블록 암호화를 위한 효율적인 연산 모드

이창두*, 권오현**, 박규석*
*경남대학교 컴퓨터공학과
**동명정보대학교 컴퓨터공학과

An Efficient Operation Mode for Block Cipher

Chang-Doo Lee*, Oh-Hyeon Kwon**, Kyoo-seok Park*
*Dept. of Computer Engineering, Kyungnam University
E-mail : lcd2@korea.com, kspark@kyungnam.ac.kr
**Dept. of Computer Engineering, Tongmyong University Of Information Technology
E-mail : ohkwon@tmic.tit.ac.kr

요 약

암호화를 위한 블록간의 연산 모드에서 어떤 연산 모드를 사용하는가에 따라 평문, 암호문 또는 키 추측 등에 많은 영향을 주게 된다. 본 논문에서는 여러 가지 블록 암호 연산 모드에 대해 알아보고, 다음 블록에 대한 키 정보의 암호화 강도를 효율적으로 높일 수 있는 RFB(Round key FeedBack) 운영 모드를 제안한다.

1. 서론

컴퓨터 보급의 증가와 통신기술의 발달로 인해 많은 데이터들이 인터넷과 같은 네트워크를 통하여 전파되고 있다. 전파되는 데이터 중에는 예금통장번호, 주민등록번호, 저작권이 있는 멀티미디어 자료 등 안전하게 전송, 저장하여야할 자료들은 암호화과정이 수행되어야 한다.

그 대표적인 암호화 알고리즘이 블록 암호화(Block cipher)기법이며, 블록 암호화에서는 자료를 일정한 크기로 나누어 블록 단위로 암호화를 수행한다.

컴퓨터의 급속한 처리 속도 향상으로 인하여 기존 64비트 암호화 알고리즘인 DES는 표준화 기간이 만료 되었고, 새로이 128, 192, 256비트 암호화 알고리즘인 AES(Advanced Encryption Standard)가 2000년 10월에 표준화 되었다[1].

본 논문에서는 기존 연산 모드를 분석하고 128블록 암호화에서의 효율적인 연산 모드를 제안한다. 논문의 구성은 다음과 같다. 먼저, 2장에서는 기존 블록 암호 연산 모드를 살펴보고, 3장에서는 128비트 블록 암호의 효율적인 연산 모드인 RFB(Round key FeedBack) 운영 모드 제안, 키 안정성 분석을 하고, 4장에서는 구현, 마지막으로 5장에서는 결론 및 향후의 연구 과제에 대해 기술한다.

2. 블록 암호 연산 모드

일반적으로 블록 암호화 알고리즘에서 임의의 고정 크기로. 메시지를 암호화하기 위해 블록 단위로 연산 운영 모드를 이용하는데, 이런 운영 모드는 암호화 알고리즘 자체만큼 안전하고 효율적으로 처리해야 한다.

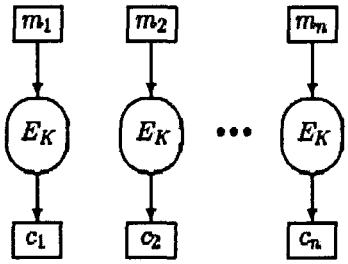
블록 암호화에도 적용할 수 있는 일반화된 운영 모드가 국제 표준인 ISO/IEC 10116으로 표준화되어 있으며, 이러한 운영 모드에는 ECB(Electronic Code Block), CBC(Cipher Block Chaining), CFB(Cipher FeedBack), 그리고 OFB(Output FeedBack) 등이 있다[2][3].

블록 암호화에서 임의의 길이가 주어졌을 때, 이를 블록 암호를 이용하여 암호화하려면, 우선 블록들을 크기만큼씩 나눈다. 이들을 $m_1, m_2, \ldots, m_n$이라 하자. 이렇게 구성된 n 개의 블록을 $m_1, m_2, \ldots, m_n$에 대하

여 ECB, CBC, CFB, OFB의 모드 등은 주어진 키 K를 이용하여 다음과 같이 수행된다.

• ECB(Electronic Code Block)

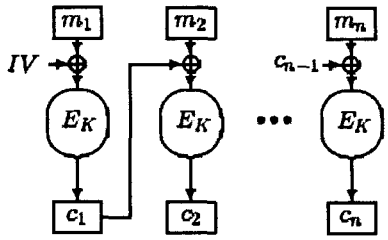
ECB 운영 모드에는 각 블록은 <그림 2-1>과 같이 다른 블록들과 무관하게 독립적으로 암호화가 수행된다. 따라서, 평문 블록 두개 m_1과 m_2가 동일한 경우 대응하는 암호문 c_1과 c_2도 같아지게 된다. 이것은 공격자가 암호문을 보고도 키와 평문의 특성을 쉽게 추측할 수 있는 여지가 있다.



<그림 2-1> ECB 운영 모드의 동작

• CBC(Cipher Block Chaining)

Cipher Block Chaining(CBC) 모드는 <그림 2-2>과 같이, i 번째 평문 블록 m_i에 대하여 이전 암호문 블록 c_{i-1}과 xor를 수행하고 이 결과에 암호화를 수행하여 i 번째 암호문 블록 c_i를 만든다. 첫 번째 평문 블록은 초기 치환 IV를 c_0로 설정하여 암호화가 수행된다. 그러나 이전 블록들의 영향을 전달하는 매개체가 c_{i-1}은 알고 있는 값이다. 즉, 입력과 출력인 $m_i \oplus c_{i-1}$과 c_i를 알 수 있으므로, CBC 모드에 의해 암호화된 n 개의 블록에 대하여 n 개의 평문/암호문 쌍이 얻어진다. 이것은 네트워크를 오가는 데이터가 CBC모드로 암호화된 경우 공격자가 알려진 평문 공격을 수행하면 데이터 크기에 비례하는 평문/암호문 쌍을 얻을 수 있음을 의미한다. 즉, CBC모드는 ECB 모드보다 안전하긴 하나 차분 해독법 등에 의한 공격에는 안전하지 못하다.

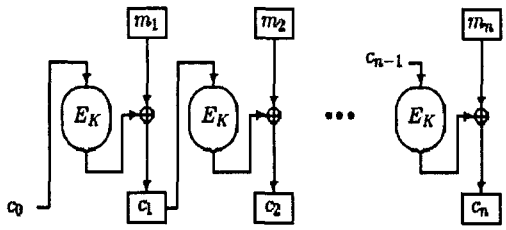


<그림 2-2> CBC 운영 모드의 동작

• CFB(Cipher FeedBack) 모드

CFB모드에서 각 블록은 <그림 2-3>과 같이 이전 암호문 블록을 암호화한 결과와 xor를 수행하여 해당 암호문을 생성한다. 이 경우에도 블록 암호의 입력은 c_{i-1}로 출력은 $m_i \oplus c_i$로 구해지므로 알려진 평문 공격을 수행하면 암호화된 n 개의 블록에 대하여 n 개의 평문/암호문 쌍을 얻는다.

CFB모드에서는 블록 암호를 수행해야 하는 횟수가 다른 모드들보다 많아지는 단점이 있다.

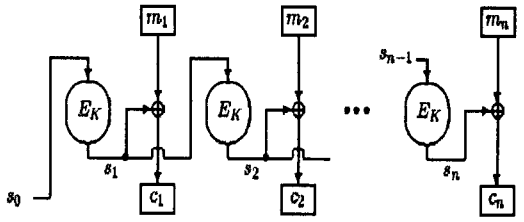


<그림 2-3> CFB 운영 모드의 동작

• OFB(Output FeedBack)모드

OFB모드는 CFB모드와 유사한데 평문과 xor되는 값이 이전 평문이나 암호문의 영향을 전혀 받지 않는다는 점만이 다르다.

OFB모드에서는 <그림 2-4>과 같이 블록 암호의 입력과 출력을 $m_{i-1} \oplus c_{i-1}$, $m_i \oplus c_i$로 구할 수 있어, 알려진 평문 공격을 수행하면 암호화된 n 개의 블록에 대하여 n 개의 평문/암호문 쌍을 얻을 수 있다.

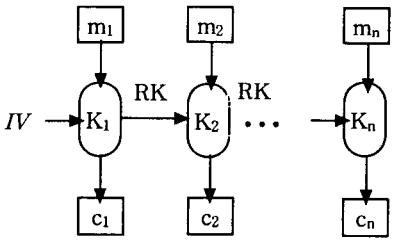


<그림 2-4> OFB 운영 모드의 동작

3. 제안 128비트 블록 암호 연산 모드

블록 암호에서 암호화 알고리즘 못지않게 암호 연산 모드 운영 방식이 중요하다. 제안하는 운영 모드는 128비트 이상 블록 암호에서 라운드 키(Round key FeedBack)를 이용하여 매 블록마다 키를 다르게 생성하여 차분 해독법에 강하게 설계하였다.

RBF 운영 모드는 <그림 3-1>에서와 같이 먼저 원 키를 패딩 처리하고, 초기 치환(*IV*)을 하여 원 키를 알아보기 어렵게 함수 처리한다. 함수 처리된 키를 라운드 키 함수에 의해 라운드 키를 생성한다. 생성된 라운드 키의 마지막을 다음 블록의 키로 활용한다.



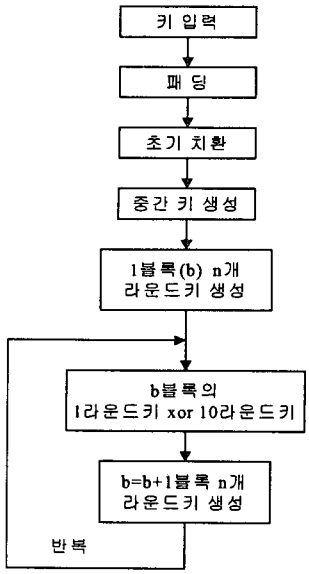
<그림 3-1> RFB 운영 모드의 동작

3.1 RFB모드 동작 개요

라운드 키(Round Key)에 의한 RFB(Round key FeedBack)모드의 동작과정은 크게 패딩(덧붙이기) 처리, 초기 치환, 중간 키 생성, 블록 라운드 키 생성 등 <그림 3-2>와 같은 과정으로 수행된다.

3.2 패딩

블록 암호는 일정한 크기로 한정되어 있으므로 키의 나머지 부분을 패딩 처리해 주어야한다. 예를 들면 128비트 암호화키에서 사용자가 암호를 영문자 9자인 72비트만 입력하였다면, 나머지 56비트는 패딩 규칙에 따라 56비트를 채우게 된다. 본 논문에서는 일반적으로 많이 활용하는 방법으로 나머지를 0으로 채운다.



<그림 3-2> RFB 운영 모드 동작 과정

3.3 초기 치환

사용자가 키를 입력할 때 같은 수를 일정하게 반복한다든지, '12345···'등 수가 순서적으로 입력하게 되면, 키 스케줄에서 키 노출의 위험성이 많아지게 된다. 따라서 사용자가 키 입력에서 노출되기 쉬운 암호를 사용하더라도 키 스케줄에서 치환시킴으로써 보다 더 견고한 라운드 키를 만들 수가 있다.

초기 치환 상수는 의사난수 50개를 발생하여 비트 나열 예측이 어려운 16진수 '7a49f06b317c8d59'를 선택한다.

3.4 중간 키 생성

본 과정은 라운드 키 함수 입력 전에 초기 치환된 키를 해독하기 어려운 키로 만드는 것으로, 1문자만 다르게 하여도 라운드 키의 배열을 변화시켜 키 암호화 강도를 높일 수 있다.

먼저 첫 8bit와 6bit좌회전한 128bit와 xor 한다. 다음으로 8bit 8행으로 행렬A1, B1, A2, B2 4개를 만들어 A1 xor B2, A2 xor B2의 결과 값으로 128bit 중간 키를 만든다.

3.5 라운드 키 생성

블록 암호화에서 각각의 라운드별 암호화를 수행하

면서 키가 필요한 경우가 많다. 이 때 라운드마다 필요한 키를 라운드 키라 한다. 이렇게 구성함으로써 암호화 강도를 높일 수 있다. 본 논문에서는 8블록 10라운드 암호화라 가정하고 라운드 키를 생성한다.

첫 8bit와 6bit좌회전한 128bit와 xor를 2회 반복하여 제 1라운드 키를 만들고, 계속해서 10라운드 키를 만든다.

1블록에서 만들어진 1라운드 키와 1블록의 10라운드 키를 xor한 후 2블록의 라운드 키 함수에 대입하여 2블록의 라운드 키를 만든다. 이 과정으로 계속해서 7블록의 1라운드 키와 7블록의 10라운드 키를 xor해서 8블록의 라운드 키를 만들어 라운드 키 생성을 종료한다.

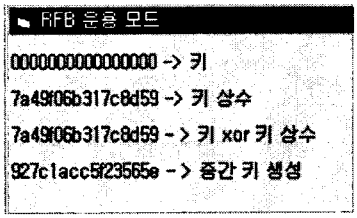
3.6 키 안정성

RBF 운영 모드에서 i 번째 블록은 $i-1$개의 블록과 각 블록당 10개의 라운드 키를 알아야 전수 탐색이 가능하다. 즉 i 번째 블록 키를 알려면 $i-1$개의 키를 모두 복호화 하여야 한다.

128비트 키에서 키 전수 탐색의 시간 복잡도는 초기 치환 상수(IV)가 공개되면 키의 전수 탐색은 $O(2^{128})$이 되고, IV가 공개되지 않은 $i+1$번째 블록 키의 전수 탐색은 $O(2^{128+64*i})$이상으로 증가되는 효과를 볼 수 있다.

4. 구현

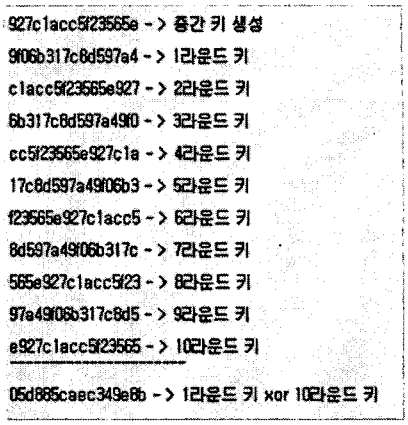
암호 공격자가 암호화 알고리즘을 공격할 때 보통 키를 모두 0로 두고 암호화 알고리즘을 공격하는 경우가 많다. 이 경우를 감안하여 <그림 4-1>과 같이 키를 모두 0으로 두고 첫 중간 키를 생성하였다.



<그림 4-1> 중간 키 생성

16진수 927c1acc5f23565e가 1블록의 라운드 키를 만드는 중간 키가 된다.

이 중간 키를 이용하여 10라운드 키를 <그림 4-2>와 같이 생성하게 된다.



<그림 4-2> 1블록 라운드 키와 2블록 중간 키

1블록의 1라운드 키와 마지막 라운드인 10라운드 키를 xor한 값이 다음 라운드 키의 생성을 위한 중간 키가 된다.

블록 암호 연산 모드 중 공격자가 가장 쉽게 평문을 추측할 수 있는 모드가 ECB로써 같은 암호문을 각 블록마다 사용하게 된다. CBC, CFB의 모드는 앞 암호문 또는 평문과 연관되어 있어 노출될 위험성이 많다.

제안하는 RFB는 각 블록에 생성되는 라운드 키를 사용하여 다음 블록에 영향을 주기 때문에 다른 방법에 비해 키 또는 평문을 추측할 수 있는 소지가 적다.

5. 결론

블록 암호화 알고리즘에서, 암호화 알고리즘도 중요하지만 키 스케줄과 운영 모드 또한 중요하다. 공격자는 블록과 블록간의 연관을 가지고 공격하는 경우가 많으므로 RFB와 같은 운영 모드를 사용함으로 다음 블록에 대한 공격을 어렵게 할 수 있다.

앞으로는 128비트 이상의 암호화 알고리즘을 주로 사용함에 따른 효율적인 키 운영 방식이 더 많이 연구되어야 할 것이다.

[참고문헌]

[1] NIST, AES(Standard for Advanced Encryption) Information, 2002
http://csrc.nist.gov/encryption/aes/rijndael/

[2] Gilles Brassard. Modern Cryptology-A tutorial, Lecture Notes on Computer Science, volume 325. 1988

[3] 이민섭, 현대암호학, 교우사, 2000