情報保護學會論文誌
第14券 第4號, 2004. 8

PMAC과 TMAC 변이 알고리즘에 대한 안전성 고찰

이 창 훈†, 김 종 성, 이 상 진‡
고려대학교 정보보호기술연구센터

A Security Analysis of PMAC and TMAC variant

Changhoon Lee†, Jongsung Kim, Sangjin Lee‡
Center for Information Security Technologies, Korea University

요 약

본 고에서는 J. Black과 P. Rogaway이 설계한 PMAC과 C. J. Mitchell이 설계한 CBC-MAC 변이 알고리즘인 TMAC*에 대해, 위조 공격(forgery attack) 관점에서 안전성을 고찰한다. Truncation을 사용하는 PMAC 경우에는 약 $2^{n/2+1}$개 선택 평문과 약 $2^{n-\tau}$번 MAC 검증 과정으로 위조 공격이 가능하고 Truncation을 사용하지 않는 경우에는 약 $2^{n/2+1}$개 선택 평문만으로 위조 공격이 가능하다. 또한, TMAC*의 경우에도 약 $2^{n/2+1}$개 선택 평문만 획득하면 위조 공격을 성공할 수 있다.

ABSTRACT

In this paper, we introduce two forgery attacks on the PMAC. If it has no truncation then the attack requires about $2^{n/2+1}$ chosen texts, otherwise, the attack requires about $2^{n/2+1}$ chosen texts and $2^{n-\tau}$ MAC verifications where τ is the size of the MAC. We also give a forgery attack on the TMAC variant which requires about $2^{n/2+1}$ texts.

Keywords : *PMAC, TMAC, Forgery Attack*

Ⅰ. 서 론

MAC(Message Authentication Codes : 메시지 인증 코드)은 키를 사용하여 임의 길이의 메시지를 고정된 길이로 압축하는 함수(Keyed hash function)로 정의할 수 있다. MAC을 사용하는 목적은 메시지에 대한 무결성을 보장하고, 메시지 출처 인증을 하기 위해서이다. 여기서, 메시지에 대한 무결성은 메시지의 변조 여부를 알 수 있다는 것을 의미하고, 메시지 출처 인증은 메시지를 보낸 사람에 대한 확인이 가능하다는 것을 말한다.

MAC의 안전성을 평가하는 방법은 크게 Practical 측면에서 MAC을 분석하는 방법과 이론적인 MAC의 안전성을 규명하는 방법(Provable Security)으로 분류할 수 있다. Practical Attack은 위조 공격(forgery attack)과 키 복구 공격(key recovery attack)을 일컫는데, 전자는 비밀키를 모르는 상태에서 기지 또는 선택 메시지와 그에 대응하는 MAC 쌍으로부터 새로운 메시지와 그에 대한 MAC 쌍을 찾는 공격이고, 후자는 MAC값을 생성하는데 사용되는 비밀키를 찾는 공격법이다. 본 고에서는 두 공격법 중 MAC에 대한 위조 공격에 초점을 둔다.

접수일 : 2004년 4월 8일 ; 채택일 : 2004년 5월 27일
* 본 연구는 정보통신부 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었습니다.
† 주저자 : crypto77@cist.korea.ac.kr
‡ 교신저자 : sangjin@korea.ac.kr

PMAC은 Eurocrypt 2002에서 P. Rogaway에 의해 제안된 MAC으로서, 하나의 키를 사용하고 일반적인 CBC-MAC 종류의 다른 MAC들과 달리, 병렬로 처리할 수 있다는 장점이 있다. 이와 더불어, PMAC은 함께 사용되는 블록 암호가 안전할 때 PMAC이 안전함이 증명되었다. [8]

CT-RSA 2003에서 소개된 TMAC은 k-비트인 키와 n-비트인 키 두 개를 사용하는 알고리즘으로서 Kurosawa와 Iwata에 의해 설계되었다[4]. 이 알고리즘은 세 개의 다른 키 (K_1, K_2, K_3)를 사용하는 XCBC-MAC[3]을 두개의 키 $(K, K' \cdot u, K')$를 사용하도록 변형한 것이다. 그런데, 이것은 키 사이에 대수적인 문제점이 있어서 키 복구 및 위조 공격에 취약점이 있음이 밝혀졌다. 그래서 C. J. Mitchell은 이러한 대수적인 문제점을 해결하고자 키 $(K, E_K(S_2), E_K(S_3))$를 사용하는 TMAC을 변형한 새로운 알고리즘을 소개하였다[7]. 편의상 이 알고리즘을 TMAC*라 부르기로 하겠다. 본 고에서는 PMAC과 TMAC*에 대한 안전성을 위조 공격 관점에서 고찰해 본다. 2절에서는 PMAC 알고리즘을 소개하고 truncation 유무에 따른 위조 공격을 수행하고, 3절에서는 TMAC* 알고리즘을 소개하고 이것에 대한 위조 공격을 수행한다. 마지막 4절에서는 본 고의 결과를 요약한다.

II. PMAC에 대한 위조 공격

본 절에서는 PMAC 알고리즘을 설명하고 이 MAC에 대한 위조 공격을 제시한다.

1. PMAC 알고리즘 소개

PMAC 알고리즘 소개하기 전에 본 논문에서 사용하는 표기들을 소개한다.

- E : n-비트 블록 암호
- K : k-비트 키
- M : MAC의 입력 메시지
- $M[i]$: i-번째 n-비트 입력 메시지 블록
- $Y[i]$: i-번째 E의 n-비트 출력 블록
- T : τ-비트 MAC 값
- $[a, b, c, d]$: 공격 복잡도를 나타내는 표기로서, a는 오프라인 상에서 블록 암호 암호화 수를, b는 기지 메시지/MAC쌍의 수를, c는 선택 메시지/MAC쌍의 수를, d는 온라인 MAC verification 수를 의미.

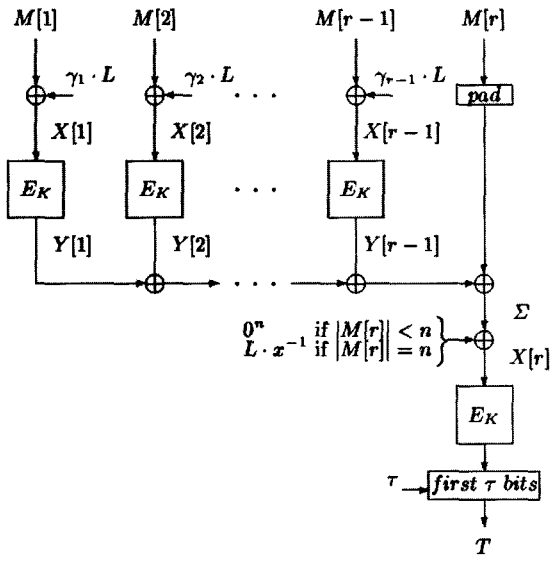


그림 1. PMAC 알고리즘

그러면, PMAC의 연동 과정(그림 1)을 다음과 같이 정의할 수 있다.

1. 입력 메시지 M을 n-비트 단위 블록들 $(M[i])$로 나눈다. 나눠진 블록의 개수를 r이라 가정할 때, 마지막 블록 $M[r]$의 크기가 n-비트가 아니라면 마지막 블록에 10...0으로 패딩한다.
2. $i=1, \ldots, r-1$에 대해, $Y[i]=E_K(M[i] \oplus \gamma_i \cdot L)$를 계산한다.
3. ㉠ $|M[r]|=n$이라면, $temp=Y[1] \oplus \cdots \oplus Y[r-1] \oplus M[r]$를 계산한 다음 $T1=E_K(temp \oplus L \cdot x^{-1})$을 계산하고, $T1$의 상위 τ-비트 truncation을 수행한 후에 MAC값 T를 출력한다.
   ㉡ $|M[r]| \neq n$라면, $M[r]$을 패딩한 후에 $(W=pad(M[r]))$, $temp=Y[1] \oplus \oplus W \cdots \oplus Y[r-1]$를 연산한 다음 $T1=E_K(temp \oplus 0^n)$을 계산하고, $T1$의 상위 τ-비트 truncation을 수행한 후에 MAC값 T를 출력한다.

2. PMAC에 대한 위조 공격

본 소절에서는 PMAC이 Truncation을 사용하는 경우와 그렇지 않는 경우를 고려하여 위조 공격을 수행한다.

2.1. Truncation을 사용하지 않는 경우

먼저, 공격자는 (q+1)개 블록을 가지는 약 $2^{n/2}$개의 다른 메시지들 $M(1)^i=(M[1],\ldots,M[q],X^i)$에 대한 MAC값들을 얻는다. 여기서, $(M[1],\ldots,M[q])$는 임의의 고정된 블록들이고 X^i는 크기가 n비트 보다 작은 임의 블록이다 $(1\le i\le 2^{n/2})$. 그런 다음, 공격자는 (q+1)개 블록을 가지는 약 $2^{n/2}$개의 다른 메시지들 $M(2)^j=(M[1],\ldots,M[q],Z^j)$에 대한 MAC값들을 더 얻는다. 여기서, $(M[1],\ldots,M[q])$는 임의의 고정된 블록들이고 Z^j는 크기가 n비트인 임의 블록이다 $(1\le j\le 2^{n/2})$.

그러면, 생일 역설(birthday paradox)에 의해, 공격자는 높은 확률로 하나의 내부 충돌쌍을 기대할 수 있다. 즉, 두 집합 $M(1)^i$과 $M(2)^j$ 사이에 높은 확률로, 같은 MAC값을 출력하는 메시지쌍이 존재한다. 이러한 메시지 충돌쌍을 각각 다음과 같다고 한다면,

$$M(1)^*=(M[1],\ldots,M[q],X^*)$$
$$M(2)^*=(M[1],\ldots,M[q],Z^*)$$

두 메시지의 $(M[1],\ldots,M[q])$는 동일한 메시지 블록이므로 두 메시지의 $\sum_{i=0}^{q}(Y[i])$값들도 같다(그림 1 참고). 따라서, 공격자는 다음 방정식을 얻을 수 있다.

$$E_K((X^*\|padding)\oplus\sum_{i=0}^{q}(Y[i])\oplus 0^n)$$
$$= E_K(Z^*\oplus\sum_{i=0}^{q}(Y[i])\oplus L\cdot x^{-1})$$

그런데, $E_K(\cdot)$는 치환 함수이므로, $L\cdot x^{-1}=(X^*\|padding)\oplus Z^*$이다. 이 사실로부터 공격자는 키 정보 $L=((X^*\|padding)\oplus Z^*)\cdot x$을 추출해 낼 수 있고, 다음처럼 새로운 메시지에 대한 위조 공격도 수행할 수 있다.

① 공격자는 첫번째 메시지들 집합 $M(1)^i$에서 충돌쌍이 아닌 다음과 같은 하나의 메시지를 선택하고, $M(1)^k=(M[1],\ldots,M[q],X^k)$, 아래 식과 같은 대응하는 MAC값을 얻는다.

$$E_K((X^k\|padding)\oplus\sum_{i=0}^{q}(Y[i])\oplus 0^n)$$

② 그러면, 공격자는 메시지 $(M[1],\ldots,M[q],(X^k\|padding\oplus L\cdot x^{-1}))$에 대한 MAC값이 $M(1)^k$에 대한 MAC값과 같음을 안다.

그러므로, 공격자는 약 $2^{n/2+1}$개의 선택 평문/암호문 쌍으로 위조 공격을 성공할 수 있다. 이 공격 복잡도는 $[0,0,2^{n/2+1},0]$이다.

2.2. Truncation을 사용하는 경우

이 공격은 Truncation을 사용하지 않을 때의 공격과 시작은 동일하다. 즉, 공격자는 약 $2^{n/2}$개의 다른 메시지들 $M(1)^i$와 $M(2)^j$ 대한 MAC값들을 각각 얻는다. 그러면, 생일 역설에 의해, 공격자는 높은 확률로 약 $2^{n-\tau}$개의 외부 충돌쌍을 기대할 수 있다. 이 외부 충돌을 발생시키는 각각의 메시지들을 다음과 같다고 하자.

$$(M[1],\ldots,M[q],X^{*(j)}),\quad (M[1],\ldots,M[q],Z^{*(j)}),\quad (1\le j\le 2^{n-\tau}).$$

그런데, 공격자가 위조 공격을 성공하기 위해서는 이 많은 외부 충돌쌍들로부터 내부 충돌쌍을 찾아야만 한다. 만약에 공격자가 내부 충돌쌍 $M[1],\ldots,M[q],X^{*(k)}$와 $M[1],\ldots,M[q],Z^{*(k)}(k\in j)$을 찾았다면, 2.1절의 경우처럼 아래 방정식을 얻을 수 있다.

$$E_K((X^{*(k)}\|padding)\oplus\sum_{i=0}^{q}(Y[i])\oplus 0^n)$$
$$= E_K(Z^{*(k)}\oplus\sum_{i=0}^{q}(Y[i])\oplus L\cdot x^{-1})$$

여기서 $E_K(\cdot)$가 치환 함수이므로, $L \cdot x^{-1} = (X*^{(k)} \| padding) \oplus Z*^{(k)}$이다. 공격자는 이 정보를 이용하여 2.1절과 유사한 방법으로 위조 공격을 할 수 있다.

① 공격자는 내부 충돌쌍(k)을 찾기 위해 다음 과정을 수행한다.

㉠ 공격자는 $L \cdot x^{-1}$의 후보들 $L \cdot x^{-1}(j) = X*^{(j)} \| padding \oplus Z*^{(j)}$, $(1 \le j \le 2^{n-\tau})$를 계산한다.

㉡ 공격자는 첫 번째 메시지 집합 $M(1)^i$에서 $(M[1], \ldots, M[q], X^s)$를 선택하고 대응하는 MAC값을 얻는다.

㉢ 공격자는 $2^{n-\tau}$개 메시지 $(M[1], \ldots, M[q], (X^s \| padding) \oplus L \cdot x^{-1}(j))$에 대한 MAC값을 얻는다.

㉣ 마지막으로, 공격자는 ㉢내의 $2^{n-\tau}$개 메시지/MAC 쌍 중에 ㉡에서 선택한 메시지에 대한 MAC값과 같은 MAC값을 갖는 메시지가 존재하는지 검사한다. 만약 ㉢내의 어떤 메시지가 이 테스트를 통과한다면, 그 때의 $L \cdot x^{-1}(j)$는 $L \cdot x^{-1}$의 후보가 된다. 공격자는 이 테스트를 통해 $2^{n-\tau}$개 내부 충돌쌍 후보들을 $2^{n-2\tau}$개 후보들로 줄일 수 있다. 따라서, 공격자는 이 과정을 $\lceil n/\tau \rceil$ 번 반복하면, 바라던 내부 충돌쌍을 찾을 수 있고 $L \cdot x^{-1}$값을 얻을 수 있다.

공격자는 획득한 $L \cdot x^{-1}$값을 이용하여 2.1절과 같은 방법으로 위조 공격을 수행 할 수 있다. 이 공격은 약 $2^{n/2+1}$개의 선택 메시지와 약 $2^{n-\tau} + 2^{n-2\tau} + \ldots + 2^{n-\lceil n/\tau \rceil \tau} \approx 2^{n-\tau}$ 번의 MAC verification 과정을 필요로 한다.

따라서, 이 공격의 복잡도는 $[0, 0, 2^{n/2+1}, 2^{n-\tau}]$이다.

Ⅲ. TMAC 변형 알고리즘에 대한 위조 공격

본 절에서는 TMAC을 변형한 알고리즘 TMAC*을 소개하고 이 MAC에 대한 위조 공격을 수행한다.

1. TMAC* 알고리즘 소개

CBC-MAC은 $CBC\text{-}MAC_K(M) = H_r$로 정의할 수 있다. 여기서, 메시지 $M = M_0 \| \ldots \| M_r$이고 $H_0 = 0$, $H_i = E_K(H_{i-1} \oplus M[i])$, $(1 \le i \le r)$이다. 그러면, TMAC*를 다음과 같이 정의 할 수 있다 (그림 2).

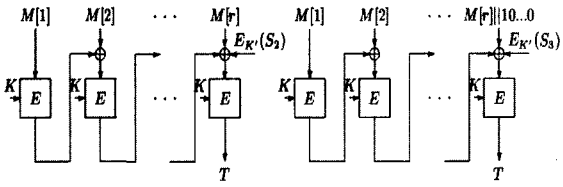


그림 2. TMAC* 알고리즘

① 메시지 M의 크기($|M|$)가 블록 암호 E의 입력 크기 n의 배수이면, MAC값은 다음과 같다.

$$TMAC_{K,K'} = E_K(H_{r-1} \oplus M[r] \oplus E_{K'}(S_2))$$

여기서, K'는 K와 다른 k-비트 키이고 S_2는 고정된 n-비트 상수이다.

② 메시지 M의 크기($|M|$)가 블록 암호 E의 입력 크기 n의 배수가 아니라면, MAC값은 다음과 같다.

$$TMAC_{K,K'} = E_K(H_{r-1} \oplus (M[r] \| padding) \oplus E_{K'}(S_3))$$

여기서, S_3은 S_2와 다른 고정된 n-비트 상수이다.

2. TMAC*에 대한 위조 공격

본 소절에서는 선택 평문을 이용하는 TMAC*에 대한 위조 공격을 보인다. 공격자는 먼저 1개 블록을 가지는 약 $2^{n/2}$개의 다른 메시지들 $M(1)^i = X^i$에 대한 MAC값들을 얻는다. 여기서, X^i는 크기가 n비트 보다 작은 임의 블록이라고 하면, 이 메시지들은 패딩 과정이 필요하다($1 \le i \le 2^{n/2}$). 그런 다음, 공격자는 1개 블록을 가지는 약 $2^{n/2}$개의 다른 메

시지들 $M(2)^j = Z^j$에 대한 MAC값들을 더 얻는다. 여기서, Z^j는 크기가 n비트인 임의 블록 이다 ($1 \le j \le 2^{n/2}$). 그러면, 생일 역설에 의해, 공격자는 높은 확률로 하나의 내부 충돌쌍을 기대할 수 있다. 이 충돌쌍을 각각 $M(1)^* = X^*$와 $M(2)^* = Z^*$라고 하면, TMAC*의 정의에 의해, 다음 방정식을 얻을 수 있다.

$$E_K((X^* \| padding) \oplus E_K(S_3)) = E_K(Z^* \oplus E_K(S_2))$$

그런데, $E_K(\cdot)$는 치환 함수이므로, 공격자는 $(X^* \| padding) \oplus Z^* = E_K(S_2) \oplus E_K(S_3)$임을 알고, 이 사실을 이용하여 다음과 같은 위조 공격을 수행할 수 있다.

① 공격자는 첫 번째 메시지들 집합 $M(1)^i$에서 충돌쌍이 아닌 한 메시지 $M(1)^k = X^k$를 선택하고, 대응하는 MAC값 $E_K((X^k \| padding) \oplus E_K(S_3))$을 얻는다.

⑤ 그러면 공격자는 TMAC*의 정의에 의해 $(X^k \| padding) \oplus (X^* \| padding) \oplus Z^*$에대한 MAC값이 메시지 X^k에 대한 MAC값과 같음을 공격자는 알 수 있다.

그러므로, 이 공격은 약 $2^{n/2+1}$개 선택 평문만으로 새로운 메시지에 대한 위조를 성공할 수 있다. 즉, 이 공격의 복잡도는 $[0, 0, 2^{n/2+1}, 0]$이다.

Ⅳ. 결 론

현재 PMAC과 TMAC은 많은 관심을 받고 있는 알고리즘들이다. PMAC은 일반적인 CBC-MAC과 달리 병렬 연산이 가능하도록 설계하여 효율성을 극대화한 알고리즘이고 TMAC은 서로 다른 세 개의 키를 사용하는 XCBC-MAC을 개선한 알고리즘이다. 본 고에서는 PMAC과 TMAC을 변형한 알고리즘(TMAC*)에 대한 위조 공격을 보임으로써 이 알고리즘에 대한 안전성을 살펴보았다. Truncation을 사용하지 않는 PMAC은 약 $2^{n/2+1}$개 선택 평문을 이용하면 새로운 메시지의 MAC을 위조할 수 있고 Truncation을 사용하는 PMAC의 경우에는 약 $2^{n/2+1}$개 선택 평문과 약 2^{n-r}번의 MAC verification 과정을 사용하면 위조 공격을 수행할 수 있다. 또한, TMAC*의 경우에도 약 $2^{n/2+1}$개 선택 평문만 있으면 위조 공격이 가능함을 보였다.

참 고 문 헌

[1] ISO/IEC 9797-1, Information technology-Security techniques-Message Authentication Codes 35:1626-1627, 1999

[2] M. Bellare, J. Kilian, and P. Rogaway, "The Security of the Cipher Block Chaining Message Authentication Code", Advanced in Cryptology-CRYPTO'94, LNCS 839, pp. 341-358, Springer-Verlag, 1994.

[3] J. Black and P. Rogaway. "CBC MACs for arbitrary-length messages: The three key construction", Advances in Cryptology - Crypto 2000, LNCS 1880, pp.197-215, Springer-Verlag, 2000.

[4] K. Kurosawa and T. Iwata, "TMAC : Two-Key CBC-MAC", Topics in Cryptology-CT-RSA 2003, LNCS 2612, pp. 33-49, Springer-Verlag, 2003

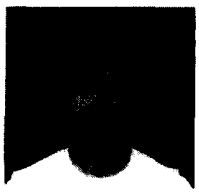
[5] J. Sung, D. Hong, and S. Lee, "Key Recovery Attacks on the RMAC, TMAC, and IACBC", ACISP 2003, LNCS 2727, pp 265-273, Springer-Verlag, 2003.

[6] T. Iwata and K. Kurosawa, "OMAC : One-Key CBC MAC, FSE 2003, LNCS 2887, pp 137-162, Springer-Verlag, 2003.

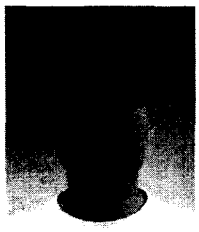
[7] C. J. Mitchell, "On the Security of XCBC, TMAC, and OMAC", Technical Report RHUL-MA-2003-4, 19, August, 2003.

[8] J. Black and P. Rogaway. "A Block Cipher Mode of Operation for Parallelizable Message Authentication" Advances in Cryptology-Eurocrypt 2002, LNCS 2332, pp.384-397, Springer-Verlag, 2002.

〈著者紹介〉



이 창 훈 (Changhoon Lee)
2001년 2월 : 한양대학교 수학과 학사
2003년 2월 : 고려대학교 정보보호대학원 석사
2003년 3월~현재 : 고려대학교 정보보호대학원 박사과정
〈관심분야〉 블록 암호, 스트림 암호, 운영모드, MAC 알고리즘 설계 및 분석



김 종 성 (Jongsung Kim)
2000년 8월 : 고려대학교 수학과 학사
2002년 8월 : 고려대학교 수학과 석사
2002년 8월~현재 : 고려대학교 정보보호대학원 박사 과정
〈관심분야〉 블록 암호, 스트림 암호, 운영모드, 해쉬함수, MAC 알고리즘 설계 및 분석

이 상 진 (Sangjin Lee) 정회원
1987년 2월 : 고려대학교 수학과 학사
1989년 2월 : 고려대학교 수학과 석사
1994년 2월 : 고려대학교 수학과 박사
1989년 2월~1999년 2월 : 한국전자통신연구원 선임 연구원,
1999년 2월~2001년 8월 : 고려대학교 자연과학대학 조교수,
2001년 9월~현재 : 고려대학교 정보보호대학원 부교수
〈관심분야〉 대칭키 암호의 분석 및 설계, 정보은닉이론, 컴퓨터 포렌식