情報保護學會論文誌
第 21 卷 第 5 號, 2011. 10

오류 주입 공격을 이용한 HMAC에 대한 키 복구 공격*

정 기 태,[1†] 이 유 섭,[1] 성 재 철,[2‡] 홍 석 희[1]
[1]고려대학교 정보보호연구원, [2]서울시립대학교 수학과

A Key Recovery Attack on HMAC using Fault Injection Attack*

Kitae Jeong,[1†] Yuseop Lee,[1] Jaechul Sung,[2‡] Seokhie Hong[1]
[1]Center for Information Security Technologies, Korea University
[2]Department of Mathematics, University of Seoul

요 약

FDTC'05와 CISC-W'10에서는 오류 주입을 통하여 타깃 알고리즘의 라운드 수를 감소시킴으로써 AES와 Triple-DES의 비밀키를 각각 찾을 수 있음을 보였다. 본 논문에서는 이 공격 아이디어를 HMAC에 적용하여 비밀키를 복구할 수 있음을 보인다. 본 논문에서 제안하는 공격은 MD 계열의 해쉬 함수가 사용된 HMAC에 적용 가능하며, 매우 적은 계산 복잡도로 비밀키를 복구할 수 있다. 특히, HMAC-SHA-2에 대한 공격 결과는 HMAC-SHA-2에 대한 첫 번째 키 복구 공격 결과이다.

ABSTRACT

At FDTC'05 and CISC-W'10, the authors showed that if they decrease the number of rounds of AES and Triple-DES by using the fault injections, it is possible to recover the secret key of the target algorithms, respectively. In this paper, we propose a key recovery attack on HMAC by using the main idea of these attacks. This attack is applicable to HMAC based on MD-family hash functions and can recover the secret key with the negligible computational complexity. Particularly, the attack result on HMAC-SHA-2 is the first known key recovery attack result on this algorithm.

Keywords: Side channel analysis, Fault injection attack, Cryptanalysis, HMAC, Hash function

I. 서 론

최근 블록 암호의 안전성 분석 기법 중 하나로서, 암호 시스템의 실질적인 구현 과정에서 얻어지는 정보들을 이용하는 기법인 부채널 공격(side channel attack)에 대한 연구가 활발히 진행 중이다. 부채널 공격은 암호 알고리즘을 구현하였을 때 발생하는 연산 시간, 전력, 전자기파, 오류 등의 부가적인 정보를 이용하는 공격 방법으로서, 오류 주입 공격(fault attack), 시차 공격(timing attack), 전력 분석 공격(power attack) 등이 있다. 최초의 부채널 공격은 시차 공격으로서 Kocher에 의해 공개키 암호의 분석 방법으로 제안되었다[6]. 이 공격이 제안된 이후 전력 분석 공격, 오류 주입 공격 등의 다양한 기법들이 제안되었다.

오류 주입 공격[4]은 공격 대상 알고리즘에 전력 변화, 강제 클럭킹 등을 이용하여 오류를 발생시키고 이를 이용하여 비밀키 정보를 얻는 공격 기법이다. 이 공격은 공격자의 능력과 발생하는 오류의 유형에 따라 일시적(transient) 오류 주입 공격과 영구적(permanent) 오류 주입 공격으로 나뉜다. 일시적

접수일(2011년 1월 3일), 게재확정일(2011년 4월 19일)
* 본 연구는 지식경제부 IT R&D 사업의 일환으로 수행하였음(유비쿼터스 환경에서의 정보보호 서비스를 위한 프라이버시 강화 암호 기술 개발)
† 주저자, kite@cist.korea.ac.kr
‡ 교신저자, jcsung@uos.ac.kr

오류 주입 공격에서는 알고리즘 수행 시 순간적인 클록 값 변경 혹은 불규칙적인 전력 공급 등을 이용하여 특정 지점에서만 오류가 발생한다. 따라서 이 오류는 한시적으로만 공격 대상 알고리즘에 영향을 준다. 이와 달리, 영구적 오류 주입 공격에서는 알고리즘 수행 전에 레지스터 값을 특정한 값으로 고정하거나 레지스터를 파괴함으로서 영구적인 오류를 발생시킨다. 따라서 이 오류는 공격 대상 알고리즘에 계속 영향을 준다. 이 공격 기법은 단순히 암호 연산 시 발생하는 시간을 체크하는 시간 공격이나 전력을 체크하는 전력 분석 공격 보다는 좀 더 능동적인 공격 형태이다.

한편, FDTC'05[5]와 CISC-W'10[2]에서는 오류 주입을 통하여 타깃 알고리즘의 라운드 수를 감소시킴으로써 AES와 Triple-DES의 비밀키를 각각 찾을 수 있음을 보였다. 이 공격의 오류 주입 가정은 "for"문과 같은 반복문에 오류를 주입하여 타깃 알고리즘의 라운드 수를 1(AES), 15(Triple-DES)로 감소시킨다는 것이다.

본 논문에서는 [2,5]의 공격 아이디어를 대표적인 해쉬 함수 기반 메시지 인증 코드인 HMAC[3,8]에 적용하여 비밀키를 복구할 수 있음을 보인다. 본 논문에서 제안하는 공격의 오류 주입 가정은 오류 주입을 통하여 HMAC을 구성하는 압축 함수의 단계 수를 감소시킨다는 것이다. 그 결과, HMAC의 비밀키의 길이가 n 워드(32 비트, 64 비트 등)일 경우, $2n$개의 오류를 주입하면 간단한 산술 연산으로 n-워드 비밀키를 복구할 수 있다. 본 논문에서 제안하는 공격은 MD 계열의 해쉬 함수가 사용된 HMAC에 대해 적용 가능하며, 기제안된 HMAC-MD4와 HMAC-MD5에 대한 키 복구 공격[1,12] 보다 효율적으로 키 복구가 가능하다. 특히, HMAC-SHA-2에 대한 공격 결과는 HMAC-SHA-2에 대한 첫 번째 키 복구 공격 결과이다.

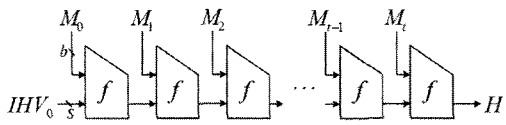
본 논문은 다음과 같이 구성되어 있다. 먼저, 2장에서는 공격을 소개하기에 앞서 MD 계열 함수, HMAC 알고리즘을 간략히 소개한 후, [2,5]에서 제안된 오류 주입 공격의 아이디어를 소개한다. 3장에서는 HMAC에 대한 키 복구 공격을 제안한다. 마지막으로 4장에서 결론을 맺는다.

II. 배경 이론

본 절에서는 MD 계열 해쉬 함수와 HMAC 알고리즘에 대하여 간략히 소개한 후, [2,5]에서 제안된 오류 주입 공격의 아이디어를 소개한다.

2.1 MD 계열 해쉬 함수

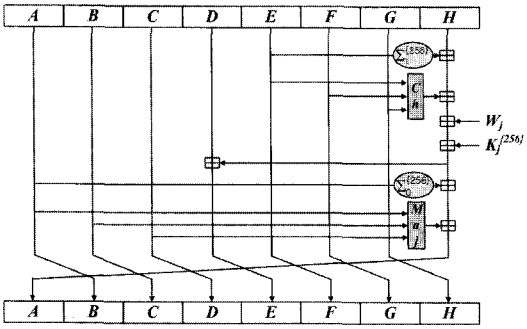
1990년, Rivest가 MD4[10]를 제안한 이후, MD5[11], RIPEMD[9], SHA-1, SHA-2[7] 등의 MD4의 설계 논리를 바탕으로 개발된 MD 계열 해쉬 함수가 발표되었다. MD 계열 해쉬 함수는 임의의 길이의 메시지에 대한 해쉬값을 구하기 위해, 메시지 M을 고정된 길이 b-비트 메시지 블록 $(M_0,\cdots,M_t)$로 나누고, [그림 1]과 같이 압축 함수 f를 이용하여 해쉬값 H를 구한다. MD-계열 해쉬 함수의 압축 함수는 고정된 길이의 입력값 (IHV_{i-1},M_{i-1})을 고정된 길이의 출력값 IHV_i으로 변환하는 함수이다. 즉, s-비트 IHV_{i-1}와 b-비트 메시지 블록 M_{i-1}을 입력 받아 새로운 s-비트 IHV_i를 출력한다. MD-계열 해쉬 함수의 압축 함수는 단계 함수를 반복하는 방법으로 구성되며, 단계 함수는 덧셈 연산, 부울 함수, 비트 단위



[그림 1] MD 계열 해쉬 함수 구조

[표 1] MD4, MD5, SHA-1, SHA-2의 주요 파라미터

해쉬 함수	메시지 블록(비트)	중간 해쉬값(비트)	해쉬값 길이(비트)	단계 수	워드 크기(비트)
MD4	512	128	128	48	32
MD5	512	128	128	64	32
SHA-1	512	160	160	80	32
SHA-224	512	256	224	64	32
SHA-256	512	256	256	64	32
SHA-384	1024	512	384	80	64
SHA-512	1024	512	512	80	64



〔그림 2〕 SHA-224/256의 단계 연산

순환 이동 연산으로 구성된다. 단계 함수 연산을 모두 마친 내부 상태값과 IHV_{i-1}을 더하여 IHV_i를 출력한다. 〔표 1〕은 대표적인 MD-계열 해쉬 함수인 MD4, MD5, SHA-1, SHA-2에 대한 주요 파라미터를 나타낸 것이다.

대표적인 MD-계열 해쉬 함수 중 하나인 SHA-2는 해쉬값의 길이에 따라 SHA-224/256/384/512로 구분된다. SHA-224/256은 512-비트 메시지 블록과 256-비트 IHV_{i-1}를 입력 받아 64번의 단계 함수 연산을 통해 갱신된 값에 IHV_{i-1}을 더하여 새로운 IHV_i를 생성한다. 〔그림 2〕는 SHA-224/256의 단계 함수를 나타낸 것이다. 여기서, 각각의 워드 A,B,C,D,E,F,G,H는 32 비트이고, W_j는 32-비트 메시지 워드, $C_j^{[256]}$는 32-비트 상수이다. 각 단계 함수에 사용되는 연산은 다음과 같다.

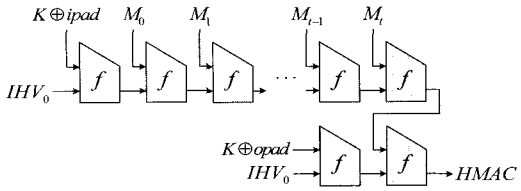
- $Ch(x,y,z)=(x\vee y)\oplus(\neg x\vee z)$.
- $Maj(x,y,z)=(x\vee y)\oplus(x\vee z)\oplus(y\vee z)$.
- $\Sigma_0(x)=(x^{\gg 2})\oplus(x^{\gg 13})\oplus(x^{\gg 22})$.
- $\Sigma_1(x)=(x^{\gg 6})\oplus(x^{\gg 11})\oplus(x^{\gg 25})$.

메시지 $M=(M_0,\cdots,M_t)$에 대해, 압축 함수를 수행한 후, SHA-224는 압축 함수의 출력값 IHV_{t+1}의 최상위 224-비트 값을 해쉬값으로 출력하고, SHA-256은 IHV_{t+1}을 해쉬값으로 출력한다.

SHA-384/512의 구조는 SHA-224/256과 유사하며, 연산의 기본이 되는 워드의 크기가 32 비트에서 64 비트로 확장된다. 단계 함수의 경우, 총 80 단계가 수행된다.

2.2 HMAC

HMAC은 Bellare 등이 제안한 해쉬 함수 기반



〔그림 3〕 HMAC 구조

메시지 인증 코드로서, 단순한 구조로 설계되었으며 안전성에 대한 증명이 이루어졌다. 현재, ANSI, IETF, ISO, NIST 표준으로 제정되어 있으며, SSL, TLS, SSH, IPsec 등 다양한 프로토콜에서 사용되고 있다.

HMAC은 〔그림 3〕과 같이 두 번의 해쉬 함수 연산을 통하여 메시지 인증 코드를 계산한다. 첫 번째로 비밀키 K와 512-비트 상수 $ipad(=0x363636\cdots)$를 XOR한 값 $K\oplus ipad$를 메시지의 앞부분에 연접하여 해쉬값을 구한다. 그리고 이 해쉬값의 앞부분에 비밀키 K와 512-비트 상수 $opad(=0x5c5c5c5c\cdots)$를 XOR한 값 $K\oplus opad$를 연접한 후, 이 값에 대한 해쉬값을 메시지 인증 코드값($HMAC$)으로 출력한다. 이를 수식으로 나타내면 비밀키 K와 메시지 M에 대한 HMAC은 다음과 같다.

$$HMAC(K,M)=H((K\oplus opad)\|H((K\oplus ipad)\|M)$$

2.3 라운드 수에 대한 오류 주입 공격

〔5〕에서는 "for"문과 같은 반복문에 오류를 주입하여 라운드 수를 감소시킴으로써 AES의 비밀키를 복구하는 공격을 제안하였다. 이 공격에서는 오류 주입을 통하여 AES를 1 라운드만을 수행시킨 뒤, 라운드 1의 출력값을 암호문으로 출력한다. 이를 통해 라운드 1의 라운드 키를 복구함으로써, AES의 비밀키를 복구할 수 있다.

한편, 〔2〕에서는 〔5〕에서 제안된 공격 아이디어를 Triple-DES에 적용하였다. 오류 주입을 이용하여 Triple-DES에서 동작하는 세 개의 DES 중 타깃 DES를 15 라운드만을 수행하도록 한 후, 라운드 16의 라운드 키를 복구하였다. 그 결과, 9개의 오류를 주입하여 2^{24}의 계산 복잡도로 Triple-DES의 192-비트 비밀키를 복구할 수 있다.

III. HMAC에 대한 키 복구 공격

본 논문에서 제안하는 공격은 MD 계열 해쉬 함수

가 사용된 HMAC에 대해 적용 가능하므로, 본 절에서는 대표적인 MD 계열 해쉬 함수인 SHA-2가 사용된 HMAC-SHA-2에 대한 키 복구 공격만을 소개한다. 다른 MD 계열 해쉬 함수가 사용된 경우에도 유사한 방법으로 키 복구가 가능하다.

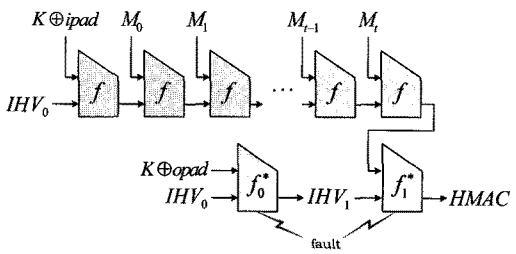
3.1 오류 주입 가정

본 논문에서 제안하는 공격에서는 [2,5]의 오류 주입 가정을 HMAC의 압축 함수 f에 적용하여 HMAC의 비밀키를 복구한다. 즉, HMAC에서 마지막으로 수행되는 두 개의 압축 함수에 오류를 주입하는 것을 가정한 후, 공격자는 오류 주입을 통해 마지막으로 수행되는 두 개의 압축 함수의 단계 수를 조절한다. [그림 4]는 본 논문에서 제안하는 공격의 공격 모델을 나타낸 것이다. 설명의 용이성을 위해, 오류를 주입하는 압축 함수를 f_0^*, f_1^*로 표기하기로 한다. 그래서 공격자는 f_0^*, f_1^*에 오류를 주입하여 공격자가 원하는 단계 수만을 수행하도록 한 후, HMAC-SHA-2의 비밀키 K를 복구한다.

3.2 HMAC-SHA-256에 대한 키 복구 공격

앞 절에서 언급한 것처럼, SHA-224/256/384/512는 워드 크기와 같은 주요 파라미터를 제외하고 동일한 구조를 갖는다. 따라서 본 소절에서는 HMAC-SHA-256에 대한 키 복구 공격만을 자세히 설명한다. 그리고 3.3절에서 나머지 HMAC-SHA-224/384/512의 경우에 대해 간단히 소개한다.

먼저, SHA-256의 워드 크기는 32 비트이므로, 비밀키 $K(=K_0\|K_1\|\cdots\|K_{n-1})$의 길이가 $(32\cdot n)$ 비트라고 가정한다. HMAC-SHA-256에 대한 키 복구 공격은 다음과 같이 32-비트 K_i를 복구하는 과정을 순차적으로 n번 수행한다$(i=0,\cdots,n-1)$.

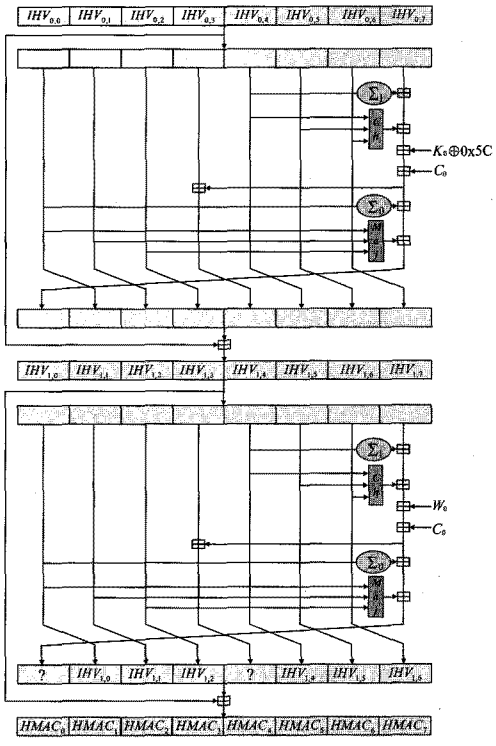


[그림 4] HMAC에 대한 오류 주입 공격 모델

1. [메시지 선택] 임의의 메시지 M을 선택한다.
2. [K_i 복구] K_i를 복구하기 위해, 다음과 같은 과정을 n번 수행한다.
   2-1. 오류를 주입하여 f_1^*와 f_2^*를 각각 $i+1$, 1 단계만을 수행하도록 한 후 얻은 $HMAC$으로부터, IHV_1에 대한 방정식을 구성한다.
   2-2. 오류를 주입하여 f_1^*와 f_2^*를 각각 $i+1$, 2 단계만을 수행하도록 한 후 얻은 $HMAC$으로부터, IHV_1에 대한 방정식을 구성한다.
   2-3. 구성한 방정식으로부터, IHV_1을 계산한다.
   2-4. IHV_0와 $K_t(t<i)$를 이용하여 K_i를 복구한다.

공격자는 임의의 메시지 M을 선택하고, f_1^*와 f_2^*에 오류를 주입하여 각각 1 단계만을 수행하도록 하여 [그림 5]와 같이 $HMAC$을 생성한다. 이를 통해 식 (1)과 같이 IHV_1에 대한 6개의 방정식을 구성한다.

① $IHV_{1,0}+IHV_{1,1}=HMAC_1$.
② $IHV_{1,1}+IHV_{1,2}=HMAC_2$.
③ $IHV_{1,2}+IHV_{1,3}=HMAC_3$.

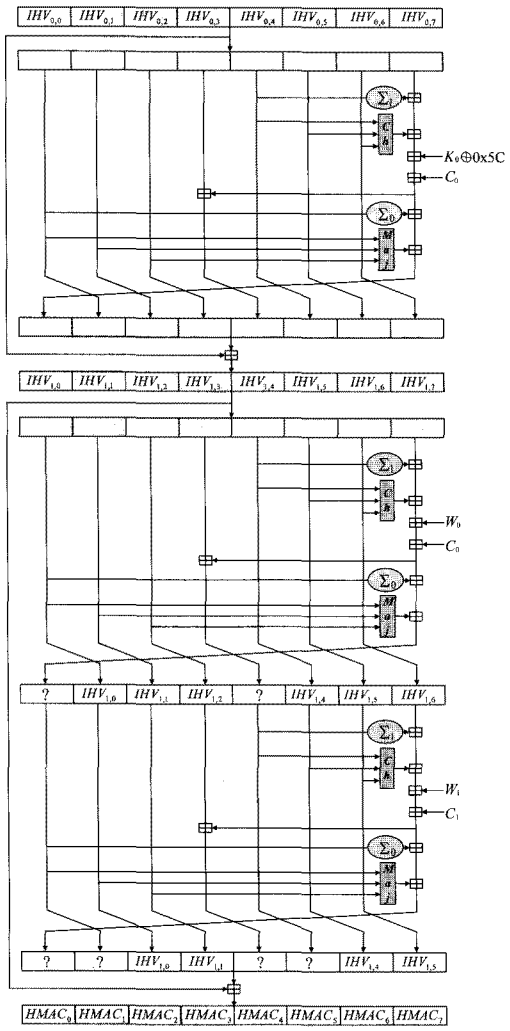


[그림 5] f_1^*, f_2^*에 오류를 주입하여 각각 1 단계만을 수행

④ $IHV_{1,4}+IHV_{1,5}=HMAC_5$.
⑤ $IHV_{1,5}+IHV_{1,6}=HMAC_6$. (1)
⑥ $IHV_{1,6}+IHV_{1,7}=HMAC_7$.

그리고 f_1^*와 f_2^*에 오류를 주입하여 각각 1, 2 단계만을 수행하도록 하여 [그림 6]과 같이 $HMAC$을 수행한다. 이를 통해 식 (2)와 같은 IHV_1에 대한 4개의 방정식을 추가적으로 구성하고, 방정식의 해를 구함으로써 IHV_1을 복구한다.

① $IHV_{1,0}+IHV_{1,2}=HMAC_2$.
② $IHV_{1,1}+IHV_{1,3}=HMAC_3$.
③ $IHV_{1,4}+IHV_{1,6}=HMAC_6$. (2)
④ $IHV_{1,5}+IHV_{1,7}=HMAC_7$.



[그림 6] f_1^*, f_2^*에 오류를 주입하여 각각 1, 2 단계만을 수행

IHV_0와 복구한 IHV_1에 대해, 다음과 같은 식을 통하여 K_0를 복구한다.

$$\begin{aligned} K_0 \oplus 0x5C = {} & IHV_{0,7}+\Sigma_1(IHV_{0,4}) \\ & +Ch(IHV_{0,4},IHV_{0,4},IHV_{0,4})-IHV_{1,1} \\ & +IHV_{0,0}+Maj(IHV_{0,0},IHV_{0,1},IHV_{0,2}) \\ & +\Sigma_0(IHV_{0,0})+C_0. \end{aligned} \quad (3)$$

K_0를 복구한 후, 동일한 메시지 M에 대해서 f_1^*와 f_2^*에 오류를 주입하여 각각 2, 1 단계만을 수행하도록 하여 위와 유사하게 IHV_1에 대한 방정식을 구성한다. 그리고 f_1^*와 f_2^*에 오류를 주입하여 각각 2, 2 단계만을 수행하도록 하여 IHV_1에 대한 추가적인 방정식을 구성하여 IHV_1을 복구한다. 그리고 이전 단계에서 복구한 K_0를 이용하여 IHV_1과 IHV_0로부터 K_1을 복구한다. 이와 같은 과정을 반복하여 K를 모두 복구한다.

단계 2에서는 각각의 K_i를 복구하기 위해서 2개의 오류를 주입하여 얻은 $HMAC$으로부터 10개의 방정식을 통해 IHV_1을 복구한다. 그리고 복구된 값을 이용하여 SHA-2의 i 단계의 연산을 통하여 K_i를 복구한다. 그러므로 $(32\cdot n)$-비트 비밀키를 복구하기 위한 계산량은 $O(n)$으로 거의 없다. 또한, 공격에 필요한 오류 주입수는 총 $2n$이다.

3.3 HMAC-SHA-224/384/512에 대한 키 복구 공격

SHA-224/384/512에 대해서도 위 소절에서 소개한 공격과 유사한 방법을 통해 비밀키를 복구할 수 있다. SHA-224의 워드 크기는 32 비트이므로, HMAC-SHA-224에 대한 키 복구 공격을 수행할 경우에는 $(32\cdot n)$-비트 비밀키를 가정한다. 또한, HMAC-SHA-384/512에 적용할 경우, $(64\cdot n)$-비트 비밀키를 가정한다.

우선, SHA-512는 워드 크기가 64 비트로 증가한 것을 제외하고는 동일한 방법으로 공격이 가능하다. 하지만 SHA-224/384의 경우, $HMAC$의 값으로 좌측 224/384 비트를 출력하기 때문에 단계 2-1과 단계 2-2에서 보다 적은 수의 식만을 구성할 수 있다. SHA-224의 경우는 식 (1)-⑥과 식 (2)-④를 제외한 8개의 방정식을 구성하여, $IHV_{1,0}, IHV_{1,1}, \cdots IHV_{1,6}$을 복구할 수 있다. 하지만, 식 (3)에서 $IHV_{1,0}$만이 사용되기 때문에 동일하게 K_i를

복구할 수 있다. SHA-384의 경우, 식 (1)-⑤,⑥과 식 (2)-③,④를 제외한 6개의 방정식을 구성하여 $IHV_{1,0}, IHV_{1,1}, IHV_{1,2}, IHV_{1,3}$을 복구할 수 있으므로 동일하게 K_i를 복구할 수 있다.

IV. 결 론

본 논문에서는 HMAC에 대한 오류 주입 공격을 제안하였다. 본 논문에서 제안하는 공격은 MD4, MD5, SHA-1, SHA-2 등의 MD 계열 해쉬 함수가 적용된 HMAC에 대해 적용 가능하며, HMAC-SHA-2에 대한 공격 결과는 이 알고리즘에 대한 첫 번째 키 복구 공격 결과이다.

본 논문에서 제안한 공격을 이용하여 HMAC에서 가장 마지막에 수행되는 두 개의 압축 함수에 오류를 주입하여 단계 수를 감소시킴으로서 중간 해쉬값 (IHV_1)에 대한 방정식을 구성할 수 있다. 이를 통해, $2n$개의 오류를 주입하여 매우 낮은 계산 복잡도로 n-워드 비밀키를 복구할 수 있다. 하지만 [2,5]에서 제안된 공격과는 달리, 본 공격에서는 두 개의 압축 함수에 동시에 오류를 주입하여야 한다. 따라서 이론적으로는 본 공격이 HMAC에 적용되지만, 실제 환경에서의 적용 가능성은 구현을 통해서만 확인 가능하다. 따라서 향후 계획으로서, 본 공격의 실제 적용 가능성을 연구할 것이다.

참고문헌

[1] 강진건, 이제상, 성재철, 홍석희, 류희수, "HMAC/NMAC-MD4에 대한 향상된 키 복구 공격," 정보보학회논문지, 19(2), pp. 63-74, 2009년 4월.

[2] 최두식, 오두환, 배기석, 문상재, 하재철, "반복문 오류 주입을 이용한 Triple DES 차분 오류 공격," 한국정보보호학회 동계학술대회발표집, pp. 308-312, 2010년 12월.

[3] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," Crypto'96, LNCS 1109, pp. 1-15, 1996.

[4] D. Boneh, R. DeMillo, and R. Lipton, "On the importance of checking cryptographic protocols for faults," Eurocrypt'97, LNCS 1233, pp. 37-51, 1997.

[5] H. Choukri and M. Tunstall, "Round Reduction Using Faults," Fault Diagnosis an Tolerance in Cryptography - FDTC'05, pp. 13-24, Sep. 2005.

[6] P. Kocher, "Timing attacks on implementation of Diffie-Hellman," Crypto'96, LNCS 1109, pp. 104-113, 1996.

[7] National Institute of Standards and Technology, "FIPS PUB 180-2: Secure Hash Standard," FIPS PUB 180-2, Aug. 2002.

[8] National Institute of Standards and Technology, "FIPS PUB 198: The Keyed-Hash Message Authentication Code (HMAC)," FIPS PUB 198, Mar. 2002.

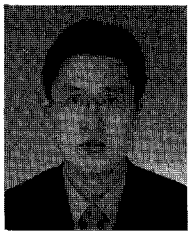
[9] B. Preneel, A. Bosselaers, and H. Dobbertin, "The Cryptographic Hash Function RIPEMD-160," RSA'97, CryptoBytes 3(2), pp 9-14, Sep. 1997.

[10] R. Rivest, "The MD4 Message Digest Algorithm," RFC 1320, Apr. 1992.

[11] R. Rivest, "The MD5 Message Digest Algorithm," RFC 1321, Apr. 1992.

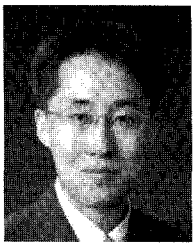
[12] L. Wang, K. Ohta, and N. Kunihiro, "New Key-Recovery Attacks on HMAC/NMAC-MD4 and NMAC-MD5," Eurocrypt'08, LNCS 4965, pp. 237-253, 2008.

〈著者紹介〉



정 기 태 (Kitae Jeong) 학생회원
2004년 2월: 고려대학교 수학과 학사
2006년 2월: 고려대학교 정보보호대학원 석사
2011년 8월: 고려대학교 정보경영공학전문대학원 박사
2011년 9월~현재: 고려대학교 정보보호연구원 박사후연구원
〈관심분야〉 대칭키 암호의 분석 및 설계

이 유 섭 (Yuseop Lee) 학생회원
2007년 2월: 서울시립대학교 수학과 학사
2007년 3월~현재: 고려대학교 정보경영공학전문대학원 석박사 통합과정
〈관심분야〉 스트림 암호, 해쉬 함수의 분석 및 설계



성 재 철 (Jaechul Sung) 종신회원
1997년 8월: 고려대학교 수학과 학사
1999년 8월: 고려대학교 수학과 석사
2002년 8월: 고려대학교 수학과 박사
2002년 8월~2004년 1월: 한국정보보호진흥원 선임연구원
2004년 2월~현재: 서울시립대학교 수학과 부교수
〈관심분야〉 암호 알고리즘 설계 및 분석

홍 석 희 (Seokhie Hong) 종신회원
1995년 2월: 고려대학교 수학과 학사
1997년 2월: 고려대학교 수학과 석사
2001년 8월: 고려대학교 수학과 박사
1999년 8월~2004년 2월: (주) 시큐리티 테크놀로지스 선임연구원
2003년 8월~2004년 2월: 고려대학교 정보보호기술연구센터 선임연구원
2004년 4월~2005년 2월: K.U. Leuven, ESAT/SCD-COSIC 박사후연구원
2005년 3월~2008년 8월: 고려대학교 정보보호대학원 조교수
2008년 9월~현재: 고려대학교 정보경영공학전문대학원 부교수
〈관심분야〉 대칭키·공개키 암호 분석 및 설계, 컴퓨터 포렌식