Enabling the Use of Strongly-Private Algorithms

Kubilay Ahmet Küçük
(PhD candidate, 4th year)
kucuk@cs.ox.ac.uk

Supervisor: Prof. Andrew MARTIN

Content from Journal Article, accepted on 03th May 2019. To be appear online in Jul/Aug 2019.
Enclave binary is inspectable by its’ nature

Questions or Contact: kucuk@cs.ox.ac.uk
Enclave-Aware or Enclave-Independent development

Algorithm Owner (AO)
Has Secret Algorithm or Business Logic

Hardware Owner (HO)
has Cloud Infrastructure for Remote Execution

Enclave Developer (ED)
Responsible for Secure Implementation

Approach 1: AO and ED on the same side
Protect the Secret-Code Before Release

AO
ED
Sending an Enclave-aware Algorithm

Ensuring the code secrecy before its release

Approach 2: HO and ED together
Protect the Secret-Code After Release

AO
ED
HO
Sending an Enclave-independent Algorithm

Maintaining the code secrecy after its release

Questions or Contact: kucuk@cs.ox.ac.uk
Enclave-Independent Private Algorithms
# Enclave-Aware Private Algorithms

<table>
<thead>
<tr>
<th>Action</th>
<th>State</th>
<th>Asset in Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ED Produces Asset 2 (A2)</td>
<td>State 2 (S2)  Standard Release Mode Creating the Asset 2 (A2)</td>
<td>A2: Enclave Library incl. TI, ECALLs, OCALLs, PIEF, PC Loader</td>
</tr>
<tr>
<td>(1.) ED sends the enclave binary A2 to HO</td>
<td>State 3 (S3) Before Execution Binaries are open for reverse engineering and inspection</td>
<td>A2 in Encrypted Enclave Memory</td>
</tr>
<tr>
<td>(2.) ED attests the enclave (Establishes Secure Channel Verifies the Enclave Identity)</td>
<td>State 4 (S4) At Runtime Binary is loaded into memory</td>
<td>Attested A2 in Encrypted Enclave Memory</td>
</tr>
<tr>
<td>(3.) AO sends the SSIEF</td>
<td>State 5 (S5) After Remote Attestation SSIEF is loaded into allocated memory via PC Loader</td>
<td>A1: Serialized Secret Internal Enclave Functions (SSIEF)</td>
</tr>
<tr>
<td>(4.) A2 extracts the A1 in HO’s environment at runtime</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New Possibilities with 3 Stake Holders

- Computational Power as a Service (CPaaS)
- Algorithm Querying as a Service (AQaaS)
- Data Querying as a Service (DQaaS)

Questions or Contact: kucuk@cs.ox.ac.uk
New Problems / Future Work

- Integrate Key sharing and separation (KSS)
- Modular Attestation
- Secure Erase
- Ownership Transfer
- ...

Contact: kucuk@cs.ox.ac.uk

Managing confidentiality leaks through private algorithms on Software Guard eXtensions (SGX) enclaves
Minimised TCB on secret-code execution with Early-Private Mode (EPM)

* Kubilay Ahmet Kıcı̇k̇1, David Grawrock2, and Andrew Martin1

1 kucuk@cs.ox.ac.uk, andrew.martin@cs.ox.ac.uk
Robert Hooke Building, Cyber Security Centre
Computer Science Department, University of Oxford

2 david.grawrock@twentyseven34.com
www.TwentySeven34.com

Abstract. Many applications are built upon private algorithms, and executing them in untrusted, remote environments poses confidentiality issues. To some extent, these problems can be addressed by ensuring the use of secure hardware in the execution environment; however, an insecure software-stack can only provide limited algorithm secrecy.

This paper aims to address this problem, by exploring the components of the Trusted Computing Base (TCB) in hardware-supported enclaves. First, we provide a taxonomy and give an extensive understanding of trade-offs during secure enclave development. Next, we present a case study on existing secret-code execution frameworks; which have bad TCB design due to processing secrets with commodity software in enclaves. This increased attack surface introduces additional footprints on memory that breaks the confidentiality guarantees; as a result, the private algorithms are leaked. Finally, we propose an alternative approach for remote secret-code execution of private algorithms. Our solution removes the potentially untrusted commodity software from the TCB, and provides a minimal loader for secret-code execution. Based on our new enclave development paradigm, we demonstrate three industrial templates for cloud applications: 1) computational power as a service, 2) algorithm querying as a service, and 3) data querying as a service.

Keywords: Trusted Computing Base (TCB) · Software Guard eXtensions (SGX) Enclave · Private Algorithms · Secret-Code Execution (SCE) · Algorithm Owner (AO) · Hardware Owner (HO) · Data Owner (DO) · Enclave Developer’s (ED) Responsibilities · Side-Channels · Early Private Mode (EPM) · Internal Enclave Functions (IEF) · Public Internal Enclave Functions (PIEF) · Serialised Secret Internal Enclave Functions (SSIEF)