ARM TrustZone 기반 신뢰실행환경의 취약점과 방어기법에 대한 연구

유준승*, 서지원*, 방인영*, 백윤흥*
*서울대학교 전기정보공학부, 반도체공동연구소
jsyou@sor.snu.ac.kr, jwseo@sor.snu.ac.kr, iybang@sor.snu.ac.kr, ypaek@snu.ac.kr

A Study on Vulnerabilities and Defense Systems of ARM TruztZone-assisted Trusted Execution Environment

Jun-Seung You*, Jiwon Seo*, In-young Bang*, Yunheung Paek*
*Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center (ISRC),
Seoul National University

요 약

현재 전 세계 수많은 모바일 기기들은 보안에 민감한 애플리케이션들과 운영체제 요소들을 보호하기 위하여 ARM TrustZone 기반 신뢰실행환경 (Trusted Execution Environment) 을 사용한다. 하지만, 신뢰실행환경이 제공하는 높은 보안성에도 불구하고, 이에 대한 성공적인 공격 사례들이 지속적으로 확인되고 있다. 본 논문에서는 이러한 공격들을 가능하게 하는 ARM TrustZone 기반 신뢰실행환경의 취약점들을 소개한다. 이와 더불어 취약점들을 보완하기 위한 다양한 방어 기법 연구에 대해 살펴본다.

1. 서론

신뢰실행환경(Trusted Execution Environment)은 최근 애플리케이션들의 무결성과 기밀성을 보호하기 위한 핵심 보안 기법으로 떠오르고 있다. 해당 기법은 전용 하드웨어를 사용하여 보안에 민감한 애플리케이션들을 시스템의 운영체제로부터 격리된 보호구역에서 실행하는 기능을 제공한다. 다양한 프로세서 제조사들(AMD, ARM, Intel, IBM 등)이 신뢰실행환경 기능을 제공하는 가운데, 모바일 및 IoT 시장에서 가장 많이 사용되는 ARM 프로세서의 ARM TrustZone[1] 기술이 모바일 기기들에 신뢰실행환경을 제공하기 위해 활발히 도입되고 있다.

ARM TrustZone 기반 신뢰실행환경이 제공하는 높은 보안 수준에 힘입어 해당 기술은 사용자 인증, 온라인 뱅킹 등 보안에 민감한 다양한 애플리케이션들을 보호하기 위해 채택되었다. 안타깝게도, ARM TrustZone 기반 보안 시스템들에 대한 성공적인 공격 사례들은 지난 몇 년 동안 지속해서 발견되고 있다. 이러한 공격들은 ARM TrustZone 기반 시스템들이 지니는 취약점들(신뢰실행환경의 큰 코드 베이스, 격리된 보호구역의 관리 방법 등)에 기인한다.

본 논문에서는 ARM TrustZone 기반 신뢰실행환경이 지니는 취약점들을 분석한다. 이와 함께 취약점들을 해결하기 위해 사용되고 있는 방어 기법들을 살펴본다.

2. 배경이론

2.1 신뢰실행환경과 ARM TrustZone

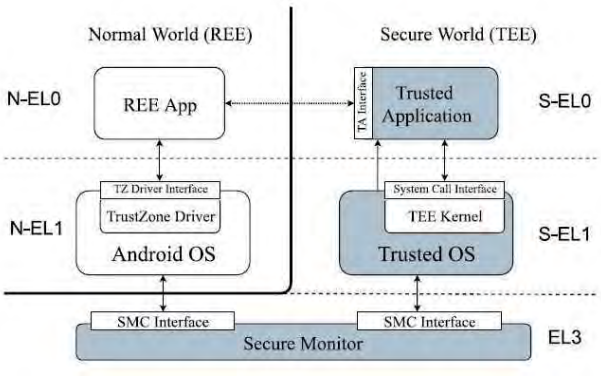
신뢰실행환경은 본래 기존 운영체제를 통하는 민감한 데이터를 격리된 환경에서 처리한다. 즉, 신뢰실행환경은 TA(Trusted Application)라고 불리는 프로그램들의 안전한 실행 보장을 목표로 한다. 신뢰실행환경은 하드웨어 기술을 기반으로 하며, 해당 기술 중 하나인 ARM TrustZone은 2004년부터[2] ARM Cortex-A 프로세서들에 제공되었으며, 최근 Cortex-M[3] 프로세서들에 제공을 위해 재개발되었다.

TrustZone은 'secure world(SW)'와 'normal world(NW)'라는 두 구역으로 운영된다. 각 물리적 프로세서는 하드웨어를 통해 구역별로 가상 프로세서를 할당하고 다른 구역으로의 안전한 전환을 가능케 한다. 시스템의 상태는 프로세서의 NS 비트를

통해 어느 구역에 있는지 확인되며, SW에 있는 자원들은 NW에서 접근할 수 없다.

2.2 ARM TrustZone의 소프트웨어 구조

ARM TrustZone의 기본적인 소프트웨어 구조는 그림1과 같다. 앞서 언급되었듯이 기본적으로 두



(그림 1) ARM TrustZone의 소프트웨어 구조

구역(normal world와 secure world)으로 운영되며, 각 구역은 3가지 실행 레벨(Execution Level; EL)을 가진다. 실행 레벨이 높을수록 구역에서 더 높은 권한을 가지며, 보통 EL0에서 애플리케이션들이 실행되며, EL1에서는 해당 구역의 운영체제가 실행된다. Secure world의 운영체제는 해당 구역에서 애플리케이션이 실행되기 위한 기본적인 기능들뿐만 아니라 암호 라이브러리, 신뢰 I/O 등의 신뢰실행환경 동작에 필요한 기능들을 제공한다. 가장 높은 실행 레벨인 EL3에는 secure monitor이라 불리는 소프트웨어가 존재하여 두 구역 간의 안전한 전환을 제공하며, 이는 SMC(Secure Monitor Call) 명령어를 통해 두 구역에서 접근할 수 있다. Secure world의 운영체제와 secure monitor를 합쳐 신뢰실행환경 시스템의 TCB(Trusted Computing Base)라 일컫는다.

3. ARM TrustZone 기반 신뢰실행환경 취약점들

3.1 신뢰실행환경 공격 반경

신뢰실행환경은 기본적으로 작은 TCB를 가정한다. 다시 말해 격리된 보호구역에서 작동하는 요소들을 최대한 줄이고자 한다. 격리된 구역(secure world)에 많은 요소가 들어갈수록 해당 구역 내에서 작동하는 애플리케이션들이 공격할 수 있는 부분들이 증가할 뿐만 아니라, 다른 구역(normal world)에서 격리된 구역으로 접근할 통로들도 증가하기 때문이다. 하지만 현재 ARM TrustZone 기반 신뢰실행환경 시스템들은 너무 큰 TCB를 지닌다. 즉, secure world에서 작동하는 애플리케이션들 및 운영체제의 크기가 공격하기에 충분하다. 이는 다양한 ARM TrustZone 기반 신뢰실행환경 시스템들의 크기와 시스템에서 발견된 취약점들의 개수(표1)를 통해 확인할 수 있다.

<표 1> 다양한 신뢰실행환경 크기 및 취약점 개수

신뢰실행환경	바이너리	CVE
Qualcomm TEE	1.61 MB	92
Trustonic TEE	350 KB	5
Huawei TEE	744 KB	3
Nvidia TEE	97 KB	10
Linaro TEE	365 KB	3

3.2 Secure, normal world 간의 격리

신뢰실행환경은 SW와 NW 사이를 강력하게 격리함과 동시에 두 구역 사이의 안전한 통신을 제공해야 한다. 하지만 격리 체계는 secure world에서의 시스템 콜 등에 의해서 우회될 수 있다. 문제는 SW에서 실행되는 애플리케이션들이 NW의 물리 메모리를 변조시킬 수 있다는 점이다. 즉, 두 구역 간의 통신을 위해 공유 메모리가 있을 때, SW에서 실행되는 프로그램이 NW의 프로그램보다 더 높은 권한을 지니고 공유 메모리를 변조시킬 수 있다. 예를 들어, Qualcomm TEE에서는 SW에서 실행되는 프로그램이 특정 시스템 콜을 사용하여 NW의 OS 커널이 관리하는 물리 메모리 할당을 허용한다. 그뿐만 아니라 신뢰실행환경 디버깅 메커니즘의 허점을 통해 두 구역 간의 격리가 우회될 수 있다. 디버깅 과정에서 나오는 중요 정보들(스택 트레이스, 로그 등)이 NW의 메모리로 전달되어 시스템 관련 중요 정보들이 취약해진다.

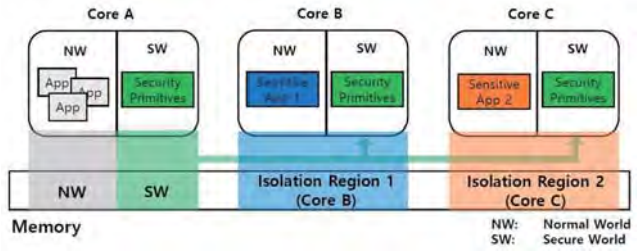
3.3 메모리 보호 메커니즘

많은 신뢰실행환경 시스템들은 취약한 메모리 보호 메커니즘을 지니고 있거나, 아예 보호 장치가 없다. 대표적으로 기본적인 메모리 보호 기법인 ASLR(Address Space Layout Randomization)은 많은 신뢰실행환경 시스템에 제대로 구현되어 있지 않다. 이와 더불어 스택 쿠키나 guard pages 등의 추가적인 메모리 보호 기법들 또한 구현되어 있지 않다.

4. ARM TrustZone 기반 신뢰실행환경 보호기법

4.1 다중 격리 환경

3.1에서 분석한 신뢰실행환경의 넓은 공격 반경에 의한 취약점의 근본적인 문제점은 하나의 구역(secure world)에 너무 많은 애플리케이션 및 신뢰실행환경 구성 요소들이 밀집되어 있다는 점이다. 이를 해결하기 위한 보호 기법은 그림2처럼 현재 하나의 거대한 구역으로 운영되고 있는 격리 보호구역을 나누어 다중 격리 환경으로 운영한다. 즉,



(그림 2) 다중 격리 환경의 기본적인 구조

여러 개의 애플리케이션과 운영체제가 한 구역에서 같이 실행되지 않고, 애플리케이션별로 서로 다른 격리 보호구역에서 실행된다. SANCTUARY[4]는 ARM TrustZone이 제공하는 TZASC(TrustZone Address Space Controller)를 사용하여 다중 격리 환경을 구현하며, vTZ[5]는 잘 사용되지 않는 실행 레벨(EL2)과 하드웨어 가상화 기법을 사용하여 이를 구현한다.

4.2 구역 간 안전한 통신 채널 및 격리

3.2에서 소개한 두 구역(secure, normal world)간의 격리 우회는 근본적으로 두 구역간의 통신 채널의 허점에 기인한다. 즉, NW에서 SW의 메모리에 접근할 때 충분한 인증 작업이 이루어지지 않고, 두 구역간의 메모리 공유 메커니즘이 불안전하다. 이러한 두 구역 간의 통신 취약점은 normal world에 있는 애플리케이션과 secure world에 있는 애플리케이션이 통신할 때마다 세션 키를 사용하여 보호할 수 있다. 즉, 특정 애플리케이션들만 알고 있는 키를 사용하여 상호 통신을 암호화하여 외부 공격을 방어한다. 또 다른 방어 기법은 구역 간 통신을 공유 메모리 없이, 즉 데이터 복사로, 구현한다.

4.3 암호화된 메모리

3.3에서 소개한 메모리 보호 기법의 부재는 기본적으로 기존의 보호 기법을 구현하여 해결할 수 있다. 메모리 보호 기법이 오작동하기보다는 구현이 되어있지 않기 때문이다. 하지만 이와 함께 추가로 구현할 수 있는 방어 기법도 있다. 해당 기법은 secure world에서 보호되고 있는 데이터를 암호화한다. 이는 이미 다른 신뢰실행환경 기술 (Intel SGX)에서 구현되어 있지만, ARM TrustZone에서는 제공되지 않는 기능이다. Secure world에 있는 데이터를 모두 암호화하여 보관하고, 데이터를 처리할 때만 격리 보호구역 안에서 해독하면 normal world에서 secure world의 메모리에 접근할 수 있더라도 데이터들은 안전하다.

5. 추가 취약점들 및 보안 강화 방안

앞서 분석한 ARM TrustZone 기반 신뢰실행환경의 취약점들은 가장 대표적인 약점들 및 방어 기법들이며, 다른 취약점들도 존재한다. 대표적으로는 부채널 공격으로부터의 취약점들이다. 해당 취약점들은 근본적으로 하드웨어 설계 자체의 문제인 경우가 많아 큰 비중을 두고 다루지 않았지만, 캐시나 분기 예측기, DRAM을 이용한 부채널 공격들[6-8]이 존재한다. 이를 방어하기 위해서 하드웨어 명령어를 추가하거나 캐시를 빈번하게 플러시하는 방어 기법들이 존재하지만, 최적화되어 설계된 하드웨어를 변형시키기 때문에 감수해야 하는 오버헤드가 크다.

이와 더불어 ARM TrustZone이 제공하지 않는 신뢰실행환경 기능들에 의한 취약점들이 존재한다. 예를 들면, 앞서 언급한 데이터 암호화 기능이나 원격 증명(remote attestation) 기능의 부재가 있다. 원격 증명이란 원격 신뢰실행환경끼리 상호 간의 신뢰를 형성하는 메커니즘인데, Intel SGX는 제공하지만, ARM TrustZone은 제공하지 않는 기능이다. 즉, 원격에 있는 ARM TrustZone 기반 신뢰실행환경은 상대방도 신뢰실행환경이라고 착각하고 민감한 데이터들을 통신할 수 있다. 이를 방어하기 위해서는 ARM 프로세서의 고유 키나 특성을 가지고 상호 인증할 수 있는 메커니즘을 고안해야 할 것이다.

6. 결론

본 논문에서는 ARM TrustZone 기반 신뢰실행환경이 지니는 취약점들을 분석하고, 이에 대한 방어 기법들을 살펴보았다. ARM TrustZone의 통합적 격리 메커니즘으로 인한 공격 반경 증가, 격리 보호구역과 일반 구역간의 안전한 통신 메커니즘 부재, 그리고 메모리 보호 기법 부재로 인한 취약점들이 존재한다. 이를 방어하기 위해 다중 격리 환경, 세션 키를 활용한 안전한 통신 메커니즘, 메모리 암호화 등의 방어 기법을 사용한다. 많은 취약점에 대한 방

어 기법이 존재하지만, 하드웨어 부채널 공격이나 제공되지 않는 신뢰실행환경 기능으로 인한 취약점들은 아직 ARM TrustZone 기반 신뢰실행환경에 위협적이다. 이를 위해 오버헤드를 줄여 실용적인 방어 기법 연구 및 제공되지 않는 신뢰실행환경 구현 연구가 계속되어야 할 것이다.

7. ACKNOWLEDGEMENT

본 연구는 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단 (NRF-2017R1A2A1A17069478), 2020년도 두뇌한국21플러스사업 , 2020년도 정부 과학기술정보통신부의 재원으로 정보통신기술진흥센터 (No.2017-0-00213, 능동적 사전보안을 위한 사이버 자가변이 기술 개발)의 지원을 받아 수행된 연구임.

참고문헌

[1] Arm, “ARM Security Technology. Building a Secure System using TrustZone Technology ARM,” Arm whitepaper, p. 108, 2009.

[2] T. Alves and D. Felton, “TrustZone: Integrated Hardware and Software Security,” Tech. In-Depth, vol. 3, no. 4, pp. 18 - 24, 2004.

[3] S. Pinto, H. Araújo, D. Oliveira, J. Martins, and A. Tavares, “Virtualization on TrustZone-enabled Microcontrollers? Voilà!” in 25th IEEE IEEE Real-Time and Embedded Technology and Applications Symposium, Montreal, Canada, 2019.

[4] F. Brasser, D. Gens, P. Jauernig, A.-R. Sadeghi, and E. Stapf, “SANCTUARY: ARMing TrustZone with Userspace Enclaves,” in Network and Distributed Systems Security (NDSS) Symposium, 2019.

[5] Z. Hua, J. Gu, Y. Xia, H. Chen, B. Zang, and H. Guan, “vTZ: Virtualizing ARM TrustZone,” in USENIX Security Symposium. Vancouver, BC. 2017, pp. 541 - 556.

[6] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice, and S. Mangard, “ARMageddon: Cache Attacks on Mobile Devices,” in USENIX Conference on Security Symposium. Denver: 2016, pp. 549 - 564.

[7] S. Lee, M.-W. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, “Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing,” in 26th USENIX Security Symposium (USENIX Security 17). Vancouver, BC: USENIX Association, 2017, pp. 557 - 574.

[8] V. van der Veen, Y. Fratantonio, M. Lindorfer, D. Gruss, C. Maurice, G. Vigna, H. Bos, K. Razavi, and C. Giuffrida, “Drammer: Deterministic Rowhammer Attacks on Mobile Platforms,” in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM, 2016, pp. 1675 - 1689.