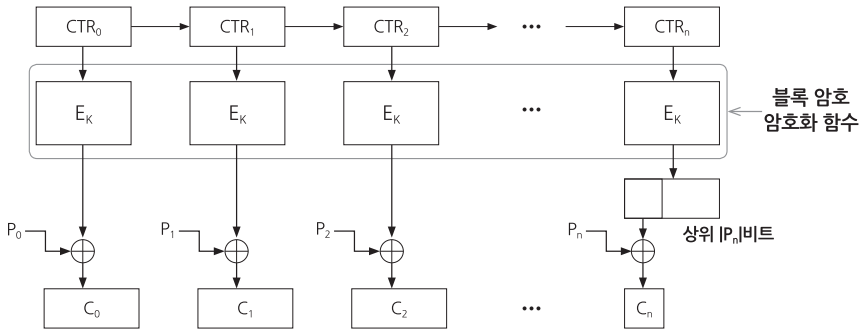
# n 비트 블록 암호 운영 모드

**박제홍** 정보보호기반 프로젝트그룹(PG501) 부의장
국가보안기술연구소 책임연구원

## 1. 머리말

암호 알고리즘은 데이터에 대한 기밀성(confidentiality), 무결성(integrity) 등 다양한 정보보호 서비스를 제공하는 데 사용되는 핵심 기술이다. 블록 암호(block cipher)는 특히 기밀성 보장을 위해 필요한 암호화 기술의 핵심 요소로 인식되고 있다. 일반적으로 블록 암호는 고정 길이의 암호키와 데이터를 입력으로 받아 라운드 함수라고 하는 연산 구성을 반복 수행하는 방식으로 입력 데이터와 동일한 길이의 암호문(ciphertext)을 생성한다. 현재 많이 사용되고 있는 범용 블록 암호의 데이터 입력 길이(이하 블록 길이)는 128비트이며, 경량 환경을 대상으로 저면적 구현이 용이하게 설계된 블록 암호의 경우 이보다 짧은 64비트를 많이 채택하고 있다. 참고로 국내에서 개발되어 현재 사용되고 있는 블록 암호 LEA[2], ARIA[1], SEED[3]는 블록 길이가 128비트이고 HIGHT[4]는 64비트이다.

그러나 실제 응용(application)에서 처리하는 데이터의 크기는 가변적이다. 블록 암호 알고리즘을 이용하여 이러한 다양한 크기의 데이터를 암호화할 때 가장 단순하게 고려할 수 있는 방법은 데이터를




[그림 1] 블록 암호 운영 모드 동작 예(CTR 모드)

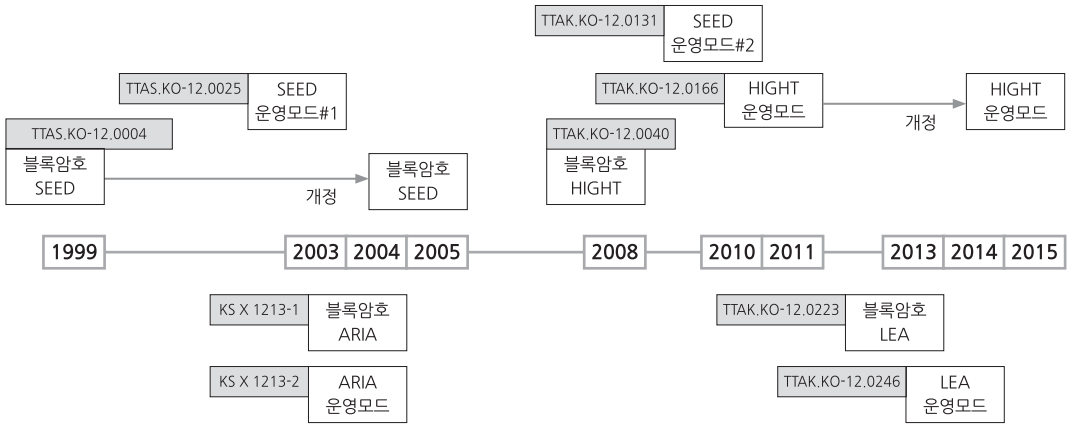
블록 길이 단위로 분할하고 개별 블록을 암호화하여 그 결과(암호문 블록)를 순서대로 연접하는 것이다. 그러나 동일한 데이터 블록을 대상으로 암호키를 고정하여 암호화를 반복 수행할 경우 항상 동일한 암호문 블록을 생성하게 된다. 이러한 특성으로 인해, 데이터를 구성하는 다수의 블록이 동일한 값을 가지는 경우 암호화된 데이터에서도 그 구성이 드러나게 된다. 따라서 암호화를 통해 데이터와 관련된 모든 정보의 노출을 차단하려는 기밀성 요구조건을 충족시킬 수 없게 된다. 이는 안전한 블록 암호 알고리즘을 사용한다고 하더라도 가변 길이 데이터에 대한 안전성을 보장하기 위해서는 별도의 방법이 필요하다는 것을 보여준다.

[그림 1]과 같이 단일 블록을 처리하는 블록 암호를 반복 적용하여 가변 길이 데이터에 대한 정보보호 서비스를 제공하는 방법을 정의한 것이 바로 블록 암호 운영 모드(block cipher mode of operation)이다. 블록 암호 운영 모드의 등장은 미국 연방정부에 도입되는 암호제품에 사용할 수 있는 블록 암호 DES 기반 암호화 방법으로 규격화된 것(NIST FIPS 81)에서 출발하였으며, 제안된 4종의 운영 모드들(ECB, CBC, CFB, OFB)은 데이터의 기밀성 보장을 주요 설계 목표로 하였다. 이후 보호 대상이 되는 데이터의 종류와 특성이 세분화되고 정교해짐에 따라 블록 암호 운영 모드에 대한 기능 및 안전성 요구조건 또한 다양해지고 있다. 이러한 기술 수요를 반영하여 학계에서는 기밀성 뿐만 아니라 무결성과 같은 다른 정보보호 서비스를 효율적으로 제공하거나 특정 환경에 최적화된 운영 모드 설계 연구를 진행하고 있다. 그 결과로 현재 여러 가지 블록 암호 운영 모드가 개발되어 다양한 응용에 사용되고 있다. 실제로 인터넷 서비스 보안을 위해 많이 사용되고 있는 SSL/TLS[7], IPsec[6], SRTP[5] 등의 암호 프로토콜에서는 데이터에 대한 기밀성(과 무결성)을 제공하기 위한 용도로 운영 모드의 적용 방법을 상세하게 정의하고 있다. 그러나 프로토콜 규격에서 제시하는 운영 모드 적용 방법은 초기값(IV, Initialization Value)과 같은 주요 운영 모드 파라미터의 조건을 충족시킬 수 있도록 프로토콜의 구조와 특성을 파라미터 설정에 반영하는 것을 위주로 한다. 따라서 프로토콜에서 허용하는 블록 암호 운영 모드를 실제 구현하여 활용하기 위해서는 운영 모드 규격의 참조가 선행되어야 한다.

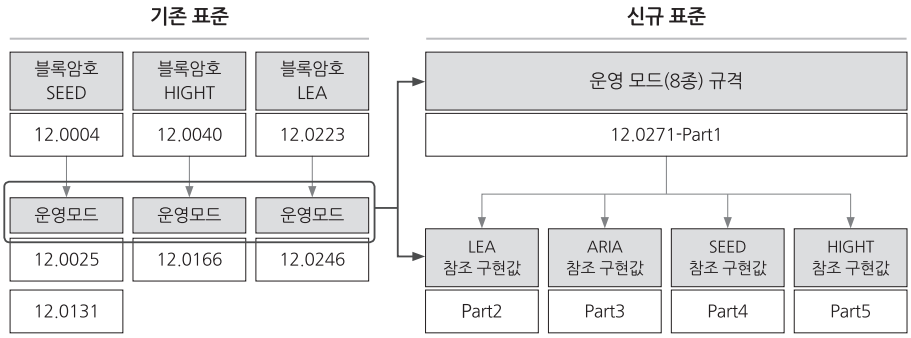
블록 암호 운영 모드의 기능과 역할을 반영하여 국가・공공기관 도입 암호제품에 탑재되는 암호모듈에 대한 구현 정확성을 검증하는 암호모듈 검증제도(KCMVP)에서는 블록 암호 운영 모드를 별도의 검증 대상 항목으로 분류하고 다양한 운영 모드를 검증 대상 알고리즘(보호함수)으로 지정하여 관리하고 있다. 본고에서 소개하는 운영 모드 표준은 보호함수로 지정된 블록 암호 LEA, ARIA, SEED, HIGHT를 암호제품에 적용함에 있어 조합 가능한 운영 모드의 규격과 참조 구현값을 제시하는 것을 목표로 개발되었다.

## 2. 표준 개발

해당 표준이 개발되기 전에도 국내에서는 블록 암호 운영 모드 표준이 제정되어 암호제품 개발에 활용되고 있었다. 그러나 국제 표준과는 달리 국내에서는 개별 블록 암호 알고리즘에 종속되는 형태로 운영 모드 표준이 개발되었다. [그림 2]는 국내 개발 블록 암호 알고리즘과 운영 모드의 표준 제정 시기를 도시한 것으로 개별 블록 암호 알고리즘의 이용 활성화를 위한 후속 표준으로 운영 모드 표준이 개발된 것을 확인할 수 있다. 이들 블록 암호 알고리




[그림 2] 국내 블록 암호 운영 모드 표준 개발




[그림 3] 운영 모드 표준의 개정

즘이 KCMVP 보호함수로 지정됨에 따라 암호제품 개발 및 검증에 활용하기 위한 참조 표준으로 블록 암호 알고리즘과 운영 모드 표준들이 모두 활용되고 있다.

[그림 1]과 같이 블록 암호 운영 모드는 기반 함수로 사용하는 블록 암호 알고리즘의 내부 구조와 독립적으로 설계된다. 따라서 입출력 길이가 동일한 블록 암호 알고리즘에 대해서 운영 모드는 별도의 수정 없이 동일하게 적용할 수 있다. 운영 모드의 이러한 특성과 향후 새로운 운영 모드의 추가 가능성을 고려하여 TTA 정보보호기반 PG(PG501)에서는 2014년 블록 암호 운영 모드 표준의 통합 관리방안을 수립하고 기존 표준을 정비하기 위한 작업을 추진하였다. 주요 관리방안을 정리하면 다음과 같다.

- 난수발생기(TTAK.KO-12.0189), 키 유도 함수(TTAK.KO-12.272) 등 다른 블록 암호 기반 암호 알고리즘 표준과 동일하게 블록 암호 운영 모드를 하나의 독립된 암호 알고리즘 표준으로 통합 관리
- 운영 모드 규격은 제1부(Part 1)에 정리
- 참조 구현값 생성 방식을 통일하고 개별 블록 암호 알고리즘(LEA, ARIA, SEED, HIGHT)을 적용한 운영 모드 참조 구현값을 생성하여 제2부를 시작으로 하는 연계 표준(family standard)으로 관리

[그림 3]은 통합 관리방안에 의한 기존 표준의 개정 방향을 도시한 것이다.

<표 1> n 비트 블록 암호 운영 모드 표준

| Part | 표준명<br>(n 비트 블록 암호 운영 모드) | 내용 | 제정 | 개정 |
|---|---|---|---|---|
| 1 | 제1부: 일반 | 운영 모드 규격 정의 | 2015 | 2016 |
| 2 | 제2부: 블록 암호 LEA | LEA를 적용한 참조 구현값 | | 2017 |
| 3 | 제3부: 블록 암호 ARIA | ARIA를 적용한 참조 구현값 | 2017 | - |
| 4 | 제4부: 블록 암호 SEED | SEED를 적용한 참조 구현값 | | |
| 5 | 제5부: 블록 암호 HIGHT | HIGHT를 적용한 참조 구현값 | | |

이러한 블록 암호 운영 모드 표준의 정비를 통해 기대하는 효과는 다음과 같다.

- 새로운 블록 암호 알고리즘이 표준으로 제정될 경우, 관련 운영 모드 참조 구현값을 운영 모드 표준에서 설정한 기준에 따라 생성하고 이를 연계 표준으로 추가하여 표준 사이의 일관성과 통일성 유지
- 새로운 운영 모드를 추가할 경우, 연계 표준 전체를 일괄적으로 개정하여 현행화 가능

2017년 완료된 정비 작업의 결과로 개발된 표준을 정리하면 〈표 1〉과 같다. TTA로부터 부여받은 표준 번호는 TTAK.KO-12.0271이다.

## 3. 표준 내용

### 3.1 운영 모드 규격(제1부)

제1부 일반에서는 KCMVP 보호함수로 지정된 블록 암호 운영 모드 8종의 규격을 제시하고 있다. 일반적으로 운영 모드는 제공하는 정보보호 서비스를 기준으로 크게 기밀성 전용(confidentiality only), 메시지 인증 전용(message authentication only), 그리고 인증 암호화(authenticated encryption)로 분류할 수 있다. 이러한 분류 기준에 따라 제1부에 포함된 운영 모드를 구분하면 다음과 같다.

- **기밀성 운영 모드:** ECB, CBC, CFB, OFB, CTR
- **메시지 인증 운영 모드:** CMAC
- **인증 암호화 운영 모드:** CCM, GCM

기밀성 운영 모드는 암호화를 통해 데이터에 대한 어떠한 정보도 노출시키지 않는 것을 목표로 설계되었으며 암호키를 고정하더라도 초기값을 변경함으로써 동일 평문에 대해 다른 암호문이 생성되도록 하는 설계 방식을 준용한다(ECB 모드 제외). 메시지 인증 운영 모드는 블록 암호를 기반 함수로 사용하는 메시지 인증 코드(message authentication code) 방식을 정의한 것으로 데이터에 대한 무결성과 근원 인증(source authentication) 서비스를 제공한다. 인증 암호화 운영 모드는 기밀성 운영 모드와 메시지 인증 코드를 적절하게 조합하여 하나의 암호키로 데이터에 대한 기밀성과 무결성을 동시에 제공한다. 표준에 정의된 GCM과 CCM 모드는 CTR 모드를 기반으로 동작한다.

개별 운영 모드는 구조에 따른 여러 가지 특성을 가지고 있다. 이러한 특성은 암호제품 개발 과정에서 운용 환경에 적합한 운영 모드를 선택하는 근거로 활용할 수 있다. 예를 들어, 비트당 오류율(BER, bit error rate)이 높은 통신 환경에서 전송률을 높이기 위한 오류 전파(error propagation) 최소화, 계산

<표 2> 주요 운영 모드 특성

| 운영 모드 | 오류 전파 (블록) | 블록 암호 복호화 | 덧붙이기 | 초기값 조건 | 병렬 처리 | | 사전 계산 |
|---|---|---|---|---|---|---|---|
| | | | | | 암호화 | 복호화 | |
| ECB | 유 | 필요 | 필요 | - | 가능 | 가능 | 불가능 |
| CBC | 유 | 필요 | 필요 | random | 불가능 | 가능 | 불가능 |
| CFB | 유 | 불필요 | 불필요 | random | 불가능 | 가능 | 불가능 |
| OFB | 무 | 불필요 | 불필요 | random | 불가능 | 불가능 | 가능 |
| CTR | 무 | 불필요 | 불필요 | nonce | 가능 | 가능 | 가능 |
| CCM | - | 불필요 | 불필요 | nonce | 불가능 | 불가능 | 불가능 |
| GCM | - | 불필요 | 불필요 | nonce | 가능 | 가능 | 가능 |
| CMAC | - | 불필요 | 불필요 | - | 불가능 | 불가능 | 불가능 |

자원(resource) 제약에 따른 구현 면적 최소화, 또는 데이터 고속 암호화를 위한 병렬 구현/사전 계산 활용과 같은 요구사항이 도출될 수 있으며 이러한 요구사항을 반영하여 개발자는 적합한 운영 모드를 선택할 수 있다. 또한 운영 모드가 보장하는 수학적인 엄밀한 안전성을 보장하기 위해서는 초기값과 같은 주요 파라미터에 대한 조건(랜덤(random) 또는 한 번만 사용(nonce))을 충족시킬 수 있는지 여부도 사전에 검토되어야 한다. 운영 모드의 적용과 관련하여 고려해야 하는 주요 특성들을 정리하면 〈표 2〉와 같다.

부록에서는 ECB와 CBC 모드에서 사용 가능한 덧붙이기(padding) 방식을 제시하고 있다. 특히 CBC 모드를 사용하는 암호 프로토콜의 실제 운용 과정에서 취약성 이슈로 부각된 패딩 오라클 공격(padding oracle attack)[11]에 내성을 가지는 덧붙이기 방법[10]을 포함하여 타 운영 모드 표준과의 차별성을 확보하였다.

### 3.2 참조 구현값

제1부를 제외한 연계 표준(제2부~제5부)은 KCMVP 보호함수인 블록 암호 LEA, ARIA, SEED, HIGHT를 운영 모드의 기반 함수로 사용하여 생성한 참조 구현값을 제시하고 있다. 기존 운영 모드 표준에서는 표준 개발자마다 다르게 설정한 기준에 따라 생성된 참조 구현값을 제시한 반면 개정 표준 개발 과정에서는 관리 방안에 따라 각 블록 암호 알고리즘을 적용한 운영 모드에 대해 동일한 기준으로 참조 구현값을 생성하였다. 이로부터 범용 암호모듈과 같이 다수의 블록 암호 알고리즘을 구현하는 경우에 자체 검증의 편의성을 높일 수 있도록 하였다.

## 4. 표준 활용

본 표준은 KCMVP 보호함수의 참조 규격으로 활용되던 블록 암호 운영 모드 표준을 대체하기 위한 목표로 개발되었다. 표준 개발 과정에서는 이를 고려하여 보호함수로 지정된 운영 모드 8종을 대상으로 규격을 정의하고 블록 암호 보호함수 4종에 대한 개별 참조 구현값을 생성하여 제시하였다. 따라서 이 표준은 새로운 KCMVP 보호함수 참조 표준으로 활용 가능하다. KCMVP 보호함수 참조 표준은 검증기관의 검증 기준으로 활용됨과 동시에 암호제품 개발 업체에게는 구현 기준으로 활용될 수 있다.

KCMVP의 검증을 받은 암호모듈은 국가・공공기관을 포함하여 다양한 국내 암호제품 시장에서 활용

되고 있는 점을 고려할 때 본 표준을 이용한 운영 모드 사용 기준과 방법의 일원화된 관리를 통해 국가기간망에 대한 일관성 있는 정보보호 대책 시행을 용이하게 할 수 있다.

최근 ARIA가 OpenSSL 1.1.1[8]에 탑재된 것과 같이 국산 암호 알고리즘의 적용처가 점차 국제적으로 확대되고 있는 점을 고려할 때 본 표준은 국내 개발 블록 암호 알고리즘의 암호제품 적용을 위한 유일한 참조 문서로써 알고리즘의 보급 확산에 기여할 수 있다.

## 5. 맺음말

미국 연방정부 도입 암호제품에 적용 가능한 암호기술의 규격화 및 표준화를 담당하고 있는 NIST(National Institute of Standards and Technology)의 경우 본 표준에 포함된 8종의 운영 모드 이외에도 디스크 암호화, 개인정보 암호화, Key Wrap 등 특화된 용도를 위한 전용 운영 모드의 규격을 개발(NIST SP800-38)하고 사용을 승인한 바 있다. 학계에서도 인증 암호화 운영 모드를 포함한 인증 암호화 기술 수요의 증가에 대응하여 연구 역량 강화를 위한 프로젝트(CAESAR competition)[9]를 진행하고 있다. 따라서 PG501에서도 블록 암호 운영 모드에 대한 학계 및 산업계의 연구 · 개발 및 적용 현황을 지속적으로 면밀히 검토하여 필요 시 본 표준에 반영할 계획이다.

참고로 본 표준을 국가 표준으로 상정하는 작업이 현재 진행 중이다. 제1부는 2016년 이미 국가 표준으로 제정(KS X 3254)되었으며, 2017년 TTA 표준으로 제정된 참조 구현값을 후속 표준으로 제안하여 현재(2018년 9월) 적부 여부를 검토 중에 있다.

## [참고문헌]


[1] KS, 정보기술 – 보안기술 – 128비트 블록 암호 알고리즘 ARIA – 제1부: 일반, KS X 1213-1, 2004.

[2] KS, 128 비트 블록 암호 LEA, KS X 3246, 2016.

[3] TTA, 128비트 블록암호알고리즘 SEED, TTAS.KO-12.0004/R1, 2005.

[4] TTA, 64비트 블록암호 HIGHT, TTAS.KO-12.0040/R1, 2008.

[5] IETF, The Secure Real-time Transport Protocol (SRTP), RFC 3711, 2004.

[6] IETF, IP Encapsulating Security Payload, RFC 4303, 2005.

[7] IETF, The Transport Layer Security (TLS) Protocol Version 1.3, RFC 8446, 2018.

[8] OpenSSL – Cryptography and SSL/TLS Toolkit. https://www.openssl.org

[9] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. https://competitions.cr.yp.to/caesar.html

[10] H.C. Kang, M. Park, D. Moon, C. Lee, J. Kim, K. Kim, and J. Kim, New efficient padding methods secure against padding oracle attacks, Proc. of ICISC 2015, LNCS, vol. 9558, pp. 329-342, 2016.

[11] S. Vaudenay, Security flaws induced by CBC padding – Applications to SSL, IPSEC, WTLS..., Proc. of EUROCRYPT 2002, LNCS, vol. 2332, pp. 534-546, 2002.