A Comparison Study of Intel SGX and AMD Memory Encryption Technology

Saeid Mofrad, Fengwei Zhang

Shiyong Lu

Wayne State University

{saeid.mofrad, Fengwei, Shiyong}@wayne.edu

Weidong Shi (Larry)

University of Houston

wshi3@uh.edu

Outline

- Introduction
- Technology Background
- Comparison and Results
- Conclusions and Future Work

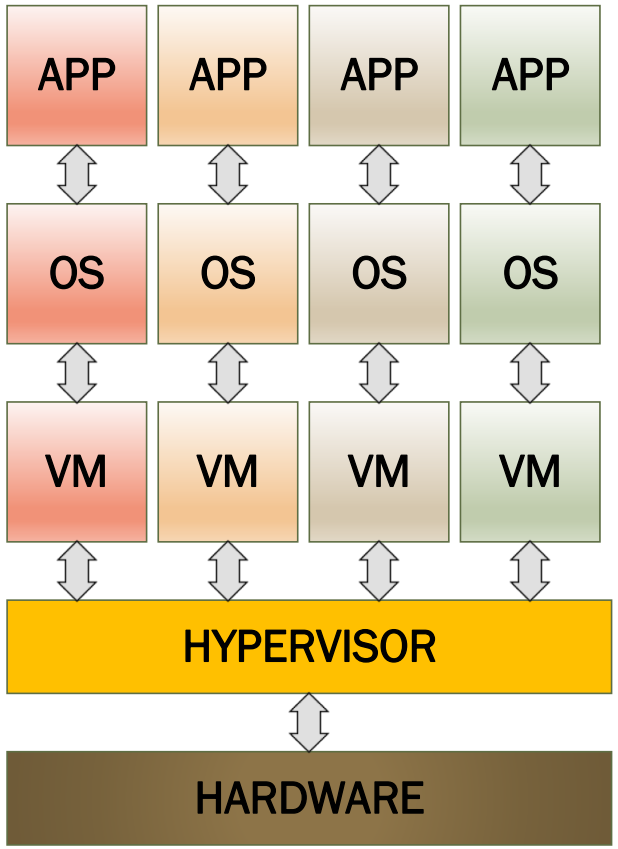
Outline

- Introduction
- Technology Background
- Comparison and Results
- Conclusions and Future Work

Trusted Execution Environment

- Trusted Execution Environment can be achieved with isolation.
- Isolation through Virtual Machine is a common approach to achieve security at runtime.
- Downsides of software only virtualization:

1) Virtualization uses OS and Hypervisor and puts them in the TCB.
2) OS or Hypervisor contains thousands of lines of code and may have security flaws.
3) Many OS and Hypervisor exploits have been reported.
4) Increased TCB means less security.



Hardware-Assisted Trusted Execution Environment

- Hardware-Assisted TEE couples hardware with TEE technology mitigates the downsides of the software only TEEs.
- Hardware-Assisted TEE is faster since it uses dedicated hardware.
- Hardware-Assisted TEE exposes small TCB and smaller TCB means better security.
- Early Hardware-Assisted TEE: Intel ME, AMD PSP, and x86 SMM.

- Two general purpose Hardware-Assisted TEE have been proposed recently in x86 architecture:

1. Intel Software Guard eXtensions (SGX). (HASP 2013)
2. AMD Memory Encryption Technology. (White Paper 2016)

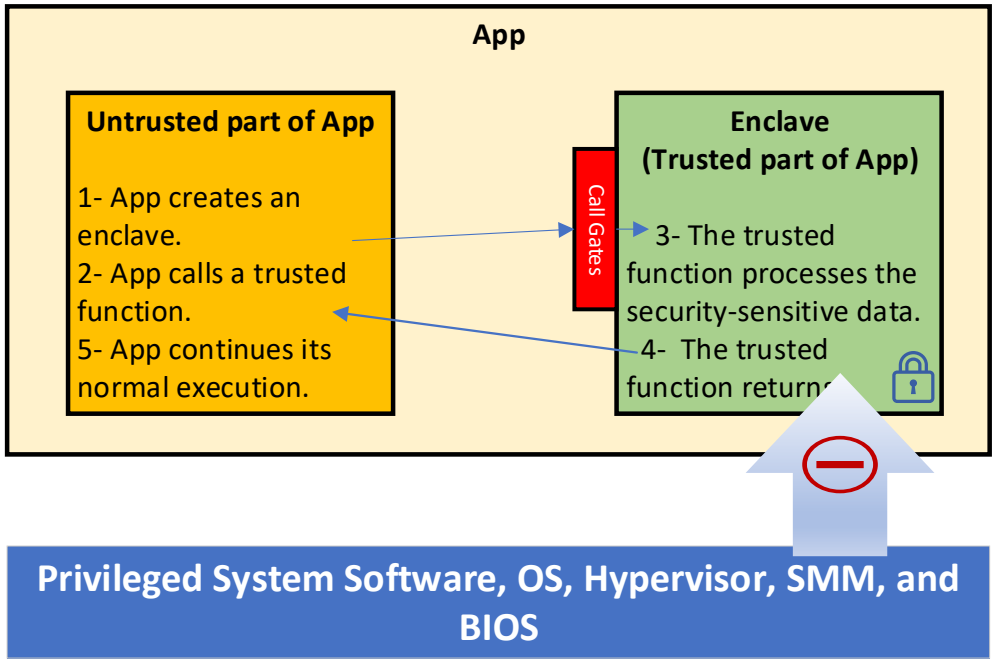
Outline

- Introduction
- Technology Background
- Comparison and Results
- Conclusions and Future Work

Background: Intel Software Guard eXtensions (SGX)

➢ **SGX Application execution flow:**

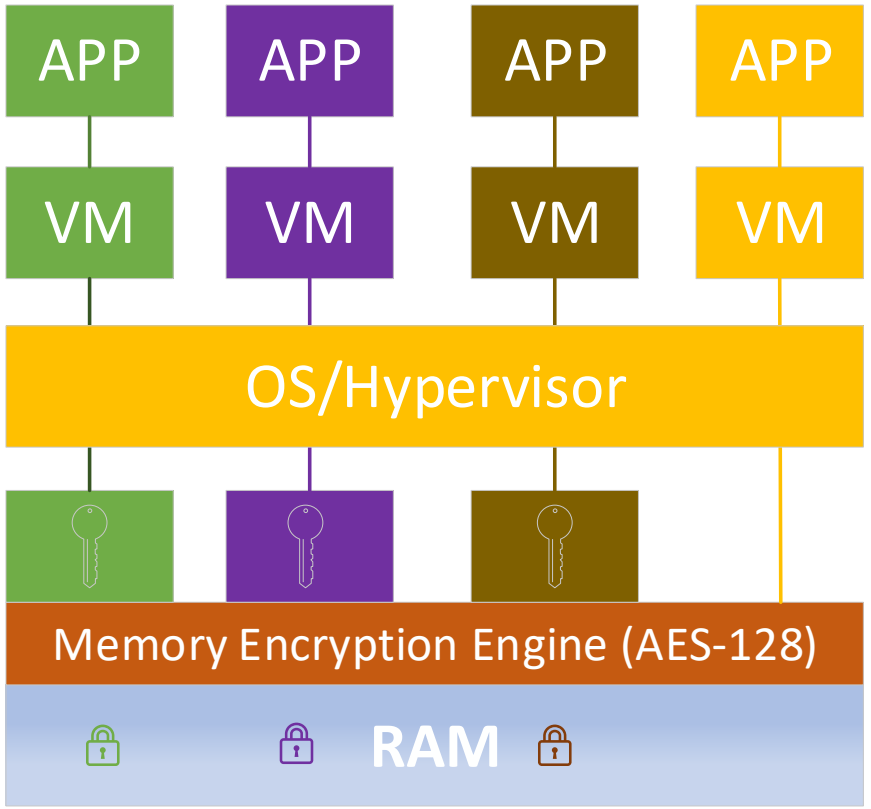
1) App is built with trusted and untrusted part.
2) Untrusted part creates and executes the enclave that is placed in the encrypted and trusted memory referred to as EPC.
3) Trusted function is called and the execution is transferred into the enclave where all data will be in clear text, then the security-sensitive data is processed.
4) Trusted function returns.
5) App continues its normal execution.



Background: AMD Memory Encryption Technology

- Addresses the physical access and the system software class of attacks in the public cloud.
- **Introduces SME, TSME, and SEV**

1) Secure Memory Encryption (SME) and Transparent Secure Memory Encryption (TSME) protect against the physical access attacks.
2) Secure Encrypted Virtualization (SEV) protects against system software class of attacks.
3) SME and TSME encrypt the system memory providing the confidentiality for illegally physical access attempts.
4) SEV encrypts VM's memory image with an unique key to the VM providing confidentiality for VM's memory image and protects against system software class of attacks.



Outline

- Introduction
- Technology Background
- Comparison and Results
- Conclusions and Future Work

Testbeds Configuration

Testbed Machine	Intel	AMD
CPU Model	Core i7-6700	EPYC-7251
CPU Physical Core Number	4	8
CPU Logical Core Number	8	16
CPU Base Clock	3.4 GHz	2.1 GHz
CPU Max Clock	4.0 GHz	2.9 GHz
Cache Type	Smart Cache	L3
Cache Size	8MB	32MB
Motherboard	DELL OptiPlex 7040	GIGABYTE MZ31-AR0
Memory	8GB DDR4 No-ECC	32GB DDR4-ECC
Storage	1TB 7200 RPM HDD	512GB SSD
Operating System	Linux 16.04 LTS	Linux 16.04 LTS
OS, Hypervisor kernel	4.15.7-041507-generic	4.15.0-rc1-kvm
VM Kernel	N/A	4.14.0-rc5-tip
TEE SDK Version	SGX SDK Ver 2.00	N/A

SGX VS SEV

TEE Technology	Highest Access Level	Memory Size Limits	SDK	Software Change	Platform Attestation Mechanism	Protection Level
Intel SGX	Ring 3	Up to 128MB EPC	Provided	Required	Attested through Intel Remote Attestation Protocol and IAS	Confidentiality and Integrity of the Code and Data in the Enclave At Runtime
AMD SEV	Ring 0	Up to Available System Ram	Not Required	Only Hypervisor and VM's Kernel	Attested through AMD Secure Processor	Confidentiality of the VM's Memory Image At Runtime

Function and Use Cases Comparison

Intel SGX	AMD Memory Encryption Technology (SEV)
Initial design targeted microservices and small workload. (small amount of secure memory and was featured mainly in mobile and desktop family processors)	Initial design targeted cloud and Infrastructure as a Service. (Large amount of secure memory featured in server family processors)
Requires major software changes and code refactoring. (Not suitable for securing legacy applications)	Does not require software changes and code refactoring. (Suitable for securing legacy applications)
SGX works with ring 3 and is not suitable for workloads with many system calls.	SEV works with ring 0 and is suitable for broader range of workloads especially those with many system calls.
SGX is suitable for small but security-sensitive workload. (SGX has small TCB)	SEV is suitable for securing legacy, large and enterprise level application. (SEV has large TCB)

Security and Vulnerability Comparison

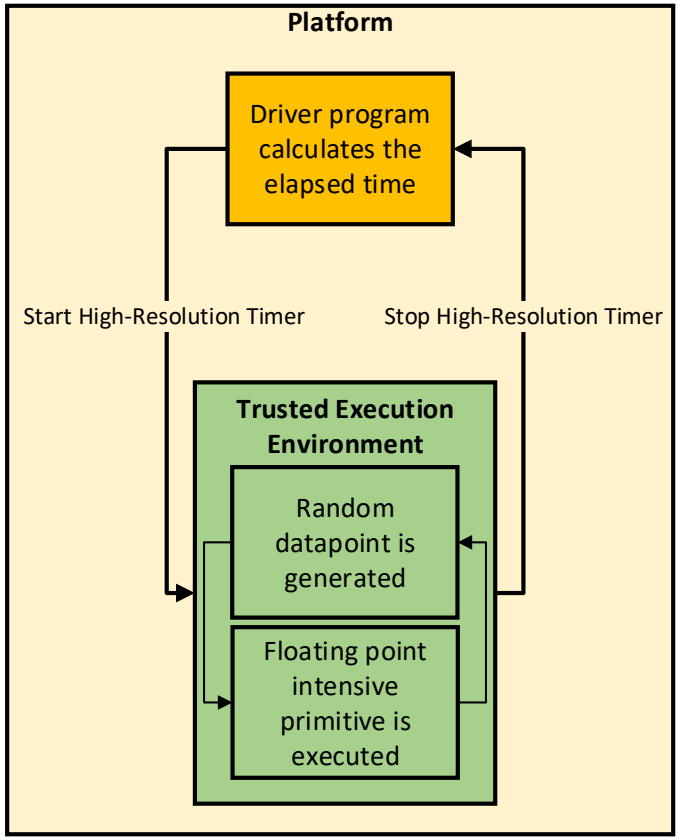
Intel SGX	AMD Memory Encryption Technology (SME, SEV)
Provides Memory Integrity Protection.	Does Not Provide Memory Integrity Protection.
Vulnerable to Memory Side Channels.	Vulnerable to Memory Side Channels.
Vulnerable to Denial of Service Attacks. (OS Handles System Calls)	Vulnerable to Denial of Service Attacks. (Hypervisor Handles VM Requests)
Small TCB. (TCB is CPU package)	Large TCB. (VM's OS is located inside TCB)
Vulnerable to Synchronization Attacks. (TOCTTOU, Use-After-Free)	AMD Secure Processor Firmware Bug Discovered. (MASTERKEY and FALLOUT)

Intel SGX carefully separates the trusted and untrusted environments, provides a narrow and protected enclave gateway, enforces memory access control, and applies memory integrity protection, thus making it a suitable TEE for protecting workloads that interact with security-sensitive data.

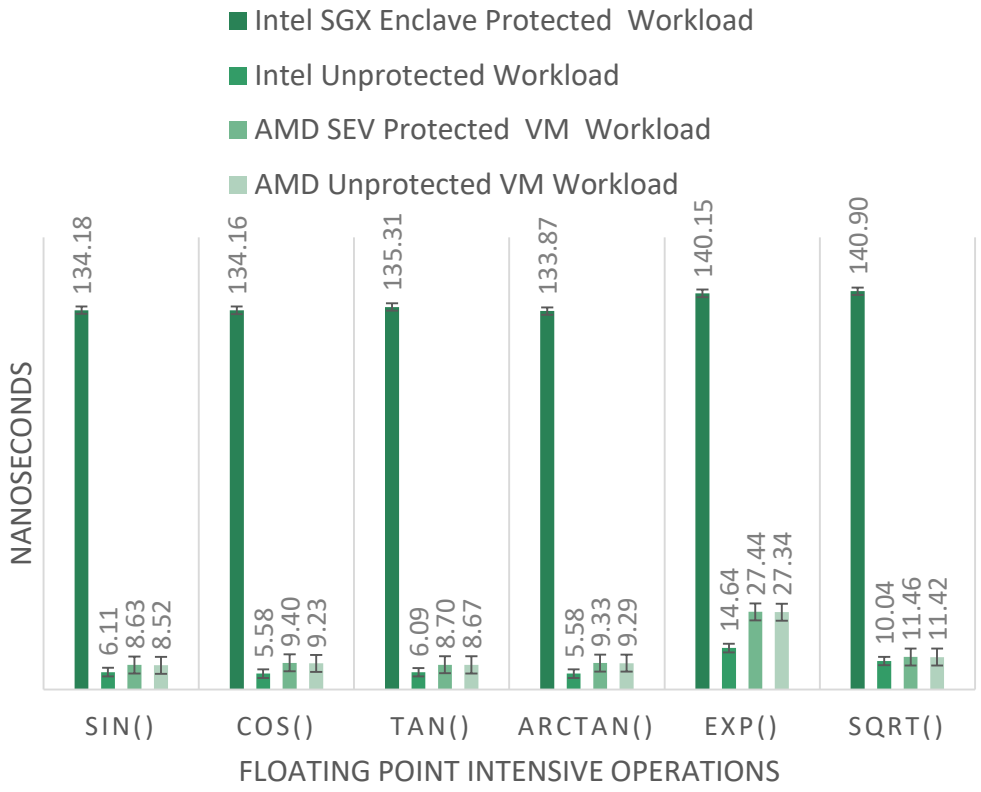
Floating Point Intensive Workload Comparison

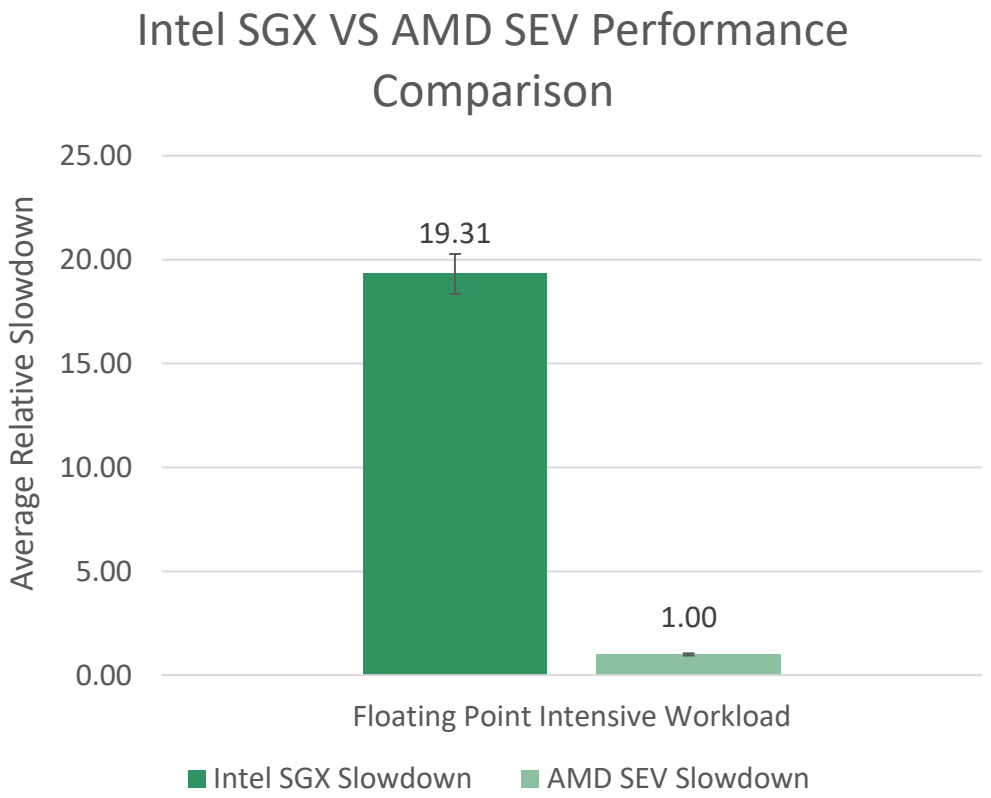
- Measures the execution performance of the TEE.
- Methodology:

1) Codebase is identical for SGX and SEV.
2) SGX uses different random number generator (Provided by SGX SDK).
3) Datapoints are generated inside the TEE.
4) Benchmark applies floating point intensive primitives and calculates the elapsed time.



Floating Point Intensive Workload Comparison Results



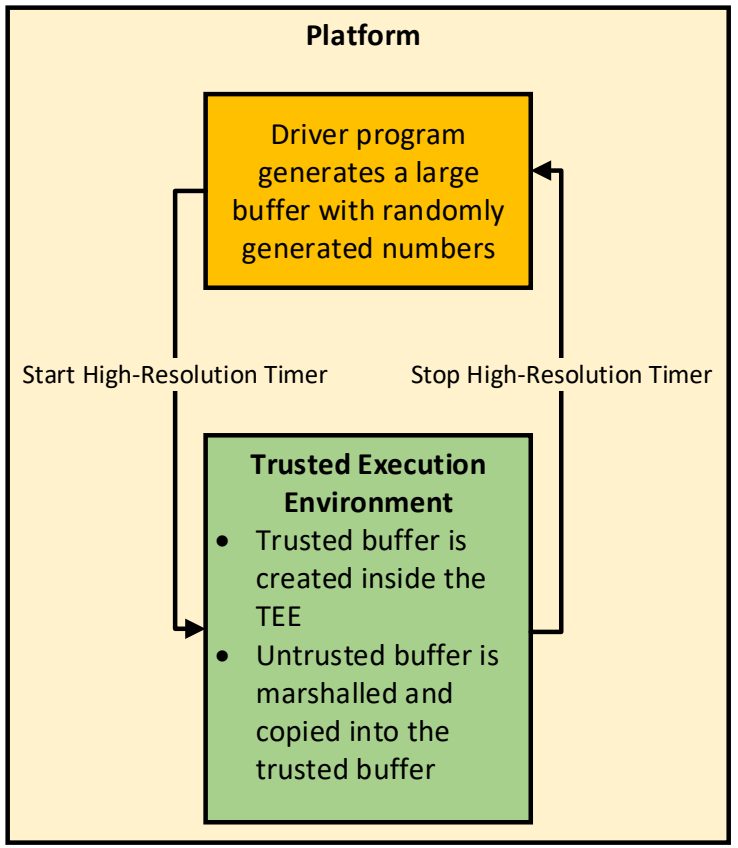


Memory Encryption Engine Performance Comparison

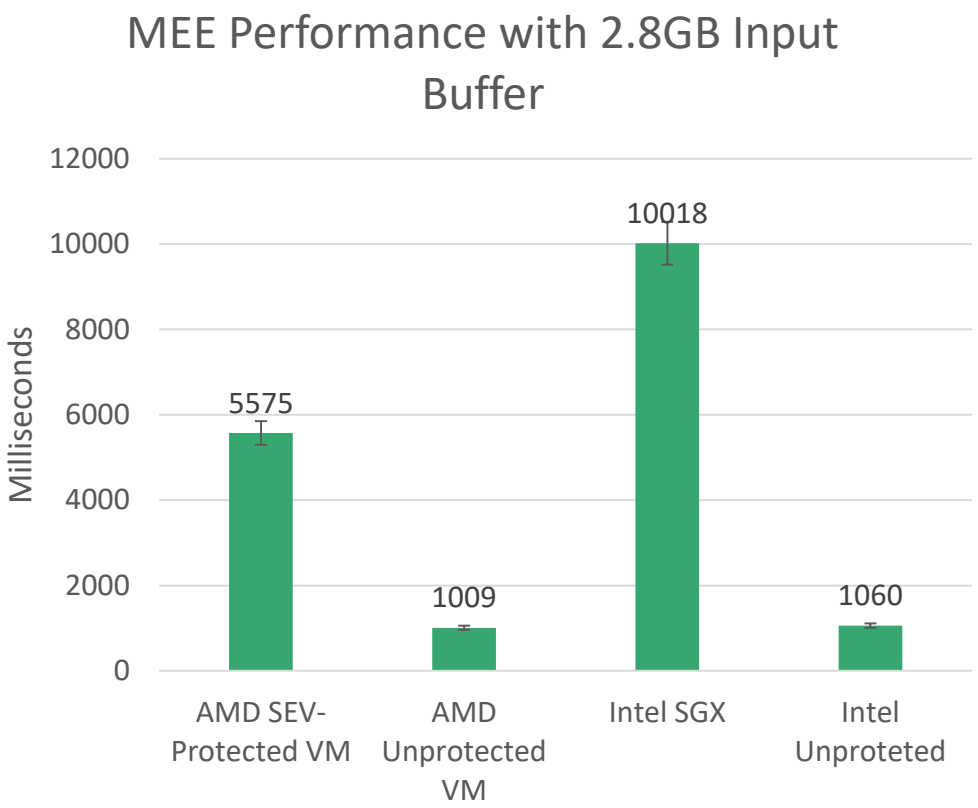
- Measures the MEE performance of the TEE.
- Codebase is identical for SGX and SEV.

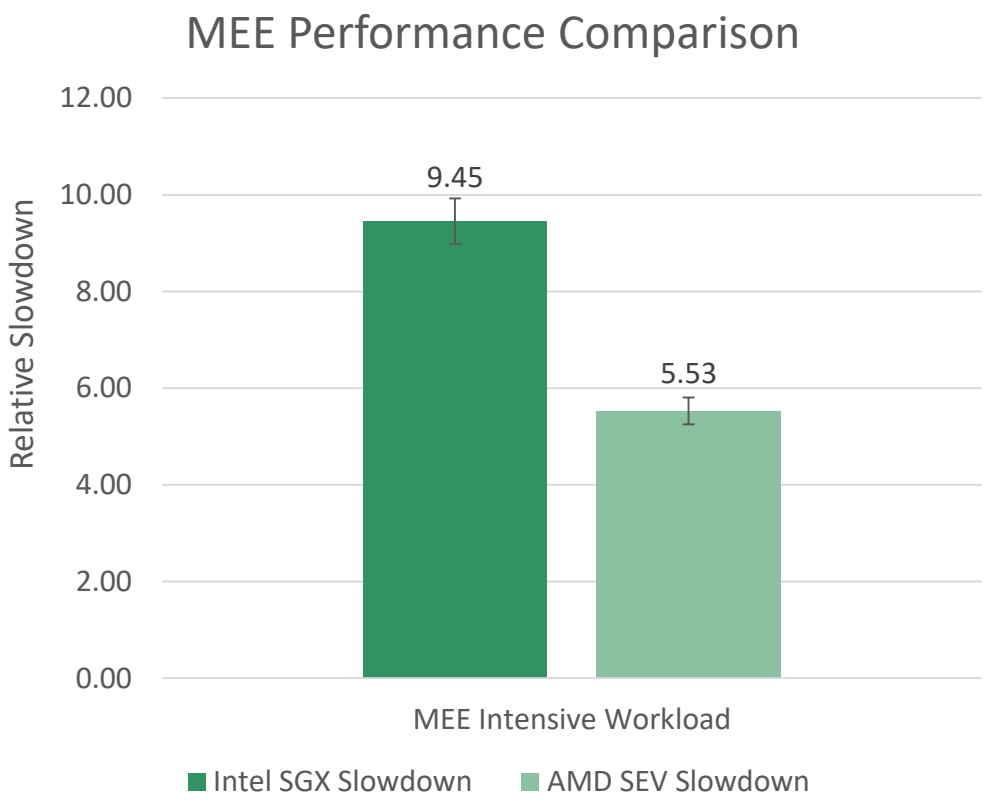
Methodology:

1) Large buffer is generated outside of the TEE.
2) Large buffer is sent and copied inside the TEE.
3) Elapsed time is calculated.



Memory Encryption Engine Performance Comparison Results



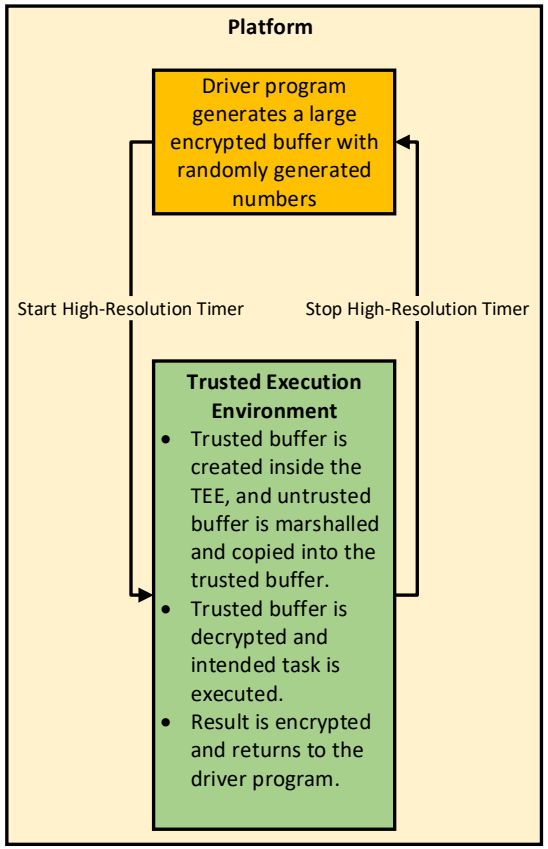


Comprehensive Workload Comparison

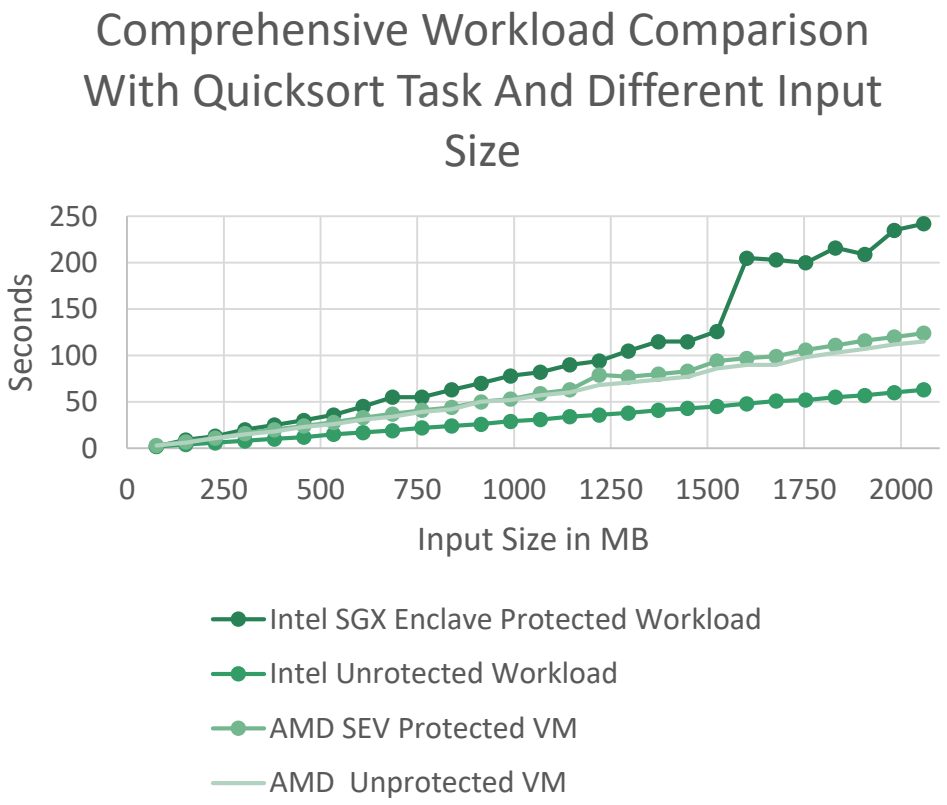
- Measures the performance of the TEE while a secure protocol for public cloud data provisioning and workload is followed.
- Codebase is identical for SGX and SEV.
- Task performed: Quicksort and MD5 message digest.
- SEV simulates enclave model.

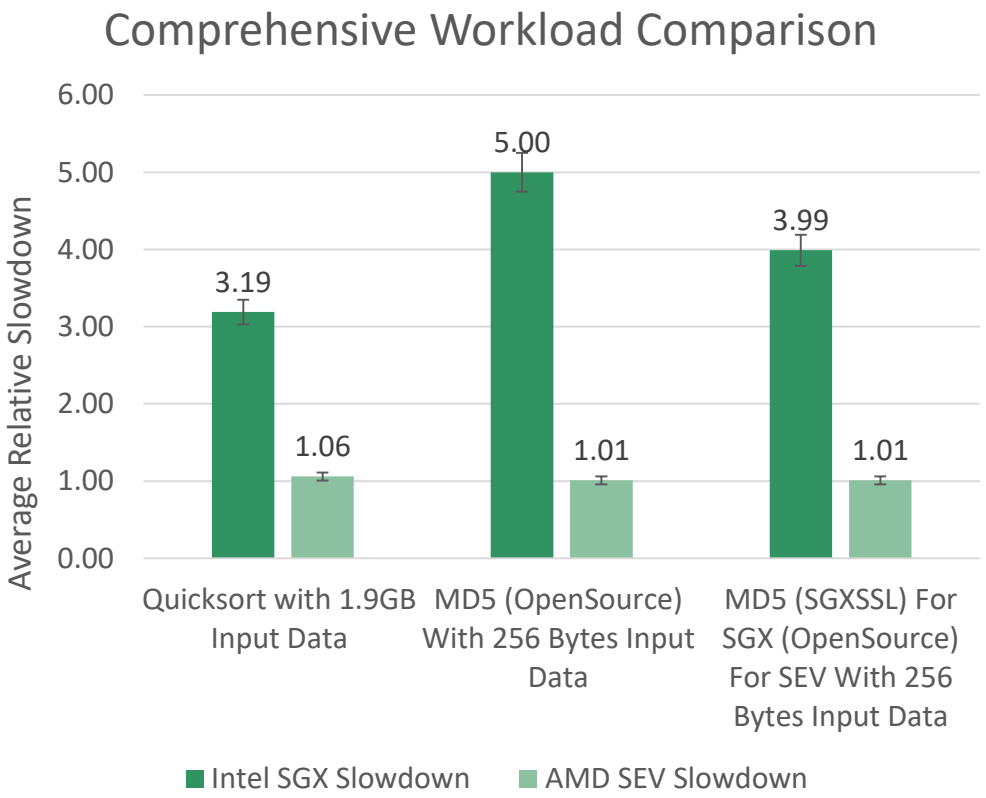
Methodology:

1) Driver program generates and encrypts datapoints with a key known to the enclave.
2) Encrypted data is sent and copied inside the TEE.
3) Inside the enclave the received buffer is decrypted, task is executed, result is encrypted, and returns to the driver program.
4) Elapsed time is calculated.



Comprehensive Workload Comparison Results





Outline

- Introduction
- Technology Background
- Comparison and Results
- Conclusions and Future Work

Conclusions and Future Work

- This paper is the first comparison study between AMD Memory Encryption Technology and Intel Software Guard eXtensions (SGX).
- This paper illustrates comparison information regarding the functionality and use cases, security, and performance of Intel SGX and AMD Memory Encryption Technology.
- We conclude that Intel SGX is suited for highly security-sensitive but small workloads since it enforces the memory integrity protection and has a limited amount of secure resources.
- AMD SME and SEV do not provide memory integrity protection. However, providing a greater amount of secure resources to applications, performing faster than Intel SGX (when an application requires a large amount of secure memory), and no code refactoring, make them more suitable for complex or legacy applications and services.
- Future work: SEV-ES and SGX2.

Thank You!

Email: saeid.mofrad@wayne.edu

Technical Report of this work will be available at:

http://compass.cs.wayne.edu/compass/publications.html