



128비트 블록 암호 LEA



박제홍 국가보안기술연구소 선임연구원

1. 머리말

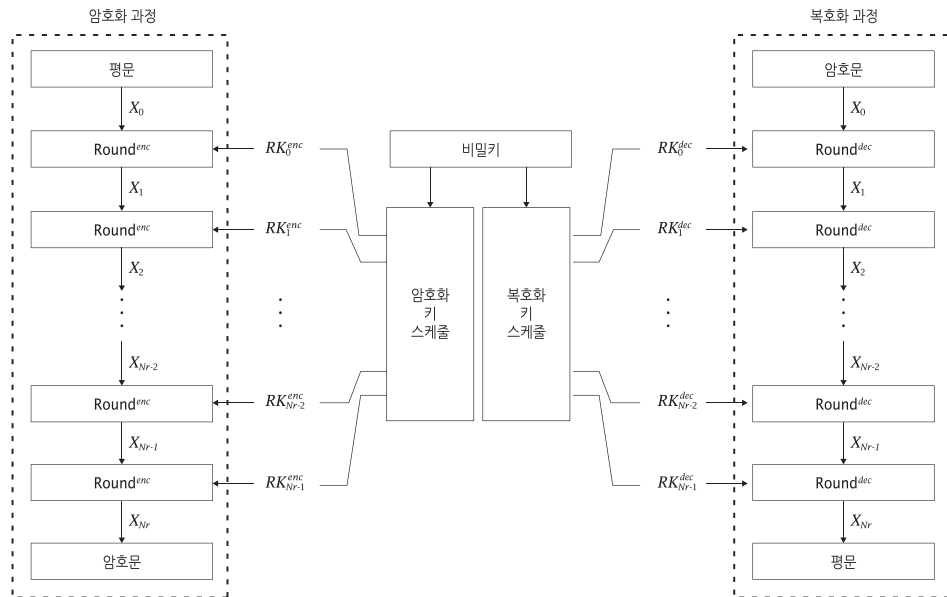
1.1 LEA 개발 배경

기거나 네트워크 보안을 위해서는 운용 환경에 따른 다양한 보안 요구사항을 충족시킬 수 있는 암호기술의 사용이 필요하다. 64비트, 128비트 혹은 그 이상 길이의 블록 단위로 데이터 암호·복호화 연산을 처리하는 블록 암호는, 기밀성이나 무결성과 같은 정보보호 서비스를 제공하는 핵심 암호기술이다. 블록 암호의 암호·복호화 절차는 라운드 함수를 주어진 라운드 수만큼 반복하는 구조로 되어 있다. 라운드 함수는 동일한 구조이지만, 라운드마다 비밀키로부터 생성되는 라운드 키가 반영되어 독립성이 부여된다. 블록 암호의 전체 동작 과정을 도시하면 [그림 1]과 같다.

블록 암호 설계 분야의 연구는 90년대 후반에 진행된 미국 연방정부 표준 블록 암호 AES(Advanced Encryption Standard) 공모사업을 통하여 비약적인

발전을 이루었다. AES[7]가 국제 표준으로서의 위상을 가지게 됨에 따라 범용 블록 암호 개발 사례는 급격하게 감소하였지만, 국내에서는 주요 시스템에 대한 안전성을 보장하고 시장 경쟁력을 확보하기 위해 SEED[4]와 ARIA[3]를 독자 개발하여 사용하고 있다.

최근 사물인터넷(IoT, Internet of Things) 시장의 폭발적인 성장이 전망됨에 따라, 세계 각국은 사물인터넷 서비스에 의한 다양한 부가가치 창출 가능성에 주목하면서 사물인터넷을 미래 성장 동력으로 내세우고 있다. 이에 시장 선점 및 주도권 확보를 위한 산업계의 적극적인 투자가 이루어지고 있으며, 시장 활성화를 통한 국가 경쟁력 강화를 위해 국가 단위의 지원 정책이 제시되고 있다. 이와 관련하여 미래창조과학부는 2014년 발표한 사물인터넷 기본계획[1]과 사물인터넷 정보보호 로드맵[2]에서 사물인터넷 기반 조성을 위한 정보보호 인프라의 중요성을 강조하고 있다. 특히 사물인터넷 정보보호 인프라 강화를 위한 단계적 추진 전략의 하나로 사물



[그림 1] 블록 암호·복호화 과정

인터넷 기기, 네트워크, 서비스·플랫폼의 경량·저전력 특성을 고려한 암호기술의 개발 필요성을 제시하고 있다. 이는 사물인터넷 환경을 구성하는 기기의 상당수가 메모리 크기, 프로세서 성능, 소비전력 등의 자원이 제한된 특성을 가진다는 점을 반영한 것으로, 기존 PC 기반의 범용 환경에 적합하도록 설계된 암호기술들을 사물인터넷 환경에 그대로 적용하는 데 한계가 있음을 시사한다.

2000년대 중반 RFID/센서 네트워크 분야가 부각됨에 따라, 하드웨어 형태의 보안기기 탑재를 고려한 새로운 블록 암호의 개발 수요가 발생하였다. 이 시기에 구현 면적 최소화 측면에서 AES를 능가하는 경량 블록 암호로 개발된 사례는 ISO의 경량 블록 암호 표준[6]으로 제정된 PRESENT와 CLEFIA가 있다. 그러나 현재 사물인터넷 환경에서는 비용이나 관리 편의성 등을 고려하여 암호기술을 소프트웨어 모듈로 개발하는 것이 주로 고려되고 있다[2]. 소프트웨어 환경에서는 하드웨어 경량 블록 암호가 AES에

비해 속도나 코드 크기 측면에서 장점을 보여주지 못하고 있다. 게다가 경량화 또는 고속화를 위한 AES 최적 구현 기술 개발이 지속적으로 이루어지면서, AES를 능가하는 새로운 블록 암호를 개발하는 것이 쉽지 않게 되었다.

1.2 LEA의 개발과 표준화

LEA(Lightweight Encryption Algorithm)는 미래 창조과학부의 지원으로 2010년부터 3년간의 개발 과정을 거쳐 2012년 공개된 블록 암호이다. LEA는 하드웨어 환경을 대상으로 한 경량 블록 암호 설계 사례를 참고하여, 소프트웨어 환경에서 AES에 의해 설정된 성능 한계의 극복을 목표로 개발되었다. 특히 사물인터넷 기기의 경량·저전력 특성을 고려하여 구현 환경의 계산 자원을 효율적으로 사용하면 충분한 성능 제공을 가능하게 하는 것을 주요 개발 목표로 하였다. 이러한 고속 및 경량 특성을 동시에 충족시키기 위해서, LEA는 AES를 비롯한 기존

<표 1> LEA 규격

구분	블록 길이	키 길이	라운드 수(Nr)
LEA-128	16	16	24
LEA-192	16	24	28
LEA-256	16	32	32

대다수 블록 암호의 설계에 사용된 비선형 치환 테이블(S-box, Substitution box) 기반 구조와는 차별화된 설계 기술인 ARX(Addition, Rotation, XOR) 구조를 채택하였다. ARX 구조는 프로세서가 제공하는 기본 연산을 그대로 사용할 수 있기 때문에 연산 속도가 빠르고, 비선형 치환 테이블 저장에 필요한 메모리가 불필요하여 코드 크기나 구현 면적 대비 속도를 향상시킬 수 있는 장점을 가지고 있다. 반면 비선형 치환 테이블 기반 구조에 비해 안전성 확보를 위한 라운드 수의 증가가 불가피하고, 안전성을 분석하기 어렵다는 단점을 가지고 있다. LEA는 ARX 구조의 장점을 유지하면서 주요 블록 암호 공격 방법에 대한 정량적인 안전성 분석이 가능하며, 병렬 처리(SIMD, Single Instruction Multiple Data) 구현이 용이하도록 설계되었다.

LEA의 안전성 및 고속·경량 특성과는 별개로 사물인터넷 환경을 비롯한 다양한 암호제품 시장에서 LEA가 실제 활용되기 위해서는, 산업계의 제품 탑재를 위한 규격 문서의 체계적인 관리 및 접근 용이성 보장이 필요하였다. 이에 따라 2013년 LEA 규격의 국내 표준화 추진을 결정하고, TTA 정보보호기반 프로젝트 그룹(PG501)을 통해 표준화에 착수하였다. 이미 블록 암호 SEED, HIGHT가 TTA 표준 [4][5]으로 제정되어 있었기 때문에, PG501에서는 LEA가 가지는 차별성이 중점적으로 논의되었다. 논의 과정에서 기제정 표준 블록 암호와 차별된 LEA의 설계 사상, 안전성 및 성능에 대한 검토가 이루어

졌으며, 이와 관련한 국제 학회 발표[8]와 신뢰 기관의 객관적인 평가 결과[9]를 근거로 PG501 위원들의 합의를 통해 2013년 12월에 해당 표준화를 성공적으로 마칠 수 있었다.

2장과 3장에서는 LEA 규격 표준(TTAK.KO-12.0223)의 주요 내용 및 LEA의 안전성과 효율성에 대해 소개한다.

2. 표준의 주요 내용

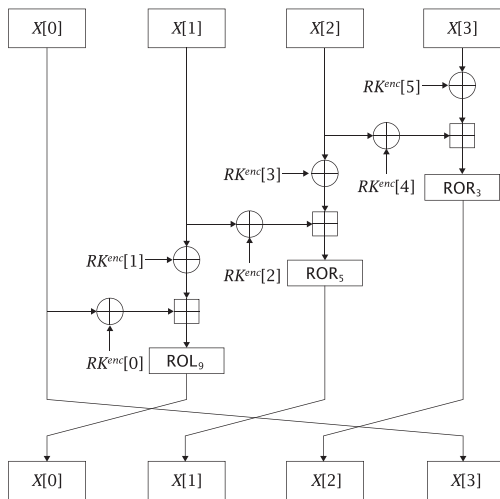
2.1 LEA의 특징

블록 암호 LEA는 128비트 데이터 블록을 암호화하는 알고리즘으로, 요구되는 안전성 기준에 따라 128, 192 또는 256비트 비밀키를 사용할 수 있다. LEA의 라운드 함수는 32비트 단위의 ARX 연산만으로 구성되어 있어, 이들 연산을 지원하는 32비트 소프트웨어 플랫폼에서 고속으로 동작한다. 또한 라운드 함수 내부의 ARX 연산 배치는 충분한 안전성을 보장하는 것과 동시에 비선형 치환 테이블 사용을 배제하여 경량 구현이 가능하도록 한다.

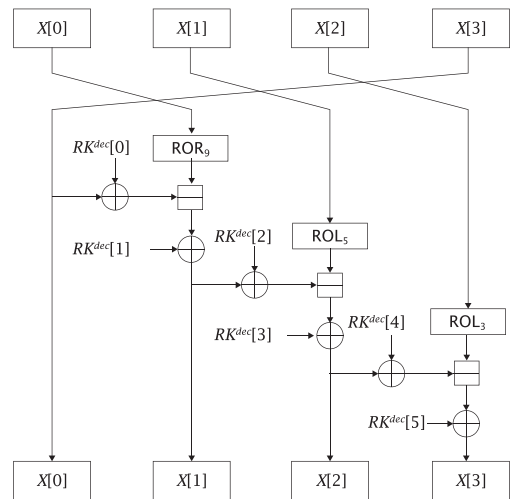
2.2 LEA 규격

2.2.1 전체 구조

LEA는 각 비밀키 길이에 따라 LEA-128, LEA-192, LEA-256으로 구분하며, LEA의 규격은 <표 1>과 같이 정리할 수 있다. <표 1>에서 블록과 키 길



[그림 2] 암호화 라운드 함수



[그림 3] 복호화 라운드 함수

이는 바이트 단위로 표기한다.

2.2.2 암·복호화 함수

[그림 1]에서 도시한 바와 같이 LEA는 암·복호화 과정에서 키 길이와 관계없이 동일한 라운드 함수를 반복적으로 수행한다. 이때 암호화 과정과 복호화 과정에 사용되는 라운드 함수의 구조는 각각 [그림 2], [그림 3]과 같다. 여기에서 라운드 함수의 입·출력 길이는 128비트이며, 라운드 키 RK의 길이는 192비트이다.

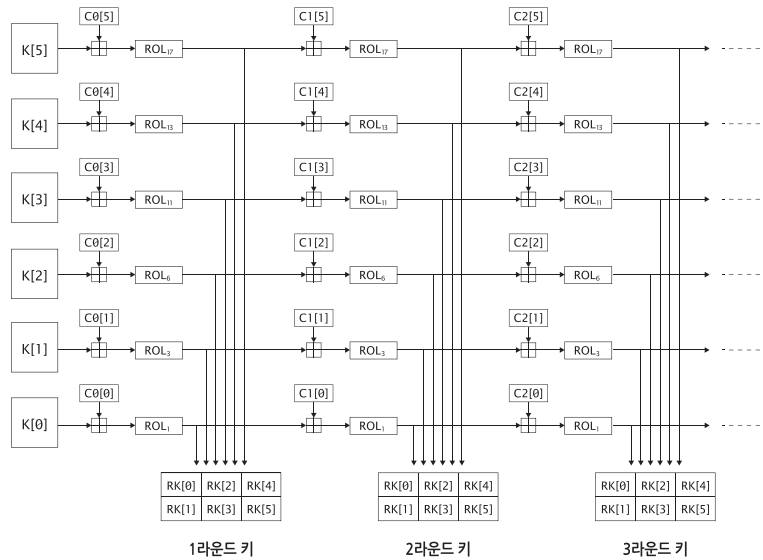
라운드 함수 입력값은 32비트 단위 배열 형태로 내부 연산에 사용된다. 라운드 함수를 구성하는 내부 연산으로 \boxplus 는 32비트 덧셈(Addition), \oplus 는 비트단위 배타적 논리합(XOR, eXclusive OR), $\text{ROLn}(\text{RORn})$ 은 n비트 좌측(우측) 회전(Rotation) 연산이다.

2.2.3 키 스케줄 함수

LEA-128, LEA-192, LEA-256을 구분하는 요소는 라운드 함수의 반복 횟수와 각 라운드에서 라운드

함수에 적용되는 라운드 키를 생성하는 방법인 키 스케줄에 있다. <표 1>에서 제시한 바와 같이 키 길이가 증가함에 따라 반복해야 하는 라운드 수가 증가하며, 이는 더 많은 라운드 키가 필요하게 됨을 의미한다. 비밀키의 길이는 다르지만 [그림 2], [그림 3]에서 보는 바와 같이 라운드 키의 길이는 192비트로 고정되어 있다. 따라서 LEA는 비밀키 길이에 따라 라운드 키를 생성하는 방법을 별도로 정의하고 있다. [그림 4]는 가장 직관적인 구조를 가진 LEA-192의 키 스케줄 함수를 도시한 것이며, 표준에는 LEA-128과 LEA-256의 키 스케줄 함수 또한 정의되어 있다.

라운드 함수와 동일하게, 키 스케줄 함수의 입력값은 32비트 단위 배열 형태로 내부 연산에 사용된다. 또한, 키 스케줄 함수는 규격에서 정의한 32비트 라운드 상수를 비트 회전 함수와 결합하여 사용하는 특징을 가진다.



[그림 4] LEA-192 키 스케줄 함수

3. LEA의 안전성과 효율성

3.1 안전성

LEA는 현재까지 알려진 모든 블록 암호 공격 방법에 대해 안전하다[8]. 특히 LEA는 AES와 다르게 키 스케줄 특성에 기인한 이론적인 취약성이 존재하지 않으며, ARX 구조의 특성을 이용하는 공격 방법에도 안전하다. 이는 AES를 설계한 기관에서 수행한 LEA의 안전성 평가를 통해서도 확인되었다[9]. <표 1>에 제시된 전체 라운드 수는 현재 알려진 모든 블록 암호 공격에 안전한 최소 라운드 수에 일정 비율의 안전성 마진을 더하여 결정된 것이다. 이러한 안전성 마진의 확보는 향후 등장할 수 있는 새로운 공격 방법에 대해서도 LEA의 안전성을 보장할 수 있다.

3.2 효율성

LEA는 범용 환경 및 사물인터넷 환경에 사용되는 프로세서에서 AES 대비 우수한 소프트웨어 구현

성능을 보여준다. 아래의 성능 비교는 128비트 키 길이를 기준으로 하며, 해당 키 길이의 AES를 AES-128로 표기한다. 스마트기기에서 사용되는 ARM 플랫폼을 대상으로 LEA-128의 속도 최적화와 코드 크기 최적화 구현 결과를 정리하면 각각 <표 2>, <표 3>과 같다. 각 표에 인용된 유사 성능의 ARM 플랫폼을 대상으로 한 AES-128의 측정값은 현재 학계에 알려진 가장 좋은 결과이다. <표 2>와 <표 3>은 LEA-128이 AES-128에 비해 1.7배 빠르게, 또는 1/4 크기의 코드로 구현 가능한 것을 보여준다. 또한 Intel, AMD의 PC/서버용 프로세서 등에서 측정된 결과를 통해, LEA-128은 AES-128 대비 약 1.5~2.7배 고속으로 동작하는 것을 확인할 수 있다[8].

LEA는 소프트웨어 구현에 최적화된 형태로 개발되었지만, 하드웨어 구현에 있어서도 우수한 성능을 보여준다. <표 4>는 하드웨어 구현 시 면적 대비 속도 최적화 결과를 정리한 것이다. AES-128의 구현은 LEA-128의 면적 최적화 구현 대비 절반 이하의 면적으로 가능하지만, 속도 측면을 동시에 고려할

<표 2> ARM 구현 효율성 비교: 속도 최적화

알고리즘	플랫폼	속도(cycles/byte)
LEA-128	ARM926EJ-S	20.06
AES-128	StrongARM SA-1110	34.00


<표 3> ARM 구현 효율성 비교: 코드 크기 최적화

알고리즘	플랫폼	코드 크기(bytes)	속도(cycles/byte)
LEA-128	ARM926EJ-S	622	326.94
AES-128	ARM7TDMI	2,468	460.50

<표 4> 하드웨어 면적 대비 속도 최적화 구현 효율성 비교

알고리즘	cycles/block	Throughput(kbps)	Tech.(μm)	Area(GE)	Throughput/Area
LEA-128	24	533.33	0.13	5,426	9.82
AES-128	226	56.64	0.13	2,400	2.35

경우 LEA-128이 AES-128보다 3배 이상 효율적인 것을 확인할 수 있다.

보호함수에 포함된 운영 모드 8종에 대한 LEA 운영 모드 규격이 2014년 TTA 표준(TTAK.KO-12.0246)으로 제정되었다. 

4. 맺음말

블록 암호 LEA는 범용 네트워크 및 사물인터넷 환경의 정보보호를 위한 핵심 기술로 활용되어 관련 암호 제품의 성능 향상 및 시장 경쟁력 확보에 기여할 수 있다. LEA 표준은 암호제품 개발에 있어 LEA 규격의 구현 가이드로 활용 가능하다. 참고로 LEA 표준은 2014년에 개최된 LEA 구현 경진대회의 출품작 상당수에서 구현 참조문서로 활용된 바 있다. 또한, 최근 LEA가 국내 암호모듈 검증제도(KCMVP)의 신규 검증대상 보호함수로 승인됨에 따라 LEA 표준은 2016년부터 시작되는 LEA 구현 검증에 활용될 예정이다.

참고로 LEA를 암호제품에 적용하기 위해서는 다중 블록 데이터 처리 방법을 정의한 운영 모드(Modes of Operation) 규격(LEA 운영 모드)이 별도로 필요하다. 이와 관련하여, 암호모듈 검증제도

[참고문헌]

- [1] 미래창조과학부, '사물인터넷 기본계획', 2014. 05. 08.
- [2] 미래창조과학부, '사물인터넷 정보보호 로드맵', 2014. 10. 31.
- [3] KS, '128비트 블록 암호 알고리즘 ARIA - 제1부: 일반 (KS X 1213-1:2009)', 2009. 12. 09.
- [4] TTA, '128비트 블록암호알고리즘 SEED(TTAS.KO-12.0004/R1)', 2005. 12. 21.
- [5] TTA, '64비트 블록암호 HIGHT(TTAK.KO-12.0040/R1)', 2008. 12. 19.
- [6] ISO, 'Information technology - Security techniques - Lightweight cryptography - Part 2: Block ciphers (ISO/IEC 29192-2)', 2012. 01. 15.
- [7] NIST, 'Advanced Encryption Standard(NIST FIPS-197)', 2001. 11. 26.
- [8] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K.H. Ryu, and D.-G. Lee, 'LEA: A 128-bit block cipher for fast encryption on common processors', Proc. of WISA 2013, LNCS, vol. 8269, 2014.
- [9] A. Bogdanov, N. Mouha, E. Tischhauser, D. Toz, K. Varici, V. Velichkov, M. Wang, Q. Wang, and V. Rijmen, 'Security Evaluation of the Block Cipher LEA Final Report', 2011. 07. 07.