하드웨어 칩 기반 보안시스템 및 해킹동향

최필주, 최원섭, 김동규
한양대학교

요 약

최근 보안기술 동향은 소프트웨어 기반 보안시스템의 한계를 보완하고자 하드웨어 기반 보안시스템으로 발전하고 있다. 그러나 이러한 하드웨어 기반 보안시스템도 부채널 공격, 메모리 공격, 버스프루빙 공격과 같은 해킹 기술에 취약하다. 본고에서는 최근 보안기술 동향을 기술하고, 하드웨어 보안시스템에 대한 해킹기술 및 이에 대한 대응책에 대하여 기술한다.

Ⅰ. 서 론

최근 국내외 여러 기관에서 해킹 사고가 일어남에 따라 소프트웨어 보안 시스템에 대한 문제점이 크게 제기되고 있다. 소프트웨어 보안시스템은 OS보다 상위 계층에서 동작함으로 OS가 안전하다는 보장이 되었을 때에 보안이 확립 될 수 있다. 또한 OS도 하드웨어가 안전하다는 보장이 있을 때 보안이 확립 될 수 있다. 이렇듯 소프트웨어는 하위 레벨의 보안이 확립되어야 안전이 보장된다는 한계가 존재한다. 또한 소프트웨어는 컴퓨터의 시스템구조상 메모리를 통해서만 연산이 수행되므로 메모리공격을 피할 수가 없다. 최근에는 이러한 소프트웨어 보안시스템의 한계를 보완하기 위해 다양한 하드웨어 보안시스템이 제안되고 있다. 그러나 하드웨어 보안시스템도 부채널공격, 메모리공격, 버스프루빙 공격과 같은 해킹기술에 정보가 노출 될 수 있다. 본고에서는 최근 하드웨어 보안시스템의 동향을 분석하고 하드웨어 보안시스템의 취약점 및 이를 해결하기 위한 대응책을 제시한다.

Ⅱ. 관련기술

1. Trusted Platform Module

Trusted Platform Module(TPM)[1]은 외부에서 접근이 불가능한 안전한 장소에 키를 보관하고 내부에서 암호화 연산을 수행하도록 고안된 하드웨어 칩이다. TPM의 내부에서 암호화된 데이터는 동일한 암호키를 가진 해당 TPM에서만 복호화가 가능하다. 또한 TPM은 Platform Configuration Register(PCR)를 통해 각종 시스템 정보(BIOS, 커널등)와 사용자가 입력한 PIN을 수집하여 이전에 저장되어 있는 정보와 비교를 통해 정보가 일치할 경우에만 시스템 사용이 가능하다. 이를 통해 공격자가 TPM칩과 암호화된 데이터를 습득하여 타 시스템에서 복호화하는 행위를 차단할 수 있다.

TPM은 다음과 같은 구조로 이루어 진다.

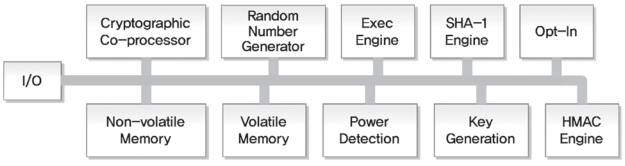


그림 1. TPM 아키텍처

TPM은 크게 명령어 처리 엔진(Exec Engine), 암호화 엔진, 비휘발성 및 휘발성 메모리으로 구성된다. 비휘발성 메모리에는 TPM의 고유키들이 저장되며 외부로 노출되지 않는다. 명령어 처리 엔진에서는 칩 운영체제와 TPM 명령어를 수행하며 이때, 비휘발성 메모리에 저장된 암호키와 암호화 엔진을 사용하여 데이터 암복호화 및 각종 인증을 수행한다.

TPM은 하드웨어적으로 메인 시스템과 분리되어 외부 접근을 차단함으로써 소프트웨어공격으로부터 암호키와 데이터를 안전하게 관리할 수 있지만, 하드웨어를 직접 공격하는 메모리 공격 및 버스 프루빙 공격에는 키와 데이터가 노출 될 수 있는 위험이 있다.

2. Trusted Execution Environment

Trusted Execution Environment(TEE)는 스마트폰과 같은 모바일 기기의 메인 프로세서를 보안영역과 일반영역으로 나누고 민감한 정보의 저장과 처리를 보안영역에서 수행함으로써

신뢰성을 보장하는 시스템이다. 일반영역은 기존의 안드로이드와 같은 일반 OS가 실행되는 환경이며 보안이 상대적으로 덜 중요한 엔터테인먼트 목적의 프로그램이 수행된다. 보안영역에서는 지불 결제나 Digital Rights Management(DRM) 등의 보안이 중요한 어플리케이션을 안전하게 수행할 수 있다. 일반영역과 보안영역은 물리적으로 분리되어 있으며 일반영역에서 보안영역으로의 접근은 〈그림 2〉에서와 같이 TEE에서 제공되는 API를 통해서만 가능하다.

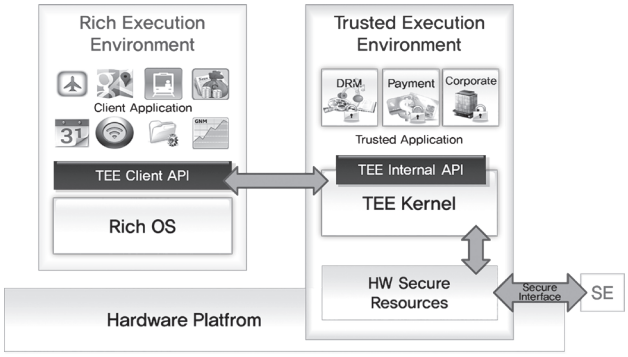


그림 2. TEE 아키텍처

TEE에서는 여러 보안 기능을 제공하는데 암호키 저장 및 데이터 암호화 기능 이외에도, 시간정보관리 기능을 통해 디지털 컨텐츠의 유효 기간 관리가 가능하며 비밀번호 입력과 같은 민감한 입출력에 대한 보호기능이 있다.

이러한 TEE에 관한 명세는 보안 칩 관련 명세에 대해 활동하고 있는 GlobalPlatform에서 정립하고 있으며 현재 TEE의 시스템 구조 [2], TEE의 2가지 API인 Internal API [3] 및 TEE Client API [4]에 대한 명세가 정립된 상태이다. TEE를 적용한 프로세서는 ARM사의 TrustZone[5]이 대표적이며 삼성전자의 KNOX[6]를 포함하여 모바일 및 다양한 임베디드 디바이스에서 TrustZone을 이용한 전용 보안플랫폼이 개발되고 있다.

3. 전용 보안플랫폼

삼성전자에서는 ARM의 TrustZone을 적용한 보안플랫폼 KNOX를 개발하였다. KNOX를 통해 사용자는 실제 안드로이드 앱을 통해 TEE 환경을 사용할 수 있다. KNOX는 여러 보안 기능을 제공한다. 우선 KNOX는 컨테이너라는 영역을 만들어 컨테이너 내부의 텍스트나 이미지, 파일 등은 외부에서 접근할 수 없도록 하였다. 또한 컨테이너 내부의 모든 정보는 암호화되어 저장된다.

그리고 KNOX는 Trusted Boot를 통해 부팅시에 허가되지 않는 OS와 소프트웨어를 사전에 방지한다. Trusted Boot는 부팅 시 디바이스에서 실행되는 펌웨어의 인증 여부를 확인하고, 이에 대한 측정값을 TrustZone에 안전하게 저장한다. 이후 시스템 런타임 시 KNOX 플랫폼의 TrustZone 애플리케이션은 이러한 측정값을 사용하여 보안 키 공개, 컨테이너 활성화 확인 등의 중요한 보안 결정을 내리게 된다.

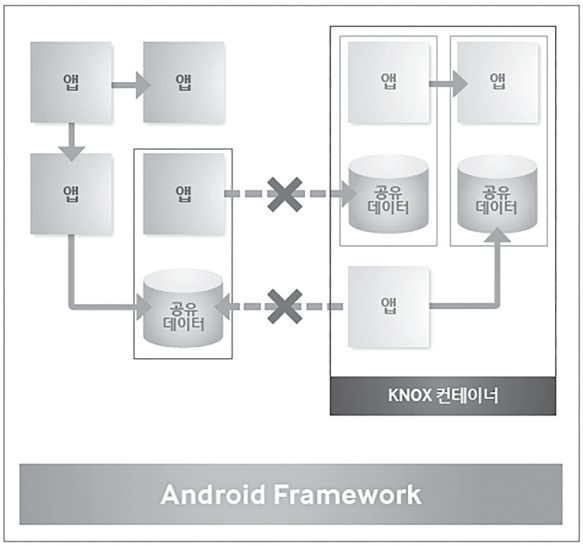


그림 3. KNOX 컨테이너

KNOX는 트러스트존 무결성 측정 아키텍처(TrustZone-based Integrity Measurement Architecture, TIMA)를 사용하여 ARM TrustZone 하드웨어와 리눅스 커널의 무결성 여부를 지속적으로 모니터링 한다. TIMA는 TrustZone의 보안영역 체계에서만 동작하며 임의로 중지시킬 수 없다. TIMA는 커널의 무결성이 침해 당할 경우 원격으로 이를 사용자에게 알려준다.

이상으로 최근 보안 관련 기술에 대해서 알아보았다. 세가지 관련 기술은 모두 하드웨어적으로 보안 시스템을 분리하는 것을 기반으로 하여 기존의 소프트웨어 보안시스템에서 발생할 수 있는 보안키 저장 문제와 메모리 해킹 문제 등을 막고 있으나, 하드웨어의 직접적인 해킹에 대해서는 취약하다.

Ⅲ. H/W 해킹기술

이번 장에서는 하드웨어 보안플랫폼에 대한 해킹기술에 대해서 알아본다.

1. 부채널 공격

부채 널 공격(SCA, Side Channel Attack)은 칩이 동작할 때

변화하는 전력 소모, 열, 연산 소모 시간, 전자기파(electro-magnetic wave) 등의 부가적인 정보, 즉 부채널 정보를 이용하는 공격방법이다. 〈그림 4〉은 부채널 공격 모델을 나타낸다.

그림 4. 부채널 공격 모델

부채널 정보를 수집 및 분석하여 비밀 데이터 및 키 등의 주요 보안 정보를 추출해낼 수 있으며 공격 시 칩의 외부에서 접근 가능한 인터페이스만을 이용하기 때문에 상대적으로 적은 시간과 비용만을 필요로 한다. 따라서 security SoC의 가장 현실적이고 위협적인 공격 방식으로 인식되고 있다.

부채널 공격은 사용되는 부채널 정보에 따라 시차 공격, 전력 분석 공격, 전자기파 분석 공격, 오류 주입 공격 등으로 분류할 수 있다. 시차 공격은 입력 데이터에 따라 달라지는 연산의 시간이나 연산 회수를 관찰하여 분석하는 방법이다. 예를 들어 대표적인 공개키 알고리즘인 RSA의 경우 키의 각 bit 값에 따라 내부에서 수행되는 연산의 종류가 달라지므로 이를 관찰하여 키 값을 추출하는 것이 가능하다[7].

전력 분석 공격은 공격 대상 암호 장치의 소비 전력과 연산 중 처리되는 데이터간의 상관도를 이용하는 것으로 부채널 공격 방법들 중에서 가장 효과적이고 위협적인 공격 방법으로 알려져 있다. 전력 분석 공격은 분석 방식의 복잡도에 따라 단순 전력 분석(Simple Power Analysis : SPA)과 차분 전력 분석(Differential Power Analysis : DPA)으로 나누어진다. SPA는 특정 명령어가 수행되는 한 시점에서 데이터에 따라 달라지는 소비전력을 분석하여 비밀정보를 유추하는 방법이다.

〈그림 5〉는 RSA의 파형을 분석한 것으로 연산 시간이나 파형 모양을 관찰하여 키 값을 분석하는 과정을 나타낸다. 그러나 이러한 SPA는 공격자가 공격하고자 하는 시점의 구현방법을 정확히 알고 있어야하는 단점이 있다. 반면 DPA는 비밀 정보 비트와 소비전력의 통계적인 상관관계를 이용하여 비밀 정보를 유추할 수 있는 방법으로 노이즈에 강인하고 아주 적은 자원을 사용하기 때문에 부채널 공격법 중 가장 강력한 공격법 중 하나로 알려져 있다.

전자기파 분석 공격은 공격 대상 암호 장치로부터 방사되는

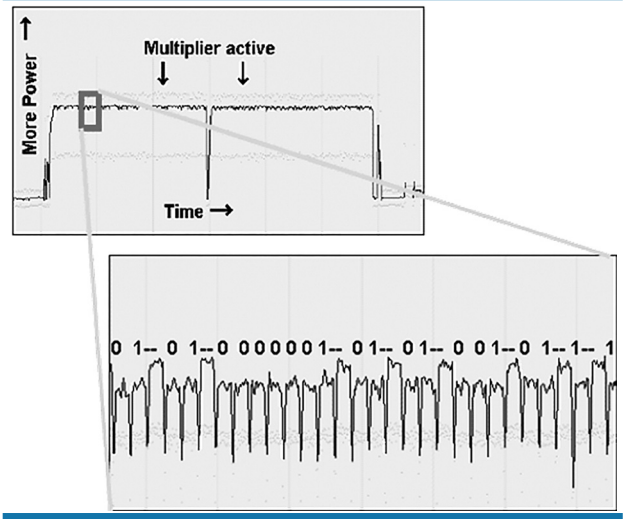


그림 5. RSA 연산 파형

전자기파 신호를 분석하는 방법으로 전력 분석 공격과 달리 원거리에서 정보의 습득이 가능하며 다중 채널로 구성되고 있어서 전력 분석 공격 대응 장치에서도 전자기파 정보의 분석이 가능하다는 장점이 있다. 전력 분석 공격과 비슷하게 단순 전자기파 분석 (Simple Electro-Magnetics Analysis : SEMA)과 차분 전자기파 분석 (Differential ElectroMagnetics Analysis : DEMA)으로 분류된다.

오류 주입 공격은 암호 연산을 위한 칩이나 하드웨어에 예상치 못한 결함을 유발시켜 발생된 잘못된 출력 값을 분석함으로써 내부의 비밀 정보를 알아내는 공격 방법이다. 칩의 패키징을 제거한 후 오류를 유발시키는 방법과 외부에서의 전기적 스파크를 일으켜 오류를 유발시키는 방법 등이 있다.

2. 메모리 공격

메모리 공격은 메모리 내의 내용을 추출하거나 복제, 변경하는 방법이다. 공격 대상은 메모리의 위치에 따라 Off-SoC 메모리와 On-SoC 메모리로 나뉠 수 있으며 메모리의 종류에 따라 휘발성 메모리와 비휘발성 메모리로 나뉠 수 있다.

Off-SoC 메모리의 경우 쉽게 메모리에 연결된 버스를 관찰하거나 메모리에 특수 처리를 하여 메모리 내의 정보를 획득할 수 있으며 공격자의 악의적인 코드가 담긴 메모리로 교체하는 등의 공격도 가능하다. On-SoC 메모리는 off-SoC 공격에 비해 상대적으로 공격이 매우 어려우나 뒤에서 설명할 역공학을 통해 칩 내부의 메모리에 접근하여 메모리의 내용을 추출하는 것이 가능하다. 그 밖에 메모리에 필요한 공급 전력을 차단하여 시스템의 불완전한 틈을 노릴 수도 있다.

메모리 공격의 대상은 비휘발성 메모리로 한정되지 않으며 휘

발성 메모리도 공격이 가능하다. 휘발성 메모리의 경우 공급 전원이 차단될 경우 저장된 데이터가 사라지기 때문에 공격하기에 어려움이 있으나 완전히 데이터가 사라지기까지 약간의 시간을 소모하며 온도를 급격히 낮출 경우 그 시간은 수분에서 수십분 이상으로 늘어난다. 따라서 전원이 차단된 비휘발성 메모리로부터도 저장되어 있던 데이터를 획득하는 것이 가능하다. 소거 기능을 사용하더라도 SRAM의 경우 반전에 의해 데이터가 소거되므로 소거 후 데이터로부터 원래 데이터를 추측할 수 있다.

3. 역공학을 통한 버스 프루빙 공격

역공학은 칩의 패키지를 제거하고 칩의 각 층을 하나씩 제거하며 레이아웃을 찍은 후 컴퓨터로 획득한 레이아웃을 분석하는 방법이다. 이를 통해 칩 내부의 레이아웃 정보를 모두 획득할 수 있으며 기술에 따라 칩이 동작하는 상태에서 외부 패키지만 제거한 후 원하는 곳의 신호를 관찰하는 것이 가능하다.

Security SoC는 하나의 칩으로 되어있어 외부로부터 공격에 안전한 편이나 역공학 기술을 사용할 경우 칩 내부까지 관찰되어 중요 정보가 노출될 수 있다. SoC 칩의 내부가 관찰 가능할 경우 취약점을 살펴보기 위해 내부 구성을 살펴보면 〈그림 6〉과 같다.

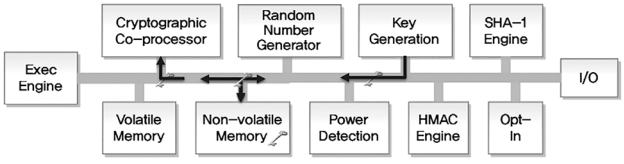


그림 6. 기본적인 내부 구성도

〈그림 6〉은 Secure SoC의 기본적인 내부 구성을 나타낸다. 기본적으로 칩의 동작을 주관하는 core인 execution engine이 있으며 데이터 저장을 위한 메모리와 외부와 인터페이스하기 위한 I/O, 마지막으로 보안 기능을 지원하기 위한 다양한 보안 모듈들로 구성되어 있다. 대부분의 보안 모듈들은 데이터의 암호화 및 복호화, 디지털 서명 생성 및 증명, HMAC 생성을 통해 중요 데이터에 대한 무결성 및 기밀성을 보장하고 사용자 인증, 부인 방지 등의 기능을 지원한다. 그러나 이러한 보안 기능들을 수행하기 위해서는 키를 사용하는데 이 키는 키 생성 모듈에서 생성되어 각 보안 모듈에서 사용되고 필요에 따라 메모리에 저장된다. 만약 bus를 관찰하는 것이 가능하다면 이러한 키 값이 노출될 수 있다.

실제로 스마트카드 칩의 내부 bus가 역공학 공격에 의해 관찰된 사례 [8]가 존재한다. 〈그림 7〉은 역공학을 하기 위해 칩의 외부 패키지를 제거한 모습이며 〈그림 8〉은 역공학에 의해 칩의 레이아웃이 분석된 모습이다.

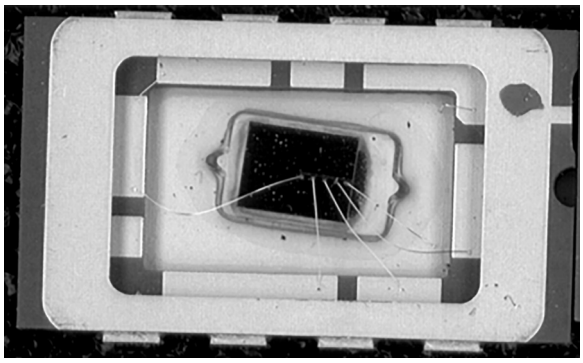


그림 7. 외부 패키지가 제거된 칩

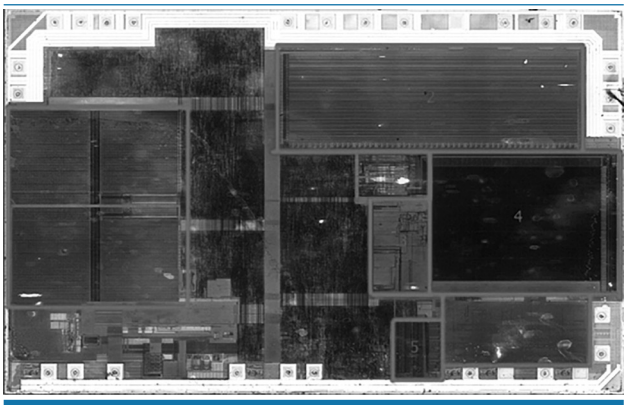


그림 8. 내부 레이아웃

역공학에 의해 내부의 레이아웃이 획득되면 〈그림 8〉과 같이 메모리나 코어 등의 IP는 쉽게 구별 가능하므로 이들 IP에 연결된 bus도 쉽게 찾아낼 수 있다. Infineon 66PE chip에 대한 해킹 사례에서는 이렇게 내부 bus를 관찰하여 지나는 데이터를 수집하고 분석하여 코드를 얻어내고 이를 이용하여 공격을 시도하였다. 이는 실제로 역공학을 통해 SoC 내부의 bus를 관찰하는 것이 가능하며 이는 bus를 통해 이동하는 데이터를 추출할 수 있다는 것을 의미한다. 비슷한 방법으로 bus를 통해 내부 코드뿐만 아니라 이동 중인 키 값도 추출될 수 있으며 특히 키가 노출될 경우에는 쉽게 내부 비밀정보를 복호화하거나 서명을 위조할 수 있다는 취약점이 존재한다.

Ⅳ. 대응 방안

이번 장에서는 위에서 설명한 하드웨어 보안플랫폼의 해킹 취약점에 대한 대응 방안을 논의한다.

1. 부채널 공격 대응 방안

부채널 공격의 대응 방법으로 가장 많이 사용되는 것은 마스킹(masking) 방법과 하이딩(hiding) 방법이다. 마스킹 방법은 중간 과정에서의 연산 값을 랜덤하게 만드는 방식으로 원래 중간 연산 값과의 관계를 약하게 만들어 주는 것을 의미한다. 하이딩 방법은 연산 중간의 전력 소모량의 데이터 연관성을 제거하기 위해 소모량을 통일하거나 랜덤하게 만드는 방식을 의미한다.

마스킹은 랜덤 수를 적용하는 연산 종류에 따라 부울 마스킹(boolean masking)과 산술 마스킹(arithmetic masking) [9]으로 구분된다. 부울 마스킹은 XOR이나 AND와 같은 선형 연산에 랜덤 수를 적용하며 산술 마스킹은 AES의 S-box와 같은 비선형 연산에 랜덤 수를 적용한다. 그러나 이렇게 추가된 랜덤 성분은 DPA 시 약해지기 때문에 고차원의 DPA [10]에 취약하다는 단점이 있다. 따라서 하이딩 기법으로 대체하거나 이와 병행해서 사용해야 한다.

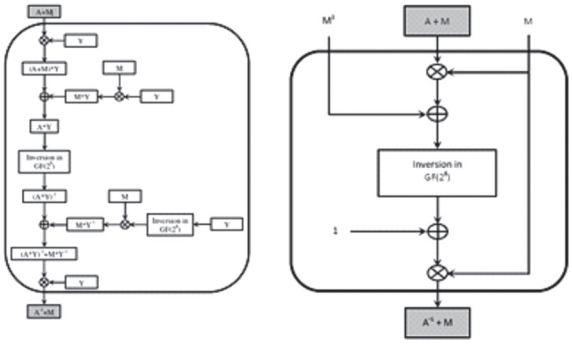


그림 9. 무작위수 M을 이용한 마스킹 기법

하이딩 방법의 주된 방식은 실행되는 동작과 데이터에 따른 전력소모량이 독립적이 되도록 회로를 구성하는 것이다. 일반적으로 DRP 회로 방식(Dual-rail precharge logic style)을 기반으로 하여 만들어진다.

일반 CMOS(<그림 10>)는 입력 값에 따라 출력 값의 경로가 다르나 DRP(<그림 11>)의 경우 입력 값과 관계없는 출력 값을 가지며 모든 경우에 대하여 일정한 load capacitance를 갖는다. SABL (Sense Amplifier Based Logic) [11]과 WDDL (Wave Dynamic Differential Logic) [12]가 대표적인 DRP 회로로 SABL은 트랜지스터 수준에서 구현되며 WDDL은 일반 셀 라이브러리를 이용하여 만들어진다. 이러한 회로를 구성할 때에는 연결의 대칭이 중요하므로 대칭되는 노드 간 capacitance의 차이가 적도록 대칭적으로 설계하는 것이 중요하다.

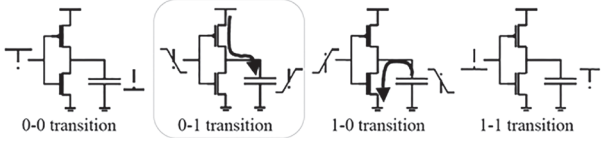


그림 10. 일반적인 CMOS

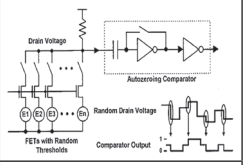
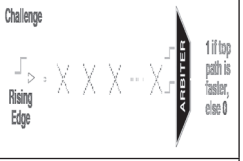
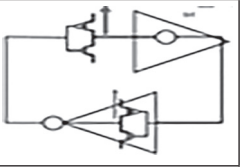
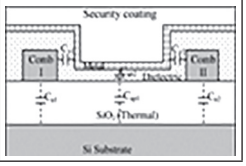
그림 11. DRP 회로

2. 메모리 공격 대응 방안

메모리 공격으로부터 중요 데이터를 보호하기 위해서는 메모리에 중요 데이터가 평문 상태로 저장되는 것을 방지해야한다. 그러나 데이터를 암호화하기 위해서는 반드시 키가 필요하므로 적어도 하나 이상의 키는 근원키로써 평문 형태로 저장되어야 한다. 따라서 이러한 문제를 해결하기 위한 방법이 필요하다.

PUF(Physical Unclonable Function)는 메모리 공격의 대응 방안으로 근원키를 구현하는데 사용될 수 있다. PUF는 말 그대로 복제 불가능한 함수로 똑같은 레이아웃을 이용하여 제작되나 칩마다 예측 불가능한 랜덤한 값을 생성하는 회로이다. PUF를 사용함으로써 각 칩은 고유의 근원키를 가지게 되며 이 키는 메모리에 저장되지 않는다. 추가적으로 키가 필요한 경우 근원키를 이용하여 암호화 후 메모리에 저장하게 된다. <표 1>은 다양한 방식의 PUF를 나타낸다.

표 1. 다양한 방식의 PUF

연도	구현 방법	제작자	layout
2000 [13]	CMOS의 랜덤 drain voltage	SiidTech, Portland State University	
2005 [14]	delay path의 속도 차	MIT	
2007 [15]	SRAM의 불안전한 초기 상태	Philips	
2009 [16]	랜덤 capacitance 값	NXP semiconductors	

PUF는 키로 사용되는 것 외에도 값싼 ID로도 사용될 수 있는 등 다양하게 활용하는 것이 가능하나 랜덤성과 안정성, 높은 수율 등의 필요조건이 있어 쉽게 구현하기 어려운 기술이다. 현재 delay path의 속도 차를 이용한 arbiter PUF [14]와 SRAM의 초기 불안전한 상태를 이용한 SRAM PUF [15]를 각각 VERAYO와 INTRINSIC ID에서 상용화 중에 있으나 시불변성과 안정성을 확보하기 위한 ECC(Error Correction Code)를 필요로 하는 등의 문제가 있어 아직 출시된 제품은 전무한 상태이다. 그러나 PUF가 가지는 보안 특성으로 인해 기존의 단점을 개선할 수 있는 PUF에 대한 연구가 계속해서 진행되고 있으며 메모리 공격의 대응 방법뿐만 아니라 다양한 보안 프로토콜에서 사용될 수 있을 것으로 기대되고 있다.

3. 역공학을 통한 버스 프루빙 공격 대응 방안

SoC 내부의 bus를 오가는 데이터 중 특히 키 값을 보호하기 위하여 앞서 메모리 공격의 대응 방안에서 사용한 PUF를 사용할 수 있다. 〈그림 12〉는 PUF가 적용 시 변경되는 내부 구성을 나타낸다.

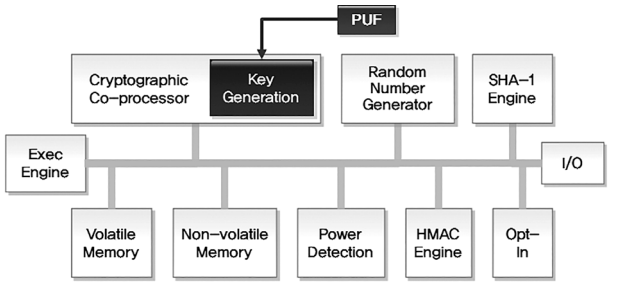


그림 12. PUF가 적용된 내부 구성도

변경된 구성에서는 키 생성 모듈이 암호 모듈 내부에 존재하고 키 생성 모듈은 PUF를 내부에 포함하고 있다. 키 생성 모듈은 PUF를 기반으로 키를 생성하므로 각 SoC 칩마다 PUF 기반의 고유 키 값을 가지며 필요에 따라 키를 동적으로 생성하고 키를 메모리에 저장하지 않는다. 또한 키가 필요한 암호 모듈은 내부의 키 생성 모듈로부터 바로 키를 전달받으므로 키가 bus를 이동할 필요성이 사라진다. 따라서 결과적으로 키가 bus를 이동하지 않으므로 버스 프루빙에 의해 키가 노출될 위험성이 사라지게 된다.

V. 결 론

최근의 보안기술은 기존 소프트웨어 보안시스템의 한계를 극복하기 위해 하드웨어 기반으로 이루어지고 있다. 하드웨어 기반의 보안시스템은 암호키를 메인시스템과 분리하고 칩 내부에서 암호화 연산을 하므로 소프트웨어 공격에는 안전하지만, 직접적인 하드웨어 공격에는 암호키가 노출될 위험이 있다. 본고에서는 하드웨어 해킹 공격으로 부채널 공격, 메모리 공격, 버스 프루빙 공격을 설명하였으며 이로부터 하드웨어 보안시스템도 위험할 수 있다는 것을 확인하였다. 최근에는 키 저장 문제를 원천적으로 해결하기 위해 PUF라는 하드웨어 지문이 활발히 연구되고 있다. PUF는 칩마다 존재하는 고유한 하드웨어 특성으로써 메모리형태로 존재하지 않으므로 공격자는 임의로 이 값을 얻어낼 수 없다. 이러한 특성을 가진 PUF를 기반으로 구축된 암호시스템은 키 노출의 위험성이 없어 기존 소프트웨어 및 하드웨어 보안시스템의 취약점을 보완함으로써 국가 정보망의 안정성 향상에 크게 기여할 것으로 기대된다.

참 고 문 헌

[1] Trusted Computing Group(TCG),"TPM Main Part1 Design Principles Specification Version 1.2", 2011.

[2] GlobalPlatform, "TEE System Architecture", (http://www.globalplatform.org)

[3] GlobalPlatform, "TEE Internal API Specification", (http://www. globalplatform.org)

[4] GlobalPlatform, "TEE Client API Specification", (http://www. globalplatform.org)

[5] ARM,"ARM Security Technology-Building a Secure System using TrustZone Technology", (http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf)

[6] Samsung, "Samsung Knox", (https://www.samsungknox.com/ko)

[7] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," CRYPTO '99, LNCS 1666, pp. 388-397, Springer-Verlag, 1999.

[8] C. Tarnovsky, "Deconstructing a 'Secure' Processor", Black Hat, Washington DC., (www.blackhat.com)

[9] Kamal, A.A. Youssef, A.M, "An-Area-optimized Implementation for AES with Hybrid Countermeasure against Power Analysis," ISSCS 2009, pp. 1-4, 2009.

[10] M. Joye, P. Paillier, and B. Schoenmaker, "On

Second-Order Differential Power Analysis," CHES 2005, LNCS 3659, pp.293-308, 2005.

[11] A. Hevia, M. Kiwi, "Strength of two data encryption standard implementations under timing attacks," ACM Trans. on Information and System Security, Vol. 2, pp.416-437, 1999.

[12] D. D. Hwang, K. Tiri, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, I. Verbauwhede, "AES-Based Security Coprocessor IC in 0.18-㎛ CMOS With Resistance to Differential Power Analysis Side-Channel Attacks," IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 41, NO. 4, pp. 781-791, 2006.

[13] K. Lofstrom, W.R. Daasch, and D. Taylor, "IC identification circuit using device mismatch," Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International, 2000.

[14] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, VOL. 13, 2005.

[15] J. Guajardo, S.S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," Cryptographic Hardware and Embedded Systems-CHES 2007, VOL. 4727, 2007.

[16] D. Roy, J. H. Klootwijk, N. Verhaegh, H. Roosen, and R. Wolters, "Comb capacitor structures for on-chip physical uncloneable function," Semiconductor Manufacturing, IEEE Transactions on, VOL. 22, 2009.

약 력

최 필 주

2010년 한양대학교 전자통신컴퓨터공학 학사
2012년 한양대학교 전자컴퓨터통신공학 석사
2012년~현재 한양대학교 전자컴퓨터통신공학 박사과정 재학
관심분야: 보안 SoC 설계, 스마트카드 보안, 전자결제시스템 보안, 자동차용 보안시스템

최 원 섭

2012년 한양대학교 전자통신컴퓨터공학 학사
2012년~현재 한양대학교 전자컴퓨터통신공학 석사과정 재학
관심분야: 모바일 결제 보안, 스마트카드 보안, TEE 보안

김 동 규

1992년 서울대학교 컴퓨터공학 학사
1994년 서울대학교 컴퓨터공학 석사
1999년 서울대학교 컴퓨터공학 박사
1999년~2006년 부산대학교 조교수
2006년~현재 한양대학교 융합전자공학부 교수
2007년~현재 한국정보보호학회 상임이사
2007년~현재 한국정보처리학회 상임이사
2010년~현재 금융보안연구원 표준화 자문위원
2011년~2012년 한국멀티미디어학회 재무부회장
2011년~2012년 기술표준원 모바일지불결제표준화협의회 부위원장
2011년~현재 한국은행 금융정보화추진협의회 자문위원
2012년~현재 한국전자공학회 반도체소사이어티 총무이사
관심분야: 보안 SoC 설계, 스마트카드 보안, 전자결제시스템 보안, 제어시스템 기능안전 및 보안, 자동차용 보안시스템