정 보 보 호 학 회 지
제 30 권 제6호, 2020. 12

# CPU 마이크로아키텍처 보안 기술 연구 동향

신 영 주*

요 약

CPU 마이크로아키텍처는 하드웨어 자원을 공유하거나 투기적 실행과 비순차 실행 등 파이프라인 효율을 극대화하는 방법을 통해 성능 최적화를 달성한다. 그러나 보안을 고려하지 않은 설계 구조로 인해 마이크로아키텍처에 심각한 보안 취약점들을 내포하고 있으며 이는 각종 시스템 보호 메커니즘들을 무력화할 수 있는 시스템 공격으로 이어지고 있다. 본 논문에서는 CPU 마이크로아키텍처의 취약점 및 이를 활용한 공격 기술을 소개하고 최근 주요 보안 학술대회에서 발표된 관련 논문들을 중심으로 최신 연구 동향을 살펴본다.

## Ⅰ. 서 론

컴퓨터 시스템의 보호 메커니즘은 보안 수준이 다르거나 서로 신뢰 관계가 없는 여러 프로그램 간에 보안 경계를 만들어 다른 보안 영역을 침범하지 않도록 격리를 제공한다. 예를 들어 운영체제 커널은 페이징 메커니즘을 통해 유저모드 프로세스로부터 보호되며, 웹 브라우저는 샌드박싱을 통해 신뢰할 수 없는 자바스크립트 코드로부터 보호된다. 이러한 보호 메커니즘들은 소프트웨어상에 심각한 취약점이 발견되지 않는 한 안전한 시스템 환경을 제공한다.

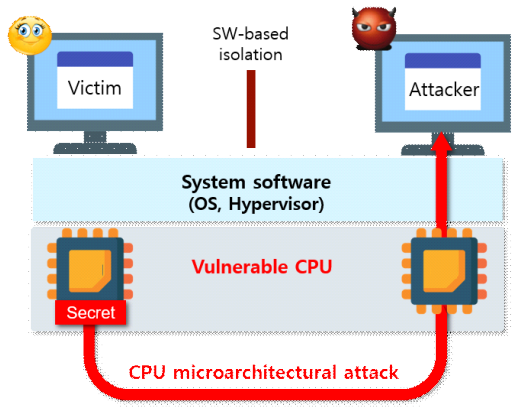
CPU 마이크로아키텍처 공격은 이러한 소프트웨어 방식의 보호 메커니즘들에 의해 만들어진 보안경계를 우회한다. 이를 통해 권한이 없는 비인가자는 다른 보안 영역 내에 존재하는 민감한 비밀정보에 접근할 수 있게 된다(그림 1).

CPU 마이크로아키텍처 공격은 CPU의 설계 및 구현에 내재한 각종 보안 취약점들을 이용한다. 대부분의 마이크로아키텍처는 설계 단계에서부터 보안에 대한 충분한 고려 없이 실행 성능과 하드웨어 자원 활용률 극대화에만 초점을 두어 개발되었다. 이는 캐시와 같은 공유자원에서의 부채널 취약점과 명령어 파이프라인 효율 최적화 과정에서의 심각한 보안 취약점들을 야기하였다.

마이크로아키텍처 내부 구조가 갖는 고도의 복잡성으로 인해 아직 발견되지 못한 보안 취약점들이 여전히 많이 존재할 것으로 보인다. 그래서 최근에는 신규 CPU 취약점을 도출하고 보안성을 강화하기 위한 각종 연구가 활발하게 진행되고 있다.

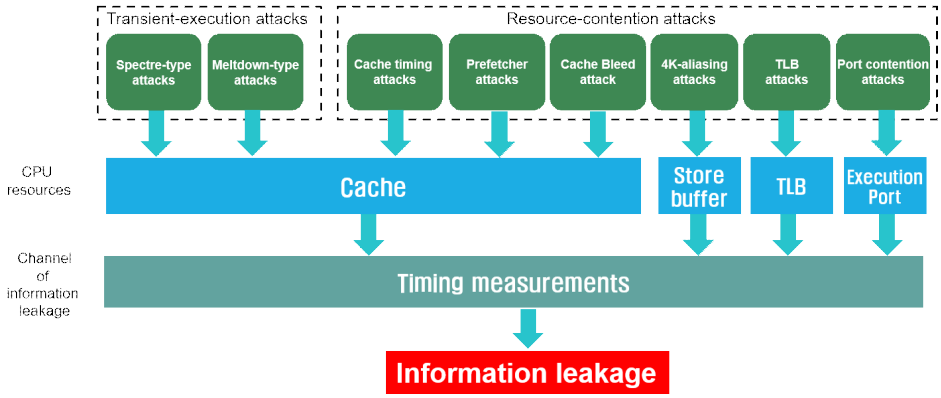
본 논문에서는 CPU 마이크로아키텍처 공격을 소개하고 학계에서 진행하고 있는 CPU 마이크로아키텍처 취약점 및 보안 기술에 관한 최근의 연구 동향을 살펴본다. CPU 마이크로아키텍처 공격은 발견된 보안 취약점의 유형과 그 공격대상에 따라 3가지 유형으로 분류할 수 있다. 유형별로 취약점과 이를 활용한 공격 방법들을 살펴보기로 한다.

현재 CPU 마이크로아키텍처 취약점 및 보안 기술에




〔그림 1〕 CPU 마이크로아키텍처 공격을 통한 보안경계 우회

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1F1A1065539)
* 고려대학교 정보보호대학원 (조교수, syoungjoo@korea.ac.kr)




〔그림 2〕 자원 경합 공격과 일시적 실행 공격

관한 연구는 보안 분야 학계에서 주도하고 있다. 따라서 4대 보안 학술대회인 ACM CCS, IEEE S&P, USENIX Security 그리고 NDSS 에 최근 발표된 연구 논문들을 중심으로 최신 연구 동향을 살펴보도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 3가지 유형의 CPU 마이크로아키텍처 공격을 소개한다. 3장에서는 최근 4대 보안 학술대회에 발표된 CPU 보안 관련 연구 논문들을 소개한다. 마지막으로 4장에서는 본 고의 내용에 대한 요약 및 결론으로 끝을 맺는다.

## Ⅱ. CPU 마이크로아키텍처 공격

CPU 마이크로아키텍처에 대한 주요 공격 기술로는 공유 하드웨어 자원의 경합을 이용한 자원 경합 공격과 투기적 실행 및 비순차 실행의 설계 취약점을 이용한 일시적 실행 공격, 그리고 신뢰 실행 환경을 대상으로 한 공격들이 있다. 본 장에서는 각 공격 기법들에 대해 살펴본다.

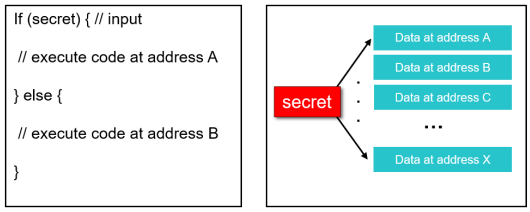
### 2.1. 자원 경합 공격

CPU 내부에는 연산을 위한 실행 유닛들과 데이터 저장에 필요한 캐시 메모리 등 다양한 하드웨어 자원들이 존재한다. 대부분의 CPU 는 이들 자원의 활용률을 극대화하기 위하여 물리적 또는 논리적 코어들 간에 자원을 공유하여 사용하고 있다. 따라서 어느 한쪽 코어에서 공유자원을 많이 사용하게 되는 경우 다른 코어는 프로그램의 실행에 영향을 받을 수밖에 없다.

자원 경합 공격 (Resource contention attack) 은 코어들이 어느 한 공유자원을 두고 선점하면서 발생하는 간섭 현상을 부채널로 활용하여 비밀정보를 획득하는 공격 방법이다 (그림 2). 여러 공유자원 중에 가장 대표적인 것은 CPU 캐시를 활용한 캐시 부채널 공격이 있다. 캐시는 일반적으로 프로그램이 시간적, 공간적 지역성을 갖는 점을 이용하여 자주 사용하는 데이터를 빠르게 접근할 수 있게 해주는 하드웨어 자원이다. 메모리보다 접근속도는 빠른 반면에 극히 작은 저장 공간을 가지고 있어 코어 간에 경합이 빈번하게 발생하는 자원 중의 하나이다.

캐시 부채널 공격 중에 대표적인 기법으로는 Flush+Reload 공격[5]이 있다. 공격자(Spy)와 공격대상(Victim) 이 물리 페이지를 공유하고 있는 조건에서 공격이 이루어진다. 예를 들어 victim 이 1비트 비밀값 s 에 따라 특정 메모리 주소를 접근하는 프로그램이라고 할 때, spy 는 해당 주소에 대응하는 캐시 라인 상태의 관찰을 통해 비밀값 s를 유추해낼 수 있다. 즉, spy 는 캐시 라인의 메모리 주소를 접근할 때 소요되는 시간을 측정하고 그 결과가 기준치보다 작을 경우 이미 victim 이 해당 주소에 접근하였다고 판단할 수 있다. 이러한 방식으로 시간 측정을 통해 캐시 내부 상태를 관찰하고 이를 토대로 비밀정보를 알아낼 수 있다.

최근에 많은 연구를 통해 캐시뿐만 아니라 다른 CPU 하드웨어 자원들에서도 코어 간 경합을 통한 부채널이 존재할 수 있다는 게 밝혀지고 있다. 예를 들어 CPU 백엔드에는 명령어의 유형 별로 6~8개의 실행 포트가 존재하는데, 한 코어에서 특정 포트를 점유하게 되면 이를 공유하는 다른 쪽 코어에서 동일한 명령어를 실행할 때 영향을 받게 된다. 실행 포트에서 발생하는




〔그림 3〕 자원 경합 공격에 취약한 소프트웨어 구현

자원 경합을 이용하여 victim 이 실행하는 명령어의 종류를 유추하고 나아가 비밀정보를 획득하는 것이 가능하다 [3]. 실행 포트 뿐만 아니라 분기 예측기, 프리페칭 유닛, 메모리 서브 시스템 그리고 TLB 등 다른 공유 하드웨어 자원들에서도 자원 경합을 통한 부채널 공격이 가능하다.

OpenSSL 이나 GnuPG 등 암호 구현 소프트웨어들이 자원 경합 공격의 주요 대상이 되어 실제로 AES나 RSA, ECDH 등 암호 알고리즘의 비밀키가 유출될 수 있음이 밝혀지기도 했다. 이 공격에 취약한 소프트웨어들은 공통적으로 그림 3과 같이 비밀값에 의존하여 다른 제어 흐름을 갖거나 다른 메모리 접근 패턴을 가지고 있다.

## 2.2. 일시적 실행 공격

CPU 는 명령어 페치와 디코딩을 담당하는 프론트엔드와 실제 메모리 접근 및 연산을 수행하는 백엔드로 구성되어있다. 하나의 명령어는 프론트엔드와 백엔드에 걸쳐있는 여러 단계의 파이프라인을 따라 처리가 이루어진다. 따라서 CPU의 실행 성능을 극대화하기 위해서는 파이프라인이 유휴상태에 있지 않도록 끊임없이 명령어를 공급해야 한다.

그런데 분기 명령어나 권한 체크가 필요한 일부 명령어들의 경우 분기 위치가 결정되거나 권한 체크가 완료될 때까지는 다음 명령어를 파이프라인에 공급할 수 없다. 이 문제를 해결하기 위해 분기를 예측하여 예측 주소에서 명령어를 가져와 실행하는 투기적 (Speculative) 실행과 권한 체크를 진행하는 동안 순서상 다음 명령어를 먼저 가져와 실행하는 비순차(Out-of-order) 실행기법을 사용하여 파이프라인 효율을 극대화한다.

이렇게 투기적 실행과 비순차 실행을 통해 미리 파이프라인으로 진입한 명령어들은 분기 예측이 적중하거나 실행 권한에 문제가 없는 경우에만 리오더 버퍼 (ROB)에서 커밋되어 그 실행 결과값이 레지스터나 메모리에 반영된다. 그렇지 않은 경우에는 즉시 파이프라인을 비우고 모든 실행을 롤백하게 된다. 그러나 명령어 실행을 롤백하더라도 캐시와 같은 CPU 마이크로아키텍처 내부에는 그 흔적이 일부 남아있게 된다.

일시적 실행 공격 (Transient execution attack) 은 ROB에서 커밋되지 못하고 사라지는 명령어들이 캐시 등에 남겨놓은 흔적을 토대로 캐시 부채널 분석 등의 기법을 통해 victim 의 비밀정보를 획득하는 공격 기법이다 (그림 2). 대표적으로 투기적 실행 과정에서의 취약점을 이용한 Spectre 유형의 공격들과 비순차 실행의 취약점을 이용한 Meltdown 유형의 공격들이 있다 [4].

Spectre 유형의 공격들은 공통적으로 spy 가 victim 과 공유하고 있는 분기 예측기를 활용한다. 먼저, spy 는 분기 예측기를 트레이닝하여 의도한 주소로 victim 의 실행을 분기하도록 유도한다. 이때 프로그램 입력이나 함수 인자 등을 통해 victim 에게 제어정보를 전달하여 투기적 실행이 진행되는 동안 내부의 특정 비밀정보에 접근하도록 제어할 수 있다. Spectre 유형의 공격들은 활용하는 분기 예측기의 종류에 따라 분류되며 대표적으로는 Spectre-PHT (변종 1), Spectre-BTB (변종 2) 그리고 Spectre-RSB (변종 5) 등이 있다.

Meltdown 유형의 공격은 일부 Intel 프로세서 등에서 명령어 권한 체크가 늦게 이루어지는 문제점을 이용한다. 우선 spy 는 커널 또는 존재하지 않는 영역의 메모리 주소에 대해 로드 명령어를 실행한다. 해당 주소 영역에 대한 접근 권한 체크는 이 명령어가 ROB에서 커밋되기 직전에 이루어지는데, 그 사이에 비순차 실행으로 실행된 명령어들이 그 주소에서 읽어 들인 값을 가지고 캐시에 인코딩을 수행한다. 인코딩 된 값은 캐시 부채널 분석을 통해 복구할 수 있다. Meltdown 유형의 공격 들은 보통 사용하는 접근 권한 비트의 종류에 따라 분류되며 대표적으로는 Meltdown-US와 Meltdown-PF (Foreshadow) 그리고 Meltdown-GP (LazyFP) 등이 있다.

## 2.3. 신뢰 실행 환경에 대한 공격

컴퓨터 시스템에 대한 사이버 공격 기술이 갈수록 고도화되고 클라우드 컴퓨팅 등 다양한 컴퓨팅 환경이 출

현하면서 운영체제 커널에 시스템의 신뢰를 전적으로 의존하는 전통적인 신뢰 모델로는 보안 문제를 해결하는 데 한계가 있다. 이에 따라 운영체제에서 CPU 로 신뢰 컴퓨팅 베이스 (TCB) 를 옮겨 CPU 내부에서 안전한 실행 환경을 제공하는 신뢰 실행 환경 (TEE) 기술이 최근에 새롭게 출현하였다.

TEE 는 Intel SGX와 AMD SEV 그리고 ARM 의 TrustZone 등 각 CPU 마다 독자적인 기능과 설계를 가지고 구현되었으나 공통적으로 사용자에게 데이터의 기밀성과 실행 프로그램의 무결성 등 보안 기능을 제공한다. 그러나 TEE 기술은 다양한 보안 위협에 대한 충분한 고려를 하지 않고 설계되어 여러 보안 문제를 안고 있다. 특히, 캐시 부채널 공격 등 고도화되고 있는 CPU 마이크로아키텍처 공격 기술과 시스템의 운영체제 자체가 공격자가 되는 강력한 공격 모델이 결합하여 TCB 의 핵심인 CPU 제조사 마스터키가 유출되는 등 TEE 의 심각한 보안 취약점들이 계속 발견되고 있다.

## Ⅲ. CPU 취약점 및 보안 기술 최신 연구 동향

2020년에 개최된 주요 보안 학술대회에서는 CPU 마이크로아키텍처 보안 기술 관련 주제로 총 16편의 논문들이 발표되었다 (표 1). 본 장에서는 학술대회별로 발표된 논문들을 살펴본다.

### 3.1. NDSS

NDSS‘20 에서는 총 3편의 CPU 마이크로아키텍처 보안 관련 논문들이 발표되었다.

PhantomCache [17] 는 LLC (Last level cache) 에서의 캐시 부채널 공격을 방지하기 위한 캐시 설계 구조를 제안하였다. 기본 아이디어는 LLC의 메모리-캐시 매핑 구조를 랜덤화하여 캐시 경합이 발생하는 메모리 주소를 찾기 어렵도록 하는 것이다.

SPEECHMINER [16] 는 마이크로아키텍처 내부에 일시적 실행 공격을 유발하는 근본적인 취약점을 찾고 또 취약점과 관련된 깊숙한 내부 구조를 이해할 수 있도록 도와주는 분석 자동화 프레임워크를 제안하였다.

ConTeXT [15] 는 일시적 실행 공격을 근본적으로 차단할 수 있는 소프트웨어와 CPU 하드웨어의 새로운 설계 구조를 제안하였다. CPU 에 non-transient bit 플래그만 추가하는 최소화된 형태의 설계 변경만으로 성능 저하 없이 공격을 차단할 수 있음을 보였다.

### 3.2. IEEE S&P

IEEE S&P‘20 에서는 총 5편의 관련 논문들이 발표되었다.

Spectator [2] 은 Spectre 공격에 대한 프로그램 안전성을 나타내는 개념인 SNI (Speculative Non-Interference)을 정의하고 심볼릭 실행 (Symbolic execution)을 기반으로 대상 프로그램의 SNI를 증명할 수 있는 알고리즘을 제시하였다.

NetCat [1] 은 네트워크 카드 등 주변장치의 I/O 데이터를 CPU 의 LLC 에 직접 전송하는 기술인 DDIO (Data-Direct I/O)를 이용하여 네트워크에서 원격에 위치한 대상 시스템에 캐시 부채널 공격을 수행하는 방법을 제시하였다.

SPECCFI [19] 는 CFI (Control flow integrity) 기술을 활용하여 두 Spectre 공격의 변종인 Spectre-BTB 와 Spectre-RSB 공격에 대한 대응 기법을 제시하였다. 이 방법은 분기 예측 시 발생하는 모든 실행 구간에 대해 사전에 CFG 에 의해 정의된 제어 흐름이 유지되도록 하여 공격자가 의도한 일시적 실행이 발생하지 못하도록 한다.

LVI [18] 는 기존의 Meltdown 유형의 공격들과는 다르게 역으로 victim 의 주소 공간에서 Meltdown 공격을 실행하는 새로운 방법인 LVI (Load value injection) 공격 기법을 제시하였다. LVI 공격은 기존의 모든 Meltdown 대응 기법들을 우회할 수 있어, 보다 근본적인 보안 대책이 필요함을 강조한다.

Plundervolt [14] 는 Intel 코어 동적 전압 스케일링 기술을 이용하여 CPU 의 전압을 미세하게 조정해 SGX 엔클레이브 내에서 실행 중인 프로그램에 오류를 발생시키는 방법을 제시하였다.

### 3.3. USENIX Security

USENIX Security‘20 에서는 총 4편의 관련 논문들이 발표되었다.

RELOAD+REFESH [13] 는 Intel CPU의 캐시 교체 (Replacement) 정책을 이용하여 캐시 라인을 제거

〔표 1〕 CPU 마이크로아키텍처 보안 기술 관련 2020년 주요 보안학회 발표 논문

| 학 회 | 논문 제목 (약어 표시) | 공격 대상 플랫폼 | R | T | E | 주 요 내 용 |
|---|---|---|---|---|---|---|
| NDSS | PhantomCache〔17〕 | - | ● | | | 캐시 부채널 공격을 방지하기 위하여 LLC 의 메모리 주소-캐시 매핑을 랜덤화하는 구조를 제안 |
| | SPEECHMINER〔16〕 | - | | ● | | 마이크로아키텍처 내부의 일시적 실행 공격을 유발하는 근본 취약점을 분석하기 위한 방법 제시 |
| | ConTeXT〔15〕 | - | | ● | | 일시적 실행 공격을 근본적으로 차단할 수 있는 H/W와 S/W의 co-design 제안 |
| IEEE S&P | Spectator〔2〕 | - | | ● | | 심볼릭 실행을 이용하여 프로그램의 Spectre 취약점 검증 방법 제시 |
| | NetCat〔1〕 | Intel | ● | | | DDIO를 이용하여 원격에서 실행 가능한 캐시 부채널 공격 제시 |
| | SPECCFI〔19〕 | - | | ● | | 제어 흐름 무결성 (CFI) 기술을 이용한 Spectre 공격 방어 기법 제시 |
| | LVI〔18〕 | Intel (SGX) | | ● | ● | Victim 주소공간에서 실행하는 역 Meltdown 유형 (Reverse Meltdown) 공격 제시 |
| | Plundervolt〔14〕 | Intel (SGX) | | | ● | 코어의 전압을 미세하게 조정하여 SGX 엔클레이브 내에 오류를 주입하는 공격 제시 |
| USENIX Security | RELOAD+REFESH〔13〕 | Intel | ● | | | Intel CPU 캐시 교체 정책을 이용하여 쉽게 탐지되지 않는 캐시 부채널 공격 제시 |
| | Medusa〔12〕 | Intel | | ● | | Meltdown 취약점을 찾을 수 있는 퍼징 도구를 소개하고 이를 이용하여 새로운 MDS 공격 발견 |
| | Membuster〔11〕 | Intel (SGX) | | | ● | 메모리 버스를 직접 스누핑 하여 엔클레이브의 메모리 접근 패턴을 관찰하는 물리 공격 제시 |
| | HybCache〔6〕 | - | ● | | | 캐시를 프로세스 별로 격리할 수 있는 캐시 설계 구조를 제안 |
| ACM CCS | InSpectre〔8〕 | - | | ● | | 비순차 실행과 투기적 실행의 모델링을 통해 공격 대응 방안 및 잠재적 취약점 제시 |
| | TRUSTORE〔7〕 | - | | | ● | FPGA 에 신뢰 저장 공간을 구축하여 각종 부채널 공격에 안전한 SGX 구현 방법 제시 |
| | CITM〔10〕 | ARM (TrustZone) | | | ● | 격리 실행 환경 (IEE)을 대상으로 캐시 일관성 프로토콜을 이용한 새로운 캐시 공격 소개 |
| | BlindSide〔9〕 | - | | ● | | Spectre 취약점을 이용하여 시스템 크래시를 유발하지 않고 익스플로잇을 제작할 수 있는 방법 제시 |

R: 자원 경합 공격, T: 일시적 실행 공격, E: 신뢰 실행 환경에 대한 공격

(Eviction) 하지 않고도 캐시 부채널 공격을 수행하는 방법을 제시하였다. 기존 방법과 달리 빈번한 캐시 제거가 발생하지 않으므로 공격이 쉽게 탐지되지 않는다.

Medusa [12] 는 Meltdown 유형 공격을 유발하는 취약점을 찾을 수 있는 퍼징 도구인 Transynther를 소개하였다. 최신 Intel CPU를 대상으로 퍼징을 수행한 결과 쓰기 결합 연산 과정에서 라인 필 버퍼에 데이터 누수가 발생하는 새로운 MDS 공격 (Meltdown 공격의 일종)을 발견하였다.

Membuster [11]는 Intel SGX 에 대한 물리적 공격을 제시하였다. 캐시와 DRAM 사이의 메모리 버스를 직접 스누핑하여 SGX 엔클레이브의 메모리 접근 패턴을 정밀한 수준으로 실시간 관찰할 수 있다.

HybCache [6] 는 캐시 부채널 공격을 방지하기 위한

새로운 캐시 설계 구조를 제안하였다. 캐시 웨이(Cache ways) 단위로 캐시 자원을 분할하여 격리된 환경을 제공하고 캐시 공유를 차단한다. 그리고 TEE 나 일부 보안이 요구되는 경우에만 선택적으로 캐시를 격리함으로써 시스템 성능 저하를 막을 수 있다.

### 3.4. ACM CCS

마지막으로, ACM CCS'20 에서는 총 4편의 관련 논문들이 발표되었다.

InSpectre [8] 는 비순차 실행과 투기적 실행 등 실제 CPU의 동작 구조를 모델링하고 이를 바탕으로 일시적 실행에서의 취약점을 근본적으로 제거하기 위한 요소들을 제시하였다. 또한, 이 모델을 통해 3가지의 잠재적인 Spectre 유형 취약점들을 발견하였다.

TRUSTORE [7] 는 FPGA에 신뢰 저장 공간을 구축하여 각종 부채널 공격으로부터 안전한 SGX 엔클레이브 기술을 제시하였다. Intel CPU-FPGA 하이브리드 아키텍처에서 구현을 하고 동작 성능을 평가하여 기존 ORAM 방식에 비해 성능이 우수함을 입증하였다.

CITM [10] 은 TEE 보다 경량화된 형태인 격리 실행 환경 (IEE, Isolated Execution Environment)을 대상으로 캐시 일관성 프로토콜 (Coherence protocol)을 이용한 캐시 공격들을 제시하였다. ARM TrustZone을 기반으로 한 ARM IEE를 비롯하여 SANCTUARY, Ginseng 등 다양한 IEE 에 대해서도 공격 가능성을 보였다.

BlindSide [9] 는 시스템 크래시를 유발하는 메모리 손상 (Memory corruption) 취약점에 Spectre 공격 기술을 결합하여 익스플로잇을 제작할 수 있는 방법을 제시하였다. Spectre 로 일시적 실행 영역을 만들고 그 안에 크래시가 발생할 수 있는 명령어들을 실행함으로써 실제 시스템 크래시를 일으키지 않고도 여러 번의 시도를 통한 익스플로잇 코드 제작이 가능하다.

## Ⅳ. 결 론

본 논문에서는 CPU 마이크로아키텍처의 취약점 및 이를 활용한 공격 기술들을 소개하였다. 구체적으로, 자원 경합 공격, 일시적 실행 공격, 그리고 신뢰 실행 환경에 대한 공격 등 취약점의 유형과 공격대상에 따라 분류한 3가지 유형의 공격을 소개하였다. 또한, 최근 주요 보안 학술대회에서 발표된 관련 논문들을 중심으로 최신 연구 동향을 살펴보았다.

성능 중심의 설계와 복잡한 내부 구조로 인해 아직도 여전히 많은 마이크로아키텍처 취약점들이 존재할 것으로 보인다. 따라서 신규 취약점 분석기술에 대한 연구는 물론 보안이 내재된 새로운 마이크로아키텍처 설계에 대한 연구도 앞으로 꾸준히 지속될 것으로 전망한다.

## 참 고 문 헌

[1] M. Kurth, B. Gras, D. Andriesse, C. Giuffrida, H. Bos, and K. Razavi, "NetCAT : Practical Cache Attacks from the Network," *in Proceedings of 2020 IEEE Symposium on Security and Privacy*, 2020.

[2] M. Guarnieri, B. Köpf, J. F. Morales, J. Reineke, and A. Sánchez, "SPECTECTOR: Principled Detection of Speculative Information Flows," *in Proceedings of 2020 IEEE Symposium on Security and Privacy*, 2020.

[3] A. Cabrera Aldaya, B. Bob Brumley, S. ul Hassan, C. Pereida García, and N. Tuveri, "Port Contention for Fun and Profit," *in Proceedings of 2019 IEEE Symposium on Security and Privacy*, 2019.

[4] C. Canella et al., "A Systematic Evaluation of Transient Execution Attacks and Defenses," *in Proceedings of the 28th USENIX Security Symposium*, 2019.

[5] Y. Yarom and K. Falkner, "Flush + Reload : a High Resolution , Low Noise, L3 Cache Side-Channel Attack," *in Proceedings of the 23th USENIX Security Symposium*, 2014.

[6] G. Dessouky, T. Frassetto, and A.-R. Sadeghi, "HybCache: Hybrid Side-Channel-Resilient Cachesfor Trusted Execution Environments," *in Proceedings of the 29th USENIX Security Symposium*, 2020.

[7] H. Oh, A. Ahmad, B. Lee, and Y. Paek, "TrustOre: Side-Channel Resistant Storage for SGX using Intel Hybrid CPU-FPGA," *in*

*Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[8] R. Guanciale, "InSpectre: Breaking and Fixing Microarchitectural Vulnerabilities by Formal Analysis," *in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[9] E. Göktaş, K. Razavi, K. Ch, E. Zürich, G. Portokalidis, and H. Bos, "Speculative Probing: Hacking Blind in the Spectre Era," *in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020

[10] J. Wang, K. Sun, L. Lei, S. Wan, Y. Wang, and J. Jing, "Cache-in-the-Middle (CITM) Attacks : Manipulating Sensitive Data in Isolated Execution Environments," *in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[11] D. Lee, D. Jung, I. T. Fang, C.-C. Tsai, and R. A. Popa, "An Off-Chip Attack on Hardware Enclaves via the Memory Bus," *in Proceedings of the 29th USENIX Security Symposium*, 2020.

[12] D. Moghimi, M. Lipp, B. Sunar, and M. Schwarz, "Medusa: Microarchitectural Data Leakage via Automated Attack Synthesis," *in Proceedings of the 29th USENIX Security Symposium*, 2020.

[13] S. Briongos, P. Malagón, J. M. Moya, and T. Eisenbarth, "RELOAD+REFRESH: Abusing Cache Replacement Policies to Perform Stealthy Cache Attacks," *in Proceedings of the 29th USENIX Security Symposium*, 2020.

[14] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens, "Plundervolt : Software-based Fault Injection Attacks against Intel SGX," *in Proceedings of 2020 IEEE Symposium on Security and Privacy*, 2020.

[15] M. Schwarz, M. Lipp, C. Canella, R. Schilling, F. Kargl, and D. Gruss, "ConTExT: A Generic Approach for Mitigating Spectre," *in Proceedings 2020 Network and Distributed System Security Symposium*, 2020.

[16] Y. Xiao, Y. Zhang, and R. Teodorescu, "SPEECHMINER: A Framework for Investigating and Measuring Speculative Execution Vulnerabilities," *in Proceedings 2020 Network and Distributed System Security Symposium*, 2020.

[17] Q. Tan and K. Bu, "PhantomCache : Obfuscating Cache Conflicts with Localized Randomization," *in Proceedings 2020 Network and Distributed System Security Symposium*, 2020.

[18] J. Van Bulck et al., "LVI : Hijacking Transient Execution through Microarchitectural Load Value Injection," *in Proceedings of 2020 IEEE Symposium on Security and Privacy*, 2020.

[19] E. M. Koruyeh, S. Haji, A. Shirazi, K. N. Khasawneh, C. Song, and N. Abu-Ghazaleh, "SPECCFI: Mitigating Spectre Attacks using CFI Informed Speculation," *in Proceedings of 2020 IEEE Symposium on Security and Privacy*, 2020.

## 〈저자소개〉

**신 영 주 (Youngjoo Shin)**

정회원

2006년 2월 : 고려대학교 컴퓨터학과 학사

2008년 2월 : KAIST 전산학과 석사

2014년 8월 : KAIST 전산학과 박사

2008년 4월~2017년 2월 : 국가보안기술연구소 선임연구원

2017년 3월~2020년 8월 : 광운대학교 컴퓨터정보공학부 조교수

2020년 9월~현재 : 고려대학교 정보보호대학원 조교수

<관심분야> 시스템 보안, CPU 마이크로아키텍처 취약점 분석, 클라우드 보안