情報保護學會論文誌
第19卷 第5號, 2009. 10

해쉬함수 ARIRANG의 축소된 단계에 대한 역상공격

홍 득 조,†‡ 김 우 환, 구 본 욱
ETRI 부설연구소

Preimage Attacks on Step-Reduced ARIRANG

Deukjo Hong,†‡ Woo-Hwan Kim, Bonwook Koo
The Attached Institute of ETRI

요 약

본 논문에서는 SHA-3의 1 라운드 후보로 제안된 알고리즘인 ARIRANG의 단계수가 축소된 버전들에 대한 역상 공격을 소개한다. 이 공격은 단계 1부터 단계 33까지의 ARIRANG-256 및 ARIRANG-512의 33단계 OFF(Original FeedForward1) 버전의 역상을 찾아내며, 단계 1부터 단계 34까지의 ARIRANG-256 및 ARIRANG-512의 31단계 MFF(Middle Feed-Forward1) 버전의 역상을 찾아낸다. 공격 복잡도는 ARIRANG-256에 대해서 약 2^{241}이고 ARIRANG-512에 대해서 약 2^{481}이다.

ABSTRACT

The hash function ARIRANG is one of the 1st round SHA-3 candidates. In this paper, we present preimage attacks on ARIRANG with step-reduced compression functions. Our attack finds a preimage of the 33-step OFF(Original FeedForward1) variants of ARIRANG, and a preimage of the 31-step MFF (Middle FeedForward1) variants of ARIRANG. Its time complexity is about 2^{241} for ARIRANG-256 and 2^{481} for ARIRANG-512, respectively.

Keywords: SHA-3 candidate, ARIRANG, Preimage Attack, Hash Function

I. 서 론

최근 NIST는 새로운 미국의 차세대 표준해쉬함수 개발을 위한 SHA-3 프로젝트를 추진하여 전 세계의 여러 암호연구그룹으로부터 해쉬함수 알고리즘들을 공모하였다. 2008년 10월 31일까지 64개의 알고리즘이 접수되었으며, 2008년 12월 9일에 그 중 51개의 알고리즘이 1 라운드 후보 알고리즘으로 선정되었다. 최근 2009년 7월 24일에는 14개의 2 라운드 후보 알고리즘이 공시되었다.

ARIRANG[1]은 1 라운드 SHA-3 후보로 제안된 알고리즘 중 하나이며 2 라운드에는 진출하지 못했다. ARIRANG의 전체 구조는 카운터를 사용하는 MD 구성법 (Merkle-Damgård Construction with a Counter)을 따른다. 출력길이에 따라, ARIRANG-224, ARIRANG-256, ARIRANG-384, ARIRANG-512의 네 가지 버전으로 분류하는데, ARIRANG-224는 ARIRANG-256의 출력의 단순한 32 비트 절사이며 ARIRANG-384 또한 ARIRANG-512의 출력의 단순한 128비트 절사이다. 따라서 ARIRANG-256과 ARIRANG-512을 기본 알고리즘으로 볼 수 있다. 전체 해쉬 구조의 기반이 되는 압축함수들은 40 단계로 구성되어 있다.

본 논문에서는 압축함수의 단계수가 축소된 버전의 ARIRANG에 대한 역상 공격을 소개한다. 공격대상으로서 두 가지 버전의 축소된 압축함수를 고려한다. OFF(Original Feedforward1)이라 부르는 첫 번째 버전은 오리지널 알고리즘과 동일한 위치에 Feedforward1이 작용한다. 두 번째 버전은 Feedfor-

접수일(2009년 4월 29일), 게재확정일(2009년 8월 13일)
† 주저자, hongdj@ensec.re.kr
‡ 교신저자, hongdj@ensec.re.kr

ward1이 가운데 단계의 출력에 위치한다. 이것을 MFF (Middle Feedforward1)이라 부른다. ARIRANG의 압축함수의 구조를 분석한 결과, 내부에 사용되는 메시지 워드들의 위치를 최대 4단계까지 위로 옮길 수 있음을 발견하였다. 이러한 워드이동성질은 본 논문에서 소개되는 공격에 주요하게 사용되었다.

본 논문의 공격은 단계 1부터 단계 33까지의 ARIRANG-256 및 ARIRANG-512의 33-단계 OFF(Original FeedForward1) 버전의 역상을 찾아내며, 단계 1부터 단계 31까지의 ARIRANG-256 및 ARIRANG-512의 34단계 MFF(Middle Feed-Forward 1) 버전의 역상을 찾아낸다. ARIRANG-256의 경우, 모든 공격들은 2241번의 축소된 압축함수의 연산에 해당하는 복잡도를 갖는다. ARIRANG-512의 경우, 모든 공격들은 2481번의 축소된 압축함수의 연산에 해당하는 복잡도를 갖는다. 본 연구는 Sasaki와 Aoki의 역상공격체계[2-4]를 따른다.

II. 해쉬함수 ARIRANG

ARIRANG의 전체 구조는 연쇄변수에 카운터를 XOR하는 MD 해쉬구조이다. ARIRANG-256과 ARIRANG-512의 압축함수 F는 각각 512비트와 1024비트의 메시지 블록을, 256비트와 512비트의 연쇄변수를 입력받아 256비트와 512비트의 출력을 생성한다. ARIRANG-256과 ARIRANG-512의 구조는 ARI RANG-256의 워드 사이즈가 32비트인 반면 ARI RANG-512의 워드 사이즈가 64비트임을 제외하면 동일하다. 연쇄변수의 초기치 H^0는 설계자가 제안한 고정된 상수이다. 카운터의 초기치 Ctr^1은 0이다. 길이가 512비트 (ARIRANG-256) 또는 1024비트 (ARIRANG-512)의 배수가 되도록 패딩된 메시지 $M = M^1||...||M^N$이 주어지면 다음과 같은 과정을 통하여 해쉬값 H^N이 생성된다.

ARIRANG($M^1||...||M^N$)
1: for i = 1, ..., N do
2: $H^{i-1} \leftarrow H^{i-1} \oplus Ctr^i$
3: $H^i \leftarrow compress(H^{i-1}||M^i)$
4: endfor
5: return H^N

ARIRANG-256의 압축함수에서는 메시지 스케줄이 512비트 메시지 블록 M^i로부터 32개의 메시지 워드 $W_0, ..., W_{31}$을 생성하고, [표 1]과 같이 정의되는 인덱스 함수 σ에 따라 j번째 단계(j = 0, ..., 39)에 두 메시지 워드 $W_{\sigma(2j)}$, $W_{\sigma(2j+1)}$을 적용시킨다.

메시지 워드 $W_0, ..., W_{31}$은 다음과 같은 방법으로 생성된다.

- 512비트 메시지 블록 M^i이 16개의 워드 $W_0, ..., W_{15}$로 분할된다. ($M^i = W_0|| ... ||W_{15}$)
- 나머지 워드들은 아래와 같이 계산된다. 여기서 $K_0, ..., K_{15}$는 상수이다.

$W_{16} = (W_9 \oplus W_{11} \oplus W_{13} \oplus W_{15} \oplus K_0)^{\ll 5}$
$W_{17} = (W_8 \oplus W_{10} \oplus W_{12} \oplus W_{14} \oplus K_1)^{\ll 11}$
$W_{18} = (W_1 \oplus W_3 \oplus W_5 \oplus W_7 \oplus K_2)^{\ll 19}$
$W_{19} = (W_0 \oplus W_2 \oplus W_4 \oplus W_6 \oplus K_3)^{\ll 31}$
$W_{20} = (W_{14} \oplus W_4 \oplus W_{10} \oplus W_0 \oplus K_4)^{\ll 5}$
$W_{21} = (W_{11} \oplus W_1 \oplus W_7 \oplus W_{13} \oplus K_5)^{\ll 11}$
$W_{22} = (W_6 \oplus W_{12} \oplus W_2 \oplus W_8 \oplus K_6)^{\ll 19}$
$W_{23} = (W_3 \oplus W_9 \oplus W_{15} \oplus W_5 \oplus K_7)^{\ll 31}$
$W_{24} = (W_{13} \oplus W_{15} \oplus W_1 \oplus W_3 \oplus K_8)^{\ll 5}$

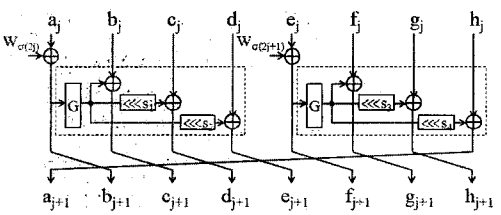
[표 1] 인덱스 함수 σ의 입출력표

j	σ(2j)	σ(2j+1)	σ(2j+20)	σ(2j+21)	σ(2j+40)	σ(2j+41)	σ(2j+60)	σ(2j+61)
0	16	17	20	21	24	25	28	29
1	0	1	3	6	12	5	7	2
2	2	3	9	12	14	7	13	8
3	4	5	15	2	0	9	3	14
4	6	7	5	8	2	11	9	4
5	18	19	22	23	26	27	30	31
6	8	9	11	14	4	13	15	10
7	10	11	1	4	6	15	5	0
8	12	13	7	10	8	1	11	6
9	14	15	13	0	10	3	1	12

$W_{25} = (W_4 \oplus W_6 \oplus W_8 \oplus W_{10} \oplus K_9)^{\lll 11}$
$W_{26} = (W_5 \oplus W_7 \oplus W_9 \oplus W_{11} \oplus K_{10})^{\lll 19}$
$W_{27} = (W_{12} \oplus W_{14} \oplus W_0 \oplus W_2 \oplus K_{11})^{\lll 31}$
$W_{28} = (W_{10} \oplus W_0 \oplus W_6 \oplus W_{12} \oplus K_{12})^{\lll 5}$
$W_{29} = (W_{15} \oplus W_5 \oplus W_{11} \oplus W_1 \oplus K_{13})^{\lll 11}$
$W_{30} = (W_2 \oplus W_8 \oplus W_{14} \oplus W_4 \oplus K_{14})^{\lll 19}$
$W_{31} = (W_7 \oplus W_{13} \oplus W_3 \oplus W_9 \oplus K_{15})^{\lll 31}$

단계함수 "step"은 [그림 1]과 같으며 압축함수 내부에서 총 40회 반복 적용된다. 단계함수를 step으로 표현하자. 단계함수 내부에 사용된 G 함수는 ARIRANG-256의 경우 AES와 동일한 네 개의 8비트 입출력 S-box와 1개의 4×4 MDS 행렬의 합성이며, ARIRANG-512의 경우 여덟 개의 8비트 입출력 S-box와 1개의 8×8 MDS 행렬의 합성이다. G 함수는 비선형 치환(nonlinear permutation)이라는 사실만 공격에 이용되므로, 그 이상의 상세한 설명은 생략하기로 한다. i번째 블록에서 연쇄변수 H^{i-1}과 메시지 블록 M^i가 압축함수 "compress"에 입력되면 우선 메시지워드 $W_0,\ldots,W_{31}$이 생성되고 단계함수 step을 반복 적용하는 다음과 같은 과정을 거쳐 그 다음 연쇄변수인 H^i가 출력된다. $state_j$는 $state_j = (a_j,b_j,c_j, d_j,e_j,f_j,g_j,h_j)$로 정의된다.

compress(H^{i-1},M^i)
1: $state_0 \leftarrow H^{i-1}$
2: for j = 0, ..., 19 do
3: $state_{j+1} \leftarrow step(state_j, W_{\sigma(2j)}, W_{\sigma(2j+1)})$
4: endfor
5: $state_{20} \leftarrow state_{20} \oplus state_0$ // $Feedforward_1$
6: for j = 20, ..., 39 do
7: $state_{j+1} \leftarrow step(state_j, W_{\sigma(2j)}, W_{\sigma(2j+1)})$
8: endfor
9: $H^i \leftarrow state_{40} \oplus state_0$ // $Feedforward_2$
10: return H^i



[그림 1] j번째 단계함수 $step_j$

III. 역상공격기법 및 공격에 이용된 성질들

3.1 Aoki와 Sasaki의 역상공격기법

어떤 해쉬값 H^N의 역상(preimage)은 그 해쉬값을 생성해주는 메시지 $M=M^1||\ldots||M^N$을 의미한다. MD 구성법을 따르는 해쉬함수들에 대하여 Sasaki와 Aoki의 역상공격[2-4]은 해쉬값 H^N이 주어지면 압축함수 F에 대하여 $F(H^{N-1},M^N) = H^N$을 만족시키는 H^{N-1}, M^N을 찾는 알고리즘을 구성한 후, 그것을 이용하여 역상을 찾는 알고리즘을 구성한다. 보통, 이 때 M^N에서 패딩되었을 때 메시지 길이의 이진코드에 해당하는 부분을 상수로 고정시킬 수 있다면 1-블록 또는 2-블록 역상을 찾는 공격이 가능하게 된다. 본 논문에서 공격의 대상인 ARIRANG의 축소된 압축함수들에 대해서는 그것이 가능하다. Sasaki와 Aoki의 역상공격의 프레임워크를 적용하는 ARIRANG에 대한 2-블록 역상을 찾는 과정은 다음과 같이 구성된다.

1. 해쉬값 H^2이 주어지면 압축함수 F에 대하여 $F(H^1,M^2) = H^2$를 만족시키는 H^1와 M^2을 찾는 알고리즘을 구성한다. 이 때 1개의 (H^1,M^2)을 찾는데 소요되는 계산복잡도를 2^x라 하고 H^1의 길이를 n이라 하자.
2. 1의 알고리즘을 반복실행하여 $2^{(n-x)/2}$개의 (H^1, M^2)을 생성하여 테이블에 저장한다.
3. $2^{(n+x)/2}$개의 메시지 M^1에 대해 $X = F(H^0 \oplus Ctr^1, M^1)$값을 구하여 테이블에 $X \oplus Ctr^2 = H^1$을 만족하는 (H^1,M^2)쌍이 있는지 체크한다. 그러한 쌍에 대하여, 연결된 메시지 $M = M^1||M^2$은 H^2의 역상이 된다.

위와 같은 역상공격은 $2^{(n+x)/2+1}$의 복잡도를 요구한다.

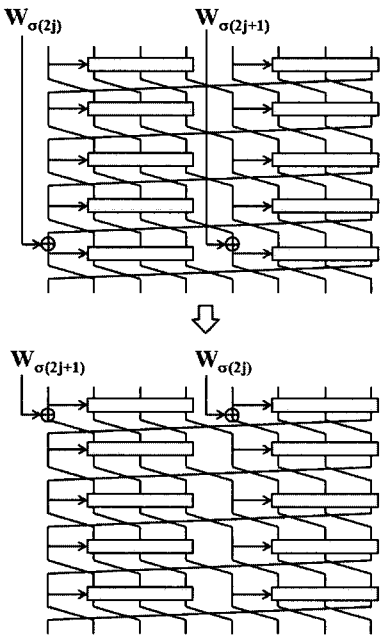
위의 역상공격과정에서 첫 번째 단계의 (H^1,M^2)을 찾는 알고리즘은 중간일치공격(Meet-in-the-Middle Attack) 기법을 이용한다. 압축함수의 내부상태를 표시하는 변수 $state_i$ (i=0, ..., 40)는 모든 i에서 $state_{i-1}$로부터 $state_i$를 계산할 수 있으며, 심지어는 $state_0$ 또한 $state_{40}$으로부터 H^2의 XOR을 통해 계산될 수 있으므로 $state_0 \rightarrow state_1 \rightarrow \ldots \rightarrow state_{40} \rightarrow state_0$과 같은 순환수열로 볼 수 있다. 압축함수에 중간일치기법을 적용하는 가장 기본적인 방법은 이 순

환수열을 적절하게 두 개의 독립적인 부분으로 분리하는 것이다. Sasaki와 Aoki[2-4]는 이러한 독립된 부분을 chunk라고 명명하였다. 독립성을 갖기 위해서는 각 Chunk는 다른 Chunk에서는 전혀 사용되지 않는 중립 워드를 적어도 하나 이상 포함해야한다. 두 chunk 중 첫 번째 및 마지막 단계를 포함하는 것을 Outer Chunk, 중간 단계들을 포함하는 것을 Inner Chunk라 부른다. 전체 단계를 두 개의 Chunk로 분리할 경우 두 개의 경계선이 형성된다. 이 때 한 경계선은 공격이 시작되는 부분이며, 다른 경계선은 일치성을 조사하는 부분이 된다.

Sasaki와 Aoki는 중간일치기법의 효과를 극대화시키기 위하여 두 Chunk간의 경계선들이 연속된 몇 단계로 구성되도록하고 그 단계들을 뛰어넘어 일치성 체크를 성공시키도록 Partial-Matching, Partial-Fixing, Local-Collision과 같은 여러 기법들을 제시하였다[2-4]. 그러나 Sasaki와 Aoki가 공격하였던 이전 해쉬함수 알고리즘들에 비하여 ARIRANG의 단계함수는 연쇄변수들 간의 연관성이 높고 G 함수가 블록암호와 같은 비선형성과 차분전파성질 등을 제공하기 때문에 Partial-Fixing 및 Local-Collision 기법은 적용되기 어려움을 확인하였다. 특히, Local-Collision 기법은 공격이 시작되는 경계선 구간 (두 Chunk간의 두 간격 중 하나)을 공략하기 위한 것인데 이것의 적용이 불가능하므로 ARIRANG의 공격에서는 두 Chunk간의 간격이 한 개 뿐인 형태로 공격이 시작된다. Partial-Matching 기법은 두 Chunk의 일치성을 확인하는 경계선인 Matching-Check 구간에서 중립 워드와 관계없는 부분을 우선적으로 확인하는 것으로서, 시간복잡도를 절약하고 여러 단계를 건너뛰어 일치성을 확인할 수 있도록 하는 방법이다. ARIRANG에서는 최대 6단계까지 Partial-Matching을 적용할 수 있다.

3.2 워드이동성질

ARIRANG의 압축함수에서는 각 메시지 워드를 필요에 따라 최대 4단계까지 위로 옮길 수 있다. 이것을 ARIRANG의 워드이동성질이라 부른다. ARIRANG의 압축함수에 워드이동성질이 존재하는 이유는 G함수를 제외한 모든 부분이 비트연산 관점에서 선형이기 때문이다. 정방향으로 계산하면 G함수의 입력에 메시지 워드가 영향을 주게 되어 있으므로 워드의 이동이 어려우나, 역방향으로 계산하면 G함수와 독립적으로 4단계까지 이동이 가능하다. 이 성질을 적절하게 활용하면 선택된 중립워드에 대하여 가장 공격에 유리하도록 Chunk를 구성할 수 있다.



[그림 2] 메시지 워드 $W_{\sigma(2j)}$와 $W_{\sigma(2j+1)}$의 이동 (단계 j에서 단계 j-4까지)

3.3 Feedforward1과 중간일치기법의 관계

ARIRANG의 설계자들은 압축함수에 대한 중간일치공격을 차단하기 위한 방법으로 $Feedforward_1$을 적용하였음을 주장하였다[1]. 중간일치기법을 적용해 본 결과, $Feedforward_1$의 위치가 공격의 가능성 여부를 결정하는 중요한 요소임을 파악할 수 있었으며 그것은 다음과 같이 정리될 수 있다.

1. Inner Chunk가 $Feedforward_1$을 포함하면 공격불가능
2. Matching-Check 구간이 Inner Chunk와 Outer Chunk의 상위 경계선을 포함하고, Outer Chunk의 하위 구간이 $Feedforward_1$을 포함하면 공격불가능
3. Matching-Check 구간이 Inner Chunk와 Outer Chunk의 하위 경계선을 포함하고, Outer Chunk의 상위 구간이 $Feedforward_1$을 포함하면 공격불가능

IV. ARIRANG의 단계 축소 버전들에 대한 공격

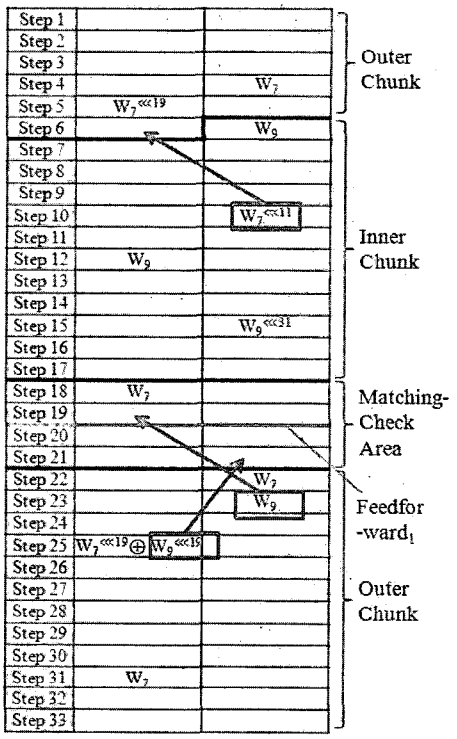
4.1 33단계 OFF 버전에 대한 공격

OFF 버전의 압축함수에 대하여 공격에 가장 적합한 chunk를 조사한 결과, 단계 1부터 단계 33까지의 33단계 OFF 압축함수에 대하여 W_7과 W_9를 중립워드로 이용하는 것이 가장 좋은 형태라는 결론을 얻을 수 있었다. OFF 버전이므로, $Feedforward_1$은 여전히 단계 19의 출력에 작용한다. W_7과 W_9를 중립워드로 이용하고 워드 이동 성질을 적절하게 적용하면, 전체 단계를 [그림 3]과 같이 Inner Chunk, Outer Chunk, Matching-Check 구간으로 나눌 수 있다. 워드의 이동에 따라 발생하는 단계 함수 정의에 대한 약간의 변화에 대해서는 설명을 생략하며, 각 i번째 단계함수의 연산 및 역연산을 각각 $state_{i+1} \leftarrow step_i(state_i)$과 $state_i \leftarrow step_i^{-1}(state_{i+1})$로 표시하기로 하자.

Matching-Check 구간에서 수행되는 Partial-Matching은 다음과 같다. 우선, 단계 18의 입력 $state_{18} = (a_{18}, b_{18}, c_{18}, d_{18}, e_{18}, f_{18}, g_{18}, h_{18})$과 단계 21의 출력 $state_{22} = (a_{22}, b_{22}, c_{22}, d_{22}, e_{22}, f_{22}, g_{22}, h_{22})$이 주어졌다고 가정하자. 그러면 $state_{18}$로부터 W_7과 상관없이 부분계산 $x_I = h_{18} \oplus G(e_{18} \oplus W_{10})^{\lll 7} \oplus W_9$을 수행할 수 있다. 또한, $state_{22}$로부터 W_9와 상관없이 부분계산 $x_O = G(G(b_{22}) \oplus c_{22}) \oplus G(b_{22})^{\lll 13} \oplus d_{22} \oplus b_1$을 수행할 수 있다. 만일 두 chunk가 일치한다면 반드시 $x_I = x_O$이 만족되어야하므로, 이것이 성립하는지 체크한다.

해쉬값 H^2가 주어졌을 때 $F(H^1, M^2) = H^2$를 만족시키는 (H^1, M^2)을 찾는 과정은 다음과 같이 설명된다.

1. b_7, c_7, d_7, e_7, e_6, f_6, g_6, h_6을 랜덤하게 선택하여 고정한다. W_7과 W_9를 제외한 다른 모든 메시지 워드들 또한 랜덤하게 선택하고 고정시킨다. 특히 W_{13}, W_{14}, W_{15}는 이것이 2-블록 공격으로 귀결될 것을 고려하여 적절한 값으로 선택된다. (예를들면, W_{13}은 0x80...0으로 고정시키고 W_{14}에는 0, W_{15}에는 29워드의 비트길이의 이진표현을 대입한다.)
2. W_9의 가능한 모든 값들에 대하여 a_7, f_7, g_7, h_7을 계산하고 $I = 7, \ldots, 17$에 대하여 $state_{i+1} \leftarrow step_i(state_i)$을 계산하여 $state_{18} = (a_{18}, \ldots, h_{18})$을 W_9, $x_I = h_{18} \oplus G(e_{18} \oplus W_{10})^{\lll 7} \oplus W_9$와 함께 테이블에 저장한다.
3. W_7의 가능한 각 값에 대하여 a_6, b_6, c_6, d_6을 계산하고 다음과 같은 계산을 수행한다.

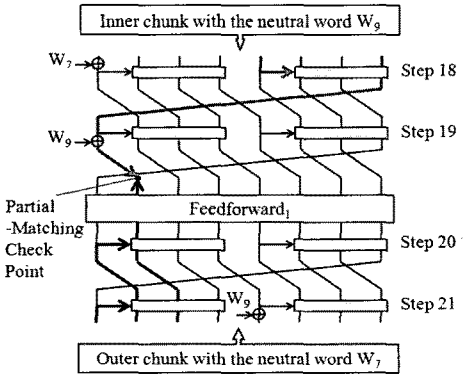


[그림 3] 33 단계 OFF 압축함수의 단계 분할
(W7과 W9외 다른 메시지 워드들은 생략)

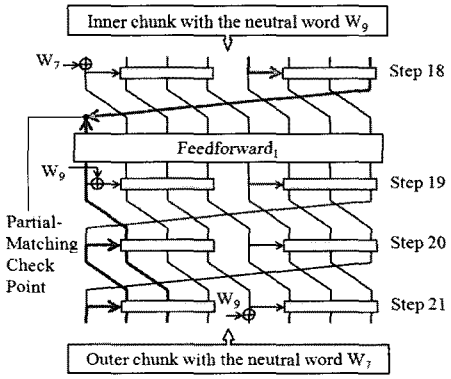
$$\begin{cases} state_i \leftarrow step_i^{-1}(state_{i+1}) \text{ for } i = 5, \ldots, 1 \\ state_{34} \leftarrow H^2 \oplus state_1 \\ state_i \leftarrow step_i^{-1}(state_{i+1}) \text{ for } i = 33, \ldots, 22 \end{cases}$$

위의 계산을 통하여 얻어지는 각 $state_{22}$로부터 $x_O = G(G(b_{22}) \oplus c_{22}) \oplus G(b_{22})^{\lll 13} \oplus d_{22} \oplus b_1$를 계산하고 $x_O = x_I$를 만족시키는 $(W_9, state_{18}, x_I)$쌍이 테이블에 있는지 조사한다. 만약 그러한 쌍이 있다면 W_7과 W_9를 이용하여 나머지부분에 대한 일치성도 조사한다. 모두 일치하는 쌍이 발견되면 그것에 대응되는 $H^1 = state_1$과 $M^2 = W_0 || \ldots || W_{15}$를 출력하고 알고리즘을 종료한다.

[그림 4]의 알고리즘을 이용하여 이전 절에서 설명된 대로 2-블록 역상을 찾는 공격알고리즘이 구성될 수 있다. ARIRANG-256의 경우, 위의 과정이 알맞은 (H^1, M^2)를 출력할 확률은 약 $2^{32} \cdot 2^{32} \cdot 2^{-256} =$



〔그림 4〕 33단계 OFF 압축함수에 적용되는 Partial-matching 기법



〔그림 5〕 31단계 MFF 압축함수에 적용되는 Partial- matching 기법

2^{-192}이다. 그러므로 위의 과정을 2^{192}번 반복하면 한 번의 성공을 기대할 수 있다. 위의 과정은 약 2^{32}번의 33단계 압축함수연산을 요구하므로 이 알고리즘의 복잡도는 2^{224}이다. 이것은 다시 2^{241}의 복잡도를 갖는 역상공격 알고리즘으로 변환될 수 있다. 같은 방식으로 계산하면, ARIRANG-512에 대한 역상공격 알고리즘의 복잡도는 2^{481}이다.

4.2 31 단계 MFF 버전에 대한 공격

MFF 압축함수에서는 Feedforward_1를 중간위치에 해당하는 단계함수의 출력값에 작용하는 것으로 고려한다. 그러므로, 이전 절에서 공격된 33단계 압축함수를 OFF가 아닌 MFF로 고려하면 Feedforward_1이 단계 16 또는 17의 출력에 작용하게되어 Partial-matching을 차단하므로 공격이 불가능하다. 그러나, MFF 압축함수의 경우 상위 단계를 제외시킴으로써, Feed forward_1의 위치를 아래로 옮기면 공격이 가능하다. 단계 3부터 단계 33까지의 31단계 MFF는 Feed forward_1이 단계 18의 출력에 작용함을 가정할 경우, 〔그림 3〕에서 단계 19에 위치했던 W_9를 다시 4단계 위로 옮김으로써 〔그림 5〕와 같은 Partial-matching을 적용할 수 있게 되어 공격이 가능하다. 복잡도는 이전 절의 공격과 동일하다.

V. 결 론

본 논문에서는 해쉬함수 ARIRANG의 축소 버전에 대한 역상공격을 제시하였다. ARIRANG의 압축함수의 축소는 OFF와 MFF의 두 가지 버전으로 고려되었는데, OFF는 33단계까지, MFF는 31단계까지 공격이 가능함이 확인되었다. 본 논문에서 소개된 모든 공격들은 ARIRANG-256에 대하여 약 2^{241}번의 압축함수 연산을, ARIRANG-512에 대하여 약 2^{481}번의 압축함수 연산을 소요한다. 이것은 ARIRANG의 역상공격에 대한 첫 안전성 분석결과이다. 여기에는 압축함수 내부에 적용되는 메시지 워드들의 이동성이 공격에 주요하게 사용되었다. 메시지 워드의 적용에 XOR 대신 덧셈 연산을 사용하면 메시지 워드의 이동을 차단할 수 있으며 알고리즘의 효율성도 저하되지 않으므로 안전성을 개선하는 간단한 방법으로서 고려될 만하다.

참 고 문 헌

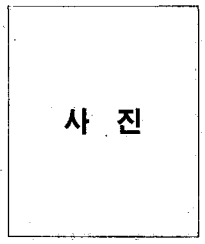
〔1〕 D.H. Chang, S.H. Hong, C.H. Kang, J.K. Kang, J.S. Kim, C.H. Lee, J.S. Lee, J.T. Lee, S.J. Lee, Y.S. Lee, J.I. Lim, and J.C. Sung, "ARIRANG: SHA-3 Proposal," available at http://csrc.nist.gov/groups/ST/hash/sha-3/Ro und1/

〔2〕 Y. Sasaki and K. Aoki, "Preimage Attacks on Step-Reduced MD5," ACISP 2008, LNCS 5107, pp. 282-296, 2008.

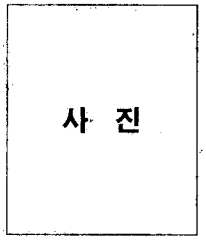
〔3〕 Y. Sasaki and K. Aoki, "Preimage Attacks on 3, 4, and 5-Pass HAVAL," ASIACRYPT 2008, LNCS 5350, pp. 253-271, 2008.

〔4〕 Y. Sasaki and K. Aoki, "A Preimage Attack for 52-Step HAS-160," ICISC 2008, LNCS 5461, pp. 302-317, 2008.

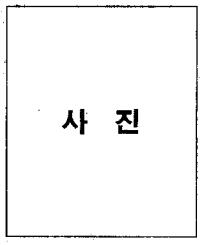
〈著者紹介〉



홍 득 조 (Deukjo Hong) 정회원
1999년 8월: 고려대학교 수학과 졸업
2001년 8월: 고려대학교 수학과 석사
2006년 2월: 고려대학교 정보보호대학원 박사
2006년 3월 ~ 2007년 12월: 고려대학교 정보보호대학원 연구교수
2007년 12월 ~ 현재: ETRI 부설연구소 연구원
〈관심분야〉 정보보호



김 우 환 (Woo-Hwan Kim) 정회원
1998년 2월: 서울대학교 수학과 졸업
2000년 2월: 서울대학교 수학과 석사
2004년 8월: 서울대학교 수학과 박사
2004년 11월 ~ 현재: ETRI 부설연구소 선임연구원
〈관심분야〉 정보보호



구 본 욱 (Bonwook Koo) 정회원
2001년 2월: 한양대학교 수학과 졸업
2003년 2월: 한양대학교 수학과 석사
2006년 2월: 한양대학교 수학과 박사과정 수료
2006년 11월 ~ 현재: ETRI 부설연구소 연구원
〈관심분야〉 정보보호