화이트박스 암호를 이용한 콘텐츠 보호 방법

이윤경* · 김신효* · 문혜란* · 정병호*

*한국전자통신연구원

Contents Protection Method usign White Box Cryptography

Yun-kyung Lee* · Sin-hyo Kim* · Hyeran Mun* · Byung-ho Chung*

*Electronics and Telecommunications Research Institute

E-mail : neohappy@etri.re.kr

요 약

2002년 S.Chow는 AES 암호 알고리즘을 화이트박스 공격에 강하게 구현하는 방법으로 화이트박스 AES(이하 WBC-AES) 암호를 제안하였다. 본 논문에서는 S.Chow가 제안한 WBC-AES에 대한 설명과 함께 이를 이용하여 콘텐츠를 보호하기 위한 방법에 관하여 기술하고자 한다.

ABSTRACT

S. Chow proposes white-box cryptography mechanism of AES algorithm(WBC-AES) in 2002. WBC mechanism is implementation method which is resistant to white-box attack. We describe the WBC-AES and contents protection method using it.

키워드

White-box cryptography, AES, contents protection

Ⅰ. 서 론

기존의 암호 알고리즘은 암호 알고리즘이 동작하는 단말과 이를 사용하는 사용자를 믿을 수 있다는 가정 하에 암호알고리즘이 공개 되어 있더라도 암호 키가 드러나지 않는다면 암호문을 획득한 공격자가 이 암호문을 해독할 수 없다는 가정 하에서 개발되어 사용되고 있다. 그러나 실제 환경에서 사용자가 공격자가 되어 암호 키를 제3자에게 노출할 수도 있고, 사용자가 사용하는 단말에 심어진 악성 프로그램이 공격자가 될 수도 있다[1]. 이에 대한 해결책으로 TPM, 스마트카드 등의 하드웨어를 이용한 방법이 제시되었으나, 하드웨어 사용에 대한 비용증가 및 설치의 어려움, 내부 결함 발생 시 업데이트 혹은 패치의 어려움 등의 문제가 있다. 그러나 화이트박스 암호 기술은 소프트웨어만으로 암호 키를 안전하게 보관할 수 있고, 신뢰할 수 없는 단말에서 암호화 알고리즘이 실행되더라도 암호 키가 드러나지 않도록 할 수 있는 기술로써 다양한 암호 알고리즘에 대한 화이트박스 암호 구현 기법에 대한 연구가 진행 중에 있다. 또한 블록암호 알고리즘을 화이트박스 암호 구현 기법으로 구현함으로써 블록암호 알고리즘의 가장 큰 고민거리인 키 분배 문제를 해결할 수 있다. 즉, 암호 키가 암호 알고리즘(화이트박스 테이블) 속에 숨겨져 있고, 사용자는 암호 알고리즘(화이트박스 테이블)을 서버로부터 다운로드 받아서 데이터의 암호화 혹은 복호화에 사용하기 때문에 공개키 암호의 공개키 분배처럼 화이트박스 테이블을 공개적으로 다운로드 가능하다(별도의 키 교환 알고리즘을 적용할 필요가 없다).

본 논문에서는 이러한 특성을 가진 화이트박스 암호 알고리즘을 이용하여 콘텐츠를 보호하는 방법에 관하여 기술하고자 한다.

Ⅱ. AES의 화이트박스 암호 구현 기법

화이트박스 암호 구현 기법은 2002년 S.Chow에 의해서 처음 제안 되었다[2]. S.Chow가 제안한 화이트박스 AES 구현 기법에 따르면 AES 암호 알고리즘을 여러 개의 룩업 테이블로 구분하여 구현하고, 이들 룩업 테이블을 통과하면 평문을 암호화 하거나 암호문을 평문으로 복호화 할 수 있다. 암호 알고리즘을 하나의 큰 룩업테이블로 만들면 암호키를 숨기는 것이 용이하지만, 테이블 크기가 지나치게 커져서 이를 실제 응용에 적용하는데 한계가 있기 때문에, 테이블을 암호학적인 기법으로 적절히 분리하되 테이블 중간값(암/복호 연산의 중간 값)이 노출되지 않도록 디코딩과 인코딩 과정을 수행한다. S.Chow의 화이트박스 암호 구현 방법의 기본 원리는 그림 1과 같다. 인코딩 과정(M_i)과 디코딩 과정(M_i^{-1})이 별도의 테이블에서 계산되므로 중간값이 노출되지 않으면서도 결국은 인코딩과 디코딩이 상쇄되면서 원래의 암호화 동작(X_i)만 수행하는 결과가 된다. 암호학적인 안전성을 위해서 외부 인코딩(F^{-1})과 디코딩(G) 과정을 거치게 되므로 동일한 암호키를 사용하더라도 AES를 이용한 암복호 결과와 화이트박스로 구현된 AES를 이용한 암복호 결과는 달라진다.

$$\underbrace{F^{-1}\circ M_1^{-1}}_{table}\circ\underbrace{M_1\circ X_1\circ M_2}_{table}\circ\underbrace{M_2^{-1}\circ M_3^{-1}}_{table}\circ\cdots\underbrace{M_{2i-1}\circ X_i\circ M_{2i}}_{table}\circ\underbrace{M_{2i}^{-1}\circ G}_{table}$$
$$\Leftrightarrow\ F^{-1}\circ X_1\circ X_2\circ\cdots\circ X_i\circ G$$

그림 1. 화이트박스 암호의 기본 원리

III. 화이트박스 암호를 이용한 콘텐츠 보호 방법

3장에서는 앞서 설명한 화이트박스 암호 테이블을 이용하여 콘텐츠를 보호하는 방법에 관하여 기술하고자 한다.

WBC-AES 암호 테이블을 이용한 데이터의 암/복호화에 소요되는 시간은 AES 암호 알고리즘을 이용한 데이터의 암/복호화에 소요되는 시간의 약 10배에 가까운 시간이 소요된다. 따라서 WBC-AES 암호 테이블을 이용하여 대용량의 고속 암/복호화가 필요한 콘텐츠를 암/복호화 하기에는 어려움이 있다. 이를 해결하기 위해서 블록암호의 모드 연산을 활용하는 방법이 있을 수 있다. 특히 AES의 CTR(counter) 모드는 nonce와 counter값을 AES 암호화 연산을 통과시킨 후 암/복호화 하고자 하는 데이터와 exclusive OR 연산 함으로써 데이터에 대한 암/복호화가 가능하므로 이를 콘텐츠의 고속 암/복호화에 적용 가능하다. 즉 특정 nonce와 counter 값을 화이트박스 AES 룩업테이블을 통과한 값을 미리 확보해 두고, 암호화 혹은 복호화 하고자 하는 콘텐츠가 입력되었을 때, WBC-AES 룩업테이블을 통과한 결과 데이터와 콘텐츠를 exclusive OR 연산함으로써 콘텐츠의 고속 암/복호화가 가능하다.

그림 2와 같이 화이트박스 AES CTR 모드를

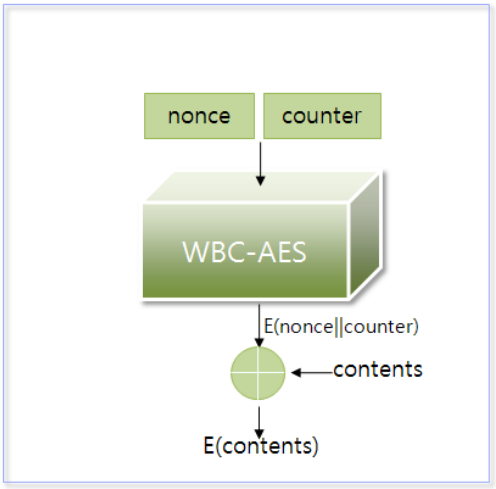

그림 2. 블록암호의 CTR 모드를 이용한 콘텐츠 암호화

콘텐츠 암/복호화에 적용함으로써 WBC-AES의 단점인 AES에 비해 느린 암/복호 속도를 극복할 수 있고, AES 이용시 겪게되는 콘텐츠 암호화 키의 분배 문제와 콘텐츠 암호화키가 암호 연산 도중 메모리에 로드되어 키가 노출되는 문제를 해결할 수 있다.

IV. 결론

본 논문에서는 화이트박스 암호 기법에 관하여 소개하고, 이를 처음으로 제안한 S.Chow의 화이트박스 구현 방법에 관하여 간략하게 기술하였다. 또한 이러한 WBC-AES를 이용하여 콘텐츠를 보호하는 방법에 관하여 기술하였다. 본 논문에서 제시한 방법을 이용하면 WBC-AES의 단점을 보완할 수 있고, AES의 단점을 보완한 WBC-AES의 장점은 그대로 살릴 수 있다.

참고문헌

[1] M. Joye, "On White-Box Cryptography," Prod. 1st Int'l Conf. Security of Information and Networks(SIN 07), Trafford Publishing, 2008, pp. 7-12.

[2] S. Chow et al., "White-Box Cryptography and an AES Implementation," Proc. 9th Ann. Workshop Selected Areas in Cryptography(SAC 02), LNCS 2595, springer, 2002, pp. 250-270.