CBMC를 이용한 AES 암호화 모듈의 정형 검증

안영정[0], 최진영

고려대학교 컴퓨터학과
{yjahn[0], choi}@formal.korea.ac.kr

Formal Verification of AES Encryption Module Using CBMC

Young-Jung Ahn[0], Jin-Young Choi

Dept of Computer Science & Engineering, Korea University

요약

정보보호 제품의 주요한 역할을 담당하는 암호 모듈의 구현 무결성을 보증하기 위해 많은 연구가 활발히 이루어지고 있다. 하지만 기존의 일반적인 테스팅 방법으로는 구현 무결성에 대해 신뢰하지 못한다. 본 논문에서는 NIST (the US National Institute of Science and Technology)에서 AES (Advanced Encryption Standard)로 제정된 Rijndael 블록암호 모듈을 Verilog로 구현하고 CBMC를 이용하여 새로운 방식의 구현 무결성 평가 방법을 제시하고자 한다.

1. 서론

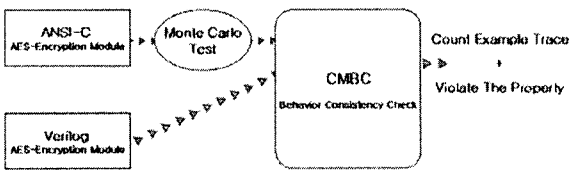
디지털 기술의 발전과 인터넷 이용의 확산으로 사회 전반에 정보화가 급진전되고 있다. 이와 더불어 정보시스템에 대한 해킹, 개인정보 유출에 따른 정보화 역기능이 커다란 사회 문제로 대두됨에 따라 정보보호에 대한 관심과 중요성이 크게 높아지고 있다. 따라서 신뢰성 있는 정보보호 기술 및 제품에 대한 요구가 높아지고 있고, 이러한 기술 및 제품을 평가하기 위해 암호 시스템 검증 기술의 필요성이 증대하고 있다.

보안 기술은 안전하고 신뢰할 수 있는 통신 환경 구축에 있어 반드시 필요한 기반 기술이다. 이러한 기술 하에 제작된 보안 시스템이 안전성과 신뢰성을 제공하지 못한다면, 보안 시스템이 제공하는 기밀성, 인증, 무결성, 부인봉쇄 서비스를 보장하지 못한다. 보안 시스템에 사용되는 기술은 DES, RSA, SEED, AES와 같은 여러 가지 암호 모듈을 기반으로 제공되고, 이러한 암호 모듈은 복잡하고 다양한 기능을 포함하고 있다. 따라서 구현에서 에러가 발생할 수 있고, 이러한 에러는 전체 보안 시스템에 치명적인 영향을 줄 수가 있다.

따라서 NIST (National Institute of Standard and Technology)에서는 CMVP (cryptographic Module Validation Program)[1], MOVS (Modes of Operation Validation System)[2], AESAVS (the Advanced Encryption Standard Algorithm Validation Suite)[3]를 통하여 개발된 암호 모듈의 구현 무결성을 평가한다[4]. CMVP, MOVS, AESAVS에서 제시한 검증 방법으로는 Known Answer Tests, Multi-block Message Test, Monte Carlo Test로 나뉜다. 이 중에 가장 탁월한 방법이 Monte Carlo Test이다. Monte Carlo Test는 의사 난수 데이터를 사용하여 추출한 입력값들이 주어지고, 해당 입력에 의한 출력값과 예상 출력값에 대한 비교로 평가한다. 그리고 이 방법은 소프트웨어 암호 모듈 개발에만 많이 사용되었다. 하드웨어 암호 모듈 검증 단계에서는 입력에 난수를 발생하고 그 결과를 비교하는 데 어렵기 때문이다. 하드웨어 암호 모듈에 대한 검증은 주어진 입력값에 대한 출력값과 예상 출력값의 비교하는 평가 방법인 Known Answer Tests을 사용한다. 하지만 Known Answer Tests는 한정된 입력값에 대한 결과만을 보기 때문에 평가하는 모듈이 해당 테스트를 통과하기 위한 악의적인 목적으로 제작된 경우 정확한 구현 적합성 평가가 어려운 단점이 있다.

이러한 문제점을 극복하기 위하여 본 논문에서는 ANSI-C 모듈과 Verilog 모듈의 행위 일치성을 정형 검증하는 도구인 CBMC (Bounded Model Checking for ANSI-C)[5,6]을 이용한다. CBMC의 입력인 ANSI-C 모듈은 Monte Carlo Test에 의해 검증된 ANSI-C AES-128[7,8] 모듈을 사용하여 개발하려는 하드웨어 AES-128 모듈에 대한 구현 과정에서 생길 수 있는 오류의 유무를 판단하는 방법을 제시하고자 한다.

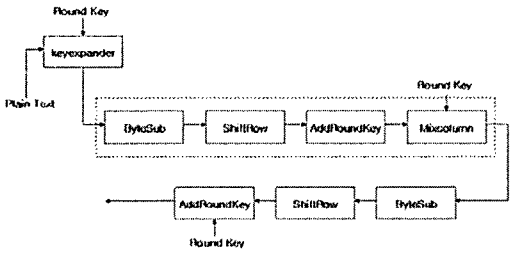
본 논문의 구성은 다음과 같다. 제 2장에서는 AES, 3장에서는 CBMC에 관해 소개하고 4장에서 AES의 구현과 실험을 통하여 일치성 검사를 한 후, 결론으로 및 향후 연구를 통하여 논문을 마무리 하고자 한다.



[그림 1] 하드웨어 암호 모듈 평가 방법

2. AES (Advanced Encryption Standard)

대부분의 대칭키 암호 시스템은 Feistel 구조의 라운드 변환을 기반으로 하는 반면에, AES 암호 모듈은 non-Feistel 구조의 라운드 변환을 수행하며, 3개의 역변환이 가능한 라운드로 구성된다. 블록의 길이는 128비트이고 키 길이는 128, 192, 256비트 중에서 선택할 수 있다. 라운드 수(Nr)는 키 길이(Nk)에 따라 10, 12, 14로 구성된다. AES 모듈의 변환 과정은 [그림 2]와 같으며 초기 라운드 키 가산 후에 Nr-1번의 반복 라운드를 수행한 후 Mixcolumn 변환이 제외된 최종 라운드 순으로 진행된다.



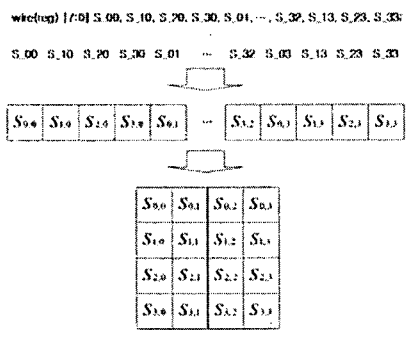
[그림 2] AES 모듈의 라운드의 라운드 변화

3. CBMC (Bounded Model Checking for ANSI-C)

CBMC는 ANSI-C 프로그램과 이것의 Verilog 구현을 CNF로 변환한 후, SAT solver인 Zchaff를 이용하여 동치성(Equivalence Checking)을 통해 두 CNF 사이의 행위 일치성 (Behavioral Consistency)을 체크하는 Bounded Model Checker이다.

4. AES-128 구현과 구현 무결성 검사

본 논문에서는 AES 표준안에 따라 블록 길이와 암호 키 길이가 128bit인 AES Encryption 모듈을 구현하였으며 단일 라운드 연산 회로를 사용하여 10번의 라운드 연산을 반복 수행한다. ByteSub 변환 블록 구현 시에는 S-Box를 8bit씩 256개를 사용하였고 ShiftRow 변환 블록은 별도의 연산 과정없이 배선만을 이용하여 구현하였다.

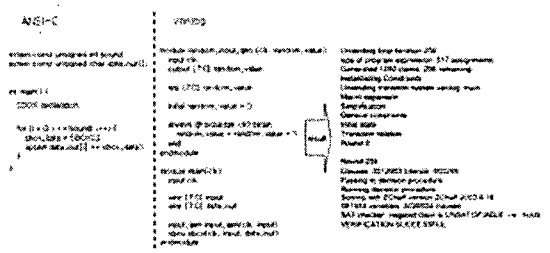


[그림 3] 128bit 평문, 키값, 암호문의 표현

MixColumn 변한 블록은 회로의 부피를 줄이기 위해서 AES 표준안에서 제시한 LogBox를 이용하지 않고 비트 연산만을 사용하여 구현하였다. 128비트의 평문, 키값 그리고 암호문의 표현은 [그림 3]과 같이 8bit 길이의 변수를 16개 나열하여 선언하였다. CBMC에서 ANSI-C 모듈과 Verilog 모듈의 행위 일치성 검사할 때, 8bit 혹은 16비트의 길이를 가진 변수에 대한 일치성 검사를 하기 때문이다. 이와 같은 설계 원칙에 따라 Verilog로 구현된 AES-128-Encryption 모듈을 다음과 같은 속성으로 CBMC 통해 Monte Carlo Test에 의해 검증된 ANSI-C AES-128 모듈과의 일치성 검사를 했다.

4.1 Sbox 입출력 과정 검증

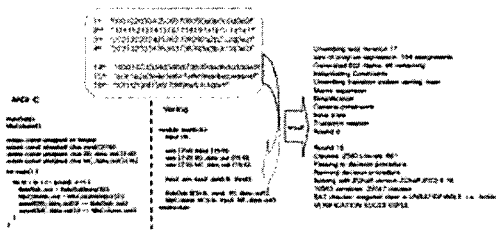
Sbox는 바이트값들의 GF(2^8)에서 곱셈의 역원을 취하는 $x \rightarrow x^{-1}$ 매핑과 GF(2) 상에서 affine 변환을 수행한 것이다. 그러므로 Sbox에 모든 가능한 입력값(0~255)을 대입함으로써 Sbox를 이루는 각 값들이 입력에 의해 정확하게 계산 되는지를 검증한다. 즉, 0~255 범위의 입력값에 의해 Sbox의 모든 값들을 확인할 수 있게 검증한다. CBMC에서 Sbox를 검증하기 위해 0~255 범위의 입력값을 대입하여 256번의 bound consistency check로 검증한다.



[그림 4] Sbox 입출력 과정 검증

4.2 ByteSub 변환 동작과 MixColumn 변환 동작 검증

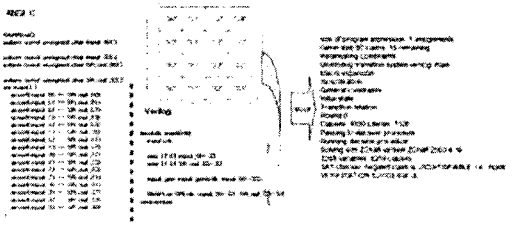
ByteSub 변환과 MixColumn 변환은 각각의 8bit 블록들이 독립적인 비선형 치환을 수행한다. 그러므로 16개의 8bit블록들이 Sbox, Mixcolumns를 호출하여 정확한 치환이 이루어지는지 확인한다. 두 변환은 128bit의 입력을 8bit 블록 단위로 연산함으로 각 바이트가 가질 수 있는 모든 입력값들("00"~"ff")을 16번으로 나누어 대입한다. [그림 5]은 CBMC를 이용하여 16번의 bound consistency check로 검증한 결과이다.



[그림 5] ByteSub 변환 및 MixColumn 변환 동작 검증

4.3 ShiftRow 변환 동작 검증

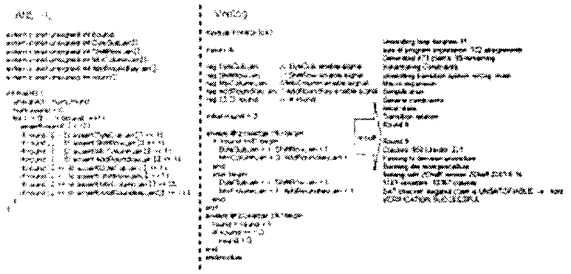
ShiftRow 변환은 상태의 값들을 변환시키지 않고 상태를 구성하는 바이트들의 위치를 교환한다. 그러므로 한번의 bound consistency check로 각 상태들의 정확한 교환을 확인한다.



[그림 6] ShiftRow 변환 동작 검증

4.4 Kontrol 모듈 동작 검증

AES는 10번의 라운드로 수행된다. 10번의 라운드는 마지막 라운드를 제외하고 ByteSub, ShiftRow, MixColumn, AddRoundKey 변환을 거친 후 마지막 라운드의 경우에만 ByteSub, ShiftRow, AddRoundKey을 수행한다. ByteSub, ShiftRow, MixColumn, AddRoundKey 변환은 각각 enable 신호가 '1'이 되면 수행한다. Kontrol 모듈은 0~8 라운드에는 ByteSub, ShiftRow, MixColumn, AddRoundKey 변환의 enable 신호를 '1'로 셋팅하고, 9 라운드에는 MixColumn 변환의 enable 신호를 '0'로 ByteSub, ShiftRow, AddRoundKey 변환의 enable 신호를 '1'로 셋팅한다. 그러므로 Kontrol 모듈 동작 검증에서는 각 라운드마다 enable 신호를 정확히 셋팅하는지 알아보고 라운드를 나타내는 변수가 10을 넘지 않는지 20 round consistency check로 검증한다.



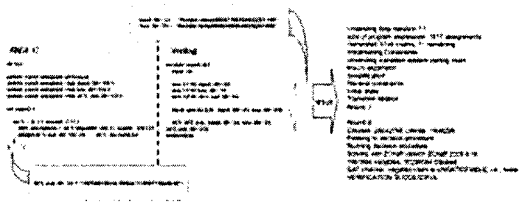
[그림 7] Kontrol 모듈 변환 동작 검증

4.5 AES encryption 모듈 검증

AES encryption 모듈에서 사용되는 모든 연산에 대해 검증하였다. 마지막으로 [그림 8]과 같이 NIST에서 인증한 ANSI-C AES-128 모듈과 Verilog AES-128 모듈에 대한 consistency check을 보였다.

5. 결론 및 향후 연구 방향

설계된 AES-128 모듈은 Monte Carlo Test에 의해 검증된 ANSI-C AES-128 모듈과 일치함을 알 수 있다.



[그림 8] AES encryption 모듈 검증

본 논문에서 제시된 방법에 따라 구현된 하드웨어 암호 모듈이 AES 표준안에 명시된 테스트 벡터를 이용하여 검증한 결과 모든 논리기능이 정상적으로 동작함을 확인하였다. 본 논문에서 사용된 CBMC는 일치성 검사할 시스템을 Propositional logic으로 변환 후 Zchaff라는 SAT 도구를 통하여 점증하기 때문에 100만개 이상의 boolean 변수를 가진 큰 시스템도 검증 할 수 있다는 장점이 있다. 하지만 CBMC의 시스템을 초기 상태에서 각 스텝마다 가지는 전위로 변환되기 때문에 검사할 수 있는 시스템의 행위가 제한적이라는 것이 가장 큰 단점이다. 본 논문에서는 하드웨어 암호 모듈은 제한적인 라운드 내에서 동작을 하기 때문에 Bounded Model Checking 도구인 CBMC의 단점이 검증을 할 때 영향을 미치지 않음을 보였다. 제안된 검증 방법론에 따라 구현된 하드웨어 AES-128 모듈은 Known Answer Tests 단점을 보완했다. 본 논문에서 제안된 하드웨어 암호 모듈의 구현 무결성 평가 방법에 대한 실험으로 하드웨어 암호 모듈의 신뢰성과 안전성을 높였다. 향후 연구 방법으로는 AES 암호화 모듈이 가져야할 속성에 대한 동치성 검사를 통해 암호 시스템의 안전성과 신뢰성을 높이고자 한다. 또한 무선 인터넷 분야에서 가장 많이 사용되는 타원 곡선 암호 모듈에도 본 논문에서 제안한 일치성 검사방법을 적용고자 한다.

참고문헌

[1] FIP Publication 140-2, Security Requirements for Cryptographic Modiles, 2001

[2] Sharon Keller and Miles Smid, NIST Special Publication 800-17, Modes of Operation Validation System (MOVS) : Requirements and Procedures, 1998

[3] Lawrence E., Bassham Encryption Standard Algorithm Validation Suite (AESAVS), 2002

[4] Cryptographic Standards and Validation Programs at NIST, http://csrc.nist.gov/cryptval/

[5] Edmund Clarke, Daniel Kroening, Karen Yorav, Behavioral Consistency of C and Verilog Programs Using Bounded Model Checking, 2003

[6] Edmund Clarke, Daniel Kroening, ANSI-C Bounded model Checker User Manual, 2003

[7] Brian Gladman, A Specification for Rijndael, the Algorithm, 2003

[8] Federal Information Processing Standards Publication 197, Announcing the ADVANCED ENCRYPTION STANDARD (AES), 2001