ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/303831164

# Toward Live Migration of SGX-Enabled Virtual Machines

**Conference Paper** · June 2016

DOI: 10.1109/SERVICES.2016.23

CITATIONS
0

READS
171

**4 authors**, including:

Jaemin Park
The Affiliated Institute of ETRI
**13** PUBLICATIONS **228** CITATIONS
SEE PROFILE

Sungjin Park
The Affiliated Institute of ETRI
**5** PUBLICATIONS **1** CITATION
SEE PROFILE

All content following this page was uploaded by Jaemin Park on 15 July 2016.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

# Toward Live Migration of SGX-Enabled Virtual Machines

Jaemin Park, Sungjin Park, Jisoo Oh and Jong-Jin Won
The Affiliated Institute of ETRI
Daejeon, Republic of Korea
{jmpark, taiji, jsoh77, wonjj}@nsr.re.kr

***Abstract*—Intel Software Guard Extensions (SGX) can address the information disclosure in cloud computing. However, the existing virtual machine managers do not provide the efficient management operations of the SGX-enabled virtual machines (VMs) like live migration. In this paper, we identify challenges and propose a novel approach and its implementation model to migrate the SGX-enabled VMs. As future work, we will design the protocol and new instructions for live migration of the SGX-enabled VMs, and implement them on top of OpenSGX, an open source SGX emulator.**

***Index Terms*—Intel Software Guard Extensions, Enclave, Live Migration, Cloud Computing**

## I. Introduction

Even though cloud computing has been widely adopted in various areas, the security administrators still hesitate to introduce cloud computing to their own organizations due to several security concerns like information disclosure. First, the malicious insider can intentionally access sensitive information owned by a victim organization's virtual machine (VM) by using the cloud management operations. Second, in the multi-tenant environment, attackers could coincidently access sensitive information that resides in other guest VMs due to hypervisor vulnerabilities such as CVE-2015-3340 [1].

Intel Software Guard Extensions (SGX) [2] can be a good candidate that may address information disclosure in cloud computing. SGX creates *enclaves* for applications that protect security sensitive code and data from malicious access. The enclave code and data are stored in Processor Reserved Memory (PRM), a part of DRAM invisible to other softwares. Because the CPU fetches the enclave code and data from the PRM as the encrypted form, the enclave code and data can be protected from the external access as well as from the probing attack on the DRAM bus. This feature prevents the insider attacker from accessing the sensitive information. SGX also promises that any software layer in a machine is not trusted, so that the enclave is designed as a secure container that can be protected from the higher privileged softwares such as the VMM. Therefore, the VMM cannot notice the contents in the enclave even though it has information leak vulnerabilities.

However, there are still challenging problems to impose SGX into cloud computing. *The existing VMMs do not provide the efficient management operations for the SGX-enabled VMs*. Especially, live migration is essential to reduce the service downtime. Thus, the SGX-enabled VMs must be able to be migrated to another host. However, it is not a trivial issue because an enclave is encrypted with a unique key for the specific CPU. If the VMM migrates an enclave in the existing manner, another host cannot decrypt the enclave memory pages from the source host due to the mismatched key.

In this paper, we propose a novel approach and its implementation model to migrate the SGX-enabled VMs. The following is the contributions of our paper.

- We identify challenges in order to support live migration of the SGX-enabled VMs.
- We propose one approach and its implementation model to address the recognized challenges.

This paper is structured as follows: Section II describes the basic features of SGX for live migration. In Section III, we propose our approach and its implementation model. The conclusion and future works are presented in Section IV.

## II. Intel SGX

Intel SGX is the extended set of instructions that supports the *enclave* where the security sensitive code and data of the applications are protected by the SGX-enabled processor. Here, three features for the enclave migration are explained.

**1) MEE.** The Memory Encryption Engine (MEE) is a hardware unit to encrypt and integrity protect the traffic between the CPU and the DRAM. To protect the enclave page cache (EPC) that holds the enclave, the MEE leverages the cryptographic key, which is changed randomly and kept inside the MEE registers, and thus each machine has the different key.

**2) Remote Attestation.** Remote attestation is a method for remote entities to attest the trustworthiness of the trusted computing base, the totality of protection mechanisms that should be trusted [3]. In SGX, the hardware platform and enclaves are the target of remote attestation, and Intel provides the remote attestation enclave as the part of the SGX framework.

**3) EPC Page Swapping.** SGX supports the EPC page swapping for the VMM to securely evict EPC pages into the untrusted system memory. The EPC paging instructions are designed to maintain the same security properties (confidentiality, anti-replay, and integrity) with the PRM.

## III. Design

In this section, we identify challenges in live migration of the SGX-enabled VMs. To address them, we propose a novel approach and its implementation model.

### A. Challenging Problems

To migrate the SGX-enabled VMs, the VM's memory pages in the source host must be copied to the destination host during the pre-copy stage and the stop and copy stage (the original process in Figure 1). However, the enclave code and data that resides in the PRM cannot be copied as usual because SGX prevents the VMM from accessing the PRM directly. We considered utilizing the EPC page swapping to evict the enclave code and data. However, the evicted enclave cannot be decrypted by the destination host because the key for the EPC page swapping is unique and cannot leave the processor.

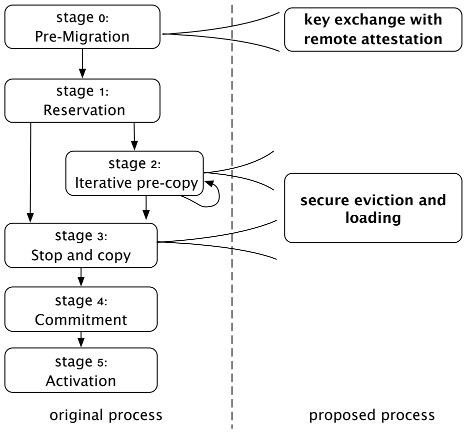

Fig. 1: Live migration of the SGX-enabled VMs in the original [4] and the proposed process.

### B. Live Migration of Enclave

To address the identified problems, we add new steps, *key exchange with remote attestation* and *secure eviction and loading*, into the original process as depicted in Figure 1.

**1) Key exchange with remote attestation.** The key agreement for live migration needs the key sharing between actual processors and integrity check of the participating enclaves. To this end, the key exchange with remote attestation is inserted into the pre-migration stage. The involved enclaves in two hosts execute mutual remote attestation to convince that the other host's SGX-enabled processors and enclave are trustworthy. It is checked whether the other host's enclave is not tampered and legitimate one. During this processes, the keying materials are exchanged, and the enclaves agree on the master secret.

**2) Secure eviction and loading.** The master secret is used to derive live migration keys (LMKs) and initial vectors (IVs), that are utilized to securely migrate the enclave in the iterative pre-copy stage and the stop and copy stage. During VM migration, the source host evicts the target enclave using LMKs when encountering the enclave. Then, the source host copies the evicted enclave to the destination host. Once the evicted enclave is copied, the destination host loads it using LMKs. This step supports the same security properties to the EPC. The source host encrypts and integrity protects the target enclave, and the destination host checks the integrity and decrypts it using LMKs with IVs (anti-replay).

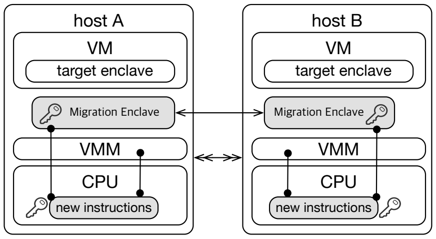

Fig. 2: Proposed implementation model; key exchange with remote attestation (⟷), evicted enclave copy (↞↠), and instruction execution (•—•).

### C. Implementation Model

To concrete our approach, we propose an implementation model for the enclave migration from host A to host B. A new external enclave called migration enclave (ME) agrees on the master secret. Like the EPC page swapping, the agreed master secret is stored in the MEE by a new instruction to make full reuse of the MEE. The VMMs on host A and host B execute new instructions for the secure eviction and loading with the derivations of LMKs and IVs using the master secret. Figure 2 shows the concept of this model, and the VMM is assumed to include the control software like Dom0 in Xen.

We considered that VMM vendors provide the live migration library and developers build their enclaves with this library to migrate the enclaves by themselves. However, if the enclave is not built with this library, live migration can be failed. Thus, if the SGX framework features ME and the SGX-enabled processor supports the new instructions for the enclave migration, the VMM executes live migration of the enclave without implementing live migration on the enclave.

## IV. Conclusion and Future Works

In this paper, we identify the challenges for live migration of the SGX-enabled VMs. We propose a novel approach and its implementation model to address the identified challenges.

This research is in progress, and we will study further to realize our implementation model on top of OpenSGX [5], an open source SGX emulator that emulates SGX instructions and provides operating components. To this end, we will design new instructions to load the master secret to the MEE and to execute the secure eviction and loading. We will design a migration protocol and formally verify it with the protocol verification language to assure that the communication is established between two actual hosts performing live migration.

## References

[1] "CVE-2015-3340: Information leak through XEN_DOMCTL_gettscinfo," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-3340.

[2] V. Costan and S. Devadas, "Intel SGX Explained," Cryptology ePrint Archive, Report 2016/086, 2016, http://eprint.iacr.org/.

[3] D. C. Latham, "Department of Defense trusted computer system evaluation criteria," *Department of Defense*, 1986.

[4] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live Migration of Virtual Machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 273–286.

[5] P. Jain, S. Desai, S. Kim, M.-W. Shih, J. Lee, C. Choi, Y. Shin, T. Kim, B. B. Kang, and D. Han, "OpenSGX: An Open Platform for SGX Research," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, Feb. 2016.

View publication stats