TTA표준 소개

블록암호알고리즘 SEED의 운영모드 표준 (TTAS.KO-12.0025)

임선간 * TTA, 정보보호기반 프로젝트그룹 위원
한국정보보호진흥원 암호인증기술팀장
강성우 * TTA, 정보보호기반 프로젝트그룹 위원
한국정보보호진흥원 암호인증기술팀 연구원

1. 서론

블록암호알고리즘 SEED는 128비트의 비밀키를 이용하여 128비트의 평문을 암호문으로 변환하는 암호알고리즘을 말한다. SEED는 1999년 9월 TTA 표준으로 제정(TTAS.KO-12.0004)된 이후 금융권, 전자상거래, 정보보호제품(VPN) 등의 다양한 분야에서 데이터의 기밀성(Confidentiality)과 무결성(Integrity) 기능을 제공하기 위해 사용되고 있다[5]. 2004년 3월 현재, nCipher, RSA Security, Chrysalis-ITS 등의 국외 주요 정보보호업체를 포함한 670개 이상의 국내외 산 · 학 · 연에서 SEED를 사용하고 있다.

블록암호알고리즘의 경우, 암호알고리즘이 적용되는 블록크기가 정해져있기 때문에 정해진 길이보다 긴 데이터를 암호화하거나 데이터 무결성 검증을 하기 위해서는 블록암호알고리즘의 운영모드를 반드시 사용하게 된다. 부가적으로, 입력 데이터의 길이가 기본 블록크기의 배수가 되지 않으면 처리가 불가능하므로 입력 메시지의 길이가 블록 길이의 배수가 되도록 하기 위해 덧붙이기 방법을 사용해야 한다. 이에, 한국정보보호진흥원에서는 안전성 측면과 효율성 측면을 고려하여 SEED의 사용을 촉진하기 위해 SEED의 운영모드를 TTA 표준(안)으로 제안하여, 2003년 12월 TTA 표준(TTAS.KO-12.0025)으로 제정하였다.

본 고에서는 2003년 12월에 TTA의 표준으로 제정된 "블록암호알고리즘 SEED의 운영모드"에서 제안하고 있는 5개의 표준 운영모드와 3개의 덧붙이기 방법의 개념 및 세부기능을 기술하고자 한다[6].

2. SEED의 운영모드

2.1 블록암호알고리즘의 운영모드에 대한 개요

지금까지 널리 알려지거나 사용되고 있는 운영모드는 ECB(Electronic CodeBook), CBC(Cipher Block Chaining), CFB(Cipher FeedBack), OFB(Output FeedBack) 등이 있고, 최근 CTR(CounTeR) 모드가 개발되어 사용되고 있다. 블록암호알고리즘의 운영모드는 크게 블록단위로 처리될 때, 동일한 키에 대해 이전 블록의 암호화 결과가 다음 블록에 미치는 영향과 암호문에서 발생한 에러가 평문의 복호화에 미치는 영향(error propagation)에 따라 분류되어진다.

ECB 모드와 CTR 모드는 이전 블록의 암호화 결과가 다음 블록에 영향을 주지 않고 독립적으로 작용한다. 따라서, ECB와 CTR 운영모드는 블록 암호화가 서로 독립적으로 작용하기 때문에 이전 블록 값에서 발생한 에러는 이후 블록값에 영향을 주지 않게 된다. 이러한 운영모드는 특정한 블록만을 대상으로 암호화 또는 복호화를 하는 환경에서는 효과적이라는 장점을 갖고 있다. 그러나 ECB의 경우, 동일한 암호문 블록의 복호화는 항상 동일하다는 문제점을 갖고 있어서, 암호화해야하는 데이터가 기본 블록단위보다 큰 경우에는 사용을 권고하지 않는다. 반면에 CTR 모드는 각 블록마다 서로 다른 카운터를 사용하여 ECB 모드의 이러한 문제점을 개선하고 있다.

CBC, CFB, OFB 모드는 이전 블록의 출력값(암호화 값)이 다음 블록의 암호화에 영향을 줌으로써 블록 단위(평문, 암호문)쌍에 대한 패턴을 숨긴다. 이 세 가지 모드는 이전 블록값에서 발생한 에러의 전파에서 각 각 서로 다른 특성을 가진다. CBC모드의 경우, 이전 출력 블록값에서 발생한 에러는 항상 그 다음 출력 블록값에 영향을 미치고, CFB모드는 이전 출력 블록값의 상위 비트에서 발생한 에러만 이후 출력 블록값에 영향을 미치고, OFB모드의 경우, 이전 출력 블록값에서 발생한 에러는 이후 출력 블록값에는 영향을 미치지 않는다.

특히, OFB 모드와 CFB 모드는 스트림 암호와 같이 초기값과 암호키를 이용하여 키 스트림 과정을 수행한 후, 이러한 과정을 통해 생성된 출력 블록값과 평문 데이터 블록값의 비트별 배타적 논리합을 수행하게 된다. OFB 모드의 경우에는 암호키를 재사용하면, 하나의 평문/암호문 쌍으로 주어진 암호문에 대한 평문을 쉽게 계산할 수 있으므로 입력 초기값을 다르게 사용해야한다. 또한, OFB 모드의 경우, 반복적으로 작용하는 변환함수의 입력값에 대한 출력값 전체를 사용하여 평문 블록과 비트별 배타적 논리합을 수행하도록 권장한다. SEED의 운영모드 표준에서는 SEED의 출력값 128비트 모두를 사용하도록 규정하고 있다.

다음 표는 운영모드의 특성에 따른 구분을 한 것이다.

구분		특징
암호화가 각 블록에 독립적으로 작용하는 운영모드	ECB	– 같은 키에 대해, 동일한 평문 블록이 동일한 암호문 블록으로 출력된다.
	CTR	– 같은 키에 대해, 서로 다른 카운터를 이용하여 동일한 평문 블록이 서로 다른 암호문 블록으로 출력된다.
이전 블록의 암호화 값이 다음 블록에 영향을 주는 운영모드	CBC	– 이전 블록 값에서 발생한 에러는 이후 블록 값에 영향을 준다. – 무결성 검증에 사용되는 MAC 값을 생성하기 위해 주로 사용하고 있다.
	CFB	– 이전 블록 값의 최상위 r 비트에서 발생한 에러는 이후 블록 값에 영향을 준다.
	OFB	– 키스트림이 평문과 암호문에 의존하지 않기 때문에, 암호화된 블록에서 발생되는 에러는 다음 블록에 영향을 주지 않는다.

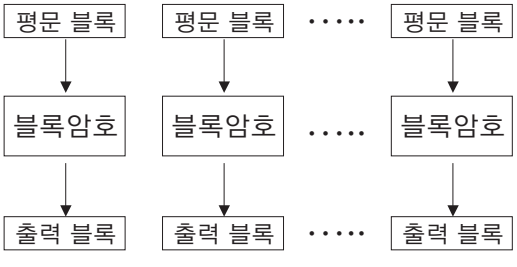
블록암호의 운영모드 표준은 표준 블록알고리즘을 구현하는데 필요한 운영모드를 명시하게된다.

현재, 가장 널리 사용되는 블록암호알고리즘의 운영모드 표준은 DES(Data Encryption Standard)에 적용할 목적으로 개발된 4개 운영모드 ECB, CBC, CFB, OFB 모드가 있다[2, 3]. 또한, 2001년에는 AES (Advanced Encryption Standard)에 적용하기 위한 새로운 운영모드인 CTR 모드를 개발하여 사용하고 있다[7]. 국내 블록암호알고리즘 표준인 SEED의 운영모드에 대해서는 본 표준에서 ECB, CBC, CFB, OFB, CTR을 명시하고 있다.

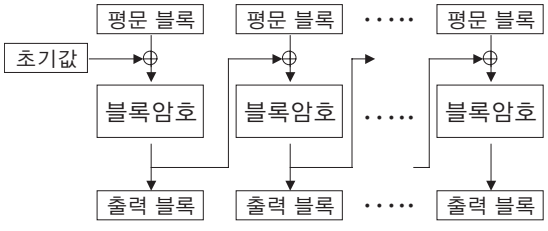
2.2 SEED의 운영모드 표준

본 표준에서 명시하고 있는 SEED에 대한 운영모드 ECB, CBC, CFB, OFB, CTR을 간략하게 소개하고자 한다.

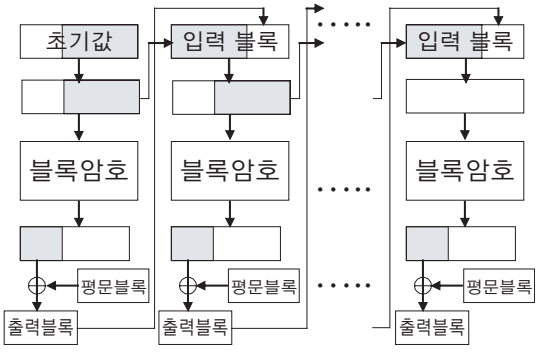
ECB 모드는 평문 데이터를 128비트 블록 단위의 독립적인 적용을 통해 암호문 데이터를 얻는 운영모드를 말한다.



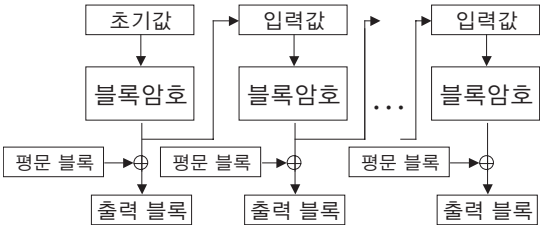
CBC 모드는 평문 데이터의 128비트 블록으로 구분하여, 이전 128비트 출력(암호문) 블록값과 128비트 평문 블록값의 배타적 논리합을 수행한 후, 블록암호의 입력값으로 처리하는 운영모드를 말한다.



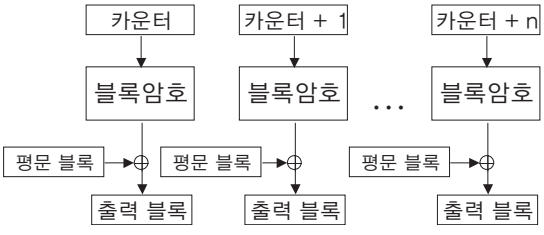
CFB 모드는 평문 데이터를 r 비트로 분할한 평문블록과 블록암호의 입력값의 변환된 값의 r 비트를 비트별 배타적 논리합을 수행한다. 이 값은 다음 블록의 입력값으로 사용된다.



OFB 모드는 평문 데이터를 블록길이로 분할한 값과 블록암호의 입력 초기값의 변환값을 비트별 배타적 논리합을 수행한다. 입력 블록의 변환값을 다음 입력 블록 값으로 사용한다.



CTR 모드는 입력값으로 카운터를 사용하고 블록암호를 통해 변환된 값은 평문 블록과 비트별 배타적 논리합을 수행한다. 이 때 사용되는 카운터는 데이터를 암호화할 때마다 서로 다른 값을 사용하도록 해야한다.

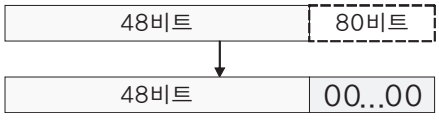


2.3 덧붙이기 방법

운영모드를 이용하여 입력 값을 처리하기 위해서는 평문 데이터의 길이는 128비트의 양의 정수배가 되도록 덧붙이기(Padding)를 해야한다. 본 표준 TTAS.KO-12.0025에서 명시된 덧붙이기 방법은 3가지로 ISO/IEC 9797-1에서 설명하고 있는 3가지 방법[4]을 SEED에 적용한 것이다.

덧붙이기 방법1은 평문 데이터의 크기가 128비트의 정수배가 아닐 경우, 최하위비트에 비트값 '0'을 추가하여 128비트의 배수가 되도록 덧붙이기한다. 그러나, 이러한 방법은 복호화할 평문 데이터와 추가로 덧붙여진 값과의 구분이 명확하지 않으므로 평문데이터의 크기를 반드시 명시해 주어야 한다.

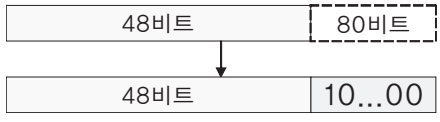
예)



평문데이터 : 4F 52 49 54 48 4D
덧붙이기 결과 : 4F 52 49 54 48 4D 00 00 00 00 00 00 00 00 00 00

덧붙이기 방법2는 평문 데이터의 크기가 128비트의 정수배가 아닐 경우, 최하위비트에 비트값 '1'을 추가하고 난 후, 128비트의 배수가 되도록 '0'을 덧붙이기 한다.

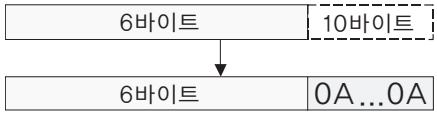
예)



평문데이터 : 4F 52 49 54 48 4D
덧붙이기 결과 : 4F 52 49 54 48 4D 80 00 00 00 00 00 00 00 00 00

덧붙이기 방법3은 평문 데이터를 바이트단위로 처리하며, 평문 데이터의 크기가 16바이트의 정수배가 아닐 경우, 16의 배수가 되기 위해 필요한 바이트의 개수를 한 바이트로 표현하여 필요한 바이트 수만큼 최하위 바이트에 덧붙인다. 예 1)에서 보는바와 같이 16의 배수가 되기 위해 10개의 바이트 덧붙이기가 필요하며, 덧붙이기에는 10에 대한 바이트 표시인 '0A'를 10번 반복하여 덧붙이게 된다.

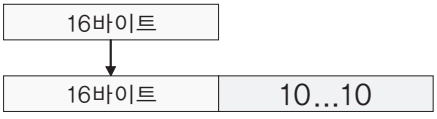
예 1)



평문데이터 : 4F 52 49 54 48 4D
덧붙이기 결과 : 4F 52 49 54 48 4D 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A

만일 평문 데이터가 16바이트의 정수배가 되면, 16바이트 '10 10... 10'를 덧붙이기 한다.

예 2)



평문데이터 : 53 45 45 44 41 4C 47 A8 3E D1 AF 07 4A 73 12 2C

덧붙이기 결과 : 53 45 45 44 41 4C 47 A8 3E D1 AF 07 4A 73 12 2C 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10

3. 결론

SEED는 국내 TTA 표준 암호알고리즘으로서, 금융권, 전자상거래 등의 다양한 분야에서 데이터 및 주요 정보를 암호화하는데 폭넓게 사용되고 있다. 또한 SEED는 2004년 3월 현재, ISO/IEC, IETF등을 통해 국제 표준화를 추진하고 있다. 2003년 12월에 제정된 TTA 표준 "블록암호알고리즘 SEED의 운영모드"는 다양한 환경 및 목적에 맞게 SEED를 구현할 수 있는 SEED의 운영모드 표준이다. 본 표준에서는 데이터의 기밀성과 무결성, 메시지 인증에 사용하기 위한 SEED의 운영모드를 정의하고 관련 테스트 벡터값을 제시함으로써, 정보보호시스템 및 서비스에서 SEED 사용에 대한 편의성을 제고하고자 하였다.

참고문헌

[1] A.J.Menezes, P.C.Van Oorschoot, and S.A.Vanstone, "Handbook of applied cryptography", CRC Press, Boca Raton 1997.

[2] FIPS 81, "DES modes of operation", Federal Information Processing Standards Publication 81, US department of Commerce/NIST, Springfield, Virginia, December 1980.

[3] FIPS, "FIPS 46-3, Data Encryption Standard(DES)", October 1999.

[4] ISO/IEC 9797-1, "Information technology - Security techniques - Message Authentication Codes(MACs) Part 1: Mechanisms using a block cipher", 1999.

[5] KISA, "128비트 블록암호알고리즘 표준 SEED", TTAS.KO-12.0004, 1999. 9. 28.

[6] KISA, "블록암호알고리즘 SEED의 운영모드", TTAS.KO-12.0025, 2003. 12. 18.

[7] NIST, " SP 800-38A 2001 ED, Recommendation for Block Cipher Modes of Operation", December 2001.