# SEE 분야의 연구 및 기술 동향

Recent Trends in Research and Technology of Secure Execution Environment

백광호 (K.H. Baek) 임베디드보안기술연구팀 연구원
강동호 (D.H. Kang) 임베디드보안기술연구팀 선임연구원
김기영 (K.Y. Kim) 임베디드보안기술연구팀 팀장

목 차



Secure execution environment는 안전한 컴퓨팅 시스템의 실행환경을 의미한다. 컴퓨터 시스템을 포함해서 프로세서를 가지고 있는 모든 종류의 단말이 secure execution enviroment 관련 연구의 대상이 될 수 있다. 기본적인 컴퓨팅 환경의 보안 수준을 높여주는 secure execution enviroment는 이미 많은 연구가 진행된 분야로 본 문서에서는 이와 관련된 연구 및 기술개발 동향에 대해서 알아본다. 또한 대학교에서 진행된 대표적인 프로젝트와 프로세서 제조업체의 기술 동향 및 관련된 산업 표준화 동향을 살펴보고 비교 분석하는 것을 목표로 한다.

# Ⅰ. 서론

컴퓨터 시스템을 비롯하여 정보를 처리하는 여러 가지의 단말들을 해킹의 위협으로부터 보호하는 것은 중요한 보안 이슈 중의 하나이다. 컴퓨팅 환경의 보호를 위해서 진행되고 있는 연구 중에서 Secure Execution Environment(이하 SEE)는 주목할 만한 가치가 있는 분야 중의 하나이다.

실행환경 자체의 보안 수준을 높여주는 것을 목적으로 하는 SEE는 컴퓨팅 환경의 안정성을 높여주는 효과가 있으며, 다른 보안 기술과 병행해서 사용될 수 있는 장점이 있다. 본 문서는 이러한 SEE와 관련된 연구 및 기술 동향을 분석하는 것을 목적으로 작성되었으며, 다음과 같은 순서로 기술될 것이다.

Ⅱ장에서 다루게 될 TPM은 TCG의 신뢰컴퓨팅(trust computing)의 핵심이 되는 모듈이다. 일반적으로 하드웨어 칩으로 구현되는 TPM에 대해서 기술하고, SEE와 유사한 개념인 신뢰컴퓨팅에 관한 산업 표준을 만드는 TCG의 연구동향에서 대해서 알아본다.

Ⅲ장과 Ⅳ장에서 언급할 AEGIS와 XOM은 미국의 대학에서 진행되었던 SEE 관련 프로젝트이다. 두 프로젝트 모두 프로세서를 통한 SEE 구현을 목적으로 하고 있지만, 그 접근방법에는 약간의 차이가 있다.

대표적인 프로세서 관련 업체인 ARM사와 Intel사의 SEE 관련 기술인 TrustZone과 Trust Execution Technology(이하 TET)에 대해서는 Ⅴ장과 Ⅵ장에 걸쳐서 알아본다. TrustZone은 이미 상용화가 된 기술이고 TET는 개발이 진행중인 기술로서, 둘 다 SEE를 제공하기 위한 프로세서 차원의 보안 기술이다.

본 문서는 주로 하드웨어적인 접근방법을 다룬다. 일반적으로 SEE의 경우, 하드웨어 기반의 접근방법이 소프트웨어 기반의 접근 방법에 비해서 안전한 것으로 알려져 있기 때문이다. 물론 본 문서는 하드웨어와 소프트웨어를 모두 사용하는 기술을 포함하고 있으므로 소프트웨어 기반의 접근 방법을 제외한다고는 볼 수 없을 것이다.

**● 용 어 해 설 ●**

Secure Execution Environment: 프로세서, OS 등의 지원을 통해서 프로그램의 안전한 수행을 보장하는 환경을 의미한다. 안전한 수행을 보장하는 방법으로는 무결성, 기밀성 보장 등이 있다.

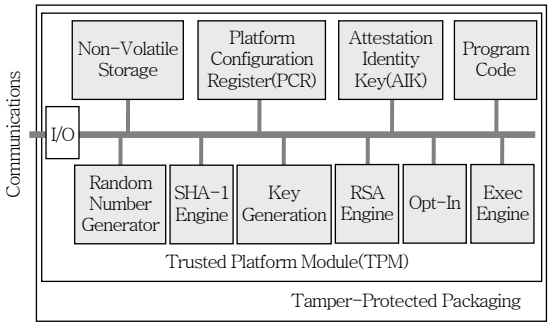
TCG(Trusted Computing Group): 신뢰 컴퓨팅 산업 표준을 개발하고 지원하기 위해 구성된 조직인 TCG는 이 분야의 세계표준을 주도하고 있는 단체이다.

# Ⅱ. TCG와 TPM 기술

TPM[1]은 TCG[2]에서 정의하는 신뢰컴퓨팅을 구축하기 위해서 필요한 여러 하위 기능—기본적인 신뢰 관련 연산—을 제공하는 모듈이다. 신뢰컴퓨팅의 가장 하위에 위치하는 TPM은 훼손 방지(tamper protected)가 필수적이기 때문에 하드웨어 칩으로 구현하는 것이 일반적이지만, 소프트웨어로 구현하는 것도 배제하지 않는다[1]. 칩으로 구현된 TPM은 소프트웨어 방식의 공격과 더불어 물리적인 도난의 경우에도 정보의 노출이 용이하지 않다는 장점을 가진다.

일반적으로 컴퓨터의 마더보드(mother board)에 부착되는 TPM 칩은 통신 채널로 LPC를 이용하며, 디바이스 드라이버를 통해서 상위 기능과 인터페이스를 가진다. (그림 1)은 TPM 내부의 세부 기능을 나타내고 있다. 모듈 내부의 세부 기능을 나타내는 (그림 1)에서 보여지듯이, TPM의 신뢰 관련 연산은 1) 암호화 키의 생성과 저장, 2) 패스워드의 저장, 3) 무결성(integrity)의 검증—플랫폼과 프로그램을 대상으로 하는—을 위한 측정값의 저장, 4) 디지털 인증서 관련 신뢰 연산의 제공 등을 포함한다.

TCG가 구현하고자 하는 신뢰컴퓨팅을 위해서 TPM은 1) 무결성 측정값의 생성 및 저장, 2) 플랫폼의 신원 증명 및 이를 통한 플랫폼 인증, 3) 데이터 저장의 보호 및 안정성 등을 지원하기 위한 핵심 부품과 같은 역할을 한다.




(그림 1) TPM 세부 기능도

TCG가 추구하는 신뢰컴퓨팅은 SEE보다 포괄적인 개념이라고 볼 수 있는데, TCG의 여러 워크그룹(work group)들은 PC 플랫폼을 포함하는 유무선 상의 컴퓨팅 디바이스와 관련된 산업 표준을 만들고 있다. TCG는 기본적으로 TPM이라는 가볍고 저렴한 하드웨어 칩을 바탕으로 신뢰컴퓨팅을 구축하고 있으며, 이를 지원하기 위한 소프트웨어 표준인 TSS[3]도 정의하고 있다.

TCG의 접근방법은 신뢰컴퓨팅을 위한 최소한의 하드웨어인 TPM과 이를 지원하는 TSS의 상호연동에 기초하고 있으며, TSS가 적용된 운영체제가 신뢰컴퓨팅의 많은 부분을 담당하는 형태로 파악할 수 있다.

## Ⅲ. AEGIS 프로젝트

### 1. AEGIS 프로젝트의 개요

미국 MIT대의 CSAIL에서 수행된 AEGIS[4]-[7] 프로젝트는 물리적인 공격과 소프트웨어 기반의 공격 모두로부터 안전한 컴퓨팅 시스템을 구축하기 위해서 단일 칩 프로세서 구조의 SEE를 제시한다.

“Architecture EnGines for Information Security”의 약어인 AEGIS는 Tamper-Evident(이하 TE)와 Private Tamper-Resistant(이하 PTR)의 환경을 제공하면서도 성능의 저하는 최소로 할 수 있는 구조를 제시하였고, 이를 FPGA 칩으로 구현하였다.

### 2. AEGIS 프로젝트의 내용 및 결과

물리적인 공격에 대해서 프로세서 자체는 안전하고 신뢰할 수 있는 반면에, 외부 메모리나 주변장치(peripheral) 그리고 운영체제는 신뢰할 수 없다는 것이 AEGIS의 기본 가정이다. 애플리케이션의 결함(bug)으로 인한 보안 문제는 AEGIS가 다루는 영역이 아니다. AEGIS는 기존의 프로세서와 운영체제에 몇 가지의 기능을 추가한 형태의 SEE를 구현하는 것을 목적으로 한다.

외부 메모리에 저장되어 있는 데이터의 유효성(validity)은 무결성의 검증으로 확인이 가능하다는 가정을 바탕으로 AEGIS는 TE 환경을 제공한다. TE는 시스템에서 동작하는 프로그램의 비정상 행동을 초래할 수 있는 모든 물리적인 혹은 소프트웨어 기반의 훼손(tampering) 행위를 검증할 수 있다는 것을 의미한다. 해시(hash)와 키(key)를 이용해서 프로그램 실행의 무결성을 검증하는 TE 환경은 프로그램의 초기상태, 인터럽트가 발생하였을 때의 상태, 내외부의 메모리 등을 관리 대상으로 한다.

AEGIS가 제공하는 PTR 환경은 TE 환경에 프로그램이나 데이터를 위한 은닉(privacy) 기능을 추가한 것으로 이해할 수 있다. 훼손 행위를 시도하는 공격자가 프로그램이나 데이터에 대한 정보를 획득하는 것이 불가능하게 만들기 위한 PTR은 AES를 통한 암호화로 구현된다.

AEGIS는 해시나 암호화에 사용되는 키의 안정성을 높이기 위해서 Physical Random Function (Physical Unclonable Function이라고도 사용, 이하 PUF)[4]이라는 기능을 추가하였다. PUF는 일종의 난수 발생장치로 집적회로의 예측 불가능한 지연(delay) 성분을 이용한다. 주어진 집적회로의 지연 성분의 편차는 임의적(random)이고 이를 측정하는 것도 불가능한 것으로 여겨지기 때문에 보안의 측면에서 PUF는 큰 장점을 가지는 것으로 평가할 수 있다.

TE와 PTR 환경을 지원하기 위해서 프로세서와 운영장치의 커널(kernel)에 무결성 및 은닉과 관련

된 기능을 구현하였고 외부메모리의 데이터까지 암호화 기능을 적용시킨 AEGIS는 RTL 수준의 구현을 통해서 그 기능과 성능을 검증하였다. 구현에 따른 성능의 저하는 크지 않은 것으로 평가되었으며[7], 프로세서 및 운영체제의 수정을 통해서 SEE를 지원할 수 있는 구조를 제안한 AEGIS는 신뢰 컴퓨팅과 관련해서 참고할 가치가 있는 프로젝트이다.

## Ⅳ. XOM 프로젝트

### 1. XOM 프로젝트의 개요

미국 Stanford대의 CSL에서는 Copy Protection(이하 CP)과 Tamper Resistance(이하 TR)를 지원하는 XOM 프로젝트를 진행하였다[8]-[10]. 실행전용(execute only) 프로그램이란 암호화되어서 저장되고, 프로세서에 로딩(loading)이 될 때만 복호화되며, 원래 목적 이외에는 사용할 수 없도록 만들어진 프로그램을 의미한다. 실행전용 프로그램의 개념을 구현하기 위해서 XOM은 프로세서와 운영체제의 수정을 최소로 하면서, 복잡도와 성능의 트레이드오프(trade off)를 적절하게 조절하는 것을 목표로 하였다.

### 2. XOM 프로젝트의 내용 및 결과

AEGIS의 경우와 마찬가지로, XOM은 운영체제 및 외부 메모리는 신뢰할 수 없다는 것을 바탕으로 연구를 시작하였다. 따라서 XOM은 CP와 TR의 환경을 제공하기 위해서 프로세서와 운영체제를 수정하는 구조를 제시하였다. XOM은 SimOS라는 MIPS R10000 기반의 프로세서 시뮬레이션 시스템을 이용하여 XOM 구조가 적용된 프로세서를 검증하였고, IRIX 6.5 운영체제를 수정하여 XOM 구조가 적용된 운영체제를 구현하고 그 성능을 평가하였다[8].

실행 전용 메모리(execute only memory)는 실행 전용 코드(execute only code)가 저장되어 있는 메모리를 의미하며, 실행 전용 코드는 코드가 프로세서에서 실행이 될 때만 복호화되고 그 외에는 암호화되어 보안이 유지되는 코드를 말한다. 실행 전용 코드를 위한 프로세서 및 메모리 환경을 구현하는 것이 XOM의 목적이며, 이는 CP와 TR 환경과 같은 맥락이라고 볼 수 있다.

실제로 XOM은 보호하고자 하는 실행 코드를 암호화해서 저장하는 동시에 코드의 해시값도 관리하는데, 이 해시값은 실행 코드의 무결성 보장 및 권한 관리에 사용된다. XOM은 구획(compartment)을 제공하는데, 이 구획은 정보의 입출입을 제한하는 논리적인 범위로 한 구획의 프로그램이나 사용자는 다른 구획의 데이터에 접근할 수 없도록 한다. 구획은 실제 실행환경을 구분하고 분리해주는 역할을 담당하며, 아래에 설명할 TrustZone과 TET 기술에도 이와 비슷한 개념이 적용되어 있다.

XOM은 권한 관리와 구획의 분할에 사용되는 정보의 암/복호화에는 비대칭 암호화 알고리듬을, 실행 코드의 암/복호화에는 대칭 암호화 알고리듬을 사용한다. 이러한 혼합적인 암호화 알고리듬의 사용은 성능과 보안강도를 모두 고려한 것이다.

SEE와 연관되는 CP/TR 환경의 제공을 목표로 하는 XOM은 프로세서와 운영체제의 수정을 통해서 목적을 달성하고자 했으며, 이러한 수정에 따른 비용증가(overhead)가 대체로 30% 이내에 그친다는 실험 결과를 보여준다[7]. AEGIS 보다 먼저 시작된 XOM은 실행코드와 같이 주로 읽기 연산만이 수행되는 데이터를 대상으로 했다는 점에서 AEGIS와 차이를 보여주며, CP/TR 환경은 AEGIS의 TE/PTR 환경과 비슷한 개념으로 볼 수 있다.

## Ⅴ. TrustZone 기술

### 1. TrustZone 기술의 개요

ARM사가 제안하는 TrustZone은 휴대폰, PDA와 같은 장치에 적용되는 임베디드 프로세서를 대상으로 한 보안 구조이다[11]-[14]. TrustZone과 관련된 기능은 프로세서에 구현되며, 프로세서의 동작

모드를 일반 모드와 보안 모드로 구분한다. 프로세서의 '보안 상태 관련 비트(a single secure bit, 이하 s-bit)'가 프로세서와 캐시(cache) 및 MMU의 동작 상태를 구분해주는 역할을 한다.

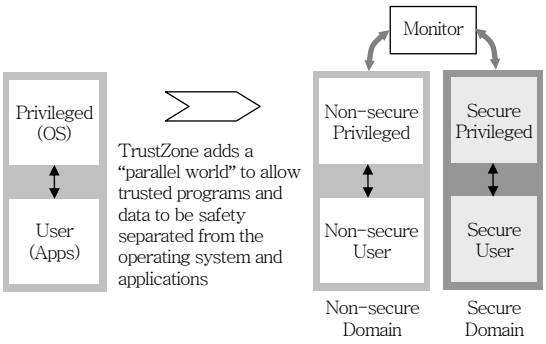
일반 프로세서에 비해서 칩의 크기가 증가하고 전력의 소모가 커지기는 하지만, TrustZone이 구현된 프로세서는 그 위에서 동작하는 소프트웨어가 마치 2개의 분리된 프로세서에서 동작하는 것처럼 보이게 한다. 이러한 추상화(virtualization)를 통해서 사용자는 단일 프로세서 상에서 보안관련 애플리케이션과 일반 애플리케이션을 구분해서 운영할 수 있는 이득을 누릴 수 있다.

## 2. TrustZone 기술의 내용

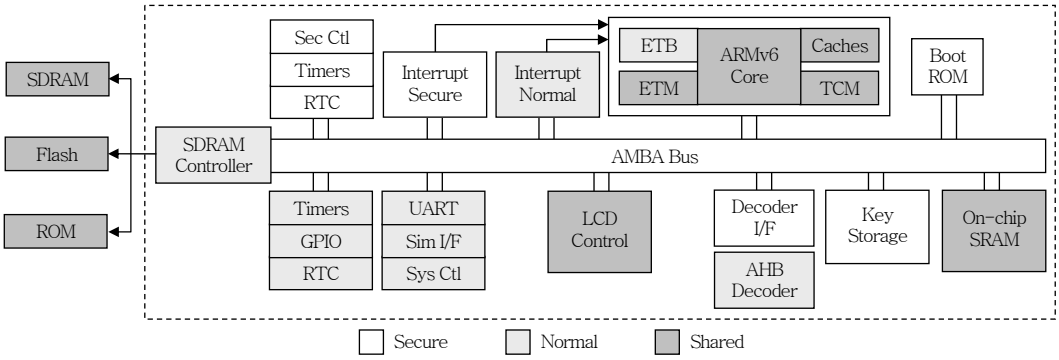
ARMv6 구조에 추가된 TrustZone은 프로세서를 사용하는 임베디드 시스템의 안정성을 보장하기 위해서 그 프로세서와 주변장치 및 저장장치를 대상으로 하는 칩 수준의 보안 기술을 제공하는 것을 목적으로 한다. (그림 2)와 같이 일반 실행 환경과 보안 실행 환경이라는 두 개의 분리된 실행환경을 하나의 프로세서로 제공하며, 이러한 추상화는 시스템의 보안과 직결되는 중요 프로그램의 수행과 일반 프로그램의 수행을 논리적으로 엄격히 분리시켜 준다. TrustZone은 이러한 논리적인 분할을 프로세서뿐만이 아니라 주변장치 및 저장장치를 대상으로도 적용한다. 추상화를 통한 실행 환경의 논리적인 분할은 악의적인 사용자나 프로그램의 위협으로부터 보안과 직결되는 프로그램의 수행이나 시스템의 자원을 보호하는 역할을 할 수 있다.

TrustZone의 프로세서 추상화는 하드웨어만으로 구현되는 것은 아니다. 보안 모드와 일반 모드의 엄격한 분리를 지원하는 하드웨어 기능과 이를 이용하여 기본적인 보안 서비스를 제공할 수 있는 보안 소프트웨어가 연동이 될 때, 일반 운영체제나 애플리케이션의 보안 수준이 향상될 것이다.

TrustZone의 1) 프로세서의 동작모드의 분리, 2) 프로세서의 동작모드를 주변장치 및 저장장치에 전달할 수 있는 버스(bus)상의 s-bit 지원, 3) 인터럽트(interrupt)와 예외상황(exception)의 관리를 위한 보안 모드 인터럽트(Secure Mode Interrupt, SMI), 보안 상태 레지스터(Secure Status Register), 보안 벡터 테이블(Secure Vector Table, SVT)의 지원, 4) 캐시와 메모리의 논리적인 분할을 위한 MMU, TLB의 분리 및 상태 추적 기능 등의 하드웨어 기능을 제공한다. TrustZone 적용 프로세서의 크




(그림 2) 실행환경의 분리[13]




(그림 3) TrustZone이 적용된 SoC의 구조[14]

기나 전력소비량은 일반 프로세서에 비해서 10% 정도 증가하고, 실제 구현을 위해서는 15,000~20,000개의 논리 게이트와 추가 메모리가 필요하다[14].

(그림 3)은 TrustZone 기술이 적용된 SoC의 구현을 설명하고 있다. AMBA bus의 s-bit를 통해서 주변장치와 저장장치의 모든 트랜잭션을 논리적으로 분할할 수 있고, 이 분할을 통해서 보안 관련 장치와 일반 장치의 동작을 엄격히 분리시킬 수가 있다.

TrustZone은 보안을 위한 하나의 완벽한 해결방안이라기 보다는 보안 시스템을 구성하려는 개발자를 위한 하드웨어 기반의 구조 확장(architecture extension)이다. 또한 TrustZone은 보안 기능이 필요한 임베디드 플랫폼의 프로세서 관련 SoC에 적용될 수 있는 기술이며, 아래에 설명할 TET와 더불어 SEE의 지원을 위한 프로세서 수준의 하드웨어 기반 접근 기술 중의 하나라고 볼 수 있다.

## Ⅵ. Trusted Execution Technology 개발 동향

### 1. TET의 개요

Intel이 자사의 IA-32(x86-32) 계열의 프로세서에 적용하기 위해서 개발하고 있는 TET는 플랫폼의 보안 능력(security capability) 향상을 목적으로, 프로세서와 입출력장치를 포함하는 주변장치 그리고 TPM을 연동하는 하드웨어 기반의 기술이다[15],[16]. 보호 실행(protected execution), 봉인 저장(sealed storage), 보호 입력(protected input), 보호 그래픽(protected graphics), 상호 증명(attestation) 등의 기능을 제공하는 TET는 프로세서, 운영체제, 애플리케이션에 적용되어 연동이 될 경우 플랫폼의 보안 수준을 향상시킬 수 있을 것이다.

### 2. TET의 내용

2006년 11월에 사전 구조 명세서(preliminary architecture specification)[17]가 발표된 것으로 볼 때 아직 개발이 진행중인 것으로 여겨지는 TET는 프로세서의 실행 모드를 구분함으로써 논리적인 분할을 제공한다. 일반 모드와 보안 모드라는 용어를 사용하는 TrustZone과 달리 TET는 표준 분할 부분(standard or legacy partition)과 보호 분할부분(protected partition)의 분할을 통해서 '보안이 중요한(security critical)' 프로그램 수행의 안정성을 높여준다. TET는 프로세서에 이벤트 처리(event handling), 보호 분할부분의 관리를 위한 명령어(instruction), 안전한 소프트웨어 스택(stack)의 설정을 위한 명령어 등을 추가하여 하드웨어 자원에 대한 접근 통제를 강화한다.

TET는 1) 메모리 보호 정책을 적용할 수 있는 기능의 제공, 2) 메모리 접근에 대한 보호 기능, 3) 프로세서와 메모리 그리고 입출력 장치 및 그래픽 장치 사이의 통신 채널의 보호, 4) TPM과의 연동을 위한 인터페이스 제공 등의 기능을 제공하는 칩셋을 포함하고 있다.

TET는 ARM사의 TrustZone과 유사한 기술이나 TPM과의 연동이나 상호 증명, 플랫폼의 증명(verification) 등 관점과 범위에서 차이가 있다. TET는 플랫폼, 애플리케이션 등의 증명과 이를 위한 측정 기준(measurement)의 생성/보관/처리하고 있으며, 이를 이용한 인증을 포함하고 있다. 이는 TCG의 TPM과 같은 패러다임을 가지는 것으로 볼 수 있다.

## Ⅶ. 결론

본 문서는 SEE 분야의 전반적인 동향에 대해서 살펴보았다. 기존의 PC 기반 플랫폼에 대해서는 많은 연구가 진행되었고 이 분야의 선두 업체들은 이미 관련된 제품을 출시하고 있는 상황이라고 판단된다. 프로세서를 가지고 있는 각종의 컴퓨팅 단말에도 SEE가 적용될 수 있으며, 특히 현재 많은 주목을 받고 있는 임베디드 시스템의 보안과 관련하여 많은 연구가 진행되고 있는 상황이다.

자원과 전원의 소비에서 제한적일 수 밖에 없는 임베디드 장치의 특성을 고려한 SEE 관련 연구의

중요성은 계속해서 커질 전망이어서 장치의 보안수준을 근본적으로 개선시켜 줄 수 있는 SEE 관련 기술개발은 현시점에서 시급한 과제 중의 하나라고 볼 수 있다.

## 약 어 정 리

AES Advanced Encryption Standard
AMBA Advanced Microcontroller Bus Architecture
CSAIL Computer Science and Artificial Intelligence Laboratory
CSL Computer System Laboratory
EEPROM Electrically Erasable Programmable Read-Only Memory
FPGA Field-Programmable Gate Array
GSM Global System for Mobile communication
IA-32 Intel Architecture, 32-bit
LPC Low Pin Count
MMU Memory Management Unit
PDA Personal Digital Assistant
RTL Register Transfer Level
SIM Subscriber Identity Module
SoC System on a Chip
TCG Trusted Computing Group
TPM Trusted Platform Module
TSS Trusted Software Stack
UICC Universal Integrated Circuit Card
UMTS Universal Mobile Telecommunication System
USIM Universal Subscriber Identity Module
XOM eXecute Only Memory

## 참 고 문 헌


[1] TPM Work Group, http://www.trustedcomputinggroup.org/groups/tpm/

[2] Trusted Computing Group, http://www.trustedcomputinggroup.org

[3] TSS Work Group, http://www.trustedcomputinggroup.org/groups/software/

[4] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon Physical Random Functions," *Proc. 9th ACM Conf. Computer and Communications Security*, Nov. 2002.

[5] G. Suh, D. Clarke, B. Gassend, M. Van Dijk, and S. Devadas, "AEGIS: Architecture for Tamper-evident and Tamper Resistant Processing," *Proc. 17th Int'l Conf. on Supercomputing*, June 2003.

[6] G. Suh, "AEGIS: A Single-Chip Secure Processor," PhD thesis, Massachusetts Institute of Technology, 2005.

[7] G. Suh, C. O'Donnell, I. Sachdev, and S. Devadas, "Design and Implementation of the AEGIS Single-Chip Secure Processor Using Physical Random Functions," Technical Report, MIT CSAIL CSG Technical Memo 483, Nov. 2004.

[8] D. Lie, C. Tekkath, M. Mitchell, P. Lincoln, D. Boneh, J. Mitchell, and M. Horowitz, "Architectural Support for Copy and Tamper Resistant Software," *Proc. 9th Int'l Conf. Architectural Support for Programming Languages and Operating Systems*, Nov. 2000.

[9] D. Lie, C. Thekkath, and M. Horowitz, "Implementing an Untrusted Operating System on Trusted Hardware," *19th ACM Symp. on Operating Systems Principles*, Oct. 2003.

[10] D. Lie, J. Mitchell, C. Tekkath, and M. Horowitz, "Specifying and Verifying Hardware for Tamper-resistant Software," *IEEE Symp. on Security and Privacy*, 2003.

[11] ARM, "Designing with TrustZone - Hardware Requirements," ARM white paper, http://www.arm.com/pdfs/TrustZone_Hardware_Requirements.pdf

[12] ARM, "Secure Software Development with the TrustZone Software API," ARM white paper, http://www.arm.com/pdfs/Secure_Soft_Dev_TZSoftware_API.pdf

[13] T. Alves and D. Felton, "Trustzone: Integrated Hardware and Software Security," ARM white paper, July 2004.

[14] T. Halfhill, "ARM Dons Armor: TrustZone Security Extensions Strengthen ARMv6 Architecture," Microprocessor Report, 2003.

[15] Intel, "Trusted Execution Technology Architectural Overview," Intel white paper, http://www.intel.com/technology/security/downloads/arch-overview.pdf

[16] Intel, "Trusted Execution Technology Overview," Intel white paper, http://www.intel.com/technology/security/downloads/trusted_exec_tech_over.pdf

[17] Intel, "Trusted Execution Technology - Preliminary Architecture Specification," http://www.intel.com/technology/security/downloads/315168.htm