임베디드 기기용 프로세서의 아키텍처 보안 기능에 대한 연구

박성환*, 김영주*, 권동현**
*부산대학교 정보융합공학과
**부산대학교 정보컴퓨터공학부
starjara@pusan.ac.kr, y0ungjupnu@pusan.ac.kr, kwondh@pusan.ac.kr

A Study on the Security Features of Processor Architecture for Embedded Devices

Seong-Hwan Park*, Young-Ju Kim*, Dong-Hyun Kwon**
*Dept. of Information Convergence Engineering, Pusan National University
**Dept. of School of Computer Science and Engineering, Pusan National University

요 약

사물 인터넷 기술의 발전에 따라 생활 곳곳에서 여러 가지 임베디드 기기들을 찾아볼 수 있게 되었다. 하지만 임베디드 기기의 보급이 늘어감에 따라 이러한 임베디드 기기를 노린 공격도 함께 늘어가고 있다. 이에 따라 임베디드 기기의 보안에 대한 다양한 연구들이 진행되고 있는데 본 연구에서는 대다수의 임베디드 기기에 적용된 RISC 아키텍처 기반 보안 기술에 대하여 살펴보고 향후 발전 방향에 대해 살펴보도록 하겠다.

1. 서론

사물 인터넷의 발전으로 임베디드 시스템의 수요가 늘어감에 따라 생활 곳곳에서 임베디드 기기들을 찾아볼 수 있게 되었다. 그에 따라 임베디드 기기를 목표로 한 공격이 증가하게 되었으며, 이는 임베디드 기기의 보안을 중요한 문제로 자리하게 했다. 그러나 임베디드 기기는 낮은 가격으로 특정한 기능만을 수행하기 위해 만들어진 특성상 낮은 성능과 제한된 크기를 가질 수밖에 없고, 이러한 특징으로 인해 일반 데스크탑이나 서버에 적용된 보안 기능을 그대로 탑재하기 어렵다는 문제가 있었다.

그러나 최근에는 이러한 한계를 극복하기 위해 임베디드 기기용 프로세서들에도 보안 기술들이 늘어가고 있으며 대표적으로 메모리안전을 위한 기술과 제어 흐름 무결성 검증을 위한 기술 그리고 실행환경 격리를 통한 애플리케이션 보호 및 데이터 보호 기능 등이 있다. 예를 들어 ARM 아키텍처에서는 메모리 보호를 위한 MTE(memory tagging extension)[1], 제어 흐름 무결성 검증에는 BTI(branch target identification)[1], TEE(trusted execution environment)[2]를 이용한 실행환경 격리를 통해 데이터를 보호하는 기법인 TrustZone[3]이 존재한다. 한편 최근 부상하고 있는 RISC-V 아키텍처에서는 MultiZone[4]이라고 하는 TEE기술 등을 대표적인 예로들 수 있을 것이다.

본 연구에서는 현재 널리 사용되는 RISC 아키텍처인 ARM과 활발하게 개발이 진행 중인 RISC-V의 대표적인 보안 기술에 대하여 살펴보고 차이를 비교해본 뒤 향후 발전 방향에 대해 논하도록 하겠다.

2. ARM 아키텍처의 보안 기능

현재 임베디드 시스템에서 가장 널리 사용되고 있는 프로세서 아키텍처인 ARM은 현재 스마트폰 등의 모바일 기기부터 IoT기기 및 공장 자동화와 같은 산업현장에까지 널리 쓰이고 있다. 이러한 ARM이 제공하는 보안 기능들에 대하여 이번 장에서 살펴보도록 하겠다.

2.1 ARM Memory Tagging Extension (MTE)

메모리 보호를 위해 제공하는 기능인 MTE는 메모

리 영역에 태그를 부여하는 방식으로 메모리의 접근 권한에 대해 관리한다. 구체적으로 이러한 메모리 태그는 모든 메모리 16바이트 마다 4비트의 메모리 태그가 부여된다. 이후 메모리 접근이 발생할 시 메모리 영역이 가지고 있는 Lock인 메모리 태그와 해당 영역을 가리키는 주소에 포함된 Key 즉, 상위 4비트를 비교해 적합성을 판단한다. 이러한 특징으로 인해 16개의 태그를 생성하여 메모리 공간을 관리할 수 있다. 다음의 그림 1은 MTE의 동작을 나타낸 그림으로, 첫 번째와 두 번째 접근의 경우 물리 메모리에 부여된 태그와 해당 메모리 영역을 가리키는 주소가 가지고 있는 태그가 일치하기 때문에 접근이 허용되나, 마지막 시도의 경우 메모리 영역을 가리키는 주소가 가지고 있는 태그의 값과 메모리 영역에 부여된 태그의 값이 각각 달라 접근시도가 차단되는 것을 알 수 있다.

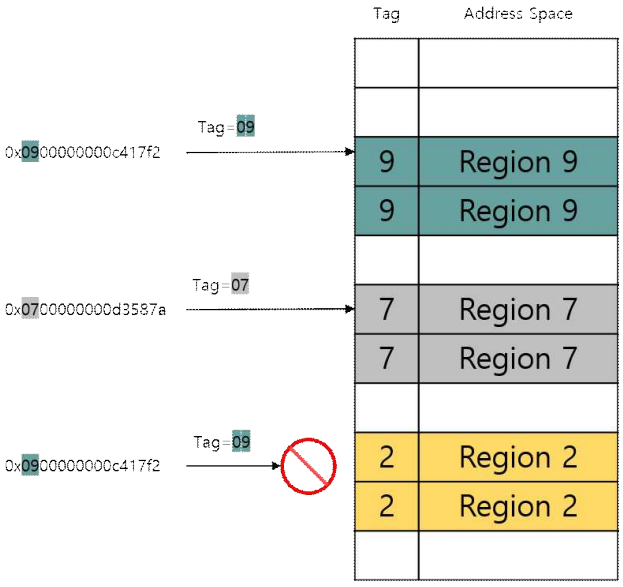

그림 1 ARM-MTE의 예시

2.2 ARM Branch Target Identification (BTI)

제어 흐름 무결성 검증을 위한 기능인 BTI는 간접분기가 일어날 때 해당 분기 직후 처음으로 수행하는 명령어가 BTI라고 하는 명령어가 아닐 경우 예외를 발생시킨다. 이를 통해 코드 생성 시 적법한 코드 위치들에 BTI 명령어를 삽입해 두면 수행 중 간접분기의 대상 주소를 항상 적법한 코드 주소로 강제할 수 있는 것이다. 또한, 간접분기의 유형에 따른 검증도 할 수 있어 결과적으로 적법한 코드 주소를 의도된 간접분기 명령어를 통해 실행할 때만 통과가 되어 수행할 수 있게 해준다.

2.3 ARM TrustZone

TrustZone은 TEE를 제공하기 위한 ARM의 보안 기술로 TEE란 신뢰 가능한 실행환경을 보장하기 위한 기술로 지문인식과 같이 높은 수준의 보안이 요구되는 애플리케이션을 위한 별도의 전용 수행공간을 그림 2와 같이 만들어 일반적인 애플리케이션과 보안이 요구되는 애플리케이션을 격리해 공격 표면을 최소화하고 중요 데이터를 보호하기 위한 기술이다.

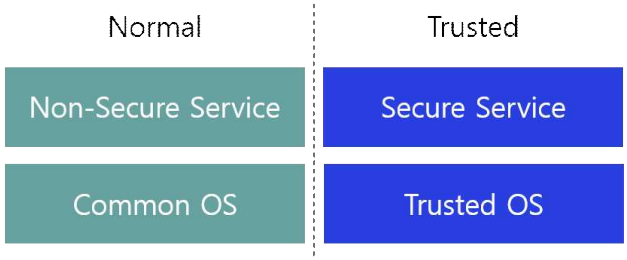

그림 2 TEE 예시

TrustZone은 이러한 TEE를 제공하기 위하여 그림 3과 같이 실행환경을 secure world와 Non-secure world로 나누며 각각의 world에 전용 자원을 할당한다. secure world는 Non-secure world에 비해 높은 권한을 가지기 때문에 Non-secure world에서는 secure world의 자원에 대해 접근이 불가하며 반대로 secure world에서는 Non-secure world로의 접근이 허가되는 특성을 이용해 각각의 실행환경을 격리하고 secure world의 데이터를 보호한다.

한 world에서 다른 world로의 전환을 위해서는 Secure Monitor Call(SMC)이라고 하는 전용 명령어의 호출이 필요하며, 이때 전이되는 world의 context저장 및 복구가 필요하게 되며, 각각의 world는 전용 자원을 소유하기 때문에 서로 간의 통신은 공유 메모리를 통해서 이루어지게 된다. 또한, 가상화를 고려하지 않은 설계로 인해 호스트 상의 모든 가상 머신이 하나의 TEE 커널을 공유하는 특징이 있다. 이러한 특성으로 점으로 미루어 보았을 때 world간의 전환 부하에 유의한 다중 사용 환경에서의 가상화에 대한 해결이 필요하다고 볼 수 있을

것이다.[5]

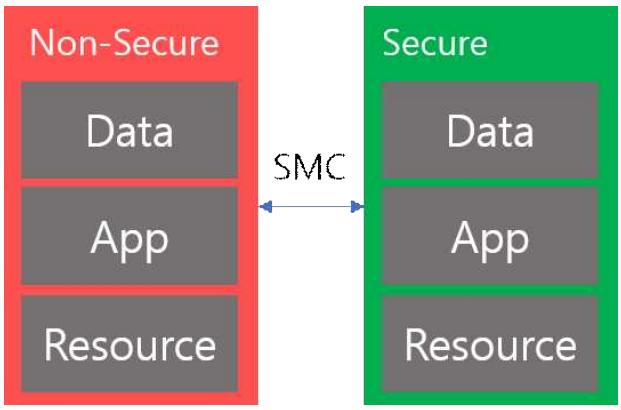

그림 3 TrustZone을 간략화한 도식

3. RISC-V의 보안 아키텍처

RISC-V는 오픈 소스 프로세서 아키텍처로서 최근 학계와 산업계에서 활발하게 연구되는 아키텍처이다. 이번 장에서는 이러한 RISC-V에 제공되고 있는 대표적인 아키텍처 기반 보안 기술들을 살펴보도록 한다.

3.1 RISC-V에서의 메모리 보안 기술

현재 공식 RISC-V 명령어 셋에서는 메모리 보호를 위한 특별한 기능은 따로 제공되지 않으나 오픈 소스 기반의 프로젝트이기 때문에 ARM의 MTE와 유사한 기능인 Tagged Memory를 비롯한 여러 아키텍처 확장을 제공하기 위해 LowRISC[6] 등과 같이 여러 연구가 활발히 진행되고 있다.

3.2 RISC-V에서의 제어 흐름 무결성 검증

최근 RISC-V에서도 return-oriented programming(ROP) attack이 발생할 수 있다는 논문[7]이 발표되는 등 RISC-V 아키텍처에서도 제어 흐름 무결성 보장을 위한 기능 탑재에 대한 필요성이 고조되고 있다. 하지만 아직 공식 RISC-V 명령어 셋에서는 이러한 제어 흐름 무결성 검증을 위한 기능을 제공하고 있지 않다. 대신 FIXER[8]와 같은 논문들에서 RISC-V 프로세서상에서 수행 흐름 무결성 보장 기술을 제안하였는데 이 논문에서는 전용 프로세서 명령어 확장이 아닌 코프로세서 기반의 모니터 방식의 보안 기법을 제안하였다.

3.3 RISC-V MultiZone

MultiZone은 ARM TrustZone과 마찬가지로 RISC-V 아키텍처에서 TEE를 제공하지만 몇 가지 차이점이 존재한다. 구체적으로 그림 4와 같이 여러 도메인을 활용한 격리정책을 지원하며 이는 RISC-V가 마이크로 커널을 기반으로 하여 상위 계층인 Machine Mode(M)와 하위 계층인 User Mode(U)로 각각의 계층을 나누어 실행 및 관리를 하는 특성에서 기인한다. 또한, 나누어진 각각의 영역들은 공유 메모리가 아닌 Secure Message를 주고받으며 통신하기 때문에 완전한 격리가 보장된다.[9][10]

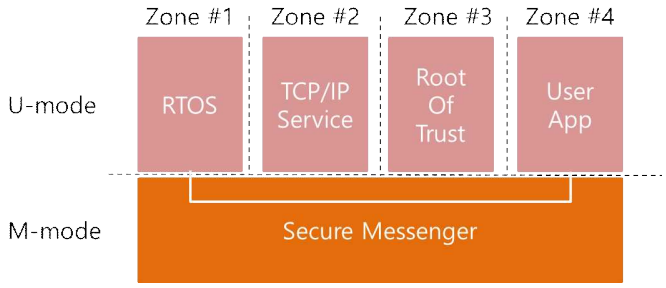

그림 4 MultiZone을 간략화한 도식

4. 결론

아키텍처 / 보호대상	ARM	RISC-V
메모리	MTE	LowRISC (진행중)
제어흐름	BTI	X
TEE	TrustZone	MultiZone

표 1 ARM과 RISC-V의 보안 기술 비교표

위의 표 1은 ARM과 RISC-V 각각에 적용된 보안 기술을 표로 나타낸 것으로 RISC-V의 경우 그 발전 역사가 짧기 때문에 현재 ARM에 비해 보안 기술 측면에서도 아직 개발이 진행 중이거나 일부 학계에서만 연구가 되고 있는 것으로 나타나 앞으로 더 많은 개선이 필요할 것으로 보인다. 즉 RISC-V의 MultiZone같은 경우에는 ARM TrustZone과 달리 복수의 TEE를 만들어 낼 수 있다는 점에서 개선되어 RISC-V를 위한 보안 기능을 연구할 때 기존의 ARM과 같은 아키텍처의 보안 기능을 단점들을 분석하여 더 나은 RISC-V용 보안 기능을 제공할 수 있을 것이다. 한편 ARM의 경우 이미 다양한

보안 기능을 제공하지만 이러한 기능들이 비교적 최근에 공개가 되었기에 앞으로 이러한 기능들을 활용해 다양한 보안 문제에 어떻게 적용할지에 대한 연구가 필요할 것이다.

Acknowledgement

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2020R1G1A1102193).

참고문헌

[1] Learn the architectre: Before debugging on Armv8-A, https://developer.arm.com/documentation/102433/0100/Stack-smashing-and-execution-permissions

[2] M. Sabt, M. Achemlal and A. Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not," 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 2015, pp. 57-64

[3] Learn the architecture: TrustZone for AArch64, https://developer.arm.com/documentation/102418/latest

[4] The First TEE For RISC-V, https://hex-five.com/multizone-security-sdk/

[5] Zhichao Hua and Jinyu Gu and Yubin Xia and Haibo Chen and Binyu Zang and Haibing Guan, "vTZ: Virtualizing ARM TrustZone", Usenix, Vancouver, BC, 2017, 541-556

[6] Tutorial for the v0.4 lowRISC release, https://www.lowrisc.org/docs/minion-v0.4/

[7] Georges-Axel Jaloyan, Konstantinos Markantonakis, Raja Naeem Akram, David Robin, Keith Mayes, and David Naccache. 2020. Return-Oriented Programming on RISC-V. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20). Association for Computing Machinery, New York, NY, USA, 471 - 480.

[8] A. De, A. Basu, S. Ghosh and T. Jaeger, "FIXER: Flow Integrity Extensions for Embedded RISC-V," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 2019, pp. 348-353

[9] G. S. Nicholas, Y. Gui and F. Saqib, "A Survey and Analysis on SoC Platform Security in ARM, Intel and RISC-V Architecture," 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, 2020, pp. 718-721

[10] Pinto, Sandro, and Jose Martins. "The industry-first secure IoT stack for RISC-V: a research project." RISC-V Workshop,(Zurich). 2019.