Tutorial: Building Distributed Enclave Applications with Sancus and SGX

Jan Tobias Mühlberg and Jo Van Bulck
imec-DistriNet, KU Leuven, Celestijnenlaan 200A, B-3001 Belgium

Abstract—**Trusted computing technology provides promising new security primitives for the development of dependable and highly secure distributed applications. Protected software components, or *enclaves*, are provided with strong integrity, confidentiality and key derivation mechanisms via a minimal, hardware-only trusted computing base. Yet, leveraging enclaved execution in practice remains not well-understood, especially for heterogeneous trusted computing networks. This tutorial aims to provide insights and hands-on exercises on building such interactive applications based on Intel SGX and the lightweight embedded Sancus architecture.**

I. Motivation and Goals

Trusted computing architectures such as Intel SGX [1], ARM TrustZone, and Sancus [2] have been around for a number of years. By enforcing strong integrity, confidentiality, and attestation guarantees with a minimal (hardware-only) trusted computing base, these architectures aim to provide a root-of-trust for the development of dependable and highly secure systems. However, few real-world applications leverage trusted computing, and approaches to interconnect *enclave* applications in heterogeneous and distributed trusted computing environments remain not well-understood. Yet, this approach provides strong security benefits in domains like the Internet of Things or interconnected Industrial Control Systems.

This tutorial outlines the development of such security-sensitive networked applications. Participants will learn how to write enclaved software modules for both the embedded open-source Sancus research architecture, as well as for off-the-shelf Intel SGX platforms. We will show how to make Sancus/SGX enclaves interact securely in a distributed fashion. Our approach is distinguished by the concept of "authentic execution" [3], a notion of end-to-end security and secure I/O where the overall application's behavior depends solely on protected device drivers and data processing enclaves.

The tutorial will cover common pitfalls for enclave development and feature hands-on exercises on programming, deploying, attesting, and interacting with basic Sancus/SGX enclaves. The tutorial setup uses an automotive industry-standard CAN bus to interconnect Intel SGX machines with Sancus-enabled microcontrollers. We will provide Sancus I/O driver modules plus a CAN authentication library [4] that allows participants to easily setup secure communication channels.

Our tutorial aims at researchers, software developers, and software architects who want to learn more about modern trusted computing architectures. Basic programming skills (in C) and familiarity with the GNU/Linux command line interface are required to follow the hands-on exercises.

II. Organisation of the Tutorial

Participants will be provided with remote SSH access to an SGX-capable machine, connected to Sancus-enabled microcontrollers over a CAN network. This machine will also have pre-installed the required software development tools for the two trusted computing architectures, as well as skeleton source code for the hands-on exercises.

The tutorial aims to fill about three hours with an equal split between lectures and practical exercises:

Duration	Content
45'	**Introductory lecture** • Trusted Computing and remote attestation • Authentic Execution on heterogeneous Trusted Computing infrastructure
45'	**Hands-on:** Developing Sancus/SGX enclaves, software deployment, remote attestation and secure communication
45'	**Hands-on:** Secure I/O on Sancus, authentic execution in a distributed setting
45'	**Pitfalls, advanced topics, wrap-up** • Pitfalls: untrusted pointers, function pointers, resource sharing, debugging • Compartmentalising applications • Side-channels and how to avoid them • Open research questions

All slides and software for this tutorial are available at https://distrinet.cs.kuleuven.be/software/sancus/tutorial.php.

Acknowledgment

This research is partially funded by the Research Fund KU Leuven. Jo Van Bulck is supported by a doctoral grant of the Research Foundation – Flanders (FWO).

References

[1] V. Costan and S. Devadas, "Intel SGX explained." *IACR Cryptology ePrint Archive*, vol. 2016, p. 86, 2016.

[2] J. Noorman, J. Van Bulck, J. T. Mühlberg, F. Piessens, P. Maene, B. Preneel, I. Verbauwhede, J. Götzfried, T. Müller, and F. Freiling, "Sancus 2.0: A low-cost security architecture for IoT devices," *ACM Transactions on Privacy and Security (TOPS)*, vol. 20, pp. 7:1–7:33, 2017.

[3] J. Noorman, J. T. Mühlberg, and F. Piessens, "Authentic execution of distributed event-driven applications with a small TCB," in *STM '17*, ser. LNCS, vol. 10547. Heidelberg: Springer, 2017, pp. 55–71.

[4] J. Van Bulck, J. T. Mühlberg, and F. Piessens, "VulCAN: Efficient component authentication and software isolation for automotive control networks," in *Proceedings of the 33th Annual Computer Security Applications Conference (ACSAC'17)*. ACM, 2017.