Exploring Intel Software Guard Extensions (SGX)

Date: February 05, 2014

Investigators:

1. Praveen Keshavamurthy
   Masters in Computer Science
   Email: keshavamurthy.p@husky.neu.edu
2. Aboobacker Rizwan Payapura
   Masters in Computer Science
   Email: rizwan@ccs.neu.edu

General Field: The project falls within Cloud Security, where the resources are shared between untrusted applications.

Keywords: cloud security, software guard extensions, isolated containers, hardware security feature, resource sharing

Project Description: Intel SGX, which provides hardware security features is a major step towards a better security in cloud computing, but it is mostly untested by researchers. So the exact limitations and capabilities of this processor is still not very clear. We will be exploring Intel SGX by implementing a sample application in this project and benchmarking the performance and integrity of the processor.

Project Responsibilities:

1. Praveen Keshavamurthy: Lead Designer, Software Developer
2. Aboobacker Rizwan Payapura: Lead Researcher, Software Developer

Total Budget: $16200

Deliverables: Sample client-server e-voting application which uses Intel SGX features, benchmark report on performance and integrity, SGX environment to carry out side channel attack.

Executive Summary

Trusted Execution Environment – Intel Software Guard Extensions (SGX)

Secure remote computation, offloading processing to a remote computer owned and maintained by an untrusted party with some integrity and privacy guarantees, has remained an unsolved problem. This is mainly due to the hierarchical security model that aims only to protect the privileged code from untrusted code and does not fare well to protect user data from access by privileged code. Although complete homomorphic encryption solves the problem for a limited family of computations, it has an impractical performance overhead. Intel SGX is the latest technology which aims to solve the secure remote computation problem by leveraging trusted processors on remote computers.

Intel SGX is an architecture extension designed to increase the security of software through an "inverse sandbox" mechanism. At its root, Intel SGX is a set of new CPU instructions, using which legitimate software can be sealed inside an enclave (secured memory) and protected from attack by the malware, irrespective of the privilege level of the latter. It also provides a novel hardware assisted mechanism to allow secure remote attestation and sealing to application software. Intel SGX technology is the need of the hour for applications related to Data Rights Management (DRMs) and in designing a novel architecture for trusted cloud. As Intel SGX was available to select group of researchers until recently, there is a lack of current research and programs that evaluate the technology with respect to performance impacts on existing state-of-the-art solutions.

In this project, we dissect the security features of Intel SGX by developing a secure e-voting client-server model. We will benchmark the performance of the Intel SGX by measuring the overhead, the technology brings in on secure software execution. We plan to repeat cross-vm side channel attacks to retrieve the secret key using applications built on top of Intel SGX.

Motivation

There is a lot of trust involved in a cloud infrastructure. In a security perspective 'trust' is a bad thing. But due to the complexity and performance issues many protocol is designed based on trust. For instance creating an app and hosting it in the cloud platform involves many level of trusts. The app developer needs to trust the Operating System (OS) in the cloud (which have high privilege access), the cloud provider, and worst of all other applications hosted in the same platform. The code base of OS is really huge and written by thousands of developers around the world. There is a very high possibility for a bad code in such a huge code base. Even though the bad codes are found and fixed everyday more complex features are getting added to the code base introducing more bugs. Since the OS kernel operates in ring 0 (high privileged), the attacker can make use of these bad codes to get a high privileged access and thus can manipulate the private data of other applications hosted in that platform, which operates in lower privileged user space (ring 3). So trusting such platform is a very risky thing to do, but the app developers don't have any other options. This problem can be solved by reducing the level of trust which will substantially reduce the attack surface. For achieving this hardware support is a very important step. It can help reducing the trust we place on OS and other software. If processor can control the OS access on private data that will be a huge step towards reducing the level of trust.

Intel Software Guard Extensions (Intel SGX) is designed for this purpose through an inverse sandbox mechanism. It solves the above problem by preserving the confidentiality and integrity of sensitive code and data of the applications without disrupting the ability of legitimate system software to schedule and manage the use of platform resources. In this architecture, rather than identifying malware or any other adversaries on the platform, applications can use sealed enclaves to protect the data and code from misuse by higher privileged applications such as OS with bad code. Since applications create their own enclave and the processor makes sure that the access to the enclaves is limited to only that application, the trusted components of the application are protected. Untrusted part of the application (which mostly contain interface to communicate with OS and Input/Output) can call to the trusted functions in the enclave in order to access the data in the enclave. A remote user can verify that the user is communicating with the untampered application in the cloud running inside an Intel SGX enclave by measuring the application's trusted code and produce a signed attestation (rooted in the processor) that includes this measurement and other certification that the code has been correctly initialized in a trustable environment.

Intel SGX is a major step towards better security in cloud computing, but it is mostly untested by researchers. So the exact limitations and capabilities of this processor is still not very clear. We would be exploring Intel SGX by implementing a sample application in this project.

Previous Work

Intel SGX is the latest in the long list of trusted execution technologies. In this section we examine few of the previous secure technologies and their pros and cons. This will give us a clear insight towards the design principles of Intel SGX.

IBM 4765 Secure Coprocessor encapsulate an entire computer system, including a CPU, a cryptographic accelerator, caches, DRAM, and an I/O controller within a tamper-resistant environment. The secure coprocessor destroys the secrets that it stores when an attack is detected. This approach has good security properties against physical attacks, but tamper-resistant enclosures are very expensive, relatively to the cost of a computer system.

ARM's TrustZone is a collection of hardware modules that can be used to conceptually partition a system's resources between a secure world, which hosts a secure container, and a normal world, which runs an untrusted software stack. ARM's intellectual property core (IP core) blocks do provide secure boot process, prevents attacks from malicious OS with respect to direct memory probing and page fault handling, but secure world is limited to on-chip SRAM, which limits it's capability in extending it to server scale processing.

The execute-only memory (XOM) architecture introduced the approach of having sensitive code and data execute in isolated containers on DRAM, managed by untrusted host software. XOM outlined the mechanisms needed to isolate a container's data from its untrusted software environment, such as saving the register state to a protected memory area before servicing an interrupt. XOM's design cannot guarantee DRAM freshness, so the software in its containers is vulnerable to physical replay attacks.

The Trusted Platform Module (TPM) introduced software attestation model with attestation key stored in tamper resistant chip. The TPM design provides one isolation container, covering all the software running on the computer that has the TPM chip. It follows that the measurement included in an attestation signature covers the entire OS kernel and all the kernel modules, such as device drivers. However, commercial computers use a wide diversity of devices, and their system software is updated at an ever-increasing pace, so it is impossible to maintain a list of acceptable measurement hashes corresponding to a piece of trusted software. Due to this issue, the TPM's software attestation is not used in many security systems, despite its wide deployment.

Intel's Trusted Execution Technology (TXT) uses the TPM's software attestation model and auxiliary tamper-resistant chip, but reduces the software inside the secure container to a virtual machine (guest operating system and application) hosted by the CPU's hardware virtualization features. TXT isolates the software inside the container from the untrusted code by storing Static Root of Trust Measurement (SRTM) in the TPM chip during boot cycle and it updates those TPM registers when VM is initialized making up the Dynamic Root of Trust Measurement (DRTM). TXT

does not implement DRAM encryption or HMACs, and therefore is vulnerable to physical DRAM attacks, just like TPM-based designs.

Lastly, we examined Bastion architecture which introduced secure containers by maintaining an inverted page map, associating each physical memory page to its container and virtual address. The system software verifies the virtual address used for address translation matches with the expected virtual address associated in the page map before granting access. Intel SGX follows this approach.

Intel SGX has evolved from all the previous implementations of trusted execution environment. Intel SGX has a very robust design which includes DRAM Encryption & Integrity protection, invert cache map which protects from illegal memory access and an on chip tamper resistant chip which holds the processor secret key.

Intel SGX technology was only available to select group of research workers until recently. The previous works on Intel SGX include adapting legacy applications, including SQL Server and Apache, on a commodity OS (Windows) and commodity hardware. Using trusted SGX processors as a building block, a prototype was developed by adapting Hadoop distribution framework where in the simple map and reduce jobs were bounded within an enclave, thereby reducing the trusted computation base (TCB) and hence the privacy and integrity of the data. This work has also benchmarked the performance saying average runtime overhead being 8% with read/write integrity.

As part of this project, we are trying to reproduce the cross-vm side channel attack with applications using Intel SGX technology. We will also try to identify any possible new side channel attack with use of Intel SGX. We plan to evaluate key revocation functionality of Intel SGX, which we feel is one of the important feature needed in case of an adversary.

Specific Aims

In this project, we intend to achieve following goals:

1. Explore Intel SGX Architecture.
2. Familiarize Intel SGX programming and SDK support.
3. Design and develop a sample e-voting client - server application with server running on Intel SGX. Demonstrate how Intel SGX provides more security than any other existing system used in Cloud platform.
4. Find the limitations of Intel SGX during the implementation.
5. Compare the performance with a normal equivalent application.
6. Create Intel SGX environment to carry out cache based side channel attacks, by porting the libgcrypt library (which has a known memory leak) to the Intel SGX platform.
    - Docker containers will be used by the applications to share the cache for the side channel attack.

Plan

Assuming two week sprints, we will have 6 sprints. We plan as following to execute this project successfully.

- Sprint-1: Understand Design of Intel SGX Architecture. Develop a “hello-world” program which make use of Intel SGX technology.
- Sprint-2: Develop a simple e-voting client-server program
- Sprint-3: Explore the boundaries of Intel SGX. Benchmark the performance impacts of Intel SGX. Document the same.
- Sprint-4: Analyze the feasibility of setting up an environment using Intel SGX technology for side channel attacks using windows containers
- Sprint-5: Develop the application using Intel SGX required for the attack model.
- Sprint-6: Verify the results with the available side channel attack results without the use of Intel SGX. Document the same.

Deliverables

At the end of this project, we intend to deliver the following:

1. A sample e-voting client - server application with server running on Intel SGX developed in C.
2. Benchmark report on the applications performance and integrity.
    - Based on the CPU overhead created by Intel SGX features.
3. Intel SGX environment for cache based side channel attack using the libgcrypt library to check whether Intel SGX features stop such side channel attacks.

Issues

We foresee the following potential issues:

- We are yet to explore the support of Intel SGX SDK. We are not sure about the OS libraries which are ported to work alongside Intel SGX. This might impede our development.
- Intel are not yet giving production licenses for developing Intel SGX applications. But excerpt from Intel SGX FAQs says “Developers must be willing to Sign their own apps (ideally are currently doing so) and Proxy access to attestation service”. We will be sure of this once we start the implementation.

Bibliography

1. Victor Costan and Srinivas Devadas. Intel SGX Explained. https://eprint.iacr.org/2016/086.pdf
2. Andrew Baumann, Marcus Peinado, and Galen Hunt, Microsoft Research. Shielding Applications from an Untrusted Cloud with Haven.
3. Felix Schuster, Manuel Costa, Cedric Fournet, Christos Gkantsidis, Marcus Peinado, Gloria Mainar-Ruiz, and Mark Russinovich, Microsoft Research. VC3: Trustworthy Data Analytics in the Cloud.
4. Prerit Jain, Soham Desai, Seongmin Kim, Ming-Wei Shih, JaeHyuk Lee, Changho Choi, Youjung Shin, Taesoo Kim, Brent B. Kang and Dongsu Han, "OpenSGX: An Open Platform for SGX Research", In Proceedings of the 2016 Network and Distributed System Security Symposium (NDSS 2016), 2016.
5. Gehana Booth, Andrew Soknacki, and Anil Somayaji. Cloud Security: Attacks and Current Defenses.
6. Intel SGX Emulation using QEMU. [Online] https://tc.gtisc.gatech.edu/bss/2014/l/final/pjain43.pdf
7. Intel Software Guard Extensions. [Online] https://software.intel.com/sites/default/files/332680-002.pdf
8. Using Innovative Instructions to Create Trustworthy Software Solutions. [Online] https://software.intel.com/en-us/articles/using-innovative-instructions-to-create-trustworthy-software-solutions.
9. Intel Software Guard Extensions Programming Reference [Online] https://software.intel.com/sites/default/files/managed/48/88/329298-002.pdf

Biographical sketches

Praveen Keshavamurthy is pursuing Masters in Computer Science at Northeastern University. His primary area of interest is Networking and Cloud security. He was part of cloud platform development during his tenure in Nokia Networks.

Aboobacker Rizwan Payapura is pursuing Masters in Computer Science at Northeastern University. His primary area of interest is Network Security, Cloud Security and Cryptography. He worked for Platform Infrastructure Team in Akamai Technologies.

Schedule

Date	Goals
10th Feb	Understand high level architecture of Intel SGX
24th Feb	Develop simple e-voting client-server application using Intel SGX
9th March	Benchmark boundaries and performance impacts of Intel SGX
23rd March	Report of analysis of building environment for side-channel attack using intel SGX
6th April	Develop application related to side channel attack using Intel SGX
20th April	Document the findings and report the results obtained
27th April	Final Report Submission and a Demo

Budget

Direct Costs	Amount
Investigators x 2 (165 hrs @ $40/hr)	13200
Hardware x 2 (Laptop)	3000
Total	16200

Broader Impact

As of now, Intel SGX is only supported in certain upcoming versions of desktop and mobile skylake processors. This limits the application use cases to client devices such as DRMs. Nonetheless Intel have proposed to come up with server processors supporting Intel SGX by next year. This greatly increases the significance of this project where we are completely exploring the boundaries and performance impacts of using Intel SGX technology.

Although Intel do mention that Intel SGX do not explicitly enhance side channel protection, at least from the design perspective, we believe that the Intel SGX might have some impacts on the already proven side channel attacks because of the isolated enclave and the new memory encryption engine which induces additional delay. Thus we are trying to simulate the side-channel attack of retrieving the secret key from the victim virtual machine in Intel SGX environment. This will give a good insight of additional features of Intel SGX which will lay a solid foundation going forward in the development of trusted cloud.

Appendix A: Research Conference

A recent publication titled: “Intel SGX Explained”, by Victor Costan and Srinivas Devadas is most relevant to our project. It gives the summary of the Intel-specific architectural and micro-architectural details needed to understand SGX, a detailed and structured presentation of the publicly available information on SGX, a series of intelligent guesses about some important but undocumented aspects of SGX, and an analysis of SGX’s security properties.