 \| K[j]_2 \| K[j]_1 \| \mathrm{K}[\mathrm{j}]_0) \lll 1$
$(CC_3 \| CC_2 \| CC_1 \| CC_0) \lll 7$
$(DD_3 \| DD_2 \| DD_1 \| DD_0) \lll 13$

for $k \leftarrow 0$ **to** 2 // 비선형단계1
$Y_l = (X_{l-1} \oplus X_l \oplus X_{l+1}) OR (X_l \oplus \overline{X_{l+1}})$
$(K[j]_3 \| K[j]_2 \| K[j]_1 \| \mathrm{K}[\mathrm{j}]_0) \lll 1$
$(CC_3 \| CC_2 \| CC_1 \| CC_0) \lll 7$
$(DD_3 \| DD_2 \| DD_1 \| DD_0) \lll 13$

// 라운드 입력 메시지 갱신
$M[2i] \lll 16, M[2i+1] \lll 24$
$C_1 \leftarrow M[2i]_3, C_{10} \leftarrow M[2i]_2, C_{19} \leftarrow M[2i]_1, C_{28} \leftarrow M[2i]_0$
$C_7 \leftarrow M[2i+1]_3, C_{16} \leftarrow M[2i+1]_2, C_{25} \leftarrow M[2i+1]_1, C_{34} \leftarrow M[2i+1]_0$

for $k \leftarrow 0$ **to** 2 // 선형단계2
$Y_l = (X_{l-1} \oplus X_l \oplus X_{l+1})$
$(K[j]_3 \| K[j]_2 \| K[j]_1 \| \mathrm{K}[\mathrm{j}]_0) \lll 1$
$(CC_3 \| CC_2 \| CC_1 \| CC_0) \lll 7$
$(DD_3 \| DD_2 \| DD_1 \| DD_0) \lll 13$

for $k \leftarrow 0$ **to** 2 // 비선형단계2
$Y_l = (X_{l-1} \oplus X_l \oplus X_{l+1}) OR (X_l \oplus \overline{X_{l+1}})$
$(K[j]_3 \| K[j]_2 \| K[j]_1 \| \mathrm{K}[\mathrm{j}]_0) \lll 1$
$(CC_3 \| CC_2 \| CC_1 \| CC_0) \lll 7$
$(DD_3 \| DD_2 \| DD_1 \| DD_0) \lll 13$

$i += 1, j += 1$
}

3) 출력값

모든 512비트 메시지 블록을 처리한 후의 연쇄변수 H_1은 다음과 같다.

$H_{10} \leftarrow C_0 \| C_9 \| C_{18} \| C_{27}, \quad H_{11} \leftarrow C_2 \| C_{11} \| C_{20} \| C_{29},$
$H_{12} \leftarrow C_4 \| C_{13} \| C_{21} \| C_{30}, \quad H_{13} \leftarrow C_6 \| C_{15} \| C_{24} \| C_{33},$
$H_{14} \leftarrow C_8 \| C_{17} \| C_{26} \| C_{25}$

4. 2CAH에 대한 충돌쌍 공격

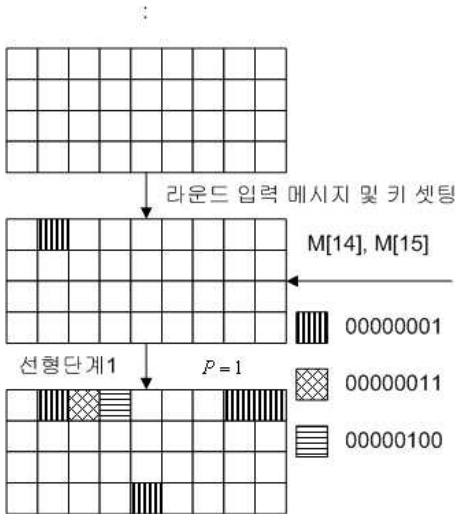
본 절에서는 2CAH의 전체라운드 차분특성을 구성하고, 구성된 차분특성을 이용한 충돌쌍 공격을 소개한다.

가. 차분 특성

본 공격에서 사용하는 메시지 워드의 차분은 다음과 같다.

$$\triangle M = M \oplus M' = (\triangle M[0], \triangle M[1], ..., \triangle M[15])$$
$$\triangle M[14] = 01000000_x, \triangle M[i] = 0 (0 \le i \le 15, i \ne 14)$$

7라운드까지 입력되는 메시지의 차분이 0이므로, 7라운드 후 전체 셀룰라 공간의 차분은 0이 된다. 이후 8라운드에 사용되는 메시지 $M[14]$에 최초로 0이 아닌 차분 01000000_x 이 입력되어 셀룰라 공간 C_1의 차분은 01_x이 되고, 나머지 공간들은 차분이 0이 된다. 선형단계 1 이후의 전체 셀룰라 공간의 차분은 확률 1로 [그림 4]와 같이 구성된다.



[그림 4] 선형단계 1 이후의 차분 특성

단계함수연산 NL에 의해 전체 셀룰라 공간의 차분이 0이 되는 확률을 계산한다. 우선, 비선형 함수 NL의 연산은 비트별로 수행된다. 그러므로 세 비트를 입력받아 한 비트를 출력하는 부울함수로 생각할 수 있다. 비선형 함수 NL의 차분 분포표를 구성하면 [표 4]와 같다.

입력차분 \ 출력차분	0	1	확 률
000	8	0	1
001	4	4	2^{-1}
010	4	4	2^{-1}
011	8	0	1
100	4	4	2^{-1}
101	4	4	2^{-1}
110	4	4	2^{-1}
111	4	4	2^{-1}

[표 4] 비선형 함수 NL의 차분 분포표

[표 4]에 의해서 선형단계 1과 비선형단계 1 이후에 전체 셀룰라 공간의 차분이 모두 0이 될 확률은 2^{-14}이다.

나. 충돌쌍 공격

앞에서 살펴본 선형함수 L과 비선형 함수 NL의 차분 특성을 이용하여 충돌쌍 공격을 제안한다.

서로 다른 메시지 M, M′에서 메시지 워드 $M[14]$에 01000000_x의 차분을 준 후, 선형함수 L과 비선형 함수 NL를 거치면, 2^{-14}의 확률로 차분이 0이 됨을 알 수 있다. 차분이 0이 된 후 라운드 입력 메시지를 갱신하여 셀룰라 공간에 입력하면 새로운 위치에 메시지 차분이 생기게 된다. 하지만, 위치만 바뀌고 최하위 비트의 차분은 그대로임을 알 수 있다. 선형단계 2와 비선형단계 2를 거친 후 모든 차분이 사라질 확률은 위와 마찬가지로 2^{-14}이 된다. 따라서 $\triangle M$을 만족하는 메시지 M, M′에 대하여 해쉬 출력값이 같을 확률은 2^{-28}이다.

그러므로 2^{28}개의 메시지 쌍이 주어졌을 때, 1개의 충돌쌍을 찾을 수 있다.

5. 결 론

본 논문에서는 2차원 셀룰라 오토마타를 기반으로 한 해쉬함수를 분석하였다. 본 논문에서 제안한 공격을 이용하여 충돌쌍을 2^{-28}의 확률로 찾을 수 있었다. 이를 통하여 이 해쉬함수는 매우 취약함을 알 수 있다. 또한 본 논문에서 이용한 차분 특성을 이용하여 더 좋은 확률로 충돌쌍을 찾을 수 있을 것으로 예상된다.

참 고 문 헌

[1] 김재겸, “이차원 셀룰라 오토마타에 기반하는 해쉬 함수”, Journal of Korea Multimedia Society Vol. 8, No. 5, 2005.

[2] 정기태, 이제상, 장동훈, 성재철, 이상진, “셀룰러 오토마타 기반 해쉬 함수 분석”, 정보보호학회 논문지, 제 14권 6호, 2004.

[3] 류한성, 이제상, 이창훈, 성재철, 홍석희, “셀룰러 오토마타 기반 블록 암호에 대한 부분키 공격”, KoreaCrypt 2007.

[4] P. Chaudhuri, D. Chowdhury, S. Nandi and S. Chatterjee,"Additive Cellular Automata - Theory and Applications", IEEE Computer Society Press, Vol. 1, CA, USA, 1997.

[5] S. Wolfram, "Cryptography with Cellular Automata", CRYPTO' 85, LNCS 218, pp. 429-432, Springer-Verlag, 1985.

[6] P. P. Chaudhuri, D. R. chowdhuri, S. Nandi, S. Chattopadhyay, "Additive Cellular Automata: Theory and Applications", IEEE Press, NewYork, 1997.

[7] S. Nandi, B. K. Kar, P. Pal Chaoudhuri, "Theory and Applications of Cellular Automata in Cryptography", IEEE Transaction on Computer, Vol. 43, No. 12, 1994.

[8] S. Wolfram, "cryptography with cellular automata", Internet request for momments 1321, R.L. Rivest, 1992.

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