신뢰 실행 환경을 위한 Processing-In-Memory 격리에 대한 연구

백재원 [1], 조영필 [2]
[1] 한양대학교 정보보안학과 석사과정
[2] 한양대학교 컴퓨터소프트웨어학과 교수

qor291@hanyang.ac.kr, ypcho@hanyang.ac.kr

Research on Processing-In-Memory Isolation for Trusted execution environment

Jae-Won Baek[1], Yeong-Pil Cho [2]
[1]Dept. of Information Security, Hanyang University
[2]Dept. of Computer Software, Hanyang University

요 약

오늘날 데이터 보안이 점점 중요한 요소로 강조되고 있으며, 기술의 발전에 따라 데이터 중심의 워크로드 증가로 인해 메모리 대역폭의 병목 현상의 문제로 데이터 처리 속도의 제약이 발생하고 있다. 이에 따라 Processing-In-Memory(PIM) 라는 새로운 형태의 메모리가 연구 및 개발되고 있다. 본 논문은 새로운 메모리인 PIM 의 취약점을 파악하고 안전하게 데이터를 처리할 수 있도록 TEE(Trusted Execution Environment) 환경을 적용하여 PIM 의 보안성을 강화하는 새로운 보호 체계를 제안한다.

1. 서론

오늘날 기술의 발전에 따라 빅데이터, 인공지능, 자율주행 등의 발전으로 처리되는 데이터 세트의 크기는 계속해서 증가하고 있다. 이에 따라 양의 데이터를 처리하는 능력이 요구되고 있으나 기술의 발전에 발맞춰 데이터 처리 속도 또한 빠르게 증가하고 있다. 하지만 메모리 읽기, 쓰기와 같은 입출력(I/O) 속도는 향상되지 못한 모습을 보여주고 있다. 오늘날의 컴퓨터 아키텍처는 데이터 이동에 소요되는 사이클이 절반 이상이다. 그렇기 때문에 현재 데이터 처리의 병목 문제로 데이터 처리 속도의 제약이 발생한다.

이를 해결하기 위해 Processing-In-Memory(PIM)라는 새로운 형태의 메모리가 연구 및 개발되고 있다. PIM의 아이디어는 앞서 연산을 메모리 내부로 가져와 메모리에 저장된 데이터의 접근 시연 시간을 줄이면서 시스템 버스를 통한 데이터의 이동을 최소화하여 빠른 데이터 처리 연산을 보여주는 것이다. 해당 기술을 유일하게 상용화한 UPMEM사의 PIM을 예시로 들 수 있다. UPMEM 사의 PIM은 메모리 내부에서 연산을 수행할 수 있는 DPU라는 연산기를 각 메모리 칩에 붙여, 데이터를 병렬적으로 메모리 칩 내부에서 처리되도록 하여 기존 메모리의 입출력(I/O) 병목 현상을 완화했다.

PIM은 표준 DDR4 R-DIMM 형태의 DRAM 엔진을 기반으로 한다. 이는 특별한 메모리 보안 메커니즘을 포함하고 있지 않음을 나타내고 보안 문제에 취약하다는 문제를 보여준다. 이는 PIM에서 연산하는 중요한 코드나 데이터의 변조나 유출을 야기할 수 있는 문제이다. 이에 따라 PIM을 외부 환경으로부터 격리를 구현한다. 이는 TEE(Trusted Execution Environment)로 외부 환경으로부터 소프트웨어 실행을 격리하는 하드웨어 수준의 기법이다. 본 논문은 대표적인 TEE 구현인 Intel의 SGX(Software Guard Extension)[1]의 Enclave를 사용하여 PIM의 중요한 데이터를 외부로부터 격리 보호한다. 이러한 하드웨어 수준의 격리는 소프트웨어의 권한 레벨과는 상관없이 이루어지기 때문에 높은 수준의 보안을 제공한다. 하지만 이런 높은 수준의 보안을 보장해도 Enclave 환경과 PIM 간의 취약점이 존재한다.

첫 번째는 Enclave 환경에 격리되어 있는 모든 데이터는 외부 환경에서 볼 수 없어야 한다. 하지만 외부 환경으로부터 격리를 구현해도 Enclave에서 PIM에 접근할 경우 데이터의 변조 및 유출 가능성이 발생하게 된다. 그 이유는 Enclave 환경에서 PIM에 접근을 하기 위해서는 외부에 있는 PIM 라이브러리[4]를 사용해야 한다. 이는 PIM과 통신 인터페이스를 담당하는 소프트웨어로 Enclave에서 PIM으로 전송한 데이터가 외부 PIM 라이브러리를 거쳐야 하므로 데이터 변조 또는 유출의 가능성이 발생하게 된다. 두 번째는 PIM 디바이스는 Enclave 환경에서 격리하는 대상이 아니다. Enclave는 PIM을 사용하기 위한 특정 코드나 데이터를 격리할 뿐 하드웨어 자체를 격리하고 보호

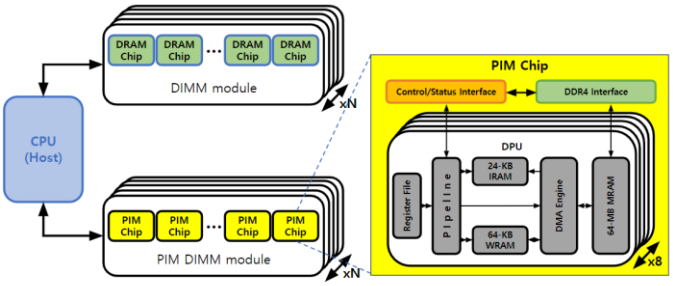
하지는 않는다. 이는 악의적인 사용자가 루트 권한을 얻을 경우 Enclave와 통신하는 PIM 디바이스에 직접적인 접근이 가능하게 된다.

본 논문은 Intel의 SGX Enclave와 같은 TEE 환경과 PIM 간의 안전한 데이터의 전송을 보장하기 위해 PIM 라이브러리의 데이터 송수신 프로세스를 Enclave 내부에서 이를 동작하게 수정하여 외부에 존재하는 PIM 라이브러리를 통하지 않고 Enclave 내부에서 PIM 간의 통신을 구현하고 마이크로 하이퍼바이저를 사용하여 Enclave 환경에서 사용 중인 PIM을 외부 실행환경으로부터 격리하는 연구를 제안한다.

2. 배경지식

2.1. UPMEM PIM

그림1은 호스트(CPU)와 PIM 간의 통신을 나타낸다. 현재 PIM은 2개의 랭크를 가지며, 각 랭크 당 8개의 UPMEM PIM 칩을 포함하고 있다. 각 PIM 칩 내부에



(그림1) UPMEM PIM 아키텍처[5]

는 8개의 DPU으로 구성되며, 각 DPU는 자체적으로 64MB의 메모리 뱅크(MRAM)와 또한 SRAM 기반의 WRAM, 24KB의 크기로 명령어를 저장하는 IRAM을 포함하고 있다.

2.2. Intel SGX

Intel SGX는 사용자 코드와 데이터의 기밀성과 무결성을 보장한다. SGX는 공격에 대한 하드웨어 지원을 통해 주소 공간 내에서 보호된 메모리 영역, Enclave를 제공한다. 이는 외부 환경(운영체제, 하이퍼바이저 등)의 접근이 불가능하여 안전하게 작업을 수행할 수 있는 독립된 보호 구역이다.

2.3. Hypervisor

하이퍼바이저는 호스트 컴퓨터에서 여러 운영 체제가 동시에 실행될 수 있게 하는 논리적인 플랫폼이다. 가상화 기술의 핵심 구성 요소로 가상 머신을 생성, 실행 등 컴퓨터의 하드웨어 자원(CPU, 메모리, 저장소 등)을 가상화하여 각 운영 체제에 할당하고 관리한다. 가상화 환경에서는 하이퍼바이저는 링 0에서 실행되며 가상 머신 내부의 운영 체제는 이보다 낮은 권한을 가진 상태로 실행되어 하이퍼바이저는 이를 제어할 수 있다.

3. 관련 연구

이 섹션에서는 Intel SGX가 가지는 단점에 대해 말하고 이에 대한 이전 연구에 대해 소개한다.

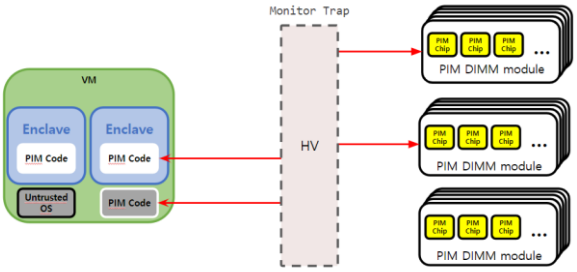
3.1. 보안 도메인 결합과 증명

격리된 실행 환경에 대한 연구는 이전부터 많은 연구의 관심이었다. 특히 Intel SGX는 Enclave와 I/O 디바이스 사이의 사용자 입출력을 보호하는 I/O 경로를 지원하지 않는다. 주로 하이퍼바이저와 같은 운영체제보다 높은 권한의 보안 도메인을 사용하여 입출력을 제어하게 된다. 하지만 Intel SGX와 하이퍼바이저 같은 보안 도메인은 서로 협업하도록 설계되지 않았으며, 서로를 신뢰할 수 있도록 이 두 도메인을 연결하는 것에 대한 노력이 있다[7]. 이에 따라 SGX와 하이퍼바이저 같은 상호 도메인에 신뢰를 바인딩하는 연구가 꾸준히 진행되고 있다. SGXIO[7] 같은 경우는 신뢰 부팅과 TPM(Trusted Platform Module)을 활용한 양방향 바인딩을 시도하는 연구가 있었다. eOPF[9]는 하이퍼바이저를 통해 악의적인 사용자가 Enclave을 실행하는 것을 방지하고 이를 모니터링, 제어를 시도한 연구가 진행되었다.

3.2. GPU TEE

PIM 과 같은 대표적인 가속기는 GPU 가 있다. GEVisor[8]은 기존 하드웨어 수정 없이 안전한 GPU TEE 를 구축하기 위한 연구이다. 루트 권한을 얻은 공격자는 GPU 드라이버를 제어하여 Memory-Mapped I/O(MMIO) 및 Direct Memory Access(DMA) 인터페이스를 통해 GPU 내부의 민감한 데이터에 액세스할 수 있다. 이러한 취약점을 Intel SGX Enclave 를 사용하여 GPU TEE 를 구축하고 하이퍼바이저를 통한 Enclave와 GPU 간의 신뢰할 수 있는 I/O 경로를 구축하는 연구이다.

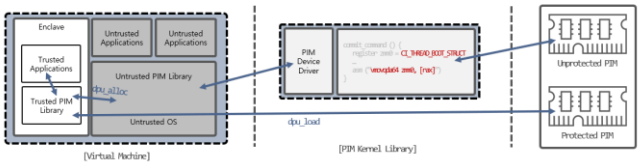
4. 디자인



(그림2) System 아키텍처

그림 2 는 시스템의 전체적인 아키텍처 디자인이다. 신뢰 환경 Enclave 와 신뢰할 수 없는 환경 OS, 사용자 코드는 VM 에서 실행된다. 안전하게 데이터를 보호하는 신뢰 환경 Enclave 의 사용자 코드와 신뢰할 수 없는 환경의 사용자 코드 모두 PIM 디바이스에 접근할 수 있다. 하지만 신뢰 환경에서 사용하는 PIM 디바이스에 관해 신뢰할 수 없는 환경에서의 접근이 가능하기 때문에 이를 하이퍼바이저 VMM 이 트랩하여 정상적인 접근인지 판단하고 접근 허가 또는 거부한다. 본 시스템의 세부적인 디자인은 SGX-PIM 과 PIM-Hypervisor 으로 나뉜다.

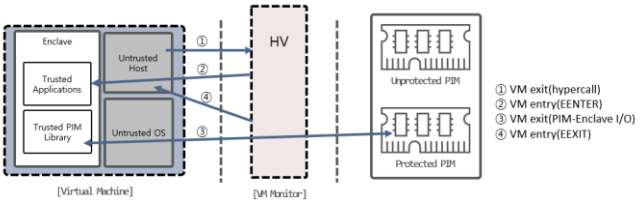
4.1. SGX-PIM



(그림 3) SGX-PIM Design

일반적인 PIM 의 실행은 PIM 라이브러리를 이용하여 PIM 바이너리 코드를 작성하고 DPU 에 적재하여 실행한다. 본 연구에서는 DPU 에 적재하고 안전하게 실행할 데이터와 코드를 Enclave 에서 보호하고 PIM 디바이스 간의 안전한 데이터 전송을 보장하기 위해 PIM 라이브러리의 데이터 전송 로직만 Enclave 내부에 구현한다. PIM 은 대표적으로 할당, 바이너리 로드, 실행 부분으로 나뉜다. PIM 라이브러리 사용이 강제되는 할당은 사용자에게 사용할 PIM 을 할당만 해주기 때문에 데이터의 이동이 없어 문제가 발생하지 않는다. 이러한 할당 과정을 통해 할당받은 DPU 는 해제하지 않는 한 독점적으로 사용할 수 있어 이 이후부터는 특별한 동작 없이는 PIM 라이브러리 사용이 강제되지 않는다. 그러나 데이터의 변조 또는 유출의 여지가 있는 데이터 전송은 PIM 디바이스 간의 통신에서 발생하는데 이는 PIM 라이브러리의 사용이 강제되지 않고 데이터 전송 로직을 Enclave 내부에 구현하여 사용하여도 정상적인 동작이 가능하다. 결과적으로 Enclave 와 PIM 간의 외부로부터의 데이터 노출 없이 통신이 가능하다.

4.2. PIM-Hypervisor



(그림 4) PIM-Hypervisor Design

이전 연구에서 주목된 것처럼 Intel SGX 는 Enclave 와 I/O 디바이스 사이의 사용자 입출력을 보호하는 I/O 경로를 지원하지 않는다. 이는 PIM 디바이스에도 해당한다. SGX-PIM 간의 보호는 Enclave 내부에서 보호되는 PIM 데이터 및 코드를 외부 요소로부터 보호를 적용하지만 PIM 디바이스 간의 입출력은 취약점이 발생한다. 이를 해결하기 위해 마이크로 하이퍼바이저를 사용하여 Enclave 에서 보호 중인 PIM 을 외부 실행환경으로부터 격리한다. 현재 시스템에서 가장 높은 권한을 가진 하이퍼바이저가 Enclave 내부에서 동작하는 PIM 디바이스에는 해당 Enclave 에게 접근 권한을 주고 커널에는 주지 않음으로써, PIM 을 외부 환경으로부터 격리할 수 있다. 이는 공격자가 루트 권한을 획득하더라도 PIM 디바이스는 하이퍼바이저로 제어되기 때문에 접근이 불가능하다. 이에 따라 Enclave 로부터 격리된 PIM 코드는 할당 받은 PIM 디바이스에 관한 접근을 하이퍼바이저로부터 제어 받으며 권한이 없는 신뢰 없는 접근으로부터 제한한다.

5. 기대효과 및 향후 연구

5.1. 보안

PIM 은 데이터 중심의 워크로드 증가로 인한 메모리 대역폭의 병목 현상의 문제를 해결하기 위한 해결책으로 데이터 처리를 가속할 수 있으며 본 논문에서는 이러한 PIM 을 TEE 환경에서 안전하게 데이터 처리를 가속화할 수 있는 보호 체계를 제시한다.

보안 측면에서는 Intel SGX Enclave 와 하이퍼바이저를 사용하여 외부 환경으로의 격리를 구현한다. 루트 권한을 얻은 공격자의 접근으로부터 SGX 의 보안 메커니즘에 따라 안전하게 데이터, 코드 보호가 보장된다. 또한 이전 연구부터 관심 받아온 SGX 의 I/O 경로의 보호는 높은 수준의 권한을 가진 하이퍼바이저를 통해 보호해야 할 I/O 디바이스 간의 접근을 제어하고 모니터링할 수 있어 안전하게 TEE 환경에서 PIM 을 사용할 수 있다.

5.2. 오버헤드

PIM 은 빠른 처리 속도를 목표로 하는 메모리 장치이다. 이 섹션에서는 해당 연구에서 발생하는 시스템 오버헤드를 향후 연구로 남겨둔다.

관련 연구를 살펴보면 시스템 구축 시작점에서Intel SGX Enclave을 구축과 하이퍼바이저와 Enclave 간의 신뢰를 증명하는 과정에서 발생하는 오버헤드는 높을 것으로 예측하고 있다. 또한 추가적으로 Enclave 암호화 과정과 하이퍼바이저로 인한 문맥 전환이 가져오는 높은 오버헤드를 예상하고 있다. 이에 대한 대책으로 GEVisor[8]은 새로운 비동기 하이퍼콜 메커니즘을 제시하기도 하였다. 하지만 SEGIVE[10]은 GPU 사용 오버헤드에 있어 적은 수의 GPU 런타임 응용프로그램에서는 오버헤드는 기존과 차이를 보이지 않고 있으며 16개 이상 응용 프로그램에서 오버헤드를 보이고 있지만 성능 이점이 크게 차이나지 않는다고 말하고 있다. 이 결과로 본 연구에서는 Enclave 내부에 적재한 TEE-PIM 간의 데이터 전송 로직은 기존 PIM SDK[4]에서 구현된 로직 중 불필요한 API를 제거하

고 데이터 전송 부분만을 사용하는 점에 대해 적은 오버헤드를 기대해 볼 수 있으며, SGX와 Hypervisor의 결합에서도 높은 오버헤드를 우려하지 않는 점을 기대해볼 수 있다. 앞서 오버헤드 측정을 향후 연구로 남겨둔다. 빠른 데이터 처리를 필요로 하는 PIM의 이점을 최대화하고 PIM에서 사용하는 데이터를 안전하게 보호할 수 있도록 한다.

6. 결론

데이터 중심의 워크로드 증가로 인해 메모리 대역폭의 병목 현상의 문제로 데이터 처리 속도의 제약이 발생하였다. 이에 대한 해결책으로 Processing-In-Memory(PIM) 기술을 도입하여 데이터 처리 속도를 향상하고 PIM을 더 안전하게 사용하기 위해 PIM을 외부 환경으로부터 격리를 구현하고 PIM의 보안성을 강화해 TEE에서 안전하게 데이터 처리를 가속화하는 새로운 보호 체계를 제안한다.

이 논문은 과학기술정보통신부의 재원으로 정보통신기획평가원(No. 2020-0-01840, 스마트폰의 내부데이터 접근 및 보호 기술 분석)과 한국연구재단(No. NRF-2022R1A4A1032361, Processing-in-Memory 보안 기술 개발)의 지원을 받아 수행된 연구임

참고문헌

[1] Intel Software Guard Extensions ProgrammingReference, Oct. 2014. Reference no. 329298-002US.

[2] Intel Software Guard Extensions Developer Guide,2016.

[3] Intel Software Guard Extensions Evaluation SDK forWindows OS. User's Guide, Jan. 2016. Revision 1.1.1.

[4] UPMEM SAS. 2021. UPMEM SDK.

[5] UPMEM Driver(Kernel Module) 2023.

[6] AFARI Research Group, 2020. Understanding a ModernProcessing-in–Memory Architecture

[7] S. Weiser and M. Werner, "SGXIO: Generic trusted I/O path for IntelSGX," in Proc. 7th ACM Conf. Data Appl. Secur. Privacy, Mar. 2017,pp. 261–268

[8] Wu, Xiaolong, Dave Jing Tian, and Chung Hwan Kim. "Building GPU TEEs using CPU Secure Enclaves with GEVisor." Proceedings of the 2023 ACM Symposium on Cloud Computing . 2023.

[9] Ahmad, Adil, et al. "An Extensible Orchestration and Protection Framework for Confidential Cloud Computing." 17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23) . 2023.

[10] Wang, Ziyang, et al. "SEGIVE: A practical framework of secure GPU execution in virtualization environment." *2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 2020.