해쉬 함수의 근사적 모델과 연쇄패턴

Approximated Model and Chaining Pattern of Hash Functions

이 선 영
Sun Young Lee

요 약

MD4를 기반으로 하는 MDx계열 해쉬 함수는 입력 워드의 치환, 순환 시프트, 변수의 연쇄, 비선형 함수 등을 이용하여 구성된다. 그러나, 그 구성 방법에 대한 이론적, 실험적 배경은 알려져 있지 않고, 평가를 위한 일반적인 모델도 알려져 있지 않다. 본 논문에서는 해쉬 함수의 설계와 평가를 위하여 해쉬 함수를 일반화하기 위한 근사 모델을 제안하고, MD5에 근사 모델을 적용하여 입력 차분의 확산을 고찰하였다. 그 결과, MD5의 약점이 완전 확산을 제공하지 않는데 있다는 것을 확인하였다. 제안된 근사 모델에서 완전 확산을 제공하기 위하여 해쉬 함수 내에 여러 가지 연쇄 패턴을 이용하는 다중 연쇄 패턴을 제안하고, 여분의 계산과 메모리 없이 완전 확산을 제공함을 MD5를이용한 실험을 통하여 보이고, 그 차분 특성이 갖는 확률이 MD5보다 작음을 증명하였다.

Abstract

The evaluation of MDx family hash functions such as MD5 is difficult because the design background or a generalized model is unknown. In this paper, an approximated model is proposed to generalize hash functions. The diffusion of a input difference is tested by an approximated model for MD5. The results show that MD5 does not provide perfect diffusion, so MD5 is weak against some attacks. We propose a multiple chaining pattern which provides perfect diffusion in approximated model of hash function without extra calculation or memory. And We show the probability of differential characteristics of our proposal.

☞ Keyword : Hash Function, Perfect diffusion, security, approximated model, chaining pattern, differential characteristic

1. 서 론

임의 길이의 입력을 고정 길이의 해쉬값으로 출력하는 해쉬 함수는 전자 서명, 인증 코드등과 같이 메시지 또는 사용자를 인증하는데 효율성과 안전성에서 매우 중요한 역할을 하는 함수이다. 현재 가장 실용적으로 사용되고 있는 해쉬 함수는 MD5[1], SHA-1[2], RIPEMD-160[3]과 같이 MD4 [4]에 기반을 둔 MDx 계열 해쉬 함수이다. MDx 계열 해쉬 함수들은 MD4를 변형한 것들인데, 그 변형을 선택한 이유에 대한 이론적, 실험적 근거나 설명이 없고, 블록 암호에 있어서의 Feistel, SPN 구조와 같은 일반화된 모델이 없어 그 안전성을 분석, 평가하는 것이 매우 어렵다. 그리고, 안전한 해쉬 함수를 설계하기 위한 근거도 알려져 있지 않다. MDx 계열 해쉬 함수에 관한 안전성은 해당 해쉬 함수에 대한 충돌을 찾는 공격의 존재로서 평가하는데 입력의 작은 변화가 반복적으로 수행되는 해쉬 연산 과정에서 없어지는 점을 이용한 차분 공격이 해쉬 함수에 있어서의 가장 강력한 공격으로 알려져 있다[4-7].

차분 공격에 강한 해쉬 함수가 되기 위해서는 해쉬 연산이 이루어지는 각 단계에서 입력의 차분이 계속 유지 되도록하는 것이 바람직하다. 이를 완전 확산(perfect diffusion)이라 한다[8]. 완전 확산에 대한 평가는 해쉬 함수의 설계 단계에서 고려되어야 할 부분이나, 대부분의 해쉬 함수에서 설계의 근거가 제시되어 있지 않으므로 실제 어떤 기준에 의하여 해쉬 함수가 설계되었는지 알기 어렵다. 본 논문에서는 해쉬 함수를 설계할 때 고려되어야 할 것들 중 완전 확산을 평가하기 위한 방법으로서

* 정 회 원 : 순천향대학교 정보보호학과 전임강사
sunlee@sch.ac.kr(제1저자)
[2005/08/23 투고 - 2005/09/12 심사 - 2005/11/01 심사완료]

해쉬 함수의 근사 모델을 제안한다. 근사 모델을 MD5에 적용하여 MD5의 약점을 이해하고, 그 원인이 연쇄 변수의 사용 순서가 적절하지 않은 데 있다는데 주목하여 각 단계에서 다양한 연쇄 패턴을 사용하는 방법을 제안하고 여분의 메모리와 작업 없이 완전 확산을 제공할 수 있음을 보인다.

본 논문은 제 2장 해쉬 함수의 개요, 제 3장 제안하는 근사 모델, 제4장 다중 연쇄 패턴의 제안 및 분석, 제5장 결론으로 구성된다.

2. 해쉬 함수

2.1 해쉬 함수의 개요

MDx 계열 해쉬 함수는 MD4의 발표 이후 발전해 왔고, 또한 많은 분석이 이루어져 왔다. 본 논문에서는 여러 가지 MDx 계열 해쉬 함수 중에서 이해하기 쉬운 MD5로 해쉬 함수의 구조를 설명한다.

MD5의 압축 함수를 위해 변수와 연산자를 다음과 같이 정의 한다.

$X \wedge Y, X \vee Y, X \oplus Y$: 각각 비트단위의 AND, OR, XOR

$X^{<<<s}$: X의 값을 s 비트만큼 왼쪽으로 순환 시프트

$\overline{X}$: X의 2의 보수

+ : mod 232 덧셈

$V := E$: 변수 V에 식 E의 값을 할당

$Y^{(i)}$: i번째 단계에서 사용되는 Y의 값

MD5의 압축 함수는 4개의 연쇄 변수 (A, B, C, D)와 16개의 입력 워드 $(X^{(0)}, X^{(1)}, \cdots, X^{(15)})$로 구성된다. 이때, 각 연쇄 변수와 입력 워드는 32비트이다. MD5의 압축 함수는 총 4라운드이며, 각 라운드는 16단계로 구성된다. 각 단계에서는 다음과 같은 연산이 수행된다.

for i=1 to 64

$$A^{(i)} := B^{(i-1)} + (A^{(i-1)} + f^{(i)}(B^{(i-1)}, C^{(i-1)}, D^{(i-1)}) + X^{(j)} + K^{(i)})^{<<<s}.$$

단,

- $X^{(j)}$는 1입력 메시지 워드 $(0 \le j \le 15)$
- $K^{(i)}$는 각 단계에서 유일하게 정해진 상수
- s 는 순환 시프트의 양.
- $f^{(i)}(a,b,c)$는 비선형 함수

64단계 이후, 4개의 연쇄 변수 (A, B, C, D)를 연결하여 128비트의 해쉬값을 출력한다.

2.2 해쉬 함수에서의 차분 공격

차분 공격에 기반을 둔 MD4에 대한 공격이 [6]에서 제안되었다. MDx 계열 해쉬 함수에 서는 압축함수에서 입력의 값을 고정시켜 두고 충돌을 발견하는 것이 가장 일반적인 방법이다. $\tilde{X} = (\tilde{X}^{(0)}, \tilde{X}^{(1)}, \cdots, \tilde{X}^{(15)})$는 다음과 같이 정의 한다.

$$\tilde{X}^{(j)} = X^{(j)} \quad , (j \neq k)$$
$$\tilde{X}^{(k)} = X^{(k)} + 1.$$

따라서, 입력의 차분 $\triangle X^{(i)}$은

$$\triangle X^{(i)} = X^{(i)} - \tilde{X}^{(i)}$$

이 된다. i단계 후의 연쇄 변수를 $(A^{(i)}, B^{(i)}, C^{(i)}, D^{(i)})$라 하고 $\tilde{X}^{(i)}$을 입력으로 한, i단계에서의 연쇄 변수를 $(A'^{(i)}, B'^{(i)}, C'^{(i)}, D'^{(i)})$라 하면, 출력의 차분은 다음과 같다.

$$\Delta^{(i)} = (A^{(i)} - A'^{(i)}, B^{(i)} - B'^{(i)}, C^{(i)} - C'^{(i)}, D^{(i)} - D'^{(i)}).$$

입력 차분$\triangle X^{(i)}$는 반복적인 연산에 의해 다음 식과 같이 보상되기도 한다.

$$\Delta^{(i)} = (0, \alpha^{<<<m}, \beta^{<<<n}, 0),$$
$$f^{(i)}(B^{(i)}, C^{(i)}, D^{(i)}) = f^{(i)}(B'^{(i)}, C'^{(i)}, D^{(i)}). \quad (1)$$

단, $0 \le \alpha \le 2^{31}$이고, $0 \le m, n \le 32$이다.

입력 차분은 반복적 연산에 의해 보상되어 충돌을 발견하도록 하기 때문에 공격에서 식(1)과 같은 차분$\Delta^{(i)}$를 검색하는 것은 매우 중요하다.

3. 근사 모델

3.1 덧셈 연산

$(A + 2^{31}) \bmod 2^{32} = A \oplus 2^{31}$이라는 사실에 의해 2^{32} 법가산은 배타적 논리합(XOR)으로 대체할 수 있다. 즉, 캐리가 없는 덧셈의 결과는 XOR 연산 결과와 동일하다. MD5의 연쇄 패턴의 특성만을 관찰하기 위하여 순환 시프트를 생략하면 MD5의 근사적 모델은 다음과 같이 쓸 수 있다.

$$A := B \oplus (A \oplus F(B, C, D) \oplus X^{(j)} \oplus K^{(i)}). \quad (2)$$

3.2 비선형 함수

차분 공격을 사용한 해쉬 함수의 공격은 입력 변수의 작은 차분은 반복적인 연산 중에 보상되어 없어진다는 데에 기초하고 있다[4-7]. 해쉬 함수의 차분 공격에서 비선형 함수는 식 (1)을 만족하고, 어떤 비선형 함수는 배타적 논리합과 비슷한 특성을 나타내기도 한다. 만약 어떤 비선형 함수가 임의의 확률로 배타적 논리합처럼 수행된다면 그 비선형 함수는 배타적 논리합으로 근사시킬 수 있다. 본 논문에서는 MD5의 비선형 함수가 배타적 논리합

〈표 1〉 MD5의 비선형 함수

단계 i	$f^{(i)}(X, Y, Z)$	함수의 이름
1-16	$(X \wedge Y) \vee (\overline{X} \wedge Z)$	MUX1
17-32	$(X \wedge Z) \vee (Y \wedge \overline{Z})$	MUX2
33-48	$X \oplus Y \oplus Z$	XOR
49-64	$Y \oplus (X \vee \overline{Z})$	OR-XOR

처럼 수행될 확률을 구하여, 비선형 함수들을 배타적 논리합으로 근사시킬 수 있음을 보인다. 배타적 논리합으로 근사시킬 비선형함수는 표 1과 같다.

[공리] 입력 X, Y, Z가 1비트라 하고 함수 $F(X, Y, Z)$가 배타적 논리합 함수라 두면, 배타적 논리합 함수는 다음을 만족한다.

(a) $F(\overline{X}, Y, Z) = \overline{F(X, Y, Z)}$
$F(\overline{X}, \overline{Y}, Z) = \overline{F(X, Y, Z)}$

(b) $F(\overline{X}, \overline{Y}, Z) = F(\overline{X}, Y, \overline{Z})$
$= F(X, \overline{Y}, \overline{Z}) = F(X, Y, Z)$

만일 비선형 함수가 공리의 (b)를 만족한다면, 비선형 함수에 대한 충돌을 발견할 수 있고, 압축 함수를 분석할 수 있다. 따라서, 비선형 함수에 대한 분석은 전체 해쉬 함수의 안전성 과 연관지어 생각할 수 있다. 해쉬 함수에서 사용되는 비선형 함수는 세 개의 32비트 입력을 병렬로 계산하므로 각 1비트들의 연산이 어떻게 이루어지는지를 살펴볼 필요가 있다. 비선형 함수 $f^{(i)}$의 연산 과정을 명확히 하기 위해 입력이 되는 각 변수들의 최상위비트(MSB)에 대해서만 생각하기로 한다. $f(B^{(i)}, C^{(i)}, D^{(i)})$에서 $f(B'^{(i)}, C'^{(i)}, D^{(i)})$로의 천이를 고찰하면 다음과 같은 서로 다른 경우들이 있음을 알 수 있다.

Case 1

$B^{(i)} = B'^{(i)}$, $C^{(i)} = C'^{(i)}$, $D^{(i)} = D'^{(i)}$인 경우. 출력은 $f(B^{(i)}, C^{(i)}, D^{(i)}) = f(B'^{(i)}, C'^{(i)}, D'^{(i)})$로 되어 값의 변화가 없고, 이때 $f^{(i)}$는 XOR처럼 작동한다.

Case 2

$B^{(i)}$의 31번째 비트에서 차분이 발생한 경우, 즉 $B'^{(i)} = B^{(i)} \oplus 2^{31}$인 경우. $f(B'^{(i)}, C'^{(i)}, D'^{(i)}) = f(B^{(i)}, C^{(i)}, D^{(i)}) \oplus 2^{31}$을 만족하면 $f^{(i)}$는 XOR처럼 계산된다.

Case 3

$C^{(i)}$와 $D^{(i)}$의 31번째 비트에서 각각 차분이 발생하였을 경우, 즉 , $D'^{(i)} = D \oplus 2^{31}$일 때, $f^{(i)}$가

$C^{(i)} = C \oplus 2^{31} f(B^{(i)}, C^{(i)}, D^{(i)}) = f(B^{(i)}, C^{(i)}, D^{(i)})$를 만족하면 $f^{(i)}$는 XOR처럼 작동한다.

Case 4

$D^{(i)}$의 31번째 비트에서 차분이 발생한 경우, 즉 $D^{(i)} = D \oplus 2^{31}$인 경우. $f(B^{(i)}, C^{(i)}, D^{(i)}) = f(B^{(i)}, C^{(i)}, D^{(i)}) \oplus 2^{31}$을 만족하면 $f^{(i)}$는 XOR처럼 계산된다.

Case 5

$B^{(i)}$와 $C^{(i)}$의 31번째 비트에서 각각 차분이 발생하였을 경우, 즉 $B^{(i)} = B \oplus 2^{31}$, $C^{(i)} = C \oplus 2^{31}$일 때, $f^{(i)}$가 $f(B^{(i)}, C^{(i)}, D^{(i)}) = f(B^{(i)}, C^{(i)}, D^{(i)})$를 만족하면 $f^{(i)}$는 XOR처럼 작동한다.

위의 공리에 의해 MD5에서 사용되는 비선형 함수가 배타적 논리합처럼 작동할 확률은 약 1/2이다. 각 비선형 함수가 배타적 논리합처럼 작동할 확률은 표2와 같다.

3.1절에서 고찰한 바와 같이 덧셈이 배타적 논리합처럼 작동한다고 가정하면, 비선형 함수가 배타적 논리합처럼 계산될 확률은 MD5의 각 단계 함수(Step function)가 배타적 논리합처럼 작용하는 확률이 된다. 즉, MD5의 단계 함수가 본 논문에서 제시하는 근사 모델을 따르는 확률은 1/2이다.

3.3 근사 모델을 이용한 차분 분석

MD5의 근사적 모델을 이용하여 입력 워드의 미세한 차분이 반복적으로 수행되는 단계 함수를 통하여 확산, 보상되는 과정을 분석한다. 먼저, 서로 다른 2개의 워드를 $X^{(i)}$와 $\tilde{X}^{(i)}$라 두고, 두 워드간의 차분$\Delta X^{(i)}$를 다음과 같이 정의 한다.

$$\Delta X^{(i)} = \tilde{X}^{(i)} - X^{(i)}. \quad (3)$$

〈표 2〉 비선형 함수가 배타적 논리합처럼 수행될 확률

Case	MUX1	MUX2	OR-XOR
2	1/2	3/4	1/2
3	0	1/4	1/2
4	1/2	3/4	1/2
5	1/2	1/4	1/2

$\Delta X^{(i)} = 2^{31}$라 가정하면 오직 1비트의 차분이 어떻게 변화해 나가는지 분석할 수 있다. 또한, 모든 연산은 32비트 단위로 이루어져 인접하는 비트들과는 독립적으로 계산되므로 연산을 분석하는 과정에서는 1비트에 대해서만 생각하기로 한다. 예를 들어, $\Delta X^{(i)}$의 MSB에 차분이 발생했다고 가정하면, 연쇄 변수 (A,B,C,D)에도 차분의 영향이 나타나, (A,B,C,D)와는 다른 값 (A',B',C',D')가 계산될 것이다. 본 논문에서는 구체적인 (A,B,C,D)와 (A',B',C',D')의 차이를 찾기보다는 $\Delta X^{(i)}$의 영향이 (A',B',C',D')에 어떻게 나타나는지를 분석하기 위해 차분 마스크 비트를 (ma, mb, mc, md)로 정의 한다. 차분 마스크 비트는 차분이 발생하면 1, 차분이 없거나 상쇄되었을 경우에 0이 된다. 즉, 입력 차분을 분석하기 위하여 단계 함수의 32비트의 연쇄 변수 (A,B,C,D)를 1비트 차분 마스크 비트 (ma, mb, mc, md)로 대응시키는 것이다. 차분 마스크 비트는 초기값으로 (0,0,0,0)을 가진다고 가정한다. 32비트 연쇄 변수를 1비트 차분 마스크 비트로 대응시켰듯이 입력 $X^{(i)}$도 1비트 변수 mx로 대응시키고 초기값은 0이라 한다. $\Delta X^{(i)}$가 발생하면 mx는 1로 되고, ma, mb, mc, md는 식(2)에 의해 계산된다. 이때, 상수 $K^{(i)}$를 항상 0로 둔다. 어떤 차분 마스크 비트의 값이 1이면, 반복되는 단계 함수에서 사용되는 변수들은 $\Delta X^{(i)}$의 영향을 받았음을 의미한다. $\Delta X^{(i)}$의 파급 효과를 평가하기 위한 완전 확산(perfect diffusion)은 다음과 같이 정의된다[8].

<정의> 완전 확산(Perfect Diffusion)

알파벳 Z상에서 r개의 알파벳으로 구성된 문자열 Z^r로부터 n개의 알파벳으로 구성된 문자열 Z^n로의 함수 G에 대한 (x, G(x))형태의 (r+n) 투플이 어떠한 r의 위치에서도 충돌하지 않는다고 할 때, r개의 입력 중 t개의 입력이 바뀐 후, 출력 중 적어도 n-t+1개의 값이 변한다면 함수 G는 완전 확산을 제공한다.

근사 모델에서 ma, mb, mc, md는 0 또는 1이

되고, 그 값이 1일 경우, 대응되는 연쇄 변수가 $\Delta X^{(i)}$의 영향으로 차분 값을 가지고 있다는 것을 의미한다. 이미 보고 되어 있는 MD5에 대한 공격은 $X^{(14)}$에 차분을 부여하여 반복 계산에 의해 차분이 상쇄됨을 이용한 것이었다[7]. 본 논문에서도 $X^{(14)}$에 대한 차분을 이용하기로 한다. $X^{(14)}$는 단계 15, 26, 36, 51에서 각각 입력으로 사용된다. 근사 모델의 15번째 단계에서는 연쇄 변수 C가 다음과 같이 계산된다.

Step 15 :

$$C^{(15)} := D^{(14)} + (C^{(11)} + f^{(15)}(D^{(14)}, A^{(13)}, B^{(12)}) + X^{(l)} + K^{(15)})$$

위 식을 차분 마스크 비트에 대응시켜 표현하면 다음과 같다.

$$\begin{aligned} mc &:= md \oplus (mc \oplus f(md, ma, mb) \oplus mx \oplus K^{(15)}) \\ &:= 0 \oplus (0 \oplus 0 \oplus 1 \oplus 0) \\ &:= 1. \end{aligned}$$

mc의 값이 1이라는 것은 $\Delta X^{(14)}$에 의해 연쇄 변수 C에 차분이 발생하였음을 의미한다. 그리고, MD5가 안전하기 위해서는 압축 함수가 완전 확산을 만족해야하므로 입력의 차분이 4개의 연쇄 변수(A,B,C,D)로 확산되어야만 한다. 근사 모델에서 입력 차분 $\Delta X^{(14)}$의 확산을 추적한 결과 64단계 후에는 오직 변수 B, C로만 차분이 확산됨을 확인할 수 있었다. 이것은 MD5의 의사 충돌 발견과 같은 상황을 나타내고 있고[7], 연쇄 변수의 연쇄 특성이 차분 공격에 대해 약점이 있다는 것을 의미한다. 근사 모델은 순환 시프트를 생략한 모델이므로 순환 시프트의 양에 의해 차분 확산이 더 좋아질 수도 있을 것이나, 현재 MD5에 대한 충돌이 발견된 것으로 보면 MD5의 순환 시프트양이 최적이라고는 할 수 없다. 또, Wang 등은 MD5의 23단계에서 34단계까지 입력 차분에 대한 출력 차분값이 0이 되고, 35단계 이후부터는 출력 차분값이 231 으로 고정되어 입력의 차분이 확장되지 못함을 보였다[9].

4. 다중 패턴 방식

4.1 MD5의 연쇄 패턴

3장에서 MD5의 근사 모델은 입력의 차분을 해쉬값 전체로 확산시키지 못하여 차분 공격에 취약하다는 것을 알 수 있었다. 차분 공격에의 취약점의 원인으로서는 적은 반복의 수, 간단한 비선형 함수의 사용, 단순한 워드 입력 등을 생각할 수 있다. 이를 해결하는 방법으로는 반복의 횟수 증가, 복잡한 비선형 함수 사용, 입력 워드의 확장 등이 있다. 그러나, 본 논문에서는 MD5의 변수의 흐름, 즉 변수의 연쇄에서 그 원인을 찾고자 한다.

MD5는

Step i:

$$A^{(i)} := B^{(i-1)} + (A^{(i-1)} + f^{(i)}(B^{(i-1)}, C^{(i-1)}, D^{(i-1)}) + X^{(j)} + K^{(i)})^{<<<s}$$

Step i+1:

$$D^{(i+1)} := A^{(i)} + (D^{(i)} + f^{(i+1)}(A^{(i)}, B^{(i)}, C^{(i)}) + X^{(j)} + K^{(i+1)})^{<<<s}$$

Step i+2:

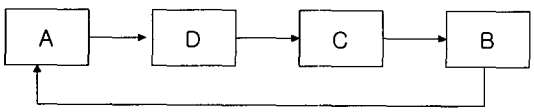
$$C^{(i+2)} := D^{(i+1)} + (C^{(i+1)} + f^{(i+2)}(D^{(i+1)}, A^{(i+1)}, B^{(i+1)}) + X^{(j)} + K^{(i+2)})^{<<<s}$$

Step i+3:

$$B^{(i+3)} := C^{(i+2)} + (B^{(i+2)} + f^{(i+3)}(C^{(i+2)}, D^{(i+2)}, A^{(i+2)}) + X^{(j)} + K^{(i+3)})^{<<<s}$$

의 과정을 64번 반복하여 마지막 Step 61~ Step 64까지의 4개의 변수값을 연결하여 128비트의 해쉬값을 계산한다. 위 식에서 각 항목의 변수의 흐름에 주목해 보면, 좌변의 값들이 $A^{(i)} \rightarrow D^{(i+1)} \rightarrow C^{(i+2)} \rightarrow B^{(i+3)}$의 순서로 계산됨을 알 수 있다. 우변의 첫째 항목은 $B^{(i-1)} \rightarrow A^{(i)} \rightarrow D^{(i+1)} \rightarrow C^{(i+2)}$순으로 사용되며, 괄호안의 첫 번째 항목은 $A^{(i-1)} \rightarrow D^{(i)} \rightarrow C^{(i+1)} \rightarrow B^{(i+2)}$순으로 사용된다. 이 순서에 순환을

허용하면, $B \rightarrow A \rightarrow D \rightarrow C$는 $A \rightarrow D \rightarrow C \rightarrow B$와 같다. 따라서, 식(4)의 각 단계에서 같은 위치에 있는 변수들을 순서대로 나열하면 결국은 $A \rightarrow D \rightarrow C \rightarrow B$가 된다. 본 논문에서는 이 변수들의 흐름을 연쇄 패턴(Chaining pattern)이라 명명한다. 그림 1에서 MD5의 연쇄 패턴을 표현하였다.



〈그림 1〉 MD5의 연쇄 패턴

2개의 입력 M과 M'에 대하여 i번째의 메시지 블록에 대하여 반복 연산했을 때의 차분을 ΔH_i, i+1번째의 반복 차분을 ΔH_{i+1}이라 하고, 그림1의 연쇄 패턴을 사용하면, 차분 특성이 갖는 확률은 다음과 같다[9].

$$P \geq \Pi_{i=1}^{4} P_j \quad \text{and} \quad P_j \geq \Pi_{t=1}^{16} P_{jt} \qquad (4)$$

단, P는 $\Delta H_i \rightarrow \Delta H_{i+1}$에 대해 차분 특성의 확률, P_j는 각 라운드에서의 차분 확률, P_{jt}는 t라운드 j 단계에서의 차분 특성 확률을 의미한다.

4.2 다중 연쇄 패턴

MD5는 하나의 연쇄 패턴을 이용하고 있으며 이 단순한 연쇄 패턴을 MD5가 완전 확장을 제공하지 못하는 요인으로 생각할 경우, 복잡한 연쇄 패턴을 사용한다면 완전 확장을 제공할 수 있을 것이다. 본 논문에서는 MD5와는 달리 각 항목별로 서로 다른 연쇄 패턴을 가지는 단계 함수를 제안한다. 제안 하는 연쇄 패턴을 다중 연쇄 패턴(multiple chaining pattern)이라 부르기로 하고 다음과 같이 정의한다.

for $i = 0$ *to* 60

$$B^{(i+1)} := C^{(i)} + (B^{(i)} + f^{(i+1)}(C^{(i)}, D^{(i)}, A^{(i)}) + X^{(j)} + K^{(i+1)})^{<<<s}$$

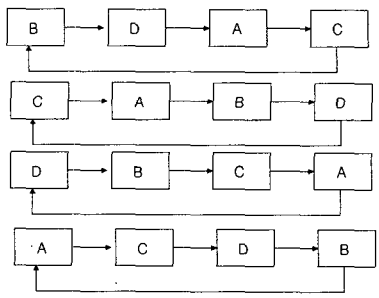
$$D^{(i+2)} := A^{(i-1)} + (D^{(i+1)} + f^{(i+2)}(A^{(i+1)}, B^{(i+1)}, C^{(i+1)}) + X^{(k)} + K^{(i+2)})^{<<<s}$$

$$A^{(i+3)} := B^{(i+2)} + (A^{(i+2)} + f^{(i+3)}(B^{(i+2)}, C^{(i+2)}, D^{(i+2)}) + X^{(l)} + K^{(i+3)})^{<<<s}$$

$$C^{(i+4)} := D^{(i+3)} + (C^{(i+3)} + f^{(i+4)}(D^{(i+3)}, A^{(i+3)}, B^{(i+3)}) + X^{(m)} + K^{(i+4)})^{<<<s}$$

$$i := i+4$$

단, $0 \leq j, k, l, m \leq 15$. 압축 함수가 64단계라 하면 MD5의 알고리즘에 의하여 최종적인 해쉬값은 $B^{(61)}, D^{(62)}, A^{(63)}, C^{(64)}$를 연결한 값이 된다. (i+1) 단계에서 (i+4) 단계의 순으로 연쇄 변수의 값 $B^{(i+1)}, D^{(i+2)}, A^{(i+3)}, C^{(i+4)}$이 계산된다. (i+1) 단계에서 $B^{(i+1)}$을 구하기 위한 식에서 첫 번째 항은 $C^{(i)}$이고, (i+2) 단계에서 같은 위치의 항은 $A^{(i+1)}$이 되며, (i+3) 단계에서 같은 항의 값은 $B^{(i+2)}$, (i+4) 단계에서는 $D^{(i+3)}$이 된다. 즉, 4단계에 걸쳐 압축 함수의 첫 번째 항에 나타나는 변수를 순서대로 나타내면 $C \rightarrow A \rightarrow B \rightarrow D$가 된다. (i+1) 단계부터 (i+4) 단계까지 괄호 안의 첫 번째 항은 $B \rightarrow D \rightarrow A \rightarrow C$의 순으로 나타난다. 비선형 함수의 두 번째 인수는 $D \rightarrow B \rightarrow C \rightarrow A$, 세 번째 인수는 $A \rightarrow C \rightarrow D \rightarrow B$의 순으로 나타난다. 즉, $B \rightarrow D \rightarrow A \rightarrow C$, $C \rightarrow A \rightarrow B \rightarrow D$, $D \rightarrow B \rightarrow C \rightarrow A$, $A \rightarrow C \rightarrow D \rightarrow B$



〈그림 2〉 다중 연쇄 패턴

의 4개의 연쇄 패턴을 사용하는 MD5의 압축 함수를 구성하였다. 이 4개의 연쇄 패턴을 그림 2에서 보인다.

다중 연쇄 패턴으로 사용 가능한 패턴의 수는 총 4!=24개이고, 이중 압축 함수를 구성하는 항의 수만큼 연쇄 패턴을 선택 할 수 있다. 즉, MD5에서는 최대 6가지의 서로 다른 패턴을 사용하여 압축 함수를 구성할 수 있다. 그러나, 모든 패턴이 완전 확산을 이루지는 않으므로, 다중 연쇄 패턴으로 사용할 패턴을 선택 할 때에는 충분히 주의를 기울여야만 한다.

4.3 다중 연쇄 패턴을 사용한 MD5의 차분 분석

4.3.1 실험에 의한 분석

3.3절의 근사 모델을 이용하여 입력의 차분의 변화를 추적하는 방법으로 제안 압축 함수의 성능을 평가하면 입력의 차분 $\Delta X^{(14)}$ 가 64 단계 이후에 모든 연쇄 변수, A, B, C, D에 영향을 미침을 실험을 통하여 알 수 있었다. 즉, n=4, t=1 일 때 완전 확산을 만족하였으므로, 제안 방식은 해쉬 함수에 대한 차분 공격에 대해 원래의 압축 함수를 사용하는 것 보다 강하다는 것을 실험을 통하여 확인하였다. 표 3은 사용 가능한 모든 연쇄 패턴에 대하여 $\Delta X^{(14)}$가 각 연쇄 변수에 미치는 영향을 테스트한 결과이다. 처음으로 4개의 변수 모두 입력 차분의 영향을 받는 51번째 단계 이후 4개의 연쇄 변수 중 입력 차분의 영향을 받는 변수와 64단계 후에 입력 차분이 확산된 연쇄 변수를 표시하였다. 진하게 표시된 부분은 제안 연쇄 패턴에 사용된 것으로, 그림3의 4개의 패턴을 의미한다.

4.3.2 차분 특성

메시지 크기가 $512\times(k-1)$비트인 서로 다른 2개의 메시지 M과 M'을 $M=(M_0,M_1,...,M_{k-1})$, $M'=(M_0',M_1',...,M_{k-1}')$라 두면, 해쉬 함수에 대한 차분은 다음과 같다.

$$\Delta H_0 \xrightarrow{(M_0,M_0')} \Delta H_1 \xrightarrow{(M_1,M_1')} \cdots \xrightarrow{(M_{k-1},M_{k-1}')} \Delta H$$

ΔH_0는 초기 차분으로 0이고, ΔH는 두 메시지에 대한 출력의 차분이다. 또, $\Delta H_i=\Delta IV_i$로서 i번째 반복에 대한 출력 차분이다. i번째 반복 차분 $\Delta H_i \rightarrow \Delta H_{i+1}$는 다음과 같이 쓸 수 있다.

$$\Delta H_i \xrightarrow{P_1} \Delta R_{i+1,1} \xrightarrow{P_2} \Delta R_{i+1,2} \xrightarrow{P_3} \Delta R_{i+1,3}$$

$$\xrightarrow{P_4} \Delta R_{i+1,4} = \Delta H_{i+1}$$

확률 P_j인 라운드 차분 $\Delta R_{j-1} \rightarrow \Delta R_j$ (j=1,2,3,4)는 $\Delta R_{j-1} \xrightarrow{P_{j1}} \Delta X_1 \xrightarrow{P_{j2}} \cdots \xrightarrow{P_{j16}} \Delta X_{16} = \Delta R_j$로 쓸 수 있다. 단, $\Delta X_{t-1} \xrightarrow{P_{jt}} \Delta X_t$, (t=1,2,...,16)는 j번째 라운드의 t번째 단계에서의 차분 특성이다. 라운드 함수의 각 단계에서 4개의 연쇄 변수는 각기 다른 연쇄

〈표 3〉 연쇄 패턴별 입력 차분의 확산 결과

연쇄 패턴	51단계 후 변화된 변수	64단계 후 변화된 변수
A-B-C-D	B, C, D	B, D
B-A-C-D	B, C ,D	A, B, D
A-C-B-D	A, D	A, C, D
C-A-B-D	**A, B, C, D**	**A, B, C, D**
C-B-A-D	A, B, C	A, D
B-C-A-D	C	A, B
A-B-D-C	B	A, D
B-A-D-C	A, B, D	C, D
A-D-B-C	A, B, C	A, C, D
D-A-B-C	A, B, C	A, C,
D-B-A-C	C, D	B, C, D
B-D-A-C	**A, B, C, D**	**A, B, C, D**
A-D-C-B (MD5)	A, C, D	B, C
D-A-C-B	A	C, D
A-C-D-B	**A, B, C, D**	**A, B, C, D**
C-A-D-B	B, C	A, B, C
C-D-A-B	A, B, D	B, D
D-C-A-B	A, B, D	B, C, D
D-B-C-A	**A, B, C, D**	**A, B, C, D**
B-D-C-A	A, B	A, B, D
D-C-B-A	B, C, D	A, B
C-D-B-A	D	B, C
C-B-D-A	A, C, D	A, B, C
B-C-D-A	A, C, D	A, C

패턴 4개중 하나를 이용하므로 차분 $\Delta H_i \rightarrow \Delta H_{i+1}$의 확률은 다음을 만족한다.

$$P \geq \Pi_{i=1}^{t} P_j \text{ and } P_j \geq \Pi_{t=1}^{6} \frac{P_\mu}{4} \quad (5)$$

위의 차분 확률은 4.1절의 식(4)의 MD5의 차분 특성이 갖는 확률보다 작다.

5. 결 론

본 논문에서는 설계 방법에 대한 이론적, 실험적 배경이 알려져 있지 않은 해쉬 함수에 대한 근사적 모델을 제안하여 MD5에 적용하였다. MD5의 근사 모델에 대한 입력 차분의 변화를 추적하여 연쇄 패턴의 특성만을 고려했을 때 MD5가 완전 확산을 제공하지 못하고, MD5의 의사 충돌 발견과 같은 결과를 나타냄을 알 수 있었다. 이것은 순환 시프트를 생략한 근사 모델의 결과가 원래의 MD5와 같은 확산을 나타낸 것을 의미하며, MD5의 순환 시프트의 양이 최적이 아님을 알 수 있다. 이들 결과에 의하여 본 논문에서 제안한 근사 모델은 향후 해쉬 함수의 설계 과정에서 순환 시프트 등의 각 연산자의 효용성을 평가하고 결정하는데 사용될 수 있으리라 생각된다.

또, 본 논문에서는 공격에 강한 해쉬 함수 설계를 위하여 연쇄 변수들의 순서를 변화시킨 다중 연쇄 패턴을 제안하여 여분의 계산이나 메모리 없이 완전 확산을 제공하는 단계 함수를 구성할 수 있음을 근사 모델을 통하여 보였고, 제안한 다중 연쇄 패턴을 적용한 MD5의 차분 특성이 갖는 확률이 MD5의 차분 확률보다 작음을 보였다. 이것은 충돌 발견 확률을 줄일 수 있음을 의미하며, 제안하는 다중 연쇄 패턴을 다른 해쉬 함수에 적용하면 충돌 확률을 줄일 수 있으리라 기대된다.

참 고 문 헌

[1] R.Rivest, "The MD5 message- digest algorithm", Request For Comments(RFC)1321, Internet Activities Board, Internet Privacy Task Force, April, 1992.

[2] NIST, "Secure hash standard", FIPS 180-1, US Department of Commerce, Washington D.C., 1993.

[3] H.Dobbertin, A.Bosselaers, B.Preneel, "RIPEMD-160:A strengthened version of RIPEMD", Fast Software Encryption-cambridge Workshop, Lecture Notes in Computer Science 1039, Springer-Verlag, pp.71-82, 1996.

[4] R.Rivest, "The MD4 message- digest algorithm", Request For Comments(RFC)1320, Internet Activities Board, Internet Privacy Task Force, April, 1992.

[5] B.den Boer and A.Bosselaers, "An attack on the last two round of MD4", Advances in Cryptology-Crypto'91, Lecture Notes in Computer Science, Springer-Verlag, pp.194-203, 1991.

[6] H.Dobbertin, " Cryptanalysis of MD4", Fast Software Encryption, Lecture Notes in Computer Science 1039, Springer-Verlag, pp.53-69, 1996.

[7] H. Dobbertin, "The status of MD5 after recent attack", CryptoBytes, 2(2), September, pp.1-6, 1996.

[8] S.Vaudenay, "On the need for multipermutations:cryptalaysis of MD4 and SAFER", Fast Software Encryption, Lecture Notes in Computer Science, Springer-Verlag, pp. 286-297, 1995.

[9] X.Y.Wang, H.B.Yu, "How to break MD5 and other hash functions", advances in Cryptology-Eurocrypt'05, Springer-Verlag, pp.19-35, 2005.

◐ 저 자 소 개 ◑

이 선 영(Sun Young Lee)
1993년 부경대학교 전자계산학과 졸업(이학사)
1995년 부경대학교 대학원 전자계산학과 졸업(이학석사)
2001년 일본 동경대학교 대학원 전자정보공학과 졸업(공학박사)
2001년 순천향대학교 강사
2004~현재 순천향대학교 정보보호학과 전임강사
관심분야 : 암호이론, 정보이론, 인터넷 보안
E-mail : sunlee@sch.ac.kr