논문 2010-47SD-2-10

스캔 설계된 AES 코아의 효과적인 비밀 키 보호 기술

(An Efficient Secrete Key Protection Technique of Scan-designed AES Core)

송 재 훈*, 정 태 진**, 정 혜 란**, 김 화 영***, 박 성 주****

(Jaehoon Song, Taejin Jung, Hyeran Jeong, Hwayoung Kim, and Sungju Park)

요 약

본 논문은 Advanced Encryption Standard(AES) 암호화 코아가 내장된 System-on-a-Chip(SoC)의 스캔 기반 사이드 채널 공격에 의해 발생될 수 있는 비밀 키 정보 누출 방지를 위한 효과적인 스캔 설계 기술을 제안한다. 본 논문에서 제안하는 시큐어 스캔 설계 기술은 IEEE 1149.1의 명령어 방식을 사용하여 거짓 키를 이용한 테스트를 진행한다. 또한 어플리케이션에 최적화 되어있는 암호화 IP 코아를 수정하지 않고 적용을 할 수 있다. SoC 상의 IEEE 1149.1 제어기 표준을 유지하며 기존 방식에 비해 낮은 면적 오버헤드 및 전력 소모량을 갖는 기술을 제안한다.

Abstract

This paper presents an efficient secure scan design technique which is based on a fake key and IEEE 1149.1 instruction to protect secret key from scan-based side channel attack for an Advanced Encryption Standard (AES) core embedded on an System-on-a-Chip (SoC). Our proposed secure scan design technique can be applied to crypto IP core which is optimized for applications without the IP core modification. The IEEE 1149.1 standard is kept, and low area, low power consumption, very robust secret-key protection and high fault coverage can be achieved compared to the existing methods.

Keywords: AES, key protection, scan design, SoC.

I. 서 론

Advanced Encryption Standard(AES) 암호화 알고리즘은 스마트 카드와 Zigbee, Bluetooth와 같은 통신용 칩이나 방송용 수신 칩 등 보안이 필요한 많은 분야에서 활발히 사용되고 있으며, IP 코아 설계물로 구현되어 정보의 암호화를 필요로 하는 System-on-a-Chip (SoC) 내부에서 많이 사용 되고 있다[1]. 이러한 AES와 같은 암호화 코아에서 사용하는 사용자 비밀 키 정보가 누출된다면 매우 큰 경제적 손실 및 사회적 문제가 야기될 수 있다. 따라서 AES 암호화 코아에 사용되는 사용자 비밀 키 누출을 방지하는 것은 매우 중요한 문제이다.

SoC의 양산 테스트 시 테스트 비용을 절감하고 고장 검출율을 효과적으로 높이기 위해 스캔 설계 기반 테스트가 널리 사용되고 있다[2]. 스캔 기반 설계는 SoC 내부 데이터의 제어와 관측을 가능하게 하는 구조적 환경을 제공한다[3~4]. 이를 통하여 테스트 엔지니어는 높은 테스트 제어도 및 관측도를 얻을 수 있으며 이는 높은 고장 검출율로 이어진다. 따라서 칩의 출하 이전에 불량품들을 선별하기 위한 단계로 일반적으로 스캔 테스

* 정회원, 트란소노
(Transono)
** 정회원, 씨앤애스테크놀로지
(C&S Technology)
*** 학생회원, 한양대학교 컴퓨터공학과
(Department of Computer Science & Engineering, Hanyang Universiry)
**** 평생회원, 한양대학교 전자컴퓨터공학부
(Department of Computer Science & Engineering, Hanyang Universiry)
접수일자: 2009년3월23일, 수정완료일:2010년1월25일

트의 과정을 거치게 된다. 또한 스캔 설계 기반 테스트는 최종 소비자의 사용 중 고장을 진단하기 위해서 쓰이기도 한다.

그런데 만약 AES IP 코아를 내장한 SoC에 일반적인 스캔 설계 기반 테스트 방법을 적용하게 되면, SoC 내부의 사용자 비밀 키와 관련된 정보를 스캔 체인을 통해 관측하는 것이 가능해진다[5~6]. 따라서 테스트 단계에서는 효과적인 테스트 수단으로써 고장 검출율을 높이면서도, 출하 후에 사용자 비밀 키와 관련된 정보의 유출을 방지할 수 있는 메커니즘이 필요하다[7].

본 논문에서 제안하는 기술은 스캔 설계 기반 테스트의 높은 고장 검출율을 유지하면서도 매우 낮은 사용자 키 누출 가능성을 가지며, 기존방법[8~10]에 비해 적은 면적 및 전력소모 오버헤드를 갖는다. 또한 최적화되어 설계된 AES 코아 내부를 수정하지 않고도 적용 가능하고, IEEE 1149.1 표준(이하 JTAG이라 함) 테스트 제어기와도 호환성을 유지한다. Ⅱ장에서는 AES 코아의 키 노출을 방지하기 위해 기존에 제시된 스캔 테스트 방법을 소개한다. Ⅲ장에서는 본 논문에서 제안하는 기술을 소개하며 Ⅳ장에서 실험 결과를 보여준다. 그리고 마지막으로 Ⅴ장에서 결론을 맺는다.

Ⅱ. AES 키 정보 누출 방지를 위한 기존 스캔 테스트 설계 기술

AES는 2001년 미국 National Institute of Standards and Technology(NIST)로부터 암호화 표준으로 선정되었다. AES는 하드웨어나 소프트웨어적으로 구현했을 때 모두 좋은 성능을 보인다. 또한 구현이 쉽고 메모리를 적게 사용하는 장점을 가지고 있어, 현재 세계적으로 많은 곳에서 하드웨어로 구현되어 사용되고 있다[1].

AES는 128비트의 데이터와 128, 192 또는 256비트의 사용자 키로 암호화가 이뤄지며 키의 길이에 따라 라운드 횟수는 각각 10, 12, 14회로 정해진다. AES 암호화 과정은 그림 1과 같이 한 번의 프리 라운드와 사용자 키의 길이에 따라 정해진 횟수만큼의 라운드의 반복으로 이루어져 있다. 프리 라운드에서는 사용자 비밀 키와 암호화 하고자 하는 평문을 사용하여 그림 1상의 KeyXOR 절차를 거친 뒤 데이터 a를 출력한다. 그 후 각 라운드에서는 사용자 비밀 키로부터 확장된 라운드 키와 데이터 a를 이용하여 SubByte, ShiftRow, MixColumn, KeyXOR 절차를 거친 뒤 데이터 e를 출력

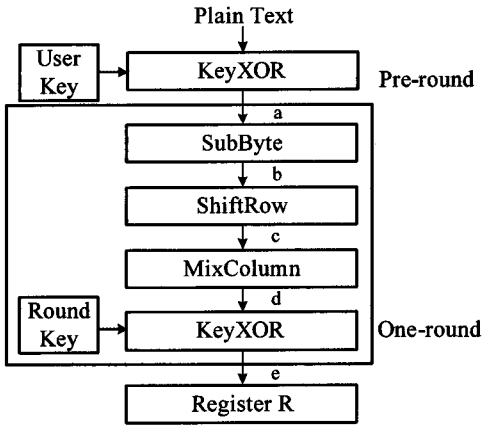

그림 1. AES 암호화 과정
Fig. 1. Process of AES encryption.

한다. 그리고 데이터 e는 각 라운드를 마칠 때마다 Register R에 저장된다.

그런데 이러한 AES 암호화 코아에 일반적인 스캔 기반 설계를 적용하게 되면, Register R과 같은 코아 내부의 레지스터들은 스캔 테스트를 위한 스캔 체인에 속하게 된다. 따라서 공격자는 암호화 과정 중에 사용자 비밀 키가 적용된 데이터를 스캔 출력포트를 통해 관측할 수 있으며, 이렇게 얻은 데이터를 이용하여 사용자 비밀 키를 유추해 내는 일이 가능하다[5~7].

[8]에서는 스캔 테스트 시 발생할 수 있는 AES 사용자 비밀 키의 누출을 방지하기위해 길이가 N인 서로 다른 M 개의 매칭 키를 정의하였다. 그리고 이 M 개의 매칭 키를 정해진 순서대로 인가해야만 스캔 출력포트가 활성화되도록 하였다. 이 매칭 키의 확인은 임의의 N 개의 스캔 플립플롭의 출력을 키 매칭 블록에 인가하여 매칭 키와 비교함으로써 이루어진다.

하지만 이는 스캔 플립플롭의 출력 팬아웃을 증가시켜 AES 코아에 라우팅 오버헤드 및 추가적인 딜레이를 가져오며, 이미 구현되어 있는 AES 코아를 수정해야하기 때문에 코아의 재사용을 어렵게 한다는 단점이 있다. 또한 스캔 테스트를 수행하기 전에 매칭 키 값을 인가해야하기 때문에 테스트 시퀀스를 복잡하게 만들고, 만약 테스트엔지니어에 의해 패턴 매칭 키 값이 누출된다면 공격자는 이를 이용하여 스캔 출력포트를 활성화시킨 뒤 AES의 내부 정보를 볼 수 있게 된다.

[9]에서는 JTAG 표준 Test Access Port(TAP) 제어기의 스테이트 머신에 새로운 스테이트를 추가하여 AES 코아를 정상 동작 모드에서 스캔 테스트 모드로 전환하고자 할 때, 반드시 파워-오프 동작을 거치도록

하였다. 정상 동작 시 남아있던 암호화 과정의 중간 값들을 스캔 테스트 전에 지움으로써 비밀 키가 적용된 데이터가 스캔 출력포트를 통해 유출되는 것을 방지한 것이다. 하지만 이 방법에는 JTAG 표준 TAP 제어기와 호환이 되지 않는 문제가 존재한다. [9]에서는 JTAG 표준 TAP 제어기에 새로운 스테이트의 추가와 더불어 AES 코아 내부에 Mirror Key Register(MKR)를 추가하고, 이 MKR에 정상 동작 시에는 사용자 키를 로드하고 스캔 테스트 동작 시에는 사용자 키가 아닌 테스트를 위한 미러 키를 설정하여 사용하는 방법을 제안하였다. 따라서 테스트 엔지니어는 스캔 테스트 시에 사용자 키를 대신하여 스캔입력 포트를 통해 MKR에 설정한 미러 키로 AES 코아를 테스트하는 것이 가능하다. 그러나 이 방법은 AES 코아 내부에 MKR을 삽입하여 면적 및 전력소모량을 증가시킨다. 또한 MKR을 AES 내부에 구현하는 경우 이미 최적화 되어있는 AES 코아의 수정을 요구하기 때문에 코아의 재사용 측면에서 불리한 단점을 가지고 있다.

[10]에서는 스캔 테스트 시 암호화 과정의 값이 누출되는 것을 막기 위하여 스캔 체인의 경로 상의 임의의 위치에 임의의 개수의 인버터를 삽입하는 기술을 제안하였다. 또한 차분해독법과 선형해독법을 이용한 공격을 방지하기 위해서, 인버터가 삽입된 스캔 체인의 출력 데이터가 m x m(m = 입, 출력 비트의 크기)의 구조를 갖는 S-box를 통해 치환된 후 스캔 체인의 최종 출력단을 통해 나가도록 하였다.

[10]의 방법을 적용할 경우에도 역시 최적화 되어있는 AES 코아를 수정해야하기 때문에 코아의 재사용 측면에서 단점을 가지고 있다. 또한 테스트 패턴 생성을 위한 자동 테스트 패턴 생성기(ATPG)는 S-box를 인식하지 못하기 때문에, 디자이너는 AES의 정상적인 스캔 테스트 데이터의 출력 값을 S-box를 통해 치환된 값으로 바꾸는 추가적인 작업을 해야만 한다. 그리고 디자이너에 의해 구현된 S-box의 구조가 유출된다면 공격자는 [11]의 방법을 이용하여 S-box를 재생성할 수 있고, 재생성한 S-box를 이용하여 그림 2에 표시된 치환 전의 인버터가 삽입된 스캔 체인의 출력 데이터인 Scan Data 1(SD1)의 값을 알아낼 수 있다. 그리고 그 후에는 다음과 같은 절차로 사용자 키 값을 복구하는 것이 가능하다.

2.1. 누출된 S-box 구조를 이용한 SD1 추출

먼저 그림 2와 같이 디자이너가 AES 코아에 m개의

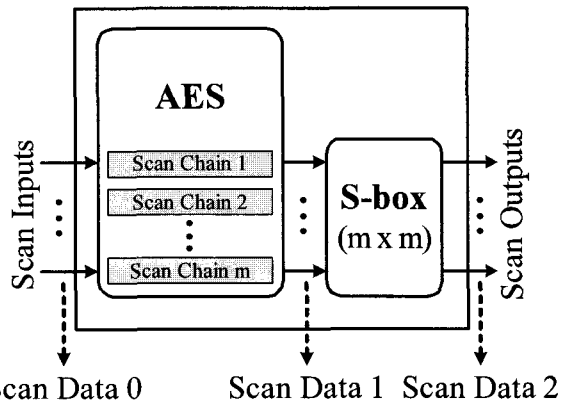

그림 2. Inverter 와 S-box를 이용한 AES 키 누출 방지 기법

Fig. 2. Key protection technique by using inverter and S-box.

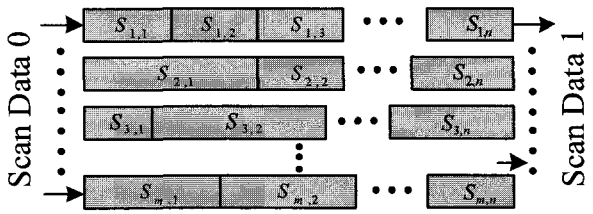

그림 3. AES 코어의 스캔 체인 구조

Fig. 3. Scan chain architecture of AES core.

스캔 체인과 m x m 구조의 S-box를 구성하였다고 가정한다. 그런데 이 AES 코아에 포함된 S-box의 구조가 유출되어 공격자가 이를 입수한다면, 스캔 출력단을 통해서 관측한 SD2 값과 입수한 S-box의 구조를 이용하여 AES 코아의 스캔 체인 출력 값이자 S-box의 입력 값인 SD1 값을 찾아낼 수 있게 된다. 그런 다음에 이렇게 알아낸 SD1 값에 [9]에서 제시한 방법을 적용하면 사용자 키 값을 유추해내는 것이 가능하다.

2.2. [9]의 방법을 이용한 데이터 레지스터 위치 탐색

공격자는 노출된 S-box 구조를 이용하여 SD1 값을 얻은 후, 다음과 같이 [9]에 제시된 절차를 통해 사용자 키 값을 유추할 수 있게 된다. 그림 2와 같이 [10]의 방법을 적용한 AES 코아의 스캔 체인 구조는 그림 3과 같이 나타내어질 수 있다. 그림 3에서 m은 AES 코아에 일반적인 스캔 기반 설계를 적용하였을 때의 스캔 체인의 개수이고, n은 [10]의 방법을 적용할 경우 각각의 스캔 체인에 삽입되는 인버터의 개수이다. 그리고 인버터가 삽입되는 위치를 기준으로 스캔 체인을 분할하였을 때, 이웃한 스캔 플립플롭들의 집합을

$S_{i,j}(i:1 \le i \le m, j:1 \le j \le n+1)$라 하면, 인버터가 삽입되지 않은 상태에서의 각각의 스캔 체인의 출력값 T_i는 식 (1)과 같이 나타낼 수 있다.

$$\begin{aligned} T_1 &= \{S_{1,1}, S_{1,2}, S_{1,3}, \cdots, S_{1,n}, S_{1,n+1}\} \\ T_2 &= \{S_{2,1}, S_{2,2}, S_{2,3}, \cdots, S_{2,n}, S_{2,n+1}\} \\ T_3 &= \{S_{3,1}, S_{3,2}, S_{3,3}, \cdots, S_{3,n}, S_{3,n+1}\} \\ T_m &= \{S_{m,1}, S_{m,2}, S_{m,3}, \cdots, S_{m,n}, S_{m,n+1}\} \end{aligned} \quad (1)$$

여기에 [10]의 방법을 적용하여 스캔 체인에 인버터를 추가할 경우, 스캔 플립플롭의 값 $S_{i,j}$는 스캔 체인의 출력단까지 전달되어지는 과정에서 경로 상에 삽입된 인버터를 지날 때마다 그 값이 반전된다. 이렇게 변경된 $S_{i,j}$ 값을 $S_{i,j} \times inv(z)(z = S_{i,j}$가 스캔 출력단까지 전달되어지는 과정에서 거치게 되는 인버터의 수, $0 \le z \le n)$라 하면, 각각의 스캔 체인의 출력 값 $Tinv_i$는 식 (2)와 같이 표현된다.

$$\begin{aligned} Tinv_1 &= \begin{Bmatrix} S_{1,1} \times inv(n), S_{1,2} \times inv(n-1), \\ S_{1,3} \times inv(n-2), \cdots, S_{1,n} \times inv(1), S_{1,n+1} \end{Bmatrix} \\ Tinv_2 &= \begin{Bmatrix} S_{2,1} \times inv(n), S_{2,2} \times inv(n-1), \\ S_{2,3} \times inv(n-2), \cdots, S_{2,n} \times inv(1), S_{2,n+1} \end{Bmatrix} \\ Tinv_3 &= \begin{Bmatrix} S_{3,1} \times inv(n), S_{3,2} \times inv(n-1), \\ S_{3,3} \times inv(n-2), \cdots, S_{3,n} \times inv(1), S_{3,n+1} \end{Bmatrix} \\ Tinv_m &= \begin{Bmatrix} S_{m,1} \times inv(n), S_{m,2} \times inv(n-1), \\ S_{m,3} \times inv(n-2), \cdots, S_{m,n} \times inv(1), S_{m,n+1} \end{Bmatrix} \end{aligned} \quad (2)$$

또한 SD1은 다음과 같이 m 개의 스캔 체인 출력 값의 집합으로 나타내어질 수 있다.

$$SD1 = \{Tinv_1, Tinv_2, Tinv_3, \cdots, Tinv_m\} \quad (3)$$

공격자는 이 *SD1* 값을 이용하여 사용자 키 값을 유추할 수 있다. 먼저 [9]의 방법과 같이 서로 1비트만 다른 두 개의 평문 a, b를 AES 코아에 입력하고 한 클락 주기 동안 AES 코아를 정상 동작하도록 한다. 그리고 변경된 데이터를 스캔 출력포트를 통해 추출한다. 인버터를 삽입하지 않았을 경우에 두 개의 평문 a, b에 대한 스캔 체인의 출력 값이 각각 $\{T_1, T_2, T_3, \cdots, T_m\}_a$, $\{T_1, T_2, T_3, \cdots, T_m\}_b$ 이었다고 하면, 이 두 데이터의 XOR 값은 식 (4)와 같다.

$$\begin{aligned} &\{T_1, T_2, T_3, \cdots, T_m\}_a \oplus \{T_1, T_2, T_3, \cdots, T_m\}_b \\ &= \begin{Bmatrix} \{S_{i,j}\}_a \oplus \{S_{i,j}\}_b, \\ 1 \le i \le m, 1 \le j \le n+1 \end{Bmatrix} \end{aligned} \quad (4)$$

그리고 스캔 체인에 인버터를 삽입했을 경우에 스캔 체인을 통해 출력되는 값은 $\{Tinv_1, Tinv_2, Tinv_3, \cdots, Tinv_m\}_a$, $\{Tinv_1, Tinv_2, Tinv_3, \cdots, Tinv_m\}_b$ 이 될 것이다. 이 두 출력 값의 XOR 값은 다음과 같다.

$$\begin{aligned} &\{Tinv_1, Tinv_2, Tinv_3, \cdots, Tinv_m\}_a \oplus \\ &\{Tinv_1, Tinv_2, Tinv_3, \cdots, Tinv_m\}_b \\ &= \begin{Bmatrix} \{S_{i,j} \times inv(z)\}_a \oplus \{S_{i,j} \times inv(z)\}_b, \\ 1 \le i \le m, 1 \le j \le n+1, 0 \le z \le n \end{Bmatrix} \end{aligned} \quad (5)$$

그런데 이 경우, 스캔 체인에 삽입된 인버터의 위치와 개수는 변하지 않으므로 두 출력 값의 모든 비트들은 같은 횟수만큼 인버터를 거쳐 반전된 값이다. 따라서 식 (5)에서의 XOR 연산은 인버터로 인한 값의 변화를 상쇄하여 식 (6)과 같은 결과를 가져온다.

$$\begin{aligned} &\{S_{i,j} \times inv(z)\}_a \oplus \{S_{i,j} \times inv(z)\}_b \\ &= \begin{Bmatrix} \{S_{i,j}\}_a \oplus \{S_{i,j}\}_b, \\ 1 \le i \le m, 1 \le j \le n+1, 0 \le z \le n \end{Bmatrix} \end{aligned} \quad (6)$$

결국 [10]에서 제안한 방법은 설계자에 의해 S-box의 구조가 누출된다면 AES 코아 내부의 스캔 체인에 인버터를 추가하지 않았을 경우와 마찬가지로 [9]에서 제시된 스캔 기반 사이드 채널 공격에 취약하다.

III. 제안하는 시큐어 스캔 설계기술

그림 4는 스캔 기반 사이드 채널 공격에 의한 AES 사용자 키 정보 누출을 방지하기위해 본 논문에서 제안하는 시큐어 스캔 설계 기술이 적용된 AES 코아의 구조이다. 제안하는 시큐어 스캔 설계 기술은 JTAG에 새로운 TAP 제어 명령어를 추가하여, AES 코아를 정상 동작 모드에서 테스트 모드로 전환할 때 이 명령어를 인가하도록 하였다. 명령어가 인가되면 코아 내부에 저장되어 있던 사용자 비밀 키가 적용된 데이터들은 거짓 키가 적용된 데이터로 변경된다. 그리고 스캔 테스트 시에는 이 거짓 키 값을 이용하여 테스트를 진행함으로써 사용자 키가 누출되는 것을 방지한다.

본 논문에서 제안하는 사용자 키 누출 방지 기술은 그림 4에 보이는 바와 같이 AES 코아 내부 회로에 추가적인 수정 없이 적용 가능하다. 따라서 어플리케이션에 최적화 되어있는 코아를 수정하지 않아도 되기 때문에 코아의 재사용성을 높일 수 있다. 다음은 본 논문에서 제안하는 기술을 두 서브섹션으로 나누어 제시한다.

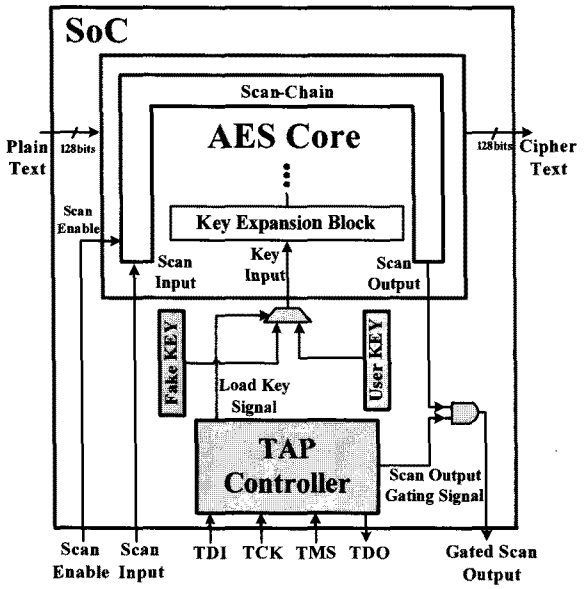

그림 4. AES 키 누출 방지를 위한 시큐어 스캔 구조
Fig. 4. Secure scan architecture for key protection.

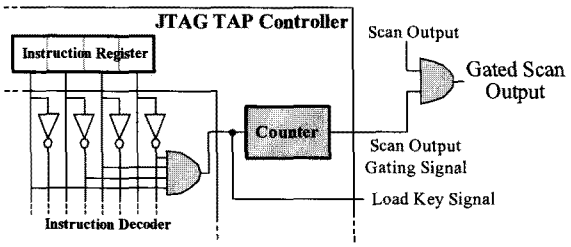

그림 5. 수정된 JTAG TAP Controller
Fig. 5. Modified JTAG TAP Controller.

3.1. AES 정상 동작 모드에서 스캔 테스트 모드로 전환 시 AES 내부 레지스터 정보 변경 기술

AES 코아를 정상 동작 모드에서 스캔 테스트 모드로 변경할 때 코아 내부에 남아있는 데이터가 스캔 체인 출력단에서 관측 가능할 경우, 이는 공격자의 표적이 된다. 따라서 정상 동작 모드에서 테스트 모드로 변경하기 전에 사용자 키 값이 적용된 내부 레지스터 값들을 변경하는 기술이 필요하다. 이를 위해 본 논문에서는 JTAG에 스캔 테스트를 위한 새로운 TAP 제어 명령어를 추가하여, 명령어 기반 방법으로 AES 코아의 모드를 변경하도록 하였다. [9]에서 JTAG의 TAP 스테이트에 새로운 스테이트를 추가하여 정상 동작 모드에서 테스트 모드로 변경 시 파워-오프 동작을 거치도록 구현한 것과 달리 이 방법은 명령어만을 추가할 뿐이므로 JTAG 표준과의 호환성에 문제가 없다. 또한 AES 코아의 외부에서 거짓 키 또는 사용자 키가 선택되어

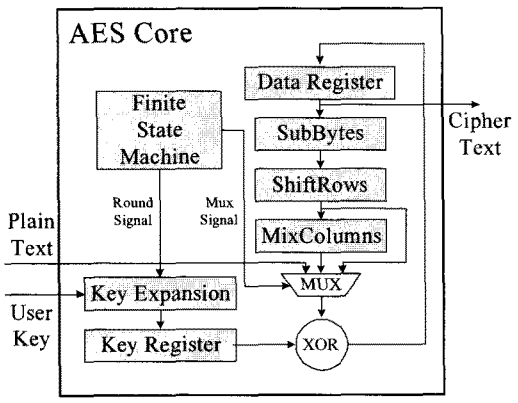

그림 6. AES 암호화 코아 설계 구조
Fig. 6. AES core architecture.

들어가기 때문에 기존에 제시되었던 방법들과는 달리 코아 내부를 수정하지 않아도 적용이 가능하다.

제안하는 방식의 구체적인 내용은 다음과 같다. 먼저 AES 코아를 정상 동작 모드에서 테스트 모드로 전환하기 위해서 스캔 테스트 명령어를 JTAG에 인가한다. JTAG 명령어가 인가되면 그림 4의 Fake KEY와 같이 사용자 키와 다른 값으로 고정되어 있는 거짓 키 값이 AES 코아에 키 인풋으로 주어진다. 그림 4에서 TAP 제어기 내부의 인스트럭션 디코더로부터 보내어지는 Load Key 신호가 이러한 제어를 위해 사용된다. 거짓 키 값이 AES 코아에 인가되면, 정상동작 시 AES 코아 내부 레지스터에 남아있던 사용자 키가 적용된 데이터가 거짓 키가 적용된 데이터로 변경될 때까지 스캔 체인 상의 값은 스캔 출력포트를 통해 볼 수 없어야 한다. 따라서 AES 내부 레지스터 값이 모두 변경될 때까지 TAP 제어기에서는 그림 4와 같이 Scan Output Gating 신호를 사용하여 스캔 출력포트를 비활성화 시킨다. 그림 5는 Load Key 및 Scan Output Gating 신호 생성을 위한 TAP 제어기 내부회로이며 회색으로 표시된 부분이 이를 위해 추가된 로직이다. 스캔 테스트 모드 전환 시 Load Key 신호에 의해 활성화/비활성화되는 카운터는 AES 내부 레지스터 값이 거짓 키가 적용된 데이터로 완전히 변경될 때까지 스캔 출력 포트을 차단하고, 그 이후에 스캔 출력포트를 활성화 시킨다. AES 코아가 정상 동작 모드일 때에는 Scan Output Gating 신호 값이 0으로 유지되기 때문에 정상 동작 시 스캔 출력포트를 통하여 AES 코아의 내부 정보가 유출될 가능성은 존재 하지 않게 된다.

본 논문에서 사용한 AES 코아는 그림 6과 같은 구조를 가지며 그림 7은 로직합성 후 게이트 수준 시뮬레

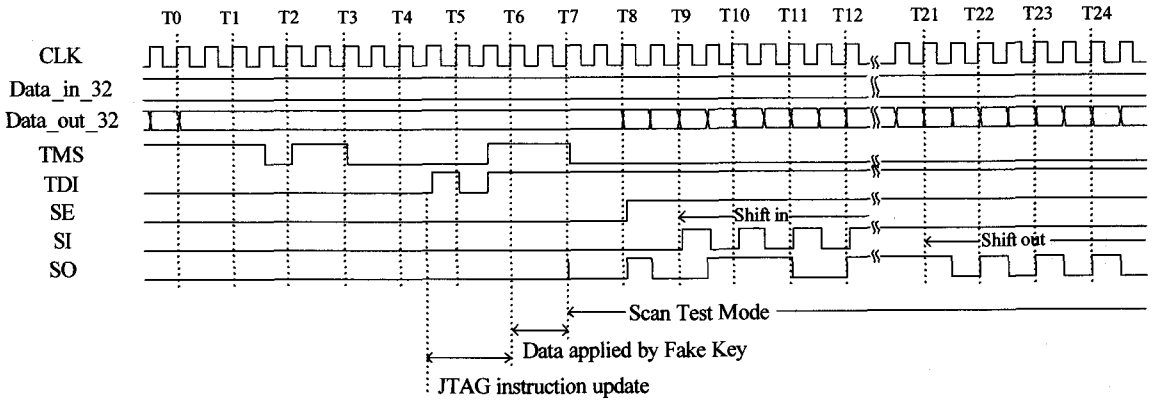

그림 7. AES의 Key 값이 적용되는 타이밍 다이어그램
Fig. 7. Timing diagram for AES key application.

이션 결과이다. 그림 7에서 정상 동작 중인 칩에 스캔 테스트를 위한 JTAG 명령어가 인가되는 시점인 T6에 사용자 키 대신 거짓 키 값이 AES 코아에 인가된다. 그림 6에서 키 레지스터 값이 변경되고 변경된 거짓 키가 적용된 데이터로 데이터 레지스터 값이 변경되기까지는 2 클락이 소요된다. 따라서 T6으로부터 2 클락 싸이클 동안 Scan Output Gating 신호를 0으로 유지시켜 스캔 출력 포트(SO)를 차단한다. 2 클락 후 AES 내부에 남아있던 정보가 제거되면 스캔 출력포트가 활성화되므로 스캔 테스트를 정상적으로 수행하면 된다.

본 논문에서 제안하는 방법을 사용하면 테스트 시 정상 동작 모드에서 암호화 진행 중 저장된 정보를 제거해 주기 위해 [9]처럼 리셋이 있는 플립플롭을 사용할 필요가 없어지므로 면적 및 전력소모 오버헤드를 줄일 수 있게 된다.

3.2. 스캔 테스트 동작 모드에서 사용자 비밀 키 누출 방지 기술

AES 코아의 스캔 테스트 시 사용자 비밀 키 누출을 방지 하면서 고장 검출율을 높이기 위해서는 테스트 모드 시 사용자 비밀 키 값을 대신할 수 있는 데이터를 인가하는 기술이 필요하다. 이를 위해 본 논문에서는 [9]에서 MKR과 같은 별도의 레지스터를 둔 것과는 달리, 그림 4에서와 같이 사용자 키와는 다른 값을 가진 고정된 거짓 키가 스캔 테스트 시 AES 코아에 인가 되도록 하였다. AES 코아에 인가되는 거짓 키 값은 TAP 제어기에서 보내지는 Load Key 신호에 의해 제어되는 멀티플렉서를 통해 사용자 비밀 키를 대신하여 AES 코아에 인가된다. 이렇게 고정된 거짓 키를 사용하면 테스트 제어도가 떨어지게 되어 고장 검출율이 낮아질 것이라 생각될 수 있으나, 이에 대한 분석을 다음과 같이 하였다.

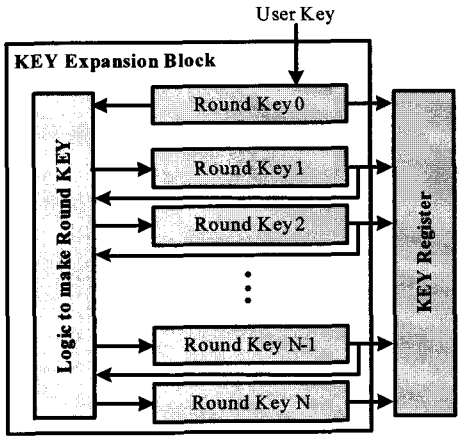

그림 8. 키 확장 블록
Fig. 8. Key expansion block.

AES 암호화 과정에서 라운드마다 적용되는 라운드 키들은 사용자 비밀 키를 이용하여 생성된다. 그러므로 AES 코아에는 각 라운드 키를 저장하기 위한 레지스터가 포함되어야 한다. 그림 5에서, 프리 라운드에서 적용되는 라운드 키는 바로 사용자 키이므로 이를 Round Key 0라고 하였다. 나머지 암호화 과정이 N 번째 라운드까지 존재한다고 할 때(본 논문에서는 128비트 키를 사용하므로 10 라운드까지 존재한다), 1 번째부터 N 번째 라운드 키 값은 전 라운드의 키 값을 이용하여 생성되므로, AES 코아의 키 확장 블록은 그림 7에서와 같이 전체적으로 피드백 구조를 이루게 된다. 따라서 본 논문에서 제안하는 고정된 거짓 키를 사용하더라도 AES 코아의 테스트 제어도에 거의 영향을 주지 않는

표 1. 기술성 비교표
Table 1. Comparison of technique.

	[8]	[9]	[10]	Prop.
Reusability of hard AES IP	N	Y	N	Y
Compatibility with JTAG	Y	N	Y	Y
Direct applicability of ATPG	Y	Y	N	Y

다. 실제로 100개의 임의로 선택된 거짓 키 값을 사용하여, 자동 테스트 패턴 생성툴인 Synopsys 사의 Tetra Max를 사용하여 고장 검출율을 실험한 결과, 모두 99.6% 이상의 고장 검출율을 얻을 수 있었다.

그러므로 본 논문에서 제안하는 시큐어 스캔 설계 기술은 스캔 테스트 시 사용자 비밀 키를 대신하여 거짓 키를 사용함으로써, 사용자 비밀 키가 누출 될 확률을 배제하는 동시에, 높은 고장 검출율을 얻을 수 있다.

Ⅳ. 실 험

본 논문에서는 100개의 거짓 키 중에서도 0의 값을 가진 거짓 키를 사용하였을 때 가장 높은 고장 검출율을 얻었으므로, 거짓 키의 값을 0으로 하여 실험을 진행하였다. 면적 및 전력소모 오버헤드 비교 실험을 위해 제안하는 기술의 설계는 AES 코아와 TAP 제어기 및 이를 이용한 테스트를 위한 로직을 포함하였다. 모든 대상 코아들은 33Mhz의 클락 속도로 동작하도록 하였다. 구현에는 MAGNA 0.18um 공정 라이브러리를 사용하였으며, 배치와 라우팅 시 다이의 크기는 1500um^{2}로 고정하고 5층의 메탈 레이어를 사용하였다. 키 누출을 방지하기 위한 기술을 적용한 AES 코아의 정상 동작과 스캔 테스트 동작 시 전력소모 오버헤드와 고장 검출율 측정을 위해서는 상업용 전력소모 측정 툴과 자동 테스트 패턴 생성 툴을 사용하였다.

본 논문에서 제안하는 기술은 스캔 설계 기반 테스트의 높은 고장 검출율을 유지하면서도 매우 낮은 사용자 키 누출 가능성을 가지며, 기존방법[8~10]에 비해 적은 면적 및 전력소모 오버헤드를 갖는다. 본 논문에서 제안한 방법의 기술적 특성을 기존 방법들[8~10]과 비교하여 표 1에 정리하였다. [8]의 방법은 JTAG과 호환되며 Automatic Test Pattern Generator(ATPG)를 이용한 테스트 패턴의 사용이 가능하나, AES 내부의 스캔 플립플롭을 키 매칭 블록과 연결해야 하므로 코아의 재사용성이 낮다. 따라서 최근 많이 활성화 되어 있는 재사용 가능한 IP 코아 기반설계 방법에 적용성이 낮다. [9]의 방법은 스캔 체인을 수정하지 않고 키 레지스터의 주변 로직만을 수정하므로 코아의 재사용성이 높고, ATPG로 생성되는 테스트 패턴을 그대로 사용할 수 있다. 하지만 IEEE 1149.1 표준 TAP 스테이트를 수정하여 표준 TAP 제어기와 호환이 되지 않는다는 단점이 있다. [10]의 방법은 스캔 체인에 인버터를 삽입하여 코아의 재사용성이 낮고 ATPG의 테스트 패턴 생성 후 추가적으로 S-box를 이용한 치환 작업을 필요로 한다. 즉, ATPG를 통해 생성된 테스트 패턴을 바로 사용할 수 없고, 추가적인 패턴 수정 작업이 필요하다.

표 2, 3, 4, 5는 본 논문에서 제안한 방법의 면적 오버헤드, 정상 동작 시 전력소모 오버헤드, 스캔 테스트 시 전력소모 오버헤드 및 고장 검출율을 [8~10]과 비교 분석한 결과이다. 표에서 [8]의 M은 스캔 출력포트 활성화를 위한 매칭 키의 매칭 횟수를 나타내며 N은 매칭 키의 비트 수를 나타낸다. [10]에서 제안된 방법에 대한 실험에서는 AES 코아의 내부에 4개의 스캔 체인(스캔 체인 길이 = {66,66,65,65})을 구현하였고 하나의 스캔 체인당 10개의 인버터를 삽입하였다.

표 2는 AES 코아에 기존 방식과 본 논문에서 제시하는 키 누출 방지 기술을 적용했을 경우의 면적 오버헤드를 보여주며, 제안한 방법의 면적 오버헤드가 가장 낮은 것을 확인할 수 있다. [8]의 결과에서 본 논문에서 제시하는 방법보다 셀 면적은 더 작은 경우가 있지만 매칭 키 비트의 길이인 N이 증가할수록 라우팅 오버헤드가 증가하여 결과적으로 전체 면적 오버헤드는 본 논문에서 제시하는 기술보다 높다. 또한 [10]의 방법은 추가된 인버터와 S-box 로직으로 인해 제안하는 방법보다 면적 오버헤드가 1.41% 더 높다.

표 2. 면적 오버헤드 비교표
Table 2. Comparison of area overhead.

		Cell Area (um2)	Routing Area (um2)	Total Area (um2)	Total Area Inc. (%)
AES	-	185221	215166	400387	Non
[9]	-	201786	219351	421137	5.18
[8] (M/N)	256/1	188948	215535	404483	1.02
	128/2	187624	215746	403370	0.75
	32/8	187647	217001	404648	1.06
	16/16	187467	219364	406831	1.61
	4/64	187920	223057	410977	2.64
	1/256	188212	227594	415806	3.85
[10]	-	188467	218430	406897	1.63
Prop.	-	185397	215886	401283	0.22

표 3. 정상 동작 시 전력소모 오버헤드 비교표
Table 3. Comparison of functional power overhead.

		Dynamic (e-02 w)	Leakage (e-06 w)	Total (e-02 w)	Total Inc. (%)
AES	-	1.1272	5.285	1.1277	Non
[9]	-	1.1498	9.739	1.1508	2.04
[8] (M/N)	256/1	1.1301	5.464	1.1306	0.26
	128/2	1.1331	5.452	1.1336	0.52
	32/8	1.1397	5.434	1.1402	1.11
	16/16	1.1444	5.424	1.1449	1.53
	4/64	1.1538	5.407	1.1543	2.36
	1/256	1.1695	5.391	1.1700	3.75
[10]	-	1.1750	7.328	1.1757	4.26
Prop.	-	1.1285	5.435	1.1290	0.12

표 4. 스캔 테스트 동작 시 전력소모 오버헤드 비교표
Table 4. Comparison of scan test power overhead.

		Dynamic (e-03 w)	Leakage (e-06 w)	Total (e-03 w)	Total Inc. (%)
AES	-	1.9546	5.285	1.9599	Non
[9]	-	2.1918	9.739	2.2015	12.33
[8] (M/N)	256/1	1.9688	5.464	1.9743	0.73
	128/2	1.9750	5.452	1.9805	1.05
	32/8	1.9861	5.434	1.9915	1.61
	16/16	1.9997	5.424	2.0051	2.31
	4/64	2.0232	5.407	2.0286	3.51
	1/256	2.0494	5.391	2.0548	4.84
[10]	-	2.0628	7.328	2.0701	5.62
Prop.	-	1.9877	5.435	1.9931	1.70

표 5. 고장 검출율 비교표
Table 5. Comparison of fault coverage.

		FC(%)	FC Inc.(%)	Scan Chain Length
AES	-	99.62	Non	262
[9]	-	99.68	0.06	390
[8] (M/N)	256/1	99.62	0	262
	128/2	99.62	0	262
	32/8	99.62	0	262
	16/16	99.62	0	262
	4/64	99.62	0	262
	1/256	99.62	0	262
[10]	-	99.62	0	262
Prop.	-	99.85	0.23	262

표 3은 AES 코아의 정상 동작 시 전력소모 오버헤드를 나타낸다. 제안하는 방법은 MKR과 같은 별도의 레지스터를 필요로 하지 않고 리셋이 있는 플립플롭을 사용하지 않아도 되므로 [9]보다 전력소모 오버헤드가 1.92% 낮다. [8]의 경우에는 스캔 플립플롭의 출력에 스캔 출력포트 활성화를 위한 매칭 블록으로의 연결선을 추가하기 때문에 N의 크기가 커질수록 스캔 플립플롭의 팬아웃 부하가 증가하여 정상 동작 시 동적 전력소모가 증가한다. 결과적으로 [8]은 본 논문에서 제시하는 방법보다 높은 전력소모 오버헤드를 갖는 것을 볼 수 있다. [10]의 방법은 인버터와 AES 스캔 체인 출력단에 추가된 S-box로직 때문에 정상 동작 시 전력소모 오버헤드가 본 논문에서 제시한 방법보다 3.14% 더 높다.

표 4는 스캔 테스트 시 전력소모량의 비교표이다. [9]는 MKR 레지스터로 인하여 동적인 전력소모가 증가하여 스캔 테스트 동작 시 전력소모 오버헤드가 본 논문의 방법보다 10.63% 더 높다. [8]의 경우는 정상 동작 시 전력소모의 경우와 마찬가지로 N의 크기가 커질수록 팬아웃의 증가에 따른 동적인 전력소모가 증가하여, 스캔 테스트 시 전력소모 오버헤드가 증가하는 것을 볼 수 있다. [10]의 방법 역시 인버터와 S-box에 의한 전력소모가 스캔 테스트 시에도 존재하기 때문에 전력소모량이 본 논문에서 제시한 방법보다 크다.

고장 검출율의 비교를 표 5에 나타내었다. [8~10] 방법과 본 논문에서 제시한 방법 모두 높은 고장 검출율을 보인다. 그러나 [9]의 방법에서 MKR의 추가는 스캔 체인의 길이를 늘리기 때문에 테스트 시간의 증가를 가져온다. [8]의 방법은 스캔 출력 포트의 활성화를 위한 매칭 키를 인가해야하므로 복잡한 테스트 시퀀스를 갖는다. [10]의 방법은 S-box로 인해 ATPG의 테스트 패턴 생성이 자유롭지 않기 때문에, AES 코아의 테스트 패턴 값을 S-box를 이용해 치환해주는 추가적인 작업을 필요로 한다.

V. 결 론

본 논문에서는 거짓 키 및 JTAG 명령어 기반 기술을 사용하여 SoC에 내장된 AES 암호화 코아의 사용자 비밀 키를 스캔 기반 사이드 채널 공격으로부터 보호하기를 위한 효과적인 방법을 제안하였다. 제안하는 방식은 공격자가 비밀 키와 관련된 정보를 스캔 출력 포트를 통해 얻는 것을 효과적으로 차단한다. 또한 제안한 기술은 IEEE 1149.1 표준과 호환될 뿐만 아니라 어플리케이션에 최적화되어 설계된 AES 코아에 적용 시에 추가적인 수정 없이 적용이 가능하므로 코아 재사용성을 높일 수 있다. 따라서 최근 많이 활성화 되어 있는 재사용 가능한 IP 코아 기반설계 방법론에 부합한다. 동시에 기존 방식보다 적은 면적 오버헤드와 전력 소모를 가지며 스캔 설계 기반 테스트의 높은 고장

검출율을 유지할 수 있는 시큐어 스캔 기술임을 실험을 통해 알 수 있었다. 본 논문에서 제안한 기술은 AES 코아 뿐만 아니라 다른 암호화 코아에도 효과적으로 적용될 수 있을 것이다.

참 고 문 헌

[1] S. Mangard, M. Aigner and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture", IEEE Transactions on Computer, vol. 52, no. 1, pp. 483-491, April, 2004.

[2] D. Josephson and S. Poehhnan, "Debug methodology for the McKinley processor", International Test Conference(ITC), pp. 451-460, Baltimore, MD, USA, Oct. 30- Nov. 1, 2001.

[3] J. Lee, M. Teharanipoor, C. Patel and J. Plusquellic, "Securing Designs Against Scan-Based Side-Channel Attacks", IEEE Transations on Depedable and Secure Computing, Vol. 4, no. 4, pp. 325-336, Oct.-Dec., 2007.

[4] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing*, Kluwer Academic Publishers, 2000.

[5] R. Kapoor, "Security vs. test quality: Are they mutually exclusive?", in Proc. ITC, pp. 1414, Charlotte, NC, USA, Oct. 26 - 28, 2004.

[6] J. Lee, M. Teharanipoor, and J. Plusquellic, "A Low-Cost Solution for Protecting IPs Against Scan-Based Side-Channel Attacks", VLSI Test Symposium, pp. 94-99, Berkeley, CA, USA, Apr. 30-May 4, 2006.

[7] B. Yang, K. Wu and R. Karri, "Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard", ITC, pp. 339-344, Charlotte, NC, USA, Oct. 26 - 28, 2004.

[8] S. Paul, R. S. Chakraborty and S. Bhunia, "VIm-Scan : A Low Overhead Scan Design Approach for Protection of Secret Key in Scan-Based Secure Chips", VLSI Test Symposium, pp. 455-460, Berkeley, CA, USA, May 6-10, 2007.

[9] B. Yang, K. Wu and R. Karri, "Secure Scan : A Design-for-Test Architecture for Crypto Chips", IEEE Transaction Computer-Aided Design of Integrated Circuits and systems, Vol. 25, No.10, pp. 2287-2293, Oct. 2006.

[10] G. Sengar, D. Mukhopadhyay and D. R. Chowdhury, "Secured Flipped Scan-Chain Model for Crypto-Architecture", IEEE Transaction Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, No.11, pp. 2080-2084, Nov.2007.

[11] J. Seberry, X. M. Zhang and Y, Zheng, "Systematic generation of cryptographically robust S-boxes", The 1st ACM Conference on Computer and Communications Security, pp. 171-182, Fairfax, Virginia, USA, Aug. 10, 1993.

[12] W. Stallings, "Cryptography and Network Security", Englewood Cliffs, NJ : Prentice-Hall, 2003.

저 자 소 개

송 재 훈(정회원)
2000년 한양대학교 전자컴퓨터 공학과 학사 졸업.
2002년 한양대학교 컴퓨터공학과 석사 졸업.
2003년 서울대학교 SoC 설계센터 연구원.
2009년 한양대학교 컴퓨터공학과 박사 졸업.
2009년~현재 트란소노 책임연구원
<주관심분야 : SoC 설계 및 테스트, 테스트를 고려한 설계>

정 혜 란(정회원)
2005년 한양대학교 컴퓨터공학과 학사 졸업.
2009년 한양대학교 컴퓨터공학과 석사 졸업.
2009년~현재 C&S Technology 연구원.
<주관심분야 : SoC 설계 및 테스트, ASIC 설계>

박 성 주(평생회원)
1983년 한양대학교 전자공학과 학사 졸업.
1983년~1986년 금성사 소프트웨어개발 연구원.
1992년 Univ. of Massachusetts 전기 및 컴퓨터공학과 박사 졸업.
1992년~1994년 IBM Microelectronics 연구스텝.
1994년~현재 한양대학교 전자컴퓨터공학부 정교수
<주관심분야 : 3D 그린 IC 테스트 설계, 테스트 합성, Built-In Self Test, Scan Design, ATPG, ASIC 설계, 고속 신호처리>

정 태 진(정회원)
2007년 한양대학교 컴퓨터공학과 학사 졸업.
2009년 한양대학교 컴퓨터공학과 석사 졸업.
2009년~현재 C&S Technology 연구원.
<주관심분야 : SoC 설계 및 테스트, ASIC 설계>

김 화 영(학생회원)
2009년 한양대학교 컴퓨터공학과 학사 졸업.
2009년~현재 한양대학교 컴퓨터공학과 석사 과정.
<주관심분야 : SoC 설계 및 테스트, ASIC 설계>