情報保護學會論文誌
第12券 第2號, 2002. 4

PKC'98에 제안된 해쉬 함수의 Original Version에 대한 전체 라운드 차분 공격

장 동 훈*, 성 재 철*, 성 수 학**, 이 상 진*, 임 종 인*

Full-Round Differential Attack on the Original Version of the Hash Function Proposed at PKC'98

Donghoon Chang*, Jaechul Sung*, Soohak Sung**, Sangjin Lee*, Jongin Lim*

요 약

신상욱 등은 PKC'98에서 기존 RIPEMD-160, HAVAL, SHA-1와 같은 해쉬 함수의 장점을 이용하여 160비트의 출력 길이를 갖는 새로운 해쉬 함수를 제안하였다.[1] 최근 FSE 2002에서 한 대완 등은 PKC'98에 제안된 해쉬 함수의 부울 함수가 당초 설계자의 의도와는 달리 일부 부울 함수가 SAC(Strict Avalanche Criterian)을 만족하지 않음을 지적하고, 설계자의 의도에 맞게 모든 부울 함수가 SAC의 성질을 만족한다는 가정 하에, 2^{-30}의 확률로 충돌 쌍을 찾는 공격 방법을 제안하였다.[2] 본 논문에서는 위의 방법을 개선하여, PKC'98에서 제안된 해쉬 함수의 original version의 전체 라운드에 대해 $2^{-37.13}$의 확률로 충돌 쌍을 찾을 수 있음을 보인다. 그리고 PKC'98에 제안된 해쉬 함수의 문제점이 메시지에 의존한 쉬프트 값의 사용에 있음을 지적한다.

ABSTRACT

Shin *et al.* proposed the new hash function with 160-bit output length at PKC'98. This hash function is based on the advantages of the existing hash functions, such as SHA-1, RIPEMD-160, HAVAL, and etc.[1] Recently, Han et al. cryptanalyzed the hash function proposed at PKC'98 and proposed the method finding a collision pair with 2^{-30} probability at FSE 2002 ,supposing that boolean functions satisfy SAC(Strict Avalanche Criterian).[2] This paper improves the method and shows that we can find a collision pair from the original version of the hash function with $2^{-37.13}$ probability through the inproved method. And we point out that the problem of the function comes from shift values dependent on message.

keyword : *Hash function, Boolean function, SAC, Collision pair*

Ⅰ. 서 론

해쉬 함수는 길이가 유한한 임의의 메시지를 입력으로 하여 고정된 길이의 비트 열을 출력하는 다대일 함수를 말한다. 이러한 성질이 암호학 분야에서 암호학적 해쉬 함수라는 이름으로 인증 및 디지털 서명 분야에 응용이 되고 있다. 암호학적으로 해쉬 함수가 만족해야할 성질은 다음과 같다.

첫째, 임의의 해쉬 값이 주어졌을 때 그것에 해당하는 입력 m을 구하는 것이 계산적으로 불가능해야 한다.

둘째, 주어진 입력에 대한 해쉬 값이 주어질 때, 같은 출력을 내는 다른 입력을 찾아내는 것이 계산

* 고려대학교 정보보호기술연구센터(pointchang@cist.korea.ac.kr)
** 배재대학교 응용수학과

상 불가능해야 한다.

셋째, 동일한 해쉬 값을 갖는 서로 다른 두 개의 메시지 m, m'를 찾는 것이 계산적으로 불가능해야 한다.

잘 알려진 전용 해쉬 함수에는 1990년 Rivest가 발표한 MD4[3] 이후로 MD5[4], HAVAL[5], SHA-1[6] 등이 나왔다. 이중 MD4는 전체 라운드에 대해 공격이 되었고[7], MD5의 경우엔 초기 값을 다르게 했을 때 공격이 되었으며[8], HAVAL은 부분적으로 공격이 되었다[9].

최근 FSE 2002에서 한 대완 등은 PKC'98에 제안된 해쉬 함수의 부울 함수가 당초 설계자의 의도와는 달리 일부 부울 함수가 SAC(Strict Avalanche Criterian)을 만족하지 않음을 지적하였다. 그리고, 설계자의 의도에 맞게 모든 부울 함수가 SAC을 만족할 경우 2^{-30}의 확률로 충돌 쌍을 찾는 방법을 제안하였다[2]. 하지만, 실제 PKC'98에 제안된 해쉬 함수의 각 라운드에 사용된 3개의 부울 함수 중에 단 1개만이 SAC의 성질을 만족한다. 따라서 FSE 2002에 발표된 공격이 실제 해쉬 함수에 적용되지는 않는다.

본 논문에서는 FSE 2002에서 제안한 공격 방법을 개선하여, PKC'98에 제안된 해쉬 함수의 original version의 전체 라운드에 대해 $2^{-37.13}$의 확률로 충돌 쌍을 찾을 수 있음을 보인다. 그리고, PKC'98에서 제안된 해쉬 함수의 문제점이 메시지에 의존한 쉬프트 값의 사용에 있음을 지적한다.

Ⅱ. PKC'98에 제안된 해쉬 함수의 소개

지금부터는 PKC'98에 제안된 해쉬 함수를 간단히 P-해쉬 함수라고 부르자. 먼저 P-해쉬 함수에 사용되는 용어와 기호를 다음과 같이 정의한다.

- 워드 : 32 비트 스트링
- 블록 : 512비트 메시지(16개의 워드)
- $+$: 법 2^{32}에서의 덧셈 연산
- $X^{\lll s}$: X를 왼쪽으로 s비트의 순환이동 연산
- $X \wedge Y$: 두 워드의 AND 연산
- $X \vee Y$: 두 워드의 OR 연산
- $X \oplus Y$: 두 워드의 XOR 연산

2.1 입력 블록의 크기와 패딩 과정

512비트 단위로 메시지를 처리한다. 512비트의 배수가 되도록 메시지 뒤에다가 패딩작업을 하는데, 마지막 메시지의 길이가 448비트가 되도록 1 다음에 필요한 0의 개수만큼 채운다. 끝의 64비트에는 메시지의 길이를 mod 2^{64}로 계산하여 채운다.

2.2 초기 값(IV)

메시지를 해쉬하는데 사용되는 5개의 연쇄변수의 초기값은 다음과 같다.

초기값(IV)
=(A,B,C,D,E)=(0x67452301,0xefcdab89,
0x98badcef,0x10325476,0xc3d2e1f0)

2.3 상수

각 라운드에서 사용되는 상수 K의 값은 다음과 같다.

$K_1 = 0$
$K_2 = 0x5a827999$
$K_3 = 0x6ed9eba1$
$K_4 = 0x8f1bbcdc$

2.4 메시지 변수의 확장

512비트인 $X_0, X_1, \cdots X_{15}$ 로부터 $X_{16}, X_{17}, \cdots X_{23}$을 다음과 같은 방식으로 확장하여 얻는다. 그래서 실제 각 라운드에는 768비트, 즉 24개의 워드가 적용된다.

$$X_{16+i} = (X_{0+i} \oplus X_{2+i} \oplus X_{7+i} \oplus X_{12+i})^{\lll 1} \quad (1)$$
$$(i = 0,1,2,\cdots,7)$$

2.5 메시지 워드의 적용 순서

메시지 워드의 적용 순서를 정의하기 위해 먼저 다음의 순열 ρ를 정의한다.

〔표 1〕 순열 ρ의 정의

i	0	1	2	3	4	5	6	7	8	9	10	11
$\rho(i)$	4	21	17	1	23	18	12	10	5	16	8	0
i	12	13	14	15	16	17	18	19	20	21	22	23
$\rho(i)$	20	3	22	6	11	19	15	2	7	14	9	13

순열 ρ에 의한 메시지 적용순서는 다음과 같다.

라운드	1	2	3	4
순 열	id	ρ	ρ^2	ρ^3

2.6 부울 함수

각 라운드에 사용되는 부울 함수는 다음과 같다. 이 중 f_2만이 SAC의 성질을 만족하고, 나머지 f_0와 f_1은 SAC의 성질을 만족하지 않는다.

$$f_0(x_1,x_2,x_3,x_4,x_5) = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus (x_2 \wedge x_3 \wedge x_4) \oplus x_5 \quad (2)$$

$$f_1(x_1,x_2,x_3,x_4,x_5) = x_2 \oplus ((x_4 \wedge x_5) \vee (x_1 \wedge x_3)) \quad (3)$$

$$f_2(x_1,x_2,x_3,x_4,x_5) = x_1 \oplus (x_2 \wedge (x_1 \oplus x_4)) \oplus (((x_1 \wedge x_4) \oplus x_3) \wedge x_5) \quad (4)$$

2.7 단계 연산

각 단계에서의 연산은 다음과 같이 정의된다.

총 4라운드(96단계)로 이루어져 있는데, 1라운드(0~23단계)에는 f_0, 2라운드(24~47단계)에는 f_1, 3라운드(48~71단계)에는 f_2, 4라운드(72~95단계)에는 다시 f_1 이 각각 적용된다.

$$A = (f(A,B,C,D,E) + X_i + K) \lll S_i,\ B = B \lll 10$$
$$B = A,\ C = B,\ D = C,\ E = D,\ A = E \quad (5)$$

2.8 순환 이동

각 단계에 사용되는 순환 이동의 값 S_i는 다음과 같이 메시지 워드에 의존하여 결정된다.

$$S_i = X_{R(i \bmod 24)} \bmod 32 \quad (6)$$

다음은 라운드별 R 함수를 나타낸 것이다.

라운드	1	2	3	4
R 함수	ρ^3	ρ^2	ρ	id

예를 들어, 20단계의 쉬프트 값인 S_{20}을 구해보자. 20단계는 1라운드이므로 R함수는 ρ^3이다. 따라서 $S_{20} = X_{\rho^3(20)} \bmod 32 = X_8 \bmod 32$ 가 된다.

2.9 표로 살펴본 각 단계별 연산

다음의 표는 단계 0~10까지의 연산에서 각 변수의 변화와 입력 워드 및 쉬프트에 사용되는 워드를 나타낸 것이다.

〔표 2〕 단계별 메시지 입력과 각 변수의 변화표

단계	A	B	C	D	E	입력	쉬프트
0	A_0	B_0	C_0	D_0	E_0	X_0	X_{13}
1	A_1	B_1	C_1	D_1	E_1	X_1	X_{22}
2	A_2	B_2	C_2	D_2	E_2	X_2	X_2
3	A_3	B_3	C_3	D_3	E_3	X_3	X_{14}
4	A_4	B_4	C_4	D_4	E_4	X_4	X_3
5	A_5	B_5	C_5	D_5	E_5	X_5	X_6
6	A_6	B_6	C_6	D_6	E_6	X_6	X_7
7	A_7	B_7	C_7	D_7	E_7	X_7	X_5
8	A_8	B_8	C_8	D_8	E_8	X_8	X_{15}
9	A_9	B_9	C_9	D_9	E_9	X_9	X_0
10	A_{10}	B_{10}	C_{10}	D_{10}	E_{10}	X_{10}	X_{18}

□ : 각 단계마다 갱신되는 부분

예를 들어, A_0, E_1는 다음과 같이 갱신된다.

$$A_0 = [f_0(A,B,C,D,E) + X_0 + K_1] \lll S_0 = (X_{13} \bmod 32)$$
$$E_1 = [f_0(E_0,A_0,B_0,C_0,D_0) + X_1 + K_1] \lll S_1 = (X_{22} \bmod 32)$$

Ⅲ. P-해쉬 함수의 분석

3.1 부울 함수에 대한 분석

각 라운드에 쓰인 부울 함수는 5개의 워드를 입력으로 받아 비트별 연산을 한 후 한 개의 워드를 출력한다.

부울 함수 f_2는 SAC을 만족한다. SAC이라함은 한비트에만 차분을 줄 때, 출력비트가 0이될 확률이 1/2이 됨을 의미한다. SAC의 성질을 공격자의 입장에서 보면, 한 비트에만 차분을 줄 때 확률 1/2로 출력 차분을 0으로 만들 수 있음을 뜻한다.

FSE 2002에서 한 대완 등은 이러한 사실을 기

초로 P-해쉬 함수의 제안자들의 의도대로 세 개의 부울 함수가 SAC을 만족한다는 가정하에 2^{-30}의 확률로 충돌쌍을 찾는 방법(부울 함수의 5개의 입력 비트 중 한 개의 입력 비트에만 차분을 주어 차분 확산을 확률적으로 막는 방법)을 제안하였다. 그러나, f_0, f_1은 SAC을 만족하지 않기에 그 방법은 P-해쉬 함수에 적용이 되지 않는다. 그 이유는 다음과 같다.

부울 함수 f_0는 마지막 입력 비트의 차분만이 1일 때, 출력 차분은 확률 1로 1이 됨을 알 수 있다. 그래서 차분 확산이 일어난다.

또한 부울 함수 f_1은 두 번째 비트의 차분만이 1일 때, 출력 차분은 확률 1로 1이 되는 성질이 있다. 이 역시 차분 확산을 일으킨다.

이러한 이유로 인해, 부울 함수의 한 개의 입력 비트에만 차분을 주는 방법으로는 P-해쉬 함수의 충돌쌍을 찾을 수 없다. 그러나, 차분을 주는 입력 비트의 개수를 늘리는 식으로 한 대완 등이 제안한 방법을 개선하여 생각할 때, 상황은 달라진다. 왜냐하면, f_0, f_1는 SAC을 만족하지는 않지만, 차분을 주는 비트의 개수를 늘려서 생각하면, 확률적으로 차분 확산을 막을 수 있기 때문이다. 다음의 표는 이를 잘 보여 준다. [표 3]은 번호 5와 13의 경우를 제외하고는 확률적으로 차분 확산을 막을 수 있음을 뜻한다.

[표 3] f_i 함수의 입력 차분 비트의 위치 변화에 따라 출력 차분 비트가 0이 될 확률

번호	함수	입력 차분 발생 비트	확률	번호	함수	입력 차분 발생 비트	확률
1	f_0	x_1	1/2	17	f_1	x_2, x_3	3/8
2	f_0	x_2	1/2	18	f_1	x_2, x_5	3/8
3	f_0	x_3	3/4	19	f_1	x_1, x_4	3/8
4	f_0	x_4	3/4	20	f_1	x_1, x_2	3/8
5	f_0	x_5	0	21	f_1	x_1, x_3	5/8
6	f_0	x_1, x_2	1/2	22	f_1	x_2, x_4	3/8
7	f_0	x_1, x_4	1/4	23	f_1	x_3, x_5	3/8
8	f_0	x_3, x_5	1/4	24	f_1	x_4, x_5	5/8
9	f_0	x_1, x_5	1/2	25	f_1	x_3, x_4	3/8
10	f_0	x_3, x_4	3/4	26	f_1	x_1, x_5	3/8
11	f_0	x_4, x_5	1/4	27	f_1	x_2, x_3, x_4	5/8
12	f_1	x_1	5/8	28	f_2	x_1	1/2
13	f_1	x_2	0	29	f_2	x_2	1/2
14	f_1	x_3	5/8	30	f_2	x_3	1/2
15	f_1	x_4	5/8	31	f_2	x_4	1/2
16	f_1	x_5	5/8	32	f_2	x_5	1/2

3.2 단계 연산에 대한 분석

P-해쉬 함수에서 사용된 연쇄 변수는 5개이다. 식 (5)를 보았을 때, 부울 함수에 5개의 연쇄 변수가 그대로 적용이 되며, 하나의 연쇄 변수가 새롭게 갱신된다. 이어서 두 번째 연쇄 변수가 왼쪽으로 10만큼 순환이동을 하고, 마지막으로 5개의 연쇄변수 (A, B, C, D, E)가 (B, C, D, E, A)로 순서가 바뀌어서 다음 단계의 부울 함수에 적용이 된다.

여기서 주목해야 할 부분은 세 가지이다. 첫째, 두 종류의 쉬프트 값의 적용, 둘째, 5개의 연쇄 변수를 모두 부울 함수의 입력 값으로 처리한 부분이다(이와 같은 설계 방식은 차분 확산 효과를 크게 높이지 못한다). 셋째, 부울 함수의 특성에 있어서이다.

차분 공격 관점에서 바라봤을 때, 두 가지의 쉬프트 값과 연쇄 변수의 순서를 바꾸는 과정은 차분 확산을 일으킨다. 그리고, 차분 확산의 효과는 f_0, f_1와 같이 확률 1로 차분 확산을 일으킬 수 있는 부울 함수를 쓸 때 커진다. 또는 3 변수 혹은 4변수 부울 함수를 사용하고, 나머지 연쇄 변수는 덧셈 연산을 하면 차분 확산의 효과는 커지게 된다. 특히 갱신된 연쇄 변수를 그 다음 단계에서 부울 함수에 작용시키는 것이 아니라 HAS-160의 경우처럼 덧셈 연산에 작용시키면, 차분의 효과는 더욱 커지게 된다.

구체적으로, 단계 연산에 있어서의 차분 분석을 위해 다음을 고려할 필요가 있다. 단계 연산에서는 $\oplus$를 쓰지 않고, 법 2^{32} 위에서의 덧셈을 사용한다. 법 2^{32} 위에서의 덧셈을 $\oplus$관점에서의 차분으로 식의 확률을 계산할 때는 carry 발생에 따른 확률을 생각해야 한다. 만약 k=31일 때, 즉 최상위비트에 대해서는 덧셈과 $\oplus$ 연산은 동일한 역할을 하기에 carry는 발생하지 않으며, 나머지 비트에 대해서는 1/2의 확률로 carry가 발생하게 된다. 이를 정리하면 다음과 같다. 여기서 p는 식이 성립하는 확률을 나타낸다.

경우 1 : k=31

$$((A \oplus 1^{\lll k}) + B) \oplus (A + B) = 1^{\lll k} \quad (p=1) \qquad (7)$$

$$((A \oplus 1^{\lll k}) + (B \oplus 1^{\lll k})) \oplus (A + B) = 0 \quad (p=1) \qquad (8)$$

경우 2 : k≠31

$$((A \oplus 1^{\lll k}) + B) \oplus (A + B) = 1^{\lll k} \quad (p=1/2) \qquad (9)$$

$$((A \oplus 1^{\ll k}) + (B \oplus 1^{\ll k})) \oplus (A + B) = 0 \quad (p = 1/2) \qquad (10)$$

3.3 입력 메시지에 의존하는 쉬프트 값에 대한 분석

P-해쉬 함수는 입력 메시지에 의존하는 쉬프트 값을 사용한다. P-해쉬 함수의 제안자들은 이러한 쉬프트 값의 사용이 기존의 공격에 대해 안전성을 증가시킬 것이라고 주장하였다.

그러나, FSE 2002에서 한 대완 등은 메시지에 의존한 쉬프트 값이 오히려 공격자로 하여금 쉬프트 값을 임의적으로 조작하게 할 수 있음을 보여 주었다. 또한 그들은 메시지 워드에 의존하는 쉬프트 값이 전용 해쉬 함수에 적절치 않을 것이라고 추측을 하며 결론을 맺었다. 이 결론은 P-해쉬 함수와 같은 형태의 전용 해쉬 함수의 경우엔 옳은 말이다. 왜냐하면, 쉬프트 값의 임의 선택이 가능한 경우 차분 공격을 가능케 하기 때문이다. 그 이유는 다음과 같다.

차분(⊕의 관점) 공격을 하기 위해서는 [표 3]과 식 (7)~(10)이 필요하다. 그런데, 쉬프트 값의 임의 선택이 가능하다는 것은 차분이 발생한 비트의 위치를 원하는 위치로 변경시킬 수 있다는 뜻이 된다. 그래서 [표 3]과 식 (7)~(10)을 이용한 차분 확산의 차단을 가능케 한다. 일반적으로 쉬프트 값이 고정되어 있는 MDx계열의 해쉬 함수에 대해 차분 공격을 하기 위해서는 쉬프트 값, 메시지 워드의 확장, 그리고 메시지의 입력 순서를 동시에 고려해야 한다. 그러나, P-해쉬 함수는 메시지 워드의 확장과 입력 순서와 상관없이 선택된 메시지 워드에 따라 쉬프트 값을 적절히 조절함으로써 차분 공격이 가능하다.

그래서 다음 Ⅳ절에서는 차분 확률이 가장 높아지도록 하는 메시지 블록 쌍을 먼저 선택하고, 선택된 메시지 블록 쌍을 기초로 적절한 쉬프트 값을 선택한다. 그 다음 Ⅴ절에서는 실제로, P-해쉬 함수 original version에 대한 차분 공격을 시도한다.

Ⅳ. 공격 확률을 높이기 위한 메시지 블록 쌍과 이에 따른 쉬프트 값의 선택

공격 확률은 차분을 주는 메시지 워드의 수가 작을수록 높아진다. 왜냐하면, 차분 공격은 확률적으로 차분 확산을 막는 것인데, 차분이 존재하는 단계가 많을수록 차분 확산을 막기 위한 확률 값이 점점 작아지기 때문이다. 이러한 의미에서 메시지 워드의 확장에 대한 분석을 보고, 공격 확률을 극대화하기 위한 메시지 블록 쌍을 선택하는 방법을 보이는 것은 의미가 있다. 메시지 워드에 대한 확장 분석은 앞서 FSE 2002에서 한 대완 등이 보여 주었다.

먼저, 식 (1)을 풀어 쓰면 다음과 같다.

$$X_{16} = (X_0 \oplus X_2 \oplus X_7 \oplus X_{12})^{\ll 1} \qquad (11)$$

$$X_{17} = (X_1 \oplus X_3 \oplus X_8 \oplus X_{13})^{\ll 1} \qquad (12)$$

$$X_{18} = (X_2 \oplus X_4 \oplus X_9 \oplus X_{14})^{\ll 1} \qquad (13)$$

$$X_{19} = (X_3 \oplus X_5 \oplus X_{10} \oplus X_{15})^{\ll 1} \qquad (14)$$

$$X_{20} = (X_4 \oplus X_6 \oplus X_{11} \oplus X_{16})^{\ll 1} \qquad (15)$$

$$X_{21} = (X_5 \oplus X_7 \oplus X_{12} \oplus X_{17})^{\ll 1} \qquad (16)$$

$$X_{22} = (X_6 \oplus X_8 \oplus X_{13} \oplus X_{18})^{\ll 1} \qquad (17)$$

$$X_{23} = (X_7 \oplus X_9 \oplus X_{14} \oplus X_{19})^{\ll 1} \qquad (18)$$

그리고, 메시지 워드의 확장을 통한 각 메시지 워드 $X_i (0 \le i \le 15)$에 의해 영향을 받는 메시지 워드들은 다음의 표와 같다.(X_i를 i로 표기함)

[표 4] 각 메시지 워드가 영향을 주는 확장된 메시지들

0	1	2	3	4	5	6	7
16	17	16	17	18	19	20	16
20	21	18	19	20	21	22	20
		20	21	22	23		21
		22	23				23
8	9	10	11	12	13	14	15
17	18	19	20	16	17	18	19
21	22	23		20	21	22	23
22	23			21	22	23	

예를 들어 위의 표를 보았을 때, X_0에 차분을 줄 경우, 동시에 X_{16}와 X_{20}도 차분을 갖게 됨을 알 수 있다. 이와 같이 한 메시지 워드는 다른 메시지 워

드에 반드시 영향을 주게 되어, 하나의 메시지 워드에만 차분을 주는 것은 불가능하다. 그러나, 두 메시지 워드에는 차분을 줄 수 있다. 위의 표를 기초로 보면, 다음과 같이 네 경우가 있다. (X_8, X_{13}), (X_9, X_{14}), (X_{10}, X_{15}), (X_{11}, X_{20}).

이와 같이 FSE 2002에서 한 대완 등이 보여준 메시지 워드의 확장에 대한 분석을 토대로, 두 메시지 워드에만 차분을 줄 수 있는 네가지 쌍을 찾을 수 있었다. 그리고 우리는 네가지 쌍에 대해 차분을 줄 경우 Ⅲ절의 분석을 기초로 높은 확률로 충돌쌍을 찾을 수 있음을 알 수 있었다. 본 논문에서는 그 중에서도, 공격확률을 극대화하기 위해서 (X_{11}, X_{20})을 선택한다. 왜냐하면 (X_{11}, X_{20})의 경우에 라운드별 입력 단계간의 차이가 가장 작기 때문이다. 두 메시지가 입력되는 단계간의 차이가 작을수록 차분이 존재하는 단계의 수가 작아져서 공격 확률이 높아지게 된다. 이를 표로 보면 다음과 같다.

〔표 5〕 각 라운드에서 두 메시지 워드가 입력되는 단계의 차이

라운드	(X_8, X_{13})	(X_9, X_{14})	(X_{10}, X_{15})	(X_{11}, X_{20})
1 라운드	5	5	5	9
2 라운드	13	1	11	4
3 라운드	3	13	13	3
4 라운드	20	18	4	7
합계	41	37	33	23

그리고 단계 연산에서 최상위 비트에 차분을 주면 2^{32} 위에서의 덧셈을 확률 1로 $\oplus$으로 바꾸어 생각할 수 있다. 또한, 차분을 주는 비트의 수가 작을수록 공격확률은 커진다. 또한, 두 메시지 블록에 대해 동일한 쉬프트 값을 적용시키기 위해 메시지 워드 X_{11}의 최상위 비트에는 차분을 주지 않는다. 왜냐하면 식 (15)와 같이 X_{11}의 최상위 비트에 차분을 줄 경우, X_{20}의 최하위 비트에 차분이 발생하여, 두 메시지 블록에 대한 쉬프트 값이 달라지기 때문이다.

이상을 기초로 하여, 다음과 같이 메시지 블록 쌍과 쉬프트 값을 선택하였다. 그리고, 다음 절에서는 이를 토대로하여 충돌 쌍을 찾을 수 있음을 보인다.

▶ $X_{11} \oplus \overline{X_{11}} = 1^{\ll 30}$ ($\overline{X_{20}} \oplus 1^{\ll 31} = X_{20}$)

▶ $\overline{X_i} = X_i$ ($i \neq 11, 20$)

▶ 고정시키는 메시지 워드의 최하위 5 비트의 값
($X_1, X_4, X_7, X_{12}, X_{15}, X_{16}, X_{17}, X_{20}, X_{23}$)
=(10, 0 , 21, 21, 11, 0, 10, 0, 13)

다음은 각 단계별 메시지 워드의 입력 순서와 단계별 쉬프트 값을 나열한 표이다.(M:메시지 워드의 순서, ·:임의의 값)

〔표 6〕 단계별 메시지 워드의 입력 순서와 쉬프트 값

단계	M	쉬프트	단계	M	쉬프트	단계	M	쉬프트	단계	M	쉬프트
0	0	.	24	4	13	48	23	0	72	13	.
1	1	.	25	21	.	49	14	.	73	22	10
2	2	.	26	17	.	50	19	10	74	2	.
3	3	.	27	1	.	51	21	10	75	14	.
4	4	.	28	23	.	52	13	13	76	3	0
5	5	.	29	18	11	53	15	.	77	6	.
6	6	21	30	12	0	54	20	21	78	7	.
7	7	.	31	10	.	55	8	.	79	5	21
8	8	11	32	5	.	56	18	.	80	15	.
9	9	.	33	16	.	57	11	0	81	0	.
10	10	.	34	8	.	58	5	.	82	18	.
11	11	13	35	0	0	59	4	.	83	23	.
12	12	.	36	20	21	60	7	0	84	10	21
13	13	.	37	3	10	61	1	.	85	21	.
14	14	0	38	22	.	62	9	.	86	16	.
15	15	0	39	6	21	63	12	.	87	20	11
16	16	0	40	11	.	64	0	.	88	4	0
17	17	10	41	19	.	65	2	.	89	17	10
18	18	21	42	15	.	66	6	11	90	12	.
19	19	.	43	2	10	67	17	.	91	19	.
20	20	.	44	7	.	68	10	21	92	8	0
21	21	.	45	14	.	69	22	.	93	9	.
22	22	.	46	9	0	70	16	.	94	11	.
23	23	10	47	13	.	71	3	.	95	1	13

Ⅴ. 전체 라운드 차분 공격

앞의 Ⅲ, Ⅳ절을 기초로 하여 이 절에서는 P-해쉬 함수를 전체 라운드에 대해 공격한다. 이 공격은 두 메시지의 차분을 이용하여 충돌 쌍을 찾는 것이다.

〔표 7〕은 전체 라운드에 대한 차분 공격을 나타낸 것이다. 그리고 다음에 나오는 소절들을 통해 차분이 발생하는 단계들을 부분별로 분석한다.

4.1 〔표 7〕의 11~21 단계의 분석

11~21단계는 1라운드이므로 f_0함수를 쓴다.

〔표 7〕의 값들은 다음과 같이 계산된다.

$$\begin{aligned}\triangle E_{11} = [&f_0(E_{10}, A_{10}, B_{10}, C_{10}, D_{10}) + X_{11} + K_1]^{\ll S_{11}=13} \\ &\oplus [f_0(E_{10}, A_{10}, B_{10}, C_{10}, D_{10}) \\ &+ (X_{11} \oplus 1^{\ll 30}) K_1]^{\ll S_{11}=13} = 1^{\ll 11} \quad (p = 1/2)\end{aligned}$$

위의 등식은 식 (9)에 의해 확률 1/2로 성립한다.

$$\begin{aligned}\triangle D_{12} = [&f_0(D_{11}, E_{11}, A_{11}, B_{11}, C_{11}) + X_{12} + K_1]^{\ll S_{12}} \\ &\oplus [f_0(D_{11}, E_{11} \oplus 1^{\ll 11}, A_{11}, B_{11}, C_{11}) \\ &+ X_{12} + K_1]^{\ll S_{12}} = 0 \quad (p = 1/2)\end{aligned}$$

위의 등식은 〔표 3〕의 2에 의해 확률 1/2로 성립한다.

$$\begin{aligned}\triangle C_{13} = [&f_0(C_{12}, D_{12}, E_{12}, A_{12}, B_{12}) + X_{13} + K_1]^{\ll S_{13}} \\ &\oplus [f_0(C_{12}, D_{12}, E_{12} \oplus 1^{\ll 21}, A_{12}, B_{12}) \\ &+ X_{13} + K_1]^{\ll S_{13}} = 0 \quad (p = 3/4)\end{aligned}$$

위의 등식은 〔표 3〕의 3에 의해 확률 3/4으로 성립한다.

$$\begin{aligned}\triangle B_{14} = [&f_0(B_{13}, C_{13}, D_{13}, E_{13}, A_{13}) + X_{14} + K_1]^{\ll S_{14}=0} \\ &\oplus [f_0(B_{13}, C_{13}, D_{13}, E_{13} \oplus 1^{\ll 21}, A_{13}) \\ &+ X_{14} + K_1]^{\ll S_{14}=0} = 0 \quad (p = 3/4)\end{aligned}$$

위의 등식은 〔표 3〕의 4에 의해 확률 3/4으로 성립한다.

$$\begin{aligned}\triangle A_{15} = [&f_0(A_{14}, B_{14}, C_{14}, D_{14}, E_{14}) + X_{15} + K_1]^{\ll S_{15}=0} \\ &\oplus [f_0(A_{14}, B_{14}, C_{14}, D_{14}, E_{14} \oplus 1^{\ll 21}) \\ &+ X_{15} + K_1]^{\ll S_{15}=0} = 1^{\ll 21} \quad (p = 1/2)\end{aligned}$$

위의 등식은 식 (9)과 〔표 3〕의 5에 의해 확률 1/2로 성립한다.

$$\begin{aligned}\triangle E_{16} = [&f_0(E_{15}, A_{15}, B_{15}, C_{15}, D_{15}) + X_{16} + K_1]^{\ll S_{16}=0} \\ &\oplus [f_0(E_{15} \oplus 1^{\ll 21}, A_{15} \oplus 1^{\ll 21}, B_{15}, C_{15}, D_{15}) \\ &+ X_{16} + K_1]^{\ll S_{16}=0} = 1^{\ll 21} \quad (p = 1/4)\end{aligned}$$

위의 등식은 식 (9)과 〔표 3〕의 6에 의해 확률 1/4로 성립한다.

$$\begin{aligned}\triangle D_{17} = [&f_0(D_{16}, E_{16}, A_{16}, B_{16}, C_{16}) + X_{17} + K_1]^{\ll S_{17}=10} \\ &\oplus [f_0(D_{16}, E_{16} \oplus 1^{\ll 21}, A_{16} \oplus 1^{\ll 31}, B_{16}, C_{16}) \\ &+ X_{17} + K_1]^{\ll S_{17}=10} = 0 \quad (p = 3/8)\end{aligned}$$

위의 등식은 〔표 3〕의 2, 3에 의해 확률 3/8로 성립한다.

$$\begin{aligned}\triangle C_{18} = [&f_0(C_{17}, D_{17}, E_{17}, A_{17}, B_{17}) + X_{18} + K_1]^{\ll S_{18}=21} \\ &\oplus [f_0(C_{17}, D_{17}, E_{17} \oplus 1^{\ll 31}, A_{17} \oplus 1^{\ll 31}, B_{17}) \\ &+ X_{18} + K_1]^{\ll S_{18}=21} = 0 \quad (p = 3/4)\end{aligned}$$

위의 등식은 〔표 3〕의 10에 의해 확률 3/4으로 성립한다.

$$\begin{aligned}\triangle B_{19} = [&f_0(B_{18}, C_{18}, D_{18}, E_{18}, A_{18}) + X_{19} + K_1]^{\ll S_{19}} \\ &\oplus [f_0(B_{18}, C_{18}, D_{18}, E_{18} \oplus 1^{\ll 31}, A_{18} \oplus 1^{\ll 31}) \\ &+ X_{19} + K_1]^{\ll S_{19}} = 0 \quad (p = 1/4)\end{aligned}$$

위의 등식은 〔표 3〕의 11에 의해 확률 1/4으로 성립한다.

$$\begin{aligned}\triangle A_{20} = [&f_0(A_{19}, B_{19}, C_{19}, D_{19}, E_{19}) + X_{20} + K_1]^{\ll S_{20}} \\ &\oplus [f_0(A_{19} \oplus 1^{\ll 31}, B_{19}, C_{19}, D_{19}, E_{19} \oplus 1^{\ll 31}) \\ &+ (X_{20} \oplus 1^{\ll 31}) + K_1]^{\ll S_{20}} = 0 \quad (p = 1/2)\end{aligned}$$

위의 등식은 식 (8)와 〔표 3〕의 9에 의해 확률 1/2로 성립한다.

$$\begin{aligned}\triangle E_{21} = [&f_0(E_{20}, A_{20}, B_{20}, C_{20}, D_{20}) + X_{21} + K_1]^{\ll S_{21}} \\ &\oplus [f_0(E_{20} \oplus 1^{\ll 31}, A_{20}, B_{20}, C_{20}, D_{20}) \\ &+ X_{21} + K_1]^{\ll S_{21}} = 0 \quad (p = 1/2)\end{aligned}$$

위의 등식은 〔표 3〕의 1에 의해 확률 1/2로 성립한다.

4.2 〔표 7〕의 36~42 단계의 분석

앞에서 분석한 방법과 마찬가지 방법으로 등식이 성립할 확률을 계산한다. 계산과정은 다음과 같다.

〔표 7〕 전체 라운드의 차분 공격

단계	△A	△B	△C	△D	△E	△X	확률
0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1
3	0	0	0	0	0	0	1
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	1
8	0	0	0	0	0	0	1
9	0	0	0	0	0	0	1
10	0	0	0	0	0	0	1
11	0	0	0	0	1 (11	1 (30	1/2
12	0	0	0	0	1 (21	0	1/2
13	0	0	0	0	1 (21	0	3/4
14	0	0	0	0	1 (21	0	3/4
15	1 (21	0	0	0	1 (21	0	1/2
16	1 (31	0	0	0	1 (21	0	1/4
17	1 (31	0	0	0	1 (31	0	3/8
18	1 (31	0	0	0	1 (31	0	3/4
19	1 (31	0	0	0	1 (31	0	1/4
20	0	0	0	0	1 (31	1 (31	1/2
21	0	0	0	0	0	0	1/2
22	0	0	0	0	0	0	1
23	0	0	0	0	0	0	1
24	0	0	0	0	0	0	1
25	0	0	0	0	0	0	1
26	0	0	0	0	0	0	1
27	0	0	0	0	0	0	1
28	0	0	0	0	0	0	1
29	0	0	0	0	0	0	1
30	0	0	0	0	0	0	1
31	0	0	0	0	0	0	1
32	0	0	0	0	0	0	1
33	0	0	0	0	0	0	1
34	0	0	0	0	0	0	1
35	0	0	0	0	0	0	1
36	0	0	0	0	1 (20	1 (31	1
37	0	0	0	1 (30	1 (30	0	1/2
38	0	0	0	1 (8	1 (30	0	3/8
39	0	0	0	1 (8	1 (30	0	25/64
40	0	0	0	1 (8	1 (30	1 (30	15/128
41	0	0	0	1 (8	0	0	25/64
42	0	0	0	0	0	0	5/8
43	0	0	0	0	0	0	1
44	0	0	0	0	0	0	1
45	0	0	0	0	0	0	1
46	0	0	0	0	0	0	1
47	0	0	0	0	0	0	1
48	0	0	0	0	0	0	1
49	0	0	0	0	0	0	1
50	0	0	0	0	0	0	1
51	0	0	0	0	0	0	1
52	0	0	0	0	0	0	1
53	0	0	0	0	0	0	1
54	0	1 (20	0	0	0	1 (31	1
55	0	1 (30	0	0	0	0	1/2
56	0	1 (30	0	0	0	0	1/2
57	0	1 (30	0	0	0	1 (30	1/4
58	0	1 (30	0	0	0	0	1/2
59	0	0	0	0	0	0	1/2
60	0	0	0	0	0	0	1
61	0	0	0	0	0	0	1
62	0	0	0	0	0	0	1
63	0	0	0	0	0	0	1
64	0	0	0	0	0	0	1
65	0	0	0	0	0	0	1
66	0	0	0	0	0	0	1
67	0	0	0	0	0	0	1
68	0	0	0	0	0	0	1
69	0	0	0	0	0	0	1
70	0	0	0	0	0	0	1
71	0	0	0	0	0	0	1
72	0	0	0	0	0	0	1
73	0	0	0	0	0	0	1
74	0	0	0	0	0	0	1
75	0	0	0	0	0	0	1
76	0	0	0	0	0	0	1
77	0	0	0	0	0	0	1
78	0	0	0	0	0	0	1
79	0	0	0	0	0	0	1
80	0	0	0	0	0	0	1
81	0	0	0	0	0	0	1
82	0	0	0	0	0	0	1
83	0	0	0	0	0	0	1
84	0	0	0	0	0	0	1
85	0	0	0	0	0	0	1
86	0	0	0	0	0	0	1
87	0	0	0	1 (10	0	1 (31	1
88	0	0	1 (10	1 (20	0	0	1/2
89	0	1 (20	1 (20	1 (20	0	0	5/16
90	0	1 (30	1 (20	1 (20	0	0	5/8
91	0	1 (30	1 (20	1 (20	0	0	25/64
92	0	1 (30	1 (20	0	0	0	15/64
93	0	1 (30	0	0	0	0	25/64
94	0	0	0	0	0	1 (30	3/16
95	0	0	0	0	0	0	1

□ : 각 단계마다 갱신되는 부분

$$\triangle E_{36}=[f_1(E_{35},A_{35},B_{35},C_{35},D_{35})+X_{20}+K_2]^{\ll S_{36}=21}$$
$$\oplus\ [f_1(E_{35},A_{35},B_{35},C_{35},D_{35})+(X_{20}\oplus 1^{\ll 31})$$
$$+K_2]^{\ll S_{36}=21}=1^{\ll 20} \qquad (p=1)$$

$$\triangle D_{37}=[f_1(D_{36},E_{36},A_{36},B_{36},C_{36})+X_3+K_2]^{\ll S_{37}=10}$$
$$\oplus\ [f_1(D_{36},E_{36}\oplus 1^{\ll 20},A_{36},B_{36},C_{36})+X_3$$
$$+K_2]^{\ll S_{37}=10}=1^{\ll 30} \qquad (p=1/2)$$

위의 등식은 식 (9)와 〔표 3〕의 13에 의해 확률 1/2로 성립한다.

$$\triangle C_{38}=[f_1(C_{37},D_{37},E_{37},A_{37},B_{37})+X_{22}+K_2]^{\ll S_{38}}$$
$$\oplus[f_1(C_{37},D_{37}\oplus 1^{\ll 30},E_{37}\oplus 1^{\ll 30},A_{37},B_{37})$$
$$+X_{22}+K_2]^{\ll S_{38}}=0 \qquad (p=3/8)$$

$$\triangle B_{39}=[f_1(B_{38},C_{38},D_{38},E_{38},A_{38})+X_6+K_2]^{\ll S_{39}=21}$$
$$\oplus\ [f_1(B_{38},C_{38},D_{38}\oplus 1^{\ll 8},E_{38}\oplus 1^{\ll 30},A_{38})$$
$$+X_6+K_2]^{\ll S_{39}=21}=0 \qquad (p=25/64)$$

위의 등식은 〔표 3〕의 14, 15에 의해 확률 25/64로 성립한다.

$$\triangle A_{40}=[f_1(A_{39},B_{39},C_{39},D_{39},E_{39})+X_{11}+K_2]^{\ll S_{40}}$$
$$\oplus\ [f_1(A_{39},B_{39},C_{39},D_{39}\oplus 1^{\ll 8},E_{39}\oplus 1^{\ll 30})$$
$$+(X_{11}\oplus 1^{\ll 30})+K_2]^{\ll S_{40}}=0 \qquad (p=15/128)$$

위의 등식은 식 (10)과 〔표 3〕의 15, 16에 의해 확률 15/128로 성립한다.

$$\triangle E_{41}=[f_1(E_{40},A_{40},B_{40},C_{40},D_{40})+X_{19}+K_2]^{\ll S_{41}}$$
$$\oplus\ [f_1(E_{40}\oplus 1^{\ll 30},A_{40},B_{40},C_{40},D_{40}\oplus 1^{\ll 8})$$
$$+X_{19}+K_2]^{\ll S_{41}}=0 \qquad (p=25/64)$$

$$\triangle D_{42}=[f_1(D_{41},E_{41},A_{41},B_{41},C_{41})+X_{15}+K_2]^{\ll S_{42}}$$
$$\oplus\ [f_1(D_{41}\oplus 1^{\ll 8},E_{41},A_{41},B_{41},C_{41})$$
$$+X_{15}+K_2]^{\ll S_{42}}=0 \qquad (p=5/8)$$

4.3 〔표 7〕의 54~59 단계의 분석

앞에서 분석한 방법과 마찬가지 방법으로 등식이 성립할 확률을 계산한다. 계산과정은 다음과 같다.

$$\triangle B_{54}=[f_2(B_{53},C_{53},D_{53},E_{53},A_{53})+X_{20}+K_3]^{\ll S_{54}=21}$$
$$\oplus\ [f_2(B_{53},C_{53},D_{53},E_{53},A_{53})+(X_{20}\oplus 1^{\ll 31})$$
$$+K_3]^{\ll S_{54}=21}=1^{\ll 20} \qquad (p=1)$$

$$\triangle A_{55}=[f_2(A_{54},B_{54},C_{54},D_{54},E_{54})+X_8+K_3]^{\ll S_{55}}$$
$$\oplus\ [f_2(A_{54},B_{54}\oplus 1^{\ll 20},C_{54},D_{54},E_{54})+X_8$$
$$+K_3]^{\ll S_{55}}=0 \qquad (p=1/2)$$

$$\triangle E_{56}=[f_2(E_{55},A_{55},B_{55},C_{55},D_{55})+X_{18}+K_3]^{\ll S_{56}}$$
$$\oplus\ [f_2(E_{55},A_{55},B_{55}\oplus 1^{\ll 30},C_{55},D_{55})$$
$$+X_{18}+K_3]^{\ll S_{56}}=0 \qquad (p=1/2)$$

$$\triangle D_{57}=[f_2(D_{56},E_{56},A_{56},B_{56},C_{56})+X_{11}+K_3]^{\ll S_{57}=0}$$
$$\oplus\ [f_2(D_{56},E_{56},A_{56},B_{56}\oplus 1^{\ll 30},C_{56})$$
$$+(X_{11}\oplus 1^{\ll 30})+K_3]^{\ll S_{57}=0}=0 \qquad (p=1/4)$$

$$\triangle C_{58}=[f_2(C_{57},D_{57},E_{57},A_{57},B_{57})+X_5+K_3]^{\ll S_{58}}$$
$$\oplus\ [f_2(C_{57},D_{57},E_{57},A_{57},B_{57}\oplus 1^{\ll 30})$$
$$+X_5+K_3]^{\ll S_{58}}=0 \qquad (p=1/2)$$

$$\triangle B_{59}=[f_2(B_{58},C_{58},D_{58},E_{58},A_{58})+X_4+K_3]^{\ll S_{59}}$$
$$\oplus\ [f_2(B_{58}\oplus 1^{\ll 30},C_{58},D_{58},E_{58},A_{58})$$
$$+X_4+K_3]^{\ll S_{59}}=0 \qquad (p=1/2)$$

4.4 〔표 7〕의 87~95 단계의 분석

앞에서 분석한 방법과 마찬가지 방법으로 등식이 성립할 확률을 계산한다. 계산과정은 다음과 같다.

$$\triangle D_{87}=[f_1(D_{86},E_{86},A_{86},B_{86},C_{86})+X_{20}+K_4]^{\ll S_{87}=11}$$
$$\oplus\ [f_1(D_{86},E_{86},A_{86},B_{86},C_{86})+(X_{20}\oplus 1^{\ll 31})$$
$$+K_4]^{\ll S_{87}=11}=1^{\ll 10} \qquad (p=1)$$

$$\triangle C_{88}=[f_1(C_{87},D_{87},E_{87},A_{87},B_{87})+X_4+K_4]^{\ll S_{88}=0}$$
$$\oplus\ [f_1(C_{87},D_{87}\oplus 1^{\ll 10},E_{87},A_{87},B_{87})$$
$$+X_4+K_4]^{\ll S_{88}=0}=1^{\ll 10} \qquad (p=1/2)$$

위의 등식은 식 (9)와 〔표 3〕의 13에 의해 확률

1/2로 성립한다.

$$\begin{aligned}\triangle B_{89} &= [f_1(B_{88}, C_{88}, D_{88}, E_{88}, A_{88}) + X_{17} + K_4]^{\lll S_{89}=10} \\ &\oplus [f_1(B_{88}, C_{88}\oplus 1^{\lll 10}, D_{88}\oplus 1^{\lll 20}, E_{88}, A_{88}) \\ &\quad + X_{17} + K_4]^{\lll S_{89}=10} = 1^{\lll 20} \qquad (p = 5/16)\end{aligned}$$

위의 등식은 식 (9)와 [표 3]의 13, 14에 의해 확률 5/16로 성립한다.

$$\begin{aligned}\triangle A_{90} &= [f_1(A_{89}, B_{89}, C_{89}, D_{89}, E_{89}) + X_{12} + K_4]^{\lll S_{90}} \\ &\oplus [f_1(A_{89}, B_{89}\oplus 1^{\lll 20}, C_{89}\oplus 1^{\lll 20}, D_{89}\oplus 1^{\lll 20}, \\ &\quad E_{89}) + X_{12} + K_4]^{\lll S_{90}} = 0 \qquad (p = 5/8)\end{aligned}$$

$$\begin{aligned}\triangle E_{91} &= [f_1(E_{90}, A_{90}, B_{90}, C_{90}, D_{90}) + X_{19} + K_4]^{\lll S_{91}} \\ &\oplus [f_1(E_{90}, A_{90}, B_{90}\oplus 1^{\lll 30}, C_{90}\oplus 1^{\lll 20}, D_{90}\oplus 1^{\lll 20}) \\ &\quad + X_{19} + K_4]^{\lll S_{91}} = 0 \qquad (p = 25/64)\end{aligned}$$

$$\begin{aligned}\triangle D_{92} &= [f_1(D_{91}, E_{91}, A_{91}, B_{91}, C_{91}) + X_8 + K_4]^{\lll S_{92}=0} \\ &\oplus [f_1(D_{91}\oplus 1^{\lll 20}, E_{91}, A_{91}, B_{91}\oplus 1^{\lll 30}, C_{91}\oplus 1^{\lll 20}) \\ &\quad + X_8 + K_4]^{\lll S_{92}=0} = 0 \qquad (p = 15/64)\end{aligned}$$

$$\begin{aligned}\triangle C_{93} &= [f_1(C_{92}, D_{92}, E_{92}, A_{92}, B_{92}) + X_9 + K_4]^{\lll S_{93}} \\ &\oplus [f_1(C_{92}\oplus 1^{\lll 20}, D_{92}, E_{92}, A_{92}, B_{92}\oplus 1^{\lll 30}) \\ &\quad + X_9 + K_4]^{\lll S_{93}} = 0 \qquad (p = 25/64)\end{aligned}$$

$$\begin{aligned}\triangle B_{94} &= [f_1(B_{93}, C_{93}, D_{93}, E_{93}, A_{93}) + X_{11} + K_4]^{\lll S_{94}} \\ &\oplus [f_1(B_{93}\oplus 1^{\lll 30}, C_{93}, D_{93}, E_{93}, A_{93}) \\ &\quad + (X_{11}\oplus 1^{\lll 30}) + K_4]^{\lll S_{94}} = 0 \qquad (p = 3/16)\end{aligned}$$

$$\begin{aligned}\triangle A_{95} &= [f_1(A_{94}, B_{94}, C_{94}, D_{94}, E_{94}) + X_1 + K_4]^{\lll S_{95}=13} \\ &\oplus [f_1(A_{94}, B_{94}, C_{94}, D_{94}, E_{94}) + X_1 + K_4]^{\lll S_{95}=13} \\ &= 0 \qquad (p = 1)\end{aligned}$$

이상의 확률을 모두 곱하면 약 $2^{-37.13}$이 되며, 따라서 PKC'98에 제안된 해쉬 함수의 충돌 쌍을 생일 공격의 확률보다 큰 $2^{-37.13}$의 확률로 찾을 수 있다.

V. 실제 충돌쌍 찾기

충돌이 발생할 평균 확률은 $2^{-37.13}$이다. 이를 구현할 경우 압축 함수를 평균 $2^{-38.13}$ 번을 돌려야 한다. 왜냐하면, 충돌 쌍은 두 메시지에 대해 고려되어야 하기 때문이다. $2^{-38.13}$번의 계산을 하는 데에는 1초에 공격 알고리즘을 10^6번을 실행하는 컴퓨터로 대략 76.45시간이 걸린다. 실제로, 충돌 쌍을 찾기 위해, 펜티엄Ⅲ 800Mhz 컴퓨터 10대를 가지고 visual C 6.0으로 공격 알고리즘을 구현(메시지 블록은 랜덤 값으로 표현하였고, 메시지의 일부를 고정시켜 원하는 쉬프트 값을 갖도록 하였음)하였다. 실제적으로 우리는 대략 10시간만에 한 컴퓨터에서 충돌 쌍을 찾을 수 있었다. 이 결과 찾은 충돌 쌍은 다음과 같다.

X_0=0xdf407f1a, X_1=0x99c0464a,
X_2=0x3380a1fa, X_3=0x0d40be50,
X_4=0x6580c1c0, X_5=0xb8803020,
X_6=0xf5c09a9e, X_7=0x388077d5,
X_8=0x1f005106, X_9=0xb080db94,
X_{10}=0xb700244c, X_{11}=0x3480cc5e,
X_{12}=0xb5c00895, X_{13}=0xa9405c59,
X_{14}=0x28c04748, X_{15}=0xba008ecb

$\hat{X}_{11} = X_{11}\oplus 1^{\lll 30}$, $\hat{X}_i = X_i$ ($i \neq 11$)

이때, 메시지 $X, \hat{X}$는 다음과 같이 동일한 해쉬값을 갖는다.

0xfe684dca 0x33524aa4 0x15ce9f59
0xd200e689 0x7b01f656

Ⅵ. 결 론

본 논문의 분석을 통해 나타난 P-해쉬 함수의 문제점은 메시지에 의존한 쉬프트 값의 사용에 있다. 즉, P-해쉬 함수와 같은 구조를 갖는 해쉬 함수의 안전성을 위해서는 쉬프트 값을 적절하게 고정시켜 놓아야 한다.

FSE 2002에서 한대완 등은 P-해쉬 함수의 부울 함수가 당초 설계자의 의도와는 달리 일부 부울 함수가 SAC을 만족하지 않음을 지적하였다. 그리고 설계자의 의도에 맞게 모든 부울 함수가 SAC의 성질을 만족한다는 가정 하에, 2^{-30}의 확률로 충돌 쌍을 찾는 공격 방법을 제안하였다.

그러나 본 논문에서는 P-해쉬 함수에 사용된 부울 함수(f_0, f_1은 SAC을 만족하지 않음)를 그대로 쓴 경우에도 $2^{-37.13}$의 확률로 충돌 쌍을 찾을 수 있음을 보였다. 그리고, P-해쉬 함수의 문제점을 지적하였다.

그러므로, 해쉬 함수 설계자들은 해쉬 함수의 설계시 쉬프트 값을 결정하는데 있어서 신중을 기해야 할 것이다.

참 고 문 헌

[1] S.U. Shin, K.H. Rhee, D.H Ryu, S.J Lee, "A New Hash Function Based on MDx-family and Its Application to MAC", *Public Key Cryptography '98*, pp. 234~246. 1998.

[2] D.W. Han, S.W. Park, S.T. Chee, "Cryptanalysis of a Hash Function Proposed at PKC'98", *Fast Software Encryption 2002*, pp. 246~256, Feb. 2002.

[3] R. Rivest, "The MD4 message digest algorithm", *RFC 1320, Internet Activities Board, Internet Privacy Task Force*, Apr. 1992.

[4] R. Rivest, "The MD5 message digest algorithm", *RFC 1321, Internet Activities Board, Internet Privacy Task Force*, Apr. 1992.

[5] Y. Zheng, J. Pieprzyk and J. Sebberry, "HAVAL-A one-way hashing algorithm with variable length of output", *Advances in Cryptology-Auscrypt'92*, LNCS 718, Springer-Verlag, 1993, pp. 83~104.

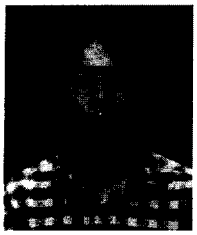
[6] *Federal Information Processing Standards Publication 180-1,1995*, April 17.

[7] H. Dobbertin, "Cryptanalysis of MD4", *Fast Software Encryption*, LNCS 1039, Springer-Verlag, 1996, pp. 53~69.

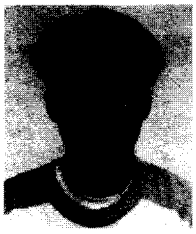
[8] H. Dobbertin, "Cryptanalysis of MD5 Compress", May. 1996.

[9] P.R. Kasselman and W.T. Penzhorn, "Cryptanalysis of reduced version of HAVAL", *Electonics Letters* 6th January 2000 Vol. 36, No. 1 pp. 30~31.

〈著者紹介〉

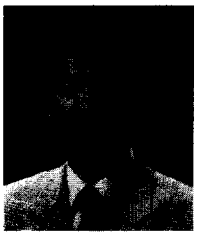


장 동 훈 (Dong-Hoon Chang)
2001년 2월 : 고려대학교 수학과 학사
2001년 3월~현재 : 고려대학교 정보보호대학원 석사 과정
〈관심분야〉 블록 암호, 해쉬 함수와 MAC

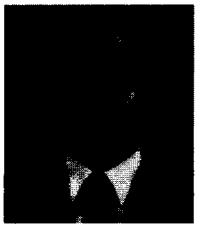


성 재 철 (Jae-Chul Sung)
1997년 8월 : 고려대학교 수학과 학사
1999년 8월 : 고려대학교 수학과 석사
1999년 9월~현재 : 고려대학교 수학과 박사 과정
〈관심분야〉 블록 암호 및 메시지 인증 코드의 설계 및 분석

성 수 학 (Soo-Hak Sung) 정회원
1982년 2월 : 경북대학교 수학과 학사
1985년 2월 : KAIST 응용수학과 석사
1988년 2월 : KAIST 응용수학과 박사
1988년~1991년 : 한국전자통신연구원 선임 연구원
1991년~현재 : 배재대학교 전산정보수학과 교수
〈관심분야〉 암호 이론, 암호 프로토콜



이 상 진 (Sang-Jin Lee) 정회원
1987년 2월 : 고려대학교 수학과 학사
1989년 2월 : 고려대학교 수학과 석사
1994년 8월 : 고려대학교 수학과 박사
1989년 2월~1999년 2월: 한국전자통신연구원 선임 연구원
1999년 3월~현재 : 고려대학교 자연과학대학 부교수, 고려대학교 정보보호대학원 겸임교수, 고려대학교 정보보호기술연구센터 연구실장
〈관심분야〉 블록 암호 및 스트림 암호의 분석 및 설계, 암호 프로토콜, 공개키 암호 알고리즘의 분석



임 종 인 (Jong-in Lim) 정회원
1980년 2월 : 고려대학교 수학과 학사
1982년 2월 : 고려대학교 수학과 석사
1986년 2월 : 고려대학교 수학과 박사
1999년 2월~현재 : 고려대학교 자연과학대학 정교수, 한국통신정보보호학회 편집위원장 고려대학교 정보보호대학원 원장, 고려대학교 정보보호기술연구센터 센터장
〈관심분야〉 블록 암호 및 스트림 암호의 분석 및 설계, 암호 프로토콜, 공개키 암호 분석