韓國 컴퓨터情報學會 論文誌
第10卷 第2號, 2005. 5.

2005-10-2-1-5

블록 암호 알고리즘을 이용한 MAC 분석

서 창 호*, 윤 보 현**, 맹 승 렬***

Security Analysis of MAC Algorithm using Block Cipher

Chang-Ho Seo*, Bo-Hyun Yun**, Sung-Reol Maeng***

요 약

본 논문에서는 전송되는 메시지의 무결성과 출처 인증을 위해 광범위하게 사용되는 메시지 인증 코드(Message Authentication Code :MAC) 알고리즘을 제안하고 안전성을 분석한다. 제안된 MAC 알고리즘은 기본 블록 암호로 64-비트 블록과 56-비트 키를 가진 64비트 블록 암호 알고리즘을 이용하여, MAC 값의 길이를 64-비트와 32-비트를 사용하였을 경우의 안전성을 비교한다. 또한, 128-비트 블록과 128-비트 키를 가진 128비트 블록 암호 알고리즘을 이용하여, MAC 값의 길이를 128비트와 64-비트를 사용하였을 경우의 안전성을 비교한다. 그래서 메시지의 길이와 MAC값의 길이에 따른 위장 공격의 안전성을 분석한다.

Abstract

This paper proposes and analyzes the MAC(Message Authentication Code) algorithm that is used for the transition integrity and the entity authentication of message. The MAC algorithm uses the DES algorithm which has 64-bit block and 56-bit key and we compare the security according to 64-bit and 32-bit length of MAC value. Moreover, we use the SEED algorithm which has 128-bit block and 128-bit key and compare the security according to 128-bit and 64-bit length of MAC value. We analyze the security the forgery attack according to length of message and length of MAC value.this paper, a coarse-to-fine optical flow detection method is proposed. Provided that optical flow gives reliable approximation to two-dimensional image motion, it can be used to recover the three-dimensional motion, but usually to get the reliable optical flows are difficult. The proposed algorithm uses Horn's algorithm for detecting initial optical flow, then Thin Plate Spline is introduced to warp a image frame of the initial optical flow to the next image frame. The optical flow for the warped image frame is again used iteratively until the mean square error between two image sequence frames is lowered. The proposed method is experimented for the real moving picture image sequence. The proposed algorithm gives dense optical flow vectors.

▸ Keyword : 메시지 인증코드(MAC), 해쉬함수(Hash Function), 위장공격(Forgery Attack), 키회수 공격(Key Recovery Attack)

• 제1저자 : 서창호
• 접수일 : 2005.03.25, 심사완료일 : 2005.05.20
* 공주대학교 응용수학과(정보보호전공) 부교수, ** 목원대학교 컴퓨터교육학과 조교수, **공주대학교 멀티미디어학부 교수
※ 본 연구는 정보통신부 기초기술연구지원사업 ITA-B1220-0401-0087-0001 연구 결과로 수행하였음

Ⅰ. 서 론

해쉬 함수란 임의의 길이를 갖는 입력을 고정된 길이의 결과로 압축하는 함수를 의미한다. 해쉬 함수는 크게 비밀키의 유무에 따라 메시지 인증 코드(Message Authentication Code :MAC)와 MDC(Manipulation Detection Code)로 나뉘며 다시 MDC는 일방향 해쉬 함수(One-way Hash Function)와 충돌 회피 해쉬 함수(Collision Resistant Hash Function)으로 나누어진다.[1] 메시지 인증 코드(Message Authentication Code :MAC)는 전송되는 메시지의 무결성과 출처 인증을 위해 광범위하게 사용된다.

MAC는 송신자와 수신자가 같은 비밀키 K를 가지고 있는 시스템 환경에 적용된다. 메시지 x를 보호하기 위하여 송신자는 K와 x의 함수인 $MAC_K(x)$를 계산하여 메시지 x에 부가하여 보낸다. 메시지 x를 받은 수신자는 공유하고 있는 비밀키 K를 이용하여 $MAC_K(x)$를 계산한 후 전송된 MAC 값과 비교함으로써 메시지 x가 변조되지 않고 정당한 사용자로부터 온 것임을 검증한다.

암호 알고리즘을 기반으로 하는 MAC 알고리즘들이 많이 제안되었고, 대표적인 MAC 알고리즘으로는 블록 암호 알고리즘을 이용하여 메시지 인증 코드(MAC)를 설계하는 여러 가지의 방법이 제기되어 왔다. 이 중 Cipher Block Chaining(CBC) 모드를 이용한 CBC-MAC이 가장 널리 상용되어 지고 있다. 국제 표준 ISO/IEC 9797-1에서도 CBC-MAC과 그 변형된 형태로 6 가지의 알고리즘이 제시되어 있다[2]. 이들은 세 가지 메시지 패딩 방법 중 한 가지를 사용하고, 두 가지 입력 변환과 세 가지 출력 변환으로 구성된다.

MAC 알고리즘의 안전성의 평가 방법으로는 위장 공격(forgery attack)과 키 복구 공격(key recovery attack)에 대한 안전성으로 나눌 수 있다. 위장공격에 안전하다는 것은 비밀키 K를 알지 못하는 공격자가 임의의 새로운 메시지 x와 대응되는 $MAC_K(x)$를 계산하는 것이 계산량적으로 불가능하다는 것이며, 키 복구 공격에 안전하다는 것은 메시지와 MAC 쌍들로부터 공격자가 비밀키 K를 얻는 것이 불가능하다는 것이다. 이들 CBC-MAC 알고리즘의 안전성 분석은 ISO/IEC 9797-1 문서와 Preneel-Oorschot[3], Knudsen[4], Brincat-Mitchell[5] 등에 의해 이루어졌다. 또한 ANSI X9.19[6] retail MAC으로 명명된 CBC-MAC 형태는 Preneel-Oorschor[7], Knudsen-Preneel[8]에 의해 자세히 분석되었다.

본 논문에서는 제안된 MAC 알고리즘에 대하여 위장 공격 관점에서 메시지의 길이와 MAC 값의 길이의 변화에 따른 공격 수행 복잡도를 구하여 안전성을 분석하였다. 제안된 MAC 알고리즘 기본 블록 암호로 64-비트 블록과 56-비트 키를 가진 DES 알고리즘을 이용하여, MAC 값의 길이를 64-비트와 32-비트를 사용하였을 경우의 안전성을 비교한다. 또한 128-비트 블록과 128-비트 키를 가진 SEED 알고리즘을 이용하여, MAC 값의 길이를 128비트와 64-비트를 사용하였을 경우의 안전성을 비교한다. 제안된 MAC 알고리즘의 메시지와 MAC값의 길이의 변화에 따른 안전성을 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 일반적인 MAC 알고리즘을 설명한다. 3장에서는 위장 공격 알고리즘의 개념을 설명하고, 3GPP-MAC 알고리즘의 수행 복잡도를 계산하여 메시지의 길이와 MAC 값의 길이의 변화에 따른 안전성을 비교 분석한다. 마지막으로, 4장에서는 결론을 맺는다.

Ⅱ. 준비 단계

2.1 MAC 알고리즘

CBC-MAC 알고리즘은 하나의 키를 이용하여 Cipher Block Chaining 모드에서의 마지막 출력의 값을 메시지에 대한 인증 코드로 사용한다. 여기서 마지막 출력의 값 중 특정 비트 부분만을 잘라내어 사용하기도 한다. 메시지의 블록 길이가 t개라고 가정하면 t번의 블록 암호를 이용한 암호화 과정으로 MAC 값을 얻을 수 있다. 본 논문에서 다루는 MAC 알고리즘은 ISO-IEC 9797-1에서 정의한 6가지 형태의 CBC-MAC과는 다른 형태의 변형된 CBC-MAC을 이용한다. 기존 CBC-MAC과의 가장 큰 차이점은 최종 출력 변환의 입력 값으로 이전의 블록 암호 출력 값으로 나온 모든 연쇄 변수를 XOR 한다는 것이다.

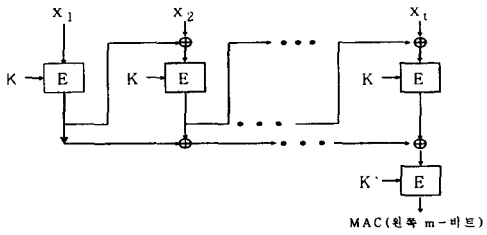

그림 1. MAC 알고리즘
Fig 1. MAC Algorithm

위의 MAC 알고리즘은 아래와 같이 동작한다. 기본 블록 암호는 n-비트 블록들을 가지고 k-비트의 비밀키를 이용한다고 가정하자. 만약 X가 n-비트 블록을 가진다면 비밀키 K를 사용하여 X의 암호화를 $E_K(X)$로 표기한다. 메시지 x는 먼저 n의 배수가 되도록 메시지의 끝에 하나의 1과 나머지 0을 이용하여 패딩하고 t개의 n-비트를 블록들 $x_1 \| x_2 \| \cdots \| x_t$로 분할된다. 여기에서 기호 $\|$는 연결을 뜻한다. 앞으로 우리는 패딩된 메시지만 고려할 것이다. 키 쌍은 K, K'을 사용하여, K'은 K로부터 다음과 같이 유도될 수 있다.

$$K' = K \oplus C$$

여기서 $C = 0101\cdots01$(128비트)인 이진 상수이다. (그림 1)에서 보듯이, MAC 값의 계산은 다음과 같다.

$$H_1 = E_K(x_1),$$
$$H_i = E_K(x_i \oplus H_{i-1}), \quad (2 \le i \le t),$$
$$MAC = E_{K'}(H_1 \oplus H_2 \oplus \cdots \oplus H_t).$$

여기서 H_i는 연쇄 변수라 불리며 MAC 값은 위의 마지막 식에서 주어진 MAC의 최상위 m-비트 $(m \le n)$만을 이용한다. 기본 블록 암호에 따라 메시지의 길이 n과 MAC 값의 길이 m, 그리고 비밀키의 길이 k가 결정된다. 또한 MAC 값의 결정 방법은 두 가지 경우로 나눌 수 있다.(즉, $m < n$인 경우와 $m = n$인 경우)

2.2 기본 정리

일반적으로 블록 암호에 기반한 CBC-MAC인 경우 블록 암호가 랜덤한 순열이라는 가정이 필요하다. 본 논문에서도 블록 암호 SEED의 랜덤성은 가정하고, 아래 보조 정리에서 SEED가 순열임을 보인다.

▶ 보조정리 1.

기본 블록 암호는 순열이다.

증명) 기본 블록 암호 알고리즘의 블록 구조는 다음과 같이

$$K(L, R) = (R, L \oplus f(R))$$

이고, K 의 역은

$K^{-1}(L, R) = (R \oplus f(L), L)$이다. 여기서 L, R은 임의의 64-비트 이진벡터이다. 그러면

$$\begin{aligned} K^{-1}(K(L,\ R)) &= K^{-1}(R,\ L \oplus f(R)) \\ &= (L \oplus f(R) \oplus f(R),\ R) \\ &= (L,\ R), \end{aligned}$$

이고 비슷한 방법으로

$$\begin{aligned} K(K^{-1}(L,\ R)) &= K(R \oplus f(L),\ L) \\ &= (L,\ R \oplus f(L) \oplus f(L)) \\ &= (L,\ R) \end{aligned}$$

이다. 따라서 기본 블록 암호 알고리즘은 순열이다.

다음 두 보조 정리는 '생일 역설'로 잘 알려진 내용이다.

▶ 보조정리 2

1에서 n까지의 번호가 적힌 n개의 공을 가진 한 개의 단지가 있다고 하자. 한 번 시행에 한 개씩 복원 추출을 통해 단지에서 r개의 공을 꺼내어 그 번호를 기록한다고 하자. n이 충분히 크고, $r = O(\sqrt{n})$일 때, 적어도 한 쌍의 충돌이 발생할 확률은 대략

$$1 - \exp\left(-\frac{r(r-1)}{2n}\right) \approx 1 - \exp\left(-\frac{r^2}{2n}\right)$$

이다.

▶ 보조정리 3

1에서 n까지의 번호가 적인 n개의 흰 공을 가진 단지와 1에서 n까지의 번호가 적인 n개의 빨간 공을 가진 단지가 있다고 하자. 한 번 시행에 한 개씩 복원 추출을 통해 두 단지에서 각각 r_1개와 r_2개의 공을 꺼내어 그 번호를 기록한다고 하자. n이 충분히 크고, $r_1 = r_2 = r$, $r = O(\sqrt{n})$이면, 적어도 한 쌍의 충돌이 발생할 확률은 대략

$$1 - \exp\left(-\frac{r^2}{n}\right)$$

이다.

▶ 보조정리 4

Preneel-Oorshot[3]에 의하면 대부분의 공격은 두 메시지가 같은 MAC 값을 가지는 충돌 쌍에 기반 한다. 따라서 본 논문에서도 두 메시지가 같은 MAC 값과 같은 마지막 연쇄 변수를 가질 때 내부 충돌 (internal collision)쌍이라 부르고, 내부 충돌 쌍이 아닌 충돌 쌍을 외부 충돌 (external collision)쌍이라 부른다.

ISO/IEC 9797-1[2]에서 사용된 표현 방법에 따라 공격자에게 필요한 자원을 명확하게 규정하기 위해 $[a, b, c, d]$ 표기를 사용한다. 여기서 a는 오프라인 블록 암호 암호화 횟수, b는 알려진 메시지/MAC 쌍의 수, c는 선택 메시지/MAC 쌍의 수, d는 온라인 MAC 검증 횟수를 나타낸다.

III. 제안된 MAC 알고리즘 안전성 분석

3.1 Knudsen-Mitchell 위장 공격법의 분석

본 절에서는 MAC의 알려진 공격법 중 가장 효율적인 Knudsen과 Mitchell[9]의 위장 공격 법을 고찰한다. 먼저 그들의 공격을 심층적으로 분석하고 구체적인 공격 수행 알고리즘의 복잡도를 계산하여 CBC-MAC의 변형 알고리즘에 대한 안전성을 분석한다. Knudsen-Mitchell의 위장 공격법은 다음과 같다. 공격자가 $2^{\frac{n+m}{2}}$ 개의 알려진 메시지/MAC 쌍을 가지고 있다고 가정하자. $2^{\frac{n+m}{2}}$ 개의 알려진 MAC 값들은 MAC 값에 따라 2^m개의 집합으로 분할하자. 적어도 한 개의 집합은 모두 MAC 값이 같은 $2^{(n-m)/2}$개 이상의 메시지를 포함할 것이고 그 집합을 선택한다. 이들 메시지 중에서 2개는 버려진 $(n-m)$-비트에서도 충돌할 것으로 기대된다. 선택한 집합의 모든 메시지 $x = x_1 \| x_2 \| \cdots \| x_t$에 대해, $2^{n/2}$개의 랜덤하게 선택한 $y(i)$로 만든 $x \| y(i)$ 메시지들과 대응하는 MAC 값들은 얻는다. 선택한 메시지/MAC 쌍의 총 개수는 $x^{(n-m)/2} \cdot 2^{n/2} = 2^{n-m/2}$ 이므로 생일 역설에 의하여 좋은 확률로 같은 MAC 값을 가지면서 $H_t \oplus y = H'_s \oplus y'$가 되는 메시지들 $x_1 \| \cdots \| x_t \| y$와 $x'_1 \| \cdots \| x'_s \| y'$가 존재한다. 즉, $2^{n-m/2}$개의 선택 메시지/MAC 쌍의 집합에서 내부 충돌 쌍이 존재한다. $2^{n-m/2}$개의 메시지 중에서 충돌이 발생할 메시지는 $2^{n-m/2}$보다 작으므로 이들 메시지 중에서 내부 충돌 쌍을 찾기 위해서는 $2^{n-m/2}$개의 메시지의 끝에 고정된 n-비트 블록을 더한 새로운 메시지의 MAC 값이 필요하다. 이것은 $2^{n-m/2}$과 비교하여 무시할 수 있는 수이다.

x와 x'의 연쇄변수들에 대해

$H_1 \oplus H_2 \oplus \cdots \oplus H_t = H'_1 \oplus H'_2 \oplus \cdots \oplus H'_s$이 성립하고 $x \| y$와 $x' \| y'$이 내부 충돌 쌍이면 $H_t \oplus y = H'_s \oplus y'$이고 $\Delta = y \oplus y' = H_t \oplus H'_s$이다. 그러면 임의의 n-비트 블록 z에 대하여

$$\begin{aligned} & H_1 \oplus H_2 \oplus \cdots \oplus H_t \oplus E_K(H_t \oplus z) \\ &= H'_1 \oplus H'_2 \oplus \cdots \oplus H'_s \oplus E_K(H_t \oplus H'_s \oplus H'_s \oplus z) \\ &= H'_1 \oplus H'_2 \oplus \cdots \oplus H'_s \oplus E_K(H'_s \oplus z \oplus \Delta) \end{aligned}$$

이므로 $x \| z$와 $x' \| (z \oplus \Delta)$는 같은 MAC 값을 가진다. 그러므로 공격자가 $x \| z$의 MAC 값을 관찰하거나 요청함으로써 $x' \| (z \oplus \Delta)$의 MAC 값을 위조할 수 있다. 이

공격의 총 복잡도는 $[0,\ 2^{(n+m)/2},\ 2^{n-m/2},\ 0]$ 이다.

$n=64,\ m=32$를 사용하는 3GPP-MAC의 경우 2^{48}개의 선택 메시지/MAC 쌍과 2^{48}개의 알려진 메시지/MAC 쌍을 가진 공격자에 의해 위장 공격이 가능하다. 하지만 그들의 증명에서 구체적으로 위장 공격에 성공하기 위한 공격 수행 시나리오와 위장 공격에 성공하기 위한 실제적인 공격 복잡도를 정확하게 알 수는 없다.

3.2 위장 공격 수행 알고리즘

본 절에서는 Knudsen-Mitchell의 위장 공격법에 기반하여 실제적으로 공격을 성공하기 위해서 필요한 위장 공격 수행 알고리즘을 제안하고 그 성공 확률 및 수행 복잡도를 계산한다.

▶ 단계 1

$2^{\frac{n+m}{2}}$ 개의 알려진 메시지/MAC 쌍을 수집한다.

▶ 단계 2

수집된 $2^{\frac{n+m}{2}}$ 개의 메시지/MAC 쌍들을 MAC 값에 따라 분류할 때 처음으로 분류 된 같은 MAC 값을 가진 $2^{(n-m)/2}$개의 메시지 집합을 선택한다. ([단계 2]가 다시 반복될 때에는 그 다음으로 분류된 메시지 집합을 택한다.)

▶ 단계 3

단계 2에서 선택한 집합의 모든 메시지 $x=x_1\|x_2\|x_3\|\cdots$에 대해, $2^{n/2}$개의 랜덤하게 선택한 $y(i)$로 만든 $x\|y(i)$ 메시지들과 대응하는 MAC 값들로 이루어진 전체 $x^{(n-m)/2}\cdot 2^{n/2}=2^{n-m/2}$개의 메시지/MAC 집합을 구한다.

▶ 단계 4.

단계 3에서 선택한 $2^{n-m/2}$개의 메시지/MAC 쌍을 같은 MAC 값을 가지는 메시지 집합들로 정렬한다.

▶ 단계 5

남아있는 메시지 집합들을 $\{X_i\}$라 할 때, 고정된 n-비트 블록 y를 첨가하여 새로운 메시지 집합들 $\{X_i\|y\}$를 구하고 대응하는 $\{MAC(X_i\|y)\}$ 집합들을 조사하여 충돌이 없으면 대응하는 메시지 X_i들을 각각의 집합에서 버린다.

▶ 단계 6

단계 5를 수행한 후 원래 메시지 집합들 $\{X_i\}$에서 버려진 메시지들을 제외하고 남아있는 메시지 집합들 중 메시지의 개수가 한 개인 집합은 버리고 남아있는 메시지 집합들을 구한다.

▶ 단계 7

단계 5와 단계 6을 $\lceil \frac{n}{m} \rceil$ 번 반복한다.

▶ 단계 8

단계 7을 수행한 후 두 개의 메시지를 가진 집합이 한 개 이상 존재한다면 이 메시지 쌍들을 내부 충돌 쌍으로 출력한다. 남아있는 집합이 없다면 단계 2로 돌아간다.

▶ 단계 9

단계 8에서 구한 메시지 쌍을 $x=x_1\|\cdots\|x_t\|y$와 $x'_1\|\cdots\|x'_s\|y'$, 그리고 $\Delta=y\oplus y'$라고 할 때 임의의 n-비트 블록 z에 대하여 $x\|z$의 MAC 값을 관찰하거나 요청함으로써 $x'\|(z\oplus\Delta)$의 MAC 값을 위조한다.

이 공격 수행 알고리즘의 성공 확률과 공격 복잡도를 알아보자.

▷ 단계 1.의 알려진 메시지/MAC 쌍의 개수가 $2^{\frac{n+m}{2}}$ 인 경우

먼저 단계 2 에서 단계 7 까지의 수행을 통하여 단계 8 에서 내부 충돌 쌍을 발견할 확률을 구해보자. 먼저 단계 2 에서 버려진 $(n-m)$-비트에서도 충돌하는 메시지들이 있을 확률은 보조정리 2에 의해 $n=2^{n-m}$, $r=2^{\frac{n-m}{2}}$ 이므로

$$1-\exp\left(-\frac{r^2}{2n}\right)$$
$$=1-\exp\left(-\frac{\left(2^{\frac{n-m}{2}}\right)^2}{2\cdot 2^{n-m}}\right)$$
$$=1-\exp\left(-\frac{2^{n-m}}{2\cdot 2^{n-m}}\right)$$
$$=1-\exp\left(\frac{-1}{2}\right)$$

대략 $1-e^{-1/2}\approx 0.39$이다.

단계 3 에서 선택한 $2^{n-m/2}$개의 메시지 중에서 $H_i\oplus y=H'_s\oplus y'$을 만족하는 내부 충돌 쌍 $x=x_1\|\cdots\|x_t\|y$와 $x'_1\|\cdots\|x'_s\|y'$을 발견할 확률은 다음과 같이 계산된다. 먼저

$$H_1\oplus\cdots\oplus H_i\oplus E_K(H_i\oplus y)=$$
$$H'_1\oplus s\cdots\oplus H'_s\oplus E_K(H'_s\oplus y')$$

인 메시지 $x=x_1\|\cdots\|x_t\|y$와 $x'_1\|\cdots\|x'_s\|y'$가 존재할 확률은 보조정리 3에 의해 $n=2^{n-\frac{m}{2}}$, $r=2^{\frac{2n-m}{4}}$ 이므로

$$1-\exp\left(-\frac{r^2}{n}\right)$$
$$=1-\exp\left(-\frac{\left(2^{\frac{2n-m}{4}}\right)^2}{2^{n-\frac{m}{2}}}\right)$$
$$=1-\exp(-1)$$

대략 $1-e^{-1}\approx 0.63$ 이다.

그리고, 동시에

$$H_1\oplus H_2\oplus\cdots\oplus H_t=H'_1\oplus H'_2\oplus\cdots\oplus H'_s$$

이 성립할 확률은 $0.63\times 0.39\approx 0.25$이다.

따라서 단계 8 에서 내부 충돌 쌍을 출력하기 위해서는 단계 2 에서 단계 7 까지 평균 4번 반복 수행이 필요하다. 이제 이 공격 알고리즘의 복잡도를 계산해 보자. [단계 5, 6, 7]을 수행하는 데에 필요한 선택 메시지의 개수는 대략

$$2^{n-m/2}+2^{n-3m/2}+2^{n-5m/2}+\cdots \approx \frac{2^{n-m/2}}{1-2^{-m}}$$
$$\approx 2^{n-m/2}$$

이다.

따라서 이 공격 알고리즘을 수행하기 위해 필요한 선택 메시지의 개수는 대략

$4\cdot 2^{n-m/2+1}=2^{n-m/2+3}$이다. 따라서 총 공격 복잡도는 $[0,\ x^{(n+m)/2},\ 2^{n-m/2+3},\ 0]$이다. $n=64$, $m=32$인 경우

$[0,\ 2^{48},\ 2^{51},\ 0]$의 공격 복잡도를 가지고 항상 위장 공격에 성공할 수 있다.

▷ 단계 1 의 알려진 메시지/MAC 쌍의 개수가 $2^{\frac{n+m}{2}+1}$ 인 경우

단계 8 에서 내부 충돌 쌍을 발견하는 성공 확률을 높이기 위해서는 단계 1의 알려진 메시지/MAC 쌍의 개수를 $2^{\frac{n+m}{2}+1}$로 증가시키면 먼저 단계 2에서 버려진 $(n-m)$-비트에서도 충돌하는 메시지들이 있을 확률은 보조정리 2에 의해

$n=2^{n-m}$, $r=2^{\frac{n-m}{2}+1}$ 이므로

$$1-\exp\left(-\frac{r^2}{2n}\right)$$
$$=1-\exp\left(-\frac{\left(2^{\frac{n-m}{2}+1}\right)^2}{2\cdot 2^{n-m}}\right)$$
$$=1-\exp\left(-\frac{2^{n-m+2}}{2\cdot 2^{n-m}}\right)$$
$$=1-\exp(-2)$$

대략 $1-e^{-2}\approx 0.86$이다.

단계 3 에서 선택한 $2^{n-m/2}$개의 메시지 중에서 $H_i\oplus y=H'_s\oplus y'$을 만족하는 내부 충돌 쌍 $x=x_1\|\cdots\|x_t\|y$와 $x'_1\|\cdots\|x'_s\|y'$을 발견할 확률은 다음과 같이 계산된다. 먼저

$H_1 \oplus \cdots \oplus H_t \oplus E_K(H_t \oplus y) = H'_1 \oplus s \cdots \oplus H'_s \oplus E_K(H'_s \oplus y')$인 메시지 $x = x_1 \| \cdots \| x_t \| y$와 $x'_1 \| \cdots \| x'_s \| y'$가 존재할 확률은 보조정리 3에 의해 대략 $1 - e^{-1} \approx 0.63$이고 동시에 $H_1 \oplus H_2 \oplus \cdots \oplus H_t = H'_1 \oplus H'_2 \oplus \cdots \oplus H'_s$ 이 성립할 확률은 $0.63 \times 0.86 \approx 0.54$이다. 따라서 단계 8 에서 내부 충돌 쌍을 출력하기 위해서는 단계 2에서 단계 7까지 평균 2번 반복 수행이 필요하다.

이제 이 공격 알고리즘의 복잡도를 계산해 보자. 단계 5, 6, 7을 수행하는 데에 필요한 선택 메시지의 개수는 대략

$$2^{n-m/2} + 2^{n-3m/2} + 2^{n-5m/2} + \cdots \approx \frac{2^{n-m/2}}{1-2^{-m}} \approx 2^{n-m/2}$$

이다.

따라서 이 공격 알고리즘을 수행하기 위해 필요한 선택 메시지의 개수는 대략 $2 \cdot 2^{n-m/2+1} = 2^{n-m/2+2}$이다. 따라서 총공격 복잡도는 $[0,\ 2^{(n+m)/2+1},\ 2^{n-m/2+2},\ 0]$이 다 . $n = 64,\ m = 32$인 경우 $[0,\ 2^{49},\ 2^{50},\ 0]$의 공격 복잡도를 가지고 항상 위장 공격에 성공할 수 있다.

따라서 단계 8 에서 내부 충돌쌍이 있을 성공 확률은 0.54로 증가하고 전체 선택 메시지/MAC 쌍의 개수는 2^{50}으로 감소한다. 하지만 증가시킬 수 있는 성공 확률이 0.63으로 제한되므로 알려진 메시지/MAC 쌍의 개수는 제한적이다.

위의 알고리즘의 일반적인 성공 확률과 공격 수행 복잡도의 관계는 〈표 1〉과 같다.

표 1. 성공 확률과 알려진 메시지/MAC, 선택 메시지/MAC 쌍의 관계(메시지 길이: n, MAC값 길이: m 인 경우)

Table1. Relation Sucess Probability, Known Plaintext MAC, and Choosen Plaintext MAC(in case of message length: m,MAC length: m)

복잡도 / 성공 확률	알려진 메시지/MAC 쌍	선택 메시지/MAC쌍	공격수행 복잡도
0.02	$2^{\frac{n+m}{2}-2}$	$1.6\times2^{n-\frac{m}{2}+6}$	$(0, 2^{\frac{n+m}{2}-2}, 1.6\times2^{n-\frac{m}{2}+6}, 0)$
0.12	$2^{\frac{n+m}{2}-1}$	$1.7\times2^{n-\frac{m}{2}+4}$	$(0, 2^{\frac{n+m}{2}-1}, 1.7\times2^{n-\frac{m}{2}+4}, 0)$
0.25	$2^{\frac{n+m}{2}}$	$2^{n-\frac{m}{2}+3}$	$(0, 2^{\frac{n+m}{2}}, 2^{n-\frac{m}{2}+3}, 0)$
0.54	$2^{\frac{n+m}{2}+1}$	$2^{n-\frac{m}{2}+2}$	$(0, 2^{\frac{n+m}{2}+1}, 2^{n-\frac{m}{2}+2}, 0)$
0.63	$2^{\frac{n+m}{2}+2}$	$1.6\times2^{n-\frac{m}{2}+1}$	$(0, 2^{\frac{n+m}{2}+2}, 1.6\times2^{n-\frac{m}{2}+1}, 0)$
0.63	$2^{\frac{n+m}{2}+3}$	$1.6\times2^{n-\frac{m}{2}+1}$	$(0, 2^{\frac{n+m}{2}+3}, 1.6\times2^{n-\frac{m}{2}+1}, 0)$

다음은 메시지 길이와 MAC값의 길이를 비교한 표이다. 다음 〈표 2〉는 기본 블록 암호로 DES를 사용하고, $n = 64, m = 32$ 인 경우의 공격수행 복잡도를 계산한 표이다.

표 2. 성공 확률과 알려진 메시지/MAC, 선택 메시지/MAC 쌍의 관계(DES, $n = 64,\ m = 32$경우)

Table2. Relation Sucess Probability, Known Plaintext MAC, and Choosen Plaintext MAC(in case of DES, $n = 64,\ m = 32$)

복잡도 / 성공 확률	알려진 메시지/MAC 쌍	선택 메시지/MAC쌍	공격수행 복잡도
0.02	2^{46}	1.6×2^{54}	$(0, 2^{46}, 1.6\times2^{54}, 0)$
0.12	2^{47}	1.7×2^{51}	$(0, 2^{47}, 1.7\times2^{51}, 0)$
0.25	2^{48}	2^{51}	$(0, 2^{48}, 2^{51}, 0)$
0.54	2^{49}	2^{50}	$(0, 2^{49}, 2^{50}, 0)$
0.63	2^{50}	1.6×2^{49}	$(0, 2^{50}, , 0)\ 1.6\times2^{49}$
0.63	2^{51}	1.6×2^{49}	$(0, 2^{51}, 1.6\times2^{49}, 0)$

다음 〈표 3〉는 기본 블록 암호로 DES를 사용하고, $n = 64, m = 64$ 인 경우의 공격수행 복잡도를 계산한 표이다.

표 3. 성공 확률과 알려진 메시지/MAC, 선택 메시지/MAC 쌍의 관계(DES, $n=64$, $m=64$경우)
Table3. Relation Sucess Probability, Known Plaintext MAC, and Choosen Plaintext MAC(in case of DES, $n=64$, $m=64$)

복잡도 / 성공 확률	알려진 메시지/MAC 쌍	선택 메시지/MAC쌍	공격수행 복잡도
0.02	2^{62}	1.6×2^{38}	$(0, 2^{62}, 1.6\times2^{38}, 0)$
0.12	2^{63}	1.7×2^{36}	$(0, 2^{63}, 1.7\times2^{36}, 0)$
0.25	2^{64}	2^{35}	$(0, 2^{64}, 2^{35}, 0)$
0.54	2^{65}	2^{34}	$(0, 2^{65}, 2^{34}, 0)$
0.63	2^{66}	1.6×2^{33}	$(0, 2^{66}, 1.6\times2^{33}, 0)$
0.63	2^{67}	1.6×2^{33}	$(0, 2^{67}, 1.6\times2^{33}, 0)$

다음 〈표 4〉는 기본 블록 암호로 SEED를 사용하고, $n=128, m=64$ 인 경우의 공격수행 복잡도를 계산한 표이다.

표 4. 성공 확률과 알려진 메시지/MAC, 선택 메시지/MAC 쌍의 관계(SEED, $n=128$, $m=64$경우)
Table4. Relation Sucess Probability, Known Plaintext MAC, and Choosen Plaintext MAC(in case of DES, $n=128$, $m=64$)

복잡도 / 성공 확률	알려진 메시지/MAC 쌍	선택 메시지/MAC쌍	공격수행 복잡도
0.02	2^{94}	1.6×2^{102}	$(0, 2^{94}, 1.6\times2^{102}, 0)$
0.12	2^{95}	1.7×2^{100}	$(0, 2^{95}, 1.7\times2^{100}, 0)$
0.25	2^{96}	2^{99}	$(0, 2^{96}, 2^{99}, 0)$
0.54	2^{97}	2^{98}	$(0, 2^{97}, 2^{98}, 0)$
0.63	2^{98}	1.6×2^{97}	$(0, 2^{98}, 1.6\times2^{97}, 0)$
0.63	2^{99}	1.6×2^{97}	$(0, 2^{99}, 1.6\times2^{97}, 0)$

다음 〈표 5〉는 기본 블록 암호로 SEED를 사용하고, $n=128, m=128$ 인 경우의 공격수행 복잡도를 계산한 표이다.

표 5. 성공 확률과 알려진 메시지/MAC, 선택 메시지/MAC 쌍의 관계(SEED, $n=128$, $m=128$경우)
Table5. Relation Sucess Probability, Known Plaintext MAC, and Choosen Plaintext MAC(in case of SEED, $n=128$, $m=128$)

복잡도 / 성공 확률	알려진 메시지/MAC 쌍	선택 메시지/MAC쌍	공격수행 복잡도
0.02	2^{126}	1.6×2^{70}	$(0, 2^{126}, 1.6\times2^{70}, 0)$
0.12	2^{127}	1.7×2^{68}	$(0, 2^{127}, 1.7\times2^{68}, 0)$
0.25	2^{128}	2^{67}	$(0, 2^{128}, 2^{67}, 0)$
0.54	2^{129}	2^{66}	$(0, 2^{129}, 2^{66}, 0)$
0.63	2^{130}	1.6×2^{65}	$(0, 2^{130}, 1.6\times2^{65}, 0)$
0.63	2^{131}	1.6×2^{65}	$(0, 2^{131}, 1.6\times2^{65}, 0)$

최종 출력 변환의 입력 값으로 모든 연쇄 변수를 XOR하여 사용함으로써 필요한 공격량을 증가시킬 수 있으며, 메시지의 길이가 증가하면 알려진 메시지/MAC쌍과 선택 메시지/MAC쌍이 모두 증가하므로 더 안전한 알고리즘이 되므로 블록 알고리즘을 SEED로 사용하면 MAC 알고리즘이 안전해 진다. 그리고, MAC의 값의 길이가 증가하면 알려진 메시지/MAC쌍은 증가하고 선택 메시지/MAC쌍은 감소하는 것을 알 수 있다. 따라서 m 이 작을수록 위장 공격의 복잡도는 증가하지만 추측 공격을 고려하여 $m=n/2$ 을 사용하고 있으며, 이 경우 현실적인 공격은 불가능하다.

Ⅳ. 결 론

MAC 알고리즘의 설계 및 안전성 분석은 수학적 이론이 뒷받침되어야 하는 광범위한 연구 분야이다. 또한 MAC 알고리즘은 현대 암호학에서 데이터 무결성과 인증을 제공하기 위한 도구로 매우 빈번하게 사용된다. 따라서 MAC 알고리즘은 안전성뿐 아니라 효율성 또한 고려하여야 하고 안전성과 관련하여서는 암호분석 능력을 예측하여야 한다.

본 논문에서는 기본 블록 암호 알고리즘으로 64 비트 블록 암호 알고리즘(DES)과 128비트 블록 암호 알고리즘(SEED)을 사용하는 MAC 알고리즘을 제안하고, 생일 공격과 내부 충돌 쌍을 이용한 Knudsen-Mitchell의 공격법을 심층 분석하여 구체적인 공격 수행 알고리즘을 제안하고,

이 알고리즘의 성공 확률 및 수행 복잡도를 계산하였다. 그리고, 메시지의 길이와 MAC값의 길이에 따른 위장공격의 안전성을 분석하였다. 메시지의 길이가 증가하면 알려진 메시지/MAC쌍과 선택 메시지/MAC쌍이 모두 증가하므로 더 안전한 알고리즘이 되므로 블록 알고리즘을 SEED로 사용하면 MAC 알고리즘이 안전해진다는 것을 알 수 있었다.

참고문헌

[1] Bart Preneel, Analysis and Design of Cryptographic Hash Functions, Katholieke Universiteit Leuven, Ph.D. Thesis, 1993.

[2] ISO/IEC 9797-1, Information technology - Security techniques-Messate Authenti cation Codes(MACs) -Part 1: Mechanism using a block cipher, 1999.

[3] B.Preneel and P.V van Oorschot, "On the security of iterated Message Authentication codes", IEEE Transactions on Information Theory, 45, pp. 71~82, 1999.

[4] L.R. Knudsen, "Chosen-test attack on CBC-MAC", Electronics Letters, 33, pp. 48~49, 1997.

[5] K. Brincat and C.J. Mitchell. "Mew CBC-MAC forgery attacks, preprint.

[6] American Bankers Association, Washington, DC. ANSI X9.19, "Financial institution retail message authenticaion", August 1986.

[7] B.Preneel and P.V. van Oorschot, "A key recovery attack on the ANSI X9.19 retail MAC", Electronics Letters. 32(17), pp.1568~1569, 1996.

[8] L.R. Knudsen and B. Preneel, "Mac-DES: MAC algorithm based on DES", Electronics Letters 34, pp. 871~873, 1998.

[9] L.R. Knudsen and C.J. Mitchell. "An analysis of the 3GPP-MAC scheme", WCC 2001, pp.319~328, January 2001.

[10] E. petrank and C. Rackoff, "CBC-MAC for Real -Time Data Sources", J. Cryptoloty, pp.315~338, 2000.

저 자 소 개

서 창 호

1990년 고려대학교 수학과(학사)
1992년 고려대학교 일반대학원 수학과 (이학석사)
1996년 고려대학교 일반대학원 수학과 (이학박사)
1996년~1996년 국방과학연구소 선임연구원
1996년~2000년 한국전자통신연구원 선임연구원, 팀장
2000년~현재 공주대학교 응용수학과(정보보호전공) 부교수
2001년~현재 공주대학교 바이오정보학과 부교수
〈관심분야〉 암호 알고리즘, PKI, 무선 인터넷 보안, 시스템 보안 등

윤 보 현

1999년 고려대학교 컴퓨터학과 (이학박사).
1999년~2003년 한국전자통신연구원 (ETRI) 선임연구원 및 팀장
2001년~2003년 한국소프트웨어산업협회 언어정보산업협의회 운영위원
2002년~2002년 광인터넷 기술정책 자문위원
2003년~2004년 한국전자통신연구원(ETRI) 초빙연구원
2003년~현재 목원대학교 컴퓨터교육과 교수
〈관심분야〉 정보검색, 컨텐츠보호, 시맨틱웹, 바이오인포메틱 등

맹 승 렬

2004년 한국과학기술원 컴퓨터학과 (공학박사).
1984년~1994년 한국표준과학연구원 선임연구원
1997년~1998년 미국 조오지 워싱턴대학 객원연구원
2004년~2005년 호주그리피스 대학 객원연구원
〈관심분야〉 컴퓨터 그래픽스, 유비쿼터스 컴퓨팅