ACM CCS'17 Tutorial

SGX Security and Privacy

Taesoo Kim (Georgia Tech), Zhiqiang Lin (UT Dallas)
Chia-Che Tsai (UC Berkeley/Stony Brook)

11/02/2017
Dallas, Texas

Agenda

- Part 1: SGX Introduction and Applications (by Zhiqiang)
- Part 2: SGX Shielding Frameworks and Development Tools (by Chia-che)
- Part 3: Security Issues on SGX (by Taesoo)

SGX Introduction and Applications

Zhiqiang Lin

UT Dallas

Outline

Outline

The Evolutions of Using Isolation for Malware Defense

The Evolutions of Using Isolation for Malware Defense

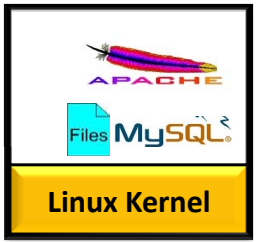
The Evolutions of Using Isolation for Malware Defense

The Evolutions of Using Isolation for Malware Defense

The Evolutions of Using Isolation for Malware Defense

The Evolutions of Using Isolation for Malware Defense

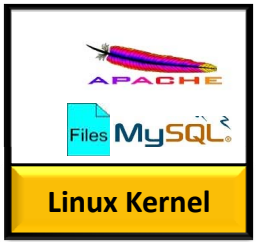
The Evolutions of Using Isolation for Malware Defense



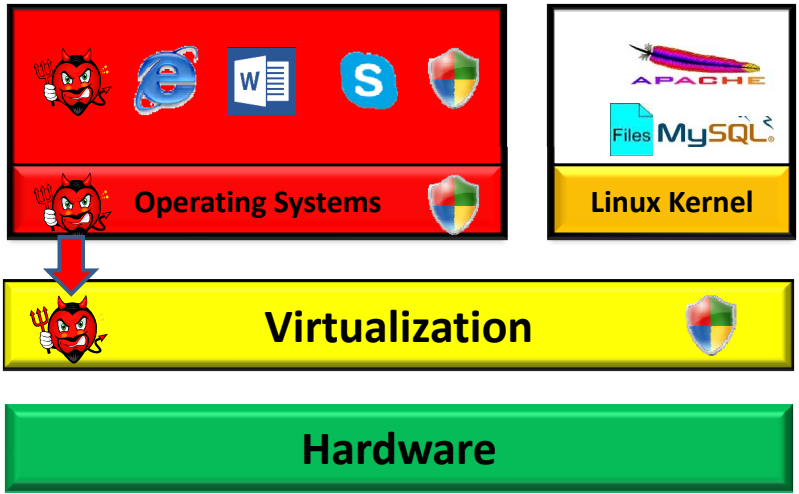
Virtualization

Hardware

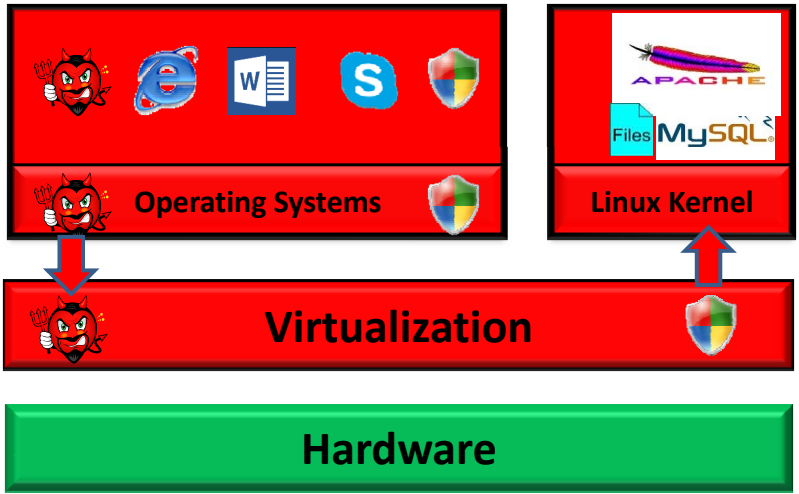
The Evolutions of Using Isolation for Malware Defense



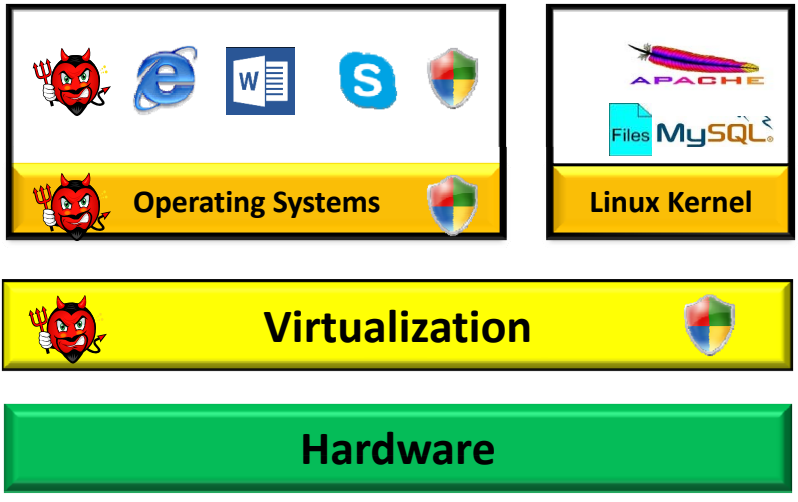
The Evolutions of Using Isolation for Malware Defense



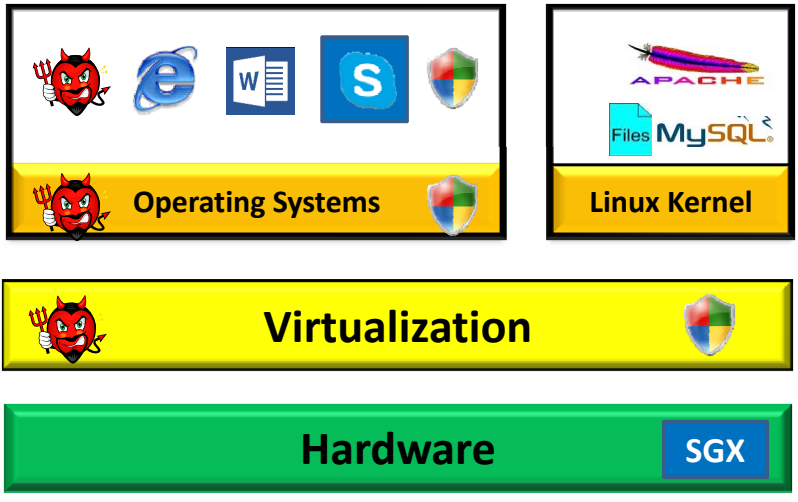
The Evolutions of Using Isolation for Malware Defense



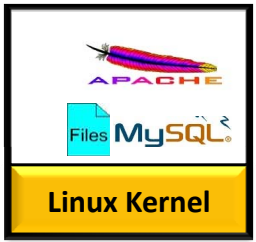
The Evolutions of Using Isolation for Malware Defense



The Evolutions of Using Isolation for Malware Defense



The Evolutions of Using Isolation for Malware Defense



Hardware Assured Security

1. Secure coprocessors [Yee94]
   - IBM 4758 [SW99]
2. Aegis secure processor [SCG+03]
3. Trusted Platform Module (TPM) [TPM03]
4. Trust Zone [Alv04]
5. AMD SVM [VD06]
6. Intel Trusted Execution Technology (TXT) [FG13]

Hardware Assured Security

1. Secure coprocessors [Yee94]
   - IBM 4758 [SW99]
2. Aegis secure processor [SCG+03]
3. Trusted Platform Module (TPM) [TPM03]
4. Trust Zone [Alv04]
5. AMD SVM [VD06]
6. Intel Trusted Execution Technology (TXT) [FG13]

Intel Software Guard eXtension (SGX) [MAB+13]

Providing **Hardware Assured Security** w/ **Minimized Attack Surface**

Existing Computer Systems

- Apps must trust
  - OS/VMM
  - BIOS, SMM
- Trust relies on software

Intel Software Guard eXtension (SGX) [MAB+13]

Providing **Hardware Assured Security** w/ **Minimized Attack Surface**

Existing Computer Systems

- Apps must trust
  - OS/VMM
  - BIOS, SMM
- Trust relies on software

Computer Systems w/ SGX

- Apps must trust
  - SGX hardware
- Trust excludes OS/VMM/BIOS/SMM

Intel Software Guard eXtension (SGX) [MAB+13]

Providing **Hardware Assured Security** w/ **Minimized Attack Surface**

Existing Computer Systems

- Apps must trust
  - OS/VMM
  - BIOS, SMM
- Trust relies on software

Computer Systems w/ SGX

- Apps must trust
  - SGX hardware
- Trust excludes OS/VMM/BIOS/SMM

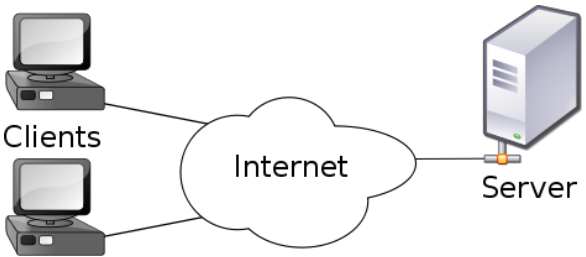
With SGX, for the first time, apps gain the ability to manage its own secret, without relying on the underlying systems software

Killer Apps

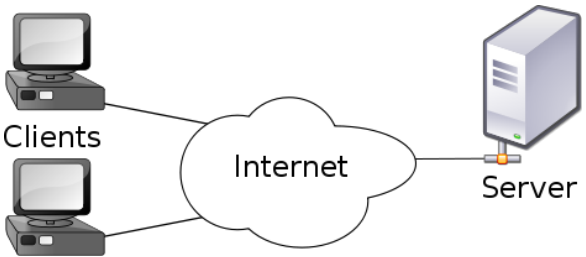
SGX provides a perfect platform for **Secure Remote Execution** [CD16]

Killer Apps

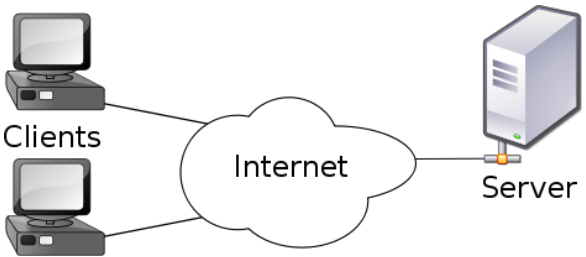
SGX provides a perfect platform for **Secure Remote Execution** [CD16]



Killer Apps



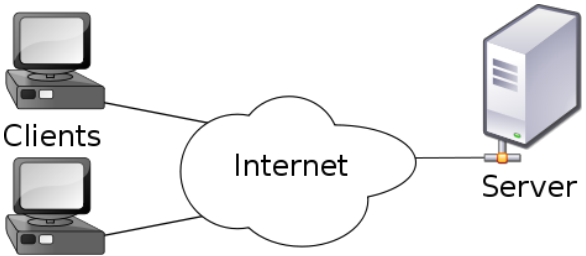
Killer Apps



Client: End Users

- **Cloud computing**
  - (Secret-preserving) data analytics
  - Healthcare record processing

Killer Apps



Client: End Users

- **Cloud computing**
  - (Secret-preserving) data analytics
  - Healthcare record processing

Server: Service Providers

- Computer game publishers
- Media streaming providers
- Software vendors (e.g., DRM)

Outline

Outline

SGX Instructions

SGX Version	User Space `enclu`	Kernel Space `encls`	Total
SGX-v1	5	13	18
SGX-v2	8	16	24

SGX Instructions

SGX Version	User Space `enclu`	Kernel Space `encls`	Total
SGX-v1	5	13	18
SGX-v2	8	16	24

`sysenter/sysexit` Instructions

SGX Instructions Overview

Privilege	Type	Instruction	Description	Version
P	MEM	EADD	Add a page	v1
P	MEM	EBLOCK	Block an EPC page	v1
P	EXE	ECREATE	Create an enclave	v1
P	DBG	EDBGRD	Read data by debugger	v1
P	DBG	EDBGWR	Write data by debugger	v1
P	MEM	EEXTEND	Extend EPC page measurement	v1
P	EXE	EINIT	Initialize an enclave	v1
P	MEM	ELDB	Load an EPC page as blocked	v1
P	MEM	ELDU	Load an EPC page as unblocked	v1
P	SEC	EPA	Add version array	v1
P	MEM	EREMOVE	Remove a page from EPC	v1
P	MEM	ETRACK	Activate EBLOCK checks	v1
P	MEM	EWB	Write back/invalidate an EPC page	v1
P	MEM	EAUG	Allocate page to an existing enclave	v2
P	SEC	EMODPR	Restrict page permissions	v2
P	EXE	EMODT	Change the type of an EPC page	v2
U	EXE	EENTER	Enter an enclave	v1
U	EXE	EEXIT	Exit an enclave	v1
U	SEC	EGETKEY	Create a cryptographic key	v1
U	SEC	EREPORT	Create a cryptographic report	v1
U	EXE	ERESUME	Re-enter an enclave	v1
U	MEM	EACCEPT	Accept changes to a page	v2
U	SEC	EMODPE	Enhance access rights	v2
U	MEM	EACCEPTCOPY	Copy page to a new location	v2

SGX Instructions Overview

Privilege	Type	Instruction	Description	Version
P	MEM	EADD	Add a page	v1
P	MEM	EBLOCK	Block an EPC page	v1
P	EXE	ECREATE	Create an enclave	v1
P	DBG	EDBGRD	Read data by debugger	v1
P	DBG	EDBGWR	Write data by debugger	v1
P	MEM	EEXTEND	Extend EPC page measurement	v1
P	EXE	EINIT	Initialize an enclave	v1
P	MEM	ELDB	Load an EPC page as blocked	v1
P	MEM	ELDU	Load an EPC page as unblocked	v1
P	SEC	EPA	Add version array	v1
P	MEM	EREMOVE	Remove a page from EPC	v1
P	MEM	ETRACK	Activate EBLOCK checks	v1
P	MEM	EWB	Write back/invalidate an EPC page	v1
P	MEM	EAUG	Allocate page to an existing enclave	v2
P	SEC	EMODPR	Restrict page permissions	v2
P	EXE	EMODT	Change the type of an EPC page	v2
U	EXE	EENTER	Enter an enclave	v1
U	EXE	EEXIT	Exit an enclave	v1
U	SEC	EGETKEY	Create a cryptographic key	v1
U	SEC	EREPORT	Create a cryptographic report	v1
U	EXE	ERESUME	Re-enter an enclave	v1
U	MEM	EACCEPT	Accept changes to a page	v2
U	SEC	EMODPE	Enhance access rights	v2
U	MEM	EACCEPTCOPY	Copy page to a new location	v2

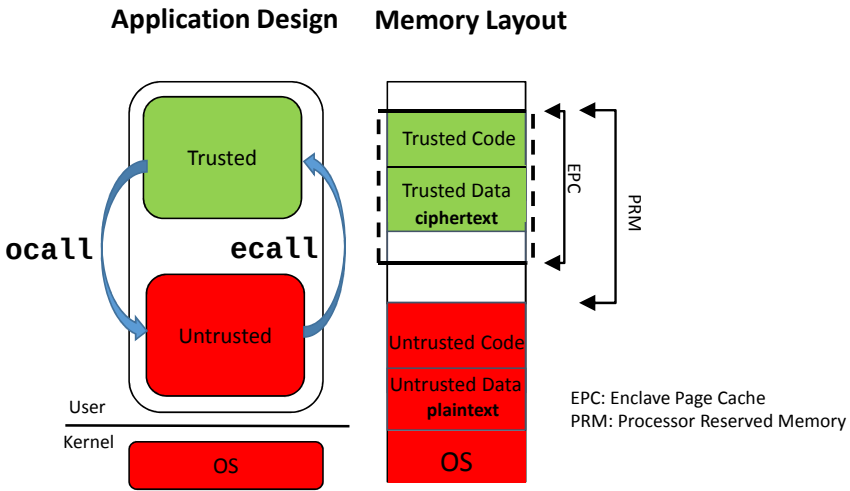
```
enclu[EEXIT]:

    mov $0x4,%rax
    enclu
```

SGX Data Structures

1. SGX Enclave Control Structure (SECS)
2. Thread Control Structure (TCS)
3. State Save Area (SSA)
4. Page Information (PAGEINFO)
5. Security Information (SECINFO)
6. Paging Crypto MetaData (PCMD)
7. Enclave Signature Structure (SIGSTRUCT)
8. EINT Token Structure (EINITTOKEN)
9. Report (REPORT)
10. Report Target Info (TARGETINFO)
11. Key Request (KEYREQUEST)
12. Version Array (VA)
13. Enclave Page Cache Map (EPCM)

SGX Application Design, and Memory Layout



EPC: Enclave Page Cache
PRM: Processor Reserved Memory

Outline

Two Ways of Having Software Protected by SGX

Approach-I: Developing SGX software from scratch through partitioning

- Partitioning software into **trusted** and **untrusted component**
- Using software development (e.g., SGX SDK [sgx]) tools to create an enclave module (shared object), which contains implementation for the trusted component
  - Defining enclave (executing the trusted component) interface
  - Using tools to generate stubs/proxies for `ecalls` and `ocalls`
  - Linking w/ SGX libraries
- Build and debug

Two Ways of Having Software Protected by SGX

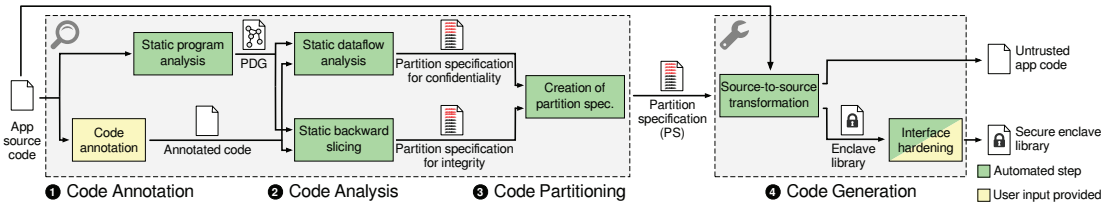
Approach-I: Developing SGX software from scratch through partitioning

- Partitioning software into **trusted** and **untrusted component**
- Using software development (e.g., SGX SDK [sgx]) tools to create an enclave module (shared object), which contains implementation for the trusted component
  - Defining enclave (executing the trusted component) interface
  - Using tools to generate stubs/proxies for `ecalls` and `ocalls`
  - Linking w/ SGX libraries
- Build and debug

Developing SGX software using a partitioning approach is what Intel intended

Teditious for the partitioning

Glamdring: Automatic App Partitioning for Intel SGX [LPM+17]



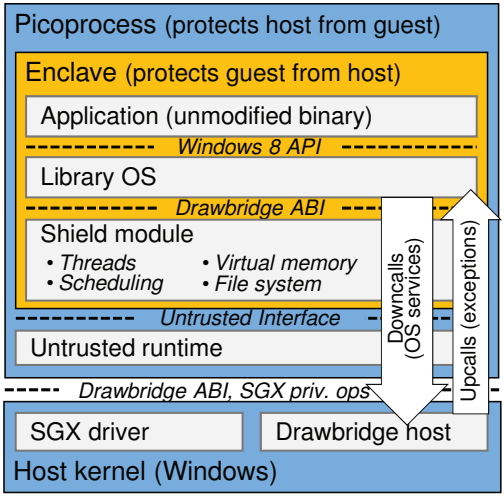
Source: https://www.usenix.org/system/files/conference/atc17/atc17-lind.pdf

Two Ways of Having Software Protected by SGX

Approach-II: Enabling Legacy Software with SGX Protection w/o Parititioning

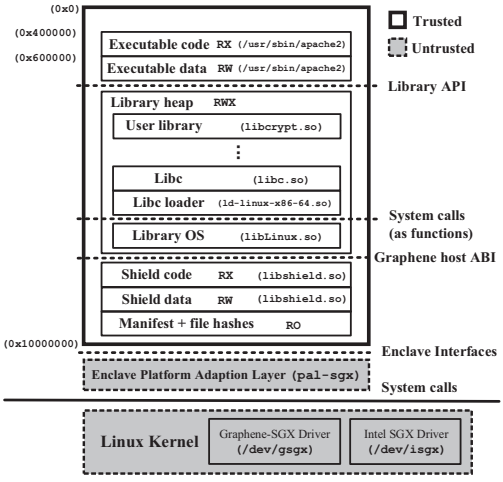
- SGX apps run at ring-3; system level code cannot be executed inside enclave.
- If SGX apps are executed with libOS (which is ring-3), then the problem gets solved
- No partitioning: the library OS approach
  1. Haven [BPH14, BPH15]
  2. Graphene-SGX [TPV17]

Haven [BPH14, BPH15]



Source: https://www.usenix.org/system/files/conference/osdi14/osdi14-paper-baumann.pdf

Graphene-SGX [TPV17]



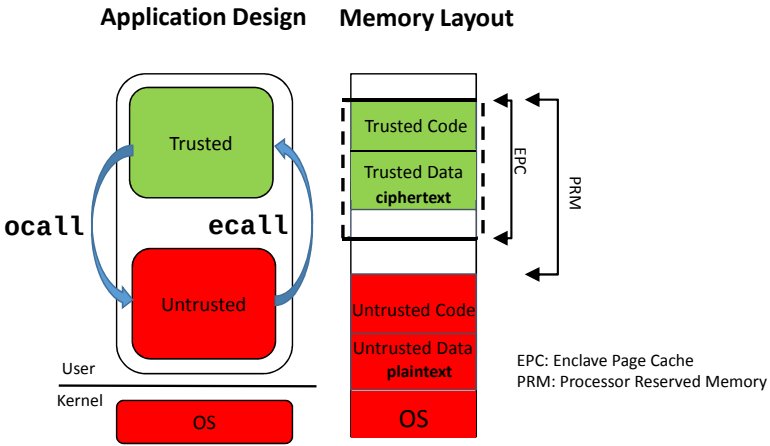
Source: https://www.usenix.org/system/files/conference/atc17/atc17-tsai.pdf

Two Ways of Having Software Protected by SGX

Approach	TCB	Effort	Library
Partitioning	Small	High	Intel SDK
No Partitoning	Large	Low	e.g., Graphene

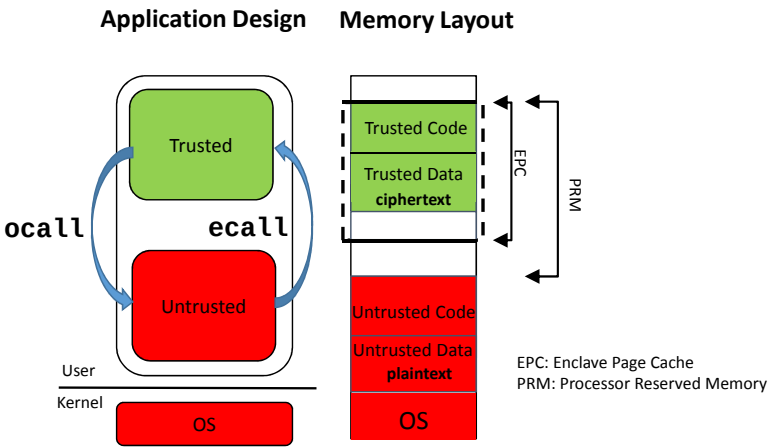
Outline

Performance Overhead



EPC: Enclave Page Cache
PRM: Processor Reserved Memory

Performance Overhead



EPC: Enclave Page Cache
PRM: Processor Reserved Memory

Runtime Overhead

1. `ecall`
2. `ocall`
3. enclave cache
4. memory encryption

Other Overhead

1. enclave creation
2. enclave deletion
3. SGX attestation

Instruction Level Overhead

Micro-benchmark	Description	Latency (cycles)
`ecall` (warm cache)	Calling an enclave w/o parameters, and immediately return	8,640
`ecall` (cold cache)	Same as above, but the entire cache is flushed	14,170
`ecall` (buffer)	to enclave	9,816
	Calling an enclave func, passing 2K buffer, from the enclave	11,172
	to and from enclave	10,827
`ocall` (warm cache)	Calling untrusted w/o parameters, and immediately return	8,314
`ocall` (cold cache)	Same as above, but the entire cache is flushed	14,160
`ocall` (buffer)	to untrusted	9,254
	Calling untrusted func, passing 2K buffer, from the untrusted	11,418
	to and from untrusted	9,801

Source: “Regaining Lost Cycles with HotCalls” [WBA17] http://www.ofirweisse.com/ISCA17_Ofir_Weisse.pdf

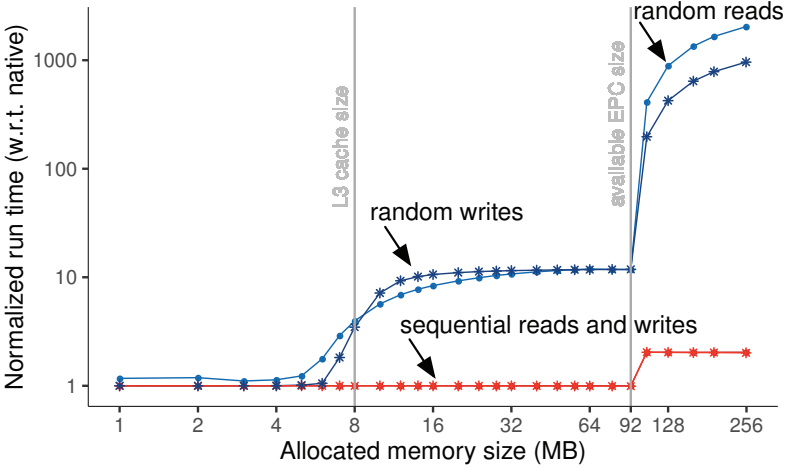
Instruction Level Overhead

Micro-benchmark	Description	Latency (cycles)
`ecall` (warm cache)	Calling an enclave w/o parameters, and immediately return	8,640
`ecall` (cold cache)	Same as above, but the entire cache is flushed	14,170
	to enclave	9,816
`ecall` (buffer)	Calling an enclave func, passing 2K buffer, from the enclave	11,172
	to and from enclave	10,827
`ocall` (warm cache)	Calling untrusted w/o parameters, and immediately return	8,314
`ocall` (cold cache)	Same as above, but the entire cache is flushed	14,160
	to untrusted	9,254
`ocall` (buffer)	Calling untrusted func, passing 2K buffer, from the untrusted	11,418
	to and from untrusted	9,801

Source: "Regaining Lost Cycles with HotCalls" [WBA17] http://www.ofirweisse.com/ISCA17_Ofir_Weisse.pdf

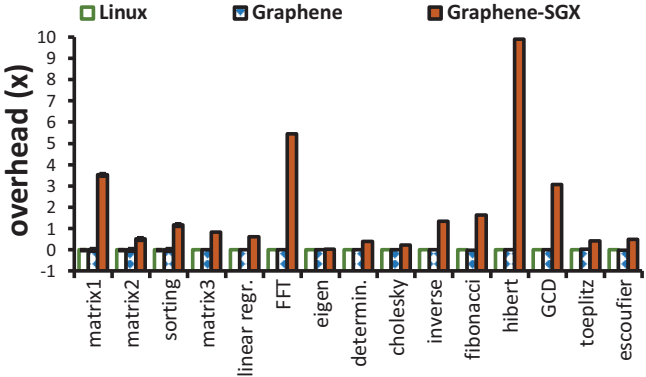
Hyper-calls to the hypervisor take about 1,300 cycles (KVM hypervisor on x86) [DLL+16]. A round-trip time for an Exception-Less System gettsc is modest at 150 cycles [SS10].

Memory Access Overhead Measured by SCONE [ATG+16]

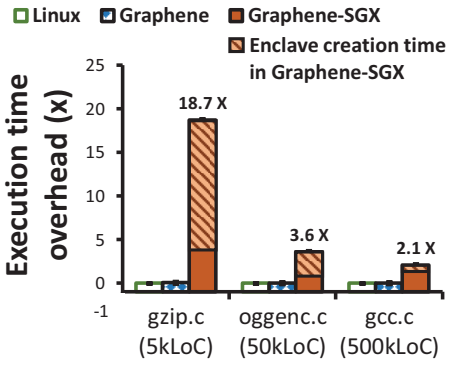


Source: https://www.usenix.org/system/files/conference/osdi16/osdi16-arnautov.pdf

Desktop Applications: Tested by Graphene-SGX [TPV17]



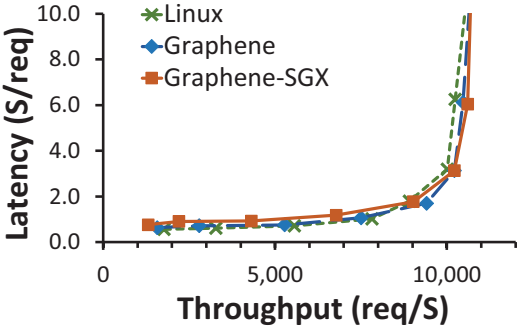
(a) R



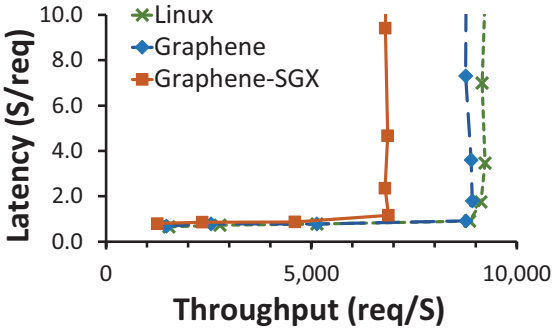
(b) GCC

Source: https://www.usenix.org/system/files/conference/atc17/atc17-tsai.pdf

Throughput of HTTP Servers: Tested by Graphene-SGX [TPV17]



(a) Lighttpd (25 threads)



(b) Apache (5 processes)

Source: https://www.usenix.org/system/files/conference/atc17/atc17-tsai.pdf

Outline

Outline

Cloud Applications

Source: https://www.rishabhsoft.com/wp-content/uploads/2014/10/Cloud_development_main.jpg

Server Side (e.g., Cloud) Applications

Paper Title	Venue	Application
Shielding Applications from an Untrusted Cloud with Haven	OSDI'14	App Hardening
VC3: trustworthy data analytics in the cloud using SGX	SP'15	Data Analytics
M2R: Enabling Stronger Privacy in MapReduce Computation	USENIX SEC'15	Data Analytics
Oblivious Multi-Party Machine Learning on Trusted Processors	USENIX SEC'16	Data Analytics
SCONE: Secure Linux Containers with Intel SGX	OSDI'16	App Hardening
SecureKeeper: Confidential ZooKeeper using Intel SGX	Middleware'16	Data Analytics
Attestation Transparency: Building secure Internet services for legacy clients	ASIACCS'16	Secure Service
S-NFV: Securing NFV states by using SGX	SDN-NFVSec'16	Cloud NFV
PANOPLY: Low-TCB Linux Applications with SGX Enclaves	NDSS'17	App Hardening
SGX-Log: Securing System Logs With SGX	ASIACCS'17	Securing Logging
Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX	ATC'17	App Hardening
Secure Live Migration of SGX Enclaves on Untrusted Cloud	DSN'17	Live Migration
Securing Data Analytics on SGX With Randomization	ESORICS'17	Data Analytics
SGX-BigMatrix: A Practical Encrypted Data Analytic Framework With SGX	CCS'17	Data Analytics
Enclave-Based Privacy-Preserving Alignment of Raw Genomic Information	SysTex'17	Genomic Computing

Server Side (e.g., Cloud) Applications

Paper Title	Venue	Application
Shielding Applications from an Untrusted Cloud with Haven	OSDI'14	**App Hardening**
VC3: trustworthy data analytics in the cloud using SGX	SP'15	Data Analytics
M2R: Enabling Stronger Privacy in MapReduce Computation	USENIX SEC'15	Data Analytics
Oblivious Multi-Party Machine Learning on Trusted Processors	USENIX SEC'16	Data Analytics
SCONE: Secure Linux Containers with Intel SGX	OSDI'16	**App Hardening**
SecureKeeper: Confidential ZooKeeper using Intel SGX	Middleware'16	Data Analytics
Attestation Transparency: Building secure Internet services for legacy clients	ASIACCS'16	Secure Service
S-NFV: Securing NFV states by using SGX	SDN-NFVSec'16	Cloud NFV
PANOPLY: Low-TCB Linux Applications with SGX Enclaves	NDSS'17	**App Hardening**
SGX-Log: Securing System Logs With SGX	ASIACCS'17	Securing Logging
Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX	ATC'17	**App Hardening**
Secure Live Migration of SGX Enclaves on Untrusted Cloud	DSN'17	Live Migration
Securing Data Analytics on SGX With Randomization	ESORICS'17	Data Analytics
SGX-BigMatrix: A Practical Encrypted Data Analytic Framework With SGX	CCS'17	Data Analytics
Enclave-Based Privacy-Preserving Alignment of Raw Genomic Information	SysTex'17	Genomic Computing

Server Side (e.g., Cloud) Applications

Paper Title	Venue	Application
Shielding Applications from an Untrusted Cloud with Haven	OSDI'14	App Hardening
VC3: trustworthy data analytics in the cloud using SGX	SP'15	**Data Analytics**
M2R: Enabling Stronger Privacy in MapReduce Computation	USENIX SEC'15	**Data Analytics**
Oblivious Multi-Party Machine Learning on Trusted Processors	USENIX SEC'16	**Data Analytics**
SCONE: Secure Linux Containers with Intel SGX	OSDI'16	App Hardening
SecureKeeper: Confidential ZooKeeper using Intel SGX	Middleware'16	**Data Analytics**
Attestation Transparency: Building secure Internet services for legacy clients	ASIACCS'16	Secure Service
S-NFV: Securing NFV states by using SGX	SDN-NFVSec'16	Cloud NFV
PANOPLY: Low-TCB Linux Applications with SGX Enclaves	NDSS'17	App Hardening
SGX-Log: Securing System Logs With SGX	ASIACCS'17	Securing Logging
Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX	ATC'17	App Hardening
Secure Live Migration of SGX Enclaves on Untrusted Cloud	DSN'17	Live Migration
Securing Data Analytics on SGX With Randomization	ESORICS'17	**Data Analytics**
SGX-BigMatrix: A Practical Encrypted Data Analytic Framework With SGX	CCS'17	**Data Analytics**
Enclave-Based Privacy-Preserving Alignment of Raw Genomic Information	SysTex'17	Genomic Computing

Server Side (e.g., Cloud) Applications

Paper Title	Venue	Application
Shielding Applications from an Untrusted Cloud with Haven	OSDI'14	App Hardening
VC3: trustworthy data analytics in the cloud using SGX	SP'15	Data Analytics
M2R: Enabling Stronger Privacy in MapReduce Computation	USENIX SEC'15	Data Analytics
Oblivious Multi-Party Machine Learning on Trusted Processors	USENIX SEC'16	Data Analytics
SCONE: Secure Linux Containers with Intel SGX	OSDI'16	App Hardening
SecureKeeper: Confidential ZooKeeper using Intel SGX	Middleware'16	Data Analytics
Attestation Transparency: Building secure Internet services for legacy clients	ASIACCS'16	**Secure Service**
S-NFV: Securing NFV states by using SGX	SDN-NFVSec'16	**Cloud NFV**
PANOPLY: Low-TCB Linux Applications with SGX Enclaves	NDSS'17	App Hardening
SGX-Log: Securing System Logs With SGX	ASIACCS'17	**Securing Logging**
Graphene-SGX: A Practical Library OS for Unmodified Applications on SGX	ATC'17	App Hardening
Secure Live Migration of SGX Enclaves on Untrusted Cloud	DSN'17	**Live Migration**
Securing Data Analytics on SGX With Randomization	ESORICS'17	Data Analytics
SGX-BigMatrix: A Practical Encrypted Data Analytic Framework With SGX	CCS'17	Data Analytics
Enclave-Based Privacy-Preserving Alignment of Raw Genomic Information	SysTex'17	**Genomic Computing**

Outline

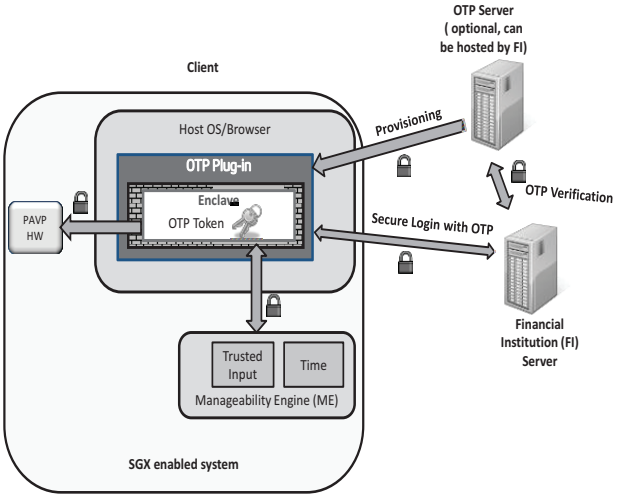
Desktop Applications

Source: https://www.azernews.az/media/pictures/personality-profiling-software-and-techniques.jpg

Client Side Applications

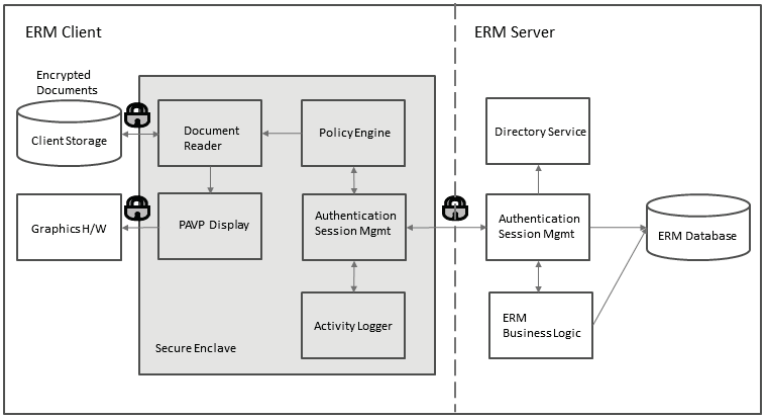
Paper/News Title	Venue	Application
Using Innovative Instructions to Create Trustworthy Software Solutions	Hasp'13	OTP, ERM, Video Conferencing
Password Manager with Intel SGX	Tutorial@16	Password Manager
Numecent to Show Off Pioneering Application Delivery Platform	IDF'16	DRM
A Case for Protecting Computer Games With SGX	SysTEX'16	Game Protection
TrustJS: Trusted Client-side Execution of JavaScript	EuroSec'17	Trusted Script Execution

Client Applications: One Time Pad (OTP)



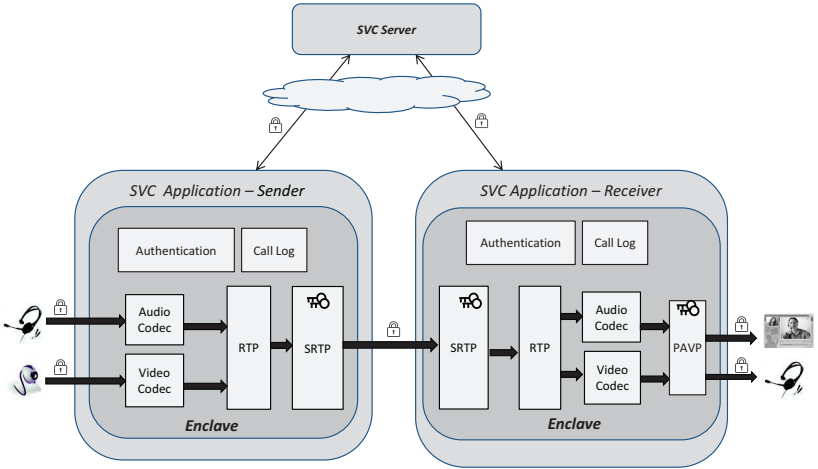
Source: https://software.intel.com/sites/default/files/article/413936/hasp-2013-innovative-instructions-and-software-model-for-isolated-execution.pdf

Client Applications: Enterprise Rights Management



Source: https://software.intel.com/sites/default/files/article/413936/hasp-2013-innovative-instructions-and-software-model-for-isolated-execution.pdf

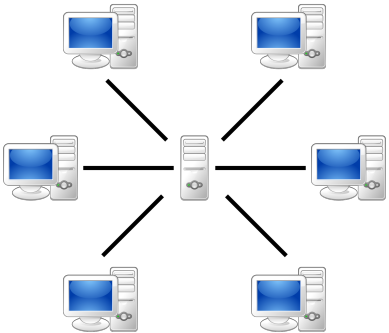
Client Applications: Secure Video Chatting

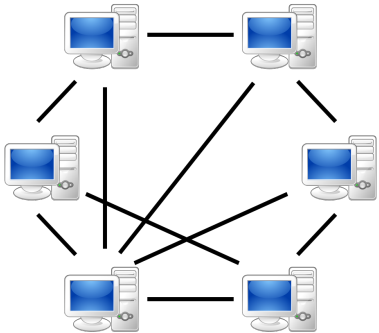


Source: https://software.intel.com/sites/default/files/article/413936/hasp-2013-innovative-instructions-and-software-model-for-isolated-execution.pdf

Outline

Client Server Model, and Distributed Computing Model





Source: https://en.wikipedia.org/wiki/Peer-to-peer

Distributed Computing Applications

Paper Title	Venue	Application
Ryoan: a distributed sandbox for untrusted computation on secret data	OSDI'16	App Hardening
Proof of Luck: an Efficient Blockchain Consensus Protocol	SysTEX'16	Byzantine fault tolerance
Town Crier: An Authenticated Data Feed for Smart Contracts	CCS'16	Smart Contracts
Secure Content-Based Routing Using Intel SGX	Middleware'16	Content-Based Routing
Enhancing Security and Privacy of Tor's Ecosystem by Using TEE	NSDI'17	Tor nework
Hybrids on Steroids: SGX-Based High Performance BFT	EuroSys'17	Byzantine fault tolerance

Outline

SGX Summary

1. A hardware assured TEE that secures applications w/ minimal attack surface

SGX Summary

1. A hardware assured TEE that secures applications w/ minimal attack surface
2. Two ways of developing SGX applications
   1. Developing from scratch by partitioning applications
   2. Executing the native (legacy) apps in an enclave w/o partitioning

SGX Summary

1. A hardware assured TEE that secures applications w/ minimal attack surface
2. Two ways of developing SGX applications
   1. Developing from scratch by partitioning applications
   2. Executing the native (legacy) apps in an enclave w/o partitioning
3. The major runtime overhead of SGX applications come from I/O, enclave encrypted memory access, and enclave cache misses.

SGX Summary

1. A hardware assured TEE that secures applications w/ minimal attack surface
2. Two ways of developing SGX applications
   1. Developing from scratch by partitioning applications
   2. Executing the native (legacy) apps in an enclave w/o partitioning
3. The major runtime overhead of SGX applications come from I/O, enclave encrypted memory access, and enclave cache misses.
4. The killer applications of SGX is for securing remote execution, such as data analytics (e.g., MapReduce, Machine learning) in the cloud, and distributed computing (e.g., Tor, bitcoins).

Outline

References I

Tiago Alves, Trustzone: Integrated hardware and software security, White Paper (2004).

Sergei Arnautov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumaran, Daniel O'Keeffe, Mark L Stillwell, David Goltzsche, Dave Eyers, Rüdiger Kapitza, Peter Pietzuch, and Christof Fetzer, Scone: Secure linux containers with intel sgx, 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI) (Savannah, GA, USA), USENIX, 11/2016 2016.

Andrew Baumann, Marcus Peinado, and Galen Hunt, Shielding applications from an untrusted cloud with haven, 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14) (CO), USENIX Association, 2014, pp. 267–283.

Andrew Baumann, Marcus Peinado, and Galen Hunt, Shielding applications from an untrusted cloud with haven, ACM Transactions on Computer Systems (TOCS) **33** (2015), no. 3, 8.

Victor Costan and Srinivas Devadas, Intel sgx explained, IACR Cryptology ePrint Archive **2016** (2016), 86.

Christoffer Dall, Shih-Wei Li, Jin Tack Lim, Jason Nieh, and Georgios Koloventzos, Arm virtualization: performance and architectural implications, Proceedings of the 43rd International Symposium on Computer Architecture, IEEE Press, 2016, pp. 304–316.

William Futral and James Greene, Intel trusted execution technology for server platforms: A guide to more secure datacenters, Apress, 2013.

Joshua Lind, Christian Priebe, Divya Muthukumaran, Dan O'Keeffe, Pierre-Louis Aublin, Florian Kelbert, Tobias Reiher, David Goltzsche, David Eyers, Rüdiger Kapitza, Christof Fetzer, and Peter Pietzuch, Glamdring: Automatic application partitioning for intel SGX, 2017 USENIX Annual Technical Conference (USENIX ATC 17) (Santa Clara, CA), USENIX Association, 2017, pp. 285–298.

References II

Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar, Innovative instructions and software model for isolated execution, Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP) (Tel-Aviv, Israel), 2013, pp. 1–8.

G. Edward Suh, Dwaine Clarke, Blaise Gassend, Marten van Dijk, and Srinivas Devadas, Aegis: Architecture for tamper-evident and tamper-resistant processing, Proceedings of the 17th Annual International Conference on Supercomputing (New York, NY, USA), ICS '03, ACM, 2003, pp. 160–171.

Intel software guard extensions (intel sgx) sdk, https://software.intel.com/en-us/sgx-sdk.

Livio Soares and Michael Stumm, Flexsc: Flexible system call scheduling with exception-less system calls, Proceedings of the 9th USENIX conference on Operating systems design and implementation, USENIX Association, 2010, pp. 33–46.

Sean W Smith and Steve Weingart, Building a high-performance, programmable secure coprocessor, Computer Networks **31** (1999), no. 8, 831–860.

TPM, Tpm main specification, http://www.trustedcomputinggroup.org/resources, 2003.

Chiache Tsai, Donald E. Porter, and Mona Vij, Graphene-sgx: A practical library OS for unmodified applications on SGX, 2017 USENIX Annual Technical Conference (USENIX ATC 17) (Santa Clara, CA), USENIX Association, 2017, pp. 645–658.

Leendert Van Doorn, Hardware virtualization trends, ACM/Usenix International Conference On Virtual Execution Environments: Proceedings of the 2 nd international conference on Virtual execution environments, vol. 14, 2006, pp. 45–45.

References III


Ofir Weisse, Valeria Bertacco, and Todd Austin, Regaining lost cycles with hotcalls: A fast interface for sgx secure enclaves, Proceedings of the 44th Annual International Symposium on Computer Architecture (New York, NY, USA), ISCA '17, ACM, 2017, pp. 81–93.


Bennet Yee, Using secure coprocessors, Tech. report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1994.