CCS’17 Tutorial Abstract / SGX Security and Privacy

Taesoo Kim Zhiqiang Lin[†] Chia-che Tsai[‡]

Georgia Institute of Technology *The University of Texas at Dallas*[†] *Stony Brook University & UC Berkeley*[‡]

ABSTRACT

In this tutorial, we will first introduce the basic concepts of Intel SGX, its development workflows, potential applications and performance characteristics. Then, we will explain known security concerns, including cache/branch side-channel attacks and memory safety issues, and corresponding defenses with various working demos. Last but not least, we will introduce various ways to quickly start writing SGX applications, especially by utilizing library OSes or thin shielding layers; we will explain the pros and cons of each approach in terms of security and usability.

CCS CONCEPTS

• **Security and privacy** → Systems security; Security in hardware;

KEYWORDS

Intel SGX; TEE; Library OS

1 INTRODUCTION

The Intel Software Guard Extensions (SGX)—a game-changing feature introduced in the recent Intel Skylake CPU—is a new technology likely to make secure and trustworthy computing in a hostile environment practical. At a high level, SGX consists of a set of new instructions that can be used to create secure regions (i.e., enclaves) to defeat attacks that aim to steal or tamper with the data within an enclave. Without a doubt, we expect that SGX will allow developers to protect sensitive code and data from unauthorized access or modification by software running at higher privilege levels such as an OS or a hypervisor.

However, SGX is merely a set of instructions; it lacks support from the OS and libraries. These deficiencies allow programmers to easily introduce naive yet preventable bugs that often lead to critical security holes in an enclave program [2]. Further, designing a correct and secure SGX infrastructure is also far from straightforward; enclave programs rely on the support of an underlying OS, but their security models exclude the OS from the TCB. This unconventional dependency makes various attack vectors, which are often considered impractical in a traditional setting, immediate and practical, especially in a cloud environment.

In this tutorial, we will first provide the basics of Intel SGX, covering workflows, potential applications and performance characteristics. Then, we will explain its security concerns, including cache/branch side-channel attacks [3], controlled-channel attacks [8], and traditional memory safety issues [2, 4], and potential defenses [5] with various demos. Last but not least, we will introduce various ways to quickly start writing SGX applications on Linux, especially by utilizing library OSes [7] or thin shielding layers [1, 6]; we will explain the pros and cons of each approach in terms of security and usability.

REFERENCES

[1] Arnautox, S., Tarch, B., Gregor, F., Knauth, T., Martin, A., Priebe, C., Lind, J., Muthukumaran, D., O’Keeffe, D., Stillwell, M. L., Goltzsche, D., Eyers, D., Kapitza, R., Pietzuch, P., and Fetzer, C. SCONE: Secure Linux containers with Intel SGX. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)* (Savannah, GA, Nov. 2016).

[2] Lee, J., Jang, J., Jang, Y., Kwak, N., Choi, Y., Choi, C., Kim, T., Peinado, M., and Kang, B. B. Hacking in Darkness: Return-oriented Programming against Secure Enclaves. In *Proceedings of the 26th USENIX Security Symposium (Security)* (Vancouver, Canada, Aug. 2017).

[3] Lee, S., Shih, M.-W., Gera, P., Kim, T., Kim, H., and Peinado, M. Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing. In *Proceedings of the 26th USENIX Security Symposium (Security)* (Vancouver, Canada, Aug. 2017).

[4] Seo, J., Lee, B., Kim, S., Shih, M.-W., Shin, I., Han, D., and Kim, T. SGX-Shield: Enabling Address Space Layout Randomization for SGX Programs. In *Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS)* (San Diego, CA, Feb. 2017).

[5] Shih, M.-W., Lee, S., Kim, T., and Peinado, M. T-SGX: Eradicating Controlled-Channel Attacks Against Enclave Programs. In *Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS)* (San Diego, CA, Feb. 2017).

[6] Shinde, S., Tien, D. L., Tople, S., and Saxena, P. Panoply: Low-TCB Linux applications with SGX enclaves. In *Proceedings of the 2017 Annual Network and Distributed System Security Symposium (NDSS)* (San Diego, CA, Feb. 2017).

[7] Tsai, C.-C., Porter, D. E., and Vij, M. Graphene-SGX: A practical library OS for unmodified applications on SGX. In *Proceedings of the 2017 USENIX Annual Technical Conference (ATC)* (Santa Clara, CA, July 2017).

[8] Xu, Y., Cui, W., and Peinado, M. Controlled-channel attacks: Deterministic side channels for untrusted operating systems. In *Proceedings of the 36th IEEE Symposium on Security and Privacy (Oakland)* (San Jose, CA, May 2015).

Taesoo Kim. is an Assistant Professor in the School Computer Science at Georgia Tech. He also serves as the director of the Georgia Tech Systems Software and Security Center (GTS3). He is interested in building a system that has underline principles for why it should be secure. Those principles include the design of a system, analysis of its implementation, and clear separation of trusted components. His thesis work, in particular, focused on detecting and recovering from attacks on computer systems. He holds a BS from KAIST (2009), a SM (2011) and a Ph.D. (2014) from MIT in CS.

CCS ’17, October 30-November 3, 2017, Dallas, TX, USA

ACM ISBN 978-1-4503-4946-8/17/10.
https://doi.org/10.1145/3133956.3136068

Zhiqiang Lin. is an Associate Professor of Computer Science at The University of Texas at Dallas. He earned his PhD from Computer Science Department at Purdue University in 2011. His primary research interests are systems and software security, with an emphasis on developing program analysis techniques and applying them to secure both application programs including mobile apps and the underlying system software such as Operating Systems and hypervisors. Dr. Lin is a recipient of the NSF CAREER Award and the AFOSR Young Investigator Award.

Chia-Che Tsai. is a PhD candidate at Stony Brook University, and will soon join the RISE Lab at UC Berkeley as a postdoc researcher. He is also joining the Computer Science and Engineering department of Texas A&M University in Fall 2018 as a faculty. He is interested in building OSes and runtimes with a balance between usability, security, and performance. He is the main contributor to the Graphene library OS, an open-source framework for reusing unmodified Linux applications on Intel SGX and other various host options.